

A game theoretic robotic team coordination protocol for intruder herding

Simone Nardi¹, Federico Mazzitelli² and Lucia Pallottino³

Abstract—Intruder tracking and herding problems are crucial in several applications. In this paper a game theoretic coordination protocol for multi-mobile robot systems is proposed to tackle both problems simultaneously. Defender robots move according to computed Nash equilibria to herd the intruder into a safe area while preventing its access to one or more protected areas. The concept of a virtual barrier is presented to induce defenders to automatically and uniformly deploy along the barrier in order to drive intruder away from the protected areas and toward the safe one. Simulation results are reported to validate the proposed approach.

I. INTRODUCTION

Interaction of autonomous agents has attracted much interest from different disciplines and has been tackled with different approaches based on the particular goal of the agents. For example, the collective motion can be based on pure coordinated interactions, [1], [2], [3], [4] or on conflicting interactions as in case of the pursuit–evasion problem [5], [6], [7], [8], [9]. Differently, the proposed approach falls into the category of agents with conflicting interactions but focuses on a different aspect that is the *defense from intrusion*, i.e. a defense–intrusion interaction, see e.g. [10], [11], [12]. This kind of interaction is ubiquitous and fascinating both in the natural world (see, e.g., Figure 1) and the artificial one (e.g., a guarded vessel tries to protect an island by preventing approach of opposed vessels). Compared with conflicting interactions, in defense–intrusion interactions, intruders not merely escape from defenders but also try to approach a protected region. On the other hand, defenders not merely pursue the intruder but also aim at expelling it away from the protected area. The biological world provides a rich source of inspiration for this type of problems, see for example the hunting strategies adopted by dolphins [13]. With respect to defense–intrusion interaction, this work couples it with the intruder herding problem [14], [15]. In other words, we are interested in defining a coordination protocol for mobile robots that provides them the ability to track an intruder and to escort it to a predetermined safe area while keeping it far from a protected area even in presence of unknown and possibly moving obstacles in

the environment. A schematic description of the scenario is reported in Figure 2.



Fig. 1: Tracking and herding example in natural scenario. Image from Wikimedia Commons

Based on a distributed game theoretic approach [16], this work proposes a novel framework able to solve the problem of defense–intrusion interaction and intruder herding, in case of multiple autonomous defenders. The proposed framework has been chosen since it allows the coordination protocol to be applied also in scenarios where communication is forbidden, for example for security reasons or due to degraded communication channels, as in underwater environments. Indeed, the proposed approach differs from the one proposed in [2], [17] because it does not need any direct communication between defenders. Hence, each agent in the environment takes part to the game and takes autonomous decisions based on the system state information locally available. Finally, the coordination protocol has been designed to be independent from the robot kinematics and to be self adaptive to the variability of the environment.

Given a scenario with obstacles, protected and safe areas, approaches as those proposed in [18], [19] provide

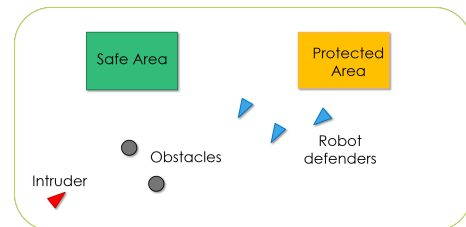


Fig. 2: Protected (yellow) and Safe (green) areas are characterized by their central position and shape. Intruder (red triangle) is located in the area and when detected by defenders (blue triangles) the herding game begins. Thanks to the proposed solution, obstacles (grey circles) do not compromise the success of the mission

¹Nardi is with Computational Geometry and Systems Laboratory, Ingegneria dei Sistemi SpA, Pisa, Italy

²Mazzitelli is with Aitronik, San Giuliano Terme, Pisa, Italy

³Pallottino is with Centro di Ricerca “E. Piaggio”, Dipartimento di Ingegneria dell’Informazione, Università di Pisa, Italy

an initial defenders deployment to detect an intruder. On the other hand, an intruder may have interest in accessing more than one protected area. Hence, once the intruder is detected, the herein proposed coordination protocol is not just used by defenders to track the intruder (see Section III) but also to identify its objectives in presence of more than one protected area (see Section III-C).

To summarize, this work proposes a game theoretic solution to the herding and tracking problem for a coordinated team of robots without the need of communication exchange. The proposed approach will be shown to work without a prior knowledge of the environment and on the area that will be attacked in case of more than one protected area. Based on local information the defenders will position themselves in order to prevent the intruder entering a protected area and to force the intruder toward a safe area while avoiding collisions with other robots and obstacles. The proposed coordination algorithm has been validated with numerical simulations following a Monte Carlo approach with different scenarios and different initial positions of the intruder and the defenders. The entire code is available to interested researchers¹.

II. PROBLEM FORMULATION

In this section, we briefly introduce, for reader convenience, the notation and the formalism used to define the coordination protocol designed for the tracking and herding problem. For the sake of simplicity we consider a 2D environment, however the protocol can be directly extended to 3D scenarios as in underwater or aerial applications.

Let N be the number of defender robots in a scenario in which a single intruder can enter. The configuration of every defenders, at time t , is:

$$\mathcal{D}_i(t) = [x_i, y_i, \theta_i] \quad \text{for } i = 1, 2, \dots, N.$$

Where x_i, y_i are the coordinates of the i -th defender expressed in a fixed reference system and θ_i represents the orientation of the i -th defender. Similarly the position and the orientation of the intruder are denoted with:

$$\mathcal{I}(t) = [x_{\mathcal{I}}, y_{\mathcal{I}}, \theta_{\mathcal{I}}].$$

As mentioned, the proposed coordination method is independent on the particular robot kinematics. We consider M , possibly moving, obstacles that have to be avoided by all the robots, defenders and intruder. Each obstacle's shape and dimension is supposed to be known when robots are sufficiently close. For simplicity, obstacles can be represented as discs and characterized by the center's position $(x_{\mathcal{O}}, y_{\mathcal{O}})$ and the radius $r_{\mathcal{O}}$:

$$\mathcal{O}_i = [x_{\mathcal{O}_i}, y_{\mathcal{O}_i}, r_{\mathcal{O}_i}] \quad \text{for } i = 1, 2, \dots, M.$$

Finally there are P protected areas represented by a known bounding box: for simplicity the area is considered

as a square i defined by its center position and sides' length $\mathcal{P}_i = [x_{\mathcal{P}_i}, y_{\mathcal{P}_i}, b_{\mathcal{P}_i}]$. Similarly, there is a single safe area, represented with its bounding box: $\mathcal{S} = [x_{\mathcal{S}}, y_{\mathcal{S}}, b_{\mathcal{S}}]$.

The robots are supposed to be equipped with two types of sensors able to provide the position and the orientation of the other robots, of the intruder and of the obstacles. These sensors are considered ideal (noise-free) but with limited range. The first type of sensors allows the estimation of other defenders position and orientation and the obstacles detection and localization. The sensors footprint is supposed to be a disc of radius r_c , called *coordination disc*. The second type of sensors allows the identification of non-cooperating robots (e.g., the intruder) when they enter the *identification disc* of radius r_{id} . Normally $r_{id} \leq r_c$ because it is easier to estimate states of a cooperating robots (e.g., [20]) rather than those of uncooperative ones (e.g., [21]). In the same way the intruder is supposed to be equipped with sensors able to estimate the position and the orientation of the defenders and with the same footprint. The description of the sensory systems to detect robots and obstacles are out of the scope for this work (for a brief introduction to technology refer to [20], [21]).

The goal of the intruder is to enter in one of the P protected areas while the goal of the defenders is to avoid that and to force the intruder entering in the safe area \mathcal{S} . All robots must avoid also collisions with the M obstacles, the defender must avoid collisions between each other. Robots are supposed to take decisions on the action to be performed based on the following information. Each defender is able to reconstruct information (e.g. dimension, position) on:

- other defenders in its coordination disc,
- obstacles in its coordination disc,
- intruder when it enters its identification disc,
- safe and protected areas.

The intruder is able to reconstruct information on:

- obstacles in its coordination disc,
- each defender in its identification disc,
- protected areas.

The intruder ignores the position of the safe area.

A. Notation

In the rest of the paper we use the term robot when we speak indifferently about a defender or the intruder; while we use the specific name when it is necessary to differentiate them.

- d_r^{ij} : distance between defender i -th and robot j -th, $j \in \{1, \dots, N, \mathcal{I}\}$,
- d_o^{ik} : distance between robot i -th and obstacle k -th, $i \in \{1, \dots, N, \mathcal{I}\}$
- $d_{\mathcal{P}}$: distance between intruder and protected area \mathcal{P} ,
- $\mathcal{O}^i = \{\mathcal{O}_k | d_o^{ik} \leq r_c\}$: the set of obstacles detected by robot $i \in \{1, \dots, N, \mathcal{I}\}$,
- $\mathcal{N}^i = \{j | d_r^{ij} \leq r_c, j \in \{1, \dots, N\} \text{ or } d_r^{ij} \leq r_{id} \text{ for } j = \mathcal{I}\}$: the set of indexes of robots in the sensor footprint of defender $i \in \{1, \dots, N\}$,

¹Code available at <https://github.com/SimoneNardi/game-theoretic-coordination-protocol>

- $\mathcal{N}^{\mathcal{I}} = \{j | d_r^{j\mathcal{I}} \leq r_{id}, j \in \{1, \dots, N\}\}$: the set of indexes of defenders in the sensor footprint of the intruder.

The linear velocity of each defender is supposed to depend on the intruder estimated position and speed, while the direction of motion is the outcome of the coordination protocol as described next. The idea is to set the defender velocity to move at maximum speed when the intruder is identified and thus reducing it linearly until the intruder velocity is reached. Such choice works under the typical assumption of a defender speed greater than the intruder one. The speed V_i of defender i is hence set equal to:

$$V_i = \begin{cases} \frac{v_{max} - v_{\mathcal{I}}}{r_{id}} d_r^{i\mathcal{I}} + v_{\mathcal{I}}, & \text{if } d_r^{i\mathcal{I}} \leq r_{id} \\ 0, & \text{otherwise,} \end{cases} \quad (1)$$

where $v_{\mathcal{I}}$ is the intruder linear velocity and v_{max} is the maximum velocity of the defender.

B. Game theoretic framework

In this paper we proposed to adopt a game theory based approach. Relevant concepts are briefly reported for reader convenience, for further details refer to [22].

A basic concept of the game theory used in this work is the *strategic game*, defined as follows:

Definition 1: A *strategic game* consists of

- a set of **players**
- each player has a set of **actions**
- each player has a **preference** over the set of available actions.

For the considered problem **each robot (defender or intruder) is a player** and the available **actions** to each player are chosen from the following set:

$$A_i = \left\{ (\theta_i, v_i) : \theta_i \in \Theta, v_i = \begin{cases} V_i, & \text{for defender} \\ v_{\mathcal{I}}, & \text{for intruder} \end{cases} \right\} \quad (2)$$

where, Θ is the set of admissible steering angles that depends on the robot kinematics. For example, in the performed simulations a finite set of three values have been chosen as Θ , due to computational limitation of the chosen platform (see Section IV). The linear velocity of each defender is derived in a deterministic manner, following equation (1) while the intruder velocity is estimated with on board sensors, e.g. using lightweight radar [21]. Hence, the role of the game in the control loop consists in choosing the appropriate steering angle θ_i .

The last element to define a strategic game are the functions that define the player's **preferences** over the actions profile, in this case expressed as cost functions. The cost function C_i of the i -th defender has the following structure: $C_i(\mathcal{A}^i) = f_i(\mathcal{A}^i, \mathcal{O}^i, \mathcal{P}_1, \dots, \mathcal{P}_P, \mathcal{S})$, where $\mathcal{A}^i = \{a_j | j \in \mathcal{N}^i\}$ is the set of actions of intruder and defenders detected by defender i , i.e. in its sensor footprint. The intruder's decision ($a_{\mathcal{I}}$) is considered in the computation of the game equilibrium, but that does not mean that the intruder actually will implement such

action. This will depend on the model of the intruder behaviour. For the considered problem, the cost function C_i of the i -th defender is:

$$C_i(\mathcal{A}^i) = \frac{K_1}{\min_{j \in \mathcal{N}^i, k \in \mathcal{O}^i} (d_r^{ij}, d_o^{ik})} + K_2 d_b^i \quad (3)$$

The proposed cost function has been chosen in order to penalize small distances between robots (or between robots and obstacles) and large distances, d_b^i , from the current configuration toward a possible defender target configuration. The target configuration can be defined in several ways, the one used to handle both the tracking and herding problem is defined in next section. K_1 and K_2 are gains used to tune the relevance of the two addends as shown in Section IV.

Assuming that the goal of the intruder is to reach the specific protected area \mathcal{P} among those in the environment, the intruder cost function is modeled as follow:

$$C_{\mathcal{I}}(\mathcal{A}^{\mathcal{I}}) = \frac{K_1}{\min_{i \in \mathcal{N}^{\mathcal{I}}, k \in \mathcal{O}^{\mathcal{I}}} (d_r^{i\mathcal{I}}, d_o^{ik})} + K_2 d_{\mathcal{P}} \quad (4)$$

where $\mathcal{A}^{\mathcal{I}} = \{a_j | j \in \mathcal{N}^{\mathcal{I}}\}$ is the set of intruder actions, and $d_{\mathcal{P}}$ is the distance between the intruder and the protected area \mathcal{P} , that is the intruder objective.

The action performed by the team of defenders and the one performed by the intruder are chosen based on Nash equilibria [23]. It is worth noting that an equilibrium corresponds to a local minimum of the players' cost functions. Moreover, among others, the Nash equilibrium is considered in this paper because its computation does not require communication between players. For the sake of clarity, in the reported definition we choose to consider an action set containing the actions of all robots neglecting the actual dependency from distances.

Definition 2: A decision set $(a_1^*, a_2^*, \dots, a_N^*)$ is a *Nash equilibrium* if it satisfies the following inequalities for $i = 1, \dots, N$:

$$\begin{aligned} C_{\mathcal{I}}(a_{\mathcal{I}}^*, a_1^*, \dots, a_N^*) &\leq C_{\mathcal{I}}(a_{\mathcal{I}}, a_1^*, \dots, a_N^*) \\ C_i(a_{\mathcal{I}}^*, a_1^*, \dots, a_i^*, \dots, a_N^*) &\leq C_i(a_{\mathcal{I}}^*, a_1^*, \dots, a_i, \dots, a_N^*) \end{aligned}$$

At each iteration each agent computes the set of Nash equilibria $(a_{\mathcal{I}}^*, a_1^*, a_2^*, \dots, a_N^*)$. Since it has to be guaranteed that a solution is found, the problem is solved by finding the Nash equilibrium in mixed strategies. The mixed Nash equilibrium is an optimal probability distribution over the action space. However, in real applications only pure actions can be performed and not mixed ones, therefore it has been chosen a pure action with maximal probability distribution. In case of more than one Nash equilibrium, available equilibria are sorted based on efficiency (in terms of the cost function) and the least expensive one is implemented. With this approach, even in case of no communication, each defender chooses to implement the action leading to the same equilibrium.

III. INTRUDER TRACKING AND HERDING PROBLEM

The Nash Equilibrium of the strategic game described above contains the action that will be performed by robots and strongly depends on the cost functions. The adaptation of the cost functions in equation (3) to the particular case of the tracking and herding problem is the focus of this Section. Indeed, the objective of the defenders is not only to defend the protected areas, but also, at the same time, to herd the intruder into the safe one that is unknown to the intruder. The introduction of multiple objectives for both the intruder and the defenders is one of the main differences between our approach and the one proposed by Alexopoulos et al. in [5]. For this purpose the defenders can move close to the intruder forcing it toward the safe area and this will be pursued assigning a target configuration to the defenders in (3). A virtual target set is hence proposed to deploy the defenders. Each target is characterized by a position and an orientation close to the intruder so that a barrier is created with the effect to push it toward a desired direction (see Figure 3). The barrier region is computed by each defender only based on intruder's position information and safe and protected areas. Since those are information available to all defenders with the intruder in their footprint, the same barrier region is obtained by each defender. The proposed solution is based on control radius system proposed in [24] for non-cooperative agents. When sensor measurement noise results in variation of the computed barrier region among defenders the proposed solution can be also applied. Indeed, the solution needs to impose some limitation in the sensor measurement errors to be effective. However, no positions within the virtual barrier are pre-assigned to defenders: each defender must *obtain* a place within it, based on the cost function previously defined, see equation (3). Each defender will arrange themselves along the barrier avoiding collisions. In case the virtual barrier overlaps an obstacle, the defenders are re-arranged along the collision free portion of the barrier so that obstacles are exploited to ensure a higher intruder reward toward the safe area. Hence, the herding is ensured also in presence of obstacles.

A. Virtual barrier construction and orientation

The virtual barrier has the scope to force the intruder moving toward the safe area while staying far from the protected ones. Hence, the virtual barrier is built based on following unit vectors (see Figure 3 for graphic representation):

- $\bar{\mathcal{P}}$ is a unit vector, applied on the intruder position, that lies along the line between the intruder and the associated protected area (\mathcal{P}) with direction toward the area itself.
- $\bar{\mathcal{S}}$ is a unit vector, applied on the intruder position, that lies along the line between the intruder and the safe area (\mathcal{S}) in direction opposite to the area itself.

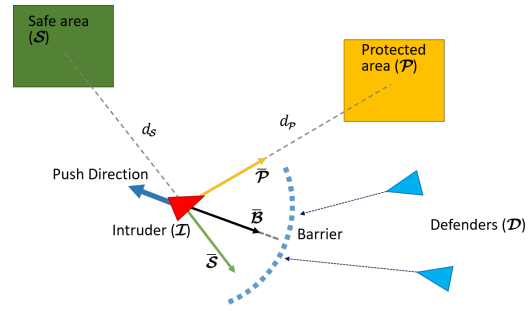


Fig. 3: Vectors and parameters to identify the barrier

- $\bar{\mathcal{B}}$ is a unit vector, applied to the intruder position, that lays between $\bar{\mathcal{S}}$ and $\bar{\mathcal{P}}$ directed toward the barrier. This vector is the one characterizing the center of the barrier and will be chosen as a convex combination of $\bar{\mathcal{P}}$ and $\bar{\mathcal{S}}$.

It is worth noting that the distances between the robot and the closest point to the area can be used instead of the one toward its center.

More formally, $\bar{\mathcal{B}} \doteq \bar{\mathcal{S}}w + \bar{\mathcal{P}}(1-w)$, where parameter $w \in [0, 1]$ determines the orientation of the barrier vector $\bar{\mathcal{B}}$. Distances from areas can be taken into account in the orientation of the barrier choosing, for example: $w \doteq \frac{d_P}{(d_S + d_P)}$, where d_P and d_S represent the distance between the intruder and the associated protected area and the distance between the intruder and the safe area respectively. If $w = 1$ then $\bar{\mathcal{B}} = \bar{\mathcal{P}}$, while if $w = 0$ then $\bar{\mathcal{B}} = \bar{\mathcal{S}}$. The behaviour of the barrier can be easily summarized as follows: when the intruder is very close to the protected area, regardless of the distance from the safe one, the barrier will be oriented to defend the protected area by positioning the defenders between the area and the intruder, ($\bar{\mathcal{B}} \approx \bar{\mathcal{P}}$). On the other hand, when the intruder is close to the safe area, regardless of the distance from the protected one, the barrier will be oriented so that defenders are deployed guaranteeing the intruder between themselves and the safe area, ($\bar{\mathcal{B}} \approx \bar{\mathcal{S}}$). In all other cases, the barrier will be oriented in intermediate positions depending on the distance of the intruder from the protected and the safe areas.

In order to increase the efficiency of the defenders pressure over the intruder, an additional term is added to the defenders cost function, w.r.t. (3), as follows:

$$C_i(\mathcal{A}^i) = \frac{K_1}{\min_{j \in \mathcal{N}^i, k \in \mathcal{O}^i} (d_r^{ij}, d_o^{ik})} + K_2 d_b^i + K_3 \gamma_i, \quad (5)$$

the idea is to use d_b^i (the distance between the current position of the i -th defender and the nearest point on the barrier) to push the defender to position itself on the barrier, while γ_i is used to push it moving toward the central position of the barrier. $\gamma_i \in [0, 1]$ is given by $\gamma_i = \frac{|\beta - y_i|}{\beta}$ for $y_i \in [0, 2\beta]$, where, referring to Figure 4, β is the barrier half-extension angle and y_i is the angle of the i -th defender position projected on

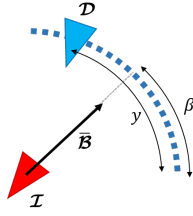


Fig. 4: Defender position on the barrier for associated cost computation

the barrier. With this choice, the cost is minimized for $\gamma_i = 0$ corresponding to $y_i = \beta$, in other words when the defender is on the central position of the barrier, i.e. when the pushing effect of the barrier is maximized.

It is worth noting that, the position of each defender over the barrier autonomously varies based on the number of robots participating in the pursuit. Indeed, each defender tries to reach a position on the barrier (second addendum of the cost function (5)), to reach the central position on the barrier (third addendum) while avoiding collisions with other defenders (first addendum).

B. Coordination algorithm

The problem of team coordination for intruder tracking and herding reveals possible conflicting defender's objectives between area protection, intruder herding and collision avoidance with other defenders. The proposed barrier based approach requires each defender to compute its own cost function based on local sensory information: defenders, obstacles and intruder's positions. Intruder can be detected based on a monitoring algorithm such as those proposed in [18], [19] and referred here as *SearchIntruder* procedure. Whenever a defender detects the intruder in the identification disc (*UpdateIntruder*), it estimates its neighbouring defenders configurations in the coordination disc (*FindNeighbours*), builds the virtual barrier, as described in Section III-A, (*CompBarrier*) to identify its target region, computes Nash equilibria of the game with the other players, i.e., the robots in its coordination disc (*CompNashEq*), and selects the most efficient action (*SetSpeed*). Thus, sensory information is updated (*UpdateIntruder*) and the procedure repeated. The Mission Control procedure of each defender, reported in Algorithm 1, monitors the tracking and herding mission success and it is based on the Action Selection procedure, reported in Algorithm 2. Procedures use the following notation:

- ID_c and D_c : initial and current defender position and orientation in the Cartesian space,
- *Protect*: protected area shape and center position,
- *Safe*: safe area shape and center position,
- I_c : intruder position and orientation,
- Ds_c : set of the other defenders' configurations.

The procedure in Algorithm 2 is used by the defenders to select the next action to be performed. In case of

Algorithm 1: Mission Control

Data: $ID_c, Protect, Safe$
Result: Mission success

- 1 $D_c \leftarrow ID_c, I_c \leftarrow Undefined$;
- 2 **while** $I_c \notin \{Safe\}$ **do**
- 3 $I_c \leftarrow UpdateIntruder(D_c)$;
- 4 **if** $I_c \neq Undefined$ **then**
- 5 **if** $I_c \in \{Protect\}$ **then**
- 6 return false;
- 7 $D_c \leftarrow ActionSelection(D_c, Protect, Safe, I_c)$;
- 8 **else**
- 9 $D_c \leftarrow SearchIntruder(D_c)$;
- 10 return true;

Algorithm 2: Action Selection

Data: $D_c, Protect, Safe, I_c$
Result: Update defender configuration D_c

- 1 $Ds_c \leftarrow FindNeighbours(D_c)$;
- 2 $Barrier \leftarrow CompBarrier(I_c, Ds_c, D_c, Protect, Safe)$;
- 3 $Move \leftarrow CompNashEq(I_c, Ds_c, Barrier, D_c)$;
- 4 $D_c \leftarrow SetSpeed(D_c, Move, Barrier)$;

unicycle-like kinematics this is the turn angle R_i . Once action has been selected, the state update equation of i -th defender, with discretization time Δt , is:

$$D_i(t + \Delta t) = \begin{bmatrix} x_i(t) \\ y_i(t) \\ \theta_i(t) \end{bmatrix} + \begin{bmatrix} \cos(\theta_i(t) + R_i) \\ \sin(\theta_i(t) + R_i) \\ 0 \end{bmatrix} V_i \Delta t + \begin{bmatrix} 0 \\ 0 \\ R_i \end{bmatrix}$$

C. Intruder objectives identification

The proposed virtual barrier has been designed to defend only one protected area. In order to manage multiple protected areas, each defender, during the intrusion tracking phase, must be able to deduce intruder's intention, i.e. the protected area the intruder is trying to reach, so that the proposed virtual barrier method can still be used. This is possible since the computed Nash equilibrium strategy contains information about the best action of each player, intruder included. Hence, each defender can compare the action performed by the intruder with the computed equilibrium deducing the intruder objective. This is possible under the assumption that the intruder is optimizing a cost function that minimizes the distance toward the desired protected area and maximizes the distance from defenders as considered in equation (4). In our solution, defenders solve a game for each protected areas with the same set of players. The intruder interest on an area is evaluated based on its last k actions (with $k = 10$ in our simulations). Intruder's performed k actions are compared by the defender with the actions computed for each performed game in each of the k time steps. The area with maximum actions correspondence is the one chosen, by the defender, to be protected. The limited number of evaluated actions k

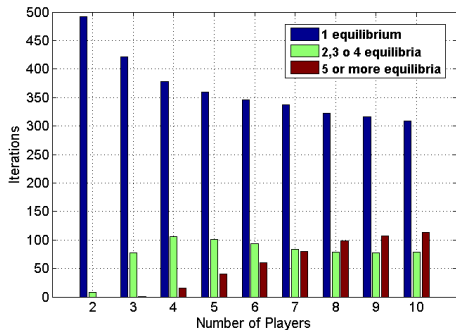


Fig. 5: Number of Nash equilibria on 500 simulations for different numbers of players

is motivated by a possibly limited memory and by the fact that, a priori, the intruder may prefer to change the area to reach as in our experiments when the intruder is guided by humans.

Based on simulation results reported in Figure 5, the number of Nash equilibria is low and it tends to increase with the number of players. In this set of Monte Carlo simulations, 500 defender initial positions are randomly selected for each configuration with assigned number of players $N \in \{2, \dots, 10\}$. The environment considered is a free square discretized in 1000×1000 pixels. Simulation results show that the intruder objective is correctly estimated after few iterations, 2 or 3 iterations.

Managing multiple protected areas has higher computational cost than the single protected area case. Indeed, each defender has to compute the Nash equilibrium of one game for each protected area. The problem hence requires an appropriated approach in case of large numbers of protected areas and this is out of the scope of this paper. Simulation results in case of few protected areas are reported in next section.

An example of the evolution of the system with two defenders and two protected areas is reported in Figure 6. The intruder starts moving toward the blue area, once it enters the identification disc of a defender (defender 2) it tries to identify the intruder's objective. Once identified the desired protected area, the defenders successfully herd the intruder toward the safe area.

IV. SIMULATION RESULTS

The robustness of the algorithm has been verified with the use of Monte Carlo simulations. A square environment is chosen, discretized with 1000×1000 pixels. Each pixels with no obstacles, represents an available position for the robots. The scenario on which the approach is validated is composed by two protected areas (the blue and violet squares respectively of Figure 7 and a single safe area (the green square). In a first set of simulation defenders knows the area in which the intruder will try to enter while in the second set the desired area is unknown a priori. Simulations have been performed with different defenders initial positions.

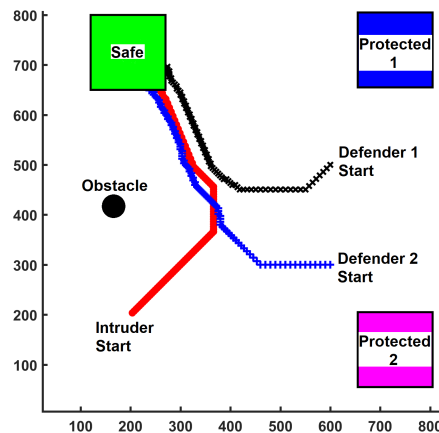


Fig. 6: Trajectories performed by defenders (blue and black) and intruder (red), when the intruder objective is the protected area number one

Nash equilibria are computed with the tool Gambit [25]: a multi-platform and open-source software. Gambit is a set of software tools for computation of finite, non-cooperative games. The solver receives input from defenders and intruder, processes the players payoff according to the surrounding environment information, and computes Nash equilibria of the game.

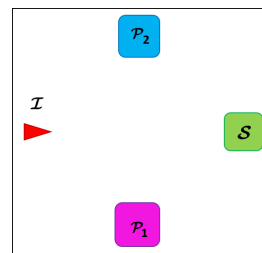


Fig. 7: Multiple protected areas environment

Each simulation returns: **success** when the intruder enters the safe area without touching the protected one, **failure** when the intruder enters the protected area and **deadlock** when the simulation exceeds 1000 iterations. In the following simulations each robot (defenders and intruder) can choose action in the set $A = \{\pm 45^\circ, 0^\circ\}$. Moreover, in order to prevent possible collisions each agent can limit its own set of actions removing those that lead to a conflict.

The performance of the approach depends on the Nash equilibrium solver. Computational cost increases exponentially with the size of the game (number of defenders, available actions and number of protected areas). For this reason, software optimization and high computational power are necessary in order to execute the proposed solution in a more realistic scenarios. However, the numerical simulations performed on a single core system, show the applicability of the proposed solution to very crowded environments with the use of a team of 10 defenders. Indeed, only the defenders sufficiently close to

the intruder are involved in the game and, in most cases, only three defenders have been proved to be sufficient to herd the intruder keeping the other out of the game. Simulations in more complex scenarios and with a large number of defenders are reported in the video attached to the paper.²

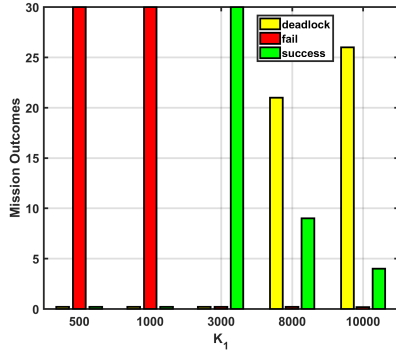


Fig. 8: Mission outcomes for different values of K_1

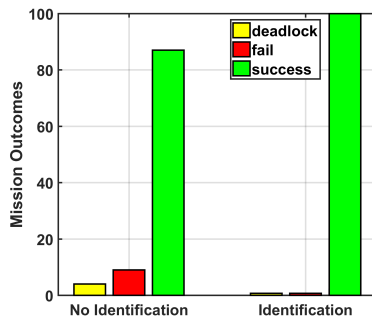


Fig. 9: Performance in case of multiple protected areas

A. Parameters tuning

Given the selected scenario, the parameters gain of the cost function defined in equation (5) has been tuned based on the motion ability of the intruder and on the considered scenario. The simulation results of the performance variation (in terms of successful simulations) with respect to the collision avoidance gain K_1 is reported in Figure 8 for 30 different missions for each value of parameter K_1 . Low values of parameter K_1 correspond to a low influence of the collision between players (between defenders and between a defender and the intruder) in the cost function. Hence the intruder may have higher payoff moving toward the protected area also in case of a nearby defender, leading to a failure in the proposed coordination protocol. On the other hand, for higher values of parameter K_1 , coordination algorithm success rate increases. However, if too large values of parameter K_1 are chosen the number of deadlock situations increases (yellow column) while the number of success decreases.

²https://youtu.be/eS_HjBaNxkCQ

Indeed, a high weight of the collision avoidance may lead to defenders keeping a higher distance between each other and from the intruder, preventing the defender to reach the virtual barrier and failing in forcing the intruder toward the safe area. Concluding, tuning of the cost function gains has been achieved experimentally through a series of simulations varying the parameters K_1 , K_2 and K_3 following procedure described in [16].

B. Results

To test and validate the proposed tracking and herding solution, different intruder behaviours may be considered. For space limitations we report simulation results of the most challenging scenario that is the case of an intelligent intruder that moves according to the Nash Equilibrium of the game against the defender.

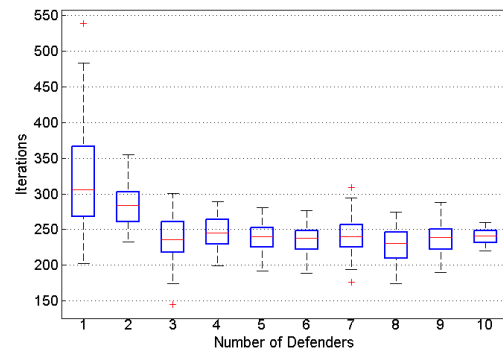


Fig. 10: Number of iteration for the success of the tracking and herding mission for different number of defenders

Figure 10 highlights the number of iterations necessary for herding success in case of different number of defenders, when the parameter K_1 is set to 3000, as suggested by results reported in Figure 8. It is worth noting that the number of iterations decreases when the number of defenders is larger than 1. This fact responds to the conviction that a coordinated team is able to complete a mission better than a single defender.

A second set of simulations has been conducted to evaluate the performance of the intruder objective identification approach in case of multiple protected areas and a priori unknown intruder objective. In this case, each of the two defenders is equipped with the same sensors as in the previous simulations set. A set of 100 different initial position of each defender is considered. Comparisons between the performance of the coordination protocol with or without the intruder objective identification procedure is shown in Figure 9. When the identification algorithm is not used each defender is responsible of one protected area and creates a different virtual barrier, based on the protected area assigned to it. In this case, in more than 10% of simulated missions the defenders fails in herding the intruder. On the other hand, when identification

algorithm is used, defenders are able to correctly identify the intruder objective and then they cooperate to push intruder towards the safe area. Indeed performance result increases up to the 100% of success.

The proposed coordination protocol, involving two defenders, has been finally validated against an intruder piloted by a human operator. The human intruder behaviour is modeled as a smart intruder where gains of cost function are tuned as in previous set of simulations. Experiment results show that every human operators have not been able to avoid to be herd towards the safe area (whose position is unknown to them) in the 100% of the missions. Note that in this case when the intruder collides with a defender the herding mission is considered a successful mission.

V. CONCLUSIONS

In this paper a game theoretic coordination algorithm for a multi-robot system is presented. The purpose of the team coordination is to defend one or more protected areas from an intruder and then herd it to a predefined safe area. A game theoretic approach has been chosen to solve the considered problem. The proposed framework is independent from the robot kinematics and can be used in environments with obstacles. The coordination between the members of the team of defenders occurs without any direct communication, thanks to the introduced concept of virtual barriers. In case of multiple protected area and unknown intruder objective a modified protocol is proposed. The protocols have been tested and validated with Monte Carlo simulations for different intruder's behaviour.

As future developments, it would be helpful to determine the minimum number of defenders necessary to successfully herd the intruder. Finally, the short prediction horizon used in the proposed game theoretic protocol could be relaxed to possibly improve performance.

REFERENCES

- [1] I. D. Couzin, J. Krause, N. R. Franks, and S. A. Levin, "Effective leadership and decision-making in animal groups on the move," *Nature*, vol. 433, no. 7025, pp. 513–516, 2005.
- [2] K. D. Do, "Output-feedback formation tracking control of unicycle-type mobile robots with limited sensing ranges," *Robotics and Autonomous Systems*, vol. 57, no. 1, pp. 34–47, 2009.
- [3] X. Liang and Y. Xiao, "Studying bio-inspired coalition formation of robots for detecting intrusions using game theory," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 40, no. 3, pp. 683–693, 2010.
- [4] M. Innocenti, L. Pollini, G. Franzini, and A. Salvetti, "Swarm obstacle and collision avoidance using descriptor functions," in *Control Applications (CCA), 2016 IEEE Conference on*. IEEE, 2016, pp. 487–492.
- [5] A. Alexopoulos, T. Schmidt, and E. Badreddin, "Cooperative pursue in pursuit-evasion games with unmanned aerial vehicles," in *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*. IEEE, 2015, pp. 4538–4543.
- [6] R. Escobedo, C. Muro, L. Spector, and R. Coppinger, "Group size, individual role differentiation and effectiveness of cooperation in a homogeneous group of hunters," *Journal of the Royal Society Interface*, vol. 11, no. 95, p. 20140204, 2014.
- [7] S. Keshmiri and S. Payandeh, "On confinement of the initial location of an intruder in a multi-robot pursuit game," *Journal of Intelligent & Robotic Systems*, pp. 1–29, 2013.
- [8] M. Haque, A. Rahmani, and M. Egerstedt, "A hybrid, multi-agent model of foraging bottlenose dolphins," *IFAC Proceedings Volumes*, vol. 42, no. 17, pp. 262–267, 2009.
- [9] C. Undeger and F. Polat, "Multi-agent real-time pursuit," *Autonomous Agents and Multi-Agent Systems*, vol. 21, no. 1, pp. 69–107, 2010.
- [10] J. Wang and W. Li, "Motion patterns and phase-transition of a defender–intruder problem and optimal interception strategy of the defender," *Communications in Nonlinear Science and Numerical Simulation*, vol. 27, no. 1, pp. 294–301, 2015.
- [11] G. E. Mullins and S. K. Gupta, "Adversarial blocking techniques for autonomous surface vehicles using model-predictive motion goal computation," in *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*. IEEE, 2015, pp. 2272–2278.
- [12] H. Huang, J. Ding, W. Zhang, and C. J. Tomlin, "Automation-assisted capture-the-flag: A differential game approach," *IEEE Transactions on Control Systems Technology*, vol. 23, no. 3, pp. 1014–1028, 2015.
- [13] B. L. Sargeant, J. Mann, P. Berggren, and M. Krützen, "Specialization and development of beach hunting, a rare foraging behavior, by wild bottlenose dolphins (*tursiops* sp.)," *Canadian Journal of Zoology*, vol. 83, no. 11, pp. 1400–1410, 2005.
- [14] A. D. Khalafi and M. Toroghi, "Capture zone in the herding pursuit evasion games," *Applied Mathematical Sciences*, vol. 5, no. 39, pp. 1935–1945, 2011.
- [15] S. Gade, A. A. Paranjape, and S.-J. Chung, "Robotic herding using wavefront algorithm: Performance and stability," in *AIAA Guidance, Navigation, and Control Conference*, 2016, p. 1378.
- [16] I. Harmati and K. Skrzypczyk, "Robot team coordination for target tracking using fuzzy logic controller in game theoretic framework," *Robotics and Autonomous Systems*, vol. 57, no. 1, pp. 75–86, 2009.
- [17] O. Purwin, R. D'Andrea, and J.-W. Lee, "Theory and implementation of path planning by negotiation for decentralized agents," *Robotics and Autonomous Systems*, vol. 56, no. 5, pp. 422–436, 2008.
- [18] 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.
- [19] S. Nardi, T. Fabbri, A. Caiti, and L. Pallottino, "A game theoretic approach for antagonistic-task coordination of underwater autonomous robots in asymmetric threats scenarios," in *OCEANS 2016-Monterey*. IEEE, 2016.
- [20] H. Li, C. Hua, C. Chen, and X. Guan, "Ads-b aided robust relay selection for cooperative communications in aircraft approach," *International Journal of Communication Systems*, vol. 29, no. 1, pp. 113–129, 2016.
- [21] L. Yang, S. Zhou, L. Zhao, and G. Bi, "A data-driven approach for monitoring forward velocity for small and lightweight drone," in *Aerospace Electronics and Remote Sensing Technology (ICARES), 2015 IEEE International Conference on*. IEEE, 2015, pp. 1–6.
- [22] M. J. Osborne and A. Rubinstein, *A course in game theory*. MIT press, 1994.
- [23] Y. Shoham and K. Leyton-Brown, *Multiagent systems: Algorithmic, game-theoretic, and logical foundations*. Cambridge University Press, 2008.
- [24] A. Pierson and M. Schwager, "Bio-inspired non-cooperative multi-robot herding," in *Robotics and Automation (ICRA), 2015 IEEE International Conference on*. IEEE, 2015, pp. 1843–1849.
- [25] McKelvey, D. Richard, McLennan, M. Andrew, Turocy, and L. Theodore. (2014) Gambit: Software tools for game theory, version 16.0.0. [Online]. Available: <http://www.gambit-project.org>.