A game theoretic approach for antagonistic-task coordination of underwater autonomous robots in asymmetric threats scenarios

Simone Nardi*†, Tommaso Fabbri‡, Andrea Caiti†‡ and Lucia Pallottino†‡
*Department of Mathematics, University of Pisa, Italy
†Research Center "E. Piaggio", University of Pisa, Italy
‡Department of Information Engineering, University of Pisa, Italy

***Abstract*—This work proposes a game theoretic approach to tackle the problem of multi-robot coordination in critical scenarios where communication is limited and the robots must accomplish different tasks simultaneously. An important application falls in underwater robotic framework where robots are used to protect a ship against asymmetric threats guaranteeing simultaneously the coverage of the area around the ship and the tracking of a possible intruder. The problem is modelled as a potential game for which novel learning protocols are introduced. Indeed, a general extension of pay-off based algorithms is herein proposed where the main difference with state-of-the-art protocols is that the trajectory optimization is considered instead of single action optimization. Moreover, the proposed T-algorithms, steer the robots toward Nash equilibria that will be shown to correspond to the accomplishment of different, possibly antagonistic, goals. Finally, performances of the algorithms, under different scenarios, have been evaluated in simulations.**

I. Introduction

It is well known that the problem of detecting and accordingly reacting to an asymmetric threat in marine environments is still an open problem from methodological and technological points of view [1]. Even though the available surveillance sensors on naval platforms are sufficient to identify and classify asymmetric threats, they are able to give a quick alert only in nominal working conditions. Indeed, adverse weather conditions easily lead to degradation of sensors performance. One of the possible consequences is the drastic reduction of the time available for a possible reaction after the detection, identification and classification procedures [2], [3]. This occurs also in particular situations when, for example, there exists an obstruction to the line of sight of the sensor system such as in presence of an island. The short time–to–reaction may increase the possibility of human errors especially in stressful situations (e.g. an incorrect assessment of the necessary reaction).

Autonomous surveillance systems can guarantee an adequate supervision of the area in any working conditions even though the entire area is not fully monitored at any time instant. Indeed, the mobility abilities of autonomous marine vehicles can be exploited to deploy the team of robots to monitor the environment. For example, in case of a static environment (e.g. fixed area of interest in the scenario) the coverage problem has been largely studied, see e.g. [4]–[6]. Such algorithms are proven to converge to a static configuration maximizing the number of interested area covered by the robot sensors' footprint. In case of dynamic environment algorithms have been designed to explore the entire area without selecting the sub–regions of major interest [7] or doing it with high communication costs [8].

In this work we focus on the particular problem of monitoring an area with a set of autonomous robots based on partial knowledge of the environment due to limited sensors footprint and communication range. The coordination of the robot must also guarantee the accomplishment of other tasks in a framework in which communication is lim-

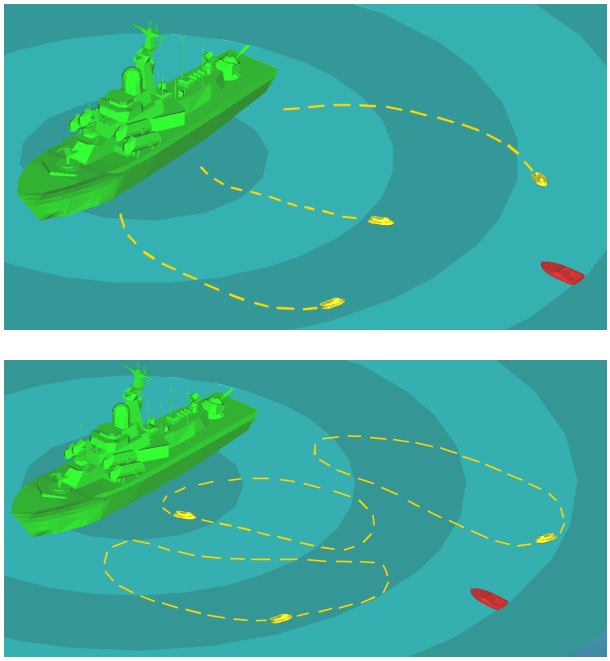


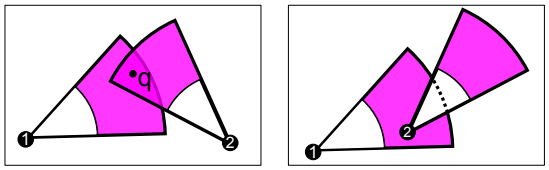
Fig. 1: Example of an asymmetric threat detected by a team of autonomous robots. The team of robots efficiently supervises the area around the ship. In the bottom image an example of antagonistic tasks are the monitoring of the area around the main ship while detecting and tracking an intruder.

ited due to security issues or deteriorated communication channels (e.g. underwater scenarios). Referring to Fig. 1, an example of antagonistic tasks are the monitoring of the area around the main ship while detecting and tracking an intruder. It is worth noting that the marine scenario is only a possible application of the proposed methodology that is valid whenever the goal is to detect, localize and react to any environmental changes of interest, e.g. high variation of temperature, water pollution, terrorists attacks etc.

The main idea is to use a game theoretic approach to tackle the considered problem. 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 [9]. In particular, the considered control problem is transformed into a non–cooperative game where the goal is to reach specific equilibria. Moreover, in case of "payoff–based" scenarios [10], i.e., robots get a reward in the reached regions based on the action performed by other robots, and this helps in capturing the coverage requirement into the problem formulation. Learning algorithms that can steer the robots toward to Nash equilibria are demonstrated to partially solve the problem, see e.g., [11].

In case of dynamic environment, the Distributed Homogeneous Synchronous Learning (*DHSL*) and the Payoff-based Homogeneous Partially Irrational Play (PHPIP) have been presented in [12]. Such algorithms have not been designed to deal with antagonistic goals as in case of asymmetric threat scenarios, where intruders have to be tracked while patrolling the area around the ship. Hence, in this paper, we propose an extension of *DHSL* and *PHPIP* to cope with environments characterized by *low rates of dynamicity*, i.e., low velocity of the threat or in general of environmental changes with respect to robot speed. More formally, we refer to a *low dynamic* environment whenever the team of robots is able to reach a steady state before a change in the environment is detected. In such scenario, we propose two novel learning algorithms, Trajectory *DHSL* (*T–DHSL*) and Trajectory *PHPIP* (*T–PHPIP*) that inherits several important features from the original ones: (i) require finite and limited memory, (ii) are applicable to payoff–based scenario, (iii) allow synchronous action, (iv) use simple rules for action selection, (v) allow local constraints on actions. The main difference with state-of-the-art algorithms is that the proposed *T*–Algorithms are based on trajectory optimization instead of single action optimization. It is worth mentioning that the same extension can be applied to any pay-off based algorithm. With *T*-Algorithms, teams of robots are able to manage different goals as required by moving toward Nash equilibria. Performances, under different scenarios, are evaluated in Monte Carlo simulations. Moreover, *T*-Algorithms have been integrated with the Robot Operating System [13] (ROS) in order to verify algorithm robustness under different robot dynamics [14].

The remain of the paper is organized as follow: the considered problem is formalized in Section II while the necessary background on the game theoretic framework is briefly reported in Section III. The main contribution of the paper, i.e. the *T*-Algorithms, is reported in Section IV while simulation results are presented in Section V.



(a) (b)

Fig. 2: (a) Neighboring robots, (b) Non–neighboring robots.

II. Problem Formulation

To deal with dynamic environments, a different formalization of the dynamic coverage problem, e.g. from the one in [15], is proposed. Pay–off based potential games have been shown to efficiently represent frameworks in which the team objective is to cover a space of interest, see e.g., [12]. For reader convenience a brief introduction of the topic is reported.

Consider a discretized non–convex workspace where each element, or 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 in the workspace. The graph $\mathcal{G}$ is assumed to be fixed and connected. N robots are deployed in $\mathcal{Q}$ to measure areas of interest that may change in time (e.g., an intruder moving in the environment). Let V be the set of robot identifiers, $x_i = \left[q_i^T c_i^T\right]^T$ be the state of the i-th robot which comprehends both the position $q_i \in \mathcal{Q}$ and the sensors configuration $c_i \in \mathcal{C}$. For example, in case of robots with an on board camera, $(\theta, r) \in \mathcal{C} = [0, 2\pi] \times [r_{min}, r_{max}]$ corresponds to the camera orientation (θ) and the focal length r.

Let X be the configuration space of the robots and $x = (x_1, \dots, x_N) \in X$ the vector of the current configuration of the robots. In order to avoid that more than one robot is monitoring the same sub–region and hence to increase the area covered we introduce the following:

Definition 1 (Robot Neighbour)**.** Let $\mathcal{D}(x_i)$ be the sensors footprint of robot i. The set of *neighbours* of robot i, is $\mathcal{N}_i(x) = \{j \in V \backslash \{i\} | \mathcal{D}(x_i) \cap \mathcal{D}(x_j) \cap \mathcal{Q} \neq \emptyset\}$, i.e. the robots that monitor the same sub–region of i, see Fig. 2.

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 q inside its sensor footprint. We assume that robots monitoring the same sub–region determine the same value W_q. Values W_q can be interpreted as the interest of monitoring the area q, such as probability of finding intruders in the monitored sub–region. How such value W_q is computed by robots is out of the target of this paper and can be based on supervisor based techniques, see e.g., [16], [17]. 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 vector function W is not fully known by the robots, due to limited sensing capabilities. Indeed the problem is a partially observable problem, i.e., each robot only has access to limited information regarding its surrounding environment.

In the framework of asymmetric threats, intruders are supposed to be able to move in the environment and such environmental changes are encoded in a time-dependent interest function $W(t)$.

Definition 3 (Dynamic Coverage Problem)**.** Given a space graph $\mathcal{G} = (\mathcal{Q}, E)$, an interest function $W : \mathbb{Z}_+ \to \mathbb{R}^{|\mathcal{Q}|}$ and a quality coverage metric $\phi : \mathbb{R}^{|\mathcal{Q}|} \times X \to \mathbb{R}$, the *dynamic coverage 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

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

As a consequence, the function π^* will be such that the probability of the event "the sub–region q is in the footprint of the robots" (i.e. the event $\{q \in \cup_{i=1}^{N} \mathcal{D}(x_i(t))\}$) tends to increase when W_q grows.

The function ϕ, in Definition 3, measures the coverage quality. An example of measure of quality is: $\phi(W, x) = \sum_{q \in \mathcal{Q}} \sum_{l=1}^{n_q(x)} \frac{W_q}{l}$, where $n_q(x)$ is the cardinality of the set $\{k \in V | q \in \mathcal{D}(x_k) \cap \mathcal{Q}\}$, i.e., the number of robots whose position and configuration of sensors allow to monitor the same sub–region q.

The function ϕ can take into account also some local cost functions f_i that depend only on robot configuration x_i. Those can be used to model private robot cost functions such as the energy consumption consumed by the robot. Hence, the considered quality coverage measure ϕ is:

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

The goal of the distributed coordination protocol is hence to steer the team of robots toward the maximum of the quality measure ϕ. The control laws π^* in (1) will be determined in next section.

III. Game Theoretic Formulation

Some basic and necessary game-theoretic concepts [18] used to obtain the optimal control laws in (1) are here briefly described.

A. Constrained Potential Games

Definition 4. [Constrained Strategic Game [10], [5]] A constrained strategic game is defined as $\Gamma = \big(V,\ A,\ \{ U_i(\cdot) \}_{i \in V},\ \{ R_i(\cdot) \}_{i \in V} \big)$, with $V = \{1, \ldots, N\}$ the set of robots. The collective action set A is $A = A_1 \times \cdots \times A_N$, 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.

The joint action of the group is denoted by $a = (a_1, \ldots, a_N) \in A$ and the collection of actions other than robot i by $a_{-i} = (a_1, \ldots, a_{i-1}, a_{i+1}, \ldots, a_N)$, hence $a = (a_i, a_{-i})$.

Definition 5 (Constrained Potential Games [9])**.** A constrained strategic game Γ is said to be a *constrained potential game* with potential function $\phi : A \to \mathbb{R}$ if for all $i \in V$, $a_i \in A_i$ and $a_{-i} \in \prod_{j \neq i} A_j$, the following equation holds for every $a_i' \in R_i(a_i)$.

$$U_i(a_i', a_{-i}) - U_i(a_i, a_{-i}) = \phi(a_i', a_{-i}) - \phi(a_i, a_{-i}) \tag{3}$$

Condition (3) implies that if robot changes its action, the change of the local objective function is equal to that of the group.
In non-cooperative game theory, the most important concept is the well known (pure) Nash Equilibrium (NE), see e.g. [19]. For our problem, we refer to the more general concept of Constrained (pure) Nash Equilibrium (CNE):

Definition 6 (Constrained Nash Equilibria)**.** For a constrained strategic game Γ, a collection of actions $a^* \in A$ is said to be a *constrained pure Nash equilibrium* if the following equation holds for all $i \in V$:

$$U_i(a_i^*, a_{-i}^*) = max_{a_i \in R_i(a_i^*)} U_i(a_i, a_{-i}^*) \tag{4}$$

Notice that, from (4), CNE are characterized by the subset of actions determined by the function R_i.
It is known that any constrained potential game has at least one pure CNE and each pure CNE is a potential function maximizer [20].

Throughout this paper, we use the following assumptions on the constrain action functions:

Assumption 1. The function $R_i : A_i \to 2^{A_i}$ satisfies the following conditions.

(i) [Reversibility] For any $i \in V$ and any a_i^1 and a_i^2 the inclusion $a_i^2 \in R_i(a_i^1)$ is equivalent to $a_i^1 \in R(a_i^2)$.
(ii) [Feasibility] For any $i \in V$ and any $a_i^1, a_i^m \in A_i$, there exists a sequence of actions $a_i^1 \to \cdots \to a_i^m$ satisfying $a_i^l \in R_i(a_i^{l-1})$ for all $l \in \{1, \ldots, m\}$.

B. Coverage Problem as Potential Game

We are now interested in defining the proposed coverage problem as a constrained potential game. The benefit that robot i obtains through sensing is chosen as $\sum_{q\in\mathcal{D}(x_i)\cap\mathcal{Q}} \frac{W_q}{n_q(x)}$. Such utility function splits the benefit W_q among all the robots that monitor the same sub–region q. The purpose of this choice is to give robots movement a boost to look for areas with highest value of W_q shared with as less robots as possible. In the considered framework each robot is supposed to both gain (a reward) and to lose (e.g. consume of energy) while monitoring sub–regions. The capture of this trade-off is the scope of the utility functions of robot i:

$$u_i(W,x) = \sum_{q\in\mathcal{D}(x_i)\cap\mathcal{Q}} \frac{W_q}{n_q(x)} - f_i(x_i), \tag{5}$$

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 set A of the collective actions is, in our problem, the state space X where there are constraints on the feasible states, such as restricted/forbidden areas. On the other hand, function R_i takes into account reachability characteristics of the robot kinematics, e.g. the impossibility for robots to walk through walls, the constraint to move slower than the maximum allowed speed or any other kinematic or mobility constraint. Finally, the coverage problem can now be defined as a constrained game $\Gamma_{cov} = (V, X, \{u_i\}_{i\in V}, \{R_i\}_{i\in V})$ for which it holds the following

Proposition 1. *The coverage game Γ_{cov} is a constrained potential game with potential function defined in* (2).

The proof is based on a direct verification of the definitions and it is omitted for brevity.

As a consequence of Proposition 1, the set of pure CNE of the dynamic coverage game Γ_{cov} is not an empty set [10].
We are now able to show known algorithms that reach pure CNE of constrained potential games.

C. Learning Algorithms

A learning algorithm is an algorithm that induces a closed–loop dynamic π, introduced in Definition 3, to converge to a pure CNE of the constrained potential game.

We restrict the analysis to two distributed learning algorithms: the Distributed Inhomogeneous Synchronous Learning (*DISL*) algorithm, [5], and the Payoff–based Inhomogeneous Partially Irrational Play (*PIPIP*) algorithm, [6].

At each iteration $t\in\mathbb{Z}_+$, the learning algorithms choose 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 $t\geq 1$, based on $R_i(x_i(t))$, each robot i updates its state following a specific learning rule.
3) At the new position, every robot computes its neighbours, utility function and next feasible action set. The process is repeated from point 2.

The two algorithms differ in their learning rule: each robot updates a parameter ϵ called *exploration rate* by

$$\epsilon(t) = t^{-\frac{1}{N(D+1)}}, \tag{6}$$

where D is the diameter of the graph $G=(\mathcal{Q},E)$ and N is the number of robots.
In particular, the learning rule for *DISL* is:

- With low probability, ϵ, the robot i *experiments*, i.e., it chooses the next action uniformly from the set $R_i(x_i(t))\setminus\{x_i^{opt}(t)\}$, where $x_i^{opt}(t)$ is defined as the position with the best utility reached in the two past steps.
- With high probability, $1-\epsilon$, the robot i does *not experiment*, i.e., it chooses next action as $x_i^{opt}(t)$.

On the other hand, the learning rule for *PIPIP* is:

- If $u_i(x(t))\geq u_i(x(t-1))$ holds, it follows the same rule of *DISL* algorithm.
- Otherwise, when $u_i(x(t)) < u_i(x(t-1))$ the robot i:
  - With probability ϵ, chooses its action uniformly from the set $R_i(x_i(t))\setminus\{x_i(t), x_i(t-1)\}$.
  - With probability $(1-\epsilon)\kappa\epsilon^{\Delta_i(t)}$, where $\Delta_i(t) = u_i(x(t-1)) - u_i(x(t))$, chooses $x_i(t)$ (the *irrational* decision).
  - With probability $(1-\epsilon)(1-\kappa\epsilon^{\Delta_i(t)})$, chooses $x_i(t-1)$.

  where κ is the *irrational factor* chosen to satisfy

$$\kappa\in\left(\frac{1}{C-1},\frac{1}{2}\right], C = \max_{i\in V}\max_{x\in X}|R_i(x)| \tag{7}$$

The main difference from *DISL* and *PIPIP* is the irrational factor k of the *PIPIP* algorithm that is used to avoid local maximizer of the potential function. In [5] it is shown that any team of robots playing a constrained potential game Γ satisfying Assumption 1, and following *DISL* rules converges to a CNE. In [6] is shown that any team of robots playing the same game Γ, and following *PIPIP* rules converges to a potential maximizer which is an efficient CNE, i.e., it is the global maximizer of the potential function.

IV. *T–DISL* and *T–PIPIP* Algorithms

In this section an extension of pay-off based algorithm is proposed and applied to the algorithms described in previous section, called *Trajectory DISL* (*T–DISL*) and *Trajectory PIPIP* (*T–PIPIP*).

T–DISL (and *T–PIPIP*) will allow the implementation of a controlled dynamics π that converges to a periodical steady state trajectory of period T, instead of a steady state configuration as the original algorithms do. In case of dynamic environments, a steady state configuration can prevent robots to detect changes in the interest function values, W_q. On the other hand, with the periodical steady state trajectory, provided by T–Algorithms, such problem may be overcome. An important improvement due to T–algorithms is the capability to solve problems having more conflicting goals without changing the action selection rules. Another important motivation in using *T–DISL* and *T–PIPIP* algorithms is given by scenarios in which robots are not able or suitable to stop for stability reasons or practical limitations in ignition, e.g. some types of underwater and aerial robots.

Those algorithms are based on robot utility functions of the form

$$u_i^T(W, x_i(t-(T-1)), \ldots, x_i(t)) = \sum_{h=0}^{T-1} [u_i(W, x_i(t-h))] - \psi_i\left(x_i(t-(T-1)), \ldots, x_i(t)\right), \tag{8}$$

where $T \in \mathbb{Z}_+$ is the period of the trajectory, i.e. in case $T = 1$ the algorithms correspond to their original version. The function $\psi_i(x_i(t-(T-1)), \ldots, x_i(t))$ is used to optimize a second performance index that can be antagonistic with respect to the first one but it depends only on robot i and its past actions. In the considered scenario, ψ is introduced to force robots to move and avoid a steady state configuration. For example, for a period $T = 2$, a possible choice is

$$\psi_i(x_i(t-1), x_i(t)) = ||x_i(t-1) - x_i(t)||_\infty^{-1}. \tag{9}$$

Indeed, by maximizing own utility functions, the robots tends to maximize $||x_i(t-1) - x_i(t)||$ and hence are forced to move.

As a consequence, the coverage quality measure is:

$$\phi^T(W(t), x(t-(T-1)), \ldots, x(t)) = \sum_{h=0}^{T-1} (\phi(W, x(t-h))) - \sum_{i=1}^{n} \psi_i(x_i(t-(T-1)), \ldots, x_i(t)). \tag{10}$$

The function ϕ^T will be proved to be a potential function associated to the utility functions u_i^T, see Theorem 2 in the following.

A. T–DISL *Algorithm*

For space limitations, only *T–DISL* algorithm is described in detail. A sligthly different approach is used for the *T–PIPIP* protocol.

In order to simplify the protocol description we introduce the following technical definition.

Definition 7. Let $R_i^n(x_i(t), x_i(k)) \subseteq R_i(x_i(t))$, with $k, t, n \in \mathbb{Z}_+$ and $k \leq t$, be the set of actions in $R_i(x_i(t))$ for which it does not exist a path of control actions, that ends at $x_i(k)$, whose length is shorter than n.

In the *T–DISL* algorithm the *exploration rate* ϵ is

$$\epsilon(t) = \frac{1}{T}\, t^{-\frac{1}{N(D+1)}}, \tag{11}$$

where D is the diameter of the graph $G = (\mathcal{Q}, E)$, N is the number of robots and T is the period.

The *T–DISL* is a learning rule of period T where the *DISL* learning rule is applied only every T time steps. As stated later, this approach maximize the coverage index in (10). Hence, in the remaining $T-1$ steps of a generic time period $[0, T-1]$, other possible strategies may be applied by robot i to cope with the optimization of the private performance index ψ_i. In particular, for convergence purposes, during the $T-1$ steps the robot actions must steer the robot, at time $T-1$, in a configuration reachable by the configuration in which the robot was at time 0. More formally:

1) Initialization
   - At $t = 0$, all robots are uniformly placed in Q.
   - At each time $t < T-2$, each robot:
     - chooses the best next action in the set $R_i(x_i(t)) \backslash R_i^{T-t}(x_i(t), x_i(0))$,
     - for every $q \in \mathcal{D}(x_i(t))$ computes $n_q(x(t))$ and $u_i(x_i(t))$ and,
     - moves to next position.
   - At $t = T-2$, each robot
     - chooses the best next action in the set $R_i(x_i(T-2)) \cap R_i(x_i(0))$,
     - for every $q \in \mathcal{D}(x_i(T-2))$ computes $n_q(x(T-2))$, $u_i(x_i(T-2))$ and
     - moves to next position.
2) Update: At each time $t = k(T-1)$ where $k \in \mathbb{Z}_+$, each robots computes $u_i^T(W, x_i(t-(T-1)), \ldots, x_i(t))$ and updates its trajectory according to the following rules:
   - With probability ϵ: robot i *experiments*. It selects best next action from the set $R_i(x_i(t))$.
   - With probability $1-\epsilon$: robot i does *not experiment*. It selects the first position of the trajectory $\underline{x}_i^{opt}(t)$. Where $\underline{x}_i^{opt}(t)$ is the more successful trajectory of robot i in the last two period, i.e., the one which gains the best value of $u_i^T(W, \cdot)$ in $(x_i(t-(2T-1)), \ldots, x_i(t-T))$ and $(x_i(t-(T-1)), ..., x_i(t))$.
3) Free Movement Update: at each time t such that $k(T-1) < t < (k+1)(T-1)$ for $k \in \mathbb{Z}_+$, if the robot at time $k(T-1)$ has chosen to experiment, it now chooses the next action from the set $R_i(x_i(t)) \backslash R_i^{(T-(t-k(T-1)))}(x_i(t), x_i(k(T-1)+1))$. Otherwise, it selects the next position of the best trajectory, already selected at time $k(T-1)$.
4) Neighbour Computation: at the new position, every robot computes its neighbours, utility function and

next feasible action set. The process is repeated from point 2.

It is worth specifying that, in the free movement update, the next action for robot i is the feasible action that maximize the private performance index ψ_i. In this way while once, any T time steps, the function ϕ^T is used to chose the best action, the remaining $T-1$ steps the other function ψ^T is optimized leading to the accomplishment of antagonistic tasks.

In order to apply de facto the algorithm, robots must be able to reach a peak velocity at least three times higher than the nominal one. Indeed, referring to Fig. 3, while non experimenting, robots have to go back to $x_i(t-(2T-1))$. To do that, in worst case, robots have to go from $x_i(t)$ through $x_i(t-(T-1))$, $x_i(t-T)$ and $x_i(t-(2T-1))$. Hence, they have to cover three steps in one, i.e., at triple speed.

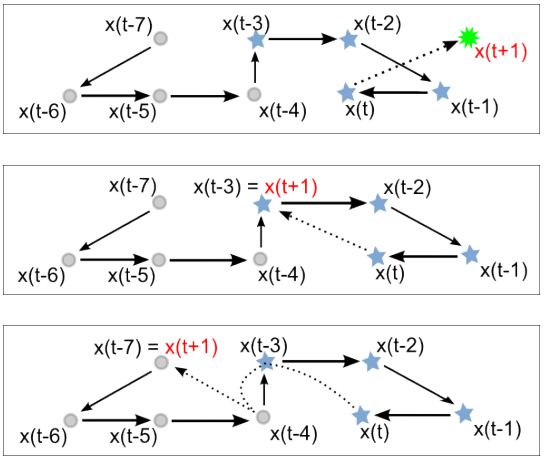


Fig. 3: *T–DISL* algorithm (with $T = 4$) at time step $t = k(T-1)$ when the *DISL* learning rule is applied. Top: robot i experiments a new action and will move toward $x(t+1)$. Center: the robot does not experiment and the trajectory of the last interval $[t-3, t]$ happens to be the maximizer of the utility function u_i^T. Bottom: the robot does not experiment but the trajectory maximizing the utility function u_i^T is the one of the second last interval $[t-7, t-4]$ whose initial configuration can be reached in at most three steps and hence in a one time step when triplicating the speed.

Under the given assumption it holds that CNE configurations are reached by the team following *T–DISL* algorithm. More formally it holds:

Theorem 2. *By applying the* T–DISL *algorithm, robots evolution converges to a steady state periodic trajectory, of period T, that is an optimizer of the index ϕ^T in* (10).

The proof of this Theorem and its extension to other dynamic learning algorithms are omitted for brevity.

B. Simulations of T–Algorithms in static environment

In this section the T–Algorithms performance is evaluated with respect to *DISL* and *PIPIP* performance in case of static environment. In such scenarios the interest function W is constant. Algorithms have been tested for different values of $T \in \{1, \ldots, 20\}$ and different number of robots $N \in \{5, 10, 15, 20\}$, in a convex area with 400 sub–regions. Moreover, for any values of N and T, 50 different initial conditions have been considered. The robot sensor footprint is a circle centred on the robot and of radius corresponding to the dimension of two sub–regions. The interest function 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^T defined in (8) where the ψ_i is described in (9).

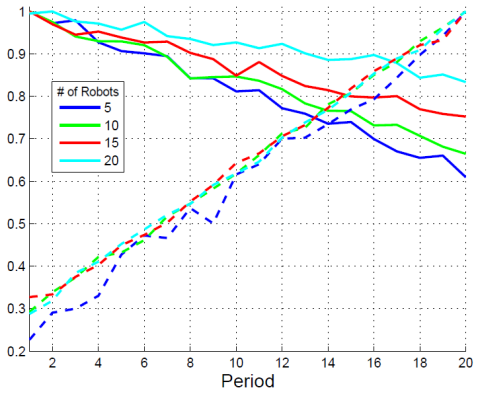


Fig. 4: The mean variation of the normalized number of covered sub–regions (dashed lines) and the normalized optimal values of the potential function (10) (continuous lines), over different values of the period T and different numbers of robots.

In order to compare T–Algorithms for different number of robots N and different values of T a normalization factor must be introduced. Indeed, for each value of N the minimum (maximum) mean number of covered sub–region is obtained for $T = 1$ ($T = 20$) while the opposite holds for the mean value of the potential function. The mean values for each N are hence normalized with the maximum obtained (in $T = 1$ for the potential function and $T = 20$ for the number of covered sub–regions). Referring to *T–DISL*, in Fig. 4 the mean variation of the normalized number of covered sub–regions (dashed lines) and the normalized optimal values of the potential function (continuous lines) for different values of the period T are reported. It is worth noting that, with respect to the original algorithm (i.e., $T = 1$) the proposed T–Algorithms provide an improving coverage (in terms of the number of covered sub–regions) by loosing (less) in optimality, when T grows up. Indeed, the potential optimal values decrease with lower rate with respect to the rate of increasing of the coverage. This behaviour depends on the chosen function ψ^T. Indeed, when the function $\psi^T \equiv 0$ robots do not get a reward if moving. In this case the number of coverage sub–regions does not increase and the potential optimal values is constant with $T > 1$. The same behaviour is obtained by *T–PIPIP* algorithms and hence their simulation results are omitted for space limitations.

This improvement in coverage with limited loss of optimality plays a crucial role in application scenarios characterized by dynamic environment, as described in next section.

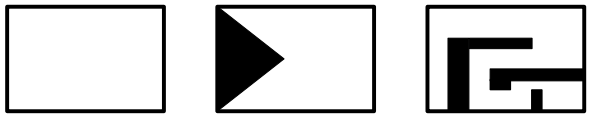
V. Learning algorithms in slow dynamic environments

The algorithms *DISL* and *PIPIP* described in previous section are designed for static environments. On the other hand, the proposed T–DISL and T–PIPIP algorithms, designed for dynamic environments, have been proved to converge for static ones. In many real applications scenarios the interest function is time–varying, e.g. intruders to be tracked can move in the environment or multiple intruders may appear at different time. It has been already proved that payoff–based algorithms naturally adapt to such environmental changes without altering action selection rules when prior knowledge on environments is not assumed [12]. In case of T–Algorithms, this statement is verified using a Gaussian density function whose mean changes in time, as interest function W. Moreover, in the following set of simulations, $W(t)$ is supposed to change any M time steps, i.e. $1/M$ may represents the intruder velocity. Since the evolution of the interest function $W(t)$ is unknown and unpredictable by robots, the use of T-Algorithms allows robots to maximize the reward received by $W(t)$ while optimizing the ψ index which can be designed to solve a different goal. Hence, in our scenario, robots may track an intruder while still performing an area patrolling task.

Simulation Setup

The algorithms have been tested for different changes of the W function (or intruder speed), $M \in \{100, 200, \ldots, 1000\}$, different number of robots, $N \in \{5, 10, 15, 20\}$, and different period, $T \in \{1, \ldots, 20\}$. The exploration rate $\epsilon = 0.1$ is constant and the irrational rate of the PIPIP algorithm is $\kappa = \frac{1}{3}$, as in test cases reported in [12]. T–Algorithms have been tested fully integrated with the well known Robot Operating System (ROS) [14], in order to consider different robot dynamics. Each simulation has been run for 20000 time steps. Moreover, for any values of M, N and T, 50 different 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^T defined in (8) where the ψ_i is described in (9). Three scenarios characterized by different degrees of convexity are considered (convexity measure is described in [21]), refer to Fig. 5 for the scenarios representation. The free space of each one 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. To represent the time variability of the environment, in the simulation, the following Gaussian density function whose mean $\mu(t)$ randomly changes every M steps have been chosen:

$$W(q) = e^{-\frac{||q-\mu(t)||^2}{2\sigma^2}}, \sigma = 5 \qquad (12)$$



(a) Open sea (b) Island (c) Harbor

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, [21].

Simulation Results

For the purpose of comparisons the following error index $\mathcal{I} \in [0, 1]$ is considered

$$\mathcal{I}(t) = 1 - \frac{1}{t}\sum_{k=1}^{t} \frac{\sum_{i \in V} u_i(k)}{\sum_{q \in \mathcal{Q}} W_q(k)} \qquad (13)$$

where u_i 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 on the whole environment) and the total benefit reached by the team (determined by the robot utility functions u_i). Hence, a null index corresponds to the ideal case in which each sub–region is covered by one and only one robot sensor footprint at any time instant.

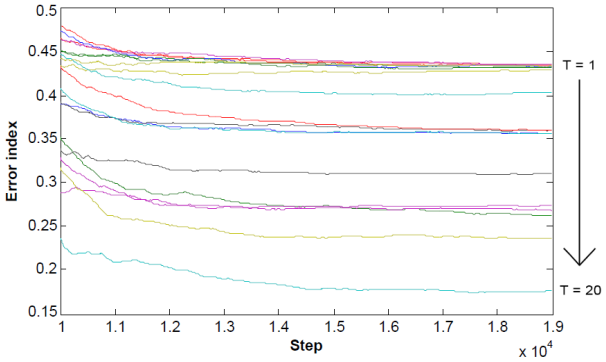
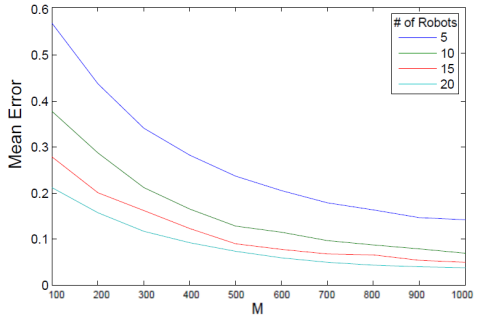
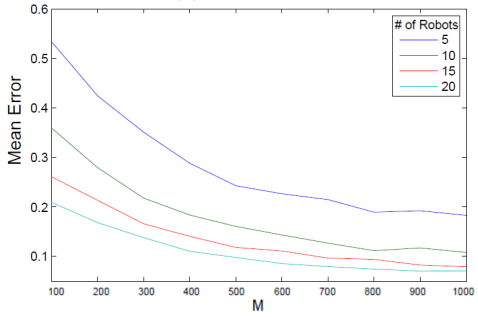


Fig. 6: Evolution of the T–*DHSL* mean value of the error index for different values of T in convex environment and $N = 5$ robots.

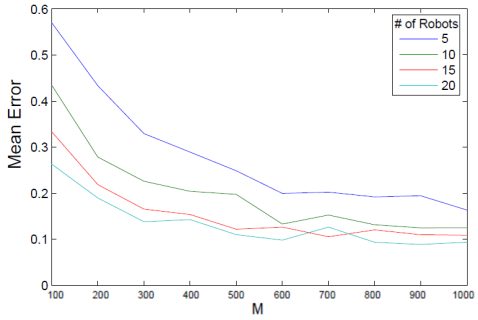
Fig. 6 reports the evolution of the mean value of the error index $\mathcal{I}$, in case of a convex scenario with $N = 5$ robots, for different values of T. In particular, the almost constant value of $\mathcal{I}$, for each T, corresponds to the fact that, although moving, robots gather around the region of higher interest that changes in time. For the purpose of comparison, the evolution of the mean value of the error index, for different values of M, N and scenarios, is reported in Fig. 7 for 10–*DHSL*. It can be noted that, for decreasing intruder velocities, the mean error decreases despite of the number of robots N and of the scenario convexity. On the other hand, lower convexity (that corresponds to a higher complexity) of the environment induces



(a) Open sea



(b) Island



(c) Harbor

Fig. 7: Evolution of the *T–DHSL* mean value of the error index for different M and N for different scenarios, when $T = 4$.

a larger value of the mean error and of its variance for any value of N. For space limitations results of *T–PHPIP* are omitted, however no great differences can be perceived in the simulation results with respect to the *T–DHSL* algorithm.

Simulations results confirm that robots executing T–Algorithms successfully adapt to the environmental changes without the necessity to change the action selection rule. Robots executing T–Algorithms with higher values of T are able to better detect environmental changes with respect to robots executing T–Algorithms with lower values of T. Concluding, simulation results confirm the validity of the proposed methodology to extend static algorithm in case of dynamic environments also in case of simultaneous possible antagonistic tasks.

VI. Conclusions

The problem of coordinating a system of underwater robots for protection against asymmetric threats has been modelled as a potential games. Two learning algorithms have been proposed based on trajectory optimization to allow the robots to manage different goals such as the tracking of the intruder and the patrol of the area. Convergence in probability toward Nash equilibria has been proved in static environment. Finally, for dynamic environments, the proposed algorithms have been evaluated in simulation in different scenarios.

The simulation results can be used to design the team (N) based on intruders characteristics (M) while maintaining a given level of performance.

The proof of convergence in case of dynamic environments is still an open problem. On the other hand, also the extension to the case of learning algorithm of different periods for different robots may be evaluated together with an automatic procedure for the choice of the best value of T for each robot.

References

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

[2] 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.

[3] 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.

[4] 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.

[5] 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.

[6] 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.

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

[8] 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.

[9] 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.

[10] 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.

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

[12] S. Nardi, C. Della Santina, D. Meucci, and L. Pallottino, "Coordination of unmanned marine vehicles for asymmetric threats protection," in *OCEANS 2015-Genova*. IEEE, 2015, pp. 1–7.

[13] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng, "Ros: an open-source robot operating system," in *ICRA workshop on open source software*, vol. 3, no. 3.2, 2009, p. 5.

[14] S. Nardi and L. Pallottino, "Nostop: an open source framework for design and test of coordination protocol for asymmetric threats protection in marine environment," in *MESAS 2016-Rome*. Springer, 2016.
[15] 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.
[16] T. Fabbri, R. Vicen-Bueno, R. Grasso, G. Pallotta, L. M. Millefiori, and L. Cazzanti, "Optimization of surveillance vessel network planning in maritime command and control systems by fusing metoc & ais vessel traffic information," in *OCEANS 2015-Genova*. IEEE, 2015, pp. 1–7.
[17] B.-N. Vo and W.-K. Ma, "The gaussian mixture probability hypothesis density filter," *IEEE Transactions on signal processing*, vol. 54, no. 11, pp. 4091–4104, 2006.
[18] M. I. Freidlin, J. Szücs, and A. D. Wentzell, *Random perturbations of dynamical systems*. Springer, 2012, vol. 260.
[19] D. Fudenberg and J. Tirole, "Game theory, 1991," *Cambridge, Massachusetts*, 1991.
[20] D. Monderer and L. S. Shapley, "Potential games," *Games and economic behavior*, vol. 14, no. 1, pp. 124–143, 1996.
[21] 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.