Coordination of unmanned marine vehicles for asymmetric threats protection

Simone Nardi[1,2], Cosimo Della Santina[2], Daniele Meucci[3], and Lucia Pallottino[2,3]

[1]Department of Mathematics, University of Pisa, Italy.
[2]Research Center “E. Piaggio”, University of Pisa, Italy.
[3]Department of Information Engineering, University of Pisa, Italy.

***Abstract*— A coordination protocol for systems of unmanned marine vehicles is proposed for protection against asymmetric threats. The problem is first modelled in a game theoretic framework, as a potential game. Then an extension of existing learning algorithms is proposed to address the problem of tracking the possibly moving threat. The approach is evaluated in scenarios of different geometric complexity such as open sea, bay, and harbours. Performance of the approach is evaluated in terms of a security index that will allow us to obtain a tool for team sizing. The tool provides the minimum number of marine robots to be used in the system, given a desired security level to be guaranteed and the maximum threat velocity.**

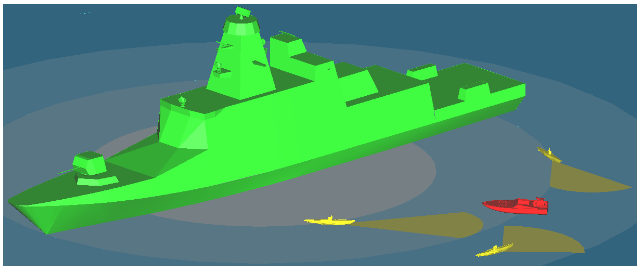


Fig. 1: Example of an asymmetric threat detected by a team of marine autonomous robots. The team of robots must efficiently monitor the area around the ship.

I. Introduction

The problem of detecting and accordingly reacting to an asymmetric threat[1] in marine environments is a challenge both from research and technological points of view [1].

Surveillance sensors, currently available on naval platforms, have reached a high level of performance in nominal working conditions [2]. However, they remain insufficient in real–world applications [3], [4] (e.g. in case of obstruction to sensor systems line–of–sight, or in case of adverse weather conditions) leading to a drastically reduced available time–to–reaction that may increase the possibility of human errors, especially in stressful situations.

The goal of innovative surveillance systems is hence to guarantee an adequate supervised area in any working conditions. In this paper, we tackle such area monitoring problem assuming that every region of the environment can not be under the robots sensor footprint (i.e. monitored) at every time instant (e.g. the scenario in Fig. 1).

In case of a static environment (e.g. fixed threats or points of interest in general) the monitoring problem has been largely studied, see e.g. [5], [6]. The goal of these algorithms is to converge to a static configuration maximizing the area monitored by the robot sensors' footprints. On the other hand, dynamic monitoring algorithms have been designed to explore the entire area without selecting the sub–regions of major interest [8] or doing it with high communication costs [9].

We focus on the problem of steering the team of robots to monitor a sea area where communication between robots is reduced, for example for security issues or deteriorated communication channels (e.g. in case of underwater vehicles). It is worth noting that the same approach can be used in several different application scenarios where the goal is to detect, localize and react to any environmental changes of interest, e.g. high variation of temperature, water pollution, etc.

The coordination protocol developed in this paper takes into account all those aspects in the unifying framework of game theory. Indeed, it is well known that the particular class of potential games solves several cooperative control problems with a reduced amount of communication between robots [10]. In particular, the considered control problem can be transformed, with a game theoretic approach, into a non–cooperative game where the goal is to reach specific equilibria. Moreover, in case of “payoff–based” scenarios [11], i.e., scenarios in which robots get a reward in the reached regions based on the action performed by other robots, it has been proved that there exist learning algorithms that can lead to Nash Equilibria [12]. Based on those considerations a novel formulation of the monitoring problem, in presence of moving points of interest, is proposed.

Among learning algorithms for static environment, the Distributed Inhomogeneous Synchronous Learning [6] (DISL) has been developed. Such algorithm, under strict conditions, guarantees that the robots system reaches a configuration of equilibrium that corresponds to a maximizer of the potential function. A drawback

[1] Asymmetric threats are a version of not "fighting fair", which can include the use of surprise in all its operational and strategic dimensions and the use of weapons in unplanned ways.

of DISL is that it has not been designed to deal with dynamical environments where the points of interest may move or change in time.

In this paper, the DISL algorithm is modified to cope with environments characterized by *high rates of dynamicity*, i.e., high velocity of the threat or fast modification in the points of interest with respect to robot speed. More formally, we refer to a *high dynamic* environment whenever the team of robots is not able to reach a steady configuration before a change in the environment is detected. The proposed modification allows robots to react efficiently to environmental changes. As a consequence teams of unmanned marine robots can track a threat without knowing a priori its behaviour, as in case of asymmetric threats.

Finally, the algorithm is evaluated through Monte Carlo simulations in case of scenarios of different geometric complexity such as open sea, bay, and harbours. Moreover, the performance is analysed in terms of a security index that provides a tool for team sizing. Indeed, given the maximum threat velocity, we are able to determine the minimum number of marine robots to be used in the system guaranteeing a desired security level of the area.

The paper is organized as follows. The dynamic monitoring problem is first introduced in Section II while it is formulated as a potential game in Section III. In Section IV the proposed algorithm is reported as a modification of the DISL learning algorithm. Finally in Section V the simulation results are reported, while in Section VI the proposed tool is described.

II. Dynamic Monitoring Problem

To solve the problem of multi-robot coordination against asymmetric threat, we first formalize the problem as a monitoring problem. In order to track moving threats, we are interest in dynamic monitoring different from existing formulations such as in [13]. Moreover, we are interested in including the sensing model in the problem formalization. For those purposes we first consider a discretized workspace (e.g. a sea area). Each sub–region is associated with its centre position $q \in \mathbb{R}^p$ ($p = 2$ for terrestrial vehicles, $p = 3$ for underwater or aerial vehicles). Let $\mathcal{Q}$ be the collection of all the labels q in the discretization. Furthermore, consider the graph $\mathcal{G} = (\mathcal{Q}, E)$ where $(q, q') \in E$ if and only if the sub–regions q and q' are adjacent. The graph $\mathcal{G}$ is assumed to be fixed and connected.

The N robots are deployed in $\mathcal{Q}$ to monitor areas of interest based on sensor footprints. Let V be the set of robot identifiers, each robot i is modeled as a point mass in $\mathcal{Q}$, with location $q_i \in \mathcal{Q}$. Each robot has on board sensors that scan a limited portion of its neighbourhood. Moreover, the scan operation has associated cost in terms of energy consumption that is roughly proportional to the scanned area, as it occurs in case of radar and sonar–like sensors.

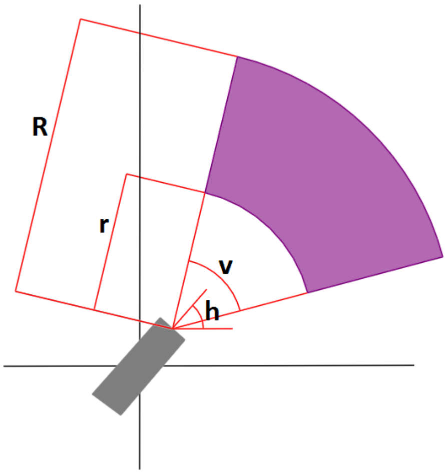


Fig. 2: Example of 2D–sensor footprint: h is the sensor orientation, v is the amplitude of view, while R and r are the maximum and minimum scanning range respectively.

We assume that the footprint of the sensor is directional, has limited range, and has a finite angle of view centred at the sensor position. Following a geometric simplification, we model the sensing region of robot i as a truncated cone in the p–dimensional space, see Fig.2 for $p = 2$. The sensor footprint is completely characterized by the following parameters:

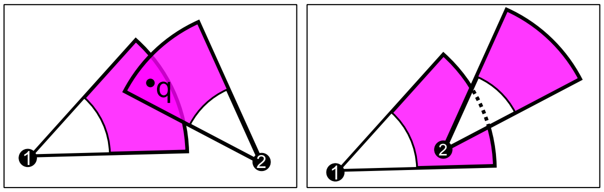
- the position of robot i, $q_i \in \mathcal{Q}$;
- the sensor orientation, given by an angle $h_i \in [0, 2\pi)$;
- the angle of view, $v_i \in [v_{min}, v_{max}]$;
- and the shortest range r (respectively, longest range R) between robot i and the nearest (respectively, farthest) object that can be identified by scanning the sensor footprint, $r_i, R_i \in [r_{min}, r_{max}]$ with $r_i < R_i$.

We assume that each robot can set its own sensor parameters r_i, R_i, v_i, h_i influencing the cost associated to the sensor operation as shown in next section. In other words $c_i = (r_i, R_i, v_i, h_i) \in \mathcal{C} = [r_{min}, r_{max}]^2 \times [v_{min}, v_{max}] \times [0, 2\pi)$ is the sensor configuration vector of robot i. We set $x_i = (q_i, c_i)$ as the state of the i-th robot which comprehends both the position $q_i \in \mathcal{Q}$ and the sensors configuration $c_i \in \mathcal{C}$.

Let X be the configuration space of the robots and $x = (x_1, \ldots, x_N) \in X$ the vector of the current robots configurations. In order to maximize the dimension of the area currently monitored by the N robots, we want to avoid to have more than one robot monitoring the same sub–region q. To this purpose we introduce the notion of neighbour:

Definition 1 (Robot Neighbour)**.** Let $\mathcal{D}(x_i)$ be the sensors footprint of robot i in $\mathcal{Q}$. The set of *neighbours* of robot i, is $\mathcal{N}_i(x) = \{j \in V \backslash \{i\} | \mathcal{D}(x_i) \cap \mathcal{D}(x_j) \neq \emptyset\}$, i.e. the set of robots that monitor one of the sub–regions monitored by robot i, see Fig. 3.

Based on the number of neighbouring robots, robot i



(a) (b)

Fig. 3: Following Definition 1, in (a) *Neighboring* robots, while in (b) *Non–neighboring* robots.

should move to minimize its neighbours. The neighbour concept is hence used to differentiate robots that influence the movement of robot i from those that don't.

In order to formalize the presence of a threat in the environment we now introduce an interest function W that corresponds to the level of criticality of each sub–region of $\mathcal{Q}$, i.e. the probability to find a threat in the sub–region. However, each robot does not have the knowledge of the value of the interest function in any sub–region until monitored by the robot itself.

Definition 2 (Interest Function)**.** The *interest function* $W : \mathbb{Z}_+ \to \mathbb{R}^{|\mathcal{Q}|}$ is the function which assigns to every time instant the vector $W(t) = (W_{q_1}, W_{q_2}, \dots) \in \mathbb{R}^{|\mathcal{Q}|}$ where $W_{q_i} \in \mathbb{R}$ is the element relative to the sub–region (with centre) $q_i \in \mathcal{Q}$.

Each robot is assumed to be able to determine the value $W_q \geq 0$ for each and only those q inside its sensor footprint. Moreover, we suppose that two robots that are monitoring the same sub–region determine the same W_q. As mentioned, values W_q can be interpreted as the probability of finding intruders in the monitored sub–region. Larger values of W_q correspond to sub–region of higher interest, while, when $W_q = 0$ the sub–region q is of no interest or does not belong to the sensor footprint. Hence, the limited sensing capabilities are encoded in the fact that the interest function W is not fully known by the robots. Indeed, the problem we tackle is a partially observable problem, i.e., each robot only has access to limited information regarding its surrounding environment. How such value W_q is computed by robots can be based on distributed supervisor based techniques that are out of the target of this paper, see e.g. [14].

It is worth noting that in asymmetric threats, intruders are supposed to be able to move in the area. Such dynamic environment characteristic is encoded in the time varying property of the interest function $W(t)$.

The problem we want to solve is hence to coordinate a multi–robot system to monitor only those sub–regions with highest probability of finding a threats inside them. For this purpose, we now define an optimal control problem in which a given quality measure is maximized.

Problem (Dynamic Monitoring Problem)**.** *Given a space graph $\mathcal{G} = (\mathcal{Q}, E)$, an interest function $W : \mathbb{Z}_+ \to \mathbb{R}^{|\mathcal{Q}|}$ and a quality measure $\phi : \mathbb{R}^{|\mathcal{Q}|} \times X \to \mathbb{R}$, the* dynamic monitoring problem *is the problem to find an evolution function π, representing the resulting closed–loops dynamic, feasible with respect to the dynamic of every robot (e.g. limited range of movements in one time step), such that*

$$\begin{cases}\pi^* = \underset{\pi}{\operatorname{argmin}} \sum_t \left\|\phi(W(t), x(t)) - \max_{y \in X} \phi(W(t), y)\right\|^2 \\ s.t.\ x(t+1) = \pi(W(t), x(t)).\end{cases} \tag{1}$$

An example of quality measure associated to the monitoring problem is

$$\phi(W(t), x) = \sum_{q \in \mathcal{Q}} \sum_{l=1}^{n_q(x)} \frac{W_q(t)}{l},$$

where $n_q(x)$ is the cardinality of the set $\{k \in V | q \in \mathcal{D}(x_k)\}$, i.e., the number of robots currently monitoring sub–region q. The function ϕ can take into account also a local cost function f_i that depends only on robot configuration x_i. Those functions can be used to model private costs such as the energy consumption of the robot.

For example, a cost that can be associated with the use of the sensor is $f_i(x_i) = f_i(c_i) = kv_i \left(R_i^2 - r_i^2\right)$, i.e. proportional to the area of the sensor footprint as occurs with radar and sonar–like sensors.

To conclude, in this paper we consider the following quality measure ϕ:

$$\phi(W(t), x) = \sum_{q \in \mathcal{Q}} \sum_{l=1}^{n_q(x)} \frac{W_q(t)}{l} - \sum_{i=1}^{n} f_i(x_i). \tag{2}$$

With this choice, the function π^*, solution of the optimal control problem, is such that the probability of the event $\{q \in \cup_{i=1}^{N} \mathcal{D}(x_i(t))\}$ (i.e., "the sub–region q is in the footprint of at least one robot") increases with $W_q(t)$.

III. Game Theoretic Formulation

In this paper we tackle the dynamic monitorin problem in (1) with a game-theoretic approach whose basic concepts are now introduced for reader convenience. Refer to [15] for a more comprehensive introduction.

The first concept of the game theory is the set of player (e.g. marine unmanned robots) that we indicates with $V = \{1, \dots, N\}$. The collective action set A is denoted with $A = A_1 \times \cdots \times A_N$, where A_i is a finite set of actions for robot $i \in V$. The function $u_i : A \to \mathbb{R}$ is the utility function of robot $i \in V$ and each robot behaves so as to maximize u_i. The function $R_i : A_i \to 2^{A_i}$ provides a so-called constrained action set, e.g. corresponding to kinematic constraints. The joint action of the group is denoted by $a = (a_1, \dots, a_N) \in A$ and the collection of actions other than robot i by $a_{-i} = (a_1, \dots, a_{i-1}, a_{i+1}, \dots, a_N)$, hence $a = (a_i, a_{-i})$.

Due to the limited knowledge on the environment, the utility function u_i of each robot depends on the neighbour set and on local available information. To solve the dynamic monitoring problem, the benefit that robot i obtains through sensing is chosen as $\sum_{q \in \mathcal{D}(x_i)} \frac{W_q(t)}{n_q(x)}$.

Such utility function splits the benefit $W_q(t)$ among all the robots that monitor the same sub–region q. The purpose of this choice is to give robot a boost to move toward, and then monitor, areas with highest value of $W_q(t)$ shared with as less robots as possible, i.e. small values of $n_q(x)$. In the considered framework each robot is supposed to gain a reward while monitoring sub–regions but at some price (e.g. energy consumption). The capture of this trade-off is the scope of the utility functions of robot i:

$$u_i(W, x) = \sum_{q \in \mathcal{D}(x_i)} \frac{W_q(t)}{n_q(x)} - f_i(x_i), \tag{3}$$

where $n_q(x)$ can be distributively computed or obtained based on sensor capabilities. The utility function u_i is distributed along the team, because it only depends on the points q within the sensing range $\mathcal{D}(x_i)$ and the actions of $\{i\} \cup \mathcal{N}_i(x)$.

The monitoring problem can be defined as a constrained game $\Gamma = (V, X, \{u_i\}_{i \in V}, \{R_i\}_{i \in V})$ where the set of the collective actions is the state space X and the action selection rules are limited by reachability characteristics of the robot kinematics $\{R_i\}_{i \in V}$. For example, limited selection rules represent the presence of possible obstacles or the constraint on the maximum allowed speed.

Proposition 1. *The monitoring game* Γ *is a constrained potential game with potential function defined in* (2).

The proof is omitted for the sake of brevity. It is based on a direct verification of the potential game definition following a procedure similar to those used in [10].

As a consequence of Proposition 1, the set of pure constrained Nash equilibria (in the following, CNE) of the dynamic monitoring game Γ is not an empty set [11].

In the following we briefly report an algorithm that reach pure CNE of constrained potential games.

A. DISL Learning Algorithm

In case of a time invariant interest function W there exist several algorithms that converge toward a CNE. We now briefly present the Distributed Inhomogeneous Synchronous Learning (DISL) algorithm, [6] that will be next modified to cope with a time varying $W(t)$, i.e. in case of dynamic environment.

At each iteration $t \in \mathbb{Z}_+$, the DISL algorithm chooses an action according to a specific procedure assuming that each robot $i \in V$ stores last two chosen actions $x_i(t-1)$, $x_i(t)$ (i.e., its last two states) and the outcomes $u_i(x(t-1))$ and $u_i(x(t))$ (i.e., the associated gains). The main steps of the algorithms are:

1) At $t = 0$, all robots are placed in $\mathcal{Q}$ and sensors configurations are initialized. Each robot i computes its neighbourhood and $u_i(x(0))$.
2) At each time $t \geq 1$ each robot i executes its action (moves in a configuration and sets its sensor footprint) following a specific learning rule.
3) At the new state, every robot computes its neighbours, utility function and next feasible action set. The process is repeated from point 2.

In step 2 each robot updates a parameter ϵ called *exploration rate* based on

$$\epsilon(t) = t^{-\frac{1}{N(D+1)}}, \tag{4}$$

where D is the diameter of the graph $G = (\mathcal{Q}, E)$, i.e. a measure of the environment discretization, and N is the number of robots. Such parameter is the probability of experimentation: when robot i *experiments*, it chooses next action uniformly from the set of feasible actions, excluded the action with the best utility collected in the last two past steps. On the other hand, with probability $1 - \epsilon$, the robot i does *not experiment*, i.e., it chooses as next action the one with highest utility in the last two steps.

In [6] it is shown that while playing a constrained potential game Γ with DISL rules the team of robots converges to a state that corresponds to a CNE. This holds if every action can be reversed (*reversibility*) and every sub–regions of the environment can be reached in a finite time (*feasibility*). For those reasons in the rest of the paper, we assume the reversibility and the feasibility of the constrain action functions $\{R_i\}_{i \in V}$.

IV. Learning algorithms in dynamic environments

The algorithms DISL, described in previous section, is designed for static environments, i.e., constant W. In this section, we tackle the problem of design learning algorithms in dynamic environments.

A. Homogeneous algorithms

The convergence of DISL, in the static environment framework, depends on the convergence to zero of the the parameter ϵ that is updated based on (4). However, this update rule takes long time to reach a sufficiently small ϵ when the size of the game, in terms of the number of robots (N) and of the space graph diameter (D), is large. Thus, from the practical point of view we might require to decrease ϵ based on heuristic or use algorithms with sufficiently small and constant ϵ. *Homogeneous algorithms* are those in which the parameter ϵ is constant.

In this section, the Distributed Homogeneous Synchronous Learning Algorithms (DHSL) is proposed to solve the monitoring problem in dynamic environments. It is worth noting that the convergence of DISL algorithm in dynamic environment has not yet been proved. On the other hand, for homogeneous algorithms and in particular for DHSL, the convergence can not be ensured even in static environments. However, it holds the following:

Proposition 2. *Consider a constrained potential game* Γ *and suppose that each robot behaves according to the DHSL algorithm. Then, given any probability* $0 < p < 1$, *if the exploration rate* ϵ *is sufficiently small, for all sufficiently large time* $t \in \mathbb{Z}_+$, *the configuration of robots leads to a CNE with probability greater than* p.

Also in this case the proof is omitted for the sake of brevity. It is based on a direct computation of the probability of reaching particular states following a procedure similar to those used in [11].

Proposition 2 provide theoretical foundations to the proposed extension and the application of DHSL in dynamic environments. Indeed, the proposition assures that the optimal actions are selected with high probability if the final value of the exploration rate is sufficiently small.

B. *Validation of homogeneous algorithm in static environment*

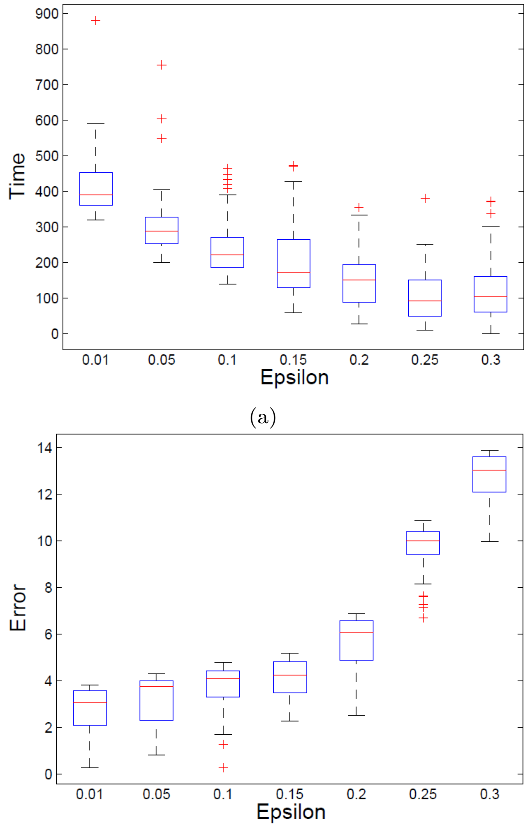
To determine the best value of the exploration rate ϵ, we evaluate in simulations how the required performance changes with respect to ϵ. For this purpose, the algorithm has been tested for different $\epsilon \in [0.01, 0.3]$ and different number of robots $N \in \{5, 10, 15, 20\}$, in a static environment with 400 sub–regions representing the open sea scenario. Moreover, for any values of ϵ and N, 50 random initial conditions have been considered.
The robot sensor footprint considered in this set of simulations is a circle centred on the robot and of radius corresponding to the dimension of two sub–regions. The interest function W is a Gaussian density function whose mean is fixed at the centre of the space $\mathcal{Q}$ and variance is 25. Each robot i optimizes the utility function u_i defined in (3) where f_i is supposed to be constant and equal for every robots.

Fig. 4 reports the results of the simulations as a function of the exploration rates ϵ. The Fig. 4a shows the time to reach a steady configuration, i.e., a CNE configuration, for different ϵ. On the other hand, the Fig. 4b shows the difference between the reached value and the maximum of the potential function, i.e., the value of the potential in the CNE. Such difference represents a mean error with respect to the optimal value of the potential function.

Simulations show that by increasing the exploration rate, the time of convergence to a CNE configuration decreases while the CNE can be reached with decreasing probability. Hence, the displacement with respect to the steady configuration increases with ϵ. Based on those results, the chosen value of ϵ is a trade–off between the convergence time and the mean potential error. The value $\epsilon = 0.1$ has been chosen, for the next sets of simulations, due to the limited variation of the two examined indexes around that value.

V. Simulations in dynamic environments

As mentioned, in case of protection against asymmetric threats, relevant situations are those where an interesting subject have to be tracked. To tackle the problem, we consider interest function W as a Gaussian density function whose mean changes in time. Hence, in the following set of simulations, $W(t)$ is supposed to shift any M time steps, i.e., M^{-1} may represent the intruder velocity. If M is such that the system can not reach the steady states, a reasonable choice is still to generate an evolution that tends to the optimal value of the potential function, defined in (2), with the current state of $W(t)$. Since the evolution of the interest function $W(t)$ is unknown and unpredictable by robots, learning algorithms has to allow robots to maximize the reward received by the current, available, $W(t)$.



(a)

(b)

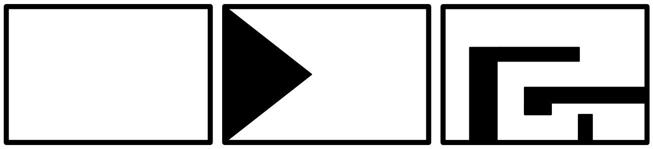
Fig. 4: (a) Time of convergence to a CNE configuration and (b) displacement with respect to the CNE, with respect to ϵ.

We are interested to evaluate the proposed algorithm in different scenarios. In particular three scenarios characterized by different degrees of convexity are considered (convexity measure is described in [17]), see Fig. 5. Such kind of scenario represents a limited area in an open sea environment, $\mathcal{C} = 1$, in presence of an island, $\mathcal{C} = 0.75$, and in an harbour, $\mathcal{C} = 0.43$.

To the authors best knowledge there is no proof of convergence of learning algorithm in dynamic environments. Hence, the convergence in dynamical context is now tested with Monte Carlo based simulations.

Simulation Setup

The DHSL algorithm has been tested for different values $M \in \{1, \dots, 10\}$ at which the interest W function changes (i.e., for different intruder velocities) and for different number of robots $N \in \{5, 10, 15, 20\}$. The exploration rate is $\epsilon = 0.1$. Each simulation has been run for 10^4 time steps. Moreover, for any values of M and



(a) Open sea (b) Bay (c) Harbour

Fig. 5: Considered scenarios are characterized by different degrees of convexity $\mathcal{C}$, decreasing from the left to right: $\mathcal{C}_a = 1$, $\mathcal{C}_b = 0.75$, $\mathcal{C}_c = 0.43$, respectively. See [17] for details on convexity measure.

N, 50 random initial conditions have been considered. The robot sensor footprint is the same as the one used in previous section. Each robot i optimizes the utility function u_i defined in (3) where f_i is supposed constant and equal for every robots. The free space of each scenario has been discretized in 400 sub–regions. Higher number of discretized sub–regions would require a higher number of robots to maintain the same performance. In the experiments, we use the following Gaussian density function whose mean $\mu(t)$ randomly shifts every M steps, moving from a sub–region to the adjacent:

$$W(q) = e^{-\frac{||q-\mu(t)||^2}{2\sigma^2}}, \sigma = 5. \tag{5}$$

Simulation Results

To evaluate the DHSL algorithm, we use the error index $\mathcal{I} \in [0, 1]$ defined as follow

$$\mathcal{I}(T) = 1 - \frac{1}{T}\sum_{t=1}^{T}\frac{\sum_{i\in V} u_i(t)}{\sum_{q\in\mathcal{Q}} W_q(t)} \tag{6}$$

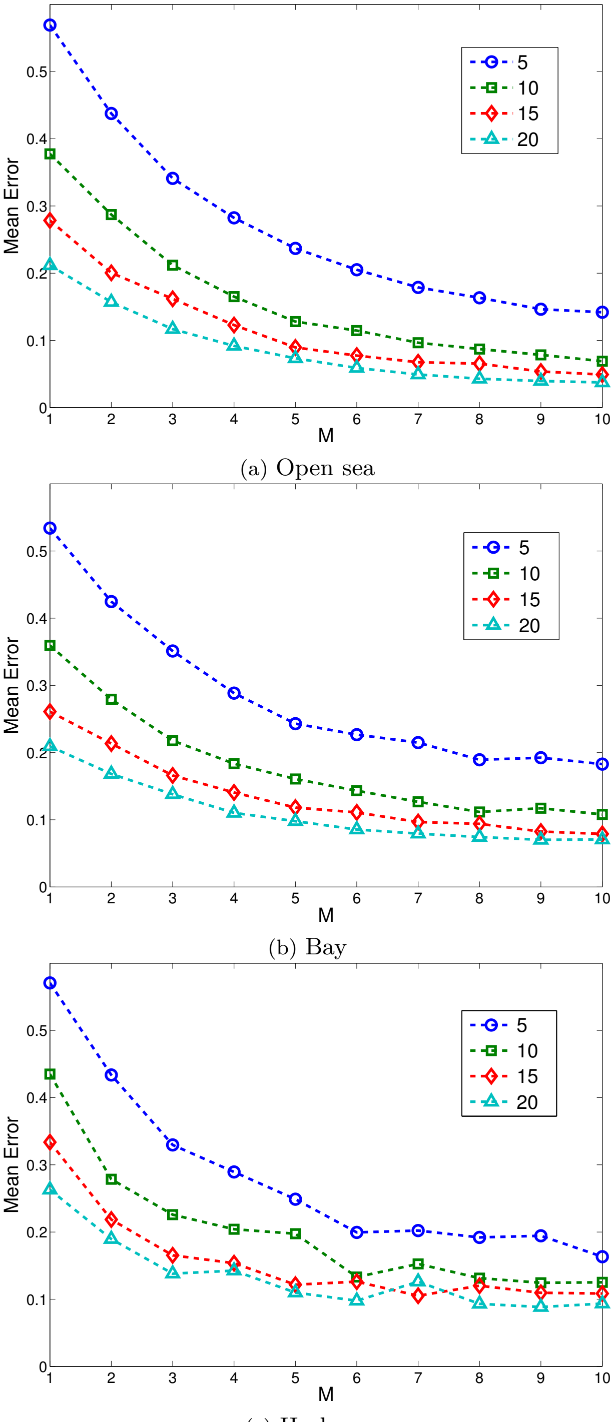
where $u_i(t)$ is the utility of robot i and $T \in \mathbb{Z}_+$ is the time step. The proposed index $\mathcal{I}$ represents the cumulative error and it is inspired to the well known IAE index. It depends on the total benefit of the scenario (determined by the W in the whole environment) and the total benefit reached by the team (determined by the robot utility functions u_i).

Graphic in Fig. 6 shows how the mean of the error decreases when the intruder velocity decreases. Notice that, as expected, the value of the mean error at steady state is coincident with the one obtained in the static case. It is worth noting that the mean error decreases despite of the number of robots N and the geometry of the scenario.

Simulations have shown that the error variance does not change with N. On the other hand, it increases in non–convex scenarios (bay and harbour) with respect to the convex one (i.e., open sea) and in the harbour scenario with respect to the bay one. Moreover, the mean error grows with loss of convexity. Despite this, Fig. 6 shows a regularity with respect to the variation over the intruder velocity M and the number of robots N.

VI. Team Sizing Tool

Based on the results obtained in the simulations, we now propose a tool to design the multi–robot system: the minimum number of robots of the team (N) is a function of the intruder velocity (M) and the security index (i.e., rate of coverage) of the area. The similarities in the convergence behaviour presented in previous section permit a collective representation of performance evolution, which is presented in Fig. 7. The graph is composed by the overlapping of the iso–level curves of



(a) Open sea

(b) Bay

(c) Harbour

Fig. 6: Mean error evolution over different intruder velocities (M) and number of robots (N), for DHSL algorithm.

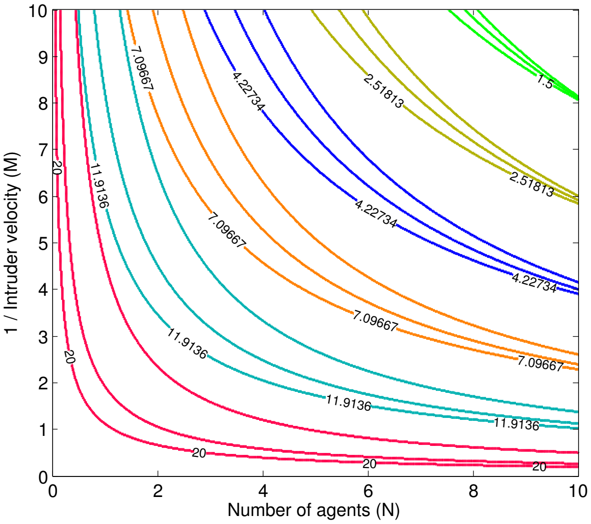


Fig. 7: Tool for design the size of the team of robots: the minimal number of robots is identified with the maximum intruder velocity and the maximal security index of the area.

the mean error fitting function in the three scenarios.

The chosen error fitting function is:

$$(1-\mathcal{S})e^{-(aM^bN^c)}+\mathcal{S} \tag{7}$$

where, for every scenario, $\mathcal{S}$ is the estimated static error that is the minimum mean error obtained with M_{max} and N_{max} and a, b and c are the fitting parameters. The chosen fitting function is motivated, in addition to the error evolution, also by the facts that error needs to be 1 when M or N is zero and $\mathcal{S}$ when M or N grows to infinity. The values in Fig. 7 are referred to the triplet of curves of same colour. The triplets are composed by α-level with $\alpha = r\mathcal{S}$, where the label r (the *security index* of the area) indicates how much the error is worse than the static case $\mathcal{S}$.

Fig. 7 has been designed as a tool to determine the size of the team of robots. Indeed, the minimal number of robots in a team is function of the maximum intruder velocity and the maximum security index accepted. For example, setting $M \geq 5$ and the given a maximum security index equals to 7, the minimum number of robots to guarantee the desired performance is between 4 and 6.

VII. Conclusions

The problem of coordinating a system of marine unmanned robots for protection against asymmetric threats has been modelled as a potential games. A learning algorithm has been extended in case of dynamic environment to allow robots to track an intruder. The proposed extension has been evaluated in simulation in different scenarios. The simulation results are used to generate a tool to size the team (N) based on intruders characteristics (M) while maintaining a given level of performance. As a future work is the formal proof of convergence of the proposed algorithm in case of dynamic environments.

VIII. Acknowledgements

This work has been partially supported by the Ligurian Cluster on Marine Technologies (DLTM), project ProDifCon.

References

[1] S. J. Blank, “Rethinking asymmetric threats,” DTIC Document, Tech. Rep., 2003.

[2] D. Darren and J. Sutton, “Maritime Force Protection Operations Analysis Methodology Development,” International Maritime Protection Symposium, USA, 12-14 December 2005.

[3] R. T. Kessel, C. Strode, and R. D. Hollett, “Nonlethal weapons for port protection: Scenarios and methodology,” in *5th European Symposium on Non-lethal Weapons*, 2009.

[4] A. Caiti, A. Munafò, and G. Vettori, “A geographical information system (gis)-based simulation tool to assess civilian harbor protection levels,” *Oceanic Engineering, IEEE Journal of*, vol. 37, no. 1, pp. 85–102, 2012.

[5] A. Howard, M. J. Matarić, and G. S. Sukhatme, “Mobile sensor network deployment using potential fields: A distributed, scalable solution to the area coverage problem,” in *Distributed autonomous robotic systems 5*. Springer, 2002, pp. 299–308.

[6] M. Zhu and S. Martínez, “Distributed coverage games for energy-aware mobile sensor networks,” *SIAM Journal on Control and Optimization*, vol. 51, no. 1, pp. 1–27, 2013.

[7] T. Goto, T. Hatanaka, and M. Fujita, “Payoff-based inhomogeneous partially irrational play for potential game theoretic cooperative control: Convergence analysis,” in *American Control Conference (ACC), 2012*. IEEE, 2012, pp. 2380–2387.

[8] M. A. Batalin and G. S. Sukhatme, “Multi-robot dynamic coverage of a planar bounded environment,” DTIC Document, Tech. Rep., 2003.

[9] E. Frazzoli and F. Bullo, “Decentralized algorithms for vehicle routing in a stochastic time-varying environment,” in *Decision and Control, 2004. CDC. 43rd IEEE Conference on*, vol. 4. IEEE, 2004, pp. 3357–3363.

[10] J. R. Marden, G. Arslan, and J. S. Shamma, “Cooperative control and potential games,” *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, vol. 39, no. 6, pp. 1393–1407, 2009.

[11] J. R. Marden, H. P. Young, G. Arslan, and J. S. Shamma, “Payoff-based dynamics for multiplayer weakly acyclic games,” *SIAM Journal on Control and Optimization*, vol. 48, no. 1, pp. 373–396, 2009.

[12] D. P. Foster, H. P. Young *et al.*, “Regret testing: learning to play nash equilibrium without knowing you have an opponent,” 2006.

[13] J. Cortes, S. Martinez, and F. Bullo, “Spatially-distributed coverage optimization and control with limited-range interactions,” *ESAIM: Control, Optimisation & Calculus of Variations*, vol. 11, pp. 691–719, 2005.

[14] A. Fagiolini, G. Valenti, L. Pallottino, G. Dini, and A. Bicchi, “Decentralized intrusion detection for secure cooperative multi-agent systems,” in *Proc. IEEE Int. Conf. on Decision and Control*, 2007, pp. 1553–1558.

[15] M. I. Freidlin, J. Szücs, and A. D. Wentzell, *Random perturbations of dynamical systems*. Springer, 2012, vol. 260.

[16] J. R. Marden and J. S. Shamma, “Revisiting log-linear learning: Asynchrony, completeness and payoff-based implementation,” *Games and Economic Behavior*, vol. 75, no. 2, pp. 788–808, 2012.

[17] J. Zunic and P. L. Rosin, “A new convexity measure for polygons,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 26, no. 7, pp. 923–934, 2004.