

© 2022 IOP Science. Personal use of this material is permitted. Permission from IOP Science must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.

PACNav: A Collective Navigation Approach for UAV Swarms Deprived of Communication and External Localization

Afzal Ahmad^{1*}, Daniel Bonilla Licea¹, Giuseppe Silano^{2,1}, Tomáš Báča¹, and Martin Saska¹

¹Department of Cybernetics, Faculty of Electrical Engineering, Czech Technical University in Prague, 16636, Prague 6, Czech Republic.

²Department of Generation Technologies and Materials, Ricerca sul Sistema Energetico (RSE) S.p.A., 20134, Milan, Italy.

*Author to whom any correspondence should be addressed.

E-mail: ahmadafz@fel.cvut.cz, bonildan@fel.cvut.cz, giuseppe.silano@fel.cvut.cz, tomas.baca@fel.cvut.cz, and martin.saska@fel.cvut.cz

Received 20 July 2022

Revised 26 September 2022

Accepted 7 October 2022

Published 11 October 2022

Abstract.

This article proposes Persistence Administered Collective Navigation (PACNav) as an approach for achieving decentralized collective navigation of Unmanned Aerial Vehicle (UAV) swarms. The technique is based on the flocking and collective navigation behavior observed in natural swarms, such as cattle herds, bird flocks, and even large groups of humans. As global and concurrent information of all swarm members is not available in natural swarms, these systems use local observations to achieve the desired behavior. Similarly, PACNav relies only on local observations of relative positions of UAVs, making it suitable for large swarms deprived of communication capabilities and external localization systems. We introduce the novel concepts of **path persistence** and **path similarity** that allow each swarm member to analyze the motion of other members in order to determine its own future motion. PACNav is based on two main principles: (1) UAVs with little variation in motion direction have high **path persistence**, and are considered by other UAVs to be reliable leaders; (2) groups of UAVs that move in a similar direction have high **path similarity**, and such groups are assumed to contain a reliable leader. The proposed approach also embeds a reactive collision avoidance mechanism to avoid collisions with swarm members and environmental obstacles. This collision avoidance ensures safety while reducing deviations from the assigned path. Along with several simulated experiments, we present a real-world experiment in a natural forest, showcasing the validity and effectiveness of the proposed collective navigation approach in challenging environments. The source code is released as open-source, making it possible to replicate the obtained results and facilitate the continuation of research by the community.

Keywords: Swarm Robotics, Relative Localization, Decentralized Control, Unmanned Aerial Vehicles.

Submitted to: *Bioinspir. Biomim.*



Figure 1: Picture from the flight experiment performed in a natural forest.

1. Introduction

The collective motion of a tightly cooperating group of UAVs has been intensively investigated in recent years [1, 2]. The use of a group of UAVs can reduce mission time and provide the redundancy and safety that is critical in many real-world applications, such as mapping large areas [3], construction [4], agriculture [5], and search-and-rescue missions [6]. These applications have further motivated research on the collective motion of a group of UAVs [7–9]. Indeed, large groups of UAVs are effective in some scenarios where facing the problem with a single robot may be unfeasible or difficult to solve. UAV swarms have also been useful for applications where redundancy is needed to cope with failure of individual UAVs [10] and in cases when multiple UAVs need to gather data simultaneously [11].

The deployment of a group of UAVs requires a complex system composed of several intricate subsystems handling vehicle control, environment perception, absolute or relative localization, mapping, navigation, and communication. Therefore, the use of centralized control architectures which require UAVs to communicate and exchange information over a shared network may explode in complexity as the number of vehicles and the size of the working area increases. Thus, it is important to introduce decentralized control architectures, fault detection systems, and feasible information sharing over a low-to-none bandwidth network.

Animals, including fish and birds, are an important source of inspiration to tackle this problem due to their use of decentralized decision making for collective motion [12–14]. Reynolds [15] described this motion using a set of simple rules addressing attraction to the group, repulsion from neighbors, and alignment to move in the same

direction as the group. These rules have since been adapted for use in robotic swarms with additional components for obstacle avoidance [16]. The simplicity of the rules and the decentralized nature of the control strategy provides robustness against the failure of individual robots. In many cases, the decentralized decision making system only depends on local information about the neighbors, which makes these methods scalable to a large number of robots.

Decentralized control of a swarm of UAVs can also be achieved using predictive controllers and trajectory planning. The swarm of UAVs in [17] uses reactive avoidance and Model Predictive Control (MPC) strategies for collective navigation. On the other hand, [18] presented a nonlinear optimization method for decentralized trajectory planning. However, these methods require a shared reference frame to localize all the UAVs and often rely on shared information about the position and future trajectory of other UAVs. Establishing a shared reference frame and communication links is often unfeasible or difficult to achieve in practice due to unreliable Global Navigation Satellite System (GNSS) signals, as well as losses in wireless communication caused by environmental occlusions and reflections. Thus, dependence on localization and communication infrastructure can severely limit the use of these methods in environments cluttered with obstacles. In cluttered spaces such as forests and construction sites, it is common practice to use onboard sensors for the localization and relative pose estimation of other UAVs. Although these sensors often have a higher computational demand than those used for global localization and communication, they can provide information at a rate sufficient for stable motion. The work in [19] presents a decentralized swarm that does not use communication, but still relies on a shared global reference frame (GNSS). In our previous works [20, 21], we described a UAV swarm that navigates towards a goal completely independent of any shared reference frame and communication infrastructure. However, both [20, 21] do not perform well in the presence of occlusions from the surrounding obstacles and scale poorly with a growing number of UAVs.

Dependence on communication has a negative influence on the scalability of the swarm. Moreover, sharing a reference frame may require exchange of mapping information and synchronization in GNSS-denied environments [22], which has adverse effects on robustness. Therefore, this article proposes a bioinspired decentralized approach for collective navigation of a swarm of UAVs without using GNSS and any communication. The UAVs are controlled using only onboard sensor data, which is used for detecting and localizing obstacles and other team members. The proposed approach builds upon the analysis of collective motion of groups of animals and humans [23] in order to design path similarity and path persistence met-

rics for comparing the trajectories of UAVs. Target UAVs are selected based on these metrics and consequently followed by individual UAVs, resulting in a collective motion. The approach does not rely on any prior information about the environment or the swarm and can be deployed in an unknown environment cluttered with obstacles. The collision avoidance mechanism proposed as part of the approach places high emphasis on safety and is specifically designed to operate in complex unknown environments. In particular, collision avoidance is designed in such a way so as to reduce deviations from the assigned paths by only reacting to the UAVs showing an immediate threat of collision. This is similar to the emergency instincts which are often observed in swarms of animals in nature. Simulated experiments and real-world flight in a natural forest have been used to validate and analyze the performance and robustness of the proposed approach. Compared to our previous works [20, 21], the approach proposed in this article uses relative localization information, recorded over a finite time horizon, to overcome the challenges of occlusion of UAVs by obstacles. The source code related to this work has been released as open-source[‡] for easy replication and future work by the community. Additionally, videos from the simulations and real-world experiments have been provided as supplementary multimedia material and made available at <http://mrs.felk.cvut.cz/pacnav>.

1.1. Related works

UAV swarms have been a topic of several studies and have recently gained a lot of attention due to their useful properties, such as adaptability, scalability, reliability, and fault-tolerance [2, 24, 25]. The idea of using group intelligence stems from natural biological systems, such as flocks of birds, schools of fish, or swarms of bees. One of the first simulated models of the natural flocking behavior of birds was introduced in [15], where the motion of a swarm of dimensionless particles was controlled by a simple set of rules. [2] describes several different methods to control a swarm of UAVs, including physics-based models and MPC. However, most of the recent research has been limited to laboratory-like conditions, as presented in [26–29].

1.1.1. No obstacle avoidance Recent works presented in [19, 30, 31] study the real-world deployment challenges of UAV swarms. [19] uses the method developed in [15] for controlling a swarm of UAVs in a real-world environment, where each UAV uses GNSS for localization and shares this position information with other UAVs over a communication network. The swarm in [30] also uses GNSS and communication for flocking, but has additional constraints on energy consumption. The reliance of [19, 30] on GNSS makes them unsuitable for several real-world scenarios where GNSS is unreliable or unavailable, *e.g.*, forests and indoor construction sites. [31] presented a swarm of UAVs for GNSS-denied environments that uses inter-UAV communication to control the swarm. However, inter-UAV communication scales poorly with a growing number of UAVs in the swarm and can become

a bottleneck in cluttered environments where obstacles reduce the communication range. The work presented in [29] uses a computer vision based technique for relative localization of UAVs. The use of on-board sensors for relative localization makes this method independent of any communication infrastructure and associated scalability issues. However, [19, 29–31] do not implement any collision avoidance, thus limiting the deployment to only obstacle-free environments.

1.1.2. Obstacle avoidance with relative localization and communication The swarm system presented in [18] uses cameras for relative localization and obstacle avoidance in a cluttered environment. The motion planning problem of moving the UAVs towards a goal is solved as a nonlinear optimization problem on-board each vehicle. However, this method uses a broadcast communication network to share UAVs' trajectories for collision avoidance between vehicles. The method presented in [32] uses a UltraViolet (UV) light based visual relative localization system for the UAVs. Cameras placed on the UAVs localize other UAVs in a relative reference frame, removing the need for sharing position information. Although the swarm in [32] can avoid static obstacles, the real-world experiments use artificial obstacles and very low environment density. Moreover, the obstacle positions are known *a priori*, which is difficult to achieve in arbitrary real-world deployment scenarios. The bioinspired method introduced in [16] models the obstacles as agents or as a group of agents depending on the obstacles number. This approach is easy to implement, but relies on precise obstacle positions and shape estimates which are not trivial to obtain in complex environments. Furthermore, such an approach can often create a virtual deadlock when the UAV agent is surrounded by several obstacles, as is common in cluttered real-world environments, such as forests or indoor construction sites.

1.2. Contributions

This article proposes Persistence Administered Collective Navigation (PACNav) as a decentralized approach to navigating a UAV swarm without communication and without global localization infrastructure from its initial position to a goal point. We address several challenges related to the real-world deployment of a swarm of UAVs and their collective motion in a cluttered environment. The main contributions going beyond the previously presented literature include:

- a bioinspired decentralized approach to infer goal direction using motion of UAVs observed directly from on-board sensors, similar to the sensory organs of animals. This approach is based on novel concepts of path persistence and path similarity which make it resilient to sensor uncertainties observed during real-world deployment;
- an approach for motion planning and reactive collision avoidance to safely navigate the swarm in cluttered environments without communication among the UAVs and any global localization system.

[‡] <https://github.com/ctu-mrs/pacnav>

1.3. Notation

This article will use $[\mathbf{A}]_i$ to denote the i -th column of a matrix \mathbf{A} , and $[\mathbf{A}]_{ij}$ to denote the i -th row and the j -th column of the matrix \mathbf{A} . For any two matrices \mathbf{A} and \mathbf{B} , $[\mathbf{A}, \mathbf{B}]$ is used to represent column concatenation. If \mathcal{S} is a set, then $|\mathcal{S}|$ denotes its cardinality. For any two vectors \mathbf{a} and \mathbf{b} , their inner product is written as $\mathbf{a} \cdot \mathbf{b}$. The symbol $\|\bullet\|$ denotes the Euclidean norm.

2. System Model

We consider a swarm composed of N -UAVs that move on a horizontal plane (XY -plane) and assume that we can directly control their velocities along these axes. Thus, the system dynamics of the i -th UAV, at discrete time index k , can be described with a point-mass model as:

$$\dot{\mathbf{p}}_i[k] = \mathbf{u}_i[k], \quad (1)$$

where $\mathbf{p}_i[k] \in \mathbb{R}^2$ is the position in the world coordinate frame \mathcal{F}_W , and $\mathbf{u}_i[k] \in \mathbb{R}^2$ is the velocity control input expressed in the same reference system (\mathcal{F}_W).

We assume that UAVs in the swarm do not have prior information about other UAVs and cannot communicate among themselves. Each UAV is equipped with an omnidirectional camera, allowing for position estimation of the other UAVs using an image processing method [33–35]. Obstacles in the environment may occlude the camera view and, consequently, direct observation of other UAVs' positions is not always possible. Thus, the estimate $\check{\mathbf{p}}_{ij}[k]$ of the j -th UAV position when observed by the i -th UAV, at discrete time k , is modeled as:

$$\check{\mathbf{p}}_{ij}[k] = \mathbf{q}_{ij}[k] + \mathbf{j}_{ij}[k], \quad (2a)$$

$$\mathbf{q}_{ij}[k] = \mathbf{p}_{ij}[k]f_{ij}[k] + \check{\mathbf{p}}_{ij}[k-1]\bar{f}_{ij}[k], \quad (2b)$$

where $\mathbf{p}_{ij}[k] \in \mathbb{R}^2$ is the relative ground truth position of the j -th UAV with respect to the i -th UAV. $f_{ij}[k] = 1$ if there is Line of Sight (LoS) between the i -th and j -th UAVs and $f_{ij}[k] = 0$ otherwise. $\bar{f}_{ij}[k]$ is the complementary function of $f_{ij}[k]$. $\mathbf{j}_{ij}[k] \in \mathbb{R}^2$ is the estimation error which is assumed to be a Gaussian random process with zero-mean and covariance matrix $\sigma^2[k]\mathbf{I}$, where $\mathbf{I} \in \mathbb{R}^{2 \times 2}$ represents the identity matrix, and:

$$\sigma^2[k] = f_{ij}[k]\sigma_{\text{LoS}}^2 + \bar{f}_{ij}[k]\sigma_{\text{NLoS}}^2. \quad (3)$$

The terms σ_{LoS} and σ_{NLoS} capture the different sources of estimation error. In the case of LoS, the estimation error is mostly influenced by the sensor noise. However, when the LoS is lost, factors, including UAV motion or other random processes, further contribute to the estimation error. For simplicity, we assume omnidirectional sensing, although (2) can be adapted to consider any other directional sensor.

When there is LoS between the i -th and the j -th UAVs, the error in the position estimate is modeled by the additive Gaussian noise (see (2)) with zero-mean and covariance matrix $\sigma_{\text{LoS}}^2\mathbf{I}$. Depending on the available LoS, the i -th UAV performs two actions: i) it adds the index j to

Algorithm 1: Updating set $\mathcal{N}_i[k]$

Data: $\mathcal{N}_i[k-1], f_{ij}[k], \delta_{ij}[k-1]$

Result: $\mathcal{N}_i[k]$

```

/* add indices of UAVs that have LoS
   with the i-th UAV at time instant k
*/
 $\mathcal{N}_i[k] \leftarrow \mathcal{N}_i[k-1] \cup \{j : f_{ij}[k] = 1, j \notin \mathcal{N}_i[k-1]\}$ 
for  $j \in \mathcal{N}_i[k]$  do
    if  $f_{ij}[k] = 1$  then
        /*  $\delta_{ij}$ : latest recorded time
           when there was LoS between
           the i-th and j-th UAVs */
         $\delta_{ij}[k] \leftarrow k$ 
    else
        |  $\delta_{ij}[k] \leftarrow \delta_{ij}[k-1]$ 
    end
    /* remove j from  $\mathcal{N}_i[k]$  if LoS with
       the i-th UAV has been lost for
       more than  $K^m$  time instants */
    if  $k - \delta_{ij}[k] > K^m$  then
        |  $\mathcal{N}_i[k] \leftarrow \mathcal{N}_i[k] \setminus j$ 
    end
end

```

the set of neighbors $\mathcal{N}_i[k]$ whenever LoS is obtained between them. However, when the LoS is lost, the estimate $\check{\mathbf{p}}_{ij}$ becomes an autoregressive process which provides feedback to the noise process \mathbf{j}_{ij} . Its expected value is the last known position of the j -th UAV, and its covariance matrix grows linearly at a rate of σ_{NLoS}^2 per sampling instant; ii) the i -th UAV removes the index j from $\mathcal{N}_i[k]$ if the LoS with the j -th UAV is lost for more than K^m time instants, where $K^m \in \mathbb{R}$ is a design parameter. A δ_{ij} variable keeps track of the latest known time instant when there was LoS between the i -th and j -th UAVs. Algorithm 1 describes the update of $\mathcal{N}_i[k]$ in more detail.

The approach proposed in this article relies on a sequence of position estimates which are stored in a matrix $\mathbf{H}_{ij}[k]$, called the *path history* matrix. When index j is added to $\mathcal{N}_i[k]$, the i -th UAV starts to store the position estimates $\check{\mathbf{p}}_{ij}[k]$ into a path history matrix $\mathbf{H}_{ij}[k]$, along with the corresponding time instant of the estimate into the matrix $\Gamma_{ij}[k]$. Algorithm 2 describes the update process for $\mathbf{H}_{ij}[k]$, where the estimates are sequentially concatenated in $\mathbf{H}_{ij}[k]$. $[\mathbf{H}_{ij}]_1$ contains the newest position estimate of the j -th UAV, and $[\mathbf{H}_{ij}]_{L_{ij}}$ the oldest one, where $L_{ij} \in \mathbb{N}_{>0}$ is the number of columns of the matrix \mathbf{H}_{ij} . When an estimate is older than the design parameter $K^p \in \mathbb{Z}_{>0}$, it is removed from $\mathbf{H}_{ij}[k]$, and so $L_{ij} \leq K^p$.

We now use the previously described data structures to introduce some metrics and notation that will be helpful for the description of the proposed PACNav approach detailed in the following sections.

Definition 1 (Path Similarity): This metric describes the similarity between the motion of two

Algorithm 2: Updating $\mathbf{H}_{ij}[k]$

Data: $\mathbf{H}_{ij}[k-1], \Gamma_{ij}[k-1], \check{\mathbf{p}}_{ij}[k]$
Result: $\mathbf{H}_{ij}[k]$

```

if  $j \in \mathcal{N}_i[k]$  then
|    $\mathbf{H}_{ij}[k] \leftarrow [\check{\mathbf{p}}_{ij}[k], \mathbf{H}_{ij}[k-1]]$ 
|    $\Gamma_{ij}[k] \leftarrow [k, \Gamma_{ij}[k-1]]$ 
else
|    $\mathbf{H}_{ij}[k] \leftarrow \mathbf{H}_{ij}[k-1]$ 
|    $\Gamma_{ij}[k] \leftarrow \Gamma_{ij}[k-1]$ 
end

/* remove last column if older than  $K^p$  time instances */

if  $k - [\Gamma_{ij}[k]]_{L_{ij}} > K^p$  then
|   remove  $([\mathbf{H}_{ij}[k]]_{L_{ij}}, [\Gamma_{ij}[k]]_{L_{ij}})$ 
end

```

different UAVs. We define the path similarity between the j -th and l -th UAVs when observed by the i -th UAV as follows:

$$\sigma_{ijl} \triangleq \frac{1}{L-1} \sum_{m=1}^{L-1} \frac{\mathbf{h}_{ij}^m \cdot \mathbf{h}_{il}^m}{\|\mathbf{h}_{ij}^m\| \|\mathbf{h}_{il}^m\|}, \quad (4)$$

$$\mathbf{h}_{ij}^m = [\mathbf{H}_{ij}]_m - [\mathbf{H}_{ij}]_{m+1}, \quad (5)$$

where $\{\mathbf{h}_{ij}^m\}_{m=1}^{L-1}$ is the recent history of the displacement of the j -th UAV, estimated by the i -th UAV, and $L = \min(L_{ij}, L_{il})$. The path similarity $\sigma_{ijl} \in [-1, 1]$ in (4) is the moving average of the inner product between the estimated displacement vectors of the j -th and l -th UAVs. ■

Note that, when both UAVs move in the same direction during the previous L time instants, then $\sigma_{ijl} = 1$; if they move in opposite directions, then $\sigma_{ijl} = -1$; and, if they move in orthogonal directions, then $\sigma_{ijl} = 0$.

The *path similarity* σ_{ijl} compares two different paths, but it does not provide information about the individual path. To this end, we introduce the following metric.

Definition 2 (Path Persistence): This metric describes variability in the direction of motion of a UAV. We define the *path persistence* of the j -th UAV when observed by the i -th UAV as:

$$\gamma_{ij} \triangleq \frac{1}{L_{ij}-2} \sum_{m=1}^{L_{ij}-2} \frac{\mathbf{h}_{ij}^{m+1} \cdot \mathbf{h}_{ij}^m}{\|\mathbf{h}_{ij}^{m+1}\| \|\mathbf{h}_{ij}^m\|}. \quad (6)$$

The path persistence γ_{ij} is the moving average of the inner product between all consecutive displacement estimations of the j -th UAV observed by the i -th UAV. It measures how much the j -th UAV motion direction has changed recently. ■

When the j -th UAV moves in a straight line, then (6) is maximized and $\gamma_{ij} = 1$. On the other hand, when it moves in a random fashion and the velocity often changes, γ_{ij} will

have lower values. From (6), we can see that the argument of the sum is the inner product between the normalized vectors \mathbf{h}_{ij}^{m+1} and \mathbf{h}_{ij}^m . Thus, $\frac{\mathbf{h}_{ij}^{m+1} \cdot \mathbf{h}_{ij}^m}{\|\mathbf{h}_{ij}^{m+1}\| \|\mathbf{h}_{ij}^m\|} \in [-1, 1]$. Since (6) is the average of these normalized inner products, we obtain $\gamma_{ij} \geq -1$.

For the purpose of the proposed approach, each UAV in the swarm will belong to one of the two categories:

- (i) **Informed UAV** which knows the goal location $\mathbf{g} \in \mathbb{R}^2$ and plans a path to reach it. Because of this information, its trajectory will present, in general, a small number of direction changes (as small as the environment allows), *i.e.*, a high path persistence γ_{ij} .
- (ii) **Uninformed UAV** which does not know the goal \mathbf{g} . It will observe the motion of the other UAVs within the swarm and use it to adaptively move in order to reach the goal. As a consequence, the uninformed UAVs will initially have more irregular motion with many directional changes, resulting in low path persistence, *i.e.*, low values of γ_{ij} .

Finally, we denote \mathcal{I} the set that contains the indices of all the informed UAVs, and $\bar{\mathcal{I}}$ the set that contains the indices of all the uninformed UAVs.

3. Problem Statement

We consider the problem of navigating a UAV swarm, deprived of communication and global localization, in an environment with randomly distributed obstacles. The UAVs start at random locations inside a circle of radius $R^s \in \mathbb{R}_{>0}$. Their mission is accomplished once all the UAVs are inside the disk of radius $R^g \in \mathbb{R}_{>0}$, centered at the goal location \mathbf{g} . Some randomly selected UAVs belong to the informed category described before, while the rest of the UAVs belong to the uninformed category. The UAV categories remain fixed during the whole mission execution.

As mentioned before, we assume that the UAVs are equipped with on-board omnidirectional sensors [33–36] that allow for estimating the position of the surrounding UAVs, as well as detecting their IDs (IDs simplify the separation of multiple observed neighbor UAVs and retrieval of their relative position). We also assume that the individual UAVs are equipped with the necessary sensors to implement Simultaneous Localization and Mapping (SLAM) [37] for localization and navigation.

We aim to design a decentralized control approach for UAVs that only uses on-board sensors and computational resources to complete the collective navigation mission described above. Since the UAVs are deprived of communication, they cannot exchange information like the goal position \mathbf{g} or the category of individual UAVs (informed or uninformed). Thus, the uninformed UAVs must use the observed UAVs' trajectories to devise a motion plan to reach \mathbf{g} . In addition, UAVs must avoid collisions not only with the obstacles populating the environment, but also among themselves.

To this end, this article proposes the decentralized control approach PACNav, which is composed of two modules. The first module, described in Section 4,

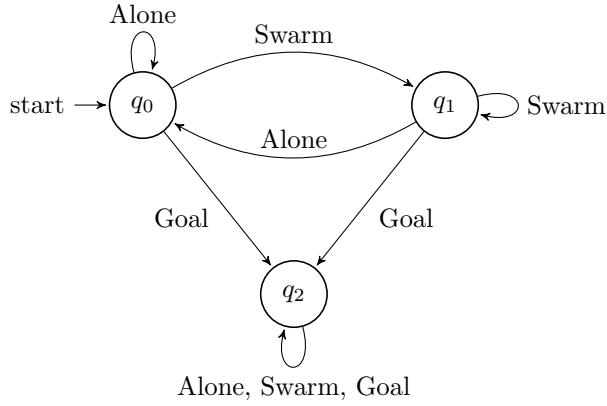


Figure 2: Finite state machine \mathcal{M} for dynamic target selection.

iteratively determines a target to be followed by the UAV. The second module, described in Section 5, controls the velocity of the UAV (\mathbf{u}_i) to reach the target provided by the previous module while avoiding collisions.

4. Iterative Target Selection

At each time instant k , the i -th UAV determines a target location $\mathbf{d}_i[k] \in \mathbb{R}^2$ and devises a path to reach it. This target can be the goal position \mathbf{g} or the position of a neighboring UAV potentially moving towards the goal \mathbf{g} . Thus, moving towards a target UAV can lead the i -th UAV to the goal. This section discusses how to determine the target $\mathbf{d}_i[k]$. Let us start by defining the finite state machine $\mathcal{M} = (\mathcal{S}, \mathcal{Q}, \Delta, q_0, \mathcal{L})$, where:

- $\mathcal{S} = \{\text{Alone, Swarm, Goal}\}$ is the input alphabet,
- $\mathcal{Q} = \{q_0, q_1, q_2\}$ represents the set of all possible states of the state machine \mathcal{M} (illustrated in Fig. 2),
- Δ is the state transition function of \mathcal{M} ,
- q_0 is the initial state, and
- $\mathcal{L} = \emptyset$ is the set of final states. As the machine \mathcal{M} is designed to run indefinitely, there are no final states.

$x_i[k] \in \mathcal{S}$ is the input to the state machine \mathcal{M} of i -th UAV at time k , and it is given by:

$$x_i[k] = \begin{cases} \text{Alone} & , |\mathcal{T}_i[k]| = 0 \wedge i \in \bar{\mathcal{I}}, \\ \text{Swarm} & , |\mathcal{T}_i[k]| > 0 \wedge i \in \mathcal{I}, \\ \text{Goal} & , i \in \mathcal{I}, \end{cases} \quad (7)$$

where $\mathcal{T}_i[k] \subseteq \mathcal{N}_i[k]$ contains the indices of UAVs that are the potential targets of the uninformed i -th UAV. The objective is to design $\mathcal{T}_i[k]$ so that it contains the indices of the UAVs which are potentially moving towards the goal \mathbf{g} . These UAVs are determined based on the following criteria:

- UAVs that are not in close proximity of the i -th UAV. The trajectories of the UAVs close to the i -th UAV are mostly influenced by the collision avoidance mechanism (to avoid collision with the

i -th UAV). Consequently, at that moment, such trajectories have little information about the motion towards a target. So, the i -th UAV discards any UAV that is too close, and it considers only UAVs beyond a certain distance. In other words, the j -th UAV has to satisfy $\|[\mathbf{H}_{ij}[k]]_1 - \mathbf{p}_i[k]\| \geq R^f$, with $R^f \in \mathbb{R}_{>0}$ being a design parameter, to be considered a potential target by the i -th UAV.

- UAVs that are not moving towards the previous target position $\mathbf{d}_i[k-1]$. Since the target is one of the UAVs in $\mathcal{T}_i[k]$, any UAVs moving towards the previous target $\mathbf{d}_i[k-1]$ will not change the current direction of motion of the i -th UAV. Thus, the i -th UAV discards any j -th UAV that satisfies $\|[\mathbf{H}_{ij}[k]]_1 - \mathbf{d}_i[k-1]\| < \|[\mathbf{H}_{ij}[k]]_{L_{ij}} - \mathbf{d}_i[k-1]\|$.
- UAVs whose path history $\mathbf{H}_{ij}[k]$ contains at least three elements. The target is determined using the path persistence metric (γ_{ij}), which needs at least three elements in the path history (see (6)).

The set of potential targets $\mathcal{T}_i[k]$ contains the indices of the UAVs that satisfy all three conditions described above.

Given the current input $x_i[k]$, the state machine \mathcal{M} transitions into a state $s_i[k] \in \mathcal{Q}$. If $s_i[k] = q_2$ (i -th UAV is informed), then the target is the goal position, i.e., $\mathbf{d}_i[k] = \mathbf{g}$. If $s_i[k] = q_0$ (i -th UAV is uninformed and its set of potential targets is empty), then the UAV does not move, i.e., $\mathbf{d}_i[k] = \mathbf{p}_i[k]$. However, if $s_i[k] = q_1$ (i -th UAV is uninformed and its set of potential targets is not empty), then it proceeds to select $\mathbf{d}_i[k]$ using the potential target set $\mathcal{T}_i[k]$. This selection process is based on the path similarity (σ_{ijl}) and persistence (γ_{ij}) metrics defined in Section 2. First, note that the goal \mathbf{g} is the target position for informed UAVs and they are following a path that will lead them towards it. Any changes of direction in their movement is mainly due to the collision avoidance mechanism reacting to the environment. As a result, the informed UAV path history, in general, will present high path persistence. Since all informed UAVs have \mathbf{g} as their target, then their path similarity will also be large. Therefore, if an uninformed UAV follows a UAV which has large path persistence and path similarities with other UAVs, then it would very likely reach the goal \mathbf{g} . Thus, the target $\mathbf{d}_i[k]$, when $s_i[k] = q_1$, is obtained as:

$$\mathbf{d}_i[k] = \left[\mathbf{H}_{ij^\star}[k] \right]_1, \quad (8)$$

with

$$j^\star = \operatorname{argmax}_{j \in \mathcal{T}_i[k]} \left(\gamma_{ij} + \sum_{l \in \mathcal{T}_i[k] \setminus j} \sigma_{ijl} \right), \quad (9)$$

where the first term in (9) is the path persistence (γ_{ij}) of the j -th candidate UAV, and the second term is the sum of path similarities between the path of j -th UAV and the rest of the potential UAV targets.

The selection heuristic presented in (8) is based on the following observation. Initially, when the swarm starts to move, only the informed UAVs will have high path persistence. As the uninformed UAVs select the informed ones as their target, the path similarity between the UAVs

will increase. Additionally, as the uninformed UAVs keep moving towards the informed ones, their path persistence will also increase. As a result, the number of UAVs with high path similarity and persistence increases over time. Thus, the uninformed UAVs can select other uninformed UAVs as targets and do not necessarily need the information about the path history of informed UAVs. This is crucial when the number of UAVs in the swarm is large and all uninformed UAVs cannot observe the informed ones. Experimental results in Section 6 verify and demonstrate this selection process. To summarize, we have:

$$\mathbf{d}_i[k] = \begin{cases} \mathbf{p}_i[k], & s_i[k] = q_0, \\ [\mathbf{H}_{ij^*}[k]]_1, & s_i[k] = q_1, \\ \mathbf{g}, & s_i[k] = q_2. \end{cases} \quad (10)$$

5. Target Tracking

After selecting the target, *i.e.*, $\mathbf{d}_i[k]$, the i -th UAV constructs and follows a path to the target while avoiding collisions with the obstacles in the environment and with the surrounding UAVs. This is achieved by controlling the UAV velocity, *i.e.*, the control input \mathbf{u}_i in (1), as:

$$\mathbf{u}_i[k] = \mathbf{n}_i[k] + \mathbf{c}_i[k], \quad (11)$$

where $\mathbf{n}_i[k]$ is the navigation control vector that will move the i -th UAV towards the desired target $\mathbf{d}_i[k]$, and $\mathbf{c}_i[k]$ is the collision control vector that provides the i -th UAV with collision avoidance capability. In the rest of this section, we will explain the process from the i -th UAV perspective, and to lighten the notation, we will drop the subscript i from $\mathbf{n}_i[k]$ and $\mathbf{c}_i[k]$ variables.

5.1. Navigation control vector

The navigation control vector $\mathbf{n}[k]$ directs the UAV towards the target position $\mathbf{d}_i[k]$. This vector is obtained from the shortest collision-free path from the UAV position ($\mathbf{p}_i[k]$) to the target ($\mathbf{d}_i[k]$). To compute this path, we model the environment as a 2-dimensional grid, resulting in a discretization of the continuous space in \mathbb{R}^2 . Each grid coordinate either belongs to a set of occupied (obstacles) points \mathcal{C}^o or free points \mathcal{C}^f . The occupied points are obtained by creating an online map of the environment. This grid represents a weighted undirected graph \mathcal{G} with edges connecting coordinates to their neighbors, as shown in Fig. 3. The graph \mathcal{G} has an edge between the generic coordinates (a_x, a_y) and (b_x, b_y) if, and only if, both coordinates belong to the free set \mathcal{C}^f . The weight matrix \mathbf{W} for the coordinates in the set \mathcal{C}^f is defined as:

$$[\mathbf{W}]_{ab} = \begin{cases} \sqrt{2}, & b_x = a_x \pm 1, b_y = a_y \pm 1, \\ 1, & b_x = a_x \pm 1, b_y = a_y, \\ 1, & b_x = a_x, b_y = a_y \pm 1. \end{cases} \quad (12)$$

The shortest path problem is formulated on the graph \mathcal{G} with weights given by \mathbf{W} . To find the shortest path from the UAV position $\mathbf{p} = (p_x, p_y)$ to the target position

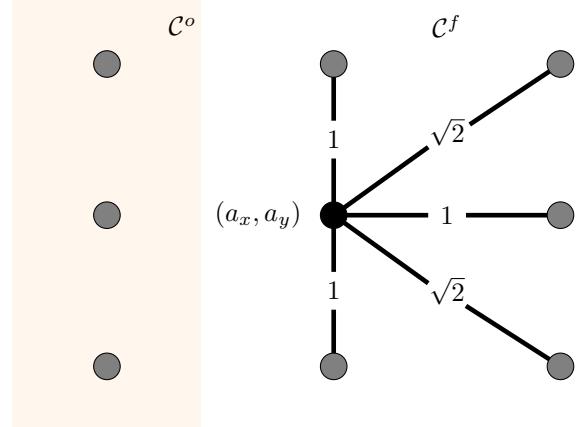


Figure 3: Graph representation of the continuous space. The edges (b_x, b_y) are shown for the node at coordinate (a_x, a_y) and depicted in gray. The free (\mathcal{C}^f) and occupied (\mathcal{C}^o) sets are represented using white and yellow colors, respectively.

$\mathbf{d} = (d_x, d_y)$ on \mathcal{G} , we use the A* algorithm [38] with the following heuristic:

$$h(a_x, a_y) = \sqrt{(a_x - d_x)^2 + (a_y - d_y)^2}. \quad (13)$$

The simultaneous mapping and planning needed for creating the graph is achieved using the methods described in [20]. The shortest path \mathcal{P} (this path is collision-free since the graph \mathcal{G} does not have any edges connecting the occupied coordinates in the set \mathcal{C}^o) generated by the A* algorithm is an ordered set of waypoints connecting the i -th UAV position $\mathbf{p}_i[k]$ to the target point $\mathbf{d}_i[k]$, whose closest waypoint to the current UAV position is denoted with the vector $\mathbf{a}_n \in \mathcal{P}$. Therefore, the navigation vector $\mathbf{n}[k]$ is given by:

$$\mathbf{n}[k] = f_{ig} \mathbf{n}_I + \bar{f}_{ig} \mathbf{n}_U, \quad (14)$$

where $f_{ig} = 1$ if, and only if, the i -th UAV has goal information, while \bar{f}_{ig} is its complementary function. So, \mathbf{n}_I is the navigation vector if the UAV is informed; otherwise, the navigation vector is \mathbf{n}_U .

As discussed in Section 4, the target selection process relies on the high path persistence of the informed UAVs. Thus, these UAVs must always move with a finite velocity towards their target. However, in order to remain in the LoS of other UAVs, the informed UAVs must slow down for the swarm to catch up. Thus, the navigation vector for informed UAVs is given as:

$$\mathbf{n}_I = \max \left(V^m, 1 - \frac{\sum_{j \in \mathcal{N}_i} \|\tilde{\mathbf{p}}_{ij} - \mathbf{p}_i\|}{2R^f |\mathcal{N}_i|} \right) K^n (\mathbf{a}_n - \mathbf{p}_i), \quad (15)$$

where $V^m \in (0, 1)$ is the minimum normalized velocity of the informed UAV, and $K^n \in \mathbb{R}$ is a scaling coefficient to rescale the position vector to form the velocity control input (\mathbf{u}). The second term in the max function depends on the average distance from the UAVs in \mathcal{N}_i . The magnitude of

Algorithm 3: Regulating navigation vector

Data: $\mathcal{N}_i, \mathbf{a}_n, \mathbf{p}_i$
Result: \mathbf{n}_U

$$\begin{aligned} \mathbf{n}_U &\leftarrow K^n(\mathbf{a}_n - \mathbf{p}_i) \\ \text{for } j \in \mathcal{N}_i \text{ do} \\ &\quad /* \text{ projection parallel to } \tilde{\mathbf{p}}_{ij} - \mathbf{p}_i */ \\ &\quad \mathbf{s} \leftarrow \frac{\mathbf{n}_U \cdot (\tilde{\mathbf{p}}_{ij} - \mathbf{p}_i)}{\|\tilde{\mathbf{p}}_{ij} - \mathbf{p}_i\|^2} (\tilde{\mathbf{p}}_{ij} - \mathbf{p}_i) \\ &\quad /* \text{ projection orthogonal to } \tilde{\mathbf{p}}_{ij} - \mathbf{p}_i */ \\ &\quad \mathbf{o}^\perp \leftarrow \mathbf{n}_U - \mathbf{s} \\ &\quad \mathbf{n}_U \leftarrow \min \left(1, \left(\frac{\|\tilde{\mathbf{p}}_{ij} - \mathbf{p}_i\|}{R_f} \right)^\alpha \right) \mathbf{s} + \mathbf{o}^\perp \\ \text{end} \end{aligned}$$

vector \mathbf{n}_I decreases as this average distance increases, which prevents the informed UAV from wandering far away from the swarm. The max function ensures that the UAV always moves with a minimum velocity of V^m .

Unlike the informed UAVs, the target of uninformed UAVs is the position of other UAVs in the swarm. Hence, the navigation vector must be designed to prevent collisions when moving towards other UAVs. The vector \mathbf{n}_U for uninformed UAVs is obtained as described in Algorithm 3. The projection parallel to the relative position vector $(\tilde{\mathbf{p}}_{ij} - \mathbf{p}_i)$ is iteratively scaled down if $\|\tilde{\mathbf{p}}_{ij} - \mathbf{p}_i\|$ is less than parameter R^f . This scaling ensures that the navigation vector will have a smaller component towards the j -th UAV as the i -th UAV moves closer to it, which is essential to prevent collisions. The parameter $\alpha \in \mathbb{R}_{>0}$ determines the rate of change of the scaling term. The orthogonal component \mathbf{o}^\perp is not scaled; thus, the motion in direction orthogonal to $(\tilde{\mathbf{p}}_{ij} - \mathbf{p}_i)$ remains unaffected.

5.2. Collision control vector

The obstacles in the environment pose a great challenge when moving towards the target. The collision free path \mathcal{P} used to obtain the navigation vector is safe in an ideal case, but it can fail to prevent collisions in the presence of motion and sensor uncertainties. As the environment is unknown, it is critical for the UAV to react to obstacles as soon as they are detected by the sensors. Thus, reactive collision avoidance is essential to ensure safety of the UAV. However, the UAVs do not need to avoid all detected obstacles, but only those that pose an immediate threat of collision. We define with $\mathcal{O}_i[k]$ the set containing the position of all the obstacles (*e.g.*, trees, other UAVs) whose distance to \mathbf{p}_i at time instant k is smaller than the reaction distance $R^o \in \mathbb{R}_{>0}$. Increasing the value of R^o would result in the UAV avoiding even far away obstacles. Thus R^o is a design parameter that needs to be tuned according to the obstacle density and the number of UAVs that make up the swarm.

The reactive collision avoidance method used by several state-of-the-art real-world systems [20, 39] often suffers from deadlocks. Some examples are shown in

Fig. 4a. To prevent such deadlocks, we propose a novel method to move away from the obstacles while avoiding the deadlocks as much as possible. Unlike [39], the proposed collision avoidance vector has components in both parallel and orthogonal directions relative to the position vector $(\mathbf{p}_i - \mathbf{o}_r)$, where $\mathbf{o}_r \in \mathbb{R}^2$ is the position of the obstacle considered. This collision vector is depicted in Fig. 4b. The parallel component takes the UAV away from the obstacle, while the orthogonal component moves it tangentially to the circle of radius $\|\mathbf{p}_i - \mathbf{o}_r\|$ and centered at \mathbf{o}_r . The combined parallel and orthogonal motions not only move the UAV away from the obstacle, but also around it. This allows for avoiding deadlocks, such as those shown in Fig. 4c.

The collision avoidance vector for an obstacle $\mathbf{o}_r \in \mathcal{O}_i[k]$ is obtained by first generating two candidate unit vectors $\hat{\mathbf{c}}^+$ and $\hat{\mathbf{c}}^-$ as:

$$\hat{\mathbf{c}}^+ = \mathbf{R}(+\varphi) \frac{(\mathbf{p}_i - \mathbf{o}_r)}{\|\mathbf{p}_i - \mathbf{o}_r\|}, \quad (16a)$$

$$\hat{\mathbf{c}}^- = \mathbf{R}(-\varphi) \frac{(\mathbf{p}_i - \mathbf{o}_r)}{\|\mathbf{p}_i - \mathbf{o}_r\|}, \quad (16b)$$

where $\mathbf{R}(\pm\varphi) \in \mathbb{R}^{2 \times 2}$ denotes the rotation matrix along the z -axis of the world frame \mathcal{F}_W , which is used to rotate the relative vector $(\mathbf{p}_i - \mathbf{o}_r)$ by angle $\pm\varphi$ in the *counterclockwise* direction. The angle φ is given as:

$$\varphi = \frac{\pi}{2R^o} \|\mathbf{p}_i - \mathbf{o}_r\|. \quad (17)$$

The angle φ varies with the distance to the obstacle. As a result, $\hat{\mathbf{c}}^+$ and $\hat{\mathbf{c}}^-$ have large components parallel to $(\mathbf{p}_i - \mathbf{o}_r)$ when the UAV is close to the obstacle. This large parallel component makes the reactive collision avoidance more focused in the parallel direction, thus preventing collisions. However, as the UAV moves away, φ increases and $\hat{\mathbf{c}}^+$ and $\hat{\mathbf{c}}^-$ have larger orthogonal components. The orthogonal components focus on moving the UAV around the obstacle.

The vectors $\hat{\mathbf{c}}^+$ and $\hat{\mathbf{c}}^-$ denote two possible directions of motion to avoid collision with the obstacle \mathbf{o}_r (see Fig. 4b). In order to keep the motion smooth, the vector with the least angular distance to the previous control input $\mathbf{u}_i[k-1]$ is used as the collision avoidance vector \mathbf{c}_r for obstacle at \mathbf{o}_r . Thus, the vector \mathbf{c}_r is obtained as:

$$\mathbf{c}_r = \max \left(0, \frac{1}{\|\mathbf{p}_i - \mathbf{o}_r\|} - \frac{1}{R^o} \right) \hat{\mathbf{c}}_r, \quad (18)$$

with

$$\hat{\mathbf{c}}_r = \underset{\hat{\mathbf{b}} \in \{\hat{\mathbf{c}}^+, \hat{\mathbf{c}}^-\}}{\operatorname{argmax}} \left(\frac{\hat{\mathbf{b}} \cdot \mathbf{u}_i[k-1]}{\|\mathbf{u}_i[k-1]\|} \right), \quad (19)$$

where $\hat{\mathbf{c}}_r$ is a unit vector. The magnitude of \mathbf{c}_r is inversely proportional to the relative distance to the obstacle. Thus, the UAV reacts more strongly to nearby obstacles in comparison to farther ones. Note that, \mathbf{c}_r is a function of the obstacle position $\mathbf{o}_r \in \mathcal{O}_i[k]$. The collision control vector $\mathbf{c}[k]$ (see (11)) is a superposition of collision avoidance vectors of all the obstacles in $\mathcal{O}_i[k]$ and is obtained as:

$$\mathbf{c}[k] = K^c \sum_{\mathcal{O}_i[k]} \mathbf{c}_r, \quad (20)$$

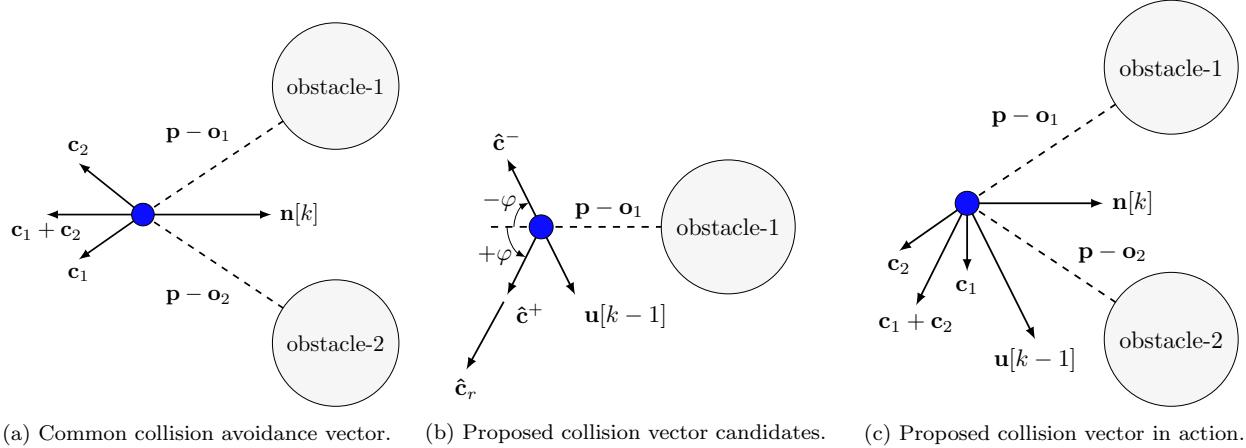


Figure 4: Collision avoidance vectors. The UAV is denoted in blue and the obstacles in gray. Fig. (4a) Common collision avoidance vector with components parallel to the obstacle position vector. Fig. (4b) Collision avoidance candidates described in (16a) and (16b). Fig. (4c) Proposed collision avoidance vector, as described in (20), resulting from the interaction with two obstacles.

where K^c is a scaling coefficient to rescale the summation. The preventive collision avoidance incorporated in $\mathbf{n}[k]$, along with the reactive collision avoidance from $\mathbf{c}[k]$, makes the control input safe even in the presence of imperfect sensor data.

6. Simulations and Experiments

In this section, we analyze the proposed collective navigation approach PACNav, as presented in Sections 4 and 5. First, we evaluate the swarm behavior using realistic simulations in Gazebo [40], exploiting the advantages of software-in-the-loop simulations [41]. Then, we present the results of real-world experiments carried out in a natural forest. Videos with the simulated and real-world experiments are available at <http://mrs.felk.cvut.cz/pacnav>, while the source code has been made available as open-source‡.

We introduce the *order metric* $\Omega[k]$ [42] that will be used to analyze the collective motion of the swarm as:

$$\Omega[k] = \frac{1}{N(N-1)} \sum_{i,j \in \mathcal{N}_i} \frac{\mathbf{v}_i[k] \cdot \mathbf{v}_j[k]}{\|\mathbf{v}_i[k]\| \|\mathbf{v}_j[k]\|}, \quad (21)$$

where $\mathbf{v}_i[k]$ is the instantaneous velocity of the i -th UAV. This metric captures the correlation between the movements of the agents and provides an indication about the overall alignment of the member UAVs. The value of $\Omega \in [-1, 1]$, where $\Omega = 1$, means that all members of the swarm are moving in the same direction. $\Omega < 1$ implies misalignment between the UAVs. Figure 5 illustrates the order between two agents.

6.1. Simulated experiments

We simulate four different cases: (case 1A) a swarm composed of $N = 3$ UAVs with only one informed UAV; (case 1B) a swarm composed of $N = 3$ UAVs with two informed UAVs; (case 2A) a swarm composed of $N = 6$ UAVs with two informed UAVs; and (case 2B) a swarm

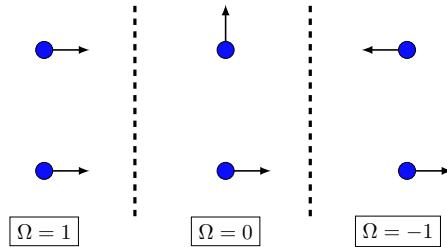


Figure 5: Illustrative scenarios of the *order metric* (21) for two UAVs. UAVs are depicted in blue, while arrows denote their direction.

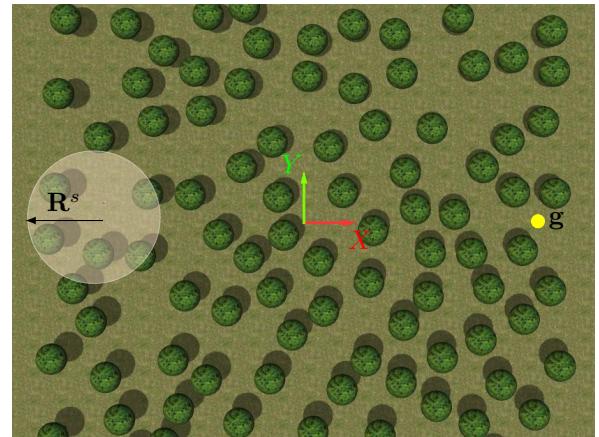


Figure 6: Simulated forest in the Gazebo simulator. The UAVs are initialized in the shaded region, while the goal is marked as the yellow dot. The red and green arrows denote the X - and Y -axis of the reference system \mathcal{F}_W , respectively.

composed of $N = 6$ UAVs with four informed UAVs. For each case scenario, we run 10 simulated experiments by varying the spatial distribution of the trees in the

simulated natural forest, while initializing the UAVs in the same shaded region and using the same fixed goal \mathbf{g} . A realization of the overall scenario is depicted in Fig. 6. Such an approach allowed us to statically characterize the behavior of the swarm with respect to the changes in the environment.

The density of the forest with randomly distributed trees over an area A can be described as:

$$\rho = \frac{N^t \pi (R^o)^2}{A}, \quad (22)$$

where N^t is the number of trees in the area A . We selected a forest of dimensions $50\text{ m} \times 50\text{ m}$ with density of $\rho = 0.4$. This density allows six UAVs to simultaneously navigate through the forest.

After trying various simulation setups, we observed that densities larger than $\rho = 0.6$ make navigation in the forest extremely difficult. Due to the high number of trees, the UAVs are trying to avoid collisions most of the time. Thus, the effect of the navigation vector $\mathbf{n}[k]$ in (11) is overtaken by the collision avoidance vector $\mathbf{c}[k]$.

The task objective consists of navigating a swarm composed of N -UAVs from a random initial position (inside the shaded circle of radius R^g in Fig. 6) to the goal \mathbf{g} , while trying to keep the mean distance between UAVs below $2R^f$ most of the time. The experiment is complete once all the N -UAVs are within a R^g distance from the goal position. Figures 7-10 present the results for the case 1A, case 1B, case 2A, and case 2B, respectively. The plots in each figure are arranged according to the increase completion time of the experiment.

As described in Section 4, only the UAVs farther than R^f are considered as potential targets by uninformed UAVs. Thus, to complete the mission, the uninformed UAVs must remain at a distance larger than R^f from the informed ones. This is one of the reasons why the mean distances between UAVs in the swarm are larger than R^f in Fig. 7a, Fig. 8a, Fig. 9a, and Fig. 10a. We also note that the minimum distance between UAVs is sometimes slightly smaller than R^o , which is when the reactive collision avoidance mechanism of the UAVs becomes active.

Comparing case 1A and case 1B, we observe that as the number of informed UAVs increases, the mean completion time is reduced from 212.4s to 189.5s. This is due to the fact that as the number of informed UAVs is increased, it becomes more likely that an uninformed UAV directly tracks an informed UAV. Similarly, increasing the number of informed UAVs from case 2A to case 2B also reduces the completion time from 231.4s to 213.3s. We also observe that as the number of UAVs are increased from three to six, the maximum and minimum distance between grows larger (see Fig. 7a, Fig. 8a, Fig. 9a, and Fig. 10a). This is because the UAVs need to avoid collisions more often when there are more UAVs moving around. This slows down their motion which leads to some of them lagging behind the others. However, the proposed approach successfully navigates the swarm in all the experiments with three and even six UAVs.

Let us focus now on the order metric (21) plotted in Fig. 7b, Fig. 8b, Fig. 9b, and Fig. 10b. This metric measures the mean alignment between the velocity

Sym.	Value	Sym.	Value	Sym.	Value
R^f	4.0 m	R^o	2.5 m	N^t	104
K^c	1.0 s^{-1}	K^n	1.2 s^{-1}	\mathbf{g}	$(20\text{ m}, 0\text{ m})^\top$

Table 1: List of parameters and their values for the simulated experiments.

of UAVs [42]. When all UAVs move in the same direction, the order takes values close to 1. However, when the UAVs move in a *disordered* manner and each UAV moves in a different direction, then the order metric takes lower values. From the plots, we observe that, in general, the order profile behaves in the following manner. In the first stage, the uninformed UAVs are trying to figure out where to go and which UAV to follow. Here, the order value is low, but as they start to follow informed UAVs (or other uninformed UAVs which are already following informed UAVs), the order starts to increase. When all the uninformed UAVs are *locked* and are tracking the informed UAVs, the order metric remains almost constant for some time. At this stage all the UAVs are moving, more or less, in the same direction at a similar speed. In the next stage, when the UAVs start to reach the goal, the UAVs lose alignment and the order metric decreases. This is due to the fact that, at the end of the experiment, the UAVs slow down and move in different directions to avoid collisions. Finally, when most of the UAVs have already reached the goal, but some UAVs are lagging behind due to collision avoidance with the trees, we observe a fourth stage. In this stage, the order metric remains almost constant at low values, as seen in the last three subplots in Fig. 7b, Fig. 8b, Fig. 9b, and Fig. 10b. Table 1 reports the values of the parameters used for the simulated experiments.

6.2. Real-world experiments

After validating the proposed approach in the simulated experiments, we performed a real-world experiment in a natural forest, as shown in Fig. 13. The complex and random arrangement of trees in the forest serve as a challenging environment to validate the feasibility and robustness of the proposed approach. The forest had a density of $\rho = 0.25$ (as calculated using (19)), and the UAVs were initialized at arbitrary positions with $R^s = 3.0\text{ m}$. The swarm was composed of four quadrotor UAV platforms (`uav1`, `uav2`, `uav3`, and `uav4`), each with a diameter of 0.5 m (the dimensions of a quadrotor can be approximated with a circle in the XY-plane) and a mass of 2.6 kg . The UAV platforms were based on the DJI F450 quadrotor, equipped with an Intel NUC on-board computer (an i7-8559U processor with 16GB of RAM) and the Pixhawk flight controller. The software stack [43] was built on the Noetic Ninjemys release of the Robot Operating System (ROS) running on Ubuntu 20.04. The UAVs were also equipped with a RPLIDAR rotary rangefinder for SLAM [37], a Garmin laser rangefinder used as an altimeter, and the UVDAR system [34] for direct relative localization of UAVs. Further details on the hardware setup can be found in [44, 45]. Fig. 11 shows the sensory equipment layout on-board the UAVs.

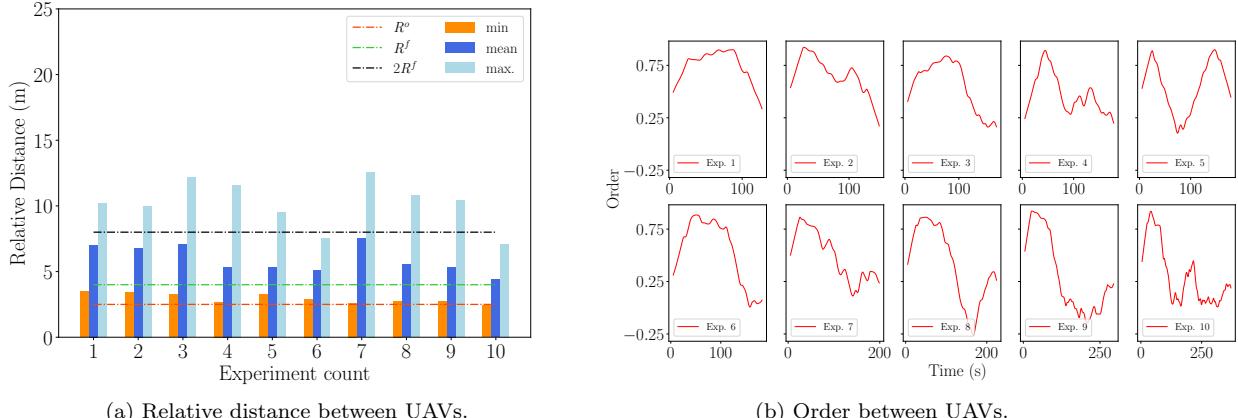


Figure 7: Relative distance and Order between UAVs in 10 independent simulated experiments with three randomly initialized UAVs, where one UAV had goal information (case 1A) with $R^s = 3.0 and $R^g = 6.0.$$

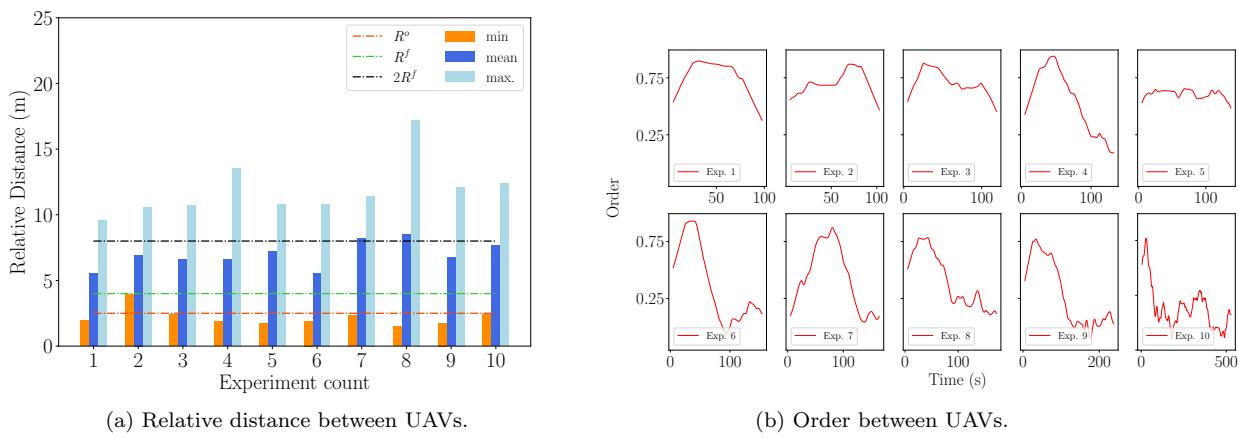


Figure 8: Relative distance and Order between UAVs in 10 independent simulated experiments with three randomly initialized UAVs, where two UAVs had goal information (case 1B) with $R^s = 3.0 and $R^g = 6.0.$$

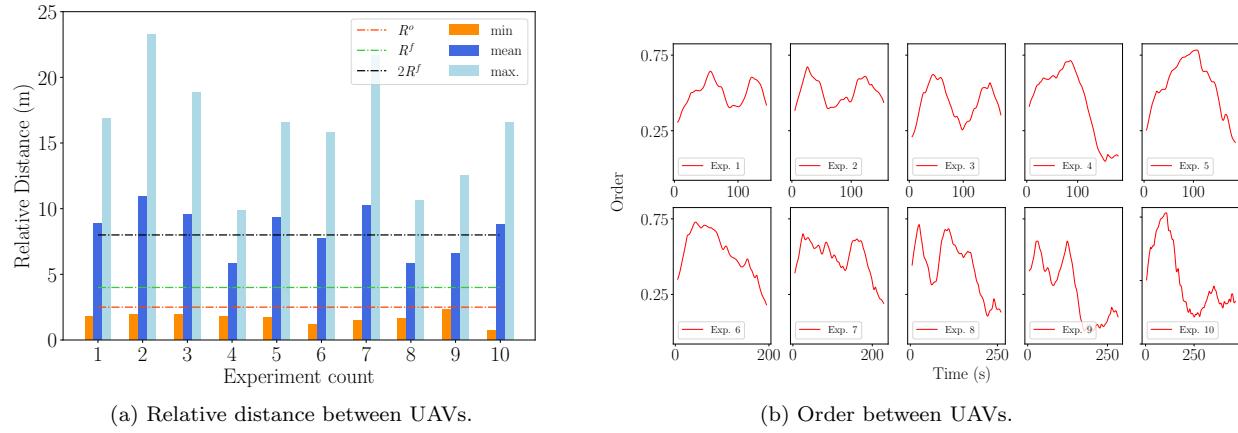
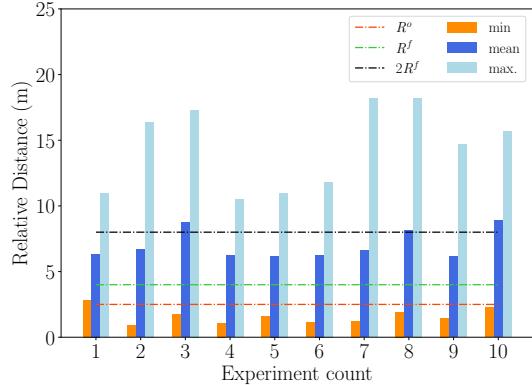


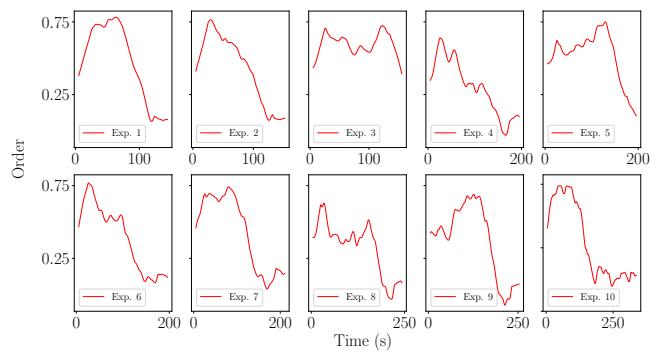
Figure 9: Relative distance and Order between UAVs in 10 independent simulated experiments with six randomly initialized UAVs, where two UAVs had goal information (case 2A) with $R^s = 4.5 and $R^g = 8.5.$$

We selected **uav3** as the informed UAV and assigned the goal at $\mathbf{g} = (0\text{ m}, 40\text{ m})^\top$ before the start of the

experiment. The rest of the UAVs were uninformed. Since only one of the UAVs is informed, the experiment



(a) Relative distance between UAVs.



(b) Order between UAVs.

Figure 10: Relative distance and Order between UAVs in 10 independent simulated experiments with six randomly initialized UAVs, where four UAVs had goal information (case 2B) with $R^s = 4.5$ m and $R^g = 8.5$ m.

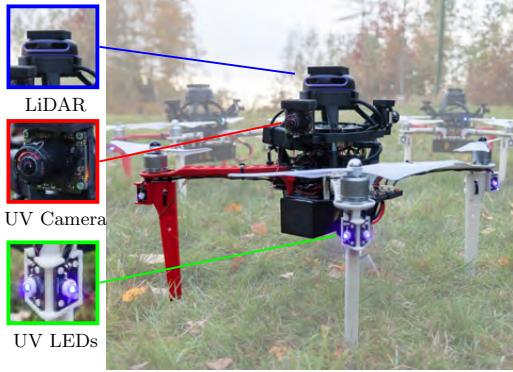


Figure 11: Quadrotor platform highlighting the UV LEDs and the UV cameras used by the UltraViolet Direction And Ranging (UVDAR) system for direct relative localization of UAVs, and the 2D Light Detection and Ranging (LiDAR) for SLAM.

constitutes the worst case scenario, which is useful to test the limits of the proposed approach.

Figure 12 shows the trajectory of all four UAVs in the forest, while Fig. 13 reports video snapshots from the real-world experiment. The marks of the UAVs in Fig. 12 correspond to the instants 0 s, 105 s, 180 s, and 300 s. At the beginning of the experiment (0-105 s), all UAVs except for **uav3**, move around to prevent any collisions with other UAVs and surrounding trees. As shown in Fig. 14, the UAVs target each other at the beginning, but when their motion stabilizes, they select **uav3** as the target. As more and more UAVs select **uav3** as their target, the swarm motion becomes more directed towards the goal. After 150 s, all of the UAVs have selected and are tracking **uav3** as their target. The swarm then reaches its goal at 300 s.

As shown in Figures 12 and 13, the UAVs move through the forest as a cohesive group, while avoiding obstacles and each other. Fig. 15 shows the relative distance of the swarm as observed by each UAV. As can be

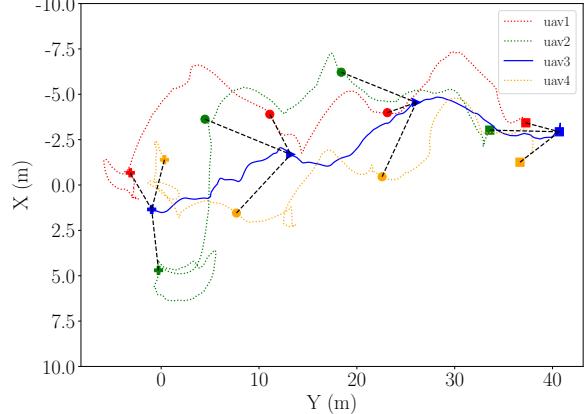


Figure 12: Recorded trajectory from the real-world flight experiment in the forest. The solid line represent the trajectory of the **uav3** (informed about the goal), while dashed lines represent the trajectories of all other UAVs (**uav1**, **uav2**, and **uav4**). Triangle and circle marks depict the position of the UAVs at time instants 105 s and 180 s, while plus and square marks show the initial (0 s) and final (300 s) positions of the UAVs, respectively.

seen from the figure, after the initial 105 s, the mean relative distance for **uav1** and **uav4** decreases rapidly as they select the informed UAV as their target. Since the mean relative distance is larger than R^f for a significant part of the experiment, the uninformed UAVs were able to select **uav3** as their target (see Fig. 14). Similarly, the order between the UAVs also increases rapidly after the initial 105 s (see Fig. 16). Note that, the order for all UAVs increases to a value above 0.5 after the initial 105 s and remains so until 250 s. As the UAVs reach the goal near the end of the experiment, they slow down and just move around the goal position. This reduces the order drastically near the end of the experiment. This behavior can also be observed in

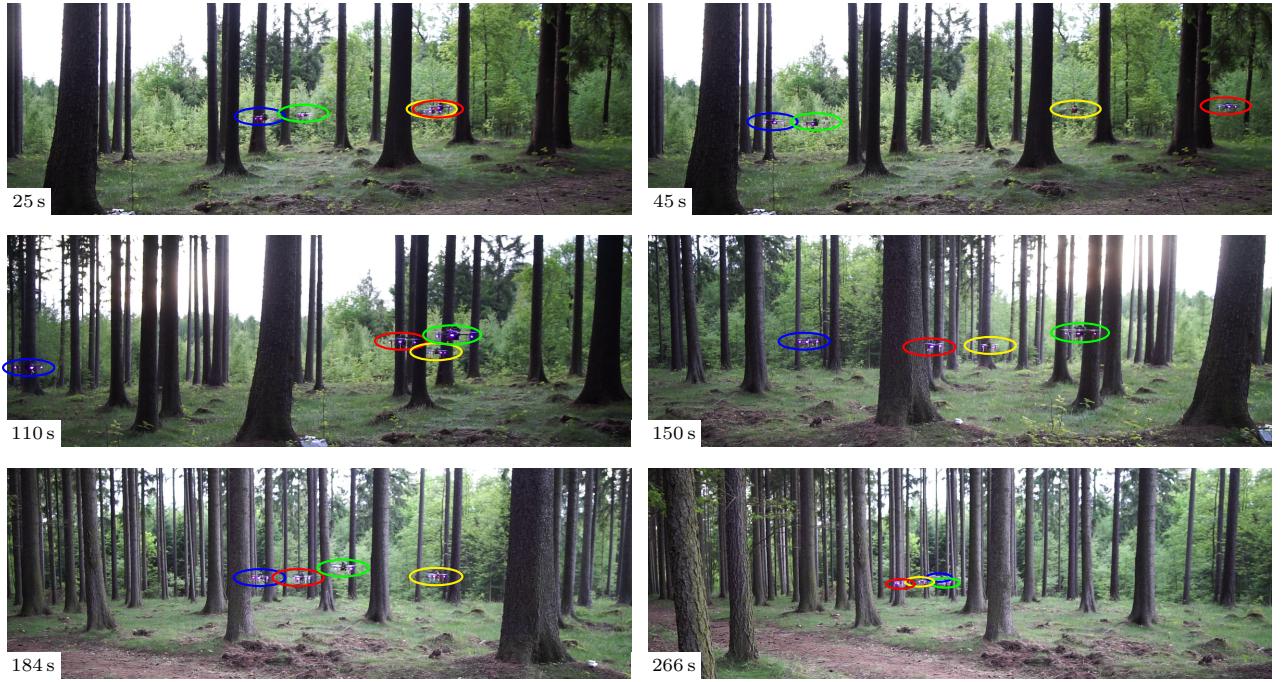


Figure 13: Video snapshots from the real-world experiment. The system evolution at different time instances is reported. Colored solid circles highlight the UAV positions and their IDs according to the legend in Fig. 12.

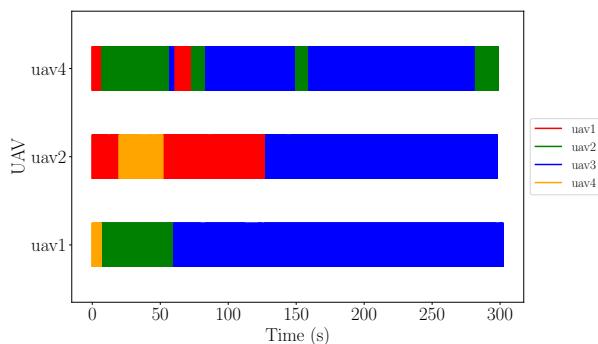


Figure 14: Target selected and tracked by each UAV during the real-world experiment in the forest. Colored lines represent the target (another UAV) selected by a particular UAV.

the supplementary multimedia material. Since the order is a measure of the alignment degree of the swarm, it is apparent that the UAVs were moving in the same direction during the experiment. This successfully demonstrates that the proposed approach can be used for collective navigation, even in the worst case scenario presented in this experiment.

The large number of trees and continuous reactive motion of other members of the swarm made the navigation challenging. While moving through the environment, UAVs were often occluded by trees or other UAVs. Since the UVDAR localization system uses cameras for relative localization, these occlusions affected its estimation accuracy. This is visible in Fig. 15 where the

estimated distance between UAVs has a high variance and often jumps by large values. The estimation inaccuracy also results in irregularity in the order (measure of velocity alignment) of the UAVs, as shown in Fig. 16. However, as the proposed path similarity and persistence metrics (see Section 2) depend on path history rather than single position estimates, the experiments show that proposed approach successfully allows the UAVs to navigate in this challenging environment.

7. Discussion

7.1. 2D vs 3D implementation

Let us discuss the difference between the 2D and 3D implementation of the proposed approach. This article focuses on a 2D implementation of PACNav, although a 3D implementation would be straightforward, providing the UAVs an additional dimension for motion and further facilitating collision avoidance. A 3D implementation of the method herein would only require expansion of dimensions for all the vector quantities. This change in dimension would result in a slight increase in the calculations involving vectors, but have no effect on the overall method. However, a 3D representation of the environment will be needed for SLAM and obstacle avoidance. This 3D representation would require heavy sensors and computationally expensive mapping algorithms. Moreover, 3D implementation of the A^* algorithm used for planning (Section 5.1) would significantly increase the memory and computational load of the overall system when running onboard the UAV.

The addition of heavy sensors and more computational power would increase the size of the UAV, making it

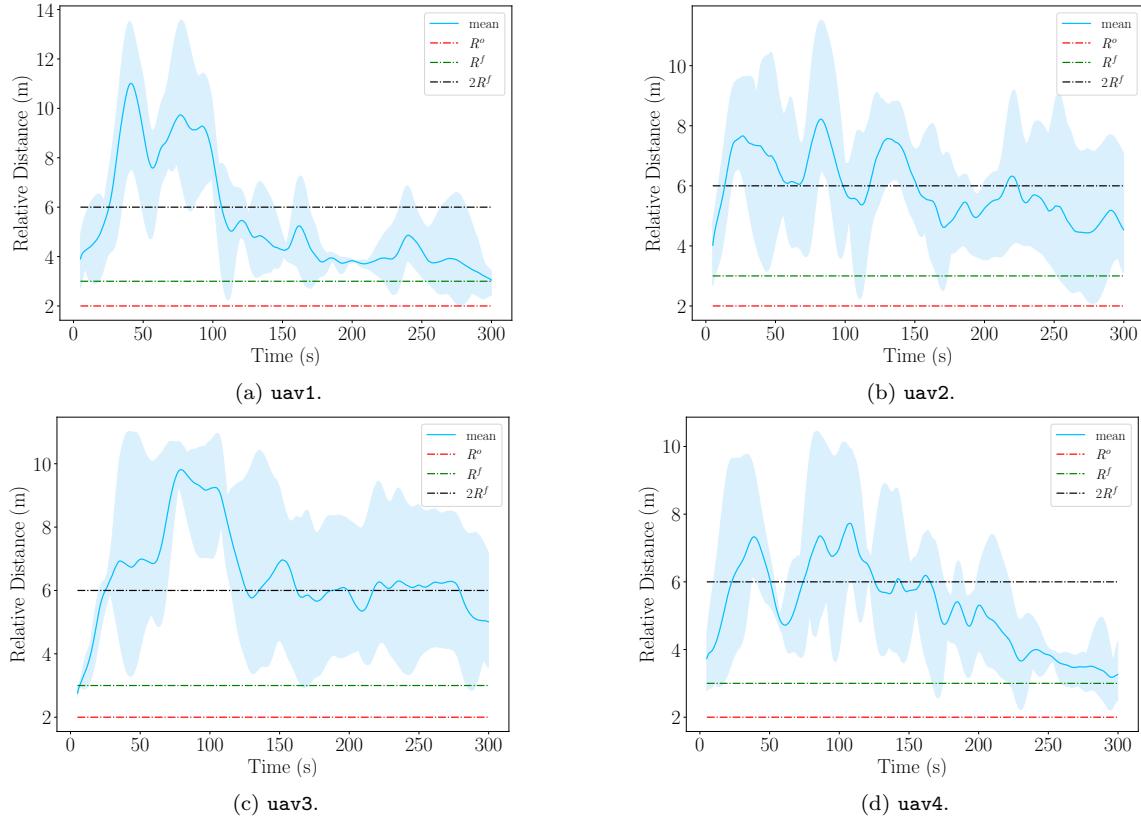


Figure 15: Recorded relative distances between the UAVs in the real-world experiment. The distance is calculated using the position estimate from the UVDAR direct localization system. The shaded region contains all the distance measurements and the solid line represents their mean at any given time instant.

difficult to operate in obstacle-rich environments, such as forests. The 2D implementation of the PACNav approach is sufficient for solving the navigation problem in many practical scenarios (some presented in this article) while also being practically feasible. Thus, it is unclear whether the extra computational load and energy requirements for 3D implementation would justify its use.

7.2. Scalability

In a scenario free of obstacles, there would be a consistent LoS between most of the UAVs throughout the entire mission. Therefore, each UAV would simply react to the motion of the other UAVs. However, in cluttered environments, the UAVs often loose LoS due to occlusions from environmental obstacles and other UAVs. As a result, each UAV may only use local information (from UAVs in LoS) to select a target. For a fixed number of informed UAVs, increasing the number of uninformed UAVs would mean that several UAVs do not have LoS with any informed UAV. This would adversely affect the collective navigation of the swarm.

For the forest simulated in our experiments, with a group size greater than six UAVs, we observed splitting of the swarm into sub-groups. Due to the loss of the target by some of the uninformed UAVs, the UAVs farther away from the informed ones were thus unable to move towards the goal, and consequently unable to guide the

other uninformed UAVs to the goal. Moreover, the uninformed UAVs were more likely to get stuck or left behind, as they did not have an appropriate target to follow. Our experiments suggest that splitting can be avoided by composing the swarm with approximately 60 percent informed UAVs.

7.3. Design parameters

The design parameters of the PACNAV approach are R^f , R^o , K^n , K^c , K^m , and K^p . In the following section, we briefly discuss their purpose and effect on the PACNAV approach.

- R^f is used to discard potential target UAVs that are too close. The i -th UAV would consider the j -th UAV as a potential target only if it is farther than the distance R^f .

If R^f is too small, most of the UAVs close to the i -th UAV are considered potential targets. Since these UAVs are close to each other, their trajectory is significantly influenced by the collision avoidance mechanism that increases the distance between the UAV. As a result, the UAVs will not exhibit the guiding behavior that arises when the UAVs are following a target. Thus, having a small R^f in a cluttered environment can lead to frequent changes in the choice of target UAV, as motion due to collision avoidance will result in high path persistence

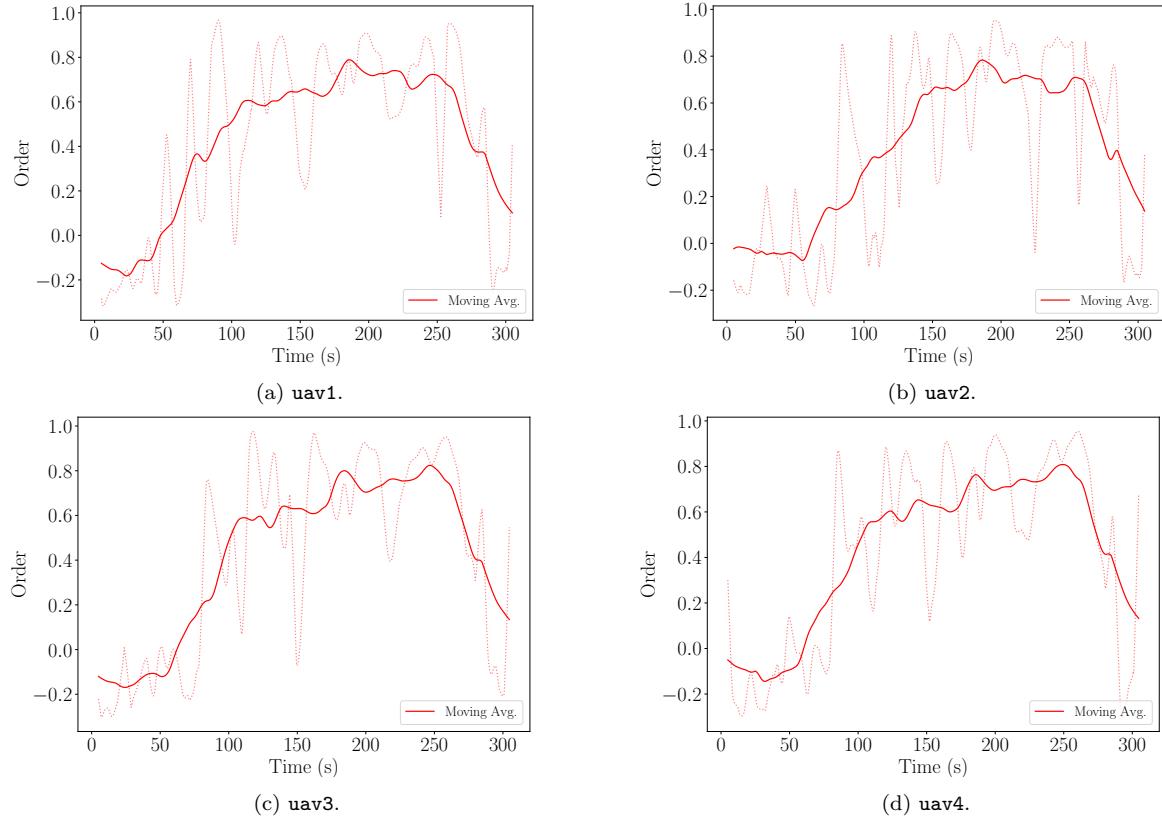


Figure 16: Recorded Order between the UAVs during the real-world experiment. The dashed and solid lines represent the recorded order and the simple moving average, respectively.

for most of the potential targets. This rapid change slows down the collective movement of the swarm while the UAVs try to follow the collision avoidance motion, rather than the goal-directed motion of the informed UAVs.

Alternatively, if R^f is too large, then the i -th UAV might not be able to find any UAV as a potential target, and the algorithm may fail.

- The parameter R^o describes the threshold distance when reacting to the obstacles (18).

If $R^o \rightarrow \infty$, the UAV reacts to every single obstacle around it. These reactions can slow down movement as the UAV will unnecessarily try to avoid far-away obstacles.

However, if R^o is too small, then the UAV will become short-sighted. Consequently, the UAV might not react in time to some of the surrounding obstacles, which can lead to collisions.

- The parameter K^n is the coefficient of the navigation vector of an informed UAV (15).

When K^n is large, the informed UAV will move to the goal with a high velocity, which may result in a loss of LoS with other UAVs when operating in a cluttered environment. Moving at high velocities can also affect the stability of the UAV when it tries to avoid collisions with obstacles.

However, when K^n is too small, the UAV will move more slowly, which in turn, slows down the entire

swarm.

- The parameter K^c is the coefficient of the collision avoidance vector \mathbf{c} (20). Small values of K^c would result in insufficient and slow reaction to obstacles, risking collisions. Large values of K^c would result in violent reaction of the UAV in the presence of obstacles. In the worst case (just as in the case of K^n), this could negatively affect the stability of the UAV.
- K^m is introduced in Algorithm 1. When the i -th UAV loses LoS with the j -th UAV for more than K^m time instants, the j -th UAV is discarded from the set \mathcal{N}_i .

When the value of K^m is large, the UAVs will be part of the set \mathcal{N}_i long after LoS has been lost. Thus, the position information of these UAVs will also be outdated. As the navigation vector depends on the number of UAVs in \mathcal{N}_i , the i -th UAV will react to the outdated position information, which can cause unnecessary delays in the collective motion of the entire swarm.

In contrast, when K^m is small, the i -th UAV will remove the UAVs as soon as the LoS is lost. As the environment is cluttered with obstacles, the loss of LoS would be frequent. Thus, at any time k , the i -th UAV will only have a few UAVs in \mathcal{N}_i , which in turn reduces the options for potential targets. As a result, having a small K^m can severely disrupt the target selection process and might produce erratic trajectories.

- K^p is introduced in Algorithm 2. It determines

the length of the path history stored for each UAV. This length is crucial for the path similarity (4) and path persistence (6) metrics. The motion of most of the UAVs is similar on very small and very large timescales. As these metrics are calculated on normalized vector quantities, the observed path histories of multiple UAVs can have similar values for metrics. Thus, for both small and large values of K^p , the metrics would not reflect the path information, resulting in a frequent change in the selected target. When a UAV is frequently switching between target UAVs, the overall motion of the swarm can be slowed down.

8. Conclusion

This article has presented PACNav as a new bioinspired decentralized approach for navigating a UAV swarm to the desired goal as a compact group. In contrast to state-of-the-art methods, the presented approach does not require communication among the members or a global localization system. Such an approach is highly beneficial in demanding real-world conditions where global localization is unavailable or has a high uncertainty and the swarm size makes communication unfeasible. In the presented approach, each UAV determines its future motion using the information derived from only on-board sensors, mimicking the sensory organs of animals moving in a group. The resultant decentralized swarm is scalable as it is not limited by communication bandwidth and information sharing. The navigation method based on the metrics derived from the collective motion of animals in nature ensures coherent movement of the swarm and collision avoidance with the environment and other members. The simulated experiments in the Gazebo robotic simulator and a real-world flight in a natural forest validated the effectiveness of the presented approach. An intensive analysis carried out in simulation demonstrated the reliability of the algorithm concerning changes in the number of informed UAVs, as well as different spatial distribution of trees in the forest. The software framework used to deploy the algorithm on a decentralized swarm of UAVs is provided as open-source[‡] in order to facilitate further research and replication of the obtained results.

Acknowledgment

This work was partially supported by the Czech Science Foundation (GAČR) under research project no. 20-10280S, by the European Union's Horizon 2020 research and innovation programme AERIAL-CORE under grant agreement no. 871479, by CTU grant no. SGS20/174/OHK3/3T/13, by the Technology Innovation Institute - Sole Proprietorship LLC, UAE, and by OP VVV project CZ.02.1.01/0.0/0.0/16_019/0000765 "Research Center for Informatics". The authors would like to thank Daniel Heřt for his help with the experimental setup.

ORCID

- Afzal Ahmad  : 0000-0002-5889-0320
 Daniel Bonilla Licea  : 0000-0002-1057-816X
 Giuseppe Silano  : 0000-0002-6816-6002
 Tomáš Báča  : 0000-0001-9649-8277
 Martin Saska  : 0000-0001-7106-3816

References

- [1] Zhou X, Wen X, Wang Z, Gao Y, Li H, Wang Q, Yang T, Lu H, Cao Y, Xu C and Gao F 2022 Swarm of micro flying robots in the wild *Science Robotics* **7** eabm5954
- [2] Chung S, Paranjape A A, Dames P, Shen S and Kumar V 2018 A Survey on Aerial Swarm Robotics *IEEE Transactions on Robotics* **34** 837–855
- [3] Hönig W, Preiss J A, Kumar T K S, Sukhatme G S and Ayanian N 2018 Trajectory Planning for Quadrotor Swarms *IEEE Transactions on Robotics* **34** 856–869
- [4] Augugliaro F, Lupashin S, Hamer M, Male C, Hehn M, Mueller M W, Willmann J S, Gramazio F, Kohler M and D'Andrea R 2014 The Flight Assembled Architecture installation: Cooperative construction with flying machines *IEEE Control Systems Magazine* **34** 46–64
- [5] Elmokadem T 2019 Distributed Coverage Control of Quadrotor Multi-UAV Systems for Precision Agriculture *IFAC-PapersOnLine* vol 52 pp 251–256
- [6] McGuire K N, De Wagter C, Tuyls K, Kappen H J and de Croon G C H E 2019 Minimal navigation solution for a swarm of tiny flying robots to explore an unknown environment *Science Robotics* **4** eaaw9710
- [7] Tagliabue A, Kamel M, Siegwart R and Nieto J 2019 Robust collaborative object transportation using multiple MAVs *The International Journal of Robotics Research* **38** 1020–1044
- [8] Inada Y and Takanobu H 2010 Flight-Formation Control of Air Vehicles Based on Collective Motion Control of Organisms *IFAC-PapersOnLine* vol 43 pp 386–391
- [9] Novak F, Walter V, Petráček P, Báča T and Saska M 2021 Fast collective evasion in self-localized swarms of unmanned aerial vehicles *Bioinspiration & Biomimetics* **16** 026009
- [10] Chamseddine A, Zhang Y and Rabbath C A 2012 Trajectory planning and re-planning for fault tolerant formation flight control of quadrotor unmanned aerial vehicles *America Control Conference* pp 3291–3296
- [11] Petráček P, Krátký V and Saska M 2020 Dronument: System for Reliable Deployment of Micro Aerial Vehicles in Dark Areas of Large Historical Monuments *IEEE Robotics and Automation Letters* **5** 2078–2085
- [12] Nagy M, Ákos Z, Biro D and Vicsek T 2010 Hierarchical group dynamics in pigeon flocks *Nature* **464** 890–893
- [13] Couzin I 2007 Collective minds *Nature* **445** 715–715
- [14] Yomosa M, Mizuguchi T, Vásárhelyi G and Nagy M 2015 Coordinated behaviour in pigeon flocks *Plos one* **10** e0140558
- [15] Reynolds C W 1987 Flocks, Herds and Schools: A Distributed Behavioral Model *Proceedings of the 14th Annual Conference on Computer Graphics and Interactive Techniques* vol 21 pp 25–34
- [16] Olfati-Saber R 2006 Flocking for multi-agent dynamic systems: algorithms and theory *IEEE Transactions on Automatic Control* **51** 401–420
- [17] Cheng H, Zhu Q, Liu Z, Xu T and Lin L 2017 Decentralized navigation of multiple agents based on ORCA and model predictive control *IEEE/RSJ International Conference on Intelligent Robots and Systems* pp 3446–3451

- [18] Zhou X, Zhu J, Zhou H, Xu C and Gao F 2021 EGO-Swarm: A Fully Autonomous and Decentralized Quadrotor Swarm System in Cluttered Environments *IEEE International Conference on Robotics and Automation* pp 4101–4107
- [19] Vásárhelyi G, Virágó C, Somorjai G, Nepusz T, Eiben A E and Vicsek T 2018 Optimized flocking of autonomous drones in confined environments *Science Robotics* **3**
- [20] Ahmad A, Walter V, Petracek P, Petrlik M, Baca T, Zaitlik D and Saska M 2021 Autonomous Aerial Swarming in GNSS-denied Environments with High Obstacle Density *IEEE International Conference on Robotics and Automation* pp 570–576
- [21] Dmytryuk A, Nascimento T, Ahmad A, Báča T and Saska M 2021 Safe Tightly-Constrained UAV Swarming in GNSS-denied Environments *International Conference on Unmanned Aircraft Systems* pp 1391–1399
- [22] Petráček P, Krátký V, Petrlík M, Báča T, Kratochvíl R and Saska M 2021 Large-Scale Exploration of Cave Environments by Unmanned Aerial Vehicles *IEEE Robotics and Automation Letters* **6** 7596–7603
- [23] Wang Q, Chen M and Li X 2017 Quantifying and Detecting Collective Motion by Manifold Learning *AAAI Conference on Artificial Intelligence* pp 4292–4298
- [24] Anam T, Jari B, Mohammad-Hashem H, Hannu T T and Juha P 2019 Swarms of Unmanned Aerial Vehicles — A Survey *Journal of Industrial Information Integration* **16** 100106
- [25] Karthik E and Spring B 2019 Mean-field models in swarm robotics: a survey *Bioinspiration & Biomimetics* **15** 015001
- [26] Verginis C K, Xu Z and Dimarogonas D V 2017 Decentralized motion planning with collision avoidance for a team of UAVs under high level goals *IEEE International Conference on Robotics and Automation* pp 781–787
- [27] Cotsakis R, St-Onge D and Beltrame G 2019 Decentralized collaborative transport of fabrics using micro-UAVs *IEEE International Conference on Robotics and Automation* pp 7734–7740
- [28] Kushleyev A, Mellinger D, Powers C and Kumar V 2013 Towards A Swarm of Agile Micro Quadrotors *Autonomous Robots* **35** 287–300
- [29] Schilling F, Schiano F and Floreano D 2021 Vision-Based Drone Flocking in Outdoor Environments *IEEE Robotics and Automation Letters* **6** 2954–2961
- [30] Dai F, Chen M, Wei X and Wang H 2019 Swarm Intelligence-Inspired Autonomous Flocking Control in UAV Networks *IEEE Access* **7** 61786–61796
- [31] Weinstein A, Cho A, Loianno G and Kumar V 2018 Visual Inertial Odometry Swarm: An Autonomous Swarm of Vision-Based Quadrotors *IEEE Robotics and Automation Letters* **3** 1801–1807
- [32] Petráček P, Walter V, Báča T and Saska M 2020 Bio-inspired compact swarms of unmanned aerial vehicles without communication and external localization *Bioinspiration & Biomimetics* **16** 026009
- [33] Walter V, Staub N, Saska M and Franchi A 2018 Mutual Localization of UAVs based on Blinking Ultraviolet Markers and 3D Time-Position Hough Transform *IEEE 14th International Conference on Automation Science and Engineering* pp 298–303
- [34] Walter V, Staub N, Franchi A and Saska M 2019 UVDAR System for Visual Relative Localization With Application to Leader–Follower Formations of Multirotor UAVs *IEEE Robotics and Automation Letters* **4** 2637–2644
- [35] Vrba M and Saska M 2020 Marker-Less Micro Aerial Vehicle Detection and Localization Using Convolutional Neural Networks *IEEE Robotics and Automation Letters* **5** 2459–2466
- [36] Horyna J, Walter V and Saska M 2022 UVDAR-COM: UV-Based Relative Localization of UAVs with Integrated Optical Communication *International Conference on Unmanned Aircraft Systems* pp 1341–1347
- [37] Kohlbrecher S, von Stryk O, Meyer J and Klingauf U 2011 A flexible and scalable SLAM system with full 3D motion estimation *IEEE International Symposium on Safety, Security, and Rescue Robotics* pp 155–160
- [38] Hart P E, Nilsson N J and Raphael B 1968 A Formal Basis for the Heuristic Determination of Minimum Cost Paths *IEEE Transactions on Systems Science and Cybernetics* **4** 100–107
- [39] Ma L, Bao W, Zhu X, Wu M, Wang Y, Ling Y and Zhou W 2020 O-Flocking: Optimized Flocking Model on Autonomous Navigation for Robotic Swarm *Advances in Swarm Intelligence* pp 628–639
- [40] Koenig N and Howard A 2004 Design and Use Paradigms for Gazebo, An Open-Source Multi-Robot Simulator *IEEE/RSJ International Conference on Intelligent Robots and Systems* pp 2149–2154
- [41] Silano G, Oppido P and Iannelli L 2019 Software-in-the-loop simulation for improving flight control system design: a quadrotor case study *IEEE International Conference on Systems, Man and Cybernetics* pp 466–471
- [42] Soria E, Schiano F and Floreano D 2020 SwarmLab: a Matlab Drone Swarm Simulator *IEEE/RSJ International Conference on Intelligent Robots and Systems* pp 8005–8011
- [43] Baca T, Petrlík M, Vrba M, Spurný V, Penicka R, Hert D and Saska M 2021 The MRS UAV System: Pushing the Frontiers of Reproducible Research, Real-world Deployment, and Education with Autonomous Unmanned Aerial Vehicles *Journal of Intelligent & Robotic Systems* **102** 1–28
- [44] Hert D, Baca T, Petráček P, Krátký V, Spurný V, Petrlík M, Matous V, Zaitlik D, Stoudek P, Walter V, Stepan P, Horyna J, Přízvl V, Silano G, Bonilla Licea D, Stibinger P, Penicka R, Nascimento T and Saska M 2022 MRS Modular UAV Hardware Platforms for Supporting Research in Real-World Outdoor and Indoor Environments *International Conference on Unmanned Aircraft Systems* pp 1303–1312
- [45] Přízvl V, Vrba M, Stepan P and Saska M 2022 Cooperative Navigation and Guidance of a Micro-Scale Aerial Vehicle by an Accompanying UAV using 3D LiDAR Relative Localization *International Conference on Unmanned Aircraft Systems* pp 517–526