Copyright Information

Harbour protection strategies with multiple autonomous marine vehicles

Gianluca Antonelli[1], Filippo Arrichiello[1], Giuseppe Casalino[2], Stefano Chiaverini[1], Alessandro Marino[3], Enrico Simetti[2], and Sandro Torelli[2]

[1] University of Cassino and Southern Lazio, Cassino, Italy
Corresponding Author email: f.arrichiello@unicas.it
[2] University of Genova, Genova, Italy
[3] University of Salerno, Salerno, Italy

Abstract. This paper presents the ongoing research activities of the Italian Interuniversity Center of Integrated Systems for the Marine Environment, ISME, in the field of harbour protection with autonomous marine vehicles. In particular, two different strategies have been developed in the recent years and have been extensively tested both in numerical simulations and in scale experiments. In the first case, a set of vehicles is positioned around an asset to be protected on the base of an optimization process of two cost functions, namely, the maximization of minimum interception distance and the minimization of maximum interception time. When an intruder is detected, an on-line optimization process selects, among the different vehicles, the one that exhibits the lowest estimated time to the menace. A motion planning algorithm with real-time obstacle avoidance is then used to drive the vehicle toward the intruder. In the second case, a team of vehicles is required to dynamically patrol a certain region by means of a decentralized control approach. The proposed solution is based on the merging of two concepts, the Voronoi tessellations and the Gaussian processes, and it allows robustness with respect to events as temporary communication or vehicle losses. It also exhibits characteristics of flexibility/scalability with respect to the number of team-mates.

Keywords. Autonomous marine vehicles, multi-vehicles system, coordinated control, harbour patrolling

1 Introduction

The problem of maintaining civilian harbours safeguarded against terroristic attacks, coming from the so-called “blue border” (i.e. the sea-side), is receiving an increasing interest in the recent years. In this context, the use of a team of “protecting” autonomous marine vehicles certainly represents a promising solution for reducing the harbour vulnerability. Indeed, under normal conditions, the vehicles can perform patrolling surveys of the more crucial waterways; instead, whenever a possible “menace” (i.e. an unauthorized vessel or a vessel moving in a suspect way) is detected, one vehicle can be used for “intercepting” the menace, allowing to determine whether the suspect vessel is “hostile” or “friend” without exposing humans directly to threats.

The Italian Interuniversity Center of Integrated Systems for the Marine Environment (ISME) is actively doing research, since more than 15 years, in the field of marine technologies and oceanic engineering. Among the different research interests, harbour protection by mean of multiple autonomous surface and underwater vehicles has recently taken an important role. Two different approaches to solve the latter problem have been developed and will be summarized in this paper.

In a first case, a set of vehicles is *silently* positioned around an asset to be protected. The positioning of the vehicles is obtained as the result of an optimization algorithm. The *optimal* vehicle is then selected in case of detection of an intruder and a real-time motion planning, capable of avoiding obstacles and current traffic, finally drives the vehicle toward the intruder ([16, 18]).

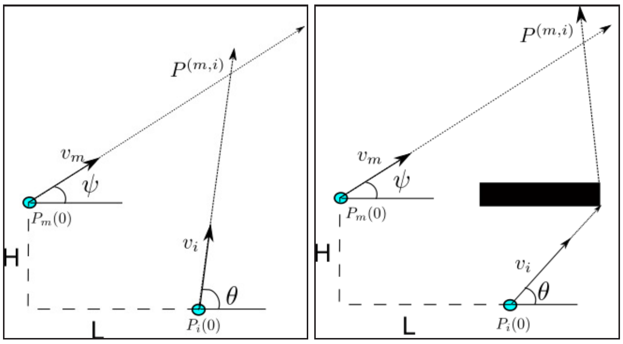
In the second case, a team of vehicles is required to patrol a certain region, i.e., to move *around* while properly collecting information. Patrolling is achieved with a sub-optimal solution satisfying the severe constraints we face in such a mission. Surface ([4], [13]) as well as underwater scenario ([12]) have been considered.

The above algorithms have been achieved in the framework of European FP7 projects such as Co3AUVs ([5]) and they have been tested via extensive numerical simulation as well as via experiments with marine vehicles.

2 Interception of suspect vessels with ASVs

Experiments with Autonomous Surface Vehicles (ASVs) are usually performed in open sea or in waterways in the absence of other unknown moving vessels. When dealing with harbour protection, however, the scenario is far different from those above and several crucial issues arise: the ship traffic is intense and the operations of tourist or merchant ships cannot be delayed or affected anyway by the security vehicles. Therefore the manoeuvres of ASVs must not perturb at all the normal harbour activities. Furthermore, there is a concrete risk of collision with other vessels, with consequent risks of personal injuries and property damages. Thus, the ASVs have to be provided with good path-following capabilities, since they need to follow a reference path with a certain accuracy. Moreover, they need reliable emergency sensory devices enabling a prompt detection of any incipient obstacle, and suitable techniques for implementing reactive obstacle avoidance capabilities, in case an unforeseen obstacle is actually detected.

In this perspective, DIST (University of Genova, Italy) and Selex Sistemi Integrati (a Finmeccanica Company, Italy), one of the international leading players in providing large systems for security and defence, are cooperating within an on-going joint research project, on the realization of the so-called Swarm Management Unit (SMU), a tool conceived for supervising the operations of a team of ASVs performing (semi-)autonomous surveillance activities within civilian harbours. Details on the SMU project are provided in [6, 17], together with a



(a) (b)

Fig. 1. The interception problem: **a**) without obstacles (**b**) with an obstacle

description of the levels of interaction between the SMU and the operator in charge of monitoring the ASVs activities.

Here we focuses on the problem of intercepting a detected menace before it could reach a particular "asset" (i.e. a crucial site to be maintained safeguarded) and of how to determine a-priori the better nominal positioning of the "interceptors" ASVs in order to increase their chances of success.

2.1 On-line selection of the interceptor

Once a menace is detected, the key factor for a prompt reaction is time. Therefore the better ASV for the interception is the one which is predicted to reach the menace in the shortest possible time. In the considered harbour scenario, all the other vessels in the area represent fixed or moving obstacles which have to be avoided by the interceptor ASV; as a consequence, for every ASV, the computation of its predicted time of interception is strictly related with the identification of its minimum-time path to reach the menace, which in turn depends on the contingent traffic situation.

A detailed description of the motion planner has been presented in [6, 17] and only its basic results are here reported for the reader's convenience. At first, the simplified problem depicted in Fig. 1 is first considered; i.e., given a single ASV, say the i-th, moving at its maximum speed v_i, find (if any) the ASV heading angle θ enabling the interception of a menace m, starting from a generic position $P_m = (-L, H)$ w.r.t. the ASV and moving at a constant speed v_m with a constant heading angle ψ.

By defining the motion of the i-th ASV as:

$$P_i(t) \triangleq \begin{bmatrix} x_i(t) \\ y_i(t) \end{bmatrix} = \begin{bmatrix} v_i cos(\theta)t \\ v_i sin(\theta)t \end{bmatrix} \tag{1}$$

and the motion of the menace as

$$P_m(t) \triangleq \begin{bmatrix} x_m(t) \\ y_m(t) \end{bmatrix} = \begin{bmatrix} v_m cos(\psi)t - L \\ v_m sin(\psi)t + H \end{bmatrix} \tag{2}$$

if no obstacles are located between the menace and the ASV (Fig. 1.a), the angle θ can be calculated as:

$$\theta = asin\left(\frac{v_m}{v_i} sin\left(\psi + \phi\right)\right) - \phi \tag{3}$$

clearly subject to

$$-1 \leq \frac{v_m}{v_i} sin\left(\psi + \phi\right) \leq 1 \tag{4}$$

where $\phi \triangleq arctan(-H/-L)$.

In case solution (3) exists, given the angle θ, the time needed for the interception can be calculated as:

$$t^{(m,i)} = \frac{-L}{v_i cos(\theta) - v_m cos(\psi)}, \quad L \neq 0 \tag{5}$$

while the point of the interception is obtained as:

$$P^{(m,i)} \triangleq \begin{bmatrix} P_x^{(m,i)} \\ P_y^{(m,i)} \end{bmatrix} = \begin{bmatrix} v_i cos(\theta) t^{(m,i)} \\ v_i sin(\theta) t^{(m,i)} \end{bmatrix} \tag{6}$$

where the notation $(\cdot)^{(m,i)}$ means that the quantity $(\cdot)$ refers to the i-th vehicle intercepting the menace m.

The more realistic situation where at least one obstacle (be it a static or a moving one) prevents the ASV from moving on a straight line is sketched in Fig. 1.b.

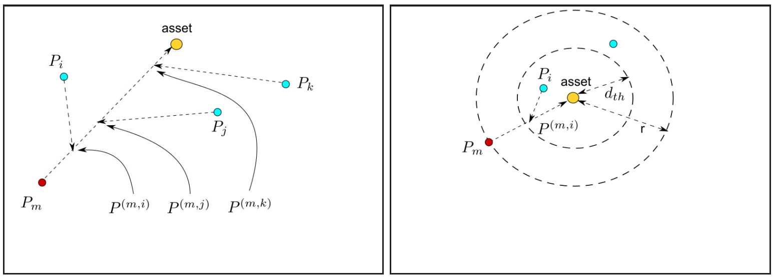
By now moving back the attention to the original problem of on-line selecting the better interceptor, consider Fig. 2, representing a situation where a menace is discovered at point P_m and is moving towards an asset, located at point $P_a = (x_a, y_a)$. A given number of vehicles $i = 1, \cdots, N$ are located in their respective positions P_i. As soon as the menace is detected, for every ASV, the predicted point of interception $P^{(m,i)}$ and related instant of interception $t^{(m,i)}$ is first of all calculated by the motion planner, as explained before. The problem of selecting the most suitable vehicle can then be simply stated as the following minimization problem:

$$\arg\min_i t^{(m,i)} \tag{7}$$

which can be easily on-line solved, given the availability of all the $t^{(m,i)}$ terms.

2.2 Off-line optimization of the ASVs position

The problem of better positioning the ASVs for achieving adequate levels of protection can be considered as a special instance of the so-called spatial resource-allocation problem, which has been studied for many years and has registered interesting results in particular in the field of fixed or mobile sensing networks (see, among the others, [9], [7] and references therein indicated). The proposed solution is based on the following two criteria. First of all, the ASVs must be



(a) (b)

Fig. 2. (**a**) Schema of the intercept problem (**b**) with a minimum distance of interception required

in the condition of *always* intercepting *any* menace before it can reach a certain security distance from the asset. As shown in the following, the adoption of such a primary criterion, other than reducing the chances of the menace to harm the asset, provides, as a by-product, an indication on the minimum number of "asset-protector" ASVs. All the (eventually available) other ASVs can then be employed for the secondary criterion, here stated in terms of reducing the maximum interception time.

Preventing menaces from getting too close to the asset The problem of guaranteeing that *any* menace can be *always* intercepted, before it gets too close to the asset, can be translated into a worst-case-scenario optimization problem. Indeed it has to be granted that, even if the menace is detected in the closest possible position to the asset (i.e. at a distance r), the ASVs are always in the condition of intercepting it on time (i.e. before the menace reaches a given security distance $d_{th} < r$, see Fig. 2b).

Let $P \triangleq \{P_1, \cdots, P_N\}$ denote the set of initial positions of a team of N ASVs, while be P_m the initial position of a detected menace m. For any given set of initial conditions (P_m, P), the distance between the interception point related to the i-th ASV and the asset can be easily computed as follows:

$$d_i\left(P_m, P\right) \triangleq \|P^{(m,i)} - P_a\| \tag{8}$$

It then follows that, for any given (P_m, P), the most suitable ASV for the interception is selected as:

$$i^o = \arg\max_{i|P_m,P} d_i\left(P_m, P\right) \tag{9}$$

whose corresponding distance between the interception point and the asset is

$$D\left(P_m, P\right) \triangleq \max_{i|P_m,P} d_i\left(P_m, P\right) \tag{10}$$

By now solving the above problem for any possible P_m (while still maintaining fixed the set P), the worst-case-scenario, i.e. the point P_m leading to the closest to the asset interception point, can be calculated as:

$$D_w\left(P\right) \triangleq \min_{P_m|P} D\left(P_m, P\right) \tag{11}$$

By finally letting the optimization variable P vary, the optimal positions of the ASVs are obtained as:

$$P^o \triangleq \arg\max_{P} D_w\left(P\right) \tag{12}$$

By combining all the above relationships, the following formulation of the original optimization problem is obtained:

$$P^o = \arg\max_{P} \left\{ \min_{P_m} \left[\max_{i} \left(\|P^{(m,i)} - P_a\| \right) \right] \right\} \tag{13}$$

whose corresponding distance from the interception point and the asset in the worst-case-scenario is clearly:

$$D^o = \max_{P} \left\{ \min_{P_m} \left[\max_{i} \left(\|P^{(m,i)} - P_a\| \right) \right] \right\} \tag{14}$$

In case D^o results lower than d_{th}, it means that the considered amount of ASVs is not sufficient to always guarantee the fulfillment of the security threshold distance, and a simple algorithm for determining the minimum number of ASVs required for protecting the asset can be applied solving the problem (14) for an increasing number of vehicles k.

Minimizing the maximum interception time In case the number of available vehicles N is greater than the number k of vehicles necessary to meet the required minimum distance of interception from the asset, the remaining $N-k$ ASVs can be exploited to solve another kind of optimization problem: minimizing the maximum interception time.

To better approach the problem, it is convenient to split the set of ASVs into two subsets: the first k vehicles with a fixed optimal positioning, as determined by the previous problem; and the remaining $N-k$ ones, whose set of positions $\hat{P} \triangleq \{P_{k+1}, \cdots, P_N\}$ is the subject of the here considered secondary optimization problem.

With these premises, for any given initial conditions (P_m, P), the most suitable ASV for the interception is now the one with the lowest interception time, that is:

$$i^o \triangleq \arg\min_{i|P_m,P} t^{(m,i)} \tag{15}$$

whose corresponding interception time is

$$T\left(P_m, P\right) \triangleq \min_{i|P_m,P} t^{(m,i)} \tag{16}$$

Then, by again considering the menace in all the allowed positions, the worst-case-scenario can be found as:

$$T_w\left(P\right) \triangleq \max_{P_m|P} T\left(P_m, P\right) \tag{17}$$

Therefore the optimal positions of the extra-ASVs can be found by minimizing the time of interception in the worst-case-scenario; that is:

$$\hat{P}^o \triangleq \arg\min_{\hat{P}} T_w\left(P\right) \tag{18}$$

Finally note that, since problem (17) considers all the possible P_m points, the extra ASVs are spread out, the farther away from the asset, the bigger the considered area is. The following more convenient formulation of problem (17) can therefore be made, by introducing a proper weighting function $0 \leq W\left(P_m\right) \leq 1$ expressing the "probability of detection" of a menace in any particular point:

$$T_w\left(P\right) \triangleq \max_{P_m|P} W(P_m)T\left(P_m, P\right) \tag{19}$$

In this way the points at the boundaries of the considered area could have a very low weight, as those inside the circle of radius r should have a zero weight.

By also considering the weighting function, the final formulation of the secondary optimization problem becomes:

$$\hat{P}^o = \arg\min_{\hat{P}} \left\{ \max_{P_m} \left[W(P_m) \min_i t^{(m,i)} \right] \right\} \tag{20}$$

2.3 Simulative results

Here we presents the results obtained when the scenario of Fig. 3 is used as area of operation. In the following pictures, the dark dot represents the asset to protect, the squares with black frame represent the position of the ASVs, while the smaller dots instead indicate that if a menace appears in such position.Moreover, the smaller circumference around the asset is a graphical representation of the minimum safety distance d_{th}, while the bigger one represent the minimum distance of detection r. Figures 4a and 4b show the paths of the ASVs when intercepting the candidate menace, which is represented with a circle.

The first simulation is depicted in Fig. 5. Between the two cases, only the detection distance r has been changed. When r is increased, the position of the ASVs can be further away from the asset, as r guarantees that the first detection of the menace cannot occur at ranges closer than it. Moreover, as it can be clearly seen, the ASVs are chosen to guard the entry points to the inner harbour area, as these points are choke points, where a surface menace must pass through if it wants to reach the asset.

The second simulation, presented through Fig. 6, shows what happens if the minimum required interception distance is increased. This increase in the required performance of the system imposes that two vehicles are now necessary to satisfy the first optimization problem. Such change will obviously decrease the performance of the system w.r.t. the secondary optimization problem.

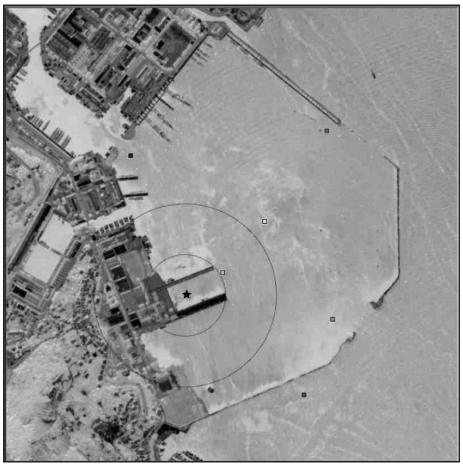
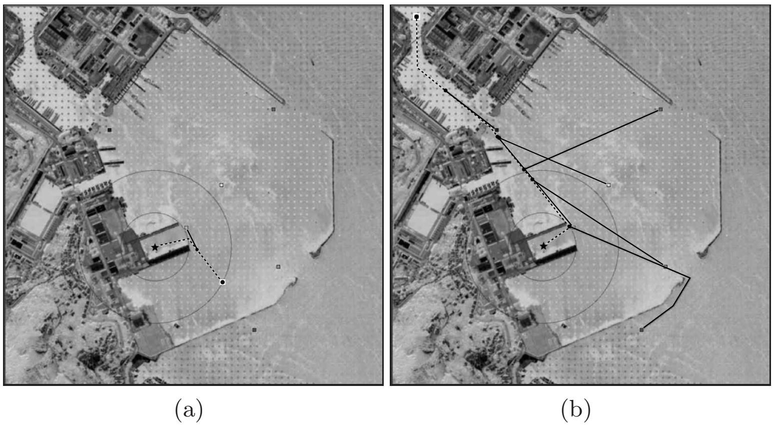


Fig. 3. Dark star: asset; Squares: ASV; Smaller circle: d_{th}; Bigger circle: r



(a)

(b)

Fig. 4. Different computed paths for the intercepting vehicles. (a) as the menace is detected just at the minimum distance r, the selected vehicle intercepts it before the minimum required distance d_{th}. (b) menace is detected far away (top left corner) and thus multiple vehicles can intercept it

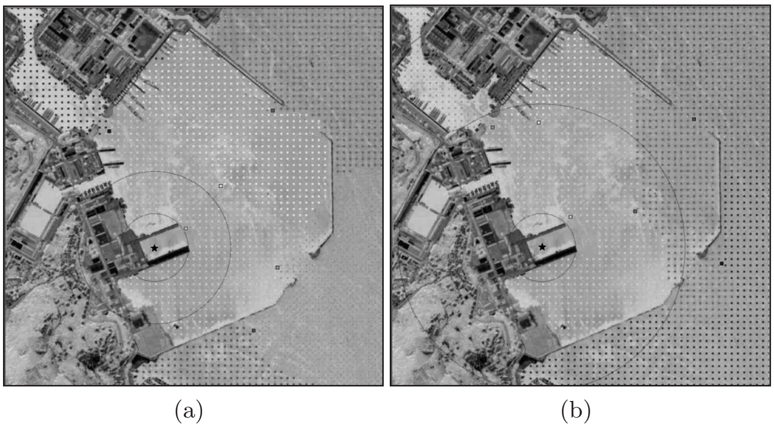


Fig. 5. Comparison by changing the detection distance r (a) $r = 400$, $d_{worst} = 214$, $t_{worst} = 22.93$, (b) $r = 750$, $d_{worst} = 373$, $t_{worst} = 22.62$

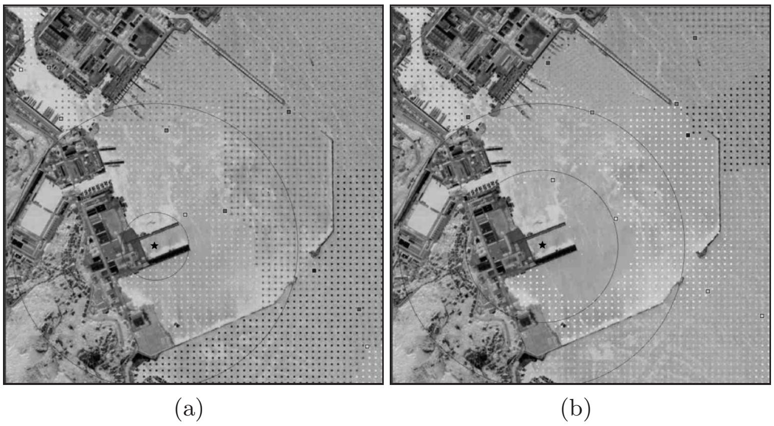


Fig. 6. Comparison by changing the minimum required interception distance d_{th} (a) $d_{th} = 178$, $d_{worst} = 374$, $t_{worst} = 20.64$, (b) $d_{th} = 400$, $d_{worst} = 460$, $t_{worst} = 20$

3 Dynamic patrolling

This section deals with the problem of dynamic patrolling with a team of marine vehicles. The patrolling task consists in traveling around an area, at regular intervals, in order to protect or supervise it [1]. Patrolling, thus, involves repeatedly visiting key locations within the working area, to assess environmental state with respect the presence of eventual intruders or any unexpected event. Despite of the fact that the patrolling tasks have been widely faced in literature from a theoretical point of view, only a few experiments have been carried out, especially in marine environment, without simplistic assumptions due to the existing technical problems. Indeed, the marine environment exhibits an additional challenge due to the extremely harsh conditions in which the vehicles need to operate.

For this reason, the problem at hand as been afforded at DIEI (University of Cassino, Italy) considering a list of realistic constraints in the development of a motion control solution for a team of autonomous vehicle that will be detailed in the following.

To the purpose, the developed strategy merges together two useful mathematical tools: Gaussian Processes and Voronoi tessellations. Given acquired samples, the Gaussian Processes [15] allow to predict the field at unknown location and to compute the uncertainty involved in that prediction [10]. Gaussian Processes allow us to address in the proposed algorithm in an elegant fashion the time and space variability, i.e., both a forgetting factor and the need to patrol more often certain regions. The Voronoi tessellations represents a subdivision of a set given a finite number of points [8, 14]. One of their main feature is that they can be calculated in a distributed way. Each vehicle, thus, is able to compute its Voronoi cell relying only on its exteroceptive sensors and/or communication capabilities.

3.1 Problem description

The problem at hand is characterized by challenging theoretical as well as implementation issues. Strongly motivated by the need to perform experimental validation of the derived algorithm, the follwing constraints have been considered:

- Coordination. Robotic missions such as the one addressed in this paper are more efficient by means of a coordinated, multi-robot strategy;
- Decentralization. One central computational unit represents a weak point for a multi-robot algorithm. This is particularly significant when a security application, such as patrolling, is considered;
- Robustness. When robots move around in the real world, they are necessarily confronted with a number of unexpected events that may seriously jeopardize the success of their missions. It is not realistic to design a multi-robot algorithm that is not *robust*, in a wide sense, with respect to the possibility that one or more robots simply stop functioning, or hold in place, or that the communication among them may experience temporary black outs;

- Scalability. As a scalability constrain we want that the computational burden associated to each robot does not change with the number of robots;
- Communications. Different communication technologies (e.g. for surface or underwater communication) come with different bandwidths and ranges that directly impact on the performance achievable with multiple vehicle patrolling algorithms. A reliable algorithm must be customizable with respect to the available communication bandwidth;
- Real-time. Each robot needs to take *decision* in real-time, thus preventing the use of off-line planning algorithms;
- In view of practical implementation, additional features such as, for example, obstacle avoidance policies, need to be considered.

These constraints are fundamental for experiments in a real scenario and, differently form other solutions, are naturally taken into account by the designed solution.

We now describe the patrolling problem addressed in this paper: consider a region $\mathcal{A} \in \mathbb{R}^l$, $l = 2, 3$ and a function $y = f(\boldsymbol{x}, t), \quad \boldsymbol{x} \in \mathcal{A}$, with $f : \mathbb{R}^l \times [0, \infty) \to \mathbb{R}_{0+}$, where f is application dependent. For example, in the case of patrolling/security applications y represents the level of safety of the environment at point $\boldsymbol{x}$ and at time t.

The function f exhibits a spatial correlation that is mainly affected by the nature of the underlying phenomenon under study and/or the sensor suite used. It is also important to stress that the function may be time-varying, at a scale that once again is determined by the phenomenon under investigation.

Our main goal is to develop a strategy to estimate the function f by taking appropriate measurements using robots equipped with sensor suites. In what follows, we let N_r denote the number of robots and $x_{r,i}(t) \in \mathbb{R}^l$, $i = 1, 2, \ldots, Nr$ the position of robot i at time t. In addition, each robot is assumed to be able to sense or receive the position of some neighbors, where the term neighbor indicates a robot $\boldsymbol{x}_{r,j}$ that is close to $\boldsymbol{x}_{r,i}$ with respect to a certain metric (for example the Euclidean distance).

The patrolling task shares several aspects with sampling. In particular, at given instant the knowledge about the safety status of a location in the area depends on the team configuration and, therefore, on the robots' positions $\boldsymbol{x}_{r,i}$, $\forall i = 1, 2, \ldots, N_r$. Thus, given a robot with position $\boldsymbol{x}_{r,i}$, it is possible to state whether this position is safe or not based on the value of function f at this point; it can be argued that this information can be used to infer the status at other locations in the neighbourhood (*spatial* dependence). An example is represented by an intruder or a toxic substance spill at location $x_{r,i}$. Finally, it can be also argued that in a dynamic scenario, a location that has been marked as safe (because it was visited in the past) but has not been visited by any of the patrolling robots for a certain amount of time should no longer be considered as safe; instead, a high uncertainty should be associated to its status. Thus, high uncertainty must be associated not only to those cells that have remained unvisited but also to the cells that were visited but long back in time (*time* dependence). An example is represented by a moving intruder, a moving oil spill, or a changing temperature field. The aim is, therefore, to estimate the map function $f(\boldsymbol{x}, t)$ by reducing the uncertainty in its knowledge; namely, by bringing the robots toward those locations characterized by a high degree of uncertainty. It is assumed that a robot will be able to measure the degree of safety, f, of a location by means of some sensor as, for example, a vision sensor. Moreover, it should be clear that the developed strategy is suitable both for patrolling and sampling as shown in [4]. In addition, the use of multi-robot systems requires a coordination mechanism among robots. Specifically a Voronoi tessellation is used for both distributing the calculus of the function f and coordinating the motion of robots. A Gaussian process strategy is, instead, used to predict function f.

3.2 The Voronoi partition

Voronoi partitions (or diagrams) are subdivisions of a set $\mathcal{D}$ characterized by a metric with respect to a finite number of seed points belonging to that set.

Assuming that at the current time t the seed points are the robots' positions $\{\boldsymbol{x}_{r,1}, \boldsymbol{x}_{r,2}, \ldots, \boldsymbol{x}_{r,N_r}\}$, the corresponding N_r Voronoi cells, $Vor(\boldsymbol{x}_{r,i})$, $i = 1, 2, \ldots, N_r$ are given by

$$Vor(\boldsymbol{x}_{r,i}) = \{\boldsymbol{x} \in \mathcal{D} \quad | \quad \|\boldsymbol{x} - \boldsymbol{x}_{r,i}\| \leq \|\boldsymbol{x} - \boldsymbol{x}_{r,j}\|, \forall j\} .$$

The union of the Voronoi cells gives back the entire set and the intersection of two cells is always empty. The most important property of the Voroni tool for the use on decentralized robotics is that each robot can compute its own cell by applying a *local* algorithm, i.e., by simply knowing its position and the *neighbors*' positions, either by direct sensing or by communication. An example is reported in Figure 7 where the Voronoi tessellation of a three-dimensional set has been generated according to three randomly generated seed points. Further details on the Voronoi-based theory and its applications can be found in [8] or [14].

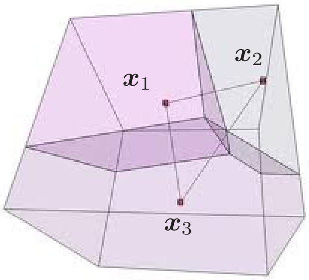


Fig. 7. Example of a Voronoi partition for 3 points in a 3D set.

3.3 The Gaussian Processes

A Gaussian process is a collection of random variables, any finite number of which have a joint Gaussian distribution. One of the key features of Gaussian Processes is their potential to yield methods to predict the value of a function at any location, given a set of previously collected observations (either in space or in time), with an explicit representation of the uncertainty of that prediction. For this reason, they will be used as a means to estimate the field f. What relates one observation to another in such cases is just the covariance function. In what follows we summarize the key facts about Gaussian Processes needed to understand the method that we propose. A comprehensive exposition of the theory can be found in [15].

We view at function $f(\boldsymbol{x}, t)$ as a zero-mean spatio-temporal Gaussian Process

$$f(\boldsymbol{x}) \sim \mathcal{GP}(0, \mathcal{K}(\boldsymbol{x}_1,\, t_1; \boldsymbol{x}_2,\, t_2) \tag{21}$$

where $\mathcal{K}(\boldsymbol{x}_1,\, t_1; \boldsymbol{x}_2,\, t_2)$ is the covariance function. We will assume that the covariance function $\mathcal{K}$ is generically defined as

$$\mathcal{K}(\boldsymbol{x}_1,\, t_1; \boldsymbol{x}_2,\, t_2) = C(\|\boldsymbol{x}_2 - \boldsymbol{x}_1\|\,, |t_2 - t_1|) \tag{22}$$

with $C : \mathbb{R}_0^+ \times \mathbb{R}_0^+ \to \mathbb{R}^+$. Notice that both space and time are taken into account to handle also the non stationary case. For the sake of simplicity, in equation (22) we assume that the process is homogeneous, second order stationary and isotropic, which basically implies that the covariance only depends on the distance between two generic points $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ and on the absolute value of the time difference $t_2 - t_1$.

Given the set $S = \{(\boldsymbol{x}_1, t_1), (\boldsymbol{x}_2, t_2), \ldots, (\boldsymbol{x}_n, t_n)\}$ made of pair of locations $\boldsymbol{x}_i \in \mathcal{A}$ and instants of time t_i and the corresponding vector of observation $\boldsymbol{y} \in \mathbb{R}^n$, the symbol $\Sigma_S \in \mathbb{R}^{n \times n}$ represents the symmetric non-negative covariance matrix whose elements (i, j) is $\mathcal{K}(\boldsymbol{x}_i,\, t_i; \boldsymbol{x}_j,\, t_j)$.

Moreover, given a single element $(\boldsymbol{x}^*,\, t)$ and the set S, $\boldsymbol{\sigma}_{Sx}(\boldsymbol{x}^*, t) \in \mathbb{R}^n$ is a column vector whose i-th element is $\mathcal{K}(\boldsymbol{x}^\star,\, t; \boldsymbol{x}_j,\, t_j)$.

The objective is to predict $y^* = f(\boldsymbol{x}^*, t)$ at the generic location $\boldsymbol{x}^*$ and at the current time instant t based on the vector of observations $\boldsymbol{y}$. In the case of a multivariate normal distribution over a set S of random variables associated with n pairs of positions and time instants, the posterior distribution of y^* is characterized by a normal distribution $y^* | \boldsymbol{y} \sim \mathcal{N}(\hat{\mu}, \hat{\Sigma})$ with [15]:

$$\hat{\mu} = \boldsymbol{\sigma}_{Sx}(\boldsymbol{x}, t)^{\mathrm{T}} \boldsymbol{\Sigma}_S^{-1} \boldsymbol{y} \tag{23}$$

$$\hat{\Sigma} = \mathcal{K}(\boldsymbol{x}^*, t; \boldsymbol{x}^*, t) - \boldsymbol{\sigma}_{Sx}(\boldsymbol{x}^*, t)^{\mathrm{T}} \boldsymbol{\Sigma}_S^{-1} \boldsymbol{\sigma}_{Sx}(\boldsymbol{x}^*, t). \tag{24}$$

The best estimate of y^* is given by (23) and the uncertainty of the estimation is captured by its variance, described in (24). Thus, while the predicted value is useful for establishing the most likely appearance of the function f based on the available sensor data, it can also be misleading if considered in isolation. One of the key advantages of Gaussian Processes is, therefore, the possibility to compute the variance of each prediction.

In the considered problem, the i th robot measures the status of location $\boldsymbol{x}$ using dedicated sensors. We assume that the measurement made by robot i at position $\boldsymbol{x}$ and time t is given by

$$y = f(\boldsymbol{x},\, t) + w_i,$$

where $w_i \sim \mathcal{N}(0, \sigma_i)$ is a white noise Gaussian Process with zero mean and standard deviation σ_i. For simplicity, we assume that $w_i = w \sim \mathcal{N}(0, \sigma)$, i.e., the robots are equipped with identical sensors. In this case, equation (23) becomes

$$\hat{\mu} = {\boldsymbol{\Sigma}_{Sx}}^{\mathrm{T}} \left(\boldsymbol{\Sigma}_S + \sigma^2 \boldsymbol{I}\right)^{-1} \boldsymbol{y} \tag{25a}$$

$$\hat{\Sigma} = \mathcal{K}(\boldsymbol{x}, t;\, \boldsymbol{x}, t) - \boldsymbol{\sigma}_{Sx}(\boldsymbol{x}, t)^{\mathrm{T}} \left(\boldsymbol{\Sigma}_S + \sigma^2 \boldsymbol{I}\right)^{-1} \boldsymbol{\sigma}_{Sx}, \tag{25b}$$

where $\boldsymbol{I}$ is the identity matrix of proper dimensions.

In the above equations, the matrices $\boldsymbol{\Sigma}_S$ and $\boldsymbol{\sigma}_{Sx}$ are completely defined once the function C in equation (22) has been specified. According to [15] and [19], one possible choice for this function is the Square Exponential Covariance Function:

$$C(\|\boldsymbol{x}_2 - \boldsymbol{x}_1\|, |t_2 - t_1|) = \phi^2 e^{-\dfrac{\|\boldsymbol{x}_2 - \boldsymbol{x}_1\|^2}{2\tau_s^2} - \dfrac{(t_2 - t_1)^2}{2\tau_t^2}}, \tag{26}$$

where ϕ is a weighting scalar parameter that will be selected to be unitary and the parameters τ_s and τ_t are positive scalars used to affect the space and time scales, respectively. In addition, it is worth noticing that the choice in equation (26) refers to isotropic domains and known constant τ_s and τ_t.

3.4 Application to the patrolling mission

From equation (24), the variance of the estimation at position $\boldsymbol{x}$ given the already acquired samples and the current time t takes the form

$$\hat{\Sigma}(\boldsymbol{x}) = C(0, 0) - {\boldsymbol{\sigma}_{Sx}}^{\mathrm{T}} \boldsymbol{\Sigma}_S^{-1} \boldsymbol{\sigma}_{Sx}. \tag{27}$$

It turns out that minimizing the uncertainty, which corresponds to minimizing the positive definite right-hand side of equation (27), is the same as maximizing (given the available degrees of freedom) the function

$$\xi_S(\boldsymbol{x}) = {\boldsymbol{\sigma}_{Sx}}^{\mathrm{T}} \boldsymbol{\Sigma}_S^{-1} \boldsymbol{\sigma}_{Sx}. \tag{28}$$

Note also that due to the time dependency of the covariance function in (26), a point that has been visited too far in the past (with respect to the time parameter τ_t) is candidate to be visited again. This feature is exploited by assigning proper time constants according to the applications.

It is interesting to reproduce graphically equation (28) for some case studies. In the simple case we consider, the region of interest is a planar square with unitary length. Three location have been visited, $S = \{(\boldsymbol{x}_1, t_1), (\boldsymbol{x}_2, t_2), (\boldsymbol{x}_3, t_3)\}$, with $\boldsymbol{x}_1 = \begin{bmatrix}0\ 0\end{bmatrix}^{\mathrm{T}}$, $\boldsymbol{x}_2 = \begin{bmatrix}1\ 0\end{bmatrix}^{\mathrm{T}}$, and $\boldsymbol{x}_3 = \begin{bmatrix}0.5\ 1\end{bmatrix}^{\mathrm{T}}$ at time t_1, t_2 and t_3, respectively.

- First example (Figure 8 (top left)): $t_1 = t_2 = t_3 = 0$, current time $t = 0$, $\tau_s = 0.5$, $\tau_t = 8$.
- Second example (Figure 8 (top right)): $t_1 = t_2 = t_3 = 0$, current time $t = 0$, $\tau_s = 0.2$, $\tau_t = 8$.
- Third example (Figure 8 (bottom)): $t_1 = 0$, $t_2 = 7$, $t_3 = 10$, current time $t = 10$, $\tau_s = 0.5$, $\tau_t = 8$.

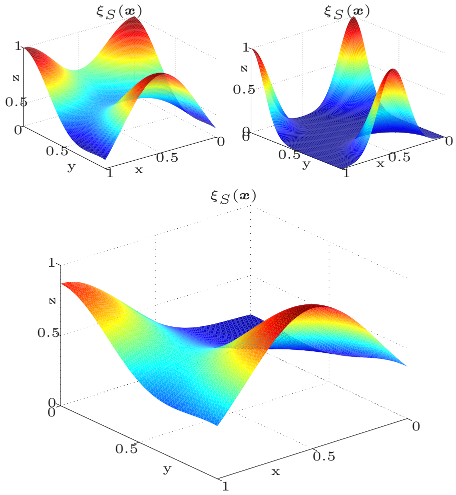


Fig. 8. Graph of the function (28). Top left: first example. Top right: second example. Bottom: third example.

In the classical sampling task as in [11, 19, 20] the field to sample is related to a physical variable such as temperature, salinity, etc.; the *hyper-parameters*, ϕ, τ_s, τ_t, are usually identified or learned by training. On the other hand, in the case of patrolling a different point of view can be used: the hyper-parameters, in particular the constant τ_s and τ_t, are not related to any physical phenomenon. On the contrary, they are decided by the user and, then, known beforehand by the robots. Their value may be chosen so as to confer to the system specific behaviors. As examples of the above concept, let us consider the following cases:

- a low value of the the parameter τ_s in equation (26) implies low *space* correlation between different locations. On the contrary, a larger value implies high correlation between even far locations. Such a feature may be used to take into account the range of robot' sensors. In addition, τ_s can be function of the location (i.e., $\tau_s = \tau_s(\boldsymbol{x})$); such a feature can be useful, e.g., for modeling different visibility conditions;
- a low value of the parameter τ_t in equation (26) implies low *time* correlation between cells. This means that the patrolling team needs to visit each

position more frequently. Also in this case, the τ_t can be functions of the location (i.e., $\tau_t = \tau_t(\boldsymbol{x})$), in order to take into account the case of environments where some locations are more exposed to unexpected events than others (eg., in an harbor patrolling scenario it may be required to visit more frequently the harbour entry).

- by setting $\tau_t = \infty$ a static field is obtained as in the case of a coverage mission.

3.5 Proposed coordination strategy

Figure 9 illustrates the proposed control architecture for a single robot. At the top level, the planner is in charge of deciding the robot trajectory. At the lower level, the Null-Space-based-Behavioral (NSB) control approach allows following reference trajectory provided by the upper level, while properly handling unexpected events such as the presence of obstacles. The NSB has been widely used by some of the authors and its description will not be given here to avoid repetion; the interested reader will find in [2], [3], the details of this strategy and its properties.

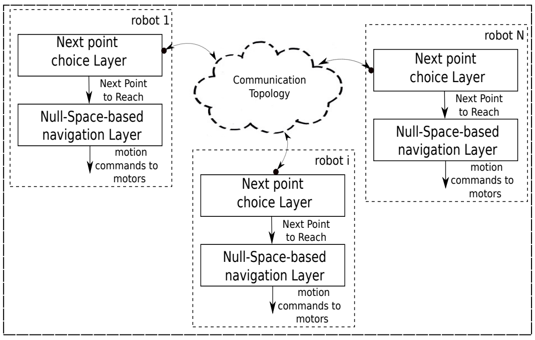

robot 1
Next point choice Layer
Next Point to Reach
Null-Space-based navigation Layer
motion commands to motors
Communication Topology
robot N
Next point choice Layer
Next Point to Reach
Null-Space-based navigation Layer
motion commands to motors
robot i
Next point choice Layer
Next Point to Reach
Null-Space-based navigation Layer
motion commands to motors

Fig. 9. Control architecture.

3.6 The top level: algorithm description

The top layer in Figure 9 represents the *core* of the proposed algorithm, i.e., the block in charge of computing the next point to be visited by each robot. As described in the previous Sections, our objective is the maximization of the function $\xi_S(\boldsymbol{x})$ expressed by equation (28). To this effect, a partition of the area $\mathcal{A}$ according to the Voronoi tessellations is computed, and each robot performs such a maximization in its own cell. The strategy designed for each robot is given as: **Algorithm**
loop

```
1. exchange data with the neighbors
2. build its own cell: $Vor(\boldsymbol{x}_{r,i})$
3. select next point $\overline{\boldsymbol{x}}_i$ in its own Voronoi cell, and move to that point
4. send $\overline{\boldsymbol{x}}_i$ to the NSB layer
end loop
```

To select the next point to, we propose the following strategy. Let $\boldsymbol{x} = \boldsymbol{h}(s) = \boldsymbol{x}_{r,i} + (\boldsymbol{x}_u - \boldsymbol{x}_{r,i})s$ be a parametrization of the line segment joining the actual position $\boldsymbol{x}_{r,i}$ of the robot and a generic point $\boldsymbol{x}_u \in S_u$, with $s \in [0,\ 1]$; the next target in S_u is determined by

$$\overline{\boldsymbol{x}}_i = \min_{\boldsymbol{x}_u \in S_u} \frac{\int_0^1 \xi_S(\boldsymbol{h}(s))ds}{\|\boldsymbol{x}_{r,i} - \boldsymbol{x}_u\|}. \tag{29}$$

The heuristics behind the strategy (29) is that, among the unvisited points in the set S_u, the one characterized by the most unvisited path (normalized by the path length) is chosen. Another possible strategy would be to simply drive the robot toward the point of global minimum of $\xi_S(\boldsymbol{x})$ inside its Voronoi cell, that is, compute

$$\overline{\boldsymbol{x}}_i = \min_{\boldsymbol{x} \in Vor(\boldsymbol{x}_{r,i})} \xi_S(\boldsymbol{x}). \tag{30}$$

As an example, Figure 10 shows a comparison between the strategy in (29) and (30) in a 50 m×100 m rectangular environment with three vehicles. The space (τ_s) and time (τ_t) constants in equation (26) are 4.7 m and 400 s, respectively. In particular, the performance index adopted is represented by the integral of $\xi_S(\boldsymbol{x})$ over the environment normalized by its area. It is worth noticing that, as $\xi_S(\boldsymbol{x}) \in\ [0, 1]$, the considered index belongs to the same interval. Because of the meaning of function $\xi_S(\boldsymbol{x})$,

- a performance index equal to 1 means that the field is completely known;
- a performance index equal to 0 means that the field is completely unknown;
- a performance index equal to 1 can be asymptotically reached only in the case of static fields;
- lower values of the time constant τ_t imply lower values of the index at steady-state.

3.7 Experimental results

Due to lack of space, only the surface experiments will be briefly reported, the underwater implementation is described in [12]. Video of the underwater as well as the surface experiments can be downloaded from `webuser.unicas.it/lai/robotica/video.html`.

The *surface* experiments were performed in July 2011 at the Parque Expo site, Lisbon, PT, using three Medusa autonomous surface robots designed and built by the marine robotics team of IST/ISR (Instituto Superior Técnico/Institute

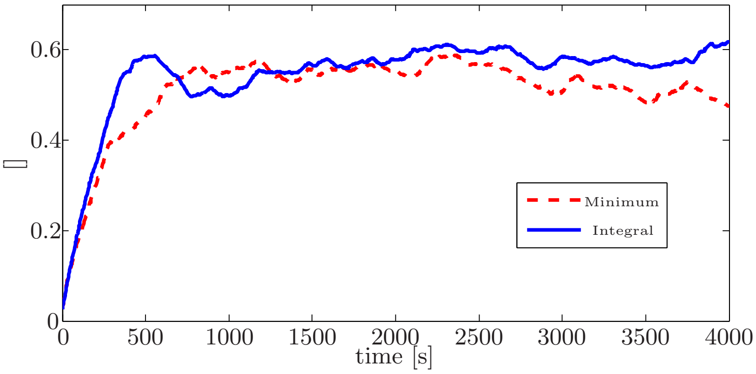


Fig. 10. Normalized integral of function $\xi_S(\boldsymbol{x})$ in a comparison between strategy in equation (29) (continuous line) and equation (30) (dashed line).

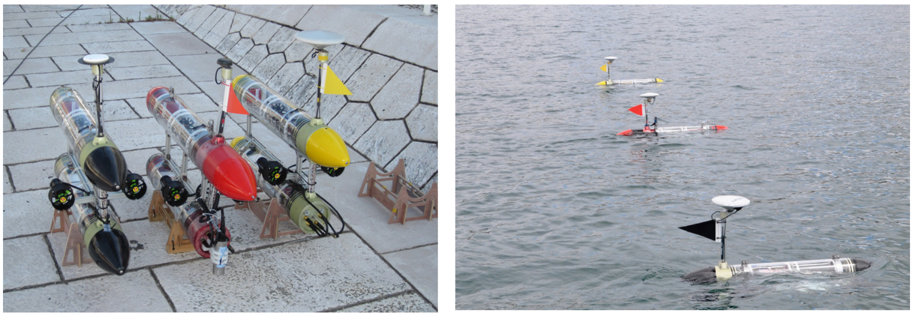


Fig. 11. The three Medusa surface robots setup (the red, black and yellow robots).

for Systems and Robotics) and shown in Figure 11 (red, black, and yellow robots).

Figure 12 shows the map of the site, with the robots moving in a 60 m×70 m rectangular map. An obstacle consisting of a buoy is placed inside the patrolling region (the dot in Figure 12). The position of the buoy is fixed and known in advance by the robots. The maximum robots speed was limited to 0.7 m/s. In what follows, we summarize the results of an experiment that run for approximately 1 hour.

The robots exchanged information via WI-FI network. The τ_s and τ_t parameters were set to 3.7 m and 200 s, respectively; the constant θ in Section 3.6 was set to 0.5.

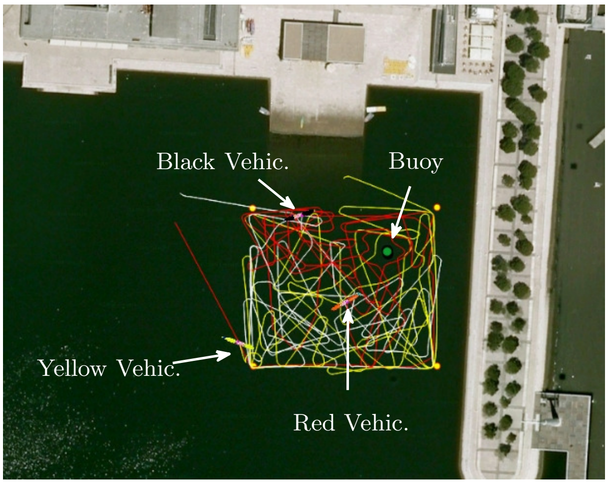


Fig. 12. The map of the Parque Expo site in Lisbon with the paths described by the robots in a typical experiment. The robots (red, black and yellow) are restricted to move in a 60 m×70 m rectangular environment.

In Figure 13, the sequence of steps performed by the robots is shown. In each frame, on the left are shown the Voronoi cells with the robots and the current targets (big bullets), while, on the right, the plot of function (28) is shown. The red color is representative of higher values of the function while the blue color of lower ones. Focusing the attention on the black robot, the following steps are shown:

1. the robot moves toward the current target as generated by the algorithm described in Section 3.5,
2. the robot reaches the current target,
3. the robot chooses the next target inside its own Voronoi cell,
4. the robot moves toward the new target.

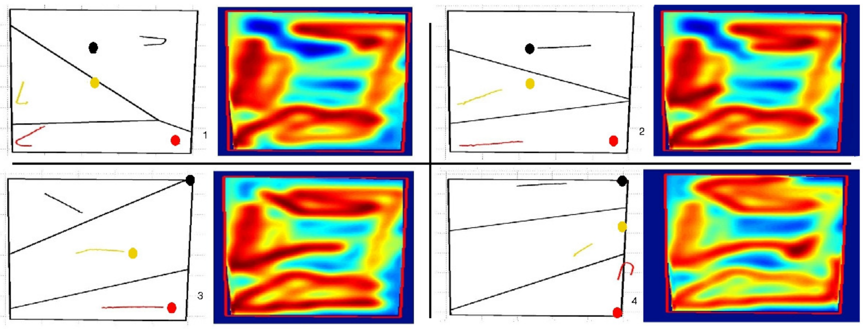


Fig. 13. Frames of the experiments. In each frame, on the left are shown the Voronoi cells with the robots and the current targets (big bullets) while, on the right, the plot function $\xi_S(\boldsymbol{x})$ in equation (28) is shown (red color is representative of high values of the function, blue color of low values).

4 Conclusions

In this paper we presented the latest research of the Italian Interuniversity Center ISME in the field of harbour protection using autonomous marine vehicles. Specifically, we at first presented a solution for the displacement of a fleet of vehicles by optimizing either the interception distance or the interception time to a menace. Then, we presented a decentralized control approach, based on Voronoi tessellations and Gaussian processes, to make a fleet of vehicles dynamically patrol a given area. The approaches have been validated via numerical simulations and experiment with autonomous surface vessels.

References

1. Almeida, A., Ramalho, G., Santana, H., Tedesco, P., Menezes, T., Corruble, V.: Recent advances on multi-agent patrolling. Proceedings of the Brazilian Symposium on Artificial Intelligence (2004)
2. Antonelli, G.: Stability analysis for prioritized closed-loop inverse kinematic algorithms for redundant robotic systems. IEEE Transactions on Robotics 25(5), 985–994 (October 2009)
3. Antonelli, G., Arrichiello, F., Chiaverini, S.: The Null-Space-based Behavioral control for autonomous robotic systems. Journal of Intelligent Service Robotics 1(1), 27–39 (Jan 2008)
4. Antonelli, G., Chiaverini, S., Marino, A.: A coordination strategy for multi-robot sampling of dynamic fields. In: Proceedings 2012 IEEE International Conference on Robotics and Automation. pp. 1113–1118. St Paul, MN (May 2012)

5. Birk, A., Pascoal, A., Antonelli, G., Caiti, A., Casalino, G., Caffaz, A.: Cooperative cognitive control for autonomous underwater vehicles (CO3AUVs): overview and progresses in the 3rd project year. In: IFAC Workshop on Navigation, Guidance and Control of Underwater Vehicles - NGCUV'12 (2012), porto, PT
6. Casalino, G., Turetta, A., Simetti, E.: A three-layered architecture for real time path planning and obstacle avoidance for surveillance USVs operating in harbour fields. In: OCEANS 09. Bremen, Germany (May 2009)
7. Cortés, J., Martínez, S., Karatas, T., Bullo, F.: Coverage control for mobile sensing networks. IEEE Transactions on Robotics an Automation 20(2), 243–255 (2004)
8. Du, Q., Faber, V., Gunzburger, M.: Centroidal Voronoi tessellations: applications and algorithms. SIAM review 41(4), 637–676 (1999)
9. Huang, C.F., Tseng, Y.C.: The coverage problem in a wireless sensor network. Mobile Networks and Applications 10, 519–528 (2005)
10. Krause, A., Gestrin, C., Gupta, A., Kleinberg, J.: Near-optimal sensor placements: Maximizing information while minimizing communication cost. In: IPSN 06. Nashville, Tennessee, USA (April, 19-21 2006)
11. Krause, A., Guestrin, C.: Nonmyopic active learning of Gaussian processes: An exploration-exploitation approach. In: International Conference on Machine Learning. pp. 449–456 (2007)
12. Marino, A., Antonelli, G.: Experimental results of coordinated coverage by autonomous underwater vehicles. In: Proceedings 2013 IEEE International Conference on Robotics and Automation. pp. 4126–4131. Karlsruhe, D (May 2013)
13. Marino, A., Antonelli, G., Aguiar, A., Pascoal, A.: Multi-robot harbor patrolling: a probabilistic approach. In: 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems. Vilamoura, PT (October 2012)
14. Okabe, A., Boots, B., Sugihara, K.: Spatial tessellations: Concepts and applications of Voronoi diagrams. John Wiley & Sons (1992)
15. Rasmussen, C.E., Williams, C.: Gaussian processes for machine learning. MIT Press (2006)
16. Simetti, E., Turetta, A., Casalino, G.: USV-based security system for civilian harbors. Sea Technology 51(11), 41–43 (2010)
17. Simetti, E., Turetta, A., Casalino, G., Storti, E., Cresta, M.: Towards the use of a team of USVs for civilian harbour protection: Real time path planning with avoidance of multiple moving obstacles. In: IEEE IROS09 3rd Workshop on Planning, Perception and Navigation for Intelligent Vehicles. St. Louis, MO, US (October 2009)
18. Simetti, E., Turetta, A., Torelli, S., Casalino, G.: Civilian harbour protection: Interception of suspect vessels with unmanned surface vehicles. In: 9th IFAC Conference on Manoeuvring and Control of Marine Craft (MCMC 2012) (September 2012)
19. Xu, Y., Choi, J.: Mobile sensor networks for learning anisotropic Gaussian processes. In: American Control Conference. St. Louis, MO, USA (June, 10-12 2009)
20. Xu, Y., Choi, J.: Adaptive sampling for learning Gaussian processes using mobile sensor networks. Sensors 11(3), 3051–3066 (2011)