

Bio-inspired control for collective motion in swarms of drones

Matthieu Verdoucq, Guy Theraulaz, Ramón Escobedo, Clément Sire, Gautier Hattenberger

▶ **To cite this version:**

Matthieu Verdoucq, Guy Theraulaz, Ramón Escobedo, Clément Sire, Gautier Hattenberger. Bio-inspired control for collective motion in swarms of drones. International Conference on Unmanned Aircraft Systems (ICUAS 2022), Jun 2022, Dubrovnik, Croatia. pp.1626-1631, 10.1109/ICUAS54217.2022.9836112 . hal-04235527

HAL Id: hal-04235527
https://hal.science/hal-04235527

Submitted on 10 Oct 2023

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Bio-inspired control for collective motion in swarms of drones

Matthieu Verdoucq[1,2,*], Guy Theraulaz[2], Ramón Escobedo[2], Clément Sire[3], and Gautier Hattenberger[1]

[1]École Nationale de l'Aviation Civile, Université de Toulouse, 31400 Toulouse, France
[2]Centre de Recherches sur la Cognition Animale, Centre de Biologie Intégrative (CBI),
Centre National de la Recherche Scientifique (CNRS) & Université Paul Sabatier, 31062 Toulouse, France
[3]Laboratoire de Physique Théorique, CNRS & Université de Toulouse Paul Sabatier, 31062 Toulouse, France

***Abstract*— To control a distributed swarm of UAVs, many solutions aim at reaching global coordination through local interactions at the agent scale. These interactions are based on the use of reactive algorithms that allow a fleet to adapt its collective behavior to specific operating contexts, but not to all. While providing some flexibility, reactive algorithms can lack speed of completion, safety, and robustness in clustered environments. Generally, the best known reactive algorithms are borrowed from behavioral patterns observed in nature, but tend to digress from the adaptation to a rigorous geometrical approach. Our approach aims at implementing a coordination model directly inspired from the interactions and behaviors that a small fish, the rummy-nose tetra (*Hemigrammus rhodostomus*), adopts to display collective movements. This article focuses on such an implementation in the deployment of a group of UAVs inside a circular arena. After carrying out experiments with a swarm of 5 drones, we develop a simulator to test the implementation in a swarm of drones of an extended version of the fish school model and evaluate its capacity to reproduce the experiments. Finally, after validating the UAV model by comparison with the experiments, the use of such collective motion algorithm in several contexts of operation is discussed.**

***Index Terms*— Collective motion, Distributed Control, Flocking algorithms, Swarm of drones, Unmanned Aerial Vehicle (UAV)**

I. INTRODUCTION

The UAS (Unmanned Aerial System) research domain has been growing in the last decade, in every of its aspects, but the predicted evolution of technologies should not allow soon an Unmanned Aerial Vehicle (UAV) to both fly for a long period of time, and give access to unlimited on-board sensing, decision-making, and mission-planning [1]. Known ways to control a swarm of UAVs can be divided considering their architecture type, reaching from fully centralized (one central node giving coordinated instructions to a swarm, knowing the environment, and making every decision as in [2]), to distributed systems, where each agent makes decisions based on limited information perceived from its environment [3].

The interest towards distributed system has been raising, as it is thought to be a solution for deploying multi-sensor, adaptive, and flexible systems in real-world scenarios [4]. The distributed approach is more recent in Multi Agent Systems (MAS), compared to centralized methods, relying on powerful computation force, but high dependence and weakness in the communication architecture. Since this approach has been showing acceptable results, and because it allows many possibilities that can enhance MAS capacities, a good deal of research have been conducted on distributed control of UAV swarms.

Here, we focus on bio-inspired approaches, as they show immediate proof-of-concept, are optimized within their context of operation, and often show optimization at many levels, even the computational one. Swarm robotics, that has emerged from the falling of technology price and the improving performances of communication, sensing and processing hardware, has been showing real potential to real-world applications: tracking, inspection, transport, surveillance, exploration,... [5], [6]. Collective motion in animal groups has been observed since the dawn of time [7], and methods to reproduce its properties in robotics systems are plenty, would it be in design [8], sensing, control, task allocation, and optimization [9]. The main advantage of bio-inspired algorithms is that, if reproducible, they can offer an appropriate and efficient solution to well-stated problems, at low computational cost. Collective behavior in biological systems results in self-organized processes based on local interactions [10]. Schools of fish can navigate along thousands of kilometers, find food sources, escape from predators, and more, without any external force guiding them, and without any apparent leader.

This type of model can be used as a source of inspiration to develop decentralized algorithms to control MAS. Such bio-inspired algorithms can offer parallel computation (as every agent is computing its own motion response according to its surroundings), robustness (as affecting one individual of the group does not put in danger the rest of the fleet), flexibility (such an algorithm can adapt to many situations despite its relative simplicity), but also resilience to communication loss and hardware failures. Compared to centralized of decentralized methods, there is no leader or pseudo-leader in such systems, ensuring the survivability of a swarm with the loss of several random members.

The paper is organized as follows. Section 2 is dedicated to the presentation of the model and its adaptation to UAVs. Then, Section 3 is devoted to the controller design. Section 4 illustrates the performance of the method based on various simulations compared to real flights experiments. Finally, a conclusion and perspectives are presented in the last section.

* Corresponding author: matthieu.verdoucq@enac.fr

II. RELATED WORK AND MOTIVATION

A. Related work

Flocking is defined as collective movement of a large number of individuals, towards a common goal, and with observable stability and order [11]. It is observed in many biological systems ((bacteria colonies, swarms of insects, flocks of birds, herds of mammals and human crowds), and shows great adaptability to environmental changes with a large variety of spatial patterns. One of the first works that investigates 3D flocking with a computer model was done by Reynolds [12], that defined a set of local interactions rules between agents resulting in a global collective motion. Since then, a lot of works, that can be found in surveys [1], [13] have been done by defining variant ways to control their formation in a reactive and distributed structures [14].

B. The biological model

Recently, a computational model of coordinated swimming in schools of Rummy-nose tetra (*H. rhodostomus*) has been developed from the analysis of more than ten hours of trajectories of single fish and pairs of fish swimming in a circular tank [15]. In this species, fish perform a burst-and-coast motion which results from sudden heading changes and straight-forward accelerations (kicks) followed by a gliding period until the next kick. The model shows that the heading change between two consecutive movements called 'kick' results from:

1) Fish display spontaneous fluctuations in their movement and react to the wall when they are close to it;
2) The heading change between two consecutive kicks depends on social pairwise interactions with close neighbors;
3) These social interactions consist of two components of attraction and alignment with some neighboring fish.

Assuming a discrete fish motion, trajectories are divided in kicking instants at which position is updated according to

$$\vec{u}_{n+1} = \vec{u}_n + l_n \vec{e}(\phi_{n+1}) \tag{1}$$

where $\vec{u}$ denotes fish position in the plane, $\vec{e}(\phi_{n+1})$ is the unit vector along the new angular direction $\phi_{n+1} = \phi_n + \delta\phi_n$ of the fish, and l_n is the length of the kick. When swimming in pairs in a circular tank, the heading angle change $\delta\phi_n$ results from the additive combination of the random fluctuations $\delta\phi_{\mathrm{R}}$, which depend on the distance to the wall r_{w} (Fig. 1), the effect of the wall ($\delta\phi_{\mathrm{w}}$, which depends on r_{w} and on the angle of incidence to the wall θ_{w} (Fig. 1), and finally, the social interaction of attraction and alignment with the other fish, $\delta\phi_{\mathrm{Att}}$ and $\delta\phi_{\mathrm{Ali}}$ respectively ([15]):

$$\begin{aligned} \delta\phi_i = {} & \delta\phi_{\mathrm{R}}(r_{\mathrm{w}}^i) + \delta\phi_{\mathrm{w}}(r_{\mathrm{w}}^i, \theta_{\mathrm{w}}^i) + \delta\phi_{\mathrm{S}}^i(d_{ij}, \psi_{ij}, \Delta\phi_{ij}), \\ \delta\phi_{\mathrm{S}}^i = {} & \delta\phi_{\mathrm{Att}}(d_{ij}, \psi_{ij}, \Delta\phi_{ij}) + \delta\phi_{\mathrm{Ali}}(d_{ij}, \psi_{ij}, \Delta\phi_{ij}), \end{aligned} \tag{2}$$

where $\delta\phi_{\mathrm{S}}^i$ is the social interaction term, d_{ij} is the distance between fish i and j, ψ_{ij} is the angle with which fish i perceives fish j, and $\phi_{ij} = \phi_j - \phi_i$ is the heading difference between both fish (Fig. 2).

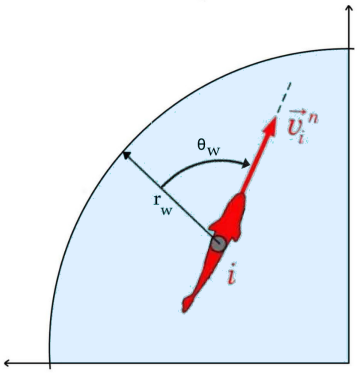


Fig. 1. Individual state variables (black) of fish i at the instant t_i^n of its n^{th} kick: r_{w} is the distance to the wall, $\vec{v}_i^n$ is the velocity vector, and θ_{w} is the angle that this vector forms with the normal to the wall.

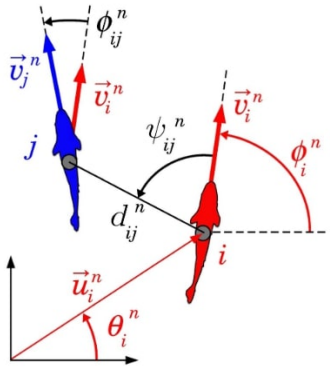


Fig. 2. Individual state variables of fish i (red) and social state variables of fish i with respect to fish j (black) at the instant t_i^n of the n^{th} kick of fish i: $\vec{u}_i^n$ and $\vec{v}_i^n$ are the position and velocity vectors of fish i, θ_i^n and ϕ_i^n are the angles that these respective vectors form with the horizontal line, $\vec{v}_j^n$ is the velocity vector of fish j at the instant of time of the n^{th} kick of fish i, d_{ij}^n is the distance between fish i and j, ψ_{ij}^n is the angle with which fish j is perceived by fish i, and $\phi_{ij}^n = \phi_j^n - \phi_i^n$ is the heading difference between them.

Interaction functions are decoupled into the product of single-variable functions, where angular functions are odd or even according to the left-right symmetry of the model. In larger groups, the social interaction is extended to other fish in the group, so that

$$\delta\phi_{\mathrm{S}}^i = \sum_{j \in I} (\delta\phi_{\mathrm{Att_{ij}}} + \delta\phi_{\mathrm{Ali_{ij}}}), \tag{3}$$

where I denotes the set of neighbors with which the focal fish interacts. As shown in [16], fish typically interact with their two most influential neighbors (*i.e.*, those which exert the largest influence on their heading angle change). This implies that each fish only processes a small quantity of information about its environment, which is sufficient for ensuring the coordinated motion of the entire group. This makes the model especially suitable for being implemented in a UAS and reproduce the observed behavior both at the individual and collective scale, thanks to the small amount of information required regarding both the number of variables determining agent's individual state and the number

of influential neighbors that must be taken into account.

C. Implementation in UAS

To implement the fish school model in a swarm of UAVs, several adaptations must be made.

The most important change brought to the original model concerns its discrete nature (considering that in the burst-and-coast type of motion a new heading decision is taken by the fish only at the end of the coast phase). From the viewpoint of UAV control, a continuous model is more adapted for real-world applications and with regard to planning constraints. Note that some realistic fish model, as the one for the barred flagtail (*Kuhlia mugil*) [17], are actually continuous in time. We consider a swarm of N UAVs flying in a planar space, and denote the state $\vec{z_i}$ and command $\vec{U_i}$ of the UAV in a 2D-Cartesian frame by

$$\vec{z_i}(t) = \big(x_i(t), y_i(t), \phi_i(t)\big), \tag{4}$$

$$\vec{U_i}(t) = \big(V_i(t), \omega_i(t)\big), \tag{5}$$

$$\dot{z_i}(t) = (V_i(t)\cos\phi_i(t), V_i(t)\sin\phi_i(t), \omega_i(t)), \tag{6}$$

where dot denotes time derivative, $V_i = |\vec{v_i}|$ is the linear velocity (speed), and ω_i is the angular turning rate.

In this system, the turning rate and speed is calculated with a PID controller, with regard to a target heading and speed. The target speed is the result of a speed cohesion interaction, often used in flocking algorithms for adding longitudinal aggregation and dispersion. The target heading $\phi_c(t)$ is the result of the adapted model, plus a migration term regarding the mission planing of a swarm, for trajectory following, aggregation around a point of interest, or target tracking. Compared to the original model Eq. (2), it has been decided not to use the fluctuation term ("cognitive noise" in the biological context), since it does not serve any purpose on a UAS, especially because of its continuous control, but also for safety and energy consumption reasons. Secondly, inherent noise is already present in the system, through air perturbations (more present with several individuals, and in outdoors experiments), communication delays, and errors in measurements. These observations will have to be kept in mind in the process of validation of the simulator, and the presence of noise in the experiments taken into account. The noise gap can be in the future estimated and simulated.

The target heading and speed are defined as follows:

$$\delta\phi_c^i = \delta\phi_{\mathrm{w}}^i + \sum_{j\in I}(\delta\phi_{\mathrm{Att}}^{ij} + \delta\phi_{\mathrm{Ali}}^{ij}) + \delta\phi_{\mathrm{Nav}}^i, \tag{7}$$

$$\delta V_i = \sum_{j\in J}\delta V_{ij}, \tag{8}$$

where I and J denote the set of neighbors considered for the heading interaction and the speed cohesion respectively, and $\delta\phi_{\mathrm{Nav}}^i$ represents the navigational term, set to null for the simulator validation. As described in the previous subsection, the social interaction strategy consists in selecting the neighbors according to their higher influence, where the influence $I_{ij}(t)$ that fish j exerts on fish i at time t is defined as the absolute value of the contribution of fish j to the instantaneous heading change of fish i [16]. In groups with more than 2 fish, the most influential neighbor is the fish having the highest influence on the focal fish heading variation [16].

In the developed system (detailed in Section III), agents calculate their new heading with a given frequency, and according to the following interaction functions [16]:

- Effect of the wall:

$$\delta\phi_{\mathrm{w}}(r_{\mathrm{w}}, \theta_{\mathrm{w}}) = \gamma_{\mathrm{w}}\sin\theta_{\mathrm{w}}\big[1 + O_{\mathrm{w}}(\theta_{\mathrm{w}})\big]f_{\mathrm{w}}(r_{\mathrm{w}}), \tag{9}$$

  with $O_{\mathrm{w}}(\theta_{\mathrm{w}}) = e_{\mathrm{w}_1}\cos\theta_{\mathrm{w}} + e_{\mathrm{w}_2}\cos 2\theta_{\mathrm{w}}$ and $f_{\mathrm{w}}(r_{\mathrm{w}}) = \exp\big(-(r_{\mathrm{w}}/l_{\mathrm{w}})^2\big)$.
- Social interaction with the most influential neighbor:
  - Alignment:

$$\delta\phi_{\mathrm{Ali}}^{ij} = \gamma_{\mathrm{Ali}}(d_{ij} + d_{0_{\mathrm{Ali}}})e^{-\left(\frac{d_{ij}}{l_{\mathrm{Ali}}}\right)^2}\sin(\Delta\phi_{ij}) \times [1 - f_{\mathrm{w}}(r_{\mathrm{w}})]. \tag{10}$$

  - Attraction:

$$\delta\phi_{\mathrm{Att}}^{ij} = \gamma_{\mathrm{Att}}\frac{d_{ij} - d_{0_{\mathrm{Att}}}}{1 + \left(\frac{d_{ij}}{l_{\mathrm{Att}}}\right)^2}\sin(\psi_{ij}) \times [1 - f_{\mathrm{w}}(r_{\mathrm{w}})]q(d_{ij}), \tag{11}$$

    where $q(d) = 2d_{ij}/(4d_{ij} - d_{0_{\mathrm{Att}}})$ if $d\mathrm{Att} \leq d_{0_{\mathrm{Att}}}$ and $q(d) = 1$ if $d > d_{0_{\mathrm{Att}}}$, $d_{0_{\mathrm{Ali}}}$ and $d_{0_{\mathrm{Att}}}$ are the respective distances of equilibrium of these interactions, l_{Ali} and l_{Ali} are the ranges of interactions, and f_{w} is a term that reduces the strength of the interaction when the agent is close to the wall.

The speed cohesion interaction is defined as follow:

$$\delta V_{ij} = \gamma_{\mathrm{Acc}}\cos(\psi)\frac{d_{ij} - d_{\mathrm{V}_0}}{1 + \frac{d_{ij}}{l_{\mathrm{Acc}}}}. \tag{12}$$

The benefit of this method lies on how well adapted it is to operational contexts such as guiding the swarm towards a point, along a line or a corridor, in static or dynamic mode, and that the addition of new features is well documented in several biological models, as, *e.g.*, a migration term [10], [18]. Regarding obstacle avoidance, an interaction as the one in Eq. (9) can be adapted to different surfaces; in particular, wall avoidance can be treated as in [19].

Finally, the model has to be adjusted to consider the physical size of the UAVs used in our experiments. In the original fish model, interaction parameters of Eq. (2) are proportional to the mean body length of fish, as well as the measures used in the validation of the model. In order to fully adapt this model to our UAVs, a new set of parameters based on the typical interaction scales of UAVs is required in order to match a faith-full behavior at large scale.

III. ARCHITECTURE

The control architecture has been developed with ROS2[1] [20], a framework that facilitates data sharing through standardized message interfaces and is widespread in the field

[1]Robotic Operating System 2.0

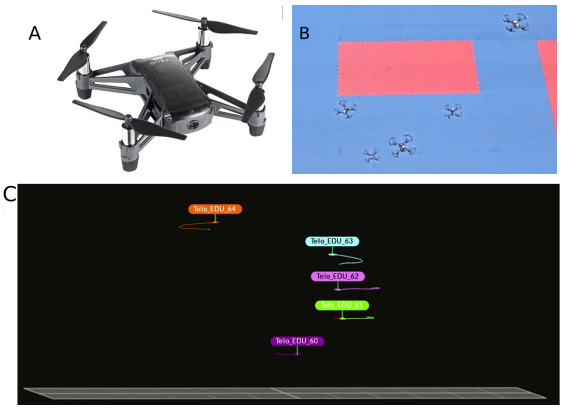


Fig. 3. [A] The experimental UAV, a Tello EDU®[3];[B] the view of the swarm within the flying arena (a 3-meter radius circle) and [C] A side view of the fleet through the motion capture tool used. UAVs fly in specific levels of altitude to avoid any collision, but are considered at the same height within the control algorithm

of robotics. ROS2 allows the separation of the modules for unit test and validation, the surveillance of the messages published in topics, and the recording of data streams, for future replays or analysis. The experimental setup consists of a UAV arena where an Optitrack® system broadcasts positions in the Wireless Network. A ROS2 node is then used to publish data in a suitable format, at high frequency (up to 100 Hz). The navigation code treats them to calculate a new heading and speed, to be sent to the UAV, a Tello EDU (Fig. 3A), which can only be controlled in speed $(v_x(t), v_y(t), v_z(t))$ and turning rate $\omega(t)$. Communication with Tello is made by UDP connection, and navigation control rate is 10 Hz (Fig. 4). Because the model is planar, the agents are separated vertically in the arena, at around 1 m from each other to avoid air perturbations and possible collisions during the experiments. A real view from the top of the arena is shown in Fig. 3B, and the vertical separation can be seen in Fig. 3C, in the motion capture view of the arena. Since the experimental UAVs don't allow on-board programming, the navigation is achieved externally. However, each drone is associated to a dedicated navigation node, only receiving information about its own state from the tracking system and states from surrounding drones from message communication.

The simulator is made as a ROS2 node reproducing the dynamics measured in the experimental setup. The same type of speed command is received, and the state of the simulated UAV is updated accordingly (see Figure 4). A Gaussian noise with a 1 degree deviation was added to the estimation of the heading, to fit experimental data.

One motivation for our work is to use such a control model to control the collective motion of many UAVs together in a distributed manner. After validating that several drones can fly together in an arena, we will need to validate the behavior of the simulator in order to match the results obtained in the experiments, with similar initial conditions and parameters.

[3]www.ryzerobotics.com

TABLE I

PARAMETERS SET FOR THE INTERACTIONS

γ_{w}	1.2	-	γ_{Ali}	1.2	-
γ_{Att}	0.4	-	γ_{Acc}	1.2	-
L_{w}	2.5	m	L_{Ali}	2.5	m
L_{Att}	2.5	m	L_{Acc}	2.5	m
$d_{0_{\mathrm{Ali}}}$	1.0	m	$d_{0_{\mathrm{Att}}}$	0.75	m
d_{Acc_0}	1.0	m	body length	0.25	m
e_{ω_1}	1.25	-	e_{ω_2}	0.0	-
V_{MIN}	0.4	m/s	V_{MAX}	0.7	m/s
V_0	0.5	m/s	$fluct$	0	-

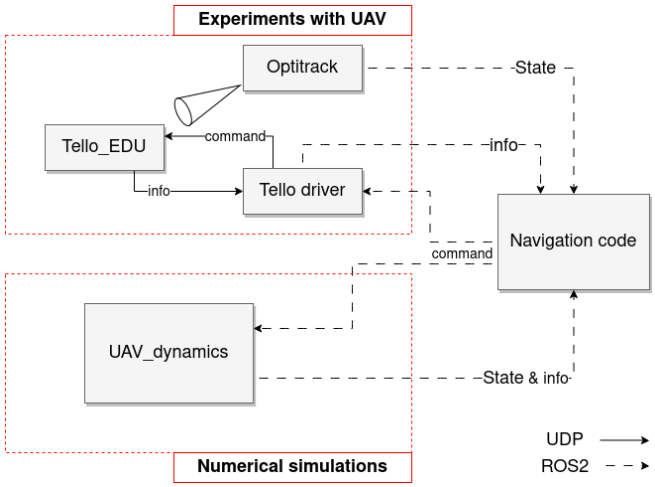


Fig. 4. System architecture. The upper red square shows the architecture used in the experiments with UAV, while the one in the bottom corresponds to the simulated dynamics. The different communications links are represented like displayed in the legend.

Once the simulator is validated, we can apply this control algorithm to real-life scenarios, and test the limits in terms of number of UAVs it can make fly while keeping the coordination.

IV. RESULTS

A set of experiments have been conducted inside a circular arena (of 3 meter radius) to validate the behavior of the swarm, and to match the values observed with those of the original model and those produced by our simulator. The results are divided into two categories:

- The validation of the control algorithm in a UAV arena, inside a fictive circular arena, using $N = 1$ and 5 UAVs.
- The validation of the simulator by comparing the measures of a set of statistical observables with those obtained in the experiments.

Regarding the adaptation of the model to a UAS, both the migration term and the fluctuation are set to zero during the experiments, as they serve for the study of navigation and phase transitions, which are not studied in this paper.

A. A single drone in the arena

The effect of the wall on the motion of a single drone inside a circular arena is described in Eq. (9).

Fig. 5 shows the trajectory of the drone both in the experiments and in the simulator, together with the probability density functions (PDF) in both the experiments and our simulations. We used 10 flights of the UAV and 10 runs of the model, each one with a mean duration of about 8 min.

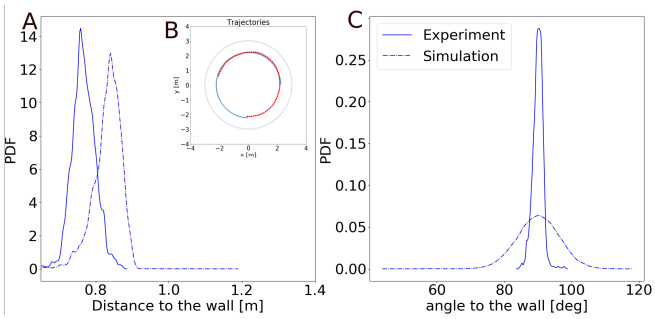


Fig. 5. Trajectories (B) and probability density functions of (A) r_w and (C) θ_w for 1 UAV flying in a circular arena of radius $R = 3\,m$. In the Trajectory (B), red markers represent the simulated trajectory, and the blue markers the trajectory in experiment. In the PDF, blue dotted lines correspond to numerical simulations of the model, full lines correspond to experimental data. We used 10 simulations and 10 experiments with a duration of 8 min on average. The plotted trajectory lasts about 17 s.

We first observe that trajectories are circular along the wall. As fluctuations are not considered in the model, the effect of the wall leads the system to an equilibrium state characterized by almost constants values of r_w and θ_w. In simulations and in experiments, the r_w values are as centered around their mean values, but the simulated mean value (83 cm) is a bit higher than in experiment (76 cm). This can be due to the inherent noise of the experimental model in measurement, control, and perturbations. Delays in the architecture can also induce a deviation in the mean value. Fig. 5C shows that θ_w is more scattered around its mean value of 90°, and the same thing happens with r_w, where the difference is negligible when compared to the size of the arena (3-meter radius). The simulation of the effect of the wall is therefore qualitatively satisfactory. The term modelling the effect of the wall being the basis of the avoidance term that we plan to introduce in the model, we consider that the simulator can be used for obstacle avoidance scenarios in lightly cluttered environment.

B. Five drones in the arena

This section shows the results of both the experiments and the simulations carried out for a swarm of $N = 5$ UAVs evolving inside the same arena as before.

In addition to the effect of physical obstacles (arena wall), the model has to consider the social interaction between drones described in Eqs. (10)–(11), together with a process of selection of the most influential neighbour with which each agent interacts.

Additional observables are considered when $N > 2$:

- The polarization of the swarm, showing the level of alignment between drones (1 for perfect alignment);
- The dispersion of the fleet, showing how close the agents are to the barycenter of the fleet;
- The distance to the closest neighbor.

Polarization and cohesion formulas can be found in [21].

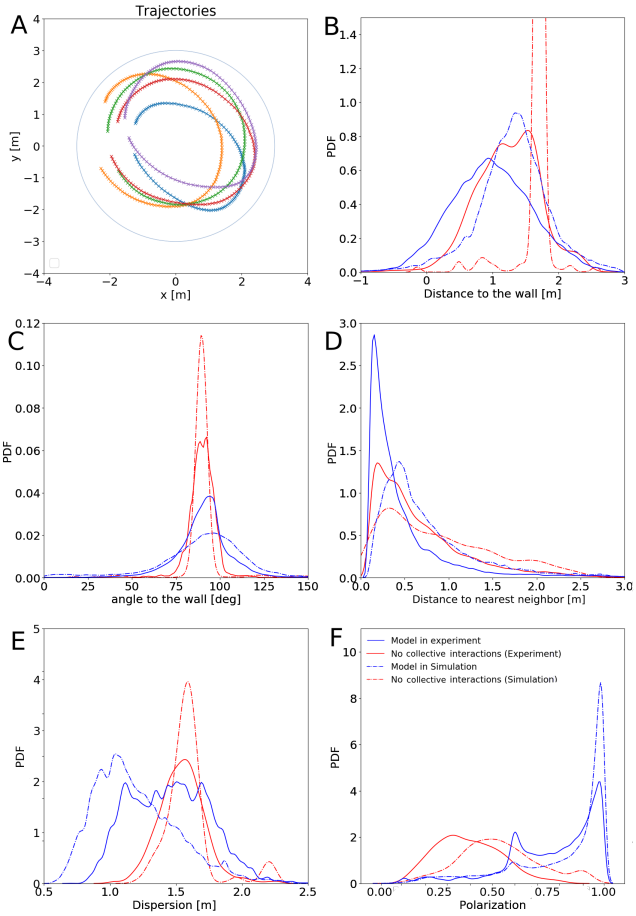


Fig. 6. (A) Trajectories of 5 UAVs in the arena of radius $R = 3\,m$ and PDF of the 5 observables: (B) Mean distance to the wall; (C): Mean angle of incidence to the wall; (D) Mean distance to nearest neighbor; (E) Dispersion, and (F) Polarization. Full lines correspond to experimental data, dotted lines to numerical simulations. Blue lines correspond to the case where drones interact with their most influential neighbor, red lines to the case where there is no social interaction between drones. Seven flights and seven runs of the model were carried out, all with a duration of about 8 min.

Although each run starts from a different initial condition, agents form a swarm in a relatively short time. Fig. 6C shows that the angle of incidence to the wall is centered around the same mean value than when flying alone, with a wider PDF when $N = 5$. Compared to the case where there is no social interactions, where agents just fly close to the wall, the balance between the distance at which the wall has an effect and the distance of equilibrium in the interactions induces a flattening in the PDFs of d_w and θ_w. The PDF of the mean distance to the nearest agent in the case where there is no social interactions (Fig. 6D) is more scattered than when social interactions are at play, despite the peak observed at low values. This can be explained by the fact that, in the absence of social interactions, agents remain

close to the border and therefore remain quite often at a more or less constant distance from each other. In Fig. 6F, the impact of social interactions is quite visible: agents are more often aligned and the polarization often reaches its maximum value. The peak of the polarization at 0.55 corresponds to frequent situations where a single agent is flying in the opposite direction than the other 4, due to the attenuation of the interaction when drones are close to the wall (via the term f_{w} in Eq. (10)). This attenuation term, coupled with social interactions, is also responsible of the smaller mean distance to the wall in Fig. 6B. Finally, Fig. 6E shows the effect that social interactions have on the dispersion of the swarm, whose PDF is shifted to smaller values than in the case where agents do not interact with each other. Agents are however confined in an arena of limited size. Experiments in a larger arena or a wider environment would have been more informative.

Qualitatively, the results obtained in the simulations are in good agreement with those observed in the experiments, the PDFs (mean value, peaks and general shape) shown in Fig. 6 being quite similar to each other. The impact of social interactions is clearly visible in the polarization of the swarm, and less visible both in dispersion and mean distance to the nearest neighbor. This is mainly due to the shape and size of the arena. The PDF of dispersion would have had a quite different profile in a larger arena, reaching higher values.

V. CONCLUSION AND FUTURE WORK

In this work, we have presented a distributed method for UAV flocking based on a fish school model. For the moment, the UAV model is a planar guidance algorithm where drones are kept flying at different heights. This feature will be implemented in a future formulation of the model through a decentralized decision-making process handled by each agent. In spite of this, the interactions implemented here allow the agents to display cohesive and polarized flies that are remarkably stable, especially taking into account the minimal number of neighbors considered in the interaction. With one influential neighbor, agents manage to maintain spatial cohesion, and alignment.

The model adapted to UAVs displays collective patterns that are very similar to those observed in nature. The simulator, qualitatively validated in this article, will allow us to test several operational contexts, especially planning, path following, obstacle avoidance, and migration (where UAVs are attracted by a destination point). The simulator allows also to test the emergence of these collective patterns in larger environments where border effects are not present. Performance will also be ensured in real scenarios to compare this method to other decentralized ones.

Moreover, as mentioned in [15] and [16], swarms of drones controlled by the UAV model are subject to phase transitions similar to those observed in physical active matter. By modifying the intensity of the interactions, the simulated flock can display different patterns of collective motion, in terms of polarization (how the agents are aligned), cohesion (how close the agents are to each other), and milling (how the flock is whirling around a common point).

This article shows the interest of adapting biological models for UAS. Distributed control through a flocking algorithm represents a proof of concept, as in [22]. The model presented in this work can be the basis for the control of larger swarms, with a wide range of practical applications such as obstacle avoidance, or air traffic management for UAV systems.

REFERENCES

[1] M. Coppola *et al.*, "A Survey on Swarming With Micro Air Vehicles: Fundamental Challenges and Constraints," vol. *Frontiers in Robotics and AI*, 2020, publisher: Frontiers.

[2] E. Soria *et al.*, "Predictive Control of Aerial Swarms in Cluttered Environments," 2020.

[3] N. Ahmed *et al.*, "Distributed Control and Estimation of Robotic Vehicle Networks: Overview of the Special Issue," vol. *IEEE Control Systems Magazine*, no. 2, pp. 36–40, Apr. 2016, conference Name: IEEE Control Systems Magazine.

[4] F.Y. Hadaegh *et al.*, "On Development of 100-Gram-Class Spacecraft for Swarm Applications," vol. *IEEE Systems Journal*, no. 2, pp. 673–684, Jun. 2016, conference Name: IEEE Systems Journal.

[5] M. Dorigo *et al.*, "Swarm Robotics: Past, Present, and Future [Point of View]," vol. *Proceedings of the IEEE*, no. 7, pp. 1152–1165, Jul. 2021.

[6] T. Verdu *et al.*, "Flight patterns for clouds exploration with a fleet of UAVs," vol. in *2019 International Conference on Unmanned Aircraft Systems (ICUAS)*, Jun. 2019, pp. 231–237, iSSN: 2575-7296.

[7] T. Vicsek *et al.*, "Collective motion," vol. *Physics Reports*, no. 3-4, pp. 71–140, Aug. 2012.

[8] Y. Chen *et al.*, "Controlled flight of a microrobot powered by soft artificial muscles," vol. *Nature*, no. 7782, pp. 324–329, Nov. 2019.

[9] L.E. Beaver *et al.*, "An overview on optimal flocking," vol. *Annual Reviews in Control*, pp. 88–99, 2021.

[10] A. Deutsch *et al.*, "Multi-scale analysis and modelling of collective migration in biological systems," vol. *Philosophical Transactions of the Royal Society B: Biological Sciences*, no. 1807, p. 20190377, Sep. 2020, publisher: Royal Society, The.

[11] A. Cavagna *et al.*, "The physics of flocking: Correlation as a compass from experiments to theory," vol. *Physics Reports*, pp. 1–62, Jan. 2018.

[12] C.W. Reynolds, " vol. "Flocks, Herds, and Schools: A Distributed Behavioral Model, p. 21, 1987.

[13] S. Chung *et al.*, "A Survey on Aerial Swarm Robotics," vol. *IEEE Transactions on Robotics*, no. 4, pp. 837–855, Aug. 2018, conference Name: IEEE Transactions on Robotics.

[14] F. Belkhouche *et al.*, "Modeling and controlling 3D formations and flocking behavior of unmanned air vehicles," vol. in *2011 IEEE International Conference on Information Reuse & Integration*. Las Vegas, NV, USA: IEEE, Aug. 2011, pp. 449–454.

[15] D. Calovi *et al.*, "Disentangling and modeling interactions in fish with burst-and-coast swimming reveal distinct alignment and attraction behaviors," vol. *PLOS Computational Biology*, p. e1005933, Jan. 2018.

[16] L. Liu *et al.*, "Computational and robotic modeling reveal parsimonious combinations of interactions between individuals in schooling fish," vol. *PLOS Computational Biology*, p. e1007194, Mar. 2020.

[17] J. Gautrais *et al.*, "Deciphering Interactions in Moving Animal Groups," vol. *PLoS Computational Biology*, no. 9, p. e1002678, Sep. 2012.

[18] A. Deutsch *et al.*, "Collective motion in biological systems," vol. *Interface Focus*, no. 6, pp. 689–692, Dec. 2012, publisher: Royal Society.

[19] R. Olfati-Saber, "Flocking for multi-agent dynamic systems: algorithms and theory," vol. *IEEE Transactions on Automatic Control*, no. 3, pp. 401–420, Mar. 2006, conference Name: IEEE Transactions on Automatic Control.

[20] "ROS2 (Robotic Operating System 2.0). Available at: https://github.com/ros2."

[21] Y. Liu *et al.*, "Collision free 4D path planning for multiple UAVs based on spatial refined voting mechanism and PSO approach," vol. *Chinese Journal of Aeronautics*, no. 6, pp. 1504–1519, Jun. 2019.

[22] G. Vásárhelyi *et al.*, "Optimized flocking of autonomous drones in confined environments," vol. *Science Robotics*, no. 20, Jul. 2018, publisher: Science Robotics Section: Research Article.