Autonomous Aerial Swarming in GNSS-denied Environments with High Obstacle Density

Afzal Ahmad, Viktor Walter, Pavel Petráček, Matěj Petrlík, Tomáš Báča, David Žaitlík and Martin Saska

***Abstract*— The compact flocking of relatively localized Unmanned Aerial Vehicles (UAVs) in high obstacle density areas is discussed in this paper. The presented work tackles realistic scenarios in which the environment map is not known apriori and the use of a global localization system and communication infrastructure is difficult due to the presence of obstacles. To achieve flocking in such a constrained environment, we propose a fully decentralized, bio-inspired control law that uses only onboard sensor data for safe flocking through the environment without any communication with other agents. In the proposed approach, each UAV agent uses onboard sensors to self-localize and estimate the relative position of other agents in its local reference frame. The usability and performance of the proposed approach were verified and evaluated using various experiments in a realistic robotic simulator and a natural forest. The presented experiments also validate the utility of onboard relative localization for autonomous multi-UAV applications in the absence of global localization information and communication.**

***Index Terms*— Multi-Robot Systems, Aerial Systems: Perception and Autonomy, Field Robotics**

I. Introduction

Recent developments in the miniaturization of computational resources and sensors have led to significant progress in the field of UAVs. These developments make it increasingly less challenging to manufacture large numbers of small, lightweight UAV units at low cost. Such units are typically not equipped with great computational or sensory resources, often due to costs, payload capacity, or unavailability of the desired form factor. However, these simple, low-cost UAVs lend themselves remarkably well for applications that require multiple units to be deployed simultaneously. For example, such applications include the measurement of spatially distributed time-dependent variables [1], [2], collaborative data collection [3], and applications where redundancy is needed due to various environmental hazards [4]. They also have the potential to be used for environmental monitoring and Search and Rescue (SAR) applications [5], [6] where large-scale teams usually provide significant advantages in area coverage. More importantly, a team of small-scale UAVs can easily operate in environments with a high obstacle density, where larger, well-equipped aerial vehicles cannot operate.

The presented work has been supported by the Czech Science Foundation (GAČR) under research project no. 20-10280S, by CTU grant no. SGS20/174/OHK3/3T/13, by the Technology Innovation Institute - Sole Proprietorship LLC, UAE, under the Research Project Contract No. TII/ATM/2032/2020, and by OP VVV funded project CZ.02.1.01/0.0/0.0/16 019/0000765 "Research Center for Informatics".

The authors are with the Faculty of Electrical Engineering, Czech Technical University in Prague, Technická 2, Prague 6
{ahmadafz|viktor.walter|pavel.petracek|matej.petrlik|tomas.baca|david.zaitlik|martin.saska}@fel.cvut.cz.

Fig. 1: A compact aerial swarm of three UAVs flocking in a complex outdoor environment.

Although these applications are highly beneficial, multi-UAV systems are still only used in limited cases due to numerous technological challenges in such environments. Primarily, the lack of reliable global localization and mutual communication becomes a significant bottleneck for the safe operation of UAV teams as the Global Navigation Satellite System (GNSS) signal and wireless communication links often experience disturbances by occlusions and reflections from obstacles. In this article, we propose a compact collective motion approach entirely based on onboard sensory information. This approach can be used as a back-up safety mechanism for high-level planning systems where even temporary global localization or communication failure would have fatal consequences. The proposed flocking behavior takes inspiration from biological systems [7], which can maintain a compact group by only using "onboard" sensory organs and without any explicit communication. In these systems, a small number of individuals with goal information are sufficient to navigate the entire group towards the goal. Thus, a multi-UAV system using the proposed approach can continue its mission with reduced capabilities or navigate towards an area where global localization and communication capabilities can be restored to obtain full performance.

A crucial piece of information for any cooperative motion is the knowledge of the relative state of other agents in the environment. However, due to the lack of global localization and communication in the discussed case, the members of the multi-UAV system cannot use a common global reference frame to localize themselves and other agents. Thus, a mutual relative localization system is required to estimate the states of other agents in a way similar to how animals use their organs to observe neighbors to achieve collective behav-

ior. In addition to the localization of other agents, obstacle avoidance is another major challenge when cooperating in an environment with a high density of obstacles. Since a prior map for an unknown real-world environment such as a forest is unavailable, we use onboard sensing for obstacle avoidance. To achieve flocking in cluttered real-world environments without using GNSS and communication, we propose a framework based on a planar LIDAR based Simultaneous Localization and Mapping (SLAM) and the relative localization system UltraViolet Direction And Ranging (*UVDAR*), which has been specifically designed for such applications (see Fig. 1). In the proposed flocking control law, information obtained from these onboard sensors is fused together using bio-inspired flocking principles. The method uses specifically designed self-aggregation to cope with the inherent uncertainty of cluttered real-world environments. The proposed flocking control law places high emphasis on safety and overcoming the unforeseen challenges encountered during multi-UAV interaction with a complex environment.

The presented swarming framework is decentralized and infrastructure-independent as each UAV agent only uses information from on-board sensors. The method is validated by performing self-aggregation and flocking experiments in a natural environment with a high density of obstacles.

II. Related Work

Decentralized robotic swarms have been extensively studied in recent years [8], but most of the research is still limited to laboratory-like conditions [9]–[12]. Previous work presented in [13] and [14] demonstrate a swarm in an outdoor environment. However, they use GNSS which is unavailable in many real-world scenarios involving UAVs. A scalable swarm of UAVs is presented in [15] for GNSS-denied outdoor environments, but it relies on inter-agent communication and does not account for obstacles in the environment. This paper proposes a framework to overcome these issues such that the UAV swarm can be used in an arbitrary outdoor environment without any dependence on external infrastructure and communication.

The presented framework uses mutual relative localization, which has been achieved in the literature using various methods. In [16], distance information from Ultra Wide Band (UWB) modules is combined with the on-board localization method to obtain a relative state estimate of other agents. [17] uses computer vision to detect the surrounding agents and then fuses this information with on-board UWB and localization estimate. These approaches obtain a precise relative position of other agents, but they rely on communication between the agents.

Alternatively, computer vision based methods such as [18], [19] use Convolutional Neural Networks (CNNs) to estimate relative position from on-board cameras. They do not require inter-agent communication, but they depend on lighting conditions, which reduces their reliability for real-world deployment as tested by our team in [20]. In order to work in an arbitrary complex environment without communication constraints, we use the *UVDAR* system [21]–[23], which was specifically developed to improve swarm usability in outdoor environments.

The flocking method proposed in this work is based on the rules of bio-inspired flocking introduced in [24], which have been used for many applications involving robot swarms. To use these rules in real-world environments with high obstacle density, it is crucial to design a reliable collision avoidance method. A common solution found in the literature is to model the obstacles as agents or as a group of agents [13], [14], [25], [26] and to use the agent repulsion law to prevent collisions. 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. An alternative approach is to design a specialized control barrier function to avoid the observed obstacles [27]. Taking inspiration from these methods, we propose a new approach for flocking that uses different methods to avoid collisions with agents and obstacles with a high emphasis on safety for the UAV. This method enabled us to navigate through a complex real-world environment safely.

III. Self-aggregation and flocking

Self-aggregation and flocking is achieved by controlling the motion of each UAV agent using a control law. Considering each UAV as a double-integrator system, the control law is used to generate the desired acceleration of the agent. Each agent uses the same control law, and the flocking emerges from their collective motion. The proposed control law is a combination of several components responsible for controlling the distance to other agents, collision avoidance to obstacles, and navigating the agents towards a goal. The selection of these components is motivated by the bio-inspired flocking presented in [24]. The resulting control law can be expressed as

$$\boldsymbol{f} = \boldsymbol{p} + \boldsymbol{n} + \boldsymbol{c}, \tag{1}$$

where $\boldsymbol{f} \in \mathbb{R}^3$ denotes the final control vector used to generate the desired acceleration of the UAV agent. Vectors $\boldsymbol{p}, \boldsymbol{n}, \boldsymbol{c} \in \mathbb{R}^3$ represent the proximal, navigation, and collision control vectors, respectively, and are described in the following subsections.

A. Proximal Control Vector

The proximal control vector is used to control the distance to the surrounding agents. For successful flocking and reliable performance of relative localization, each UAV agent must maintain a particular distance from other agents. UAV agents close to each other should recede to avoid inter-agent collisions, while agents far away should move towards the rest of the agents to maintain a cohesive group. This is implemented as repulsion and attraction to other agents. The

proximal control vector is obtained as

$$\boldsymbol{p} = \frac{1}{N_\alpha} \sum_{i=1}^{N_\alpha+N_\beta} \Big(\alpha(\|\boldsymbol{x}_i^a\|)\hat{x}_i^a\Big) + \frac{1}{N_\beta} \sum_{i=1}^{N_\alpha+N_\beta} \Big(\beta(\|\boldsymbol{x}_i^a\|)\hat{x}_i^a\Big), \tag{2}$$

where $\boldsymbol{x}_i^a \in \mathbb{R}^3$ is the current relative position of the i^{th} agent. The function $\alpha(\cdot) \in \mathbb{R}$ realizes the attraction and $\beta(\cdot) \in \mathbb{R}$ realizes the repulsion effects (see Fig. 2). N_α and N_β are the number of observed agents with a non-zero attraction or repulsion value.

1) Attraction Function: The attraction $\alpha(\cdot)$ is a piece-wise function designed such that the agents are attracted to each other only if they are separated by a distance greater than a parameter d_0. This parameter describes the distance at which the agent experiences no attraction or repulsion effects. To prevent the agents from going far apart from each other, $\alpha(\cdot)$ increases rapidly as the inter-agent distance grows greater than d_0. However, if the agents are farther apart than a threshold distance d_f, the attraction is primarily used to approach an agent, rather than maintain the distance d_0. Thus, $\alpha(\cdot)$ converges to a constant value as the distance increases beyond d_f. The attraction function is expressed as

$$\alpha(\|\boldsymbol{x}_i^a\|) = \begin{cases} 0, & 0 \le \|\boldsymbol{x}_i^a\| \le d_0 \\ k_\alpha(\|\boldsymbol{x}_i^a\| - d_0)^2, & d_0 < \|\boldsymbol{x}_i^a\| \le d_f \\ \lambda_\alpha \tan^{-1}(\|\boldsymbol{x}_i^a\| - 0.9d_f), & d_f < \|\boldsymbol{x}_i^a\| \end{cases} \tag{3}$$

where $k_\alpha \in \mathbb{R}$ is a constant that controls the rate of change of attraction as the inter-agent distance grows beyond d_0 and

$$\lambda_\alpha = \frac{k_\alpha(d_f - d_0)^2}{\tan^{-1}(0.1d_f)} \tag{4}$$

is a scaling factor used to make the piece-wise function continuous.

2) Repulsion Function: The repulsion $\beta(\cdot)$ is also a piece-wise function and is used to prevent inter-agent collisions. The repulsion function is designed to increase slowly when the agents get closer than the parameter d_0. However, when the inter-agent distance is smaller than a threshold parameter d_c, the agents are considered critically close. As such a situation would require rapid and immediate separation to avoid collisions, $\beta(\cdot)$ increases rapidly for a distance smaller than d_c. The agents beyond the distance d_0 are considered sufficiently far away, so the repulsion function converges to zero beyond d_0. The function $\beta(\cdot)$ is expressed as

$$\beta(\|\boldsymbol{x}_i^a\|) = \begin{cases} -\lambda_\beta\left(\frac{1}{\sqrt{\|\boldsymbol{x}_i^a\|}} - \frac{1}{\sqrt{d_0}}\right), & 0 < \|\boldsymbol{x}_i^a\| \le d_c \\ -k_\beta(\|\boldsymbol{x}_i^a\| - d_0)^2, & d_c < \|\boldsymbol{x}_i^a\| \le d_0 \\ 0, & d_0 < \|\boldsymbol{x}_i^a\| \end{cases} \tag{5}$$

where $k_\beta \in \mathbb{R}$ is a constant that controls the rate of change of repulsion as the inter-agent distance decreases below d_0 and

$$\lambda_\beta = \frac{k_\beta(d_c - d_0)^2\sqrt{d_c d_0}}{\sqrt{d_0} - \sqrt{d_c}} \tag{6}$$

is a scaling factor used to make the piece-wise function continuous.

The motion of the agents is strongly influenced by the parameters d_0, d_f, and d_c. The parameter d_c represents the distance after which the agents should not approach any closer. This can be selected on the basis of the safety distance required between the UAVs. Similarly, d_f describes the distance after which the attraction is primarily used for forming a cohesive group, thus d_f can be selected based on task and environment specifications.

B. Navigation Control

All agents in a swarm might not have the goal information. An agent with goal information is called an informed agent. These informed agents collectively guide the swarm to the desired goal. The navigation control vector steers the motion of an informed agent towards this desired goal. It uses a path planner (see Section IV-C) to obtain a path $\mathcal{P}$ to the desired goal. This path $\mathcal{P}$ is a set of points in $\mathbb{R}^3$, which describe the desired trajectory of the agent. The navigation control vector $\boldsymbol{n}$ is computed using this path as

$$\boldsymbol{n} = k_n\varphi(\mu_d)\hat{x}^n, \tag{7}$$

where $k_n \in \mathbb{R}$ is a constant that controls the effect of navigation control vector on the motion of the agent, $\boldsymbol{x}^n$ is the relative position of the nearest point on path $\mathcal{P}$, and the nonlinear function

$$\varphi(\mu_d) = \begin{cases} 1 - \left(\frac{\mu_d}{d_{max}}\right)^2, & 0 \le \mu_d \le d_{max} \\ 0, & d_{max} < \mu_d \end{cases} \tag{8}$$

prevents the informed agent from leaving the flock under the influence of the navigation control vector. As the relative mean distance of the observed agents μ_d increases, $\varphi(\cdot)$ decreases and converges to zero after a distance d_{max}. Parameter d_{max} is selected such that the informed agent can still detect other agents using its relative localization system. Since the navigation control is only used by the informed agents, other agents do not require any path planning.

C. Collision Control

Collision avoidance is essential to any agent for safe motion through an obstacle-rich environment, such as a forest. The collision control vector $\boldsymbol{c}$ is obtained using the relative position estimate of nearby obstacles from the local online

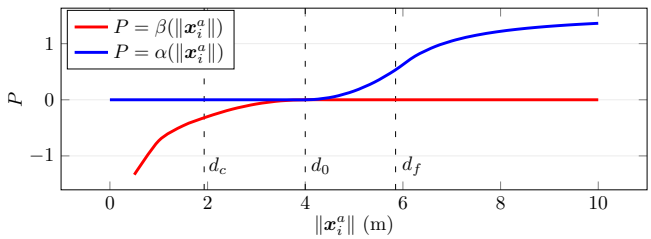

Fig. 2: Attraction and repulsion functions of the proximal control vector. The vertical lines show the values of the control vector parameters specified in Table I.

map (see Section IV-C) and is expressed as

$$\boldsymbol{c} = \frac{1}{N_c} \sum_{i=1}^{N_o} \phi(\|\boldsymbol{x}_i^o\|) \hat{x}_i^o, \tag{9}$$

where N_o is the total number of currently detected obstacles and $\boldsymbol{x}_i^o \in \mathbb{R}^3$ is the relative position of the i^{th} obstacle. N_c is the number of obstacles with a non-zero value of function

$$\phi(\|\boldsymbol{x}_i^o\|) = \begin{cases} k_o \left(\frac{1}{\sqrt{\|\boldsymbol{x}_i^o\|}} - \frac{1}{\sqrt{d_r}} \right), & 0 < \|\boldsymbol{x}_i^o\| \leq d_r \\ 0, & d_r < \|\boldsymbol{x}_i^o\|. \end{cases} \tag{10}$$

Since the agent only needs to avoid nearby obstacles, $\phi(\cdot)$ decreases to zero after a threshold distance d_r from the obstacle. The parameter d_r is selected taking into account the dynamics of the UAV agent, such that it has enough time to react to obstacles.

IV. System Architecture

The method described in Section III only requires the position of the UAV agent in an arbitrary reference frame and the relative position of neighboring agents and obstacles. The framework proposed in this section (see Fig. 4) retrieves this position information using only on-board sensors. As our framework does not need any external infrastructure or communication, the resulting swarm is strictly decentralized.

A. UVDAR

Each UAV agent is equipped with the *UVDAR* system to estimate the relative position and orientation of neighboring agents. This system uses UV LED markers to detect objects in its surroundings. Since the sun (the primary light source in natural environments) emits less radiation in the UV spectrum than visible light, the UV LEDs are distinctly visible even in broad daylight. Each UAV in the swarm is equipped with UV LEDs that act as markers and a pair of UV cameras (greyscale camera with a UV-filter) for detection. *UVDAR*

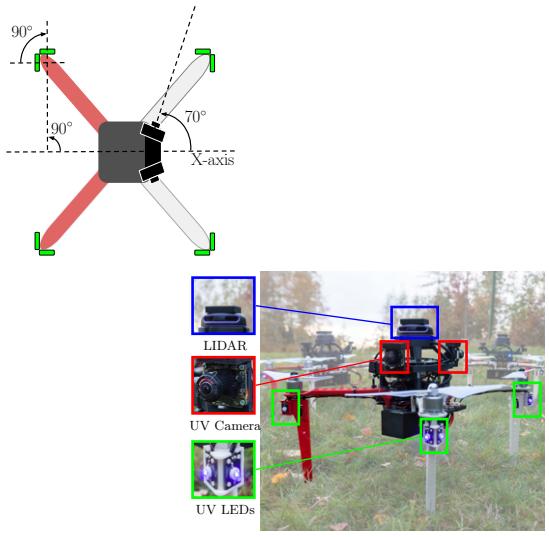

Fig. 3: The quadrotor platform highlighting the UV LEDs and the UV camera assembly used by the *UVDAR* system. The planar LIDAR was used for localization.

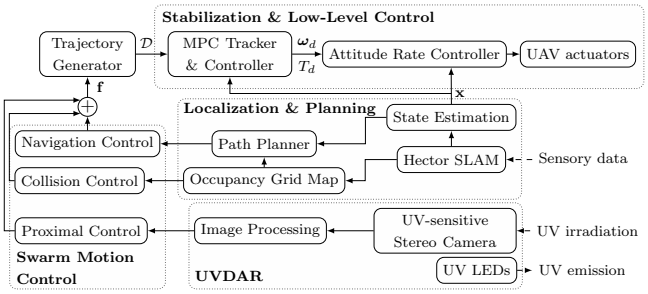

Fig. 4: The interconnected high-level system pipeline of a single UAV agent.

uses computer vision to localize the observed UV markers, retrieve the poses of neighbors, and tracks them using a linear Kalman filter.

The *UVDAR* system presented in [23] uses a single UV camera with a limited Field Of View (FOV) for detection of markers, which is not sufficient when the agent needs to observe more than one UAV. To use *UVDAR* for flocking with a swarm of UAVs, we propose the camera setup shown in Fig. 3, which significantly increases the FOV of the system and allows the UAV to react to neighboring agents. The cameras are aligned such that the setup has a 320° horizontal and a 110° vertical FOV with a 40° horizontal overlap in the frontal direction of the UAV. The resulting stereoscopic effect allows the system to make high-precision distance estimates from frontal observations. This alignment also creates a 40° blind spot at the back of the UAV. To avoid losing agents in this blind spot, each UAV continuously adjusts its orientation such that the cameras observe most of the neighboring agents. It was experimentally verified that this setup is sufficient for reliable pose estimation and flocking, and adding another camera would not increase system performance. Each UAV has a pair of UV LEDs mounted (see Fig. 3) to each leg, which blinks with a pre-assigned unique frequency. The unique frequency allows *UVDAR* to easily distinguish and track the individual UAV agents in a swarm.

B. Control

The desired acceleration of the UAV agent is obtained using the final control vector $\boldsymbol{f}$. A Trajectory Generator (see Fig. 4) uses this acceleration to generate the desired trajectory $\mathcal{D}$ as

$$\boldsymbol{r}_{j+1} = \boldsymbol{r}_j + \boldsymbol{v}\frac{j}{R_0} + \frac{1}{2}k_f \boldsymbol{f} \left(\frac{j}{R_0} \right)^2, \tag{11}$$

where $j \in \mathbb{Z}$ and $\boldsymbol{r}_j \in \mathbb{R}^3$ is the j^{th} trajectory point. $\boldsymbol{v} \in \mathbb{R}^3$ is the current velocity of the UAV, k_f is a scaling constant, and $R_0 \in \mathbb{R}$ is the sampling rate of the trajectory. Since the UAVs are moving continuously, we only calculate the trajectory for the next 1 second, i.e. $j \in \{1, 2, ..., R_0\}$.

In order to maintain the maximum number of UAVs in the FOV of the *UVDAR*, the heading direction ψ_d needs to change continuously. It is obtained as the average of angles made by the relative position vectors of neighboring agents w.r.t the body X-axis (see Fig. 3).

The Trajectory Generator supplies the sequence $\mathcal{D} = \{(\boldsymbol{r}_0, \psi_d), (\boldsymbol{r}_1, \psi_d), ..., (\boldsymbol{r}_{R_0}, \psi_d)\}$ as input to the control pipeline, as shown in Fig. 4. A linear MPC controller and tracker is used for safe and robust control of the UAV. This controller is part of the generic control pipeline developed by our group [28] for various single and multi robot applications.

C. Localization and Path Planning

A combination of LIDAR-based Hector SLAM [29] and a laser range finder for height estimation was employed for 3D localization and navigation. Hector SLAM is computationally inexpensive compared to other SLAM approaches and can work with a lightweight LIDAR, which makes it a suitable choice for small aerial vehicles. It does not use pose graph optimization for loop closures, which reduces the long-term precision of the UAV pose estimate. However, our flocking method only uses local map information for motion control, thus the Hector-provided estimate is sufficiently precise for reliable navigation.

Hector SLAM generates a 2D occupancy grid map of the environment. To account for the safety of the UAV, the obstacles in the map are inflated w.r.t the UAV size. This grid map is used by the A-star [30] algorithm to find the shortest path to the goal. We use the sum of euclidean distance to the goal and the closest obstacle as a heuristic function. The unknown cells in the grid are considered free, so that the algorithm can generate a path towards a goal point that is not yet seen by the UAV. The generated path $\mathcal{P}$ is used by the navigation control vector, as described in Section III-B. Since path planning inherently incorporates obstacle avoidance, it adds another layer of safety to UAV motion through complex environments.

V. Experimental Analysis

All the UAVs used in the experiments are homogeneous, *i.e.*, they have identical hardware and software, and use the same parameters in the flocking method (see Table I). The localization and control pipeline developed by our research group [28] was used for the simulation, as well as real-world evaluations. Since the pipeline is designed to ease the simulation-to-real-world transition of robotic software, it was extremely helpful while working with such a complex real-world system.

The experiments are supported by multimedia materials, which are available at `https://youtu.be/HH78AheC-DM`.

Control Vectors

Proximal					Navigation		Collision	
d_0 (m)	d_f (m)	k_a (-)	d_c (m)	k_r (-)	d_{max} (m)	k_n (-)	d_r (m)	k_o (-)
4.00	5.85	0.16	1.93	0.07	8.00	1.20	2.00	2.00

TABLE I: Control vector parameters used for the experiments.

A. Validation in a Simulated Forest

The simulation was used to analyze the performance of the proposed flocking approach and verify the readiness for real-world testing. It was also beneficial for the initial tuning of the parameters. The simulated UAVs have the same system architecture as described in Section IV. A simulated version of *UVDAR* is used for mutual relative localization and the UAVs do not communicate any information to other agents. The simulation is performed in the Gazebo robot simulator using the Robot Operating System (ROS). To minimize assumptions about the environment, a recorded 3D model of a natural forest (see Fig. 5), also published online, is used in the simulation.

As shown in Fig. 5, the simulated experiments successfully verify the qualitative performance of the proposed method for self-aggregation and flocking in a complex environment. Initially, the UAVs self-aggregate to form a cohesive group, after which an informed UAV navigates the swarm to a particular goal position. It is possible to provide all the UAVs with the same goal, which can potentially increase the speed of the group. However, only a single UAV is used in the experiments to verify the flocking performance with minimum possible information. The simulations for a higher number of informed agents can be found at `https://youtu.be/2QV9bDumznE`. Although only one of the UAVs is provided with the goal information in the experiment, the entire group moves towards the desired goal in order to maintain the cohesion of the swarm. The swarm deforms when moving around obstacles to avoid possible collisions with the environment and with neighboring agents. The deformation and re-organization emerge as a direct result of the flocking control law and significantly helps when navigating through the complex environment. The simulated flights validate the usability of the proposed method for autonomous navigation in complex, obstacle-rich environments, such as a forest.

B. Natural Forest Experiments

A quadrotor platform based on the *DJI f450* frame with an arm length of 22.5 cm (shown in Fig. 3) is used for the real-world experiments. Each UAV has a *Pixhawk 4* flight controller unit and an *Intel NUC8i7BEH* as an on-board computer. The UAVs are also equipped with a *Slamtec RP-LIDAR A3* for SLAM and a *Garmin LIDAR-Lite v3* line-of-sight range finder, which is used to measure the height of the UAV from the ground. A fully equipped UAV unit has a mass of 2.6 kg and is powered by a 14.8 V *LiPo* battery. Although we use a relatively high-end on-board computer, the presented self-aggregation and flocking method does not require any heavy computation and a microcontroller can easily be used to perform flocking, even when using smaller platforms.

A group of three fully equipped UAVs was deployed in a natural forest (see Fig. 1) to analyze and validate the feasibility of the proposed flocking method in a complex real-world environment. The UAVs used the *UVDAR* system for mutual relative localization and did not communicate any information to other agents.

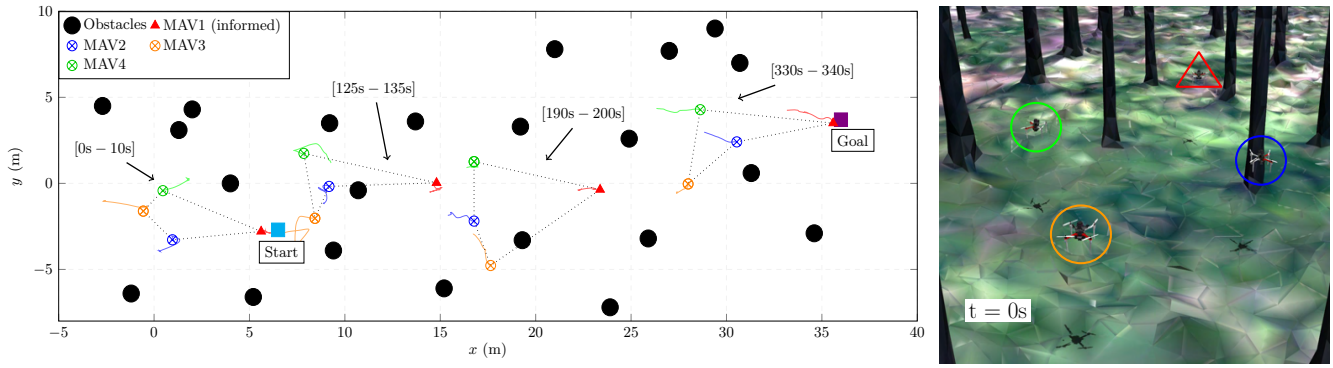

Fig. 5: Flight trajectories from the simulation. The blue square is the initial position of red UAV (goal-informed) and the purple square is the goal position for red UAV. The blue, green, and orange UAVs do not have any goal information (uninformed). The trailing line with each UAV denotes the recorded trajectory for the time interval specified in the figure.

Although a simple control law governs the swarm, it was able to navigate through the forest as a cohesive group as shown in Fig. 6. In the initial phase of the experiment, the UAVs self-aggregate to form a group, after which the informed UAV starts moving towards the goal under the influence of its navigation control vector. Other UAV agents react to this motion and try to maintain the cohesion while simultaneously avoiding collisions. This simple act of maintaining the group cohesion results in the collective motion of the entire group towards the goal. Once the informed agent reaches the goal, the other agents self-organize around it. The informed UAV then moves back to its initial position, which moves the swarm back to its initial location.

The high obstacle density in the forest and measurement noise in the real-world made the deployment extremely challenging. Occasional detection of UAVs or incorrect measurements of trees by the LIDAR would often become a part of the Hector SLAM generated map. Since the collision control vector uses this map for obstacle avoidance, it would result in a disturbed motion until subsequent measurements. To avoid these disturbances, the maximum obstacle avoidance distance d_r was reduced to react only to nearby trees. While navigating the forest, the UAVs would also get occluded by nearby trees and other agents. This affected the estimation accuracy of *UVDAR* and posed a major challenge in the highly-dense forest. The pose estimate would improve as the UAVs moved and became visible again. However, the UAVs needed to be sufficiently far apart to prevent any collisions, while the estimate was poor. We were able to provide the UAVs with more inter-agent space by increasing the d_0 parameter of the proximal control vector, but it was not sufficient in all the cases. Improving the detection and estimation of occluded objects in outdoor environments is an open research problem and is a topic for our future work. However, even a simple technique inspired by biological flocking, as presented in this work, proved to be sufficient for robust and safe flight of a swarm of UAVs.

VI. Conclusion

An approach for fully autonomous and decentralized multi-UAV flocking in a complex environment was presented in this paper. The performance of the proposed approach has been validated by numerous experiments in a realistic robot simulator and real-world experiments in a natural forest. Such experimental deployment is crucial for understanding the challenges of using multi-UAV systems in real-world scenarios. It is, to the best of our knowledge, the first work to design a fully decentralized framework to achieve flocking in a real-world environment without communication or an external source of localization. The flocking control law is designed to overcome the challenges of deployment in natural environments with a high obstacle density. Since the deployment of the proposed approach does not depend on external infrastructure, it has the potential to become a safety back-up mechanism for multi-UAV systems in applications, such as SAR scenarios or in forest fire-fighting where the loss of GNSS signal and communication connectivity is likely to occur.

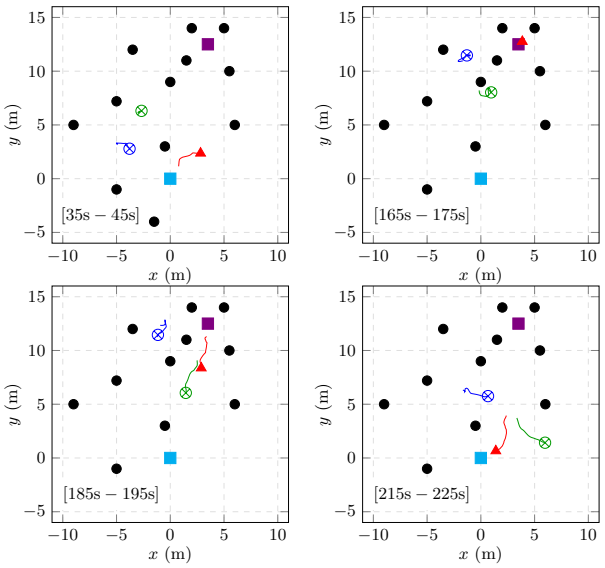

Fig. 6: Flight trajectories from the real forest experiment shown in the video `https://youtu.be/HH78AheC-DM`. The goal-informed UAV (red triangle) navigates the uninformed UAVs (green and blue crosses) through an obstacle-rich forest (black circles). The blue square is the initial position and the purple square is the goal position of the red UAV. The trailing line with each UAV denotes the recorded trajectory in the time interval specified in the figure.

REFERENCES

[1] M. Almeida *et al.*, "Distributed UAV-swarm-based real-time geomatic data collection under dynamically changing resolution requirements," *The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 42, p. 5, 2017.

[2] P. Štibinger, T. Báča, and M. Saska, "Localization of Ionizing Radiation Sources by Cooperating Micro Aerial Vehicles With Pixel Detectors in Real-Time," *IEEE RAL*, vol. 5, no. 2, pp. 3634–3641, 2020.

[3] P. Petráček, V. Krátký, and M. Saska, "Dronument: System for Reliable Deployment of Micro Aerial Vehicles in Dark Areas of Large Historical Monuments," *IEEE RAL*, vol. 5, no. 2, pp. 2078–2085, 2020.

[4] A. Chamseddine, Y. Zhang, and C. A. Rabbath, "Trajectory planning and re-planning for fault tolerant formation flight control of quadrotor unmanned aerial vehicles," in *2012 ACC*, 2012, pp. 3291–3296.

[5] M. Petrlík, T. Báča, D. Heřt, M. Vrba *et al.*, "A Robust UAV System for Operations in a Constrained Environment," *IEEE RAL*, vol. 5, no. 2, pp. 2169–2176, 2020.

[6] R. D. Arnold *et al.*, "Search and rescue with autonomous flying robots through behavior-based cooperative intelligence," *Journal of International Humanitarian Action*, vol. 3, no. 1, p. 18, 2018.

[7] C. C. Ioannou, M. Singh, and I. D. Couzin, "Potential leaders trade off goal-oriented and socially oriented behavior in mobile animal groups," *The American Naturalist*, vol. 186, no. 2, pp. 284–293, 2015, pMID: 26655156. [Online]. Available: https://doi.org/10.1086/681988

[8] S. Chung, A. A. Paranjape, P. Dames, S. Shen, and V. Kumar, "A Survey on Aerial Swarm Robotics," *IEEE Transactions on Robotics*, vol. 34, no. 4, pp. 837–855, 2018.

[9] C. K. Verginis, Z. Xu, and D. V. Dimarogonas, "Decentralized motion planning with collision avoidance for a team of UAVs under high level goals," in *2017 IEEE ICRA*, pp. 781–787.

[10] R. Cotsakis, D. St-Onge, and G. Beltrame, "Decentralized collaborative transport of fabrics using micro-UAVs," in *2019 IEEE/ICRA*.

[11] A. Kushleyev *et al.*, "Towards A Swarm of Agile Micro Quadrotors," *Auton. Robot.*, vol. 35, no. 4, pp. 287–300, Nov 2013.

[12] M. Saska, V. Vonásek, J. Chudoba, J. Thomas, G. Loianno, and V. Kumar, "Swarm Distribution and Deployment for Cooperative Surveillance by Micro-Aerial Vehicles," *Journal of Intelligent & Robotic Systems*, vol. 84, no. 1, pp. 469–492, 2016.

[13] G. Vásárhelyi, C. Virágh, G. Somorjai, T. Nepusz, A. E. Eiben, and T. Vicsek, "Optimized flocking of autonomous drones in confined environments," *Science Robotics*, vol. 3, no. 20, 2018.

[14] P. Petráček, V. Walter, T. Báča, and M. Saska, "Bio-inspired compact swarms of unmanned aerial vehicles without communication and external localization," *Bioinspiration & Biomimetics*, vol. 16, no. 2, p. 026009, dec 2020. [Online]. Available: https://doi.org/10.1088/1748-3190/abc6b3

[15] A. Weinstein, A. Cho, G. Loianno, and V. Kumar, "Visual Inertial Odometry Swarm: An Autonomous Swarm of Vision-Based Quadrotors," *IEEE RAL*, vol. 3, no. 3, pp. 1801–1807, 2018.

[16] K. Guo, X. Li, and L. Xie, "Ultra-Wideband and Odometry-Based Cooperative Relative Localization With Application to Multi-UAV Formation Control," *IEEE Trans. on Cyber.*, vol. 50, no. 6, pp. 2590–2603, 2020.

[17] H. Xu, L. Wang, Y. Zhang, K. Qiu, and S. Shen, "Decentralized Visual-Inertial-UWB Fusion for Relative State Estimation of Aerial Swarm," in *2020 IEEE ICRA*, pp. 8776–8782.

[18] M. Vrba and M. Saska, "Marker-Less Micro Aerial Vehicle Detection and Localization Using Convolutional Neural Networks," *IEEE RAL*, vol. 5, no. 2, pp. 2459–2466, April 2020.

[19] Y. Tang, Y. Hu, J. Cui, F. Liao, M. Lao, F. Lin *et al.*, "Vision-Aided Multi-UAV Autonomous Flocking in GPS-Denied Environment," *IEEE Trans. on Ind. Electronics*, vol. 66, no. 1, pp. 616–626, 2019.

[20] M. Vrba *et al.*, "Onboard Marker-Less Detection and Localization of Non-Cooperating Drones for Their Safe Interception by an Autonomous Aerial System," *IEEE RAL*, vol. 4, no. 4, pp. 3402–3409, 2019.

[21] V. Walter, M. Saska, and A. Franchi, "Fast Mutual Relative Localization of UAVs using Ultraviolet LED Markers," in *2018 ICUAS*, 2018.

[22] V. Walter, N.Staub, M. Saska, and A. Franchi, "Mutual Localization of UAVs based on Blinking Ultraviolet Markers and 3D Time-Position Hough Transform," in *CASE*, 2018.

[23] V. Walter, N. Staub *et al.*, "UVDAR System for Visual Relative Localization With Application to LeaderFollower Formations of Multirotor UAVs," *IEEE RAL*, vol. 4, no. 3, pp. 2637–2644, July 2019.

[24] C. W. Reynolds, "Flocks, Herds and Schools: A Distributed Behavioral Model," in *Proceedings of the 14th Annual Conference on Computer Graphics and Interactive Techniques*, ser. SIGGRAPH '87, New York, NY, USA, 1987, p. 2534.

[25] R. Olfati-Saber, "Flocking for multi-agent dynamic systems: algorithms and theory," *IEEE Transactions on Automatic Control*, vol. 51, no. 3, pp. 401–420, 2006.

[26] D. Sakai, H. Fukushima, and F. Matsuno, "Flocking for multirobots without distinguishing robots and obstacles," *IEEE Transactions on Control Systems Technology*, vol. 25, no. 3, pp. 1019–1027, 2017.

[27] H. Zhao, H. Liu, Y. Leung, and X. Chu, "Self-Adaptive Collective Motion of Swarm Robots," *IEEE Transactions on Automation Science and Engineering*, vol. 15, no. 4, pp. 1533–1545, 2018.

[28] T. Baca, M. Petrlik, M. Vrba, V. Spurny, R. Penicka, D. Hert, and M. Saska, "The MRS UAV System: Pushing the Frontiers of Reproducible Research, Real-world Deployment, and Education with Autonomous Unmanned Aerial Vehicles," 2020.

[29] S. Kohlbrecher, O. von Stryk, J. Meyer, and U. Klingauf, "A flexible and scalable SLAM system with full 3D motion estimation," in *2011 IEEE ISSSR Robotics*, pp. 155–160.

[30] P. E. Hart, N. J. Nilsson, and B. Raphael, "A formal basis for the heuristic determination of minimum cost paths," *IEEE Transactions on Systems Science and Cybernetics*, vol. 4, no. 2, pp. 100–107, 1968.