Local interaction and navigation guidance for hunters drones: a chase behavior approach with real-time tests

C. de Souza, Pedro Castillo Garcia, Boris Vidolov

Robotica () © Cambridge University Press
doi:

Local interaction and navigation guidance for hunters drones: A chase behavior approach with real-time tests

C. de Souza Jr, P. Castillo, B. Vidolov* †

(Accepted MONTH DAY, YEAR. First published online: MONTH DAY, YEAR)

SUMMARY
A behavioral-based strategy for cooperative hunting using drones is proposed in this paper. In this decentralized scheme, each drone acts as an individual agent computing its guidance strategy toward the target based on the relative position of its neighbors without the use of direct communication. The algorithm is based on the Deviated Pure Pursuit methodology, and the emerged behavior mimics a natural hunting formation. Simulations and real-time experiments with varying conditions were carried out to validate the effectiveness of the proposed hunting scheme. Videos of the system in action can be seen on: https://youtu.be/g2dODbd6ZLA.

KEYWORDS: hunter drones, autonomous navigation, tracking, pursuit, real-time tests.

1. Introduction

The study and analysis of pursuit applied to mobile robots has been gaining the attention of the research community, motivated by the increasing popularity of small robots, especially drones. The need for surveillance of civil areas has motivated the application of classical war-like pursuit techniques, see refs. [1–5]. Nevertheless, using a single agent for pursuit has several limitations, such as when the target has a velocity equal or superior to the pursuer, or a chaotic behavior. A recent alternative to overcome these limitations is the use of multiple pursuers; see refs. [6–9], where it is possible to increase the spatial distribution of the agents to allow for a more fault-tolerant system. However, with an increase in the number of pursuers, the task increases in complexity - as the problem now includes issues such as collision-free navigation, formation control, task, and roles allocation. Many researchers approach this complex task in an approach base on collective hunters in nature refs. [6, 7, 10–13].

The algorithms found in the literature can be distributed in four main categories: pursuit, game theory, swarming, and bio-inspired group chasing.

1.1. Pursuit

The pursuit-evasion phenomena have been observed in almost all biological relations and have been traditionally analyzed mathematically and applied to modern humans. Since the classical and traditional analyses of Pierre Bourger, in *Les courbes de Poursuite* (1732), where he describes the interception of merchant's vessels by pirates ship, many pursuit techniques have been formalized and implemented. Classical guidances laws, such as the Deviated Pursuit Guidance (DPP), the Proportional Navigation Guidance (PNG), and its variations, have been used notably in missile guidance; see refs. [14, 15].

* C. Souza Jr , P. Castillo, B. Vidolov are with the Université de Technologie de Compiègne, CNRS, Heudiasyc (Heuristics and Diagnosis of Complex Systems), CS 60 319 - 60 203 Compiègne Cedex , France. Emails: (crdesouz, castillo, bvidolov)@hds.utc.fr
† Research supported by the Ministère de l'Education Nationale, de l'Enseignement Supèrieur et de la Recherche, and National Network of Robotics Platforms (ROBOTEX), from France.

These guidance laws have been extensively explored for ground robots, see refs. [16–18]. In ref. [16], the authors explored the use of geometric rules to navigate a single agent towards a target with unknown movements and in the presence of obstacles. They applied two navigation approaches, the velocity pursuit guidance (a special case of PNG) and the DPP methodology. In ref. [17], the authors implemented a controller that keeps a constant line-of-sight from pursuers to the target, using the principle of parallel navigation to ensure, under some conditions, the rendezvous course. Similarly, in ref. [18], the authors applied a sliding-mode-based control to keep a constant line-of-sight.

In ref. [2], the author applied PP and PNG guidance algorithms into quadcopters for target interception. They compared those techniques against an optimal control trajectory planner, analyzing the time and energy required to achieve the task. However, results are only presented in simulations. In ref. [5], the author analyses three missile guidance laws in the case of a partially observable target. They formulate a pursuit-evasion game to determine the guarantees of capture under evasive maneuvers, deriving an optimal controller. However, during the approaching phase, the PP algorithm is used.

The classic guidance navigation approaches have been useful in more applications than the pursuit problem; for example, in ref. [19] the authors used the PNG scheme in a drone landing on a moving platform. Their final solution mixed PNG with proportional derivative (PD) algorithms, effectively completing the task with less oscillation and smaller errors than traditional tracking PD. The proportional navigation approach has also been used as guidance law in autonomous racing of drones in ref. [20]. In that case, the authors adapted the PNG algorithm to handle the decoupled dynamics of the quadrotor.

1.2. Differential Games

Recently, the research community has been interested in a more complex version of the pursuit problem, increasing the number of pursuers and evaders. This can be modelled using a differential game, which uses game theory to model conflict in the context of a dynamic system.

In ref. [21], the authors used a differential game formulation for multi-pursuers chasing a single evader. In their approach, each agent minimizes the action area of the evader, which is characterized by the Voronoi cells. Similarly, in ref. [22], a cost function is proposed and computed based on the relative angle and distance between agents and target. These methodologies guarantee the optimality input for pursuit. However, the large amount of assumptions and precise knowledge needed about the whole system makes it infeasible for application in the real-world.

1.3. Multi-agents

The benefits of the group work against the individual alone can be observed in many social behaviors in nature. For example, in ref. [23], the authors completed quantitative analysis of the probability to survive from predators, feeding, reproducing, and living in a group; similarly, they have discussed how hierarchical systems emerge.

In ref. [24], the authors analyzed the cooperative chase with lionesses. Although lions commonly chase alone, they can cooperate - which increases hunting success. The formation of the lions, with center and wings lions, has inspired our approach.

The attempt to use a group of robots to stalk a moving target started early, basing on the extensive multi-agents formation theory, and using a feedback controller to haunt tasks and keep formation; see refs. [7, 25, 26].

Although the hunting task can be performed in argued to be completed in all above cases, all of them rely on fixed encirclement points around the target and keeping formation, making the system not so flexible to a dynamical and real-world application. Some improvements to these strategies have been made in refs. [8, 27], although these propositions do not rely on keeping formation, they require fixed encirclement points around the target.

These points are virtual and global information, which requires, for instance, the centralization of information. Nevertheless, the complexity of these algorithms, which require optimization for path planning, is not scalable (in both cases, only four agents were used) and would be unfeasible to implement in a bigger group.

1.4. Bio-inspired group chase

Another relevant field to group pursuit is artificial swarming and flocking. This approach considers each agent as an autonomous entity, following similar rules based on local interaction, resulting in emergent collective behaviors. This methodology is based on theoretical work as in refs. [28, 29], where the authors proposed complex collectives behaviors, for flocking and herding, starting from simple local rules.

Several works consider the agent-based model to represent multi-agent pursuit; in ref. [12], the author designed an algorithm based on the Vicsek particle model. In ref. [13] the authors modeled a predation system using agents inspired by swarms of bats. In ref. [30], the authors studied the emergent behavior of agents with an interaction between pursuers in a discrete environment (two-dimensional square lattice). The simplicity of the model allowed simulations with large numbers of players (e.g. $10e^4$). In ref. [31], the hunting behavior of agents is studied in a more realistic environment using a framework to simulate a fleet of UAVs (see ref. [32]). In this methodology, the interaction of each agent is represented by physical forces in bi- and tri-dimensional space with boundary conditions. The action of each agent is calculated by the sum of different interaction factors: chase, inter-agent repulsion, alignment and, short-range collision. The 'chase' factor assumes access to information about the position and velocity vector of the evader, this allows the agent to predict of the future position of the target. Although the work showed a powerful algorithm to analyze realistic pursuit with robots, they have strong assumptions, and neglected the limited perception of the robot (often using a camera) and the coupled dynamics between perception and motion (when using drones). Furthermore, results are only validated in simulations.

In ref. [9], the author proposed a model inspired by wolf pack hunting to design a multi-robot application. This involves a hierarchical system containing at least one alpha and one beta agent and a finite state machine that determines the robot's motion. Nevertheless, the system relies on a fixed formation and fixed roles for the agents reducing the flexibility in a more reactive application. In ref. [10], they are again inspired by strategies observed in a wolf pack. They observed interaction between agents in a wolf pack which allowed them to encircle the prey and start to attack it. However, the algorithm's simplicity does not fully capture the complexity of a coordinated pursuit of a faster prey, which requires more sophisticated techniques to track and ambush the target.

1.5. Authors' contribution

Our approach is a cycling hierarchical agent-based methodology to implement behavior for a group of chasing robots. The bio-inspiration lies in allocating dynamically changing roles, and the pursuit itself. The pursuit behavior is given by the Deviated Pure Pursuit (DPP) technique. Furthermore, our approach introduces an alternative to conventional approaches based on attraction-repulsion forces to reduce the recurrent problem of oscillations. In addition, the technique is flexible to the number of pursuers and robust to abrupt changes in the formation. Moreover, the approach has no explicit communication between agents and is closer to individual pursuer agents

Our approach avoids two common problems in multi-robot cooperation: oscillations and the observation of neighbors. The first is produced by the spring effect caused by the repulsive/attractive forces. This problem is accentuated in real-world applications where delays, loss of data, or long sample times are presented. To account for this, the behavior of our system is not controlled by repulsive/attractive forces. Secondly, we model the observation of neighbors just using bearing-only information, which can be easily obtained from a single camera. This provides a more realistic observation model simplifying transfer to real-world applications.

Finally, we provide a numerical validation with different pursuers and experimental results for three different scenarios.

The paper's outline is presented as follows. Section 2 introduces the methodology and the dynamics of the agents. In Section 3, the structure of the pursuit control strategy is described. Numerical validations are done describing the performance of the proposed algorithms are shown in Section 4. Flight tests are carried out to validate the pursuit strategy. Section 5 illustrates the behavior when a drone fleet is pursuing an intruder drone, before concluding in Section 6.

2. Setup and Problem Statement

The objective of pursuit is that at least one pursuer intercepts the target. We adopt the terminology target to describe the target of the pursuit. This is not a fixed target, but has different behaviors described in Section 3. This problem can be stated as follows.

Our multi-agents pursuit-evasion problem involves N pursuers and a single target moving in a horizontal plane. The pursuit takes place inside a bounded work-space W in $\mathbb{R}^2$. We consider different bounded workspaces in this paper (square and a circle). The final goal for the pursuer team is $r_{iT} < R_{cap}$, i.e. that at least one of the pursuers has a relative distance (r_{iT}) towards the target less than a threshold R_{cap}. The target trajectories are described by $P_T(t) = [x_T(t), y_T(t)]$, where (x_T, y_T) is its Cartesian coordinates in W. Similarly, the pursuer paths are described as $P_i(t) = [x_i(t), y_i(t)]$, where $i \in 1, 2, ...N$. R#2Q8

We aim to validate the approach experimentally in real time, hence, for the safety of our prototypes, the capture radius in final capture of the intruder (target) drone will not be included in this work, i.e., the interception of the target will occur just in simulations, in the experimental results, the goal will be to corral the target.

2.1. Agent

The term agent, in this work, is equivalent to the self-propelled particles proposed by Vicsek, [31], in the sense that each individual is autonomous, has a limited range of actuation, and they are driven by simplistic rules. We extend this definition to include the definition of an agent by Muro in ref. [10]. Therefore, we consider that the agents are homogeneous (neither physically differentiation or a static hierarchy), do not use explicit communication and no explicit cooperation. All agents have a common goal, and they can differentiate between neighbors and target, however, not between individuals.

The aerial drones used in this work are quadcopter vehicles as shown in Figure 1-left. From the Newton-Euler approach, the mathematical equations for this vehicle can be written as:

$$\begin{aligned} m\dot{\boldsymbol{v}} &= \boldsymbol{F} + \mathbb{R}^T \boldsymbol{F_g} \\ \dot{\boldsymbol{\eta}} &= \mathbb{B}(\boldsymbol{\eta})\boldsymbol{\Omega} \\ \mathbb{J}\dot{\boldsymbol{\Omega}} &= \boldsymbol{\tau} - [\boldsymbol{\Omega}]^{\times}\mathbb{J}\boldsymbol{\Omega} \end{aligned} \tag{1}$$

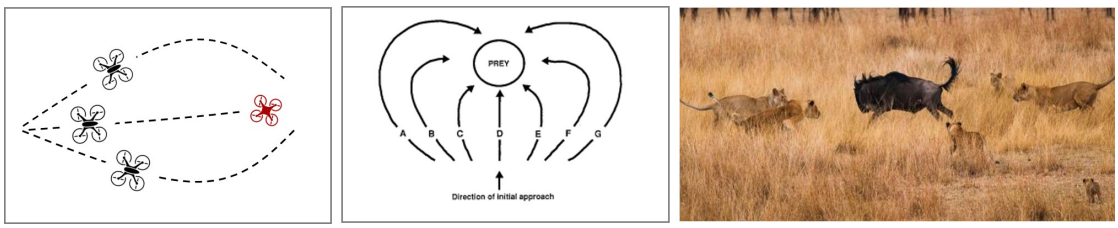

Fig. 1. The drone formation (left) mimics the behavior of lionesses hunting in group (center and right).

The bold letters represent vectors. $\boldsymbol{F}$ denotes the thrusts generated by the motors, $\boldsymbol{F_g}$ is the gravity force. $\boldsymbol{\eta}$ represents the vector of Euler angles, $\boldsymbol{\Omega}$ represents the angular velocity in the body frame. m indicates the mass of the drone, and $\boldsymbol{v} = [u\ v\ w]^T$ defines the velocity vector of the aerial robot in the body system, $\mathbb{R}$ describes the rotation matrix generated in the order yaw-pitch-roll. $\mathbb{B}(\boldsymbol{\eta})$ represents the matrix that relates the angular velocity and the derivative of the Euler angles. $\mathbb{J}$ is the inertia matrix of the drone. $[\boldsymbol{\Omega}]^\times$ represents the skew-symmetric matrix of angular velocity and $\boldsymbol{\tau}$ defines the torques applied to the vehicle.

As previously mentioned, we focus on pursuing an intruder drone, assuming it is flying at a constant altitude. We use a nonlinear controller to stabilize the orientation in the axes x, y and the altitude of the aerial vehicles. The desired altitude will be given by the z position of the intruder drone. The remaining states for the agents are ψ, x, and y as illustrated in Figure 2. The goal of this work is not to propose a control regulation for the desired trajectory; but the collective tracking of a mobile target with unknown movements in the $x - y$ plane.

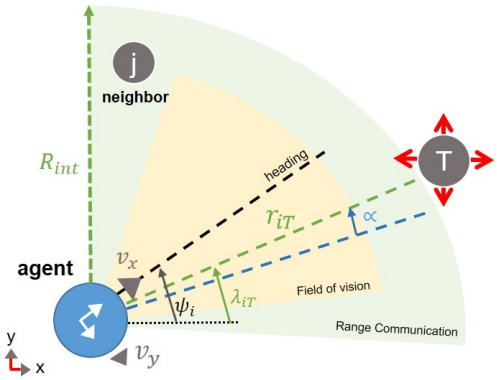

Fig. 2. Agent performance in the plane x, y. Its perception is limited for a vision field. The longitudinal velocity v_x and lateral v_y are defined in the body frame.

From Figure 2, the kinematics of the vehicle are as follows

$$\dot{x}_i = v_i \cos\psi_i, \tag{2a}$$

$$\dot{y}_i = v_i \sin\psi_i \tag{2b}$$

where x_i, y_i defines the position of the agent in the plane x, y, v_i denotes the magnitude of the vector velocity $\boldsymbol{v}_i = [v_{x_i}, v_{y_i}]^T$ in the body frame, and ψ_i represents the yaw angle of the agent i to the horizontal axis x. The linear and angular velocities define the control inputs such that

$$\boldsymbol{v}_i = \boldsymbol{f}_i, \tag{3a}$$

$$\dot{\psi}_i = f_{\psi_i} \tag{3b}$$

where $\boldsymbol{f}_i = [f_{u_i},\ f_{v_i}]$ signifies the control inputs for the longitudinal and lateral velocities, respectively and f_{ψ_i} introduces the control input for the yaw rate.

This motion constraint has an interesting property while considering the frontal camera as a navigation sensor: the drone will displace only in the direction of the field of vision, tightly coupling the perception to the motion.

For the same reason, this kind of drone motion is commonly observed in first-person-view piloting, where the only information for the pilot is the frontal camera image. In this case, the remote-pilot cancels the drone's lateral drift by combining a roll movement while turning the drones heading (yaw).

2.2. Target

The ultimate advantage of using multi-agents in pursuit is the possibility of cooperative ambushing for intercepting a more agile target. The term "agile" can mean having superior velocity or having more degrees of freedom. We assume both of these hypotheses in our work.

We consider the target will have a velocity up to twice as large as the pursuers. Furthermore, the intruder will not be constrained to non-holonomic equations (2) but by a simple particle model, which gives locomotion advantages. This configuration is a well-studied problem in differential games, commonly called "the homicidal chauffeur", see ref. [36].

We consider two strategies regarding the target behavior: (1) predefined trajectories and (2) a reactive target. First, the trajectories can be defined off-line; for example, by collecting real-flight data, or using fixed paths, e.g, circular or square trajectories. A repulsion model defines the reactive behavior of the target. Each pursuer and the arena border exert a repulsive force towards the target. Details about this repulsive force implementation is given in the subsection 3.5.

2.3. Relative Engagement

In this work, the pursuer behavior is given by a modified version of a classical guidance law, DPP. The relative kinematic engagement is a useful way to analyze convergence conditions in a pursuit-evasion problem. The kinematics equation can be deduced from the geometrical engagement of the pursuer-target, as represented in Figure 2.

From Figure 2, the Line-of-Sight (LOS) connecting the pursuer and target is denoted by r_{iT}. The bearing angle, defined between the LOS and the inertial axis-x, is denoted by λ_{iT}. The vector v_i is the velocity in the body frame, and ψ_i represents the heading (yaw) of the pursuer i.

As shown in ref. [14, 37], from the engagement a relative kinematic model can be obtained. Considering the relative velocity between, $\dot{r}_{iT} = \dot{P}_T - \dot{P}_i$. The relative velocity can be decomposed along ($v_\parallel$) and perpendicular ($v_\perp$) to the LOS :

$$\begin{aligned} v_\parallel &= \dot{r}_{iT} = v_T \cos(\alpha_T) - v_i \cos(\alpha_i), \\ v_\perp &= r_{iT}\dot{\lambda}_{iT} = v_T \sin(\alpha_T) - v_i \sin(\alpha_i). \end{aligned} \tag{4}$$

where $\alpha_T = \psi_i - \lambda_{iT}$ and $\alpha_i = \psi_T - \lambda_{iT}$. This equation represents the velocities of target seen by the pursuer along the the LOS.

3. Pursuit strategy

In the following, we introduce our group pursuit strategy. Firstly, we will describe the Deviated Pure Pursuit (DPP) guidance law, the base method for our proposition. Then, we discuss the hierarchy and the roles in our pursuit methodology, relating them with the different offsets in a DPP. Finally, our group pursuit will be explored, and details of the implementation, such as collision avoidance, will be given.

3.1. Deviated Pure Pursuit - DPP

As stated in ref. [14], the DPP can be seen as a variation of the classical Pure Pursuit (PP); it can be a result of manufacturing flaws or can be intentionally designed in order to point the agent towards a position ahead of the target, causing a "lead pursuit." However, if the offset drives the agent towards a position behind the target, it causes a "lag pursuit."

The basic principle relies in driving the heading error to a constant offset angle ($\phi_i - \lambda_{iT}) \to \alpha_0$. Considering this principle, the relative kinematics become:

R#2Q3

$$\begin{aligned} \dot{r}_{iT} &= v_T \cos(\alpha_T) - v_i \cos(\alpha_0), \\ r_{iT}\dot{\lambda}_{iT} &= v_T \sin(\alpha_T) - v_i \sin(\alpha_0). \end{aligned} \tag{5}$$

The above equations establishes one necessary condition about the offset angle: $|\alpha_0| < \pi/2$. Nevertheless, under this strategy the capture cannot be guaranteed for all $v_T > v_i$. R#2Q2

Our work explores the idea that each agent can assume different offset angles, therefore, each agent will have different degrees of "lead", and "lag" pursuit. This will cause spreading of the pursuers around the target, as shown in Figure 3. In ref. [14], an intuitive law to apply the DPP is given as R#2Q4 R#2Q3

$$f_{\psi_i} = -K_{dpp}\left(\psi_i - \lambda_{iT} - \alpha_0\right) \tag{6}$$

where K_{dpp} defines a positive constant gain.

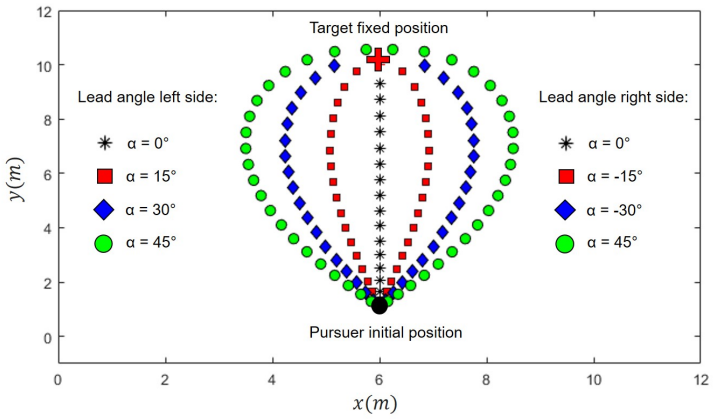

Fig. 3. Paths of deviated pursuit agents according to different angle offsets α. The agents begin the trajectory at a fixed position, given by the black dot and keep a constant velocity until reaching the target.

3.2. Hierarchy

Our system has a mixed hierarchical architecture. The *homogeneity* of the agents means that all agents are physical equals, and no identification is required. We define a hierarchy based on the advantage of information, i.e., one agent with better information will lead the group, this makes a kind of cycling hierarchy. On the other hand, they play a role in differentiation, where each agent can assume different behavior depending only on the current relative position of the pursuer to its neighbors and target.

As shown by Stander, see ref.[24], in natural systems, the central position is commonly related to dominance. In our system, the central agents have priority in moving and capturing since they have a more direct trajectory towards the target. We differentiate roles based purely on agent positioning and not due to social or biological factors (compared to natural systems). Here, each agent's trajectory is determined by computing the DPP angle assumed in the pursuit.

In Figure 4, each agent has a different angle offset, hence, each agent has a different role. The agent i (blue) and agent k (green) have consecutively positive and negative offset, therefore, they do not go in a straight line to target, rather, they ambush it from left and right, respectively. In ref. [24], the author identified this pattern in a group of hunting lionesses and called the agents at the extremities as "wings," alluding to football players' roles.

The differentiation in roles has a significant effect on the performance of the pursuit of the target since the "wings"-agents do not follow a straight line to the target but enclose it depending on the offset angle. The roles are defined such that the condition for an agent to be considered the 'center' is based on the proximity to the intruder. This allows that an agent very close to the target can execute a terminal attack independently of the position of its neighbors. R#2Q4

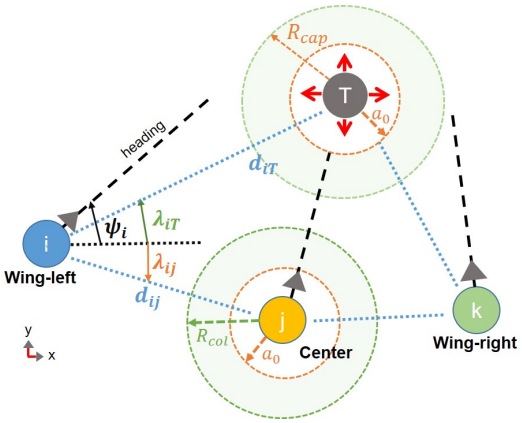

Fig. 4. Three agents pursuit an intruder drone playing different roles.

3.3. Pursuit control algorithm

The objective of the control scheme is not to control the position of each agent as it is likely to produce oscillations, due to the random and fast movements of the intruder. The control scheme's goal is to follow a deviate pursuit towards the target, with a given angle offset which will avoid collisions with the neighbors.

The heading of agent i will be controlled by its angular rate related to the pursuit conditions as shown in Figure 4. Each agent has different offset angles (α), causing different trajectories for each agent, aiming to increase the possibility of capturing the intruder using an ambush strategy. Therefore, rewriting (6) for the whole system yields R#2Q5

$$f_{\psi_i} = -K_p \left(\lambda_{iT} + \alpha \sum_{j \neq i}^{N} \delta_{i,j}(\xi_i, \xi_j) \right), \tag{7}$$

$$\delta_{i,j} = \begin{cases} -1, & \text{if } j \text{ is in the left side} \\ 1, & \text{if } j \text{ is in right the side} \end{cases}$$

$\delta_{i,j}$ can be obtained from visual information, for this work we computed it by

$$\delta_{i,j} = \frac{\boldsymbol{r}_{ij} \times \boldsymbol{r}_{iT}}{\|\boldsymbol{r}_{ij} \times \boldsymbol{r}_{iT}\|} \tag{8}$$

where $\boldsymbol{r}_{ij} = [x_j - x_i,\ y_j - y_i,\ 0]^T$ and $\boldsymbol{r}_{iT} = [x_T - x_i,\ y_T - y_i,\ 0]^T$, correspond to the planar vector of the positions for neighbor j and to target T, respectively, in the body coordinates of the pursuer i.

From Figure 4 and equation (3a), we can deduce that the longitudinal control input f_{u_i} varies with respect to the magnitude of the desired vector velocity $\boldsymbol{v}_i$ of the pursuer and also is related with the target and neighbors distances. R#2Q6

Thus, we propose

$$f_{u_i} = u_{max}\sigma_a \left(\frac{d_{iT} - R_{cap}}{R_{cap}} \right) [C_{rep}, C_{wall}]_{min} \tag{9}$$

where u_{max} is the maximal velocity imposed to the agents, d_{iT} defines the distance between agent i and the target T and R_{cap} denotes the radius to capture the intruder. $0 \leq [C_{rep}, C_{wall}]_{min} \leq 1$ is a safety coefficient that regulates the linear velocity of the agent i when approaching to agent j (collision avoidance) or when the agents are close to the limits of the workspace. Similarly, the term $\sigma_a(...)$, with $a = 1$, is set to saturate the normalized distance error ($\frac{d_{iT}-R_{cap}}{R_{cap}}$). R#2Q7

The saturation function, $\sigma_a(.)$, can be defined as follows.

$$\sigma_a(x) = \begin{cases} a, & \text{if } x > a \\ x, & \text{if } a \geq x \geq -a \\ -a, & \text{if } x < -a \end{cases} \tag{10}$$

Please note that controlling the velocity of the pursuers become relevant mainly when physical integrity of the materials are considered, such as in our experimental validations. However, in the further numerical simulation, where the collision is not a concern, the velocity of the pursuer is set constant. R#2Q9

3.4. Collision Avoidance

Most works in artificial flocking use the principle of attraction-repulsion to keep the cohesion of the group. However, those forces might cause oscillatory responses that could allow for an over-damped response of the fleet. As a consequence, this will reduce the velocity response of the agents.

We do not use attraction-repulsion principle between pursuers, but use an attraction factor towards the target, which acts as the point of convergence for all pursuers. For practical safety, a short-range repulsion was implemented, however, this is not responsible for the hunting formation.

There are two terms representing the collision avoidance (C_{rep} and ϕ_{rep_i}). The first one acts as a brake coefficient, limiting the velocity input (f_{u_i}) to zero. The second imposes a boundary in the lateral velocity, and it will be explained in Section 3.6.

For the longitudinal part this term can be denoted by

$$C_{rep} = [C_{rep_{i,j}}, C_{rep_{i,k}}, ...]_{min} \tag{11}$$

C_{rep} defines a term acting in a small rage and only aims to avoid agents collision. When an agent i detects agent j inside its repulsion area (delimited by the field of vision and a radius r_{rep}) it decreases its velocity. The coefficient of repulsion takes the minimum value of the individual repulsion from agent i to all neighbors inside the repulsion area, and it is represented as

$$C_{rep_{i,j}} = \begin{cases} 1, & \text{if } d_{ij} < R_{col} + a_0 \\ \frac{d_{ij}-a_0}{R_{col}}, & \text{otherwise} \end{cases} \tag{12}$$

where R_{col} means the radius of collision (see Figure 4) while a_0 a constant offset. When the agents are moving, the distance d_{ij} between agents i and j could be smaller than the radius of collision, then the term a_0 will impose a distance where the velocity of the agents will be completely null. This term can be defined as the shortest distance possible between two agents.

3.5. Virtual wall

Consider the scenario when the drones' fleet is approaching the workspace boundary (e.g, surveillance area). We implement a virtual repulsion force when an agent is getting closer to the border of its workspace or even a forbidden area. Along this paper, we consider two shapes for the workspace, square and circular arena. R#2Q8

For designing the virtual forces, we consider a projection of the agent in the border of the workspace, see Figure 5. Since we have one virtual agent for each agent, the break coefficient C_{wall} is computed in the same way of the agent's collision avoidance.

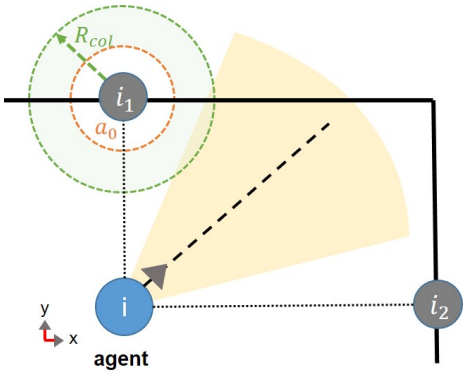

Fig. 5. Virtual wall projected when the drones approaching the border of the square workspace. Virtual agents are projected into the virtual cage

3.6. Lateral control

We design a controller to eliminate the lateral drift caused by the inertia of the drone. The drift occurs when an agent changes in its heading while moves forward. In addition, this controller also implements a lateral short-range force repulsion between two drones or between the drone and a virtual wall.

This controller is proposed as:

$$f_{v_i} = \begin{cases} 0, & \text{if } d_{ij} < R_{col} \\ \delta_{ij} \frac{R_{col} - d_{ij}}{R_{col}}, & \text{otherwise} \end{cases} \tag{13}$$

where $\phi_{rep_i} = \frac{R_{col} - d_{ij}}{R_{col}}$. The lateral velocity increases linearly when the neighbors invade the collision radius ($d_{ij} < R_{col}$). The function δ_{ij} was defined in (equation (8)), and gives the direction of the velocity in the opposite side of the collision.

4. Simulations

In order to analyze our proposition, we implemented the framework in MATLAB. The agents were modeled as a particle with constant velocity and moving in a 2D plane. We use a bounded circular workspace for these simulations. The sample period (Te) in simulations is $0.01s$. For simulation purposes, the crosses (red) represent the path of the target, while the path of the pursuers is illustrated by the diamond (blue), the asterisk (black), the circle (green), and the square (yellow). The black arrows denote the final heading (yaw) of each pursuer. The stop condition in the simulation is that the target is in a distance less than the capture radius $R_{cap} = 0.3$m of at least one agent. Three scenarios are considered:

- Fixed target and three pursuers, to illustrate the formation pattern in a simple case;
- Escape trajectories, varying number of agents, to show the flexibility of the strategy;
- Swarm of pursuers, to show our approach of these properties to deal with scalability issues, involving large amount of agents.

Finally, we analyse the effect of the angle offset in the capture rate and on the effect on collisions between pursuer.

4.1. Fixed target

The first simulation is where the target remains in a fixed position. Figure 6 shows that the patterns of the pursuers drones is similar to the trajectories executed by a group of hunter animals.

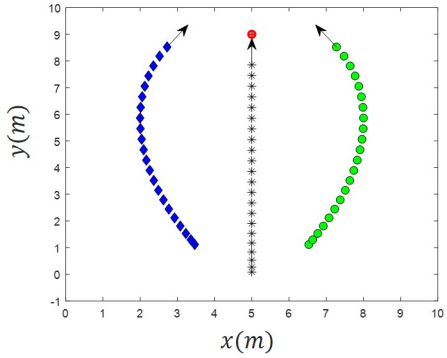

Fig. 6. Trajectories generated for three agents and a fixed target. The pursuers start in the bottom of the square and they pursuit the target with constant velocity. The deviation offset (α_0) for this pursuit is 30 degrees.

4.2. Escape trajectories

Due to this strategy's flexibility and decentralized nature, it can be easily transferred between environments with a different amount of agents. Several simulations were carried out with different numbers of pursuers ($1, 2, 3$ and 4), see Figure 7. For these simulations, the intruder velocity was set to $10m/s$ and for the pursuers $5m/s$ (half of the target speed). For visualization's sake, the maximum time for simulation was limited to $4s$. In addition, in this scenario, we propose reactive escape behavior for the target. The behavior is described in Section 2.2, all the pursuers and the borders exert a repulsive force towards the target.

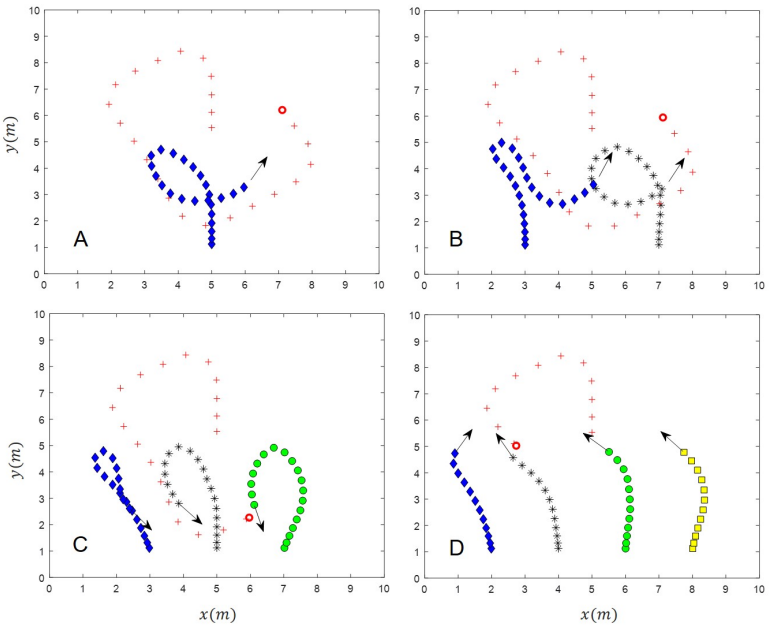

Fig. 7. Snapshots for pursuits with different numbers of agents ($1, 2, 3$ and 4). The target in all cases does an escape trajectory based in repulsion forces, with velocity of 2 times bigger than the pursuers, i.e., $\boldsymbol{v_T} = 2\boldsymbol{v_i}$. The end of the target trajectories is signalized by the red circles.

From Figure 7, in the two firsts cases (A and B), the target was not caught until the end of simulations. However, in the two lasts cases (C and D), the target was successfully intercepted before the limit of time, this shows the importance of the increasing number of pursuers in the success of the chase. In addition, in Figure 7 the velocity of the agents is implicitly indicated in the diagram. The positions of agents are shown at a constant time stamp, therefore, the sparser a trajectory of an agent, the faster it is. In Figure 7-A, for example, we can see that the target (red cross) has superior to the single pursuer (blue diamond).

4.3. Swarm

Although collective hunting is usually employed to small groups of pursuers; our algorithm can easily be applied to larger groups of agents. In the following simulation (Figure 8), we illustrate the pursuit behavior applied to a group of 15 pursuers. The agents demonstrate an emergent flocking pattern, where each agent computes independent trajectories following the target with no information of the neighbors distance.

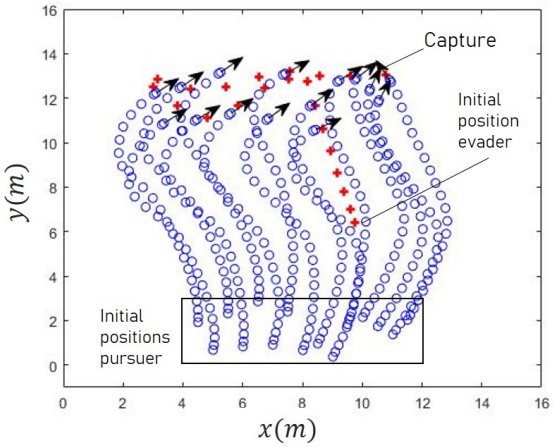

Fig. 8. Case of 15 pursuers and an intruder. The pursuers intercept the target early in their trajectory.

4.4. Effect of the offset angle on the capture

This quantitative evaluation aims to assess the effect of the angle offset (α_0) on the total number of successful captures. To this end, we subject our strategy to a group pursuit benchmark, previously proposed in ref. [34]. This benchmark consists of a circular bounded arena. We tested different angle offset values, incrementally from $\pi/32$, measuring the number of captures, as indicated in Figure 9.

For each benchmark evaluation, 700 pursuit episodes are completed, with a variable target relative velocity, $v_T/v_i = 0.8, 1.0, ..., 2$. More details about the benchmark implementation is given in ref. [34].

Figure 9-left shows the number of captures as function of the amount of pursuers. Figure 9-right, represents the average time taken to capture.

The first value ($\alpha_0 = 0$) is equivalent to the classic Pure Pursuit, where all agents point to the target's current position. However, applying this configuration, the cooperative behavior does not emerge, therefore, the total catches remain practically the same during all the numbers of agents.

Therefore, with the smallest increment of offset ($\alpha_0 = \pi/32$), the size of the group has an affect on the total amount of captures. The number of captures tends to increase, culminating in a cumulative total of 4233 (76%) catches for an offset of $\pi/8$. For an angles offset of $\alpha_0 = 3\pi/32$, our approached completed all captures during the benchmark for 5 pursuers onwards, therefore, this is as an effective strategy for chasing a faster target.

For angle offsets larger than $\pi/8$ the capture's performance tends to decay. This is due to the fact that large offset values result in very open trajectories towards the target, allowing the target to escape between two pursues.

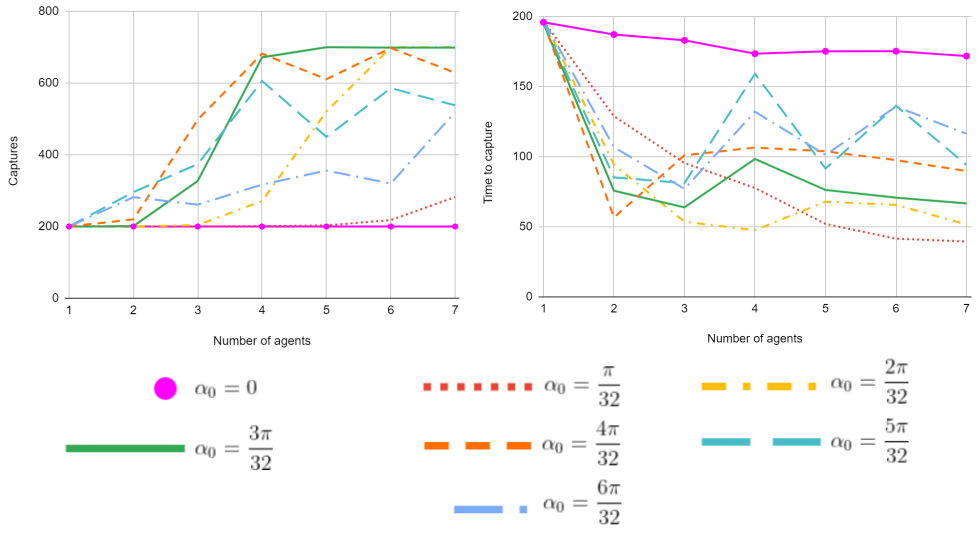

Fig. 9. Effects of the offset value in the pursuit.

4.5. Effect of the offset angle on the collision

We use the same benchmark, to analyse the amount of collisions between two pursuers, i.e., $r_{ij} < R_{col}$, where the collision radius R_{col} is equal to $0.15m$. The graph on Figure 10 clearly illustrates the effect of the offset on collision avoidance. For $\alpha_0 = 0$, collisions tend to grow exponentially, reaching values up to 395 times higher than using $\alpha_0 = 3\pi/32$.

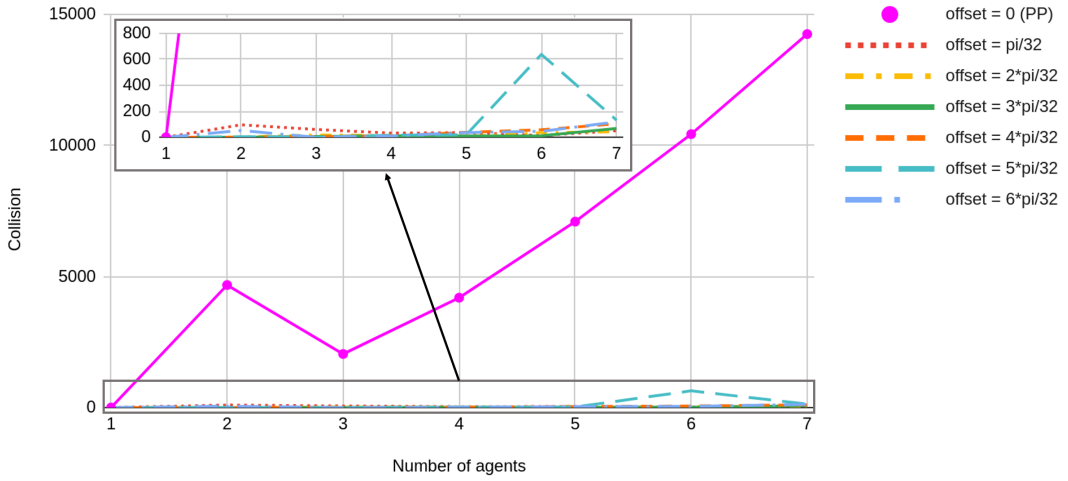

Fig. 10. Effect of the offset value (α_0) on the collision between pursuers.

5. Experimental Validation

The proposed pursuit algorithm was also validated in a real-world experiments. We use quadcopter Parrots - AR Drone 2 for aerial robots. We used Framework libre Air[1] an open source framework which provides a low-level connection to the drones. Opti-Track was used for motion capture to locate each agent. For experimental validation and security reasons, the final capture does not occur here; instead, the pursuers must stop at a security distance of $d = 1$m. The drones operate in a $6m \times 6m$ square workspace.

5.1. Drone control

In our application, the control of the quadcopters is divided into two levels, as illustrated in Figure 11. The lowest level is the control for stabilizing the drone in orientation and altitude. The higher level is responsible for controlling the pursuers ($f_{\psi i}$, f_{ui}, f_{vi}) in desired angles (θ_{i_d}, ϕ_{i_d} and ψ_{i_d}). The pursuit algorithm (the green box) is responsible for the "flocking intelligence" for each individual. Each agent can perceive the environment (neighbors, obstacles and target) and gives to the the lower levels the velocities references ($f_{\psi i}$, f_{ui}, f_{vi}).

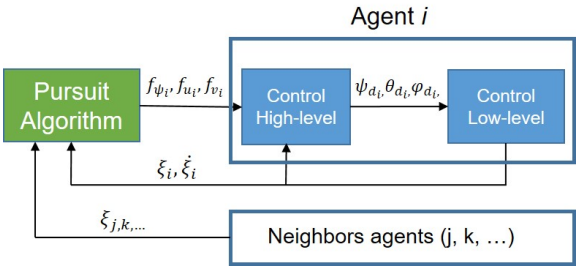

Fig. 11. Schema for the control levels for a given agent i.

Therefore, from (1) and using the PD controller in the altitude and the orientation[2], it follows that, $w_i \to 0$, $z_{B_i} \to z_{B_{i_d}}$, $\dot{\eta}_i \to 0$ and $\eta_i \to \eta_{i_d}$, where $z_{B_{i_d}}$ defines the desired altitude of the aerial vehicle i and η_{i_d} represents its desired orientation. Choosing $\eta_{i_d} = 0$ the aerial vehicle is stabilized at hover. The goal is the capture of the target drone, thus from (1) and without loss of generality for x and y (in the body frame) it follows that

$$\dot{u}_i \approx -g \sin\theta_i$$
$$\dot{v}_i \approx g \cos\theta_i \sin\phi_i$$

The prototype has an embedded controller for the attitude, thus η is small enough such that $\cos\Xi_i \approx 1$ and $\sin\Xi_i \approx \Xi_i$, being Ξ_i a generic angle representing θ_i and ϕ_i.

$$u_i = \dot{x}_{i_B} \qquad v_i = \dot{y}_{i_B}$$
$$\dot{u}_i \approx -g\theta_i \qquad \dot{v}_i \approx g\phi_i$$

where x_{i_B} and y_{i_B} are the position coordinates in the body frame. From (3a) and (13), it follows that the following stabilize the lateral dynamics

$$\dot{v}_i = -k_{v_i} v_i - f_{v_i} \tag{14}$$

where k_{v_i} denotes a positive gain. Notice from (13) that f_{v_i} defines a position lateral feedback for avoiding the obstacles. Thus, it is easy to deduce the desired roll angle as

$$\phi_{i_d} = -k_{\dot{y}_i} v - k_{y_i} f_{v_i} \tag{15}$$

[1] https://devel.hds.utc.fr/software/flair
[2] Others controllers can be also used for this purpose.

where $k_{\dot{y}_i} = \frac{k_{v_i}}{g}$ and $k_{y_i} = \frac{1}{g}$. Similarly, from (3a) and (9), it follows that

$$\dot{u}_i = -k_{u_i}(u_i - f_{u_i}) \tag{16}$$

with k_{u_i} denotes a positive gain. Notice here that f_{u_i} defines the desired velocity of the agent i that will vary with respect the position of the target T. Therefore,

$$\theta_{d_i} = k_{\dot{x}_i} e_{u_i} \tag{17}$$

where $k_{\dot{x}_i} = \frac{k_{u_i}}{g}$ and $e_{u_i} = u_i - f_{u_i}$. From (3b) and (7), it follows that

$$\dot{\psi}_i = -K_p e_{\psi_i} \tag{18}$$

where $e_{\psi_i} = \psi_i - \psi_{i_d}$.

All control laws and pursuit algorithms are computed on-board each agent, the ground station only analyzes the states and re-takes the control in manual mode in emergency cases. For improving experimental validation, a Kalman Filter was implemented to estimate states in cases where the agent does not receive the data, see ref. [33].

Lateral control takes great importance in real-world experiments, as lateral drifts appear when the agents avoid obstacles which does not happen in simulation.

5.2. Experimental Evaluation

Two scenarios are proposed to validate the pursuit algorithms.

1. The intruder drone remains in a fixed position, and three pursuers drones track it.
2. The target, moving with random movements given by a user, is tracked for three pursuers aerial drones.

5.3. Three pursuers and one static target

The target is placed in the ground at a distance of $6m$ from the pursuers. The pursuers are placed in an initial triangular formation with a distance of approximately $2m$ from their closest neighbors. This experiment aims to reproduce the pattern presented in simulation (see Figure 6). The parameters used in this experiment are presented in Table I.

Parameters	Scenarios 1 and 3	Scenario 2
α	20^o	10^o
u_{max}	1.3	1.3
R_{cap}	1.6	1.6
$Field - vision$	360^o	120^o
R_{int}	5	2
R_{col}	1.5	0
a_0	1	0

Table I . Control parameters used in the real-time scenarios..

Figure 12 shows the real-world result for this scenario. The agents take on a similar strategy compared to the hunting pattern described in Figures 1 and 6. However, the irregularities observed in the pursuer's trajectories are caused by the airflow produced by the proximity of the drones and the absence of position control.

5.4. Three pursuers and one faster intruder

An operator manually controls the target, trying to get away from the three pursuers' drones. The goal for the pursuers is to track the target, limiting their distances to the target to ensure no collisions. In addition, to further ensure the safety of our drones, the target flies at a different altitude. We choose $z_d = 0.8$m for the intruder and $z_{d_i} = 1.7$m to the pursuers. The parameters for this flight are also shown in Table I.

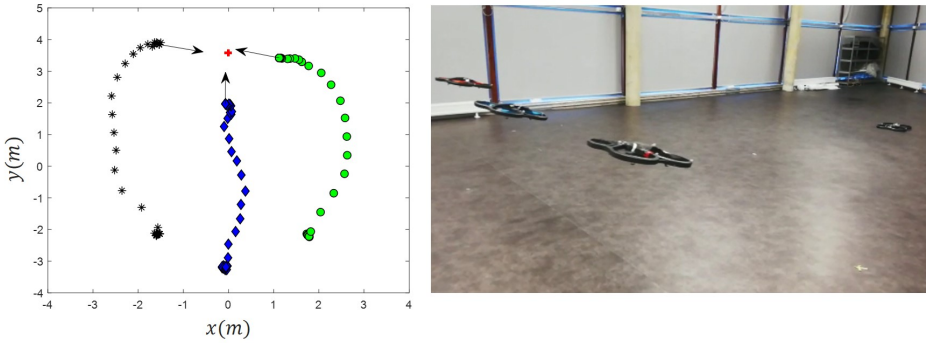

Fig. 12. Real-time performance when three aerial drones pursuit a fixed target. The first image (left) shows the displacement of the drones in the flight arena. The second one is a picture of the experiment.

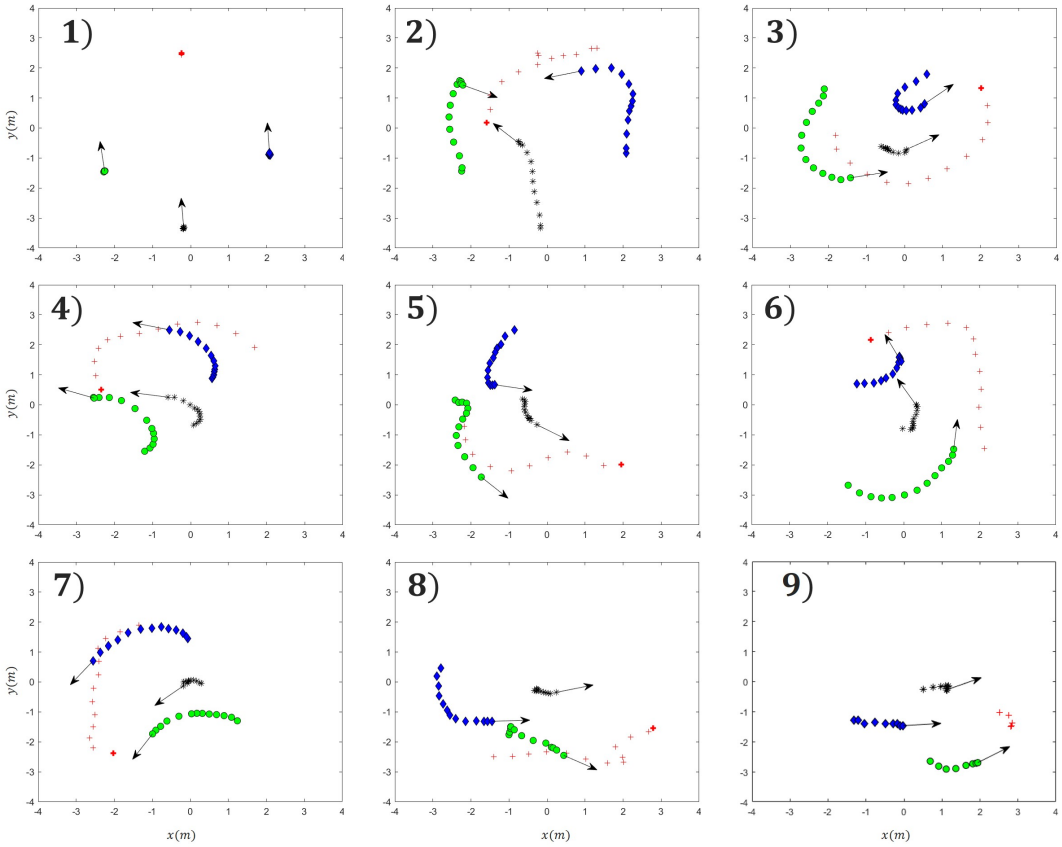

Fig. 13. Real-time performance when three aerial drones pursuit an aerial intruder drone with random movements.

In Figure 13 some snapshots are shown which illustrate the real-world performance of the drones pursuing the faster target. The figure is composed of 9 snapshots taken at an interval of $2.75s$. For each snapshot, the position of each agent was plotted ten times, with a sample time of $0.275s$. The larger velocity of the target ($2m/s$) is highlighted by the distance traveled by it in each snapshot.

The collision avoidance property was implemented in the pursuers and can be seen in Figure 14, where the inter-distances are plotted as a function of time. The minimum distance between two pursuers throughout the episode was approximately $1m$. However, since the target was piloted manually, there were no guarantees of a collision-free trajectory

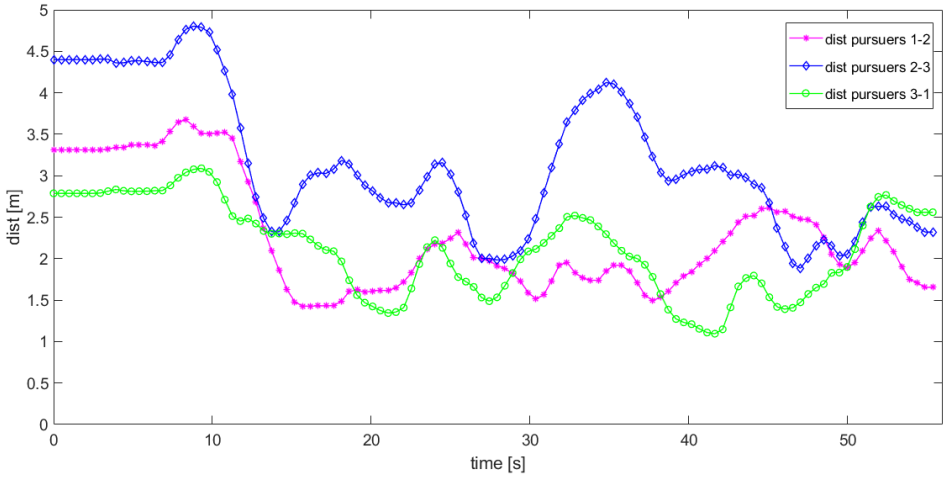

Fig. 14. Inter-distance between pursuers during the experimental test.

Observe from these figures that the cooperative hunting pattern emerged during these real-world experiments. Even facing a complex pursuit with a faster reactive target, the pursuer's drones could track and intercept the futures movements of the target. No formation is intended to be kept, instead, the drones re-arrange between themselves in other to corral the intruder. The roles of positioning (center, left-, and right-wings) are constantly changing as a function of the relative position of the fleet to the target.

Moreover, taking into account the size of the virtual arena (6×6m) and the presence of 4 drones flying inside, this approach has demonstrated a robust, practical performance with respects to the disturbances produced by the airflow of the aerial vehicles.

The flight tests can be seen on: https://youtu.be/g2dODbd6ZLA.

6. Conclusion

A decentralized strategy for cooperative group pursuit against a single intruder has been proposed and validated in this work. Our approach, inspired by lionesses' hunting behavior, applies the geometrical rules of Deviated Pure Pursuit to generate different trajectories for each agent. This strategy avoids using attractive and repulsive forces for pattern formation, which occasionally results in the system's oscillations.

Numerical simulations were carried out to demonstrate the efficiency of the proposed chase strategy. The effect of the angle offset was analyzed quantitatively through a pursuit benchmark. We observed the emergence of the behavior of collective pursuit through two main metrics, the increase in the number of target's capture and the decrease in the amount of collision between pursuers. Furthermore, a proof-of-concept with real drones is shown, were we could verify the pursuit behavior in a real-world experiment with three pursuers and one target.

This preliminary work focused mainly on the formalization of our algorithm and the qualitative description of the emerging behavior. However, our proposal also proved to be effective in capturing a faster target, as shown by the group pursuit benchmark described in section 4. Furthermore, our proposal requires very minimal assumptions; the pursuit algorithm, disregarding the safety layer of collision avoidance, requires only relative bearing angle information in relation to the target and neighbors. This minimalism of implementation puts our algorithm in a very advantageous position compared to the previous ones, such as in ref [10,12,13], which requires more information from the target and neighbors, such as position or speed in global coordinates.

We believe that a more in-depth theoretical analysis of the collision-free phenomenon observed in this work is also interesting and reaffirmed from simulations in the pursuit benchmark.

Although these first experimental results indicate the viability of this algorithm, several areas must be improved to obtain a fleet of autonomous chasing drones. The first one is perception. Pursuers must be able to extract the bearing angle information using only the embedded sensors, which are commonly the front camera and the Lidar. However, using a camera-type sensor with limited FOV arouses the interest of another field of investigation, the effect of limited visual sense on collective behavior. Recently, works, as in ref. [35], have been dedicated to evaluating the impact of this non-linearity in the observation, which could improve our work results.

Acknowledgement
We thank Rhys Newbury (Monash University - Melbourne, Australia) for the valuable effort on the full text revision. R#2Q1

References

1. T. Lefebvre and T. Dubot, "Conceptual design study of an anti-drone drone," **In**: *16th AIAA Aviation Technology, Integration, and Operations Conference*, Washinton, DC, USA, pp. 1–14 (2016).
2. R. L. Allen, *Quadrotor intercept trajectory planning and simulation.* Thesis, Naval Postgraduate School, Monterey, CA, USA (2017).
3. D. Solinger, P. Ehlert, and L. Rothkrantz, "Creating a dogfight agent," *Technical Report DKS05–01/ICE 10*, Delft University of Technology, Netherlands (2005).
4. D. Hert, *Autonomous Predictive Interception of a Flying Target by an Unmanned Aerial Vehicle.* Thesis, Czech Technical University in Prague, Prague, Czech Republic (2018).
5. J. T. Arneberg, *Guidance Laws for Partially-Observable UAV Interception Based on Linear Covariance Analysis.* Thesis, Massachusetts Institute of Technology, Massachusetts, USA (2018).
6. H. Yulan, Z. Qisong, and X. Pengfei, "Study on multi-robot cooperation stalking using finite state machine," *Procedia Engineering*, **29**, pp. 3502–3506 (2012).
7. H. Yamaguchi, "A cooperative hunting behavior by multiple nonholonomic mobile robots," **In**: *SMC'98 IEEE International Conference on Systems, Man, and Cybernetics*, **4**, pp. 3347–3352 (1998).
8. Z. Wu, Z. Cao, Y. Yu, L. Pang, C. Zhou, and E. Chen, "A multi-robot cooperative hunting approach based on dynamic prediction of target motion," **In**: *2017 IEEE International Conference on Robotics and Biomimetics* , Macau, China, pp. 587–592 (2017).
9. A. Weitzenfeld, A. Vallesa, and H. Flores, "A biologically-inspired wolf pack multiple robot hunting model," **In**: *2006 IEEE 3rd Latin American Robotics Symposium*, Santiago, Chile, pp. 120–127 (2006).
10. C. Muro, R. Escobedo, L. Spector, and R. Coppinger, "Wolf-pack (canis lupus) hunting strategies emerge from simple rules in computational simulations," *Behavioural processes*, **88**, no. 3, pp. 192–197 (2011).
11. D. Shishika, J. K. Yim, and D. A. Paley, "Robust lyapunov control design for bioinspired pursuit with autonomous hovercraft," *IEEE Transactions on Control Systems Technology*, **25**, no. 2, pp. 509–520 (2016).
12. L. Angelani, "Collective predation and escape strategies," *Physical review letters*, **109**, no. 11, pp. 1–5 (2012).
13. Y. Lin and N. Abaid *Collective behavior and predation success in a predator-prey model inspired by hunting bats.. Robotica*, **88**, no. 6, pp. 062724 (2013).
14. N. A. Shneydor, *Missile guidance and pursuit: kinematics, dynamics and control.* Elsevier (1998).
15. P. Nahin, *Chases and Escapes The Mathematic.* Princeton, New Jersey: Princeton University Press 2007.
16. F. Belkhouche and B. Belkhouche, "A method for robot navigation toward a moving goal with unknown maneuvers," *Robotica*, **23**, no. 6, pp. 709–720 (2005).
17. F. Belkhouche, B. Belkhouche, and P. Rastgoufard, "Parallel navigation for reaching a moving goal by a mobile robot," *Robotica*, **25**, no. 1, pp. 63–74 (2007).
18. H. Teimoori and A. V. Savkin, "A biologically inspired method for robot navigation in a cluttered environment," *Robotica*, **28**, no. 5, pp. 637–648 (2010).
19. R. Tan and M. Kumar, "Proportional navigation (pn) based tracking of ground targets by quadrotor uavs," **In**: *ASME 2013 Dynamic Systems and Control Conference*, Palo Alto, CA, USA, pp. 157–173 (2013).
20. S. Jung, S. Hwang, H. Shin, and D. H. Shim, "Perception, guidance, and navigation for indoor autonomous drone racing using deep learning," *IEEE Robotics and Automation Letters*, **3**, no. 3,

pp. 2539–2544 (2018).
21. H. Huang, W. Zhang, J. Ding, D. M. Stipanović, and C. J. Tomlin, "Guaranteed decentralized pursuit-evasion in the plane with multiple pursuers," **In**: *2011 50th IEEE Conference on Decision and Control and European Control Conference*, Orlando, FL, USA, pp. 4835–4840 (2011).
22. J. Li, M. Li, Y. Li, L. Dou, and Z. Wang, "Coordinated multi-robot target hunting based on extended cooperative game," **In**: *2015 IEEE International Conference on Information and Automation*, Lijiang, Yunnan, China, pp. 216–221 (2015).
23. A. Zafeiris and T. Vicsek, *Why We Live in Hierarchies?: A Quantitative Treatise*. Springer (2017).
24. P. E. Stander, "Cooperative hunting in lions: the role of the individual," *Behavioral ecology and sociobiology*, **29**, no. 6, pp. 445–454 (1992).
25. J. Gong, J. Qi, G. Xiong, H. Chen, and W. Huang, "A ga based combinatorial auction algorithm for multi-robot cooperative hunting," **In**: *2007 International Conference on Computational Intelligence and Security* , Heilongjiang, China, pp. 137–141 (2007).
26. W. Wang, J. Qi, H. Zhang, and G. Zong, "A rapid hunting algorithm for multi mobile robots system," **In**: *2007 2nd IEEE Conference on Industrial Electronics and Applications*, Harbin, China, pp. 1203–1207, IEEE (2007).
27. Y. Duan, X. Huang, and X. Yu, "Multi-robot dynamic virtual potential point hunting strategy based on fis," **In**: *2016 IEEE Chinese Guidance, Navigation and Control Conference*, Nanjing, China, pp. 332–335 (2016).
28. C. W. Reynolds, *Flocks, herds and schools: A distributed behavioral model*, **21**. ACM (1987).
29. T. Vicsek and A. Zafeiris, "Collective motion," *Physics reports*, **517**, no. 3-4, pp. 71–140 (2012).
30. T. Saito, T. Nakamura, and T. Ohira, "Group chase and escape model with chasers' interaction," *Physica A: Statistical Mechanics and its Applications*, **447**, pp. 172–179 (2016).
31. M. Janosov, C. Virágh, G. Vásárhelyi, and T. Vicsek, "Group chasing tactics: how to catch a faster prey," *New Journal of Physics*, **19**, no. 5, pp. 1–16 (2017).
32. G. Vásárhelyi, C. Virágh, G. Somorjai, N. Tarcai, T. Szörényi, T. Nepusz, and T. Vicsek, "Outdoor flocking and formation flight with autonomous aerial robots," **In**: *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Chicago, Illinois, pp. 3866–3873 (2014).
33. C. De Souza, P. Castillo, R. Lozano, and B. Vidolov, "Enhanced uav pose estimation using a kf: experimental validation," **In**: *2018 International Conference on Unmanned Aircraft Systems (ICUAS)*, Dallas, Texas, pp. 1255–1261 (2018).
34. C. De Souza, R. Newbury, A. Cosgun, P. Castillo, B. Vidolov, and D. Kulić, "Decentralized Multi-Agent Pursuit Using Deep Reinforcement Learning," **In**: *IEEE Robotics and Automation Letters (RAL)*, **6**, no. 3, pp. 4552–4559 (2021).
35. E. Soria, F. Schiano, D. Floreano, "The influence of limited visual sensing on the reynolds flocking algorithm," **In**: *Third IEEE International Conference on Robotic Computing (IRC)*, pp. 138–145 (2019).
36. I. Rufus, "Differential Games, SIAM Series in Applied Mathematics." Wiley, New York (1965).
37. F. Belkhouche, B. Belkhouche, P. Rastgoufard, "Parallel navigation for reaching a moving goal by a mobile robot," *Robotica*, **25**, no. 1, pp. 63–74 (2007).