A Minimalistic 3D Self-Organized UAV Flocking Approach for Desert Exploration

Thulio Amorim, Tiago Nascimento, Tomas Baca, Eliseo Ferrante, and Martin Saska

***Abstract*—In this work, we propose a swarm flocking approach for multi-rotor unmanned aerial vehicles (UAVs). Our approach allows the swarm to achieve cohesive and aligned flocking (collective motion). Our method efficiently achieves flocking in a random direction without externally provided directional information exchange (alignment control). The method uses only relative range and bearing throught onboard sensors on the UAV. Our method is also able to flock independent of how steep the terrain is. Furthermore, the algorithm allows for a 3D formation given a random initial height. To implement proximal control in a 3D manner, the Lennard-Jones potential function is used to maintain cohesiveness and avoid collisions between robots. The performance of the approach was tested in real-world conditions by experiments with UAVs. The last part of our experiments presents the implementation of our approach on UAVs that are independent from external positioning system such as GNSS, relying only on a relative visual localization through the use of the UVDAR system previously proposed by our group, thus, demonstrating that our system can be applied in GNSS-denied environments. Finally, to evaluate the degree of alignment and cohesiveness, we used a metric called order and the steady-state value.**

***Index Terms*—Unmanned aerial vehicles, flocking, swarm robotics, self-organization.**

I. Introduction

FLOCKING can be defined as the cohesive and ordered motion of a group of individuals in a common direction [1]. Unlike in formation control, where more elaborate rules are designed in order to achieve a complex design [2], self-organized flocking achieves a formation adaptively by imitating behaviors observed in nature. The first individual-based model of flocking, proposed in [1], consisted in three rules: separation, cohesion, and alignment. In robotics, separation and cohesion are normally achieved via proximal control (simple adaptive formation control), while alignment is achieved by alignment control [3]. Proximal control determines the cohesion of a group of robots with the use of local range-and-bearing information, while alignment control makes robots align in a common direction. Alignment control uses more elaborate sensing mechanisms to obtain the orientation of the robots, for example via local communication [4].

T. Nascimento, T. Baca and M. Saska are with the Department of Cybernetics, Czech Technical University in Prague, Prague, Czech Republic., e-mail: (see http://mrs.felk.cvut.cz/).

T. Amorim, and also T. Nascimento, are with the Lab of Systems Engineering and Robotics (LASER), Department of Computer Systems, Universidade Federal da Paraiba, Brazil.

E. Ferrante is with the Technology Innovation Institute, United Arab Emirates, and with the Vrije Universiteit Amsterdam, Netherlands.

Manuscript received May XX, 2021; revised August XX, 2021.

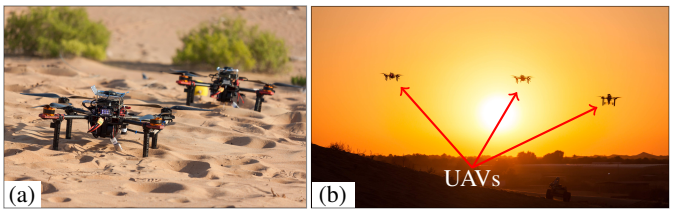

Fig. 1: Swarms of UAVs in the desert (a) using our robotic platform and (b) demonstrating our flocking approach while localized by onboard sensors only on a 3D formation.

In swarm robotics, flocking can be achieved with the use of virtual physics-based design methods [3], where each robot is considered as a virtual particle that exerts virtual forces on other nearby robots. One of the first works using this method was proposed by Kathib [5], who used the concept of an artificial potential field. The virtual physics-based design assumes that the robot is able to perceive and distinguish teammate robots and obstacles, and to estimate the relative distance and bearing from the detected robot/obstacle to calculate their relative position with respect to a local reference frame. Each robot computes a virtual force vector, which is derived from an artificial potential function. The most commonly used function is the Lennard-Jones potential function.

The most common approaches for robot flocking usually use attraction-repulsion functions, e.g. the work of Khatib [5]. Other works also add the alignment rule, e.g. the work of Turgut et al. [6] and Ferrante et al. [4]. Some works even add information about a goal direction, e.g. the work of Tarcai et al. [7], and more recently Shirazi and Jin [8]. All of these approaches have been applied to ground mobile robots, usually in an indoor environment, and some of them only in simulations. Mobile ground robots in an indoor environment are used for testing flocking algorithms, mainly due to the simplicity of the environmental setup.

Basic research in the area of UAV flocking and formation flying was recently studied in [2], [9]. UAV swarm control is a relatively new field of research, and its applications are yet to be explored. One of many possibilities being explored by the authors in this letter is the use of teams of UAVs for inspecting hard-to-access locations, especially in GNSS-denied environments [10]. Applications of this type require swarms to be flexible, and to move adaptively (see Fig. 1).

UAV flocking algorithms are often tested only in simulations, and may omit important features of real UAV systems. The work of Benedetti et al. [11] and the work of Innocente

and Grasso [12] are examples of approaches that achieve very interesting results in simulations, but cannot be used outdoors without significant modifications. Furthermore, these approaches are usually communication-dependent, which is another bottleneck of current large multi-UAV systems and a technical obstacle for the degree of scalability that is required in swarms. In contrast, recent fixed-wing UAV works have emerged with real robot experiments. Kownacki and Ołdziej [13], and Silic and Mohseni [14] have presented interesting studies, where the experiments rely on GPS as the main positioning sensor and on communication between UAVs. Generally, UAV practical approaches use ranging systems based on radio signal transmissions [15], [16], which in turn are subject to the effects of network congestion and interference. For this reason, we consider vision-based approaches more suitable for compact multi-robot groups, especially in uncontrolled outdoor environments. Multi-rotor UAV flocking has also been tested in real conditions. However, these contributions either focus on trajectory planning using indoor UAVs [17], or are GPS and broadcasting communication-dependent [18], [19]. Another recent work [20] has achieved flocking indoors with a swarm of up to 30 physical UAVs. This can be considered one of the most impressive achievements so far. Unlike our work, however, the model used alignment control, and therefore was not based purely on proximal control, as in the proposed approach. Achieving coordinated swarm behaviors without external sensing and computation is a challenging task, as has been very well explained in a recent survey [21].

The current state-of-the-art presents very important contributions and have provided useful foundations. However, to the best of our knowledge, our work is the first method based only on proximal control for ordered and cohesive 2D/3D flocking in real UAV environments applied in either GNSS or GNSS-denied environments. Thus, we can list our contributions as:

- a 3D cohesive flocking method based only on proximal control;
- the method is able to enable the UAVs to fly even on very irregular terrains such as dunes;
- the method can be applied to GNSS-defined environment, relying only on relative neighbor position measurement;
- flocking is achieved using an non-holonomic constraints, which are needed for proximal control based flocking, on unmanned multirotor aerial vehicles;
- this method has been also integrated with our double layer control architecture in a manner that imitates non-holonomic behavior in real-world conditions.

II. UAV Flocking Approach

Multi-robot group control methods based on potential functions are generally composed of three different term-functions: a proximal term, an alignment term, and an optional goal direction term, which is needed when the swarm is required to steer towards a specific target. In our work, we propose a flocking control function that uses only the proximal term ($\mathscr{P}$) in order to converge and move the UAVs into a unified direction.

Definition 1. *In a flocking of n robots, a i^{th} robot, where $i = 1..n$, is called the focal robot F_r.*

Thus, the proximal term function ($\mathscr{P}$) of the focal robot F_r results in a vector that enables the focal robot to maintain the desired distance from other neighbor robots while helping on keeping the group cohesiveness. The proximal terms encapsulates both the attraction and the repulsion behaviors, commonly seen in the potential fields approaches and other swarming approaches [3]. The proximal term function is widely used to achieve cohesive flocking. By using this function, a UAV maintains a desired distance from other neighbor robots while keeping a cohesive formation. By applying the proximal term in the 2-dimensional space, the range and bearing is the only information required to compute this term. Thus, to calculate $\mathscr{P}$ we first must estimate the relative pose of each neighbor, which in turn is required for obtaining the relative range d_n and bearing ϕ_n, and it is expressed in the body frame of reference of the focal robot.

Assumption 1. *In a flocking of n robots, the n^{th} neighbor is a robot within the sensor range delimited by the interaction range gain λ, where the range d_n of the n^{th} neighbor that the focal robot F_r takes into account is such that $d_n \leq d_{des}\lambda$.*

Furthermore, the relative position is calculated as:

$$\begin{aligned} x_r &= x_n - x_f \\ y_r &= y_n - y_f \end{aligned} \tag{1}$$

where x_n is the coordinate of the n^{th} neighbor in the X-Axis, x_f is the coordinate of the focal robot (F_r) in the X-Axis, x_r is the relative x-coordinate of the neighbor robot on the body frame of F_r, y_n is the coordinate of the n^{th} neighbor in the Y-Axis, y_f is the coordinate of the focal robot (F_r) in the Y-Axis, and y_r is the relative y-coordinate of the neighbor robot on the body frame of F_r.

Finally, we also calculate the coordinate of the neighbor in the Z-Axis on the body frame of F_r. When using the approach in the 2-dimensional space, this information is used only for maintain all UAVs at the same height. A 3-dimensional case usually is either stimulated by initializing the UAVs in different altitudes, or when an external ground irregularity forces the UAVs to be ina 3D shape. When estimating the state of F_r (and its neighbors) using the Barometer/Garmin Lindar (global positioning system) or the UVDAR sensors (relative positioning system) [22], the effect of this behavior is that when one UAV goes up to avoid an irregular terrain, the others follows it for also avoid the bump in the terrain even without seen it. Another advantages of this behavior, when using the UVDAR estimation, is that staying at the same height keeps the UVDAR LEDs of the neighbor robots in the range of view of the UVDAR Cameras of the focal robot making easier for estimate the position of the neighbors. However, an omnidirectional UVDAR system is currently under development and the extension of our approach is the subject of another work currently in development.

Thus, since it is not possible to retrieve this information in some occasions when the UAVs are in a GNSS-denied environments, we use the local origin coordinate frame. The

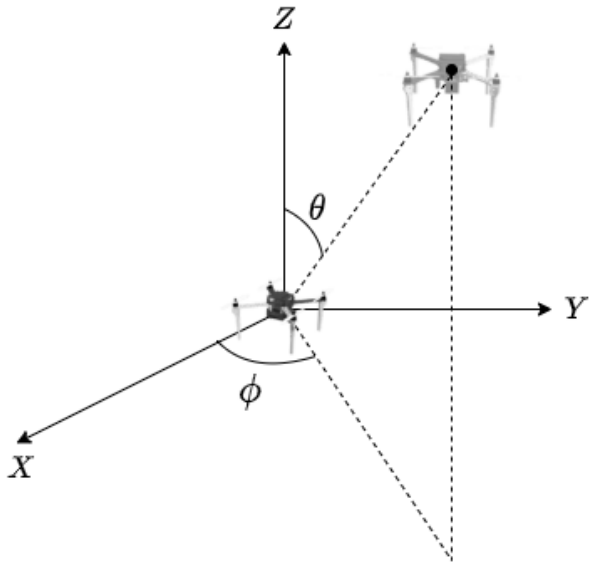

Fig. 2: Representation of the bearing ϕ and the inclination θ in the body frame of the focal robot

local origin frame is the coordinate frame with center and orientation coincident with the starting point and orientation of the UAV. The usage of the local origin frame will only provide the correct estimation if all the UAVs starts at the same height. Thus, the coordinate of the neighbor in the Z-Axis on the body frame of F_r is calculated as follows:

$$z_r = z_n - z_f \tag{2}$$

where z_n is the coordinate of the n^{th} neighbor in the Z-Axis, z_f is the coordinate of the focal robot in the Z-Axis, and z_r is the coordinate of the neighbor robot in the Z-Axis on the body frame of F_r.

Note that the previous steps are not performed when using the UVDAR estimation since the sensor already estimated the pose of the neighbor in the body frame of the robot from both UVDAR cameras, requiring only the transformation to the body frame of the focal robot coordinate frame. Thus, we can calculate the range of the n^{th} neighbor d_n as:

$$d_n = \sqrt{x_r^2 + y_r^2 + z_r^2} \tag{3}$$

To calculate the bearing is not trivial since we can have two different angles. The first angle is the normal bearing ϕ_n. The bearing of the n^{th} neighbor ϕ_n can be calculated as:

$$\phi_n = atan2(y_r, x_r) \tag{4}$$

As group can be in a 3D shape at any time, the usage of the angle between the positive Z-Axis and the line segment between the origin of the focal coordinate frame to its neighbor position θ (see Fig. 2) needs also to be calculated. The addiction of θ to the proximal term function makes possible for each UAV to going up or down. The angle θ is calculated as:

$$\theta_n = \begin{cases} 0 & \text{, if } z_r = 0 \\ atan2(\sqrt{x_r^2 + y_r^2}, z_r) & \text{, otherwise} \end{cases} \tag{5}$$

where θ_n is the angle θ of the n^{th} neighbor.

After estimating the range and bearing of each neighbor, the proximal term is calculated as:

$$\mathscr{P} = \sum_{n=1}^{m_p} p_n(d_n) e^{i\phi_n + j\theta_n} \tag{6}$$

where m_p is the number of neighboring UAVs perceived by the focal robot within the maximum interaction distance.

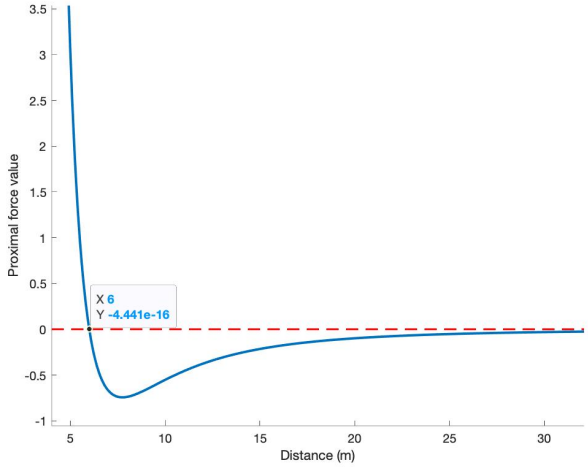

Fig. 3: The used Lennard-Jones function.

In the above equation, the term $p_n(d_n)$ is the magnitude of the proximal vector. In our case we use Lennard-Jones potential (see Fig. 3) that can be calculated as:

$$p_n(d_n) = -\frac{\partial P(d_n)}{\partial d_n} = -\frac{4\alpha\varepsilon}{d_n}\left[2\left(\frac{\sigma}{d_n}\right)^{2\alpha} - \left(\frac{\sigma}{d_n}\right)^{\alpha}\right] \tag{7}$$

where $P(d_n)$ is a virtual potential function, ε is the strength of the potential function which determines its depth, α is the steepness of the potential function, and σ is the amount of noise calculated by equation (8). Note that the minimum value of ε should be when $-\varepsilon = d_i$. Thus, σ can be expressed as

$$\sigma = \frac{d_n}{2^{\frac{1}{\alpha}}}. \tag{8}$$

A. Magnitude-dependent Flocking Motion Control

To achieve the advantages of magnitude-dependent flocking motion control, all UAVs need to satisfy non-holonomic constraints similar to the constraints of differential drive robots. But in order to pass the information calculated to the UAV control system, we need first to decompose the flocking control vector $\mathscr{P}$ into a motion control vector. We can first decompose the value of $\mathscr{P}$ into three componentes of the force: f_x, f_y and

f_z. We call f_x, f_y and f_z the projection of the flocking control vector $\mathscr{P}$ on the XYZ-plane of the body reference frame of the focal robot. Here, f_x, f_y and f_z can be calculated as

$$\begin{aligned} f_x &= \sum_{n=1}^{m_p} p_n(d_n)\sin\theta_n\cos\phi_n \\ f_y &= \sum_{n=1}^{m_p} p_n(d_n)\sin\theta_n\sin\phi_n \\ f_z &= \begin{cases} 0 & \text{, if } z_r = 0 \\ \sum_{n=1}^{m_p} p_n(d_n)\cos\theta_n & \text{, otherwise} \end{cases} \end{aligned} \tag{9}$$

The next steps are for convert the projections f_x, f_y and f_z into a desired movement. We start by multiplying each projection by a gain as the following

$$\begin{aligned} u &= \kappa_1 f_x + B_s \\ v &= \kappa_2 f_y \\ \omega &= \kappa_3 f_z \end{aligned} \tag{10}$$

where k_i, with $i = 1..3$, are the proportional gains to transform from the respective forces vector component into the additive speed component, and B_s is the maximum forward biasing speed in the X-Axis.

The linear speed u is assumed to be directly proportional to the x component of the vector (i.e. f_x), which results in the forward movement of the robot in the body reference frame, the angular movement v will be directly proportional to the y component of the vector (i.e. f_y) and the horizontal movement ω will be directly proportional to the z component of the vector (i.e. f_z). Note here that the larger the value of f_x, the faster the robot moves forward, the larger the value of f_y, and the faster the robot turns, and the larger the value of f_z, the faster the robot moves up or down. Although this approach also allowed the robot to move backward, it tends to move forward due to the existence of the forward biasing movement of B_s.

Finally, we send to the dual-layer MPC positioning control system of the UAV the desired position ($\mathbf{r}_d = [x_d, y_d, z_d]^\intercal$) and heading ($\eta_d$) references in the global frame as input values by updating its current position using the linear movement u and horizontal movement ω as follows:

$$\begin{aligned} x_d(k+1) &= x_d(k) + u * \cos\eta \\ y_d(k+1) &= y_d(k) + u * \sin\eta \\ \eta_d(k+1) &= \eta_d(k) + \omega \end{aligned} \tag{11}$$

where η is the current heading of the focal robot.

The Z-Axis is a special case. We must first calculate the needed height gain (h_{push}). We must also take into account the minimum allowed height (h_{min}) for the case the UAV goes below the minimum height. Thus,

$$h_{push} = \max(0, h_{min} - z_f) \tag{12}$$

The initial desired height h_d is then calculated using the angular movement v as follows

$$h_d(k) = \begin{cases} z_d(k) + h_{push} & \text{, if } (z_f + h_{push} + v) \leq h_{min} \\ z_d(k) + h_{push} + v & \text{, otherwise} \end{cases} \tag{13}$$

Finally, we must take into account if it is a 2D formation or a 3D formation, respectively. Thus, the final reference height z_d is as follows

$$z_d(k+1) = \begin{cases} h_d(k) + z_n & \text{, if } z_r = 0 \\ h_d(k) & \text{, otherwise} \end{cases} \tag{14}$$

One last problem must be taken into account before sending the desired references to the MPC tracker. Even when stabilizing the UAV, the presence of rapidly variation on the bearing caused by high forces due to big differences in angles still presents a major problem. In order to minimize this problem, we re-calculate the observed bearing with a spherical linear interpolation (SLERP) technique. SLERP is a popular technique for interpolating between two 3D rotations while producing smooth paths. Thus, the recalculation of the bearing transforms the robot physical heading angle:

$$q_S(k) = \left[\frac{\sin((1-\gamma)\theta)}{\sin\theta}\right] q_\eta(k) + \left[\frac{\sin(\gamma\theta)}{\sin\theta}\right] q_S(k-1), \tag{15}$$

where $q_S(k)$ is the quartenion of the smoothed heading angle in instant k, q_η is the quartenion of the physical heading angle, θ is the angle between the two quartenions, and γ is an interpolation coefficient. For $\theta \approx 0$, the SLERP equation is

$$q_S(k) = (1-\gamma) q_\eta(k) + \gamma q_S(k-1), \tag{16}$$

thus, through quaternion q_S we find η_d, which is the smoothed final reference heading angle.

B. UAV Model application

Multi-rotor UAVs may be considered as underactuated vehicles with holonomic mobility. When designing a reliable decentralized flocking mechanism, we have to assume that the UAVs are subjected to localization problems with an error of position estimation by GPS in the order of meters. Communication of the relative position between UAVs within the flock also suffers drawbacks due to communication delays, intermittent communication interruptions, and mainly scalability to enable the operation of large swarms, which is the main aim of the swarm community. We recently developed a distance and bearing estimator based on ultra-violet light, and cameras called UVDAR [22] to solve this issue.

Nevertheless, the MRS system allows for both holonomic and nonholonomic movements. To illustrate the model of the robot that satisfies this statement, let us use the dynamical model of a multirotor aerial vehicle presented in the work of Lee et al. [24]. To better analyse this model, we must be reminded that here, we depict the world frame $\mathscr{W} = \{\hat{\mathbf{e}}_1, \hat{\mathbf{e}}_2, \hat{\mathbf{e}}_3\}$ in which the 3D position and the orientation of the UAV body is expressed. The body frame $\mathscr{B} = \{\hat{\mathbf{b}}_1, \hat{\mathbf{b}}_2, \hat{\mathbf{b}}_3\}$ relates to $\mathscr{W}$ by the translation $\mathbf{r} = [x, y, z]^\intercal$ and by rotation $\mathbf{R}^\intercal$. The

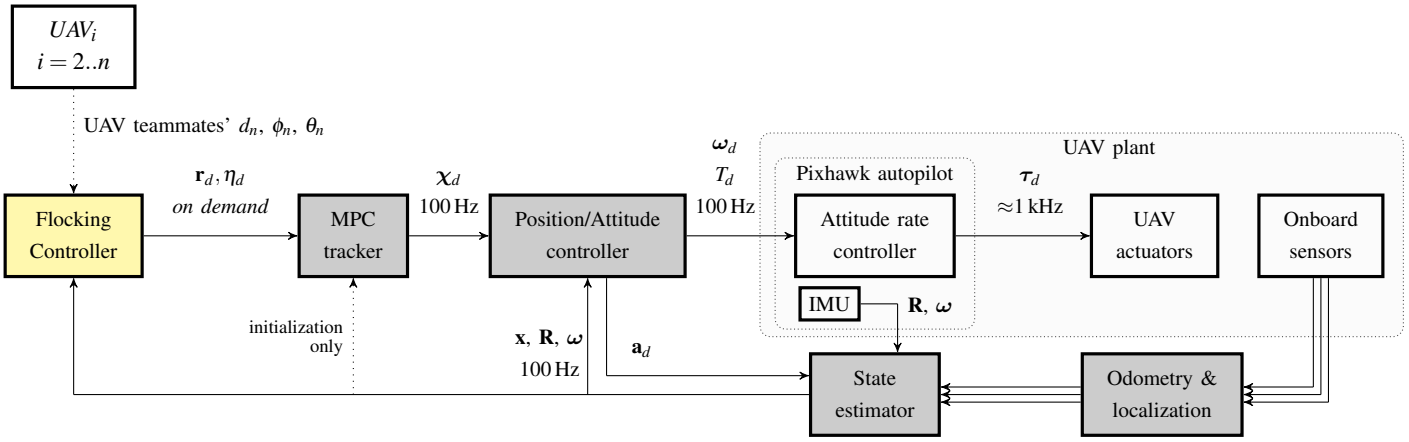

Fig. 4: A diagram of the system architecture: *Flocking Controller* in yellow is what we propose in this work. It is build upon the MRS system software (blocks in gray) [23] and it supplies the desired reference (position $\mathbf{r}_d$ and heading η_d) to the MRS system. Within the MRS system we have a first layer with an MPC tracker that processes the desired reference and gives a full-state reference to the position/attitude controller. *MPC tracker* creates a smooth and feasible reference $\boldsymbol{\chi}$ for the reference feedback controller. The feedback *Position/Attitude controller* produces the desired thrust and angular velocities (T_d, $\boldsymbol{\omega}_d$) for the Pixhawk embedded flight controller (Attitude rate controlller). The *State estimator* fuses data from *Onboard sensors* and *Odometry & localization* methods to create an estimate of the UAV translation and rotation ($\mathbf{x}$, $\mathbf{R}$).

UAV heading vector $\mathbf{h}$, which is a projection of $\hat{\mathbf{b}}_1$ to the plane $span(\hat{\mathbf{e}}_1, \hat{\mathbf{e}}_2)$, forms the heading angle η. Thus, the UAV nonlinear model can be expressed as having a translation part:

$$m\ddot{\mathbf{r}} = f_T \mathbf{R}\hat{\mathbf{e}}_3 - mg\hat{\mathbf{e}}_3, \tag{17}$$

where m is the UAV mass, g is the gravity acceleration, f_T is the thrust force created by the propellers in the direction of $\hat{\mathbf{b}}_3$, $\mathbf{r} = [x, y, z]^\mathsf{T}$ is the position of the center of the mass of a UAV in the world frame, $\ddot{\mathbf{r}} \in \mathbb{R}^3$ is the acceleration of the center of the mass of a UAV in the world frame, $\mathbf{R} \in \mathrm{SO}(3) \subseteq \mathbb{R}^{3\times 3}$ is the rotation matrix from the body frame of a UAV to the world frame, and a rotational part

$$\dot{\mathbf{R}} = \Omega\mathbf{R}, \tag{18}$$

where Ω is the tensor of angular velocity, under the condition $\Omega\mathbf{v} = \boldsymbol{\omega} \times \mathbf{v}, \forall \mathbf{v} \in \mathbb{R}^3$. The UAV is affected by downward gravitational acceleration with $g \in \mathbb{R}$.

However, as we are focused on non-agility flight, we separately consider and estimate the azimuth of the $\hat{\mathbf{b}}_1$ axis in the world as the UAV *heading*. Under the condition of $|\hat{\mathbf{e}}_3^\mathsf{T}\hat{\mathbf{b}}_1| > 0$, we define the heading as

$$\eta = \mathrm{atan2}\left(\hat{\mathbf{b}}_1^\mathsf{T}\hat{\mathbf{e}}_2, \hat{\mathbf{b}}_1^\mathsf{T}\hat{\mathbf{e}}_1\right) = \mathrm{atan2}\left(\mathbf{h}_{(2)}, \mathbf{h}_{(1)}\right). \tag{19}$$

The heading is a more intuitive alternative to the widely-used *yaw* angle as one of the 4 controllable DOFs. It is possible to use the *yaw*, but with the assumption that the tilt of the UAV ($\cos^{-1}\ \hat{\mathbf{b}}_3^\mathsf{T}\hat{\mathbf{e}}_3$) is low, near horizontal. We do not use *Euler angles*, due to the overwhelming number of conventions, which may lead to a misunderstanding. Generally, the widely-used *yaw* angle (as in *Euler angles*, *Tait-Bryan angles* [25]) has no direct meaning with respect to the particular orientation of any of the body axis in any of the conventions, since the final orientation also depends on the remaining two rotations (pitch, roll). We therefore define the heading vector by the $\hat{\mathbf{b}}_1$ axis as

$$\mathbf{h} = \left[\mathbf{R}_{(1,1)}, \mathbf{R}_{(2,1)}, 0\right]^\mathsf{T} = \left[\mathbf{b}_1^\mathsf{T}\hat{\mathbf{e}}_1, \mathbf{b}_1^\mathsf{T}\hat{\mathbf{e}}_2, 0\right]^\mathsf{T} \tag{20}$$

and its normalized form

$$\hat{\mathbf{h}} = \frac{\mathbf{h}}{\|\mathbf{h}\|} = [\cos\eta, \sin\eta, 0]^\mathsf{T}. \tag{21}$$

Therefore, we can note that all these factors directly influence the estimate of range and bearing between neighboring robots, the bearing estimation being the most problematic in teams of UAVs. Furthermore, for the algorithm to work, the robots must behave in a non-holonomic manner.

To be able to achieve self-organized ability towards the intended collective motion, let us first apply motion constraints protecting against unwanted side-motion of UAVs. In comparison with UGVs, UAVs are highly unstable and tend to drift due to errors in state estimation. The control architecture proposed in our previous work (MPC tracker [26] and MPC position controller [23]) was therefore used.

C. Control Architecture

The platforms used in this work consist of several interconnected subsystems, as depicted in Fig. 4. The *Flocking* block embeds the above discussed approach and supplies the time-parametrized sequence of the desired position and heading. This information is then sent to an *MPC tracker* based on our previous work [26], where the desired position and heading are processed. In this block, the desired position and heading are converted into a feasible, smooth, and evenly-sampled full-state control reference. This control reference contains the desired position, its derivatives up to the jerk, the heading, and the heading rate, supplied at 100 Hz. Then, the full-state control reference is used by a *MPC position controller* (based on our previous work in [23]) to provide feedback control of the translational dynamics and the orientation of the UAV. This

block creates an attitude rate ω_d and a thrust command T_d, which are sent to an embedded flight controller[1]. The flight controller encapsulates the underlying physical UAV system with motors and motor ESC, and creates 4 new controllable DOFs: the desired angular speed around $(\hat{\mathbf{b}}_1, \hat{\mathbf{b}}_2, \hat{\mathbf{b}}_3)$ and the desired thrust $\langle 0, 1 \rangle$ of all propellers. Finally, onboard sensor data (e.g., position measurements from Barometer/Garmin Lidar, velocity measurements from visual odometry, etc.) are processed by the *Odometry & Localization* and the *State estimator* blocks and supply the needed information to all these previous blocks.

III. Convergence Analysis

TABLE I: Parameters values

Parameter	Description	Value
B_S	Maximum forward speed	0.3 m/s
κ_1	Linear gain	0.5
κ_2	Angular gain	0.2
κ_3	Height gain	0.1
α	Steepness of potential function	2
ε	Strengh of potential function	6
d_{des}	Desired inter-robot distance	6 m
D_p	Maximum interaction range	7.2 m
λ	Interaction range gain	1.8
γ	Interpolation coefficient	0.95
h_{min}	Minimum desired height	2.0 m

IV. Results

Swarming algorithms are intended for large group of robots (≥ 100). When using ground robots, to gather this high number of robots is somehow feasible. However, in our experiments, we aim to analyze the behavior of autonomous middle-size UAVs, where real disturbances from outdoor environments can disturb the group cohesiveness. In this case, the use of a large number of UAVs is not a simple task. Thus, we performed real robot experiments with 10 robots and therefore succeeded in proving the efficiency of our proposed approach.

The aim of the experiments presented here is to verify local interactions between UAVs without the necessity of sharing position information among the robots in the fleet. We aim to show whether the flocking mechanism for UAVs can achieve similar properties as in the work on UGVs [4], where scalability to larger swarms was proven. The experiments used the parameters observed in Table I.

A. Metrics

To analyze the effectiveness of our approach, we use two metrics. The first, the order metric ψ, measures the degree of alignment of the orientations within the swarm. After computing the vectorial sum of the headings of all N robots,

$$\mathbf{b} = \sum_{i=1}^{N} e^{j\phi_i}, \tag{22}$$

the order can be calculated as

$$\psi = \frac{1}{N}||\mathbf{b}||. \tag{23}$$

A swarm having a common heading will result in a value of $\psi \approx 1$. A swarm where the UAVs are pointing in different directions will result in a value of $\psi \approx 0$.

In this section, we also analyze the steady-state value that is reached for a given metric (in our case the order metric). The steady-state metric is the asymptotic value reached by the order metric during the experiment, defined as

$$\bar{\mu} = \frac{\sum_{(t=T-100)}^{T} \psi_t}{100}. \tag{24}$$

B. Real Robot Experiments

V. Conclusion

In this work, we have presented an efficient and simple method for UAV flocking based only on a proximal function. This function has been used to enable the UAVs not only to maintain a cohesive flock but to move into an arbitrary direction. We performed two sets of experiments, one using GPS and communication devices to exchange UAV positions and estimate the range and bearing, while the other set of experiments used only UVDAR to estimate the range and the bearing with no communication between UAVs. Our approach managed to enable the robots to converge towards the flock and maintain the cohesiveness of the group. It also enabled the UVAs to move into an arbitrary direction. In this way, we achieved self-organized flocking with limited sensory information for aerial robots with high dynamics. Within the novel flocking approach itself, we have provided a framework and guidelines to enable fundamental achievements of swarm research to be integrated into UAV systems working in real-world conditions.

[1]Pixhawk is used for experimental verification.

References

[1] 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: Association for Computing Machinery, 1987, p. 25–34.

[2] M. Saska, D. Hert, T. Baca, V. Kratky, and T. Nascimento, "Formation Control of Unmanned Micro Aerial Vehicles for Straitened Environments," *Autonomous Robots*, vol. 44, pp. 991–1008, 2020.

[3] M. Brambilla, E. Ferrante, M. Birattari, and M. Dorigo, "Swarm robotics: A review from the swarm engineering perspective," *Swarm Intelligence*, vol. 7, pp. 1–41, 2013.

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

[5] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," *The International Journal of Robotics Research*, vol. 5, no. 1, pp. 90–98, 1986.

[6] A. E. Turgut, H. Çelikkanat, F. Gökçe, and E. Şahin, "Self-organized flocking in mobile robot swarms," *Swarm Intelligence*, vol. 2, p. 97–120, 2008.
[7] N. Tarcai, C. Virágh, D. Ábel, M. Nagy, P. L. Várkonyi, G. Vásárhelyi, and T. Vicsek, "Patterns, transitions and the role of leaders in the collective dynamics of a simple robotic flock," *J. of Statistical Mechanics: Theory and Experiment*, vol. 2011, no. 04, apr 2011.
[8] A. R. Shirazi and Y. Jin, "Regulated morphogen gradients for target surrounding and adaptive shape formation," *IEEE Transactions on Cognitive and Developmental Systems*, vol. In Press, pp. 1–11, 2020.
[9] M. Saska, "Large Sensors with Adaptive Shape Realised by Self-stabilised Compact Groups of Micro Aerial Vehicles," in *Robotics Research*. Springer International Publishing, 2020, pp. 101–107.
[10] T. Roucek *et al.*, "Darpa subterranean challenge: Multi-robotic exploration of underground environments," in *MESAS*, 2019.
[11] M. De Benedetti, F. D'Urso, G. Fortino, F. Messina, G. Pappalardo, and C. Santoro, "A fault-tolerant self-organizing flocking approach for uav aerial survey," *Journal of Network and Computer Applications*, vol. 96, pp. 14 – 30, 2017.
[12] M. S. Innocente and P. Grasso, "Self-organising swarms of firefighting drones: Harnessing the power of collective intelligence in decentralised multi-robot systems," *Journal of Computational Science*, vol. 34, pp. 80 – 101, 2019.
[13] C. Kownacki and D. Ołdziej, "Fixed-wing uavs flock control through cohesion and repulsion behaviours combined with a leadership," *Int. J. of Advanced Robotic Systems*, vol. 13, no. 1, p. 36, 2016.
[14] M. Silic and K. Mohseni, "Field deployment of a plume monitoring uav flock," *IEEE RA-L*, vol. 4, no. 2, pp. 769–775, 2019.
[15] T. Nguyen, Z. Qiu, T. H. Nguyen, M. Cao, and L. Xie, "Distance-Based Cooperative Relative Localization for Leader-Following Control of MAVs," *IEEE RA-L*, vol. 4, no. 4, pp. 3641–3648, 2019.
[16] T. Bhavana, M. Nithya, and M. Rajesh, "Leader-follower co-ordination of multiple robots with obstacle avoidance," in *SmartTechCon*, 2017.
[17] W. Hönig, J. A. Preiss, T. K. S. Kumar, G. S. Sukhatme, and N. Ayanian, "Trajectory planning for quadrotor swarms," *IEEE Transactions on Robotics*, vol. 34, no. 4, pp. 856–869, 2018.
[18] S. H. Arul, A. J. Sathyamoorthy, S. Patel, M. Otte, H. Xu, M. C. Lin, and D. Manocha, "Lswarm: Efficient collision avoidance for large swarms with coverage constraints in complex urban scenes," *IEEE Robotics and Automation Letters*, vol. 4, no. 4, pp. 3940–3947, 2019.
[19] C. Virágh, G. Vásárhelyi, N. Tarcai, T. Szörényi, G. Somorjai, T. Nepusz, and T. Vicsek, "Flocking algorithm for autonomous flying robots," *Bioinspiration & Biomimetics*, vol. 9, no. 2, 2014.
[20] 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.
[21] M. Coppola, K. N. McGuire, C. De Wagter, and G. C. H. E. de Croon, "A survey on swarming with micro air vehicles: Fundamental challenges and constraints," *Frontiers in Robotics and AI*, vol. 7, p. 18, 2020.
[22] W. Walter, N. Staub, A. Franchi, and M. Saska, "UVDAR System for Visual Relative Localization With Application to Leader–Follower Formations of Multirotor UAVs," *IEEE RA-L*, vol. 4, no. 3, pp. 2637–2644, 2019.
[23] 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," *Journal of Intelligent & Robotic Systems*, vol. 102, no. 26, pp. 1–28, May 2021.
[24] T. Lee *et al.*, "Geometric tracking control of a quadrotor UAV on SE(3)," in *IEEE CDC*, 2010, pp. 5420–5425.
[25] J. Diebel, "Representing attitude: Euler angles, unit quaternions, and rotation vectors," *Matrix*, vol. 58, no. 15-16, pp. 1–35, 2006.
[26] T. Baca, D. Hert, G. Loianno, M. Saska, and V. Kumar, "Model Predictive Trajectory Tracking and Collision Avoidance for Reliable Outdoor Deployment of Unmanned Aerial Vehicles," in *IEEE/RSJ IROS*. IEEE, 2018, pp. 1–8.