

# OBSTACLE AWARE REYNOLDS FLOCKING IN INDOOR ENVIRONMENTS

A. Felić\*, L. Tautz\*, S. Abele\*, M. Martens\*, M. Uijt de Haag\*

\* Technische Universität Berlin, Institute of Aeronautics and Astronautics, Chair of Flight Guidance and Air Traffic,  
10587 Berlin, Germany

## Abstract

This paper investigates the use of the Reynolds flocking algorithm to control the behavior of small Unmanned Aerial Systems (sUAS) within a swarm, focusing on their interactions and dynamics. The motivation for applying Reynolds flocking in swarm navigation is to optimize UAS performance in complex environments. Rule-based approaches, such as the Reynolds algorithm, present a practical solution, particularly in obstacle-rich settings where traditional optimization methods are limited by computational constraints. Through the adaptation of Reynolds flocking principles and extensive testing in simulated environments, this study advances the understanding and practical implementation of swarm intelligence in sUAS technology.

## Keywords

Reynolds Flocking; sUAS Indoor Navigation; sUAS Swarming; Obstacle Avoidance in Indoor Environment

## 1. INTRODUCTION

The rapid technological advancements in the field of unmanned aircraft systems (UAS) have, in recent years, led to the exploration of new applications within civilian and commercial sectors. Small UAS (sUAS) have garnered particular attention due to their significant potential in areas such as package delivery, surveillance, and various industrial operations. These systems offer several advantages, including operational flexibility, environmental sustainability, reduced ground risk and the efficient use of human resources. Despite these benefits, the deployment of sUAS in urban environments presents substantial challenges. A critical limiting factor is the quality of GNSS reception in densely built-up areas. Phenomena such as multipath effects and signal shadowing significantly degrade the reliability of GNSS signals, increasing the risk of collisions with buildings and potential hazards to people nearby. Additionally, the limited battery life of sUAS constrains their operational duration, necessitating frequent and inefficient returns for recharging, particularly during demanding missions such as surveillance operations.

A promising approach to overcoming these challenges is the implementation of swarm navigation, also known as swarming, in sUAS operations. This strategy leverages the coordinated behavior of multiple sUAS that interact locally with one another, enabling the swarm to complete complex tasks even in GNSS-denied environments. Notably, a high-flyer low-flyer concept can mitigate the issue of limited GNSS reception. In this scenario, high-flying sUAS feature GNSS receivers and relay their position estimates along with an estimate of the relative position of the swarm members to the remaining sUAS in the swarm allowing for an absolute position estimate of the low-flying sUAS. This cooperative method enhances both the accuracy and safety of sUAS operations in urban environments while offering a potentially more efficient solution for meeting demanding mission requirements which was shown by the authors in [1]. Moreover, swarming offers potential benefits in scenarios such as search and rescue missions, environmental mapping, and large-scale infrastructure monitoring, where

multiple sUAS must cover extensive areas efficiently and respond to dynamic conditions.

Decentralized swarm navigation and task allocation, which is required to conduct the above mentioned missions, is a complex challenge that is the subject of current research. Within this paper, the principles of Reynolds' flocking are extended to solve the task of collision-free swarm navigation. The underlying distributed behavioral model was first presented by Reynolds in 1987 in [2], which is inspired by the collective behaviors observed in nature, such as the coordinated movement of bird flocks, fish schools, and other group-living organisms. To translate these behaviors into an algorithm, specific rules must be defined to dictate the actions of individual sUAS within the swarm. A decentralized control strategy is employed, which harnesses the benefits of leaderless swarm intelligence, offering enhanced robustness and flexibility compared to centralized systems. In this decentralized model, each sUAS follows simple, defined rules based on its interactions with neighboring sUAS. These core rules typically include cohesion, separation, and alignment. Cohesion encourages sUAS to stay close to their neighbors, maintaining group unity, while separation ensures that sUAS maintain a safe distance from one another to avoid collisions. Alignment helps synchronize the movement of individual sUAS with the overall direction of the swarm. Together, these mechanisms allow the swarm to function as a cohesive unit, with complex group behaviors emerging naturally from individual, rule-based actions. To further enhance the swarm's capabilities, additional rules such as trajectory following and obstacle avoidance are integrated. Trajectory following ensures that the swarm can navigate along predetermined paths, while obstacle avoidance enables the sUAS to detect and steer clear of potential hazards in their environment. These supplementary behaviors enhance the safety and efficiency of the swarm, allowing it to not only maintain formation but also to dynamically adapt to changing conditions and environments. By incorporating these principles, swarm navigation enables UAS systems to perform complex tasks more effectively, while reducing the reliance on traditional centralized control systems. This de-

centralized, nature-inspired approach facilitates adaptive, and scalable sUAS operations.

This paper investigates the application of the Reynolds flocking algorithm for controlling sUAS swarms. The algorithm is tested in various environments, including obstacle-dense scenarios, to demonstrate the potential of nature-inspired swarm behavior in enhancing the efficiency and effectiveness of sUAS operations. Section II delineates the theoretical framework of Reynolds' flocking algorithm, including the underlying behavioral rules. Section III elaborates on the implementation methodology of the swarming algorithm, which is applied in the flight tests and simulations described in Section IV. Finally, Section V presents and critically evaluates the preliminary results of the proposed algorithm.

## 2. REYNOLDS FLOCKING

For the purpose of controlling a swarm, phenomena from nature, such as the behavior of bird flocks, are considered and applied to the existing task following the initial work of Reynolds from 1987 [2].

Developing an UAS swarm behavior according to the Reynolds rules involves implementing the principles of cohesion, separation and alignment between the individual sUAS. This enables coordinated movements within the swarm [3].

To identify the requirements for the simulation and flight testing, it is important to understand how the participating sUAS in the swarm communicate during flight, as well as the theory on which the swarming principles are based. In this case, the approach of Reynolds and the boids model is used given complete situational awareness of the other vehicles. In addition, obstacle avoidance and path planning are taken into account.

### 2.1. Boids Model by Reynolds

In Reynolds' approach [4], a model that is based on nature is provided, such as flocks of birds or schools of fish. The observation enables the establishment of three basic rules of behavior, as presented below. Every individual sUAS can be seen as a boid, which behaves random if no direction or force is applied [5]. If a force is applied, the boids are then able to find a configuration, which is (sub-)optimal for the specified parameters [5].

Depending on the given rule, an interaction range  $f$  is specified to determine the individual attraction, repulsion, and orientation area of the flock. The interaction range  $f$  is a set parameter in the swarm that constrains the size of the influenced swarm [6].

The cohesion rule describes the flock centering. It describes the effect that boids want to stay close to the center of the flock, which is determined by their neighbors in range of the attraction area [5, 7]. If one boid is too far away, the cohesion rule forces the sUAS to get closer to the center of the flock [8]. To implement this rule in a simulation, the center of the swarm is calculated as in Equation 1 [3].

$$(1) \quad \mathbf{F}_{ci} = \sum_{\forall b_j \in f} \frac{\mathbf{p}_j}{N}$$

Note,  $p_j(x_j, y_j)$  is the position of boid  $j$  and  $N$  the total number of boids.  $\mathbf{F}_{ci}$  is the center of the swarm  $f$  [3].

With the known center, the cohesion vector for boid  $i$  can be calculated by subtracting the position of boid  $i$  from the

position of the center, as shown in Equation 2 [3].

$$(2) \quad \mathbf{C}_i = \mathbf{F}_{ci} - \mathbf{p}_i$$

The cohesion vector gives the magnitude and direction of boid  $i$  towards the center of the swarm [8]. By implementing only the cohesion rule, the swarm members would converge to a single center of mass causing collisions in real-life [3].

Therefore, the counter-oriented separation rule describes the separation between each sUAS to avoid conflicts. To avoid possible collisions, the boids should stay as far away from each other as necessary to maintain a safe distance [5, 7]. In Equation 3 the separation vector for boid  $i$  is determined by taking every boid  $j$  in the repulsion zone into account [3, 8]. The repulsion zone is the threshold at which the separation rule is activated.

$$(3) \quad \mathbf{S}_i = - \sum_{\forall b_j \in f} (\mathbf{p}_i - \mathbf{p}_j)$$

The alignment rule includes velocity matching. The boids match the direction and speed of their neighbors in one of the following ways [5, 7]. They can move towards or away from each other, as well as in the same or opposite direction [5, 7]. This leads to a coordinated movement of each boid [8]. To calculate the alignment, the first step is to determine the average velocity vector for boid  $i$ , shown in Equation 4 [3].

$$(4) \quad \mathbf{F}_{vi} = \sum_{\forall b_j \in f} \frac{\mathbf{v}_j}{N}$$

Therefore, the average velocities from boid  $i$  neighbors  $\mathbf{v}_j$  in range of the orientation area are taken into account [3, 8]. In the next step Equation 5 the alignment vector is calculated [3].

$$(5) \quad \mathbf{A}_i = \mathbf{F}_{vi} - \mathbf{v}_i$$

Without this rule, the boids wouldn't mimic their neighbors and the flocking behavior couldn't be observed [3].

Figure 1 depicts the three basic principles mentioned above to describe the movements of a single boid in a swarm.

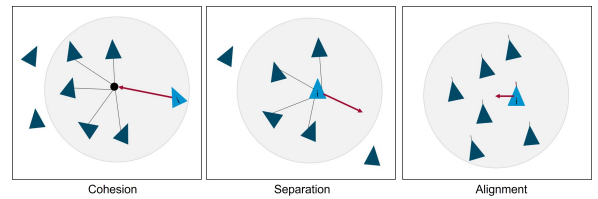


FIG 1. Reynolds rules, own illustration in accordance with [9]

With all known vectors derived from each rule the boid moving vector is computed as a linear combination, shown in Equation 6 [3, 8].

$$(6) \quad \mathbf{v}_i = w_1 \mathbf{C}_i + w_2 \mathbf{A}_i + w_3 \mathbf{S}_i$$

$w_1$ ,  $w_2$  and  $w_3$  are coefficients in an  $[0, 1]$  interval to determine the importance of each force acting as proportional gains [8].

## 2.2. Obstacle Avoidance with Trajectory Following

In order to use a sUAS swarm in a real-world scenario, an obstacle avoidance algorithm must be established to ensure the safety of an operation [3]. Here, the Force-field based approach collision avoidance algorithm as described in [10] is used.

This method uses attractive and repulsive forces to model interactions between sUAS and obstacles [10]. The Figure 2 shows the principle of this obstacle avoidance method.

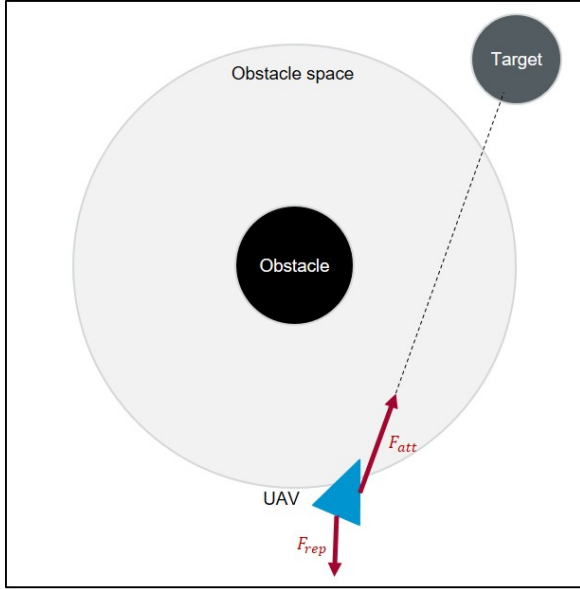


FIG 2. Obstacle avoidance with a flight plan, own illustration in accordance with [11,12]

The obstacle is surrounded by an obstacle space, which is the source of the repulsive force and sets a predefined collision range [3,12]. If the individual sUAS reaches the threshold of the obstacle space, the sUAS need to change their direction [3]. The attractive force is the force around a target, which the sUAS needs to reach [12]. Using the known forces, the direction of the resultant force can be calculated, taking into account both the attractive and repulsive components [12]. Note that the force-field based method is limited in their application. Limitations occur in specific obstacle situations, where the sUAS converged to a local minimum [13].

The attraction force is generated by the target, which in this scenario is the next waypoint. [14] uses Equation 7 to calculate the force vector to guide the sUAS swarm to the next waypoint.

$$(7) \quad \mathbf{T}_i(t) = \mathbf{p}_{des} - \mathbf{p}_i(t)$$

The vector  $\mathbf{T}_i(t)$  points from the current position of sUAS  $\mathbf{p}_i(t)$  to the position of the waypoint  $\mathbf{p}_{des}$  [14]. The force causes the sUAS to move in a specific direction towards the waypoint with a defined magnitude [14].

## 2.3. Choice of Coefficients

As described above, to achieve desired swarm behavior, it is crucial to control the magnitude of each rule output relative to others. In order to determine the weighting and thus the exact values for the coefficients, numerous tests were carried out in the simulation and during test flights. Each of the analysed scenario result in different optimum coefficient sets. For the given tasks, a feasible set of coeffi-

cients was found through experiments ensuring stability of the swarm, even under different initial conditions.

## 3. SIMULATION

The proposed rules were tested in different scenarios in simulation. This was done to validate the mechanism of the system to behave as a swarm. The alignment output is included in the implementation but was not used in further testing in this work as first results have shown almost no influence of it when using a high trajectory output.

The simulation makes use of a Unity environment mimicking the real test-range facility for sUAS indoor-operation at Technische Universität Berlin and features a realistic dynamics model of the given sUAS, including Gauß-Markov-based flight-technical-error modeling. The setup is used as Software-In-The-Loop evaluation method which builds upon the Robot Operating System 2 (ROS2), where a swarm of 10 unmanned aerial vehicles (sUAS) was modeled. Each UAV followed a simplified kinematic model for movement and interaction. The environment was created to simulate realistic conditions, including different obstacle fields. The control and communication between the sUAS were handled through one ROS2 node per sUAS communicating via the ROS2 publisher and subscriber concepts, ensuring accurate output calculations for cohesion, separation, and obstacle avoidance. Identical to the real test-range sUAS, the sensor out- and the control inputs of the simulation run at 10 Hz, allowing for detailed analysis of swarm dynamics and behavior throughout the simulation.

### 3.1. Scenario 1: Forest

In order to simulate a realistic environment for a swarm mission, a flight through a forest was chosen. Considering the average spacing of trees in a very dense forest, the distance between each cylinder, representing a tree, is fixed to 1.5 m. This environment enables to test in specially the obstacle avoidance implementation. The swarm should be splitting up to avoid the trees but find together after the obstacles to follow the common flight plan. In Figure 3 the flight track of each swarm member is shown for a sample execution of the experiment. The waypoints are plotted in orange, showing that the swarm is following the flight plan and only deviating slightly due to obstacle avoidance and the swarm dynamic. The following of a given flight path is illustrated by Figure 4 when each of the reached waypoints results in a decrease of the output until the next waypoint is targeted. During all executions of the experiment, no collisions occurred.

To ensure safe operation of the swarm, a safe distance between the swarm member needs to be maintained by applying the separation output. For this simulation experiment, no conflict was detected with a distance below 0.4 m. Only two near approaches between sUAS 1/5 and sUAS 1/8 occurred, with a distance of 0.44 m between the sUAS.

The scenario enables to test in specially the obstacle avoidance implementation. The swarm should be splitting up to avoid the trees but find together after the obstacles to follow the common flight plan. During the simulation, no problems with obstacle conflicts occurred, as a minimal distance of 0.26 m between an obstacle and all sUAS is guaranteed.

The plot of the obstacle avoidance output in Figure 4 shows the influence of each obstacle on the output visible when the sUAS enters the threshold area around an obstacle. A better on map magnitude of the repulsive output around

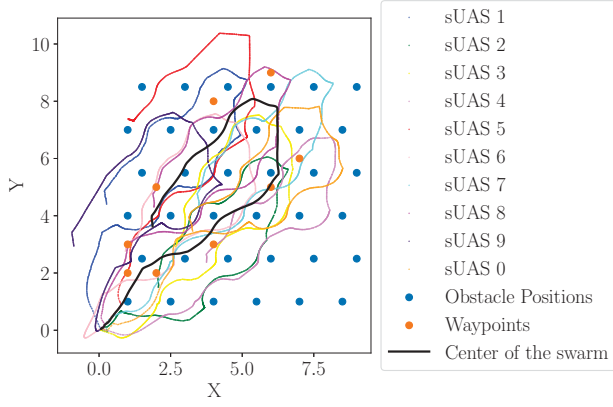


FIG 3. Forest Scenario - Flight path of sUAS

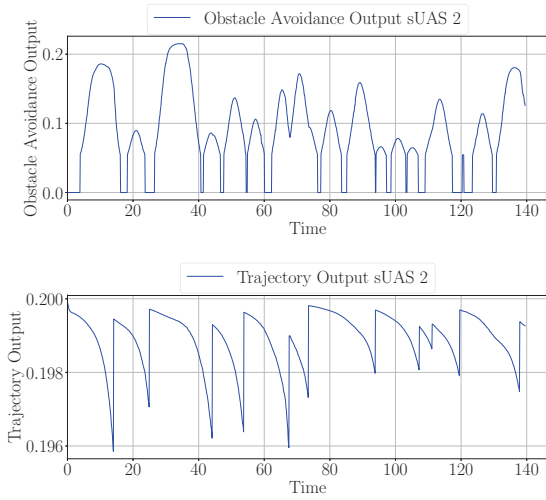


FIG 4. Forest Scenario - Trajectory and Obstacle Avoidance Output acting on sUAS 2

each obstacle is given in Figure 5. For a better visualisation, only the recorded data sets from four sUAS are used in Figure 5. The increase in the obstacle avoidance output is visible for each sUAS getting near one of the represented cylinders.

### 3.2. Scenario 2: V-Formation

In this scenario, obstacles were arranged in a V-formation to assess the boundaries of the sUAS and the swarm as an overall system. In particular, the behavior of the swarm members to act as a swarm in a highly restrictive environment is to be examined. The V-formation of obstacles created for this purpose has a distance of 0.3m in both the x- and y-directions.

In the simulation, it is observed that the sUAS disperse in front of the V-shaped obstacle formation due to the influence of the obstacle avoidance output. None of the sUAS successfully navigate through the obstacle gaps, as the pre-set threshold of 0.5m does not permit sufficient clearance between the obstacles. Instead, to reach the designated waypoints, the sUAS follow the contour of the V-formation, eventually regrouping once the end of the obstacle formation is reached.

By selecting the center of the swarm as the reference for trajectory following, the waypoints are reached success-

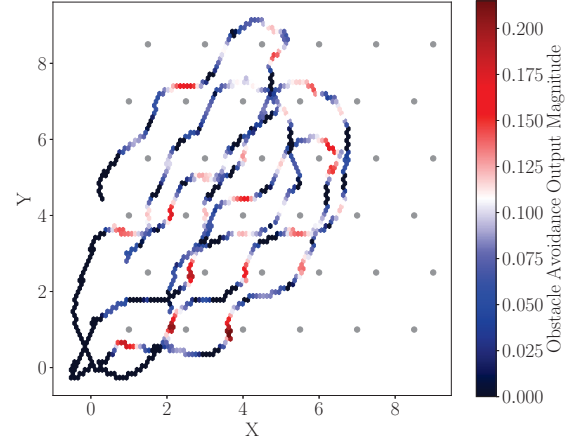


FIG 5. Forest Scenario - Heatmap of the Obstacle Avoidance Output Magnitude

fully. This approach ensures that the sUAS navigate collectively around the obstacles, rather than individually targeting each waypoint. The flight path of the swarm, along with a visualization of its center, is illustrated in Figure 6.

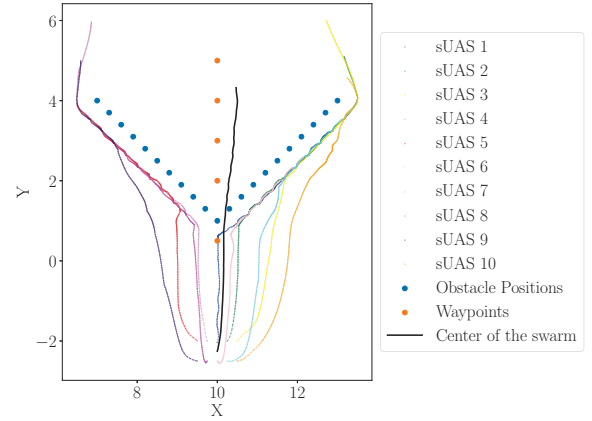


FIG 6. V-Scenario - Flight path of sUAS

When evaluating the respective outputs, several patterns emerge during the execution of the flight plan. Notably, as the swarm encounters the V-shaped obstacle formation, the sUAS become increasingly separated, leading to a gradual rise in the cohesion output over time. This trend, visible in the graph, indicates the system's effort to maintain swarm integrity as the sUAS attempt to navigate around the obstacles. The increase in cohesion output ensures that the swarm remains compact by pulling the sUAS back together once they have passed the obstacle formation. This behavior is critical for maintaining swarm coordination and preventing excessive dispersion, particularly in challenging environments. Figure 7 below illustrates the cohesion output norms for each individual sUAS, with distinct increases corresponding to the moments of swarm separation and regrouping.

In Figure 8, it can be observed that the distance to the closest waypoint decreases over time as the sUAS move towards it. Once a waypoint is reached, the next waypoint is selected as the new target. This process leads to a repeated drop in distance to the waypoint, followed by a transition

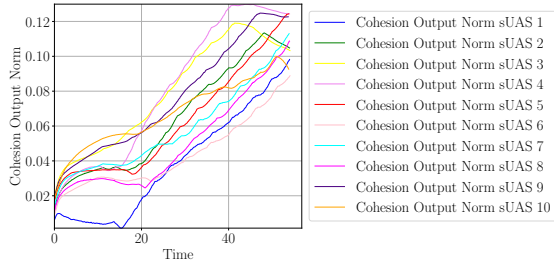


FIG 7. V-Scenario - Cohesion output over time

to the next target. As each new waypoint is selected, the trajectory output increases, guiding the sUAS toward the newly assigned target. The increase in output ensures that the swarm remains on track and adjusts its flight path accordingly.

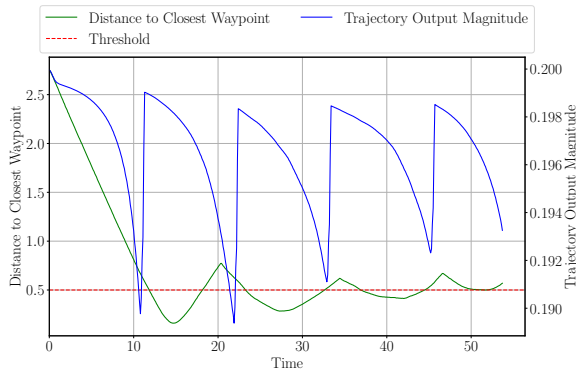


FIG 8. V-Scenario - Trajectory output over time

In Figure 9, the obstacle avoidance output becomes crucial as the swarm approaches obstacles. The obstacle threshold represents the minimum safe distance between the sUAS and the obstacles. When the sUAS get too close, the avoidance output escalates, pushing them away to avoid collisions. If the distance exceeds the threshold, the system reduces this output, allowing the sUAS to maintain their planned trajectory while keeping a safe margin from the obstacles.

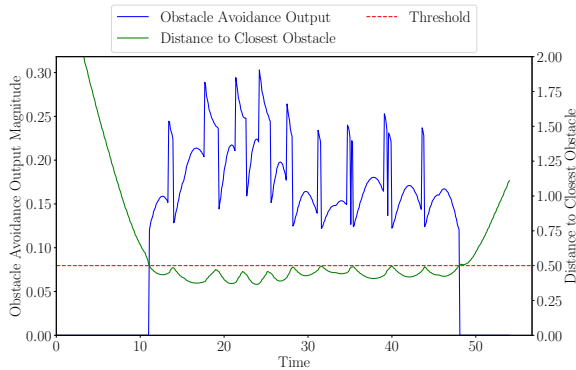


FIG 9. V-Scenario - Obstacle avoidance output over time

An analysis of obstacle avoidance in relation to the map highlights the repulsive outputs experienced by the sUAS along the obstacle formation. This phenomenon is visually represented in Figure 10, where the distribution of

repulsive outputs along the obstacle formation is clearly illustrated. Moreover, it can be observed that the avoidance output intensifies at the initial phase of the formation. This increase is attributed to the concentration of sUAS in that area simultaneously, which amplifies the influence of outputs like separation and trajectory adjustment outputs. This localized congestion leads to a higher cumulative effect of these outputs on the sUAS motion.

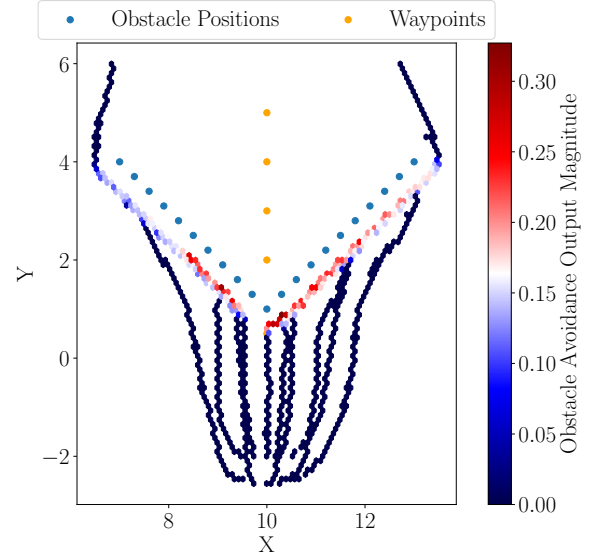


FIG 10. V-Scenario - Heatmap

#### 4. FLIGHT TEST

As an application to a real-world scenario, multiple flight tests were performed to evaluate the Reynolds flocking behavior in a practical setting. The flight test was conducted in a controlled indoor environment using a swarm of 3 sUAS. The indoor environment features Ultra-Wide-Band transceivers as anchors and beacons for centimeter-level indoor-navigation accuracy using DJI Tello EDU sUAS. The test area was defined by four waypoints, and no external obstacles were included in this scenario. The sUAS followed the same control logic as the simulation, with real-time adjustments to trajectory and separation outputs. Data was logged at 10 Hz to analyze the system's performance in maintaining stable swarm behavior and ensuring safe distances between sUAS during the flight.

The flight test implemented a modified trajectory control logic compared to the approach described in section 2. In this new logic, a waypoint is considered reached when each individual sUAS comes within a distance of 1 m from the waypoint, as opposed to basing the condition on the center of the swarm. The flight paths of the sUAS are illustrated in Figure 11 below, which demonstrates that the Reynolds flocking system successfully guided the sUAS along the intended flight plan, adhering to the proposed waypoints. The flight test allowed for a detailed analysis of several key aspects regarding the system's ability to function as a Reynolds flocking model. Both the trajectory output and the separation output exhibited significant characteristics during the test. Figure 12 illustrates the trajectory output as a function of the distance to the next waypoint.

Initially, the trajectory output appears to remain constant, prompting a more detailed analysis. Figure 13 and Fig-

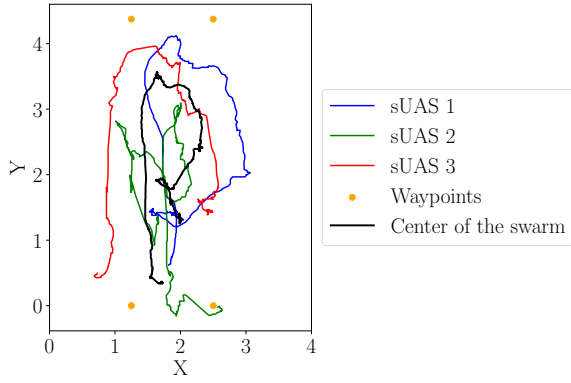


FIG 11. Flight Test - Flight path of sUAS including center of the swarm

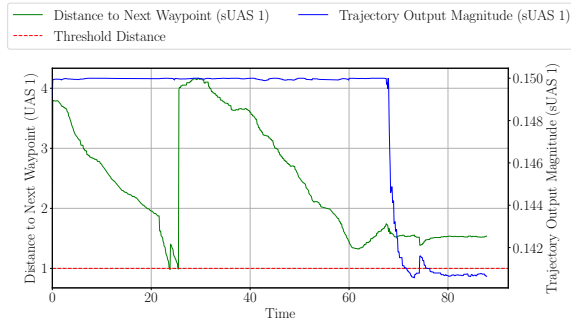


FIG 12. Flight Test - Trajectory output over time

ure 14 illustrate the components of the trajectory output for sUAS 1, plotted against the distance in the x- and y-directions to the next waypoint, respectively. According to the specified flight plan, movement towards the next waypoint results in changes solely in either the x- or y-direction. The observed discontinuities arise due to a change in the target waypoint. The relationship between the trajectory output and the distance is evident, while minor deviations can be attributed to short temporal offsets in the calculation of the relative position. These offsets lead to brief mismatches between the actual and target waypoints.

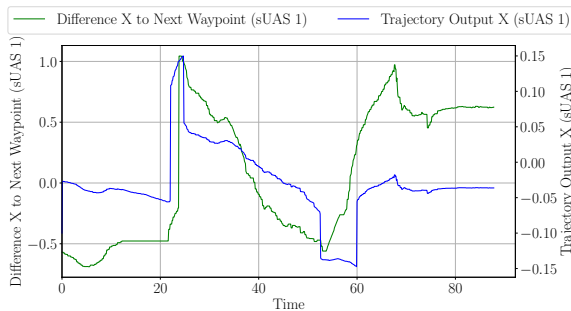


FIG 13. Flight Test - Trajectory output X component over time

The following Figure 15 depicting the average separation output in dependence of the distance to each sUAS, shows a stable pattern, indicating that the system effectively maintains safe distances between sUAS throughout the test. This stability suggests that the separation output is functioning as intended, preventing sUAS from clustering too closely and ensuring collision avoidance.

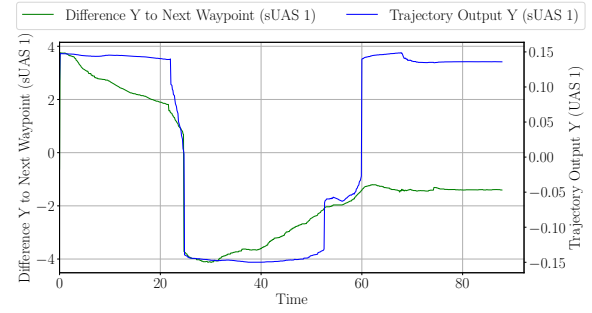


FIG 14. Flight Test - Trajectory output Y component over time

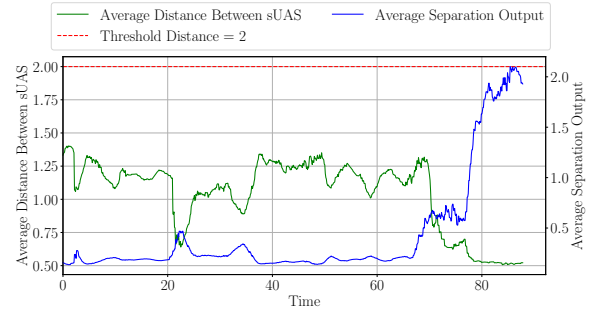


FIG 15. Flight Test - Separation output over time

In addition, the plot of the average distance between sUAS highlights the system's capability to keep the sUAS at a consistent spacing, closely following the target threshold of 2 meters. The sUAS maintained a steady formation, balancing the cohesion and separation outputs. It can be observed that with decreasing distance between the sUAS, the separation output increases to prevent any collisions. This balance is crucial for achieving coordinated flight without erratic movements or significant deviations in distance.

Overall, the test confirms that the system achieves both stability and adaptability, successfully maintaining swarm cohesion while managing inter- sUAS spacing using the experimentally chosen coefficients.

## 5. CONCLUSION

The results from the implementation and subsequent flight tests offer valuable insights and identify key areas for further investigation. In the simulation of a forest-like scenario, the swarm exhibited effective obstacle avoidance, while maintaining cohesion and adhering to a pre-determined flight path. The V-shaped scenario, in particular, highlighted the algorithm's capability to split and regroup, demonstrating its potential applicability in more complex and dynamic environments. Transitioning to indoor flight tests involving three sUAS (small Unmanned Aerial Systems) provided critical data on the algorithm's practical implementation, where the swarm successfully navigated toward predefined waypoints, all while maintaining safe inter-vehicle distances.

This study highlights the successful advancements made to the original three Reynolds rules, including the implementation of the separation and obstacle avoidance rule. One of the significant outcomes is the demonstration of effective waypoint navigation in both simulated and indoor flight scenarios. The analysis shows that the swarm maintained

stability concerning safe inter-vehicle distances and transitions between waypoints, indicating that the system strikes an effective balance between governing control outputs. However, swarm stability remains crucial for practical applications. While the system generally maintained stability, instances of oscillatory behavior were observed, particularly in constrained spaces or during sudden environmental changes. Given the dynamic nature of sUAS swarm operations, further investigation into the algorithm's stability under these conditions is required. The complexity of implementing swarm algorithms presents challenges for the exact replication of indoor test results, as behavior tends to differ between simulation and flight tests. This underscores the necessity of developing reproducible testing methodologies, including improved obstacle avoidance strategies. The behavior of the swarm is highly sensitive to the parameterization of the coefficients, making the identification of an optimal configuration difficult for different mission profiles. Dynamic thresholds based on environmental conditions or specific swarm configurations could be explored in future work. In this context, refining the conditions for waypoint achievement could further enhance the adaptability of the algorithm.

In conclusion, this paper demonstrates the potential of Reynolds rule-based algorithms for controlling sUAS swarms in both simulated and indoor environments. The results provide a foundation for further refinement of the key topics discussed.

## References

- [1] Maarten Uijt de Haag and Mats Martens. Cooperative sUAV swarm navigation for urban infrastructure monitoring. In ION GNSS+, The International Technical Meeting of the Satellite Division of The Institute of Navigation, pages 2835–2848. Institute of Navigation, Oct. 2022.
- [2] Craig W. Reynolds. Flocks, herds, and schools: A distributed behavioral model. SIGGRAPH Computer Graphics, 21(4):25–34, July 1987. ISSN: 0097-8930.
- [3] Saleh Alaliyat, Harald Yndestad, and Filippo Sanfilippo. Optimisation of boids swarm model based on genetic algorithm and particle swarm optimisation algorithm (comparative study). European Conference on Modelling and Simulation, 2014.
- [4] Craig W. Reynolds. Flocks, herds, and schools: A distributed behavioral model. SIGGRAPH '87: Computer graphics and interactive techniques, 14:25–34, 1987.
- [5] Bedrich Hartman, Christopher ans Benes. Autonomous boids. Computer animation & virtual worlds, 17:199–206, 2006.
- [6] R.V. Meshcheryakov, P.M. Trefilov, A.V. Chekhov, S.A.K Diane, and K.D. Rusakov. An application of swarm of quadcopters for searching operations. IFAC Conference on Technology, Culture and International Stability TECIS, 52:14–18, 2019.
- [7] Logan E. Beaver and Andreas A. Malikopoulos. An overview on optimal flocking. Annual Reviews in Control, 51:88–99, 2021.
- [8] Nour El Houda Bahloul, Saadi Boudjit, Marwen Abdennebi, and Djallel Eddine Boubiche. A flocking-based on demand routing protocol for unmanned aerial vehicles. Journal of Computer Science and Technology, 33:263–276, 2018.
- [9] Craig W Reynolds. Flocks, herds and schools: A distributed behavioral model. In Proceedings of the 14th annual conference on Computer graphics and interactive techniques, pages 25–34, 1987.
- [10] Jawad Naveed Yasin, Mohammad-Hashem Haghbayan, Jukka Heikkonen, Hannu Tenhunen, and Juha Plosila. Formation maintenance and collision avoidance in a swarm of drones. International Symposium on Computer Science and Intelligent Control, 3:1–6, 2019.
- [11] Weiwei Zhao, Hairong Chu, Mingyue Zhang, Tingting Sun, and Lihong Guo. Flocking control of fixed-wing uavs with cooperative obstacle avoidance capability. IEEE Access, 7:17798–17808, 2019.
- [12] Hong Shen and Ping Li. Unmanned aerial vehicle (uav) path planning based on improved pre-planning artificial potential field method. Chinese Control And Decision Conference (CCDC), 2020.
- [13] Logan E. Beaver and Andreas A. Malikopoulos. Beyond reynolds: A constraint-driven approach to cluster flocking, 2020.
- [14] Yang Huang, Jun Tang, and Songyang Lao. Collision avoidance method for self-organizing unmanned aerial vehicle flights. IEEE Access, 7:85536–85547, 2019.