From Shadows to Light: A Swarm Robotics Approach With Onboard Control for Seeking Dynamic Sources in Constrained Environments

Tugay Alperen Karagüzel, Victor Retamal, Nicolas Cambier, and Eliseo Ferrante

***Abstract*—In this letter, we present a swarm robotics control and coordination approach that can be used for locating a moving target or source in a GNSS-denied indoor setting. The approach is completely onboard and can be deployed on nano-drones such as the Crazyflies. The swarm acts on a simple set of rules to identify and trail a dynamically changing source gradient. To validate the effectiveness of our approach, we conduct experiments to detect the maxima of the dynamic gradient, which was implemented with a set of lights turned on and off with a time-varying pattern. Additionally, we introduce also a minimalistic fully onboard obstacle avoidance method, and assess the flexibility of our method by introducing an obstacle into the environment. The strategies rely on local interactions among UAVs, and the sensing of the source happens only at the individual level and is scalar, making it a viable option for UAVs with limited capabilities. Our method is adaptable to other swarm platforms with only minor parameter adjustments. Our findings demonstrate the potential of this approach as a flexible solution to tackle such tasks in constrained GNSS-denied indoor environments successfully.**

***Index Terms*—Swarm robotics, aerial systems: perception and autonomy, multi-robot systems.**

I. Introduction

UNMANNED aerial vehicles (UAVs) have proven to be a valuable tool for source seeking in a wide range of applications [1], [2]. However, detecting a source with UAVs poses significant challenges [3], particularly in complex environments. Indeed, such endeavour requires high levels of autonomy and adaptability, particularly in unknown GNSS-denied environments where communication may be limited or intermittent. Source seeking UAVs can find applications in outdoor or indoor environments. UAVs for outdoor environments, larger in size, enjoy more flexibility in sensor hardware and computational resources, allowing resource-intensive operations. Conversely, UAVs for indoor environments face space constraints and necessitate a compact design with limited sensor and computational capacities. Because of these restrictions, the design of hardware and control for indoor UAVs is more challenging compared to outdoor UAVs. This mandates strategic efficiency enhancements, particularly in algorithm design and resource utilization, to meet objectives within these spatial confines.

To circumvent the constraints inherent to indoor platforms and to achieve the above objectives, an approach based on multiple UAVs offers a compelling solution, as evidenced by previous research [4]. Particularly, swarm robotics studies collective behaviors that emerge from simple local interactions between individual agents [5]. Given that a swarm is a group of decentralized, autonomous, and simple agents, that can work together to achieve a common goal, it presents an apt framework for our constraints. Swarm robotics takes inspiration from the collective behaviours observed in nature: The collective sensing behaviours in fish schools, as explored in [6], [7], offer substantial inspiration for our previous [8] and the current study. Thus, drawing upon such insights from nature allows us to apply them beneficially within the context of our research.

The task of source seeking becomes even more difficult in the presence of dynamic or moving sources, and when requiring also collective motion towards the source, that is individuals maintaining safe and similar distances to each other while moving in the same direction [9]. Here, UAVs must coordinate their collective motion, sense and navigate the environment, detect and locate the source of interest, while constantly balancing a compromise between updating their collective estimation of the source and coordinating their movements.

To solve the problem of performing collective motion while seeking a dynamic source in a GNSS-denied indoor environment, this study introduces an approach for fully onboard control of a nano UAV swarm. The approach is not based on complicated control and state estimation methods. Rather, the swarm utilizes relative localization, limited onboard sensing and no communication on source measurements, and achieves dynamic source seeking through emergent self-organization. In this study we show, for the first time, a **fully onboard** implementation, with a **genuine and dynamically changing gradient source** (in the experiments implemented via light emitters), **sensed onboard** in the presence of **obstacles**.

Manuscript received 18 July 2023; accepted 19 October 2023. Date of publication 10 November 2023; date of current version 21 November 2023. This letter was recommended for publication by Associate Editor G. A. Sartoretti and Editor M. A. Hsieh upon evaluation of the reviewers' comments. *(Corresponding author: Tugay Alperen Karagüzel.)*

Tugay Alperen Karagüzel, Victor Retamal, and Nicolas Cambier are with the Vrije Universiteit Amsterdam, 1082MA Amsterdam, The Netherlands (e-mail: t.a.karaguzel@vu.nl; v.retamalguiberteau@student.vu.nl; n.p.a.cambier@vu.nl).

Eliseo Ferrante is with the Vrije Universiteit Amsterdam, 1082MA Amsterdam, The Netherlands, and also with the Autonomous Robotics Research Center, Technology Innovation Institute Abu Dhabi, Abu Dhabi, United Arab Emirates (e-mail: e.ferrante@vu.nl).

Digital Object Identifier 10.1109/LRA.2023.3331897

II. Related Work

Previously, several works have attempted to tackle the problem of localizing a source by using UAVs with limited sensing capabilities. In the context of single UAVs, the authors in [10] employed a deep reinforcement learning algorithm to locate a solitary, stationary light source, utilizing multiple lightweight sensors on the UAV. Their platform was also capable of obstacle avoidance. The learning pipeline incorporated a training-in-simulation phase prior to actual deployment. The authors in [11] introduced a bio-inspired solution, mimicking bacterial chemotaxis, to a source localization problem, but without employing any learning methodology. They deployed a single nano-UAV equipped with a temperature sensor to identify a heat source. The UAV leveraged historical position and measurement data to estimate an assumed form of gradient, successfully localizing the heat source.

Besides single-UAV applications, there is also work on multi-UAV approaches. The authors of [12] focus on an extensive application employs nano-drones to localize a stationary gas leak source using a specialized sensor. In this application, UAVs exchange measurement data and approximate relative distances to pinpoint the source, all while circumventing obstacles. Despite these coordinated actions, the system does not demonstrate any collective motion. The authors of [13] introduce a level of collective movement where nano UAVs localize an emulated source, implemented as a stored function onboard each UAV. The computed source function values are shared among UAVs, subsequently integrating with the flocking algorithm. In a simulation study by [14], multiple UAVs attempt to locate a single static signal source utilizing an extremum seeking algorithm. The UAVs estimate the signal strength gradient's peak by averaging each other's measurements. Lastly, the authors in [15] present centralized adaptive navigation control strategies to characterize sources in 2D and 3D scalar fields, also accounting for plume dynamics. The process incorporates the collective use of individual measurements from diverse environmental points by the cluster members. Unfortunately, despite accurate modeling of UAV dynamics, the work was only implemented in simulation.

In our previous work [8], we have implemented a set of rules used for gradient following, that has been evaluated using a centralized implementation and sensor emulation for static gradient forms devoid of obstacles. In addition, we tested various scalar field types, swarm sizes, and behavioral modifications (such as alignment control) through simulation experiments. In these simulations, our proposed method demonstrated its effectiveness not only on a realistic dynamic model of Crazyflie nano drones but also on point agents modeled at a comparatively large scale. In this work, we implement and demonstrate for the first time a fully onboard implementation of a bio-inspired algorithm for gradient following, using an approach that is a significant extension of the one presented in [8]. We have enhanced our previous implementation by adapting our methodology to accommodate the inputs from a real gradient source instead of emulated values, introducing obstacle avoidance capability with an on-board distance sensor, and manually optimizing all control rules in light of these new factors. Unlike the previous approach, all rules are now implemented on the same micro controller serving as the main flight controller, eliminating the need for centralized implementation.

We use fully onboard controlled Crazyflie nano-drones in an indoor environment (unlike control from a central computer as in [11] or only on simulation as in [14] and [15]), a physical custom-made on-board sensor to implement scalar source sensing (differently from [13] where the measurements are calculated by a stored function), and another physical on-board sensor for sensing and avoiding an obstacle (which is not achieved in [13], [14] and [15]). Each drone senses the gradient via its sensor and retrieves a scalar value (the intensity of the source at its location), but it does not communicate this measurement to other drones (unlike the cases where measurements are communicated to peers for the estimation in [12], [13], [14] and [15]). The drone does not store nor utilizes any historical data, and the method operates reactively unlike other studies we discuss in this section.

III. Methodology

In this section, we outline our methodology for addressing the challenge of dynamic source localization with obstacle avoidance in the absence of global positions and directional information about the source location. This method considers three main behavioural building blocks: collective motion, emergent source localization, and obstacle avoidance. All these building blocks are fully reactive and do not require information about past sensory readings. Moreover, they are all deployed on each drone in a fully onboard fashion. The collective motion building block requires only the relative positions and headings of neighboring agents. With this building block, we expect that the swarm will move in a cohesive and ordered fashion, maintaining safe and predictable distances between agents. For the source localization building block, the source can be sampled using an on-board sensor that gives scalar information on the source intensity at the drone position. The building block employs scalar readings of the gradient field strength at their current locations to modulate their desired distances to maintain with others [8], to achieve emergent source localization (converging to the source location in a way that is only achievable as a collective, individuals are incapable of doing that [7]). Simultaneously, the centroid of this cohesive swarm is expected to follow the increasing gradient of the source strength, ultimately converging on the source center. The hypothesis in this letter is that, when the source location is changed, the swarm will react accordingly to the changed source location. The obstacle avoidance building block uses 4 time-of-flight on-board sensors acting on the principal directions (front, back, left, right) to perceive obstacles. This building blocks combines three contributions: virtual repulsion forces produced according to the direction away from the obstacle; an attraction force towards the agent's neighbors; and a random perturbations component to avoid agents resolving directional deadlocks.

The individual controller of each UAV calculates linear speed (U_i) and angular speed (ω_i). UAVs can translate only in their heading direction by the magnitude of U_i, and this direction is changed by ω_i over time. Although the platform we use (multi-rotor aerial vehicle) can translate in any direction, we impose this constraint on the platforms to apply our methodology, which ensures smooth motions of the UAVs and improves the stability of the collective motion [16]. In addition, the translation constraint makes our methodology easily portable to any kind of platform with minor changes only in the parameters.

The controller takes the relative positions and headings of other UAVs as the input. In addition, each UAV uses its own sensor reading as a source value. Individual sensor readings are used in such a way that UAVs with higher values keep a longer distance to their neighbours, without letting others know about it. Later, the vectors are combined to produce U_i and ω_i, which are applied on each control step.

The core of our method consists in creating a virtual spring between a focal UAV i and its neighbors, by modulating their interpersonal distance according to the focal UAV's individual sensory input (L_s) and the following equation:

$$\sigma_i = \sigma_L + \left(\frac{L_s - L_{\min}}{L_{\max} - L_{\min}}\right)^{0.1} \sigma_r \tag{1}$$

In the above, σ_L and σ_r stand for the lowest allowed value and the modulation range of σ_i, respectively. This modulation increases the desired distance $d_{des} = \sigma_i\sqrt{2}$ of the focal UAV to other UAVs when it is exposed to higher sensor readings, which can also be seen as more asocial behavior. Differently from [8], (1) includes an exponent, not present in the original formulation, that is introduced to cope with realistic environmental gradient profiles. The value of the exponent must be less than 1 because the real gradient tends to change rapidly near the maximum and more smoothly in darker regions. To better respond to small changes in these darker areas, σ_i increases more quickly when L_s has lower values. The exponent of 0.1 was found manually to achieve the best performance with the real gradient source.

The first vector is the proximal vector $\vec{p_i}$ and therefore computes the velocity required to maintain d_{des} between UAV i and each UAV m, that is part of the former's neighborhood N:

$$\vec{p_i} = \sum_{m\in N} -\epsilon\left[2\frac{\sigma_i^4}{(d_i^m)^5} - \frac{\sigma_i^2}{(d_i^m)^3}\right] e^{j\phi_i^m} \tag{2}$$

where d_i^m is the relative distance between UAVs i and m, ϕ_i^m is the relative bearing angle of m on the reference frame of i and ϵ determines the strength of $\vec{p_i}$. Please note that the notation e^{jX}, where X is an angle in radians, follows Euler's formula. In this representation, the real part corresponds to the x-axis and the complex part to the y-axis in the reference frame where angle X is defined. Using this approach, $\vec{p_i}$ and other vectors are computed as 2D vectors with x and y components in the local reference frame of the focal UAV i.

Since neighboring UAVs that sense lower values than the focal UAVs, have a comparatively shorter d_{des}, and are therefore still attracted to the focal UAV. $\vec{p_i}$ enables a chase-evade phenomenon that is directed towards the increasing values of the source gradient [8].

The next vector is the alignment control vector $\vec{h_i}$, which contributes to making the heading directions of UAVs similar. The success of the swarm on the gradient following task is analyzed with and without the alignment control on our previous work [8]. The finding in [8] was that alignment control increases the order in the swarm and transforms the motion into a more reactive, smooth and faster one. To retain these properties with a dynamic gradient source without affecting the performance adversely, we have chosen to keep alignment control and showcase the best possible performance. $\vec{h_i}$ is calculated as a unit vector. The direction of $\vec{h_i}$ is the average of heading directions of all neighbors, resolved in the local frame of the focal UAV.

$$\vec{h_i} = \frac{e^{j\theta_0} + \sum_{m\in\mathcal{N}} e^{j\theta_m}}{||e^{j\theta_0} + \sum_{m\in\mathcal{N}} e^{j\theta_m}||} \tag{3}$$

In the above, θ_m represents the heading angles of neighbor UAVs and θ_0 stands for the heading of the focal UAV.

Then, the boundary avoidance vector, $\vec{r_i^b}$, is responsible for producing a repulsion from the boundary (b) if the focal UAV is closer to it than a safe distance, D_b. $\vec{r_i^b}$ is calculated as the combination of repulsion from all boundaries (B) in which the focal UAV is closer than D_b.

$$\vec{r_i^b} = \sum_{b\in B} k_{rep}\left(\frac{1}{L_b} - \frac{1}{L_0}\right)\left(\frac{\vec{p_i^b}}{L_b^3}\right) \tag{4}$$

In the above, k_{rep} designates the strength of $\vec{r_i^b}$ and L_0 sets the relaxation threshold. L_b is the shortest distance to the boundary b among all close-by boundaries, B. $\vec{p_i^b}$ points the relative direction from the boundary b to the focal UAV to produce repulsion.

The final component of $\vec{f_i}$ is the obstacle avoidance vector, $\vec{o^i}$ (not considered in [8] and added for the first time in this letter).

$$\vec{o^i} = \sum_{d\in D} \vec{o_d^i} \tag{5}$$

where, superscript d represents the direction of sensed obstacles and $\vec{o_d^i}$ represents the sub-vector for each direction. In our application, d can only be in four principal directions (the set D) of the local reference frame; front, back, left, and right, where the front is the current heading direction of the focal UAV. $\vec{o_d^i}$ for each direction is calculated as follows:

$$\vec{o_d^i} = \tau(\delta_d^i)e^{j\varphi} + \mathbb{I}(d)\cdot\frac{1}{2}\tau(\delta_d^i)e^{j\Omega} \tag{6}$$

where δ_d^i is the corrected measured distance with an offset of 15 cm, $\tau(\delta_d^i)$ is the magnitude of the sub-vector:

$$\tau(\delta) = (2.5/\delta)^2 \tag{7}$$

and

$$\mathbb{I}(d) = \begin{cases} 1 & \text{if d is front} \\ 0 & \text{otherwise} \end{cases} \quad ; d \in D \tag{8}$$

Also,

$$\Omega = \text{sign}(\angle\vec{p_i})\frac{\pi}{2}, \tag{9}$$

$$\varphi = \upsilon_i + \psi + \pi. \tag{10}$$

where υ_i is the angle of obstacle direction in the local reference frame ($\upsilon_i \in [\pi, -\frac{\pi}{2}, \frac{\pi}{2}, 0]$). π radians are added to produce the repulsion in the opposite direction. ψ is a random perturbation angle where $\psi \sim Unif(-\frac{1}{2}, \frac{1}{2})$. ψ was introduced to add randomness to the obstacle repulsion to break any local minimum that can lock the UAV's movement. Ω has a similar effect; it adds an extra perturbation in the direction angle ($\angle\vec{p_i}$) of the proximal control vector ($\vec{p_i}$). If $\angle\vec{p_i}$ is positive, the perturbation produces an effect as if there is an obstacle on the right, and if it is negative, the perturbation produces an effect as if there is an obstacle on the left. In the end, all $\vec{o^i}_d$ calculated for all directions (front, back, left, right) are combined as in (5) to get the final vector, $\vec{o}_i$.

Ultimately, the total force vector $\vec{f_i}$ is the weighted sum of all vectors with respective weights, α, β, γ, and κ.

$$\vec{f_i} = \alpha\vec{p_i} + \beta\vec{h}_i + \gamma\vec{r_i} + \kappa\vec{o_i} \tag{11}$$

The linear (U_i) and angular speeds (ω_i) are calculated as follows:

$$U_i = K_1 f_x + u_{add}, \quad \omega_i = K_2 f_y \tag{12}$$

In the above, f_x and f_y are the components of $\vec{f_i}$ on axes of the local reference frame of the focal UAV, where the x-axis is always the heading direction. K_1 and K_2 are the respective speed gains, and u_{add} is the additional propulsion we add to decrease stagnancy in the swarm motion.

U_i is clamped within the range $[0, 0.13]$ m/s and ω_i is clamped within the range $[-0.4\pi, 0.4\pi]$ rad/s. u_{add} is chosen to be 0.05 m/s. The other parameter values are chosen as follows: $\epsilon = 12.0$, $\sigma_L = 0.5$, $\sigma_r = 0.56$, $L_{\min} = 52$, $L_{\max} = 648$, $D_b = 0.5$ m, $k_{rep} = 5.0$, $L_0 = 0.5$, $\alpha = 0.8$, $\beta = 0.2$, $\gamma = 1.0$, $\kappa = 2.0$, $K_1 = 0.06$ and $K_2 = 0.6$. The parameter values are tuned and set manually according to our evaluation. Only $L_{\min}$ and $L_{\max}$ depend on the sensory hardware and environmental conditions.

IV. Implementation and Experimental Setup

The algorithm presented in the previous section is implemented on Crazyflie 2.1 nano-drones, and runs on-board by using the *App-Layer* functionality [17]. The open-source implementation can be found in the GitHub repository [18]. UAVs calculate their own global positions (X and Y on the global frame) on-board with an Ultra-wide Band Localization system. They determine their heading (current yaw angle on the global frame) using their on-board sensors (as implemented in the Crazyflie stack). They exchange positions and headings with each other via a radio communication channel using *P2P* API [19]. Crucially, global positioning and communication are only required due to our drones' inability to perform relative localization. If they were able to estimate the range and bearing of neighboring drones, communication would not be necessary for our methodology. In addition, each drone has a flow camera pointing downwards to improve planar velocity estimation and a distance sensor ranging the altitude.

For the obstacle sensing capability, our Crazyflies carry a deck (multi-ranger) that contains 4 time-of-flight distance sensors in the principal directions; front, back, left, and right. These sensors provide distance to the closest object in the directions they face.

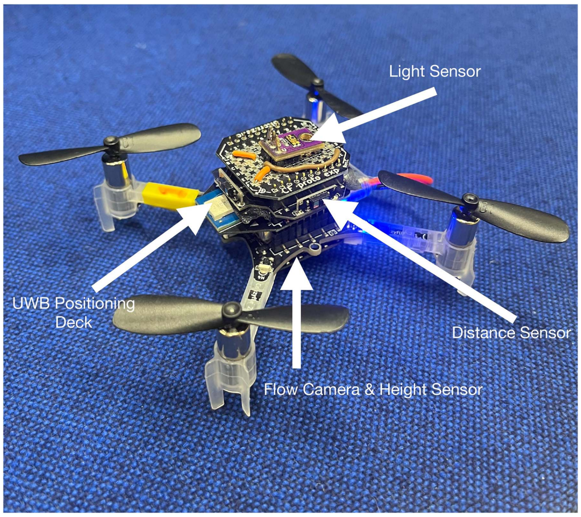


Fig. 1. Crazyflie equipped with sensors.

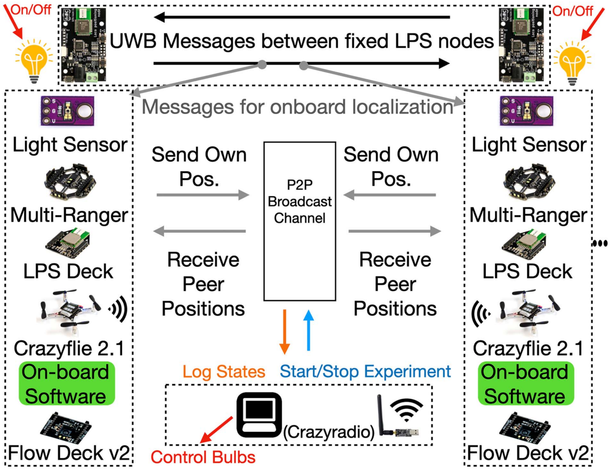


Fig. 2. System architecture.

Given the design of the sensor, blind spots could potentially appear when the obstacle is located between these directions. Distance measurement is done on a ray, so rather than a field of view, we obtain the distance for a single point. The distance values are used as δ^i_d for the corresponding direction.

Physical light sources were used and placed in the environment to realize the experimental version of the source gradient. We built and equipped the Crazyflies with a custom-made light sensor deck on top, that uses TEMT6000 [20] light sensor circuit, which changes the output voltage depending on the light intensity in the environment. The internal software reads this voltage for use in our methodology as L_s. Minimum and maximum voltage values measured in our experimental setup led to our choice of $L_{\min}$ and $L_{\max}$ values. Our custom Crazyflie platform can be seen in Fig. 1. The visualization of the system architecture can be found in Fig. 2.

The experiment arena dimensions are 7.0 m in length and 4.75 m in width. For the whole flight, a constant altitude of 0.5 m is maintained by all UAVs. The arena includes light bulbs on the top, switched on and off by smart plugs. These smart

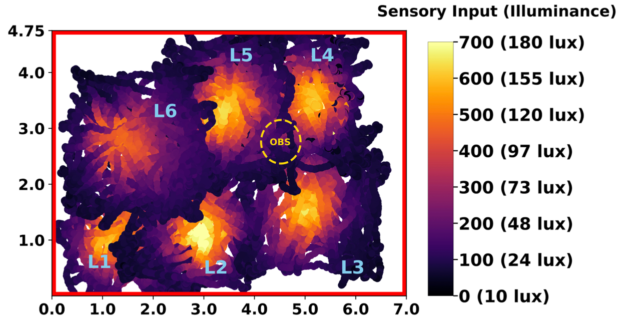


Fig. 3. Light source positions and light intensity heat map, measured sensory input and corresponding illuminance values.

plugs receive messages over the network, to which a control computer is also connected. These individually switchable light sources (which are completely extrinsic for the swarm) constitute the dynamic gradient to be followed by the swarm. Fig. 3 shows the position of the light sources as well as the measured light intensity and illuminance values collected from 6 different flights conducted with a single drone for each light source. This drone visited 100 randomly chosen points when only the corresponding light source was on, and the control computer recorded the position-light intensity data together. For the sensory input values obtained, which theoretically range from 0 to 4096 due to the analog voltage input, the corresponding illuminance levels are calculated based on the sensor's datasheet [20]. Factors such as temperature, variations in wavelength, and electrical losses are not considered in these calculations. Red lines show the arena boundary. We use the information visualized in Fig. 3 for two purposes: Deciding $L_{\min}$, $L_{\max}$ values, and locating the light centers to assess the gradient following performance of the swarm.

In order to test the performance of the swarm following the dynamic gradient, we designed 4 sets of experiments. In experiment **A**, the lights are switched on sequentially, starting from L1 to ending with L6. Each light stays on for 35 seconds. In experiment **B**, the L4-L2-L6 sequence is followed with a switch every 65 seconds. The duration for each switch is longer in **B**, since light sources are further and intersections between lightened areas are smaller. In experiment **C**, lights are switched two by two with the following sequence: [L1-L6]-[L2-L5]-[L3-L4]-[L2-L5]-[L1-L6]. By doing so, we want to test the performance of our approach on a bimodal gradient. The period for the light switch is 50 seconds at **C**. Finally, experiment **D** follows the sequence L3-L4-L5, with the addition of an obstacle, placed on the location OBS, as shown in Fig. 3. The circle-shaped obstacle is built using off-the-shelf materials such as paper and wooden sticks, shown in Fig. 4. Given the design of the sensor (possibility of blind spots), the diameter is selected to be 80 cm to guarantee that UAVs can sense the obstacle from a safe distance of approximately 14 cm. As the obstacle makes moving from L3 to L4 and L4 to L5 more challenging, the switching periods are set to 20-90-90 seconds. With this setup, we assess the swarm's capability to follow changing light sources, when an obstacle is on the way, without any crash. It should be noted that obstacle only exists in experiment **D**.

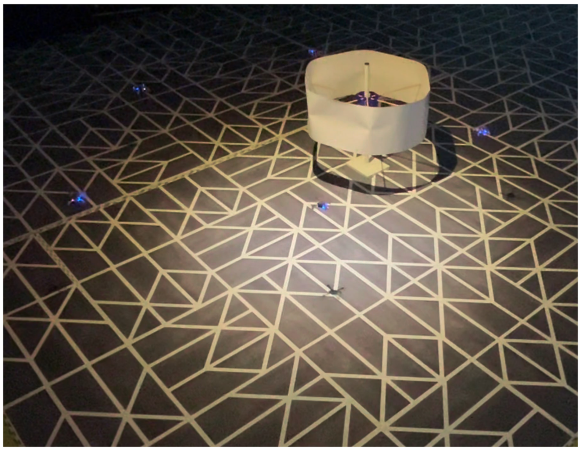


Fig. 4. Crazyflie swarm in the air, experiment **D**.

Each experiment starts with the control computer triggering an internal switch in the on-board software of Crazyflies, through radio communication. Simultaneously, a timer starts on the computer, which sends switch signals to the lights. Afterward, the positions, heading angles, and light measurements of the UAVs are obtained and recorded 2 times per second for the assessment. When the experiment timer is finished, a signal is sent, which causes all UAVs to land.

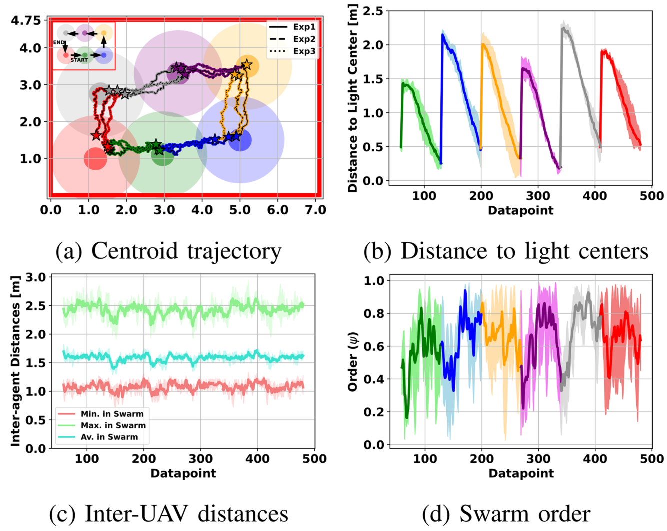
V. Results and Discussion

First, we observe the gradient following behavior by analyzing the trajectory of the swarm centroid. The positions of the UAVs are averaged for each data point to get the centroid position. We use centroid trajectory because, when the swarm arrives at the next light center successfully, UAVs position themselves around the center instead of clustering in the middle as they must maintain a distance (d_{des}) from each other by our methodology. This makes observing the centroid trajectory more intuitive than observing individual UAV paths. Nevertheless, links for animation videos of the whole set of experiments can be found on the playlist,[1] where individual UAV motions can also be seen. In addition, real flight videos can be found on this playlist.[2]

In Fig. 5(a), centroid trajectories for 3 runs of experiment **A**, can be seen together. Colors indicate the current active light source, and *stars* indicate the position of the centroid when the light is switched to the next one. Colored regions show the approximate areas illuminated by the corresponding light sources. The trajectories and the locations of the *stars* from 3 different experiments indicate a successful gradient following behavior and the emergence of collective localization of the source center. Fig. 5(b) shows the distance of the centroid to the active light center over time (light center positions are estimated from Fig. 3). Lines indicate the average of 3 experiments, whereas shaded regions are spanned between maximum and minimum values. This figure clearly shows the steady and repeatable nature of the

[1] [Online]. Available: https://www.youtube.com/playlist?list=PLQXaE6NbHSe2d1Y2zXLiO3y00d4w9OrXR or http://bit.ly/3QDp2W6

[2] [Online]. Available: https://www.youtube.com/playlist?list=PLQXaE6NbHSe39z1X2zpjZD6f6mB9f40u9 or https://bit.ly/3MnF3gq



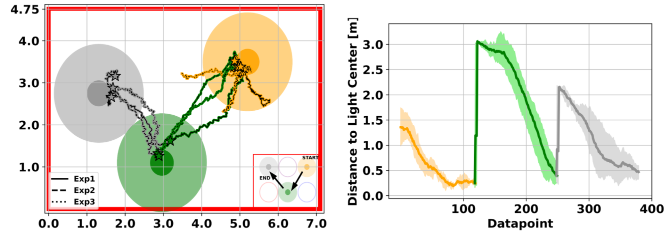
(a) Centroid trajectory

(b) Distance to light centers

(c) Inter-UAV distances

(d) Swarm order

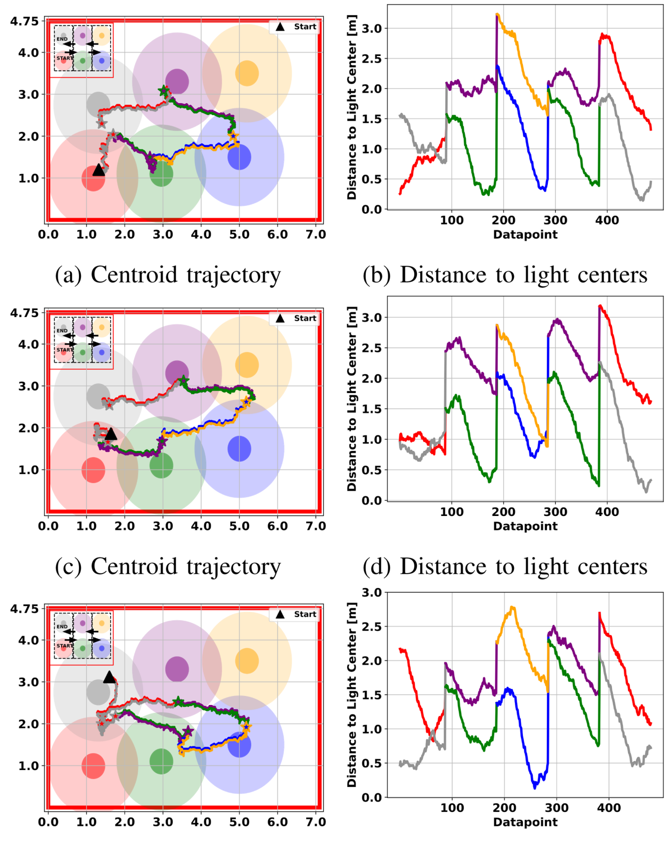
Fig. 5. Experiment **A**.



(a) Centroid trajectory

(b) Distance to light centers

Fig. 6. Experiment **B**.



(a) Centroid trajectory

(b) Distance to light centers

(c) Centroid trajectory

(d) Distance to light centers

(e) Centroid trajectory

(f) Distance to light centers

Fig. 7. Experiment **C**.

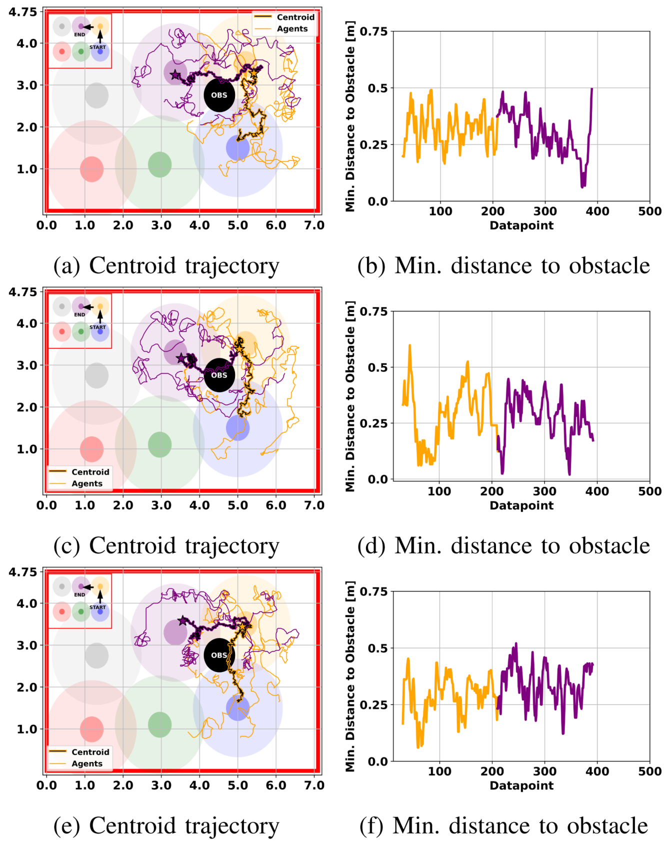
collective following of a dynamic gradient. Fig. 5(c) shows the average, maximum, and minimum distances between all UAVs of the swarm at each data point. Here we can see that UAVs never become dangerously close nor crash in any of the 3 experiments. The most important information we can extract from this data is that, although our methodology depends on the heterogeneity of desired distances of the UAVs, the difference thereof never becomes large enough to disturb the swarm's coherence or to increase the risk of inter-UAV collisions. To analyze Fig. 5(d), we define a metric for the agreement of the UAV's headings; order. The order is calculated as follows:

$$\Psi = \frac{\| \sum_{i \in N_A} e^{j\theta_i} \|}{N_A} \tag{13}$$

In the above, θ_i represents the heading angle of each UAV within the whole swarm (N_A). Ψ becomes 1 when all headings are perfectly parallel and approach zero when headings are different. When Fig. 5(d) and 5(b) are observed together, we can see that Ψ increases when the active light is switched, and the swarm starts to move in the direction that increases the new light gradient. When the swarm centroid becomes closer to the light center, Ψ decreases, and the swarm stops. This correlation between distance to the light center and the swarm order helps to validate, once more, the coupling between collective behavior and environmental change.

Fig. 6(a) shows the centroid trajectories from 3 different experimental runs **B**. Here we can observe that the yellow light region (L4) and green light region (L2) are located considerably further than consecutive ones as in experiment **A**. Despite that, the swarm successfully follows the weaker gradient (comparatively to **A**) in all 3 runs, and reaches the light center. As expected from the limited maximum speed of UAVs and increased distance between centers, reaching the green center from the yellow center takes longer. When Fig. 6(b) is also considered, we can conclude that the swarm still achieves gradient following in a steady and repeatable manner.

In experiment **C**, where the lights are switched two by two, starting positions are more diverse (closer to one of the centers or intersection region) and we could therefore expect the swarm's starting position to significantly alter the emerging behavior. To assess the flexibility of our algorithm, we consider the three cases that are possible within the first region consisting of L1 and L6; The swarm can initially be located around the L1 center, between L1 and L6, or around the L6 center. Except from scenarios where individual sources should be localized separately, we believe converging to the intersection between two sources is a valid result in scenarios that requires scanning and covering multiple sources simultaneously. Fig. 7(a), (c) and (e) show centroid trajectories from experiments in which the swarm starts from each of these 3 starting point options.



(a) Centroid trajectory

(b) Min. distance to obstacle

(c) Centroid trajectory

(d) Min. distance to obstacle

(e) Centroid trajectory

(f) Min. distance to obstacle

Fig. 8. Experiment **D**.

Different from previous figures, starting points are indicated with a black triangle, and trajectories consist of two colors corresponding to the active lights. We see in the trajectory plots that, regardless of the starting point, the swarm is able to proceed to the next light region after the light switch. Unlike other experiments, the swarm is not necessarily located in a center; it can also stay at the intersection of two light regions, which has an attractive light intensity. Fig. 7(b), (d) and (f) show distances to the light centers for each experiment. Here we can clearly observe the same phenomenon: the swarm sometimes converges towards the intersection of regions. Although 7(e) reveals different results regarding where the swarm positioned itself, our current analysis would require further refinement to answer the question: What determines whether the swarm wanders at the intersection or gravitates toward the centers? In order to answer this question, future experiments we will conduct would necessitate a more consistent set of initial conditions coupled with enhanced precision in light field control. While our current study does not encompass this in-depth behavioral analysis, it lays the foundation for such investigations, highlighting the criticality of understanding collective gradient following behavior.

In experiment **D**, we test the gradient following behavior 3 times when there is an obstacle on the way, which can be locally sensed by each UAV within a limited range (D_b). In Fig. 8(a), (c) and (e), trajectories of both swarm centroids and the UAVs from experiment **D** are presented. The UAV trajectories help validating that there was no crash between the UAVs and the obstacle during the experiments and demonstrate the avoidance behavior. For further information, Fig. 8(b), (d), and (f) report the minimum distance occurring between any UAV and the obstacle during experiments. When both types of figures are observed, we can see that the swarm successfully followed the light gradient. The *stars*, which represents the centroid location when lights are switched, clearly shows the swarm took position around the bright light center. Nevertheless, we see that the distance to the obstacle can decrease to unsafe values occasionally. Despite our model being designed to avoid such situations, hardware is a limiting factor in this instance. For instance, due to the design of the distance sensor, it is reasonably possible for UAVs to miss the obstacle when the obstacle is located between two of these principal directions, relative to the UAV. There are only two ways to compensate for these blind spots: Either allowing UAVs to get closer to the obstacle, or making sure the obstacle is large enough. Since we have a limited space to fly, the optimum point we found between these two alternatives was to set a safe obstacle diameter, as stated in Section IV. Even so, occasional unsafe moments were not entirely avoidable, given the localization errors. If we were to have smaller obstacles, the situation could be expected to be worse. The most reliable way of fixing this issue is by having sensors in more directions, possibly omnidirectional (360°). Yet we still see this as an achievement to integrate on-board obstacle avoidance capability to the swarm without diminishing the follow of the dynamic gradient. In future work, we aim to further refine the integration between gradient following and obstacle avoidance. By doing so, we anticipate smoother movement of the swarm around obstacles, eliminating the staggered motions observed in the current implementation.

In addition to the obstacle avoidance, the most pressing upgrade before tackling real-world problems is relative localization. While our current methodology needs no change to comply, our nano drones have not achieved it yet. A solution by using UWB radios currently equipped should also be scalable and similarly accurate to guarantee the same effectiveness. Additionally, although we have tested real light gradients, real-world sources like gases will introduce more noise and complexity. Future tests should adapt to these challenges.

VI. Conclusion

In this letter, we present an approach for a swarm of nano drones to follow the gradient of a dynamic source without exchanging any information about the scalar measurements made by individual UAVs. The control and sensing are implemented fully on-board, for the first time. In addition, obstacle avoidance capability is also achieved on-board and does not need an explicit information exchange within the swarm. This demonstration of on-board gradient following, with a swarm of drones where each can fit in a palm, is a promising step towards more challenging applications.

Our framework is designed to be generic, so that deployment on various platforms does not require any modification of the fundamental blocks. The only building block we use that still require external infrastructure is the relative localization, which in turn imposes the constraint of having communication.

However, (even slightly) more capable platforms that enable relative localization of peers, would not need any communication between UAVs at all. Moreover, any type of sources can be followed (gas source, radiation source, signal source, etc.), as long as the UAVs can be equipped with relevant sensors. In any case, these sensors will only need to provide scalar samples (signal strength, parts per million (PPM) values, etc.). On larger platforms, omnidirectional ranging sensors can be used to avoid obstacles in various shapes and sizes.

Considering the wide range of possibilities, we plan to extend the capabilities of our swarm in the future by developing more capable platforms, or improving our method in different ways (e.g. swarm speed, improved obstacle avoidance with multiple static or dynamic obstacles, 3D source localization, etc.). We believe this technology can be relevant to real world problems, which require reaching places that are dangerous, and efficient search behaviors, in case of emergencies. In line with the inherent requirements of real-world applications, our future vision aims at considering larger scale scenarios, and to enhance the swarm's exploration capabilities, in addition to following a newly detected source gradient. Although this objective requires a significant transformation in collective behavior, it will allow the swarm to autonomously search for an unknown source or transition between multiple non overlapping sources.

REFERENCES

[1] J. Han, Y. Xu, L. Di, and Y. Chen, "Low-cost multi-UAV technologies for contour mapping of nuclear radiation field," *J. Intell. Robot. Syst.*, vol. 70, pp. 401–410, 2013, doi: 10.1007/s10846-012-9722-5.

[2] Z. Fu, Y. Chen, Y. Ding, and D. He, "Pollution source localization based on multi-UAV cooperative communication," *IEEE Access*, vol. 7, pp. 29304–29312, 2019, doi: 10.1109/ACCESS.2019.2900475.

[3] J. Burgués and S. Marco, "Environmental chemical sensing using small drones: A review," *Sci. Total Environ.*, vol. 748, 2020, Art. no. 141172, doi: 10.1016/j.scitotenv.2020.141172.

[4] M. Campion, P. Ranganathan, and S. Faruque, "UAV swarm communication and control architectures: A review," *J. Unmanned Veh. Syst.*, vol. 7, no. 2, pp. 93–106, 2019, doi: 10.1139/juvs-2018-0009.

[5] E. Şahin, "Swarm robotics: From sources of inspiration to domains of application," in *Proc. Int. Workshop Swarm Robot.*, 2004, pp. 10–20, doi: 10.1007/978-3-540-30552-1_2.

[6] J. G. Puckett, A. R. Pokhrel, and J. A. Giannini, "Collective gradient sensing in fish schools," *Sci. Rep.*, vol. 8, no. 1, pp. 1–11, 2018, doi: 10.1038/s41598-018-26037-9.

[7] A. Berdahl, C. J. Torney, C. C. Ioannou, J. J. Faria, and I. D. Couzin, "Emergent sensing of complex environments by mobile animal groups," *Science*, vol. 339, no. 6119, pp. 574–576, 2013, doi: 10.1126/science.1225883.

[8] T. A. Karagüzel, A. E. Turgut, A. Eiben, and E. Ferrante, "Collective gradient perception with a flying robot swarm," *Swarm Intell.*, vol. 17, no. 1/2, pp. 117–146, 2022, doi: 10.1007/s11721-022-00220-1.

[9] T. Vicsek and A. Zafeiris, "Collective motion," *Phys. Rep.*, vol. 517, no. 3/4, pp. 71–140, Aug. 2012, doi: 10.1016/j.physrep.2012.03.004.

[10] B. P. Duisterhof et al., "Tiny robot learning (tinyRL) for source seeking on a nano quadcopter," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2021, pp. 7242–7248, doi: 10.1109/ICRA48506.2021.9561590.

[11] N. Elkunchwar et al., "Bio-inspired source seeking and obstacle avoidance on a palm-sized drone," in *Proc. IEEE Int. Conf. Unmanned Aircr. Syst.*, 2022, pp. 282–289, doi: 10.1109/ICUAS54217.2022.9836062.

[12] B. P. Duisterhof, S. Li, J. Burgués, V. J. Reddi, and G. C. H. E. de Croon, "Sniffy Bug: A fully autonomous swarm of gas-seeking nano quadcopters in cluttered environments," in 2021*Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2021, pp. 9099–9106, doi: 10.1109/IROS51168.2021.9636217.

[13] A. Datar, P. Paulsen, and H. Werner, "Flocking towards the source: Indoor experiments with quadrotors," in *Proc. Eur. Control Conf.*, 2020, pp. 1638–1643, doi: 10.23919/ECC51009.2020.9143867.

[14] S. Zhuo, "Source seeking of multi-UAV based on extremum seeking algorithm," in *Proc. 17th Int. Conf. Control, Autom. Syst.*, 2017, pp. 1062–1067, doi: 10.23919/ICCAS.2017.8204373.

[15] R. K. Lee, C. A. Kitts, M. A. Neumann, and R. T. Mcdonald, "Multiple uav adaptive navigation for three-dimensional scalar fields," *IEEE Access*, vol. 9, pp. 122626–122654, 2021, doi: 10.1109/ACCESS.2021.3107854.

[16] E. Ferrante, A. E. Turgut, C. Huepe, A. Stranieri, C. Pinciroli, and M. Dorigo, "Self-organized flocking with a mobile robot swarm: A novel motion control method," *Adaptive Behav.*, vol. 20, no. 6, pp. 460–477, 2012, doi: 10.1177/1059712312462248.

[17] "Bitcraze - crazyflie firmware - app layer documentation," 2022. [Online]. Available: [Online]. Available: https://www.bitcraze.io/documentation/repository/crazyflie-firmware/master/userguides/app_layer/

[18] "Project repository," 2023. [Online]. Available: [Online]. Available: https://github.com/tugayalperen/Gradient_following_firmware

[19] "Bitcraze - crazyflie firmware - p2p documentation," 2022. [Online]. Available: https://www.bitcraze.io/documentation/repository/crazyflie-firmware/master/functional-areas/p2p_api/

[20] "Vishay semiconductors - temt6000x01 datasheet," 2023. [Online]. Available: https://www.vishay.com/docs/81579/temt6000.pdf