



Collective gradient perception with a flying robot swarm

Tugay Alperen Karagüzel¹ · Ali Emre Turgut² · A. E. Eiben¹ · Eliseo Ferrante^{1,3}

Received: 3 January 2022 / Accepted: 27 September 2022 / Published online: 26 October 2022
© The Author(s) 2022

Abstract

In this paper, we study the problem of collective and emergent sensing with a flying robot swarm in which social interactions among individuals lead to following the gradient of a scalar field in the environment without the need of any gradient sensing capability. We proposed two methods—desired distance modulation and speed modulation—with and without alignment control. In the former, individuals modulate their desired distance to their neighbors and in the latter, they modulate their speed depending on the social interactions with their neighbors and measurements from the environment. Methods are systematically tested using two metrics with different scalar field models, swarm sizes and swarm densities. Experiments are conducted using: (1) a kinematic simulator, (2) a physics-based simulator, and (3) real nano-drone swarm. Results show that using the proposed methods, a swarm—composed of individuals lacking gradient sensing ability—is able to follow the gradient in a scalar field successfully. Results show that when individuals modulate their desired distances, alignment control is not needed but it still increases the performance. However, when individuals modulate their speed, alignment control is needed for collective motion. Real nano-drone experiments reveal that the proposed methods are applicable in real-life scenarios.

Keywords Collective sensing · Flocking · Collective motion · UAV swarm

✉ Tugay Alperen Karagüzel
t.a.karaguzel@vu.nl

✉ Eliseo Ferrante
e.ferrante@vu.nl

Ali Emre Turgut
aturgut@metu.edu.tr

A. E. Eiben
a.e.eiben@vu.nl

¹ Vrije Universiteit Amsterdam, Amsterdam, The Netherlands

² Middle East Technical University, Ankara, Turkey

³ Technology Innovation Institute, Abu Dhabi, UAE

1 Introduction

Understanding and designing the behavior of large swarms of robots is an endeavour that is highly relevant both in science and engineering. From the scientific perspective, a long-standing challenge is to understand how complex collective sensing, information processing, and synchronous dynamics emerge from individuals that are simple, embodied, independent in their cognitive processes, and interact only through local sensing and communication (Şahin, 2004; Brambilla et al., 2013). Indeed, there are many examples of natural systems such as animals and insects that are able to move together in a coordinated fashion while performing complex collective sensing and coordination actions (Couzin et al., 2005), such as identifying and moving toward a resource location (Couzin et al., 2005; Kearns, 2010) or avoiding predators (Olson et al., 2013). This is only possible if the rules for individuals are designed in such a way that they only use local information and local sensing.

The collective behavior of a swarm is not necessarily the direct and traceable result of the individual behaviors. In fact, it is not only the individual behaviors that determine the group behavior, but also the interactions between individuals and between individuals and the environment. Here, we address the problem of collective and emergent sensing (Berdahl et al., 2018), where social interactions among individuals lead to collective computation of an environmental property by only using local scalar measurements. A good example of this problem is the taxis behavior defined as moving toward or away from a given physical feature (Gorban & Çabukoğlu, 2018) such as light or temperature. This is known to be solvable by living organisms, for instance by golden shiner fish (Puckett et al., 2018; Berdahl et al., 2013). A school of golden shiners incapable of sensing the light gradient can collectively sense and follow the decreasing light gradient by making local light intensity measurements and considering social cues of their group members.

In this paper, we aim to develop two methods—desired distance modulation and speed modulation—with and without alignment control that can work along with a well-studied flocking algorithm (Ferrante et al., 2014) for collective gradient following with an aerial robot swarm. The proposed methods ensure collective and emergent sensing of the gradient in a scalar field and robots follow the increasing gradient direction in a cohesive and ordered manner. In order to test our goals, methods are first tested with kinematic and physics-based simulators, then by using a swarm of nano-aerial vehicles. Moreover, we test the scalability of the methods by using swarms with different sizes in simulations and we analyze behaviors in collective level on various scalar field models such as Linear, Bimodal and Circle to assess dependency of behaviors on changing environment. In addition, the density (ratio between the swarm area and the bounded area) is investigated.

2 Related work

Collective sensing and taxis have been studied in the context of both biological and artificial systems. In biological systems, bacteria groups are a good first example to study, because of their limited sensory, memory and motion capabilities (Shklarsh et al., 2011; Camley et al., 2016). Shklarsh et al. (2011) proposed a bacteria-inspired collective navigation method based on changing social interactions. In their method, simple agents vary the strength of their interactions based on the concentration of a chemical substance and

the bacteria swarm was able to reach the lowest concentration region. Another example of collective sensing is seen on fish schools. In (Puckett et al., 2018; Berdahl et al., 2013), the anti-phototaxis behavior of fish schools is studied. The results showed that a fish school can effectively steer toward darker regions collectively provided that each fish modulates its speed based on the light intensity it detects locally. This speed modulation method is the inspiration to one of the methods proposed in this paper. Kernbach et al. (2009) studied the thermotactic aggregation behavior of young honey bees. Each honeybee when encounters another honeybee stops for a certain waiting time. By only modulating the waiting time based on local temperature measurement of each bee, it was observed that bees are able to aggregate on the optimal temperature gradient on the environment.

There are also many examples of artificial systems designed to make collective level estimation of environmental properties. In (Wahby et al., 2019), previously mentioned honeybee-inspired aggregation method (Kernbach et al., 2009) is improved with an ability to aggregate on different upper and lower values of temperature gradient in the environment using only local communication. In (Campo et al., 2011), an aggregation task with shelter selection is considered. Robots using only local density estimation are able to aggregate in a shelter that fits better the swarm size. In (Valentini et al., 2016), the robot swarm collectively estimates the most frequent tile color of the ground using communication to perform voting or exchange their predictions. Some studies related to division of labor also show that modifications of social interactions due to environment can also produce collective level solutions to more realistic problems. In (Zahadat & Schmickl, 2016), underwater robots demonstrate a fully distributed dynamic and efficient task allocation among the swarm by only relying on local interactions. In this method, which is inspired by honeybee age-polyethism, interactions also involves communication. Alternatively, it is shown that (Carreón et al., 2017) social interactions and limited environmental observations of agents can be used in real-life problems like regulation of a public transport service. By locally observing the approaching trains and a virtual pheromone assigned to each stop, Carreón et al. (2017) showed that a whole metro system can be organized in a distributed fashion.

We now discuss the literature that is the most related to the work presented in this paper. In (Schmickl et al., 2016), a swarm of simulated individuals can follow the decreasing gradient by a method inspired by slime molds. The algorithm works by individuals exchanging messages (“pings”) with frequencies correlated with their local estimate of the gradient. The requirement of individuals to estimate the direction of “pings” potentially comes with challenges for a real-world application. Neither this method nor its more recent adaptations (Varughese et al., 2019) are implemented on a real platform. Bjercknes et al. (2007) present a truly distributed and emergent collective taxis on a simulated swarm with highly realistic features and constraints. Robots in an area illuminated by a beacon modify their desired distances to the rest of the swarm. The ones that are illuminated by the beacon tries to be more distant to the others, the swarm moves toward the beacon. Although there is a similarity between this approach and one of our methods, individuals move in a staggered and slower way in (Bjercknes et al., 2007) without a group order, therefore in this work collective taxis emerges but not collective motion rendering it not suitable for aerial swarms. In addition, their aim is more toward achieving single source localization rather than gradient following. Shaukat and Chitre (2016) presents an underwater application with autonomous robots to localize an RF beacon. Robots employ a method that eventually results in adaptive group coherence, depending on proximity to the source. Although they demonstrated the effectiveness of adaptive group coherence for a collective and emergent source localization on real underwater platforms, robots used directional information of

signal measurements by employing two receivers on left and right side of the platforms. Duisterhof et al. (2021) demonstrates a relevant example of a collective search for a gas source with a nano-drone swarm. They showed that local information exchange between on-board controlled drones and local gas sensing are sufficient to locate the source collectively. Differently from our approach, agents use waypoints to navigate and share their sensing information with each other. Despite that, when we consider the current challenges with aerial vehicles (Coppola et al., 2020) such as on-board peer sensing and autonomous flight without any external positioning system, Duisterhof et al. (2021) presents a very promising real-life application within the given hardware constraints.

With respect to the most similar studies just discussed, the work presented in this paper differs from them by: demonstrating an emergent taxis behavior with the constraints of real aerial platforms using only local and scalar measurements rather than directional information about the gradient, avoiding any external information exchange about the scalar measurements among the individuals while having an ordered and cohesive collective motion.

3 Methodology

We consider a swarm composed of N individuals that can move in a 2D bounded environment. The environment contains a certain fictitious substance that is spread in the arena in a non-uniform way, through a gradient. This gradient is modeled as a very fine grid where each cell has a scalar value. The grid modeling of the environment serves both to simplify the computations as well as to model the finite sensing resolution of individual robots. Individuals can sense the local value in the grid cell in which they are currently located in. Yet individuals move continuously, independent from the cell structure of the environment. In addition, individuals can sense the relative distances and bearings of other individuals within a limited sensing range (D_p). Additionally, individuals can sense their distance to the environmental boundaries if they are in the boundary sensing range (D_r). Movements of the individuals follow the non-holonomic constraints: At each time instant, the focal individual has a direction of motion, which is called the heading and it can only move along this heading direction based on its linear speed. The heading direction changes according to the angular velocity of the individual. The details of physics used during simulations depend on the specific simulator platform considered. In the following, we explain first the standard collective motion method (Sect. 3.1); subsequently, we introduce desired distance modulation (Sect. 3.2) and speed modulation (Sect. 3.3) methods that can achieve collective gradient following and a baseline method for individual gradient following (Sect. 3.4).

3.1 Standard collective motion

The standard collective motion (SCM) method is used by the individuals to reach and maintain collective motion in a cohesive and ordered fashion. At each time instant, the focal individual i calculates a virtual force \vec{f}_i as follows:

$$\vec{f}_i = \alpha \vec{p}_i + \beta \vec{h}_i + \gamma \vec{r}_i \quad (1)$$

The virtual force consists of three vectors: The proximal control vector \vec{p}_i , the alignment control vector \vec{h}_i , and the boundary avoidance vector \vec{r}_i . The proximal control vector \vec{p}_i generates a spring-like effect between neighboring individuals: If the focal individual and

its neighbor are closer than the desired distance, \vec{p}_i becomes a repulsive force, otherwise it is an attractive force unless two robots are exactly at the desired distance. The alignment control vector \vec{h}_i contributes to maintain a consensus between the heading of the focal individual and the average heading of its neighbors. It was shown in previous studies (Ferrante et al., 2012) that collective motion can be achieved without alignment control vector, therefore in this paper, we consider alignment control as “optional” and perform experiments with and without it. Lastly, the boundary avoidance vector \vec{r}_i ensures a safe distance between the environmental boundaries and the focal individual. The weights α , β and γ determine the relative contribution of the corresponding virtual force vectors.

Proximal control is always enabled by default to guarantee collision-free cohesive motion (Turgut et al., 2008; Ferrante et al., 2012, 2014). At each time instant, \vec{p}_i is calculated for each neighbor (m) of the total \mathcal{R} neighbors by the focal individual in the sensing range, D_p . The proximal control vector is calculated as:

$$\vec{p}_i = \sum_{m \in \mathcal{R}} p_i^m(d_i^m, \sigma_i) \mathcal{L}e^{j\phi_i^m} \tag{2}$$

Here, $p_i^m(d_i^m, \sigma_i)$ and $\mathcal{L}e^{j\phi_i^m}$ are the magnitude and the angle of the vector i pointing at each perceived individual, respectively. The magnitude of the i th vector is calculated using the following virtual force function derived from a modified Lennard–Jones potential function, as:

$$p_i^m(d_i^m, \sigma_i) = -\epsilon \left[2 \frac{\sigma_i^4}{(d_i^m)^5} - \frac{\sigma_i^2}{(d_i^m)^3} \right] \tag{3}$$

where d_i^m is the relative distance of m sensed by i , ϵ is the gain of the force function and σ_i is the desired distance coefficient between i and its neighbors. The relation between desired distance of i and σ_i is as follows:

$$d_{des}^i = 2^{1/2} \sigma_i \tag{4}$$

We indicate both d_i^m and σ_i as variables because the value of the former is the distance of neighbor m to focal individual i which is continuously changing and the latter is the desired distance coefficient of individual i which changes in the desired distance modulation method explained later.

Alignment control is calculated by normalizing the sum of the headings of each neighbor in the sensing range D_p , together with the heading of the focal individual i . \vec{h}_i is calculated as follows:

$$\vec{h}_i = \frac{\mathcal{L}e^{j\theta_0} + \sum_{m \in \mathcal{R}} \mathcal{L}e^{j\theta_m}}{\|\mathcal{L}e^{j\theta_0} + \sum_{m \in \mathcal{R}} \mathcal{L}e^{j\theta_m}\|} \tag{5}$$

where $\mathcal{L}e^{j\theta_m}$ refers to the heading of individual m and the heading of the focal individual i is indicated as $\mathcal{L}e^{j\theta_0}$. Headings are calculated with respect to a common frame of reference for all individuals. In a real application, this frame of reference can be implemented either by a digital compass “common north” (Turgut et al., 2008) or via a shared directional signal such as a light source (Ferrante et al., 2014).

Boundary avoidance is calculated for every edge of the polygonal boundary of environment within the boundary perception range D_r

$$\vec{r}_i = \sum_{b \in B} \vec{r}_i^b \tag{6}$$

For every perceived edge (b) among all simultaneously perceived edges (B) of the polygonal boundary of environment, the magnitude of the avoidance vector (Khaldi & Cherif, 2016) is calculated as follows:

$$\vec{r}_i^b = k_{\text{rep}} \left(\frac{1}{L_b} - \frac{1}{L_0} \right) \begin{pmatrix} \vec{p}_i^b \\ L_b^3 \end{pmatrix} \tag{7}$$

k_{rep} is the gain of the avoidance vector, L_0 is the relaxation threshold of the function and L_b is the shortest distance to the edge b . The unit vector p_i^b indicates the direction of closest point on the edge b in the frame of reference of individual i .

Motion control is achieved by using \vec{f}_i to calculate the desired linear and angular speeds of the focal individual i at a time instant as in (Ferrante et al., 2012). First, \vec{f}_i is projected on the two orthogonal axes of i 's local frame of reference. It is a right-handed reference frame in which the x -axis is parallel to the heading of individual i . Figure 1 visualizes each step of SCM for a focal agent and 2 neighbors to the point where f_x and f_y are found. The linear and angular speeds are calculated as follows:

$$U_i = K_1 f_x + U_c \tag{8}$$

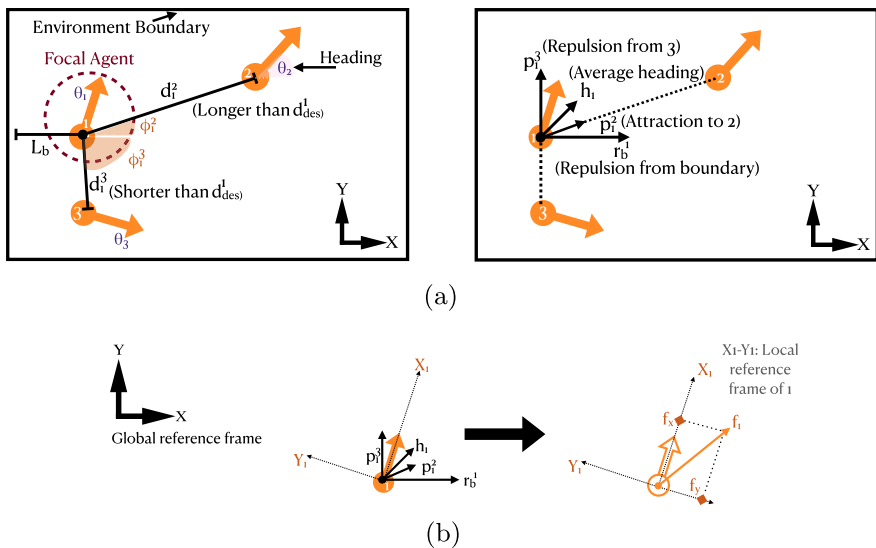


Fig. 1 Step by step visualization of how total virtual force vector on the focal agent is calculated in SCM and components on the local reference frame are found. (a) First, the focal agent measures the relative distances (d_1^2 , d_1^3) and relative angles (ϕ_1^2 , ϕ_1^3) of neighbours within the perception range D_p . In addition, the distance to any boundary (L_b) within the range D_r is measured. Optionally, the average heading of neighbours is also found (average of θ_1 , θ_2 and θ_3). Later, all measured distances and angles are used in the corresponding formulas to calculate the virtual force components for proximal control (p_1^2 , p_1^3), alignment control (h_1) and boundary avoidance r_b^1 . (b) The resultant force vector f_1 is calculated by summing all the force vectors. Finally, f_1 is projected onto local reference frame of the focal agent. Components of f_1 in this local frame (f_x and f_y) are used to calculate linear and angular speeds. α , β , and γ are set to 1 for simplicity

$$\omega_i = K_2 f_y \quad (9)$$

The linear speed U_i is determined by multiplying the x component of \vec{f}_i (f_x) by the linear speed gain K_1 and adding it to the bias speed, U_c . The angular speed is determined by multiplying the y component of \vec{f}_i (f_y) with the angular speed gain K_2 .

The linear and angular speeds of i are bounded as follows:

$$U_i = \begin{cases} 0 & U_i \leq 0 \\ U_i & 0 < U_i < U_{\max} \\ U_{\max} & U_i \geq U_{\max} \end{cases}$$

$$\omega_i = \begin{cases} -\omega_{\max} & \omega_i \leq -\omega_{\max} \\ \omega_i & 0 < \omega_i < \omega_{\max} \\ \omega_{\max} & \omega_i \geq \omega_{\max} \end{cases}$$

3.2 Desired distance modulation

In the standard collective motion (SCM) method, the desired distance between the focal individual i and its perceived neighbors is determined by the σ_i term, which is constant and the same for all individuals. In the first proposed method Desired Distance Modulation (DM), we instead change σ_i for every individual to correlate the desired distance with the local perceived value of the scalar field, G° , in the environment as follows:

$$\sigma_i = \sigma_{\min} + \left(\frac{G^\circ}{G_{\max}} \right) (\sigma_{\max} - \sigma_{\min}) \quad (10)$$

where σ_{\max} and σ_{\min} are the maximum and minimum values of σ_i , respectively. G_{\max} is the maximum local value in the environment. We expect the agents perceiving higher local values to have higher desired distance values hence they tend to be more distant to their neighbors. On the contrary, neighbors perceiving lower local values have smaller desired distance values making them to be closer to their neighbors. This symmetry breaking within the swarm creates a tendency to move toward higher local values.

3.3 Speed modulation

The second proposed method, Speed Modulation (SM), introduces two alterations on the control of the individuals:

- The focal individual i amplifies the repulsive component of the calculated proximal control vector. In particular, the neighbors of the focal agent which are closer than the desired distance starts to produce stronger repulsion forces than they would be producing with SCM. This magnification of the repulsive forces increases when the focal individual is perceiving higher local values.
- The focal individual i decreases its calculated linear speed according to the local value.

The first item on the list requires an additional element to calculate \vec{f}_i :

$$\vec{f}_i = R_a \delta \vec{p}_i^r + \alpha \vec{p}_i + \beta \vec{h}_i + \gamma \vec{r}_i \quad (11)$$

where δ is the weight of this new effect, virtual repulsion vector \vec{p}_i^r , which consists of repulsive forces in \vec{p}_i . R_a , is the repulsion weight calculated as a function of the local value as:

$$R_a = \left(\frac{G^\circ}{G_{\max}} \right) \quad (12)$$

Next, the calculated linear speed U_i of the focal individual i is modulated according the local value as:

$$U_i = \max \left\{ U_i \left[1 - \mathcal{P}_u \left(\frac{G^\circ}{G_{\max}} \right)^{\mathcal{E}_u} \right], U_{\min} \text{sgn}(U_i) \right\} \quad (13)$$

where \mathcal{P}_u is the portion of U_i to be modified, \mathcal{E}_u is the local value correlation exponent and U_{\min} is the minimum linear speed value as a lower threshold after the modulation. In SM, when the focal agent reads a higher local value, it slows down. When it slows down, its neighbors will temporarily be closer than the desired distance producing a combined repulsive force. To increase the strength of this repulsive force, we modify its strength (R_a). Since the portion of the swarm on higher local values will be slowed down and pushed out by their neighbors, the rest will be pulled toward the higher local values creating a collective gradient following effect.

3.4 Baseline method: single gradient follower

When employing one of the two proposed methods, the swarm is expected to show a collective taxis behavior by following the increasing gradient direction in the environment without explicit gradient sensing. As a baseline method, an algorithm is proposed for a solitary individual. The individual is assumed to perceive the local values on the circumference of its sensing circle. By knowing the local values, it can follow the best direction where the gradient increases. This method is used as the baseline of comparison with the proposed methods. This comparison could be helpful on deciding if the emergent sensing of the gradient by the swarm is comparable with a genuine gradient sensing of a single individual. The desired linear and angular speeds of the single individual is calculated as a function of the resultant virtual force vector as follows:

$$\vec{f}_i = \theta \vec{g}_i + \gamma \vec{r}_i \quad (14)$$

The goal direction vector, \vec{g}_i , is a unit vector in the direction of maximum positive change in local values between the individual i 's current location and perceived values on the sensing circumference. θ is the relative weight of the goal direction vector. The motion control method of the single individual is exactly the same used in the other methods.

4 Multi-agent simulations

We first perform an in-depth analysis of the proposed methods on an efficient kinematic multi-agent simulator. Multi-agent simulations are the first step to assess the success and scalability of the proposed methods in addition to their robustness over changing environment models. Despite its simplicity, the simulator is powerful enough to perform a comprehensive analysis while systematically varying a number of key parameters.

4.1 Experimental setup

In multi-agent simulations, a swarm of N point agents is considered. The positions and the headings of the agents, which are both continuous variables, are updated with discrete integration steps. For all experiment settings, the control time step dt is chosen to be 0.05 simulated seconds (ss). The position and heading updates also involve a uniformly distributed random error in the range $([-e_p, e_p])$. Another error term with the same characteristics is added to the perceived distance of neighboring peers in each local axis of the focal agent, in the range of $([-e_n, e_n])$.

Four different factors are considered in the experiments: Method, swarm size, scalar field model to represent the environmental gradient, and swarm density. Concerning the first factor, we consider the two proposed methods, Desired Distance Modulation (DM) and Speed Modulation (SM), both with and without heading alignment control (HA), in order to assess each method's individual effect and the one combined with HA. In addition, we consider the Standard Collective Motion (SCM), as well as the baseline method, Single Gradient Follower (SGF), that will be tested with a single agent only, making a total of five method combinations. For the second factor, except for SGF, different swarm sizes are considered for each of the different settings: 5, 50 and 100. For the third factor, we considered seven different scalar field models to reveal correlations between certain gradient patterns and collective responses if there is any. Their visualization and naming are depicted in Fig. 2. In addition, two snapshots with different experiment settings are shown in Fig. 3.

To implement scalar field models, the environment is divided into grids. For all experimental settings, the length of these grids (Δ_g) are taken as 0.04 units. A uniformly distributed random error $([-e_g, e_g])$ is added to the perceived local value. Concerning the last experimental factor, the swarm density level (DL) is defined as the ratio between the area the swarm covers and the environment's total area. To impose particular values of swarm density levels, we assume that the area that the swarm covers is proportional to the total number of agents. With this assumption, we can choose environments with growing areas in a way that is proportional to the total number of agents. A summary of this proportional relationship for different environments are shown in Table 1.

For each setting, experiments are repeated 100 times. For each run, a centroid location is chosen randomly in order to eliminate the effect of initial conditions on the performance of the swarm. Finally, agents are initialized around the centroid location such that they are all in the sensing range of each other. The termination criteria of each experiment is chosen to be linked to the trajectory length of the centroid of the swarm. The cutoff value for trajectory length is taken as 2.5 times of the edge length of the experiment arena. The experiment is finalized when the trajectory length reaches the cutoff value, or when total time limit (640,000 simulated seconds) is exceeded.

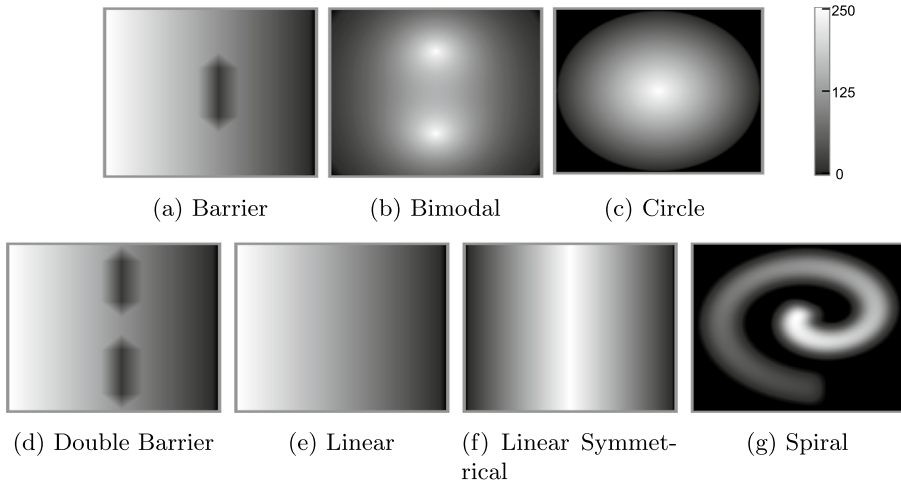


Fig. 2 Visualization of scalar field models used in the experiments; Brighter regions indicate higher scalar values

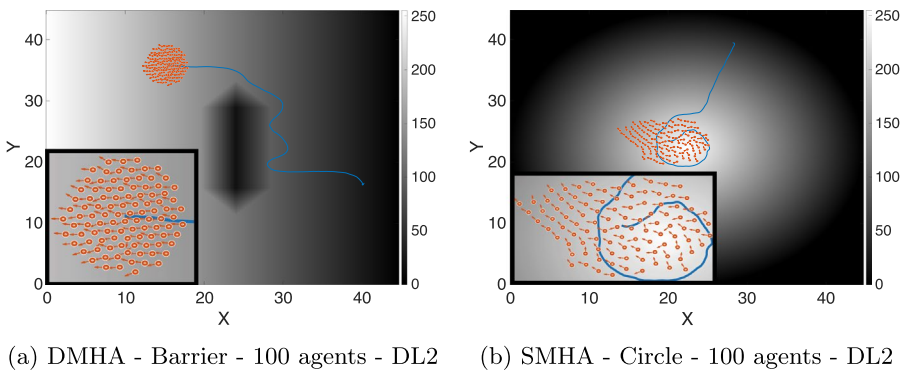


Fig. 3 Snapshots from two different multi-agent simulation runs; **a** DMHA on the Barrier model and **b** SMHA on the Circle model, blue line indicates trajectory of the swarm centroid and arrows show the heading direction of corresponding agents

Table 1 Edge dimensions and areas of square shaped environments in units, corresponding to density level and swarm size

Swarm size	Edge length	Env. area	Edge length	Env. area
	DL1	DL1	DL2	DL2
5	40 units	1600 unit ²	10 units	100 unit ²
50	126.48 units	16000 unit ²	31.62 units	1000 unit ²
100	178.88 units	32000 unit ²	44.72 units	2000 unit ²

The numerical values of the common parameters are given in Table 2. The method specific parameters are presented in Table 3. Methods require different parameter values to perform at their best and we tune these values manually for different methods. For example tuning α implies an increased weight of proximal control when compared to other effects.

Table 2 Values of common parameters in all experiment mediums

Parameter	Value
N : Total number of agents	5, 50, 100
Δ_g : Grid edge size (units)	0.04
D_p : Perception range (units)	2.0
ϵ : Proximal control strength coefficient	12.0
D_r : Boundary perception range (units)	0.5
γ : Boundary avoidance weight coefficient	1.0
k_{rep} : Boundary avoidance strength coefficient	2.0
L_0 : Avoidance vector relaxation threshold	0.5
U_{max} : Maximum linear speed (units/ t)	0.15
U_{max} (SGF): Maximum linear speed (units/ t)	0.1
U_c : Constant speed addition (units/ t)	0.05
G_{max} : Maximum grid value	255.0
e_p : Position update random error range (units)	$[\mp 0.05]$
e_n : Peer perception random error range (units)	$[\mp 0.05]$
e_g : Local value perception random error range	$[\mp 0.5]$
dt : Control time step (ss)	0.05

Table 3 Values of method specific parameters in multi-agent simulations

Parameter	DM	DMHA	SM	SMHA	SCM	SGF
α : Proximal cont. weight coeff.	2.0	0.5	0.5	0.6	1.0	–
β : Alignment cont. weight coeff.	–	2.0	–	2.0	2.0	–
K_1 : Linear speed gain	0.5	0.02	0.03	0.03	0.03	0.4
K_2 : Angular speed gain	0.05	0.4	0.4	0.4	0.4	0.1
ω_{max} : Max. angular speed (rad/ t)	$\pi/4$	$\pi/4$	$\pi/4$	$\pi/4$	$\pi/3$	$\pi/4$
σ_{min} : Max. desired distance coeff.	0.3	0.3	–	–	–	–
σ_{max} : Min. desired distance coeff.	0.6	0.6	–	–	–	–
\mathcal{P}_u : Modified U_i portion	–	–	0.5	1.0	–	–
\mathcal{E}_u : Local val. correlation exp.	–	–	2.0	10.0	–	–
θ : Goal direction weight coeff.	–	–	–	–	–	5.0

This is why DM have larger value of α when compared to DMHA since DM lacks alignment control. Speed gains also differ for the needs of different methods: Reflecting the resultant force vector on angular speed (K_2) is more important on SM and SMHA when compared to DM. Speed modulating methods perform better when agents control their headings more sensitively.

Methods have certain parameters only defined for themselves. For DM and DMHA, these parameters are σ_{min} and σ_{max} . They tune the amplitude of desired distance asymmetry within the swarm. Since they should be chosen sufficiently large but not freely. Larger values can cause separations in the group. Whereas \mathcal{P}_u and \mathcal{E}_u tune what portion of linear speed will be modulated and how strong it will get affected from the local value. Modifying the speed with insufficient strength and portion may shadow the effect of our method SM. Whereas larger than sufficient values may harm the ability of swarm to collectively move.

4.2 Metrics

In order to measure the performance of the swarm in following the gradient toward the *increasing values* of the scalar field, we define and use a *gradient following performance* (GFP) metric. This metric is calculated as follows: First, at the end of each run, the centroid trajectory -which is equal in length for all methods on a particular experiment setting- is divided into equal-length segments (Δd) and control points are specified as the end point of each segment. This step ensures all experiment runs produce the same number of control points (N_d) of the local value perceived by swarm centroid. Second, the local value of the cell where the swarm centroid falls is recorded ($G^\circ(n_i)$, $n_i \in N_d$). Third, the local values observed on each control point of a trajectory, belonging to a single experiment run, are summed. Finally, the summation is normalized by using the result of *the baseline method* (SGF) at the same experiment settings. Recorded values of SGF and each trajectory segment are denoted as $G^\circ(n_i^*)$ and n_i^* , respectively. It is also important to state that the experiments for SGF are repeated 10 times more (1000 in total) compared to all the others and the best of 10 is chosen for each experiment configuration to remove any effect of randomness. GFP can be formulated as follows:

$$\text{GFP} = \frac{\sum_{n_i \in N_d} G^\circ(n_i)}{\sum_{n_i^* \in N_d} G^\circ(n_i^*)} \quad (15)$$

GFP can take any value between 0 and 1. When GFP approximates 1 it means the performance of the chosen method producing similar results with our baseline method (SGF). The opposite case simply implies swarm always perceived local values close to zero. There are two edge cases to be considered during the calculation of GFP: a case that the swarm could not reach the designated trajectory length in total time limit and a case that the swarm is no longer a single group which means that the location of the centroid is no longer a valid indicator. For the former case, a fair way to compute the metric is to consider the trajectory pieces collected up to the time limit. In the latter case, the situation is not acceptable for our objectives. Therefore, if group separation occurs, the gradient following performance is recorded as 0 for that run. To determine whether separation occurs, we calculate the *number of groups* at a certain time instant. This is done by considering pairs of agents that are located within their sensing range. Afterwards, the pairs are appended to a list of equivalence pairs and the method of equivalence class is used to assign an equivalence class to each pair. Total number of equivalence classes gives us the total number of groups.

4.3 Results

In Fig. 4, we report GFP for swarms of the largest size (100 agents), for all scalar field models and for two different swarm densities. As expected, standard collective motion SCM fails to keep up with the baseline method SGF, since the swarm simply wanders around in the arena and averages all the local values it measures along the way. That is why SCM has different values among different scalar field models depending on bright to dark ratio in the model. Corresponding values can be observed on Fig. 4a.

We now focus on the performance of each method on different environments. Figure 4b reports the results for the Barrier scalar field model. Despite the fact that with

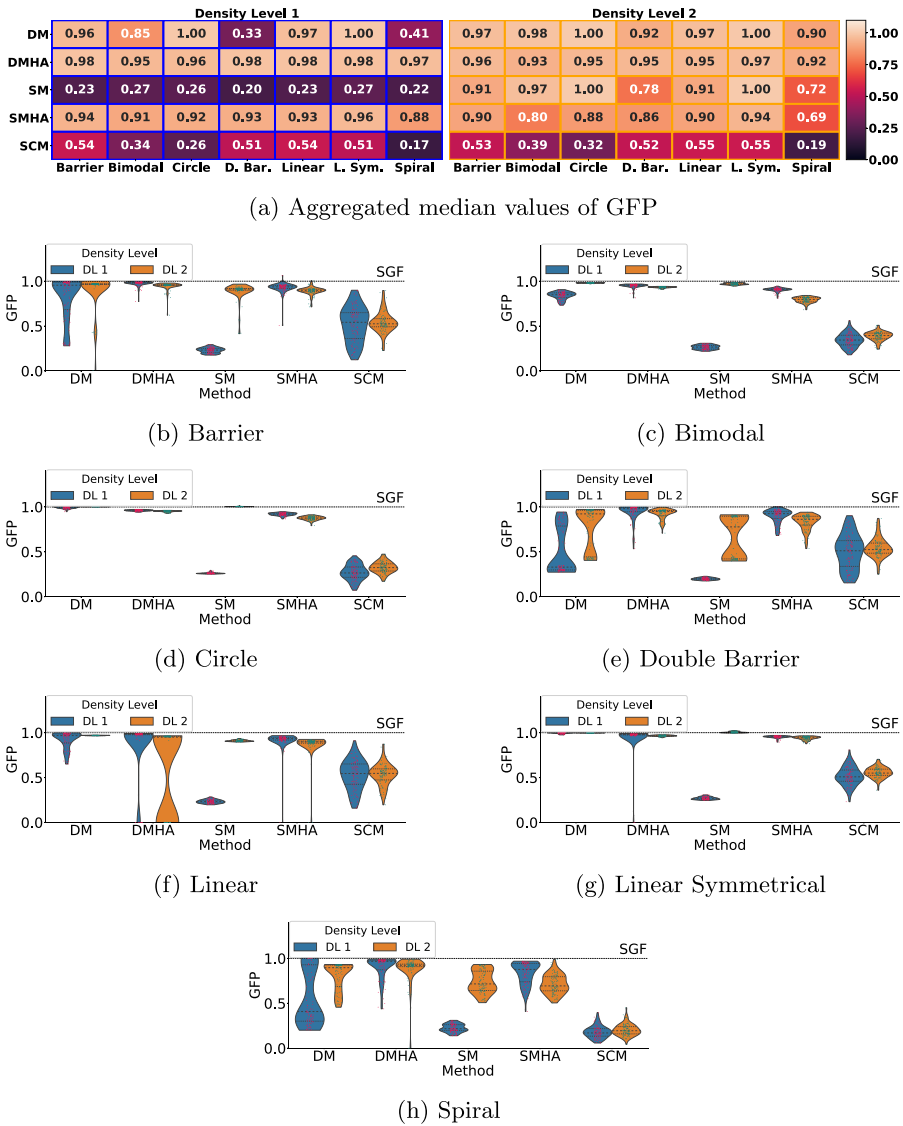


Fig. 4 Gradient following performances (GFP) of all methods in multi-agent simulations, on all scalar field models: Desired Distance Modulation (DM), its variation with alignment control (DMHA), Speed Modulation (SM), its variation with alignment control (SMHA) and Standard Collective Motion (SCM), the black dashed line at 1 indicates the best performance of Single Gradient Follower (SGF) and GFP of other methods are normalized accordingly, violins are presented for different density levels. Median GFP values of each method on each model is presented in **a**

DM numerous experiments resulted with high GFP, there is a considerable portion of runs demonstrating significant performance loss. Depending on the starting point, if swarm’s trajectory does not intersect with the "barrier" of the environment, DM shows more potential. This is the case since DM lacks the ability of a sudden turn as an ordered swarm, by reason of not having alignment control. The number of experiments

that resulted in poor performance are higher for the swarm density level of 1, indicating that DM performs slightly better in high swarm density level. In other words, it is possible that with low swarm density level, the desired distance difference among agents are not enough for repeatable success.

The situation is different when considering DMHA, in which the majority of the experimental runs achieved high GFP. Alignment control, which is the difference between DM and DMHA, plays a key role in explaining how controlling alignment of agents affects the ability of swarm to avoid sudden changes in the gradient trend. When SM is considered, the majority of the experiments demonstrated high GFP with swarm density level 2. The performance of SM is drastically different for density level of 1. The performance metric drops to a value even lower than SCM. The reason behind this sudden performance loss is that as stated earlier, an experiment is terminated if swarm centroid trajectory could not reach determined length in given time. The way we have designed the metric is the key here: The overall shortage of SM is greater since it loses the ability to move whereas SCM is always able to move although the direction of motion is not toward the most rewarding portions of scalar fields. Finally, SMHA shows a satisfactory performance with high GFP.

In Fig. 4c, which reports the results for Bimodal scalar field model, we see reduced variation in the performances among different methods compared to the Barrier model. A possible reason for that is the absence of an obstruction on the way of increasing scalar values of the gradient in Bimodal model. Apart from SM at swarm density level of 1, all methods indicate a high success. Nevertheless, DM has a slightly decreased performance at swarm density level of 1 when it is compared to 2. The same reason for that, which is stated for the Barrier model, also applies here. The performance of SMHA slightly decrease from swarm density level 1 to 2. This situation is the opposite of what is seen for DM. Although the difference is not significant, it indicates a correlation between low performance and high density for SMHA.

The results for the Circle model are reported in Fig. 4d. The performances reported here are very similar to the ones for Bimodal model, with a slight increase overall. This increase is better observable on median values at Fig. 4a. As a notable difference, DM's performance loss for swarm density level 1 does not exist for the Circle model. That absence of performance loss indicates a correlation between success of the swarm and scalar field model characteristics.

A slightly more challenging version of the Barrier model is the Double Barrier model. The results for this version which are presented in Fig. 4e clearly reflects the situation. For DM, portion of the experiments with poor performance is clearly growing. At swarm density level 1, this portion becomes the majority and drags the median line to a value lower than SCM. The fact that a swarm with DM performs worse than a method without a gradient perception, indicates that DM may cause swarm to spend more time and stuck between pitfalls if the opening is narrow while SCM still ignores the pitfalls and wander around aimlessly, including regions with higher values. Although the situation is not exactly the same for DMHA, the performance decrease is still observable.

The results for the Linear model, which is quite similar to the Barrier and Double Barrier models only with a difference on not having any barrier regions with undesirable values, is presented in Fig. 4f. For DMHA, although the median line is very close to 1 for swarm density level 1 and 2 within a close distribution, there are numerous data points valued as 0. As stated earlier, this indicates a group separation. This separation is observed more at swarm density level 2 when compared to 1. One possible reason of facing a separation on most successful method so far is continuous and directed nature of gradient in the Linear model. This nature allows the highly aligned swarm to speed up and collide with the

boundary edge. SM indicates a good performance with a narrow distribution, again only for swarm density level 2. SMHA also presents a satisfying performance on both density levels but with a small number of group separations, like DMHA.

Performances of different methods on the Linear Symmetrical model is presented at Fig. 4g. The first thing observed from a wide aspect is that almost all methods performed better than they do on the other linearly oriented scalar field models. DM and DMHA performed with a comparable success to the baseline method, apart from only 2 group separations out of total 100 experiments, observed with DMHA. The same trend of having better performance than other scalar field models can also be observed for SM and SMHA. The fact that maximum valued column is located in the center, leads to a sharper change in local values and produce a high-valued region in the middle to host the swarm far from boundary repulsion. These key features of the Linear Symmetrical model helps us to understand overall increase on the performances.

For the final and most distinguished scalar field model, the Spiral, the results are reported in Fig. 4h. As an overall trend, it is observed that all methods have decreased performances when compared to the other models. Firstly, DM has a satisfactory median line for swarm density level 2 around 0.9 while having a considerable amount of data point below it. This wide distribution exist also for swarm density level 1. Although the distribution of the performance metric is not narrow around the median line, DMHA still indicates that it is the best performing method among others in Spiral model. For SM and SMHA, the performances are distinctly better than SCM. Yet, the performances are comparably lower than they are at other scalar field models. When the distinct and narrow nature of the Spiral model is considered together with highest dark to bright region ratio, reasoning the overall performance decrease is not a challenge here.

4.4 Discussion of results

When we observe the results of multi-agent simulations (Fig. 4), we see that DM shows a satisfactory performance overall. The highest performances (Fig. 4a) are seen on the scalar models (Linear Symmetrical and Circle) where maximum is far from the borders and closer to the middle. These type of scalar fields naturally have values changing faster than the other fields. This case is in accordance with our expectations since larger local value difference leads to larger desired distance difference among agents that leads to a better performance. A different phenomenon is observed in the Bimodal and Linear models such that the performance decreases on lower swarm density levels. The performance drop can be connected to decreased variations in the scalar values measured by the individuals. In addition, DM shows significant performance drops when it encounters with pitfalls. This negative effect can be explained by the lack of the heading alignment component.

When alignment control is also employed, as in DMHA, we obtain better performances than DM. We also do not see any negative effect of the swarm density level on the performance. Yet, we observe group separations. Although we see only 2 in 100 runs on the Linear Symmetrical model, we see higher numbers on the Linear model. The Linear model have the maximum value along the edge. Since the edges have a repulsive effect, the swarm can possibly be hitting the edge like a fluid hitting a flat surface and scattering afterwards. In order to test this effect, we generated a scalar field model very similar to the Linear model with the only difference being the extended high-valued edge containing the maximum value. The performance comparison of DMHA on these scalar field models are shown in Fig. 5. We observe that number of group separations dropped to 0 with a slight

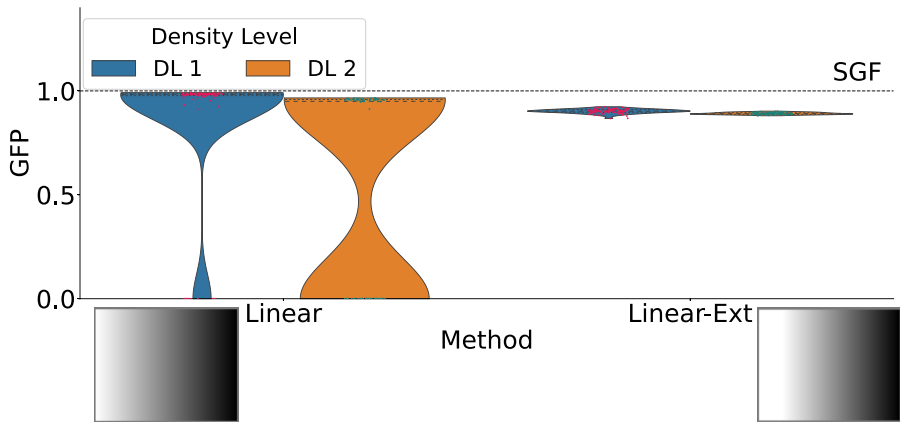


Fig. 5 The original and the extended version of Linear scalar field model and gradient following performance (GFP) of DMHA on both versions with different density levels

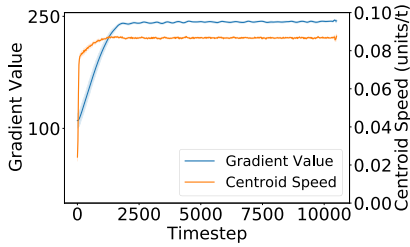
performance decrease on the extended Linear model. The reason of this decrease in GFP is not due to DMHA performing worse instead our baseline method SGF performing better, since brighter regions are extended. Extended bright region is more advantageous for a single robot when compared to a swarm. A single robot can easily stay in that region.

Another method we proposed for collective sensing in the swarm was SM. GFP results show that SM has a performance comparable with DM and DMHA (Fig. 4a). Observable drops of the performance occur in the models which involves pitfalls (Barrier and Double Barrier models) or more complex than others (Spiral model). This is expected from a swarm without alignment control. Yet, this conclusion is only valid with the high swarm density level (2).

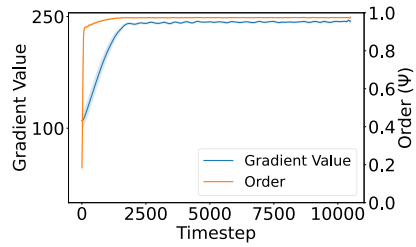
On the low swarm density level (1), experiments are terminated earlier with SM since the swarm was not able to travel long enough to meet the trajectory length criteria. That is when GFP is recorded as 0 for the rest of the "unrealized trajectory", which is observable on the violin plots in all scalar field models. Penalizing behaviors this way complies with our research goal since we require uninterrupted collective motion compared to intermittent swarm motion that can be interrupted, even if the interruption occurs in the desired region.

For a deeper understanding of this issue, Fig. 6a, c can be referred. Figures report the gradient value and speed of the swarm centroid for DMHA and SM methods on the Circle model at swarm density level 2 as an average of total 100 experiments. Level 2 is chosen since at that level, both methods are successful and results give information about method's specific differences. Firstly, two methods differ by their ascent to higher gradient value: While DMHA ascends to high gradient values sharply and maintain those values afterwards, SM shows a slow and gradual increase. This can be analyzed together with the corresponding centroid speeds, since DMHA quickly increases the speed and maintains it as well while SM increases the speed slightly at the beginning just before losing it for the rest of the experiment. Although the nature of SM suggests lower speed for swarm, the centroid speed is lower than what can be ideally desired.

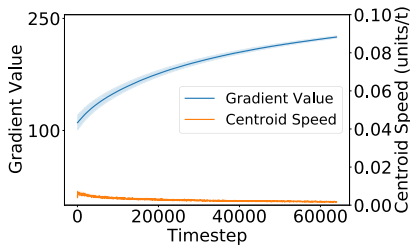
Analyzing the *order* metric (Ferrante et al., 2014), which is an important collective motion property, might explain the decreased speed performance observed with SM. The order metric is a quantitative value designating the agreement of swarm member's



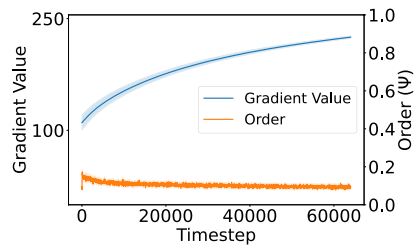
(a) DMHA, Gradient Value and Centroid Speed



(b) DMHA, Gradient Value and Order



(c) SM, Gradient Value and Centroid Speed



(d) SM, Gradient Value and Order

Fig. 6 Time evolution of swarm centroid speed, order and gradient value seen by swarm centroid of DMHA and SM from multi-agent simulations with 100 agents, on Circle scalar field model with swarm density level 2, colored lines show average of the variables over 100 experiments and colored bands show 95% confidence level

heading. It takes values closer to 1 when all members are moving in the same direction and approaches to 0 when member’s orientations are unrelated with each other. The order (Ψ) is calculated as follows where $e^{j\theta_i}$ denotes the heading angle of agent i :

$$\Psi = \frac{\| \sum_{i \in N} e^{j\theta_i} \|}{N} \tag{16}$$

Figure 6b, d report the order metric and centroid gradient value for DMHA and SM, respectively. The reason behind the difference between high and consistent speed of DMHA and decreased speed of SM can be found here. While DMHA reaches high order (as expected since it has alignment control) in a short time after the beginning, SM is never able to increase the order. That difference indicated that although a swarm with SM can move toward increasing values of gradient, it is impractically slow and chaotic. Moreover, as we see on the *gradient following performance* results for swarm density level 2, this slowness and disordered structure had to be stopped with a time-limit induced termination.

When SM is equipped with alignment control (SMHA), the performance improves significantly on the environment models with pitfalls. The performance difference with and without alignment control on these environment models is smaller with DM. In desired distance modulation methods, collective motion emerges from relative positions of the peers. Whereas in speed modulation methods, collective motion relies more on relative

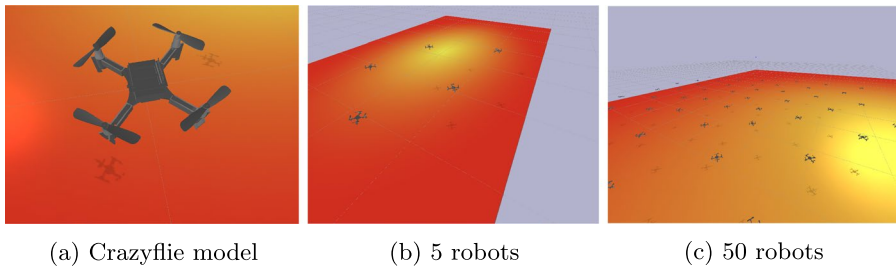


Fig. 7 Snapshots from dynamical simulation experiments with swarms of 5 and 50 robots on Circle scalar field model

speeds and headings. In other words, speed modulation is affected by a scalar variable such as the speed whereas distance modulation mainly depends on the vectorial adjustments (bearings of neighbor peers). This is a possible cause of the strong coupling between the performance of speed modulation and alignment control.

Moreover, in physical systems, obtaining better performance with lower accuracy on relative positions of entities is easier when compared to relative speeds or headings. The reason behind this is the fact that with relative positions, direction is the most important factor to the methods whereas for speed modulation, direction is a secondary product of changing relative speeds. Thus, we analyzed SMHA in physics-involved experiments to assess its real-world applicability and compare it with DM and DMHA in order to reveal the challenges in a method relying on accurate control of individuals' relative speeds. We have chosen SMHA for these experiments, instead of SM, since we can safely claim alignment control can bring improved order and this is the only possible way to get performances from speed modulation which are good enough to be compared with desired distance modulation methods'.

5 Dynamical simulations

Dynamical simulations are conducted on a physics-based simulator by employing realistic dynamic and sensory model of the Crazyflie nano aerial vehicle (Panerati et al., 2021). This particular simulator and aerial vehicle model are chosen since the real experiments are also conducted with the Crazyflie platform. Several snapshots taken during different experiment settings are shown in Fig. 7. Dynamical simulations are computationally costly when compared to multi-agent simulations. Hence, only a subset of the experiments conducted previously is considered here.

In dynamical simulations and real robot experiments, drones move in a fixed altitude and perform 2D flight. Although these aerial platforms have an advantage of moving in 3D, the proposed methods are not developed yet for 3D flight but still has the potential to do so. Instead of using the full potential of drones right away, in the first application of our methods, we aim to show the performance of the proposed methods in collective gradient following in a relatively simpler setting.

5.1 Experimental setup

In dynamical simulation setup, the platform model responds to changing values of linear speed commands at 20 Hz. Controlling a robot in a swarm with velocity commands would also enable porting of all the swarm robotics approaches normally developed on unmanned ground vehicles (UGVs) (Brambilla et al., 2013) to unmanned aerial vehicles. One further ingredient to achieve this is the ability to control the heading of a quadrotor, in a way that is compliant to the non-holonomic constraints typical of unmanned ground vehicles (Amorim et al., 2021).

Quadrotors are holonomic so that they can move in any direction in any time. Yet, if the velocity commands are generated in that manner, they may cause abrupt changes in the orientation of the quadrotor. Consequently, they might cause dynamical instabilities. Additionally, satisfying non-holonomic constraints is also required to implement SCM. SCM requires the platform to respond to linear speed commands, that move the platform in the direction of its current heading and respond to angular speed commands, that rotate the heading of the platform. To satisfy these non-holonomic constraints one option is to make the platform respond to angular speed commands through rotations of the quadrotor around the vertical axis (yaw), which effectively means to change its heading. This might bring new overhead and constraints to the controller that might cause further instabilities. For this reason, we choose to change the quadrotor heading only “virtually”, as follows: Initially, every robot has a randomly chosen virtual heading. Linear speed commands generate translation motion that are only applied in this direction, as if it is a differential drive ground vehicle. The angular speed commands, instead, modify this virtual heading, that will in turn change the axis along which we constrain the translating motion of the quadrotor.

The simulator does not model a sensor that can perform relative localization. It is also not straightforward to implement a controllable physical variable modeling the environmental gradient. Therefore, we emulate both the peer sensing and the scalar field model concepts in a way that is transparent to the proposed algorithms: Robots still can only sense peers within a certain sensing distance and only know the scalar value of the field at their particular location. In this way, the algorithms are compatible with any sensor for relative localization and any sensor for gradient sensing.

The computational complexity of the physics simulator imposes a time limitation for the experiments. Therefore, we performed experiments only within a subset of the experimental setup considered in multi-agent simulations, which excludes SM, since it shows significantly lower group order as a result of continuously changing linear speed values and needs significantly more time to move a considerable distance.

As another limitation imposed by higher computational complexity, we considered only three scalar field models: The Circle, Linear and Linear Symmetrical. They are chosen based on our judgement that they are fundamentally distinct models. Besides, we chose these models, because, we think that they are feasible to implement on our real flight arena. Additionally, we consider only two group sizes for dynamical simulations: 5 and 50. Experimenting with 5 agents allows us to make a smooth transition to the real robot experiments, since the number of aerial vehicles considered in real experiments is 5. Besides, 50 was our computational limit for physics-based simulations. In order to increase similarity of the simulation configuration with the real experiments, we chose the swarm density level equal to what we have in real robot

Table 4 Values of method specific parameters in dynamical simulations

Parameter	DM	DMHA	SMHA	SGF
α : Proximal cont. weight coeff.	1.0	1.0	0.8	–
β : Alignment cont. weight coeff.	–	2.0	2.0	–
K_1 : Linear speed gain	0.3	0.04	0.04	0.3
K_2 : Angular speed gain	0.6	0.1	0.3	0.2
ω_{\max} : Max. angular speed (rad/ t)	$\pi/4$	$\pi/2$	$\pi/3$	$\pi/4$
σ_{\min} : Max. desired distance coeff.	0.8	0.8	–	–
σ_{\max} : Min. desired distance coeff.	0.5	0.5	–	–
\mathcal{P}_u : Modified U_i portion	–	–	1.0	–
\mathcal{E}_u : Local val. correlation exp.	–	–	3.0	–
θ : Goal direction weight coeff.	–	–	–	5.0

experiments. Therefore, with 5 agents, dimensions were 6.5 m by 4 m and for 50 agents they were 16.12 m by 16.12 m.

Each experimental condition is repeated 50 times. Initial placement of the robots around initialization points are done in the same way with the multi-agent simulations. Initial centroid positions are chosen among the four corners and the center of the environment, where from each position 10 experiment runs are started. The termination criteria (the trajectory length) is specified as 2.5 times the longest edge length of the environment. None of the experiments needed and used a time limit.

The numerical values for the parameters are introduced in Sect. 3 and those related to the physics simulator settings can be found in Table 4. The chosen numerical values show a difference with multi-agent simulations, which is motivated by the fact that two robot models are completely different in terms of movement capabilities and sensing, which results in different collective dynamics. In order to reduce the gap between idealized agents and physics-based agents, parameter values are manually tuned. Parameter optimization is left out of scope of this paper.

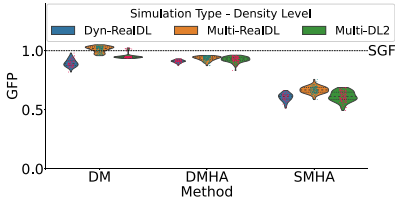
Gradient following performance (GFP) is used to measure the success of the swarm and recorded every 2 calculation steps. Hence, SGF is also implemented in the physics-based simulator and repeated 10 times more than other methods. In addition, the number of groups is checked during each experiment: No separation is ever observed, therefore the results of this metric are not shown. We also report the performances from multi-agent simulations, in order to perform a comparison. For 5 and 50 agents, multi-agent simulations are conducted with 2 different density levels, one reflects real experiments and the other one the multi-agent simulations.

5.2 Results

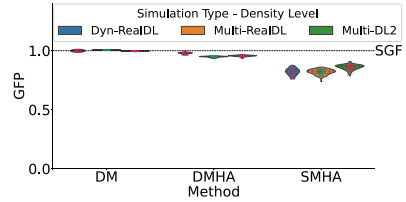
In Fig. 8, we report GFP from dynamical simulations together with the multi-agent simulation counterparts. When we check the plots, we conclude that the results of multi-agent simulations are well replicated on dynamical simulations with various methods, scalar field models and densities. This case also can be quantitatively validated in Figs. 4a and 8a.

	Dynamical Sim. - Real Density Lvl.			Multi-Agent Sim. - Real Density Lvl.			Multi-Agent Sim. - Density Lvl. 2			
DM-5	0.89	0.87	0.94	1.02	0.94	0.99	0.95	0.96	0.99	1.00 0.75 0.50 0.25 0.00
DM-50	1.00	0.91	1.00	1.01	0.95	1.00	1.00	0.97	0.99	
DMHA-5	0.91	0.90	0.93	0.94	0.91	0.94	0.93	0.94	0.94	
DMHA-50	0.98	0.90	0.98	0.95	0.93	0.96	0.96	0.95	0.97	
SMHA-5	0.61	0.63	0.80	0.67	0.73	0.87	0.61	0.82	0.85	
SMHA-50	0.82	0.85	0.97	0.82	0.83	0.93	0.86	0.90	0.93	
	Circle	Linear	Lin. Sym.	Circle	Linear	Lin. Sym.	Circle	Linear	Lin. Sym.	

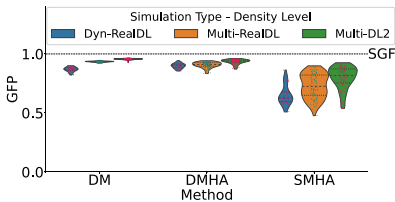
(a) Aggregated median GFP values



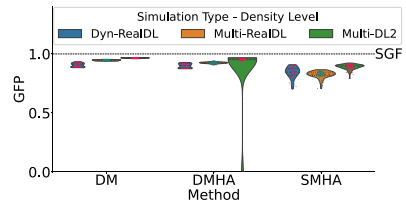
(b) Circle - 5 robots



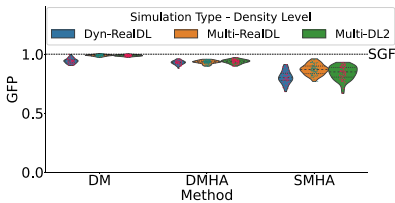
(c) Circle - 50 robots



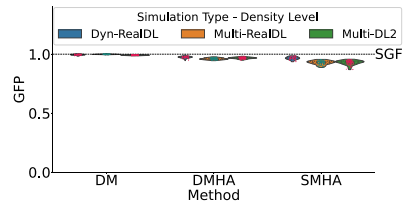
(d) Linear - 5 robots



(e) Linear - 50 robots



(f) Linear Symmetrical - 5 robots



(g) Linear Symmetrical - 50 robots

Fig. 8 Gradient following performances (GFP) of chosen methods (DM,DMHA and SMHA) in dynamical simulations, on Circle, Linear and Linear Symmetrical scalar field models with 5 and 50 robots, the black dashedline at 1 indicates the best performance of Single Gradient Follower (SGF) and GFP of other methods are normalized accordingly, violins are presented for different density levels and simulation mediums: Dynamical simulations with real robot experiments density level (Dyn-RealDL), multi-agent simulations with real robot experiments density level (Multi-RealDL) and multi-agent simulations with swarm density level 2 (Multi-DL2), median GFP values for each setting can be observed in **a**

When we observe the performances on the Circle model with 5 agents (Fig. 8b), we see an apparent GFP difference between SMHA, and the other methods. Although, finite-size effects greatly affected SMHA, this effect is less obvious with DM and DMHA since in these methods, unmatched desired distance values among peers are already producing repulsive forces for the focal agents in a direction outwards from the rest of the swarm pointing out the increasing gradient. Another point to note for this experiment setting is that DM can perform comparably better than SGF. It is only possible with low number

of agents and in multi-agent simulations. These conditions point out that when agents are moving similar to ideal case, a small swarm is better to keep their centroid at the bright and pointy center of the gradient circle than a single agent trying to wander around that bright center point. Better performance of DM over SGF is almost vanished for the same Circle model when experiments are conducted with 50 agents, as can be seen in Fig. 8c. Another difference we observed with 50 agents is the improvement of SMHA's performance.

In Linear model, (Fig. 8d, e), we see several differences from the Circle model. Due to slower and smoother changing scalar values of the Linear model, DM no longer crosses the baseline. Besides, DMHA points out several group separations with 50 agents in multi-agent simulations. Meanwhile SMHA shows a slightly better but less repeatable performance when compared to the Circle model. Although the increased median values can be explained by the larger portion of the environment with higher values for this case, the wider spread points out an inconsistency on following an increasing trend.

In Linear Symmetrical model (Fig. 8f, g), all methods demonstrate a slight improvement on their performances when compared to the other models. This can be linked to the sharper transition of values in the Linear Symmetrical model and more importantly, wider region of field with higher values to settle and keep the swarm centroid there, unlike the boundary effect the swarm is facing on the Linear model or smaller bright region in the Circle model. In addition, SMHA shows its best performance with 5 agents in this scalar field model.

5.3 Discussion of results

Physics-based simulations stand as a meeting point for three different experiment mediums. Using a realistic model for the aerial vehicles brings differences on relative and absolute performances of the methods.

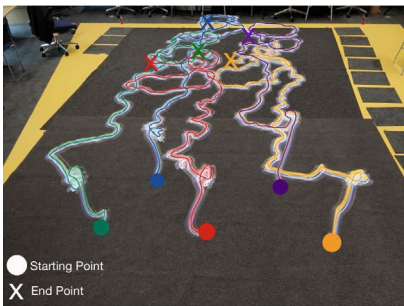
As an important outcome, performance of DM and DMHA methods stand in a strong agreement with multi-agents simulation results (Figs. 4a, 8a). This creates a strong motivation and confidence to carry these methods to real robot experiments. Although the case is similar in SMHA with 50 agents, the decreased performance with 5 agents indicates the weakness of SMHA on scalability. When the number of peers in a locality decreases, the robots that are facing higher gradient values can not become effective enough in group level to influence the others. There might be solutions to increase the effect of slowed down peers on others such as increasing the speed gap between peers that are measuring different gradient values or strengthening the effect of peers on each other. Yet, based on our knowledge and experience we expect that both possible solutions end up with losing collective motion stability. As a result of weakness of SMHA with low robot numbers in addition to our trials, we did not included this method in real robot experiments. To further motivate this exclusion, another reason is the fact that robots with SMHA need to be successful on changing their speed accurately and frequently. Yet, it is a big challenge for a robotic hardware, especially aerial vehicles on the air. That is why we believe, this method was not applicable with our real-world experimental setup.



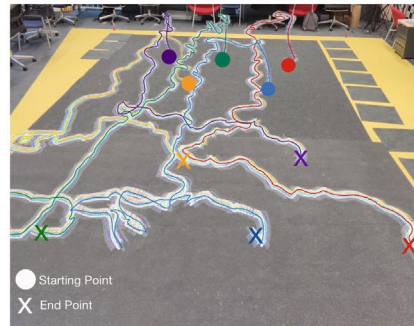
(a) Top view



(b) Bottom view



(c) Individual trajectories on Circle scalar field model



(d) Individual trajectories on Linear scalar field model

Fig. 9 Crazyflie aerial platform has a UWB deck on top **a** and combination of flow camera and laser distance sensor on bottom **b**. Trajectories in **c** and **d** are colored for each robot in the swarm. **c** in Circle scalar field model shows circulations around the bright center while **d** shows straighter paths from darker to brighter regions of Linear model

6 Real robot experiments

Final stage of the experiments is conducted with a real nano-aerial vehicle platform, the Crazyflie.¹ This platform is selected because of its small size (9.2 cm motor to motor distance) and capabilities for indoor flights. Experiments conducted with a real aerial platform constitutes a strong test bed for applicability of proposed methods with physical constraints, inaccuracies and delays.

6.1 Experimental setup

Crazyflie is able to fly in a stable manner without the need for global position information, yet still there is no fully developed option for accurate on-board peer localization. Therefore, as commonly done for this platform, we use an indoor positioning setup: The Loco Positioning System (LPS). This system works based on Ultra-Wide Band (UWB) signals

¹ <https://www.bitcraze.io/products/crazyflie-2-1/>.

exchanged between fixed anchors, which are UWB emitters, placed in the arena at positions shared with Crazyflies and UWB receivers installed on the Crazyfly board. By using LPS, the drones can localize themselves within a 10 cm accuracy. Estimations for position on horizontal and vertical axes are done on board. Crazyfly platform equipped with a UWB deck on top and a flow deck on bottom, which consists of a flow camera for velocity measurements and laser distance sensor for altitude measurements, is shown in Fig. 9. Although peer localization is not performed fully on-board, UWB applications on Crazyfly proceeds in a promising direction for a possible usage of fully on-board localization in the near future. Yet another limitation we have is not being able to process high-level control fully on board. To be precise, although in principle it is possible to implement a controller for our methods on board, the ecosystem was not developed enough by the time we conducted our experiments. Thus, our Crazyfly setup utilizes a central computer in the control loop: High-level controllers are running on the central computer that receives instantaneous positions of all robots and calculates velocity command for each drone at every time step. These commands are then send to each drone via a serial communication protocol using the Crazyradio platform. For the high level controller and corresponding communication requirements, Crazyswarm (Preiss et al., 2017) framework is employed. The peer localization and sensing of local values of the scalar field model are also implemented on the central computer following the constraints of the sensing ranges. In sensing the local values, the absolute position of each drone is used to find which cell it lies within and the value of that cell with a random uniform error in range of $[-e_g, e_g]$.

The flight arena dimensions are chosen to be 6.5 meters by 4 meters and total number of robots in the swarm is 5. Long exposure photographs of the swarm on the flight arena captured during experiments are shown in Fig. 9. Considering the limited dimensions of the Crazyfly and its battery capacity, the flight time is limited at 4.5 minutes. The same scalar field models used in dynamical simulations are chosen for experiments: Circle, Linear and Linear Symmetrical. For real robot experiments, the DM and the DMHA methods are used.

For each method and the scalar field model, 6 experiments are conducted. As in multi-agent and physics-based simulations, drones are initialized and take off within their sensing range and with the centroid located within a proximity of chosen initial point. Initial points are selected to be on each of the four corners of the arena in addition to upper and lower points of a vertical line drawn through the middle of arena's long edge. A slight difference from the physics-based simulations is that instead of using center point, two points located above and below center point are used as initial points. Since with real platforms repeating an experiment is more costly, we decided that analyzing group motion with initial conditions closer to lower gradient values is more important compared to the case where they start already closer to higher gradient values. The parameter values for the methods are chosen to be the same as in the dynamical simulations. This is the supporting evidence to the fact that physics-based simulations are realistic enough to model and predict the outcome of the real robots.

The success of the aerial swarm on following the increasing gradient in the environment is assessed through the inspection of the centroid trajectory of the swarm reported in Fig. 10 and of the gradient value that swarm centroid is experiencing as plotted in Fig. 11. In this plot, the centroid gradient values are averaged over all 6 experiments performed with a given method and a given scalar field model. The positions of the centroid, and the local value of the gradient corresponding to the centroid location are recorded at 2 Hz.

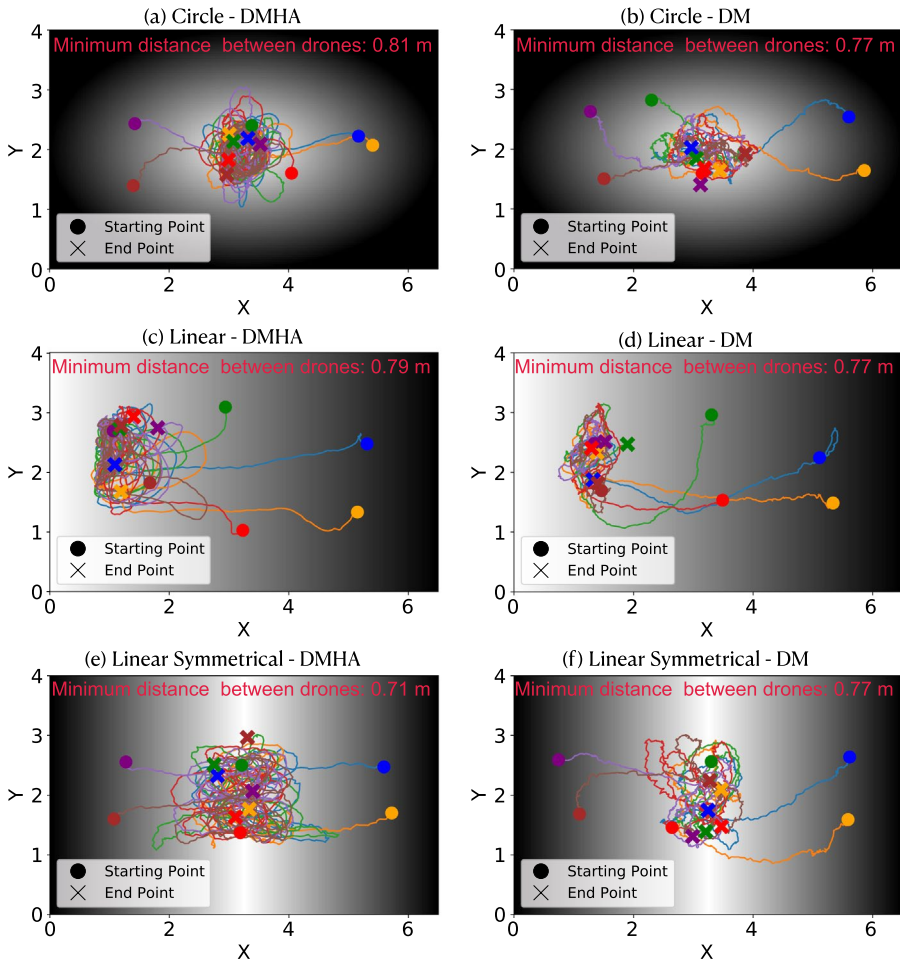


Fig. 10 Trajectories of swarm centroids from real robot experiments for DM and DMHA methods, on Circle, Linear and Linear Symmetrical scalar field models, different colors indicate different experiments, filled circle indicates start of the trajectory and filled crosses indicates the endpoint for corresponding color

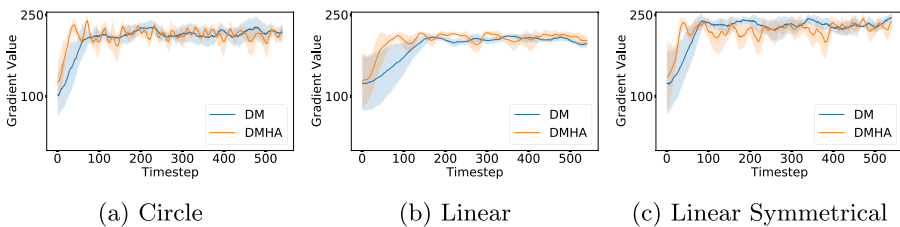


Fig. 11 Development of gradient values over time seen by swarm centroids at real-robot experiments for DM and DMHA at Circle, Linear and Linear Symmetrical scalar field models

6.2 Results

When 6 different trajectories of each method are observed on the Circle model (Fig. 10a, b), we see an increased concentration and overlapping ellipses around the center for DMHA which indicates an effective collective behavior. The reason why we consider these trajectories as effective is the behavior of the swarm in the high-value gradient region is not stationary: The swarm continues to revolve in a collective ordered manner centered around the high value region of the gradient. As stated earlier, a stationary swarm located on the bright region is not the ideal case, because we ideally want a swarm that does not lose the order when reaching the end point of the gradient, as in a real task there could be a subsequent gradient to follow and gaining back the order might take some time. When trajectories for DM is closely observed, although they seem similar at high level, the ragged structure of the trajectories and the non-elliptic tours around the center indicates a less ordered and directed motion. Yet, the greatest advantage of DM is being able to achieve gradient-following results without alignment control, which indicates the presence of a trade-off between swarm capabilities and performance.

In the Linear model (Fig. 10c, d) and the Linear Symmetrical model (Fig. 10e, f), the trajectories for both DMHA and DM are concentrated on the left (Linear model) or center (Linear Symmetrical model) of the arena where higher local values of the gradient are located. In addition, here the positive contribution of alignment control can also be observed, as in the Circle model. Trajectories of DMHA are much direct and short when compared to DM. Yet, DMHA has more circular paths spreading around a wider area while DM shows a more stable concentration around the region.

Figure 10 also reports the minimum distance observed between drones in all 6 experiments. In both methods, this distance is not significantly different from each other. The values are in the expected range (when σ_{\min} and σ_{\max} are considered in Table 4). Moreover, similarity between minimum distances of methods is also expected since both use the same values for corresponding parameters. Our observations during experiments also agree with the data: Minimum distances were as expected and safe for drones.

Figure 11a–c present the mean and confidence interval of the centroid gradient value over time for the chosen scalar field models, as averaged over experiment repetitions. In all models, we see the gradient values reach up to over 200. In all models but more apparently in the Circle and Linear Symmetrical models, DM has a more stable mean line while we can see oscillations for DMHA. In addition, DMHA increases to its marginally stable form quicker than DM. Whereas the speed of DMHA on reaching its maximum value is apparently higher in the Linear model, as expected since the distance from darker to brighter regions is longer than others for the Linear model. This may be due to the fact that, in a more stretched environment, the swarm keeps an ordered regime longer and therefore moves faster, on average on the whole trajectory. Finally, in only Linear Symmetrical model we can see that the mean DM line is above the DMHA line almost all the time, since more stationary attitude of DM is most important in this scalar field model. In addition, the value that mean lines reach for both methods are higher than they are in the Circle and Linear models. The observation of both methods showing relatively better performance on Linear Symmetrical model is in accordance with the implications done before from multi-agent and dynamical simulation results.

6.3 Discussion of results

Overall, with both methods the swarm shows motion from lower gradient values to higher ones in all scalar field models. Nevertheless, DMHA and DM have their strengths and weaknesses in different points. While DMHA is better on reaching the bright side with a shorter path and staying active there, DM is better on centering the highest value and stay stationary after. One could suggest that an adaptive usage of alignment control might serve as an optimal strategy to advantage the strengths of both methods. Yet, we want to test the isolated effect of heading control on the methods since our concern is avoiding heading control completely. Since our definition of virtual heading on the drones is not suitable for on-board sensing, alignment control certainly requires communication. We believe that a part of our novelty relies on eliminating any need of information exchange between peers on their scalar perceptions of the environment. Hence, exchanging heading information will violate this novelty. For that reason, involving heading information in any portion is something we try to avoid.

Since the velocity constraints, desired peer distances and peer sensing ranges are chosen according to our real-world hardware setup from the beginning (e.g., even for multi-agent simulations), we were able to test our methods with minimal modification. The method parameters of physics-based simulations and real robot experiments were exactly the same. We see it as an advantage for both mediums of experiments: We both increase our confidence on the physics-based simulations and on the possibility to increase robot number in real robot experiments since we already increased it up to 50 in physics-based simulations.

Even with the current inaccuracy level on robots moving with velocities of maximum 15 cm/s, we were able to produce desired collective behaviors. Moreover, as it can also be seen in Fig. 11, the success of DM was comparable to DMHA. We see it as an important outcome since we are able to state that desired distance modulation works without the need of alignment control. In a truly distributed application, this will avoid the implementation of a local communication modality since the distance and bearing of other agents can be sensed on-board in various ways. For example visually or by using specialized sensors. We believe being freed of this need can drastically increase the method's real-life applicability.

7 Conclusion

In this paper, we proposed two different methods—desired distance modulation and speed modulation—for collective gradient following with a swarm of robots. Differently from the literature, proposed methods achieve collective gradient sensing without the requirement of any information exchange or multiple sensors on board to estimate the gradient. Through systematic experiments in different gradient models and densities, we showed that both methods with and without alignment control successfully performed in different environment settings. Success of the proposed methods proves that collective motion and emergent sensing can be produced simultaneously when agents have a real aerial platform dynamics. To the best of our knowledge, it is the first time that collective gradient sensing is achieved with agents having such limited sensing capabilities. Our experimental analysis leveraged on three different experimental setups: (1) a multi-agent kinematic-based simulator aimed at performing an analysis of all the proposed methods in a rich set of environmental conditions to understand the inherent dynamics, (2) a physics-based simulator aimed at

performing real-world like experiments in a systematic way; (3) real quadrotor platforms aimed at validating the methods in a real-world setting. In the experiments, both gradient following performance and cohesiveness of the swarm are evaluated.

We can draw a conclusion from the results that the desired distance modulation method is performing successfully among different gradient models and densities in both simulations and real-world experiments. Alignment control enhances the performance and adds more robustness but it is not required when the desired distance modulation method is employed. A swarm with speed modulation is much more sensitive to individuals' ability on accurate speed control and develops a dependency on alignment control. Without alignment control, the group order is easily lost and collective motion ability of the swarm is deteriorated as observed in dynamical simulations. Yet, simulations with point-mass agents still show a potential with speed modulation: Accurate speed control with realistic/real agents or a future development on the method to make it more robust to noise in speed control could make speed modulation usable real-world applications.

Although we only tested the ability of the swarm on following a stationary gradient in the environment, proposed methods are fully compatible with dynamically changing gradient models. Since we always care about having an ordered and moving swarm, testing the performances of proposed methods with such gradient models is a highly possible future challenge. A complementary development will be further enhancing our real platform to sense a physical property such as gas particles, temperature or light. By doing so, we will be highlighting the importance and portability of our methods on real-world problems. Such real-world applications with an aerial swarm can be searching radioactive sources outdoor, gas leaks indoor or localizing any physical property source which can be suitable for our gradient modeling approach. Another future aspect we have is interpreting the gradient differently. A good example is replacing the scalar measurements of individuals with a danger indicator depending on a "predator" and use the collective sensing ability to avoid it. The opposite is also possible, track and chase a "prey" with a similar approach. This draws another exciting integration with a real-life problem.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

- Amorim, T., Nascimento, T., Petracek, P., De Masi, G., Ferrante, E., & Saska, M. (2021). Self-organized uav flocking based on proximal control. In *2021 international conference on unmanned aircraft systems (ICUAS)* (pp. 1374–1382). IEEE. <https://doi.org/10.1109/ICUAS51884.2021.9476847>
- Berdahl, A., Torney, C. J., Ioannou, C. C., Faria, J. J., & Couzin, I. D. (2013). Emergent sensing of complex environments by mobile animal groups. *Science*, *339*(6119), 574–576. <https://doi.org/10.1126/science.1225883>
- Berdahl, A. M., Kao, A. B., Flack, A., Westley, P. A., Codling, E. A., Couzin, I. D., et al. (2018). Collective animal navigation and migratory culture: From theoretical models to empirical evidence.

- Philosophical Transactions of the Royal Society B: Biological Sciences*, 373(1746), 20170009. <https://doi.org/10.1098/rstb.2017.0009>.
- Bjerknes, J. D., Winfield, A. F., & Melhuish, C. (2007). An analysis of emergent taxis in a wireless connected swarm of mobile robots. In *2007 IEEE Swarm intelligence symposium* (pp. 45–52). IEEE. <https://doi.org/10.1109/SIS.2007.368025>
- Brambilla, M., Ferrante, E., Birattari, M., & Dorigo, M. (2013). Swarm robotics: A review from the swarm engineering perspective. *Swarm Intelligence*, 7(1), 1–41. <https://doi.org/10.1007/s11721-012-0075-2>
- Camley, B. A., Zimmermann, J., Levine, H., & Rappel, W. J. (2016). Emergent collective chemotaxis without single-cell gradient sensing. *Physical Review Letters*, 116(9), 098101. <https://doi.org/10.1103/PhysRevLett.116.098101>
- Campo, A., Garnier, S., Dédriche, O., Zekkri, M., & Dorigo, M. (2011). Self-organized discrimination of resources. *PLoS One*, 6(5), e19888. <https://doi.org/10.1371/journal.pone.0019888>
- Carreón, G., Gershenson, C., & Pineda, L. A. (2017). Improving public transportation systems with self-organization: A headway-based model and regulation of passenger alighting and boarding. *PLoS one*, 12(12), e0190100. <https://doi.org/10.1371/journal.pone.0190100>
- Coppola, M., McGuire, K. N., De Wagter, C., & De Croon, G. C. (2020). A survey on swarming with micro air vehicles: Fundamental challenges and constraints. *Frontiers in Robotics and AI*, 7, 18. <https://doi.org/10.3389/frobt.2020.00018>
- Couzin, I. D., Krause, J., Franks, N. R., & Levin, S. A. (2005). Effective leadership and decision-making in animal groups on the move. *Nature*, 433(7025), 513–516. <https://doi.org/10.1038/nature03236>
- Duisterhof, B. P., Li, S., Burgués, J., Reddi, V. J., & de Croon, G. C. (2021). Sniffy bug: A fully autonomous swarm of gas-seeking nano quadcopters in cluttered environments. In *2021 IEEE/RSJ international conference on intelligent robots and systems (IROS)* (pp. 9099–9106). IEEE. <https://doi.org/10.1109/IROS51168.2021.9636217>
- Ferrante, E., Turgut, A. E., Huepe, C., Stranieri, A., Pinciroli, C., & Dorigo, M. (2012). Self-organized flocking with a mobile robot swarm: A novel motion control method. *Adaptive Behavior*, 20(6), 460–477. <https://doi.org/10.1177/1059712312462248>
- Ferrante, E., Turgut, A. E., Stranieri, A., Pinciroli, C., Birattari, M., & Dorigo, M. (2014). A self-adaptive communication strategy for flocking in stationary and non-stationary environments. *Natural Computing*, 13(2), 225–245. <https://doi.org/10.1007/s11047-013-9390-9>
- Gorban, A. N., & Çabukoğlu, N. (2018). Basic model of purposeful kinesis. *Ecological Complexity*, 33, 75–83. <https://doi.org/10.1016/j.ecocom.2018.01.002>
- Kearns, D. B. (2010). A field guide to bacterial swarming motility. *Nature Reviews Microbiology*, 8(9), 634–644. <https://doi.org/10.1038/nrmicro2405>
- Kernbach, S., Thenius, R., Kernbach, O., & Schmickl, T. (2009). Re-embodiment of honeybee aggregation behavior in an artificial micro-robotic system. *Adaptive Behavior*, 17(3), 237–259. <https://doi.org/10.1177/1059712309104966>
- Khalidi, B., & Cherif, F. (2016). A virtual viscoelastic based aggregation model for self-organization of swarm robots system. In *Annual conference towards autonomous robotic systems* (pp. 202–213). Springer. https://doi.org/10.1007/978-3-319-40379-3_21
- Olson, R. S., Hintze, A., Dyer, F. C., Knoester, D. B., & Adami, C. (2013). Predator confusion is sufficient to evolve swarming behaviour. *Journal of The Royal Society Interface*, 10(85), 20130305. <https://doi.org/10.1098/rsif.2013.0305>
- Panerati, J., Zheng, H., Zhou, S., Xu, J., Prorok, A., & Schoellig, A. P. (2021). Learning to fly—a gym environment with pybullet physics for reinforcement learning of multi-agent quadcopter control. In *2021 IEEE/RSJ international conference on intelligent robots and systems (IROS)* (pp. 7512–7519). <https://doi.org/10.1109/IROS51168.2021.9635857>
- Preiss*, J. A., Hönig*, W., Sukhatme, G. S., & Ayanian, N. (2017). CrazySwarm: A large nano-quadcopter swarm. In *IEEE international conference on robotics and automation (ICRA)* (pp. 3299–3304). IEEE. <https://doi.org/10.1109/ICRA.2017.7989376>, software available at <https://github.com/USC-ACTLab/crazyswarm>
- Puckett, J. G., Pokhrel, A. R., & Giannini, J. A. (2018). Collective gradient sensing in fish schools. *Scientific Reports*, 8(1), 1–11. <https://doi.org/10.1038/s41598-018-26037-9>
- Şahin, E. (2004). Swarm robotics: From sources of inspiration to domains of application. In *International workshop on swarm robotics* (pp. 10–20). Springer. https://doi.org/10.1007/978-3-540-30552-1_2
- Schmickl, T., Wotawa, F., Thenius, R., & Varughese, J. C. (2016). Fstaxis algorithm: Bio-inspired emergent gradient taxis. In *ALIFE 2016, the fifteenth international conference on the synthesis and simulation of living systems* (pp. 330–337). MIT Press. <https://doi.org/10.7551/978-0-262-33936-0-ch055>

- Shaukat, M., & Chitre, M. (2016). Adaptive behaviors in multi-agent source localization using passive sensing. *Adaptive Behavior*, 24(6), 446–463. <https://doi.org/10.1177/1059712316664120>
- Shklarsh, A., Ariel, G., Schneidman, E., & Ben-Jacob, E. (2011). Smart swarms of bacteria-inspired agents with performance adaptable interactions. *PLoS Computational Biology*, 7(9), e1002177. <https://doi.org/10.1371/journal.pcbi.1002177>
- Turgut, A. E., Çelikkanat, H., Gökçe, F., & Şahin, E. (2008). Self-organized flocking in mobile robot swarms. *Swarm Intelligence*, 2(2), 97–120. <https://doi.org/10.1007/s11721-008-0016-2>
- Valentini, G., Brambilla, D., Hamann, H., & Dorigo, M. (2016). Collective perception of environmental features in a robot swarm. In *International conference on swarm intelligence* (pp. 65–76). Springer. https://doi.org/10.1007/978-3-319-44427-7_6
- Varughese, J. C., Moser, D., Thenius, R., Wotawa, F., & Schmickl, T. (2019). swarmfstaxis: Borrowing a swarm communication mechanism from fireflies and slime mold. In *Complex adaptive systems* (pp. 213–222). Springer. https://doi.org/10.1007/978-3-030-20309-2_10
- Wahby, M., Petzold, J., Eschke, C., Schmickl, T., & Hamann, H. (2019). Collective change detection: Adaptivity to dynamic swarm densities and light conditions in robot swarms. In *Artificial life conference proceedings* (pp. 642–649). MIT Press. https://doi.org/10.1162/isal_a_00233
- Zahadat, P., & Schmickl, T. (2016). Division of labor in a swarm of autonomous underwater robots by improved partitioning social inhibition. *Adaptive Behavior*, 24(2), 87–101. <https://doi.org/10.1177/1059712316633028>

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.