UAV collision avoidance using multiple artificial potential functions: Practical implementation and experimental outdoor applications

Oscar F. Archila[1,2], Alain Vande Wouwer[2] and Johannes Schiffer[3]

Abstract— **In unmanned aerial vehicle (UAV) applications, fast-response collision avoidance is an essential requirement for a safe flight. This applies in particular to real-world settings with constrained hardware performance. With these challenges in mind, the authors have recently presented a collision avoidance strategy based on multiple artificial potential functions (MAPOFs). The strategy overcomes the typical drawbacks of the conventional, popular APOFs approach, such as chattering or deadlock. The main focus of the present paper is on the practical implementation and experimental demonstration of the proposed MAPOF strategy in two real-life outdoor scenarios using a commercial quadrotor and open-source software. The experiments show the effectiveness of the MAPOFs strategy in the presence of stationary obstacles as well as in the presence of another moving UAV. The experiments are complemented by additional simulation results to further illustrate and validate the proposed approach.**

I. INTRODUCTION

Unmanned aerial vehicles (UAVs), commonly known as drones, are increasingly being utilized in various applications such as smart farming, monitoring and surveillance, building inspections, image capture and filmmaking, and artistic performances [1], [2], [3]. With the growing use of drones for these purposes, safety concerns have become paramount, necessitating the implementation of efficient collision avoidance systems (CAS) as an additional layer on top of conventional flight controllers [4].

Among the different CAS approaches, a strategy that has been largely studied in robotics is artificial potential functions (APOFs) [5], [6]. The design of APOFs is inspired by the idea of applying artificial ”forces” to the low-level tracking control of the UAV. This helps the drone to avoid obstacles while following a specific trajectory or reaching a final target location. For instance, in [7], the authors integrate the APOFs in a UAV formation, adding a repulsion force around the obstacles. In [8], the authors integrate the APOFs in a path-planning method to avoid possible obstacles in the environment. Although APOFs have been effectively employed to navigate around fixed or moving obstacles in various studies, they have two significant drawbacks: chattering, which can lead to deadlocks, and the lack of experimental demonstrations in a hardware UAV platform [2].

In [9], the authors mitigate the chattering by implementing a rotating potential field that determines the direction in which obstacles should be avoided smoothly. Another approach that deals with this issue is barrier functions. For instance, in [10], the author uses barrier functions to create a smooth time-invariant controller for UAV collision avoidance. Although barrier functions can effectively handle this critical challenge, their implementation is often complex and computationally demanding. Other approaches that deal with the drawbacks of standard APOFs are based on model predictive control (MPC). In [11], the authors present a chance-constrained approach where the obstacles are modeled in a probabilistic manner in the MPC problem. Similarly, in [12], the authors propose chance constraints based on the obstacle velocity in the MPC framework. However, since a non-linear optimal problem needs to be run for each step iteration, the computation cost of this MPC approach is comparatively high. This presents significant limitations for small UAV platforms with constrained hardware, where real-time responsiveness is essential for effective obstacle avoidance [13].

Moreover, experimental applications of APOFs for outdoor UAV collision avoidance remain scarce in the literature. In [14], the authors develop experiments using APOFs to maintain a circular formation for fixed-plane UAVs. However, these experiments do not focus on collision avoidance.

To address the chattering issue and the stability analysis, the present authors have recently proposed a new collision avoidance strategy based on multiple APOFs (MAPOFs). The theoretical development of the MAPOF strategy, together with the associated closed-loop stability properties, are currently under review elsewhere [15]. The focus of the present article is on the technical implementation aspects of the MAPOF strategy and its experimental validation in outdoor conditions. The contributions of this paper are, therefore, as follows:

- We revisit the novel MAPOFs strategy from [15] and provide a detailed description of the algorithm's implementation on an actual experimental UAV platform.
- We describe the hardware platform, detailing the quadrotor specifications, middleware, and control adaptation in an open-source environment. This also includes

*This work is partially supported by the BMBF project iCampus2 (#16ME0420K).

[1]Oscar F. Archila is with the Control Systems and Network Control Technology Group, Brandenburg University of Technology Cottbus-Senftenberg (BTU C-S), 03046 Cottbus, Germany and with the Department of Systems, Estimation, Control and Optimization (SECO), University of Mons, 7000 Mons, Belgium archilacr@b-tu.de

[2]Alain Vande Wouwer is with the Department of Systems, Estimation, Control and Optimization (SECO), University of Mons, 7000 Mons, Belgium alain.vandewouwer@umons.ac.be

[3]J. Schiffer is with the Control Systems and Network Control Technology Group, Brandenburg University of Technology Cottbus-Senftenberg (BTU C-S), 03046 Cottbus, Germany and the Fraunhofer IEG, Fraunhofer Research Institution for Energy Infrastructures and Geothermal Systems IEG, 03046 Cottbus, Germany schiffer@b-tu.de

testing the strategy performance through simulation.
- We present experimental results of the MAPOF strategy for two real-life scenarios in outdoor conditions: one with multiple fixed obstacles and another with an additional UAV in the spatial domain.

The remainder of this paper is organized as follows. In Section II, the collision avoidance strategy is detailed. The UAV setup, including software and hardware specifications, is presented in Section III. In Section IV, the modeling and simulations used to evaluate the strategy are described. Experimental results are presented in Section V, followed by performance analysis in Section VI. Finally, conclusions are discussed in Section VII.

II. Collision Avoidance Strategy

A collision avoidance system with fast response and low computational cost is required. The APOF approach is a simple method used in robotics that satisfies these requirements. However, its main disadvantages include its unsmooth trajectory and the fact that it can generate deadlocks [16]. To overcome these limitations, we have introduced in [15] a new method that, by using *multiple* APOFs, avoids the previous problems and is based on an additional suitable space partition strategy. Then, for each partition, an appropriate APOF is defined. This method is revisited next.

A. *Multiple artificial potential functions (MAPOFs)*

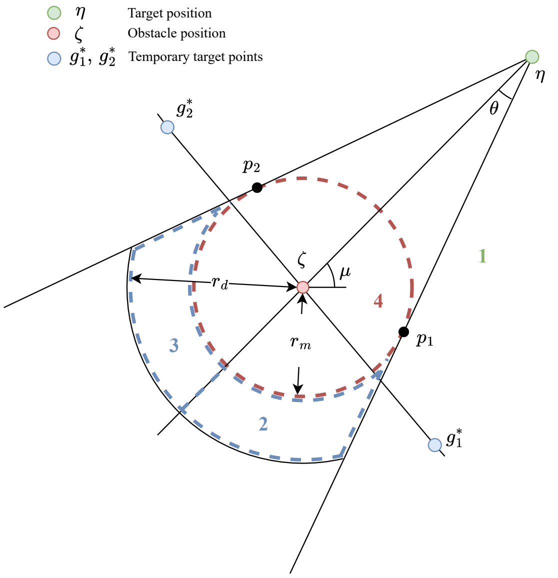

Fig. 1. Space partition for the MAPOFs strategy for collision avoidance.

MAPOFs are used to address collision avoidance. Instead of using an additive force that sums up the attraction and repulsive forces, we propose generating a space partition (see Figure 1) based on the obstacle's position and the target. This allows using a specific force in each sub-region, generating a smooth trajectory. The strategy is based on three criteria:

- An attraction force F_1 pointing to the target position η is applied when the UAV is not detecting obstacles.
- Avoidance forces F_2 or F_3 are applied to change the UAV direction, when it detects an obstacle. These forces point to temporary target points g_1^* and g_2^*.
- A repulsion force F_4 is applied, when the UAV is close to the obstacle ζ.

According to the previous logic, the applied force depends on the space partition illustrated in Figure 1. The numbers represent the regions where specific forces are applied. The latter are defined as

$$\begin{aligned} F_1(\xi) &= k_\eta(\eta - \xi), \\ F_2(\xi) &= k_g(g_1^* - \xi), \\ F_3(\xi) &= k_g(g_2^* - \xi), \\ F_4(\xi) &= -k_\zeta(\zeta - \xi), \end{aligned} \tag{1}$$

where $k_\eta, k_g, k_\zeta > 0$ and $\xi \in \mathbb{R}^2$ is the current position of the UAV. From (1), it follows that F_1 is an attraction force in the direction of the target position, F_4 is a repulsion force in the opposite direction of the obstacle, and F_2 and F_3 are attraction forces to the temporary target points that allow the creation of ”escape” trajectories and the mitigation of potential oscillations or, worse, deadlocks.

B. *Space partition*

Consider the space $\mathbb{R}^2$ in Figure 1 segmented in four regions, where every region applies a different APOF. This segmentation is used when an obstacle is spotted in the detection range $r_d > 0$ of the UAV. Let the current obstacle position be denoted by $\zeta \in \mathbb{R}^2$ and the security radius by $r_m > 0$. The objective of the UAV is to reach the target position $\eta \in \mathbb{R}^2$. The numbers in Figure 1 represent the region numbers. Additionally, we define the following quantities:

$$\begin{aligned} \mu &= \arctan\left(\frac{\eta_y - \zeta_y}{\eta_x - \zeta_x}\right), \\ \theta &= \arctan\left(\frac{r_m}{\|\eta - \zeta\|_2}\right), \end{aligned} \tag{2}$$

where $\mu \in [0, 2\pi]$ is the relative angle between the obstacle and the goal, and $\theta \in [0, 2\pi]$ is the relative angle between the lines projected onto the obstacle and the points $p_1 \in \mathbb{R}^2$ or $p_2 \in \mathbb{R}^2$. The subscripts x and y denote the first and second components of the vectors η and ζ, respectively.

We define the points p_1 and p_2 as

$$p_1 = [r_m \cos(\theta), r_m \sin(\theta)]\, \mathbf{R}(-\mu) + \zeta, \tag{3}$$

$$p_2 = [r_m \cos(\theta), -r_m \sin(\theta)]\, \mathbf{R}(-\mu) + \zeta, \tag{4}$$

where $\mathbf{R}(\cdot)$ is the 2-D rotation matrix

$$\mathbf{R}(\phi) = \begin{bmatrix} \cos\phi & -\sin\phi \\ \sin\phi & \cos\phi \end{bmatrix}.$$

The temporary points g_1^* and g_2^* are placed perpendicular to the line generated by the target position and the obstacle as follows:

$$g_1^* = \left([r_m, 0] \, \mathbf{R} \left(\mu + \frac{\pi}{2} \right) \right) d + \zeta, \tag{5}$$

$$g_2^* = \left([r_m, 0] \, \mathbf{R} \left(\mu - \frac{\pi}{2} \right) \right) d + \zeta, \tag{6}$$

where $d > 1 + \frac{r_d}{r_m}$ ensures the points outside of the detection area.

C. State dependent switching function

Since the force is applied depending on the position of the UAV, we define a switching function that depends on ξ, η, and ζ. This function activates the correct APOF according to (1) and Figure 1. First, let $z_c, z_1, z_2, z_3 \in \mathbb{R}^2$, and consider the function $\Gamma : \mathbb{R}^2 \times \mathbb{R}^2 \times \mathbb{R}^2 \times \mathbb{R}^2 \to \mathbb{R}$,

$$\Gamma(z_c, z_1, z_2, z_3) = \begin{cases} 1, & z_c \text{ is inside of the lines generated} \\ & \text{by } z_1, z_2 \text{ and the origin } z_3 \\ 0, & \text{otherwise.} \end{cases} \tag{7}$$

Now, we can use Γ to define the switching function $\sigma : [0, \infty) \to \{1, 2, 3, 4\}$ that generates the force $F_{\sigma(t)}$ to be used at each instant of time, i.e.,

$$\sigma(t) = \begin{cases} 4, & \|\xi - \zeta\|_2 < r_m, \\ 3, & \Gamma(\xi, p_1, \zeta, \eta) = 1, \\ & \Gamma(\xi - \zeta, (\eta - \zeta)\mathbf{R}(\pi), p_1 - \zeta, [0, 0]^\top) = 1, \\ & \|\eta - \zeta\|_2 > r_m, \; \|\eta - \zeta\|_2 < r_d, \\ 2, & \Gamma(\xi, p_2, \zeta, \eta) = 1, \\ & \Gamma(\xi - \zeta, (\eta - \zeta)\mathbf{R}(\pi), p_2 - \zeta, [0, 0]^\top) = 1, \\ & \|\eta - \zeta\|_2 > r_m, \; \|\eta - \zeta\|_2 < r_d, \\ 1, & \text{otherwise.} \end{cases} \tag{8}$$

The function in (8) is constructed based on the space partition of Figure 1 and represents the geometrical restriction to delimit the four regions:

- For $\sigma(t) = 4$ the restriction is given by the circle of radius r_m.
- For $\sigma(t) = 3$, the first restriction is given by the cone generated by the points p_1, ζ, and η. The second restriction ensures that the region of the cone taken is the furthest from the obstacle. In this case, the sub-cone restriction generated by p_1, ζ, and η rotates π with respect to the obstacle. Finally, the last two restrictions are given by the circles with radius r_d and r_m.
- In the case of $\sigma(t) = 2$, the analysis is the same that $\sigma(t) = 3$ but instead of the point p_1 is used p_2.
- The $\sigma(t) = 1$ corresponds when an obstacle is not detected, or the obstacle is not interfering with reaching the target.

D. Closed-loop stability

The multiple APOF CAS algorithm is amenable to stability analysis, which is the subject of another article [15]. In this analysis, the closed-loop system is considered like a switched system, where $\sigma(t)$ is the switching signal. The concept of dwell time [17] is then used to provide sufficient tuning conditions, which ensure the stability of the closed-loop system. Since the focus of the present paper is on the practical implementation of the MAPOF strategy (1), (8), the stability conditions are not fully revisited here.

III. PRACTICAL IMPLEMENTATION OF THE UAV SETUP

The experiments are developed considering a commercial quadrotor for outdoor flights. The hardware includes the drone frame with the motors, a flight controller unit (FCU) in charge of the low-level control, an onboard computer (OBC) where the CAS strategy is implemented, the ground control station (GCS) to manage the mission, and a router that generates a WiFi network. For the software, Ardupilot is used as the firmware of the FCU, and ROS 2 is the framework for running the high-level control and managing the communication with the UAV and the GCS.

A. Quadrotor

The quadrotor used for the experiments is the commercial kit QAV25 from Holybro. The kit includes a Pixhawk 6c as an FCU. As OBC, a Raspberry Pi zero 2w is attached to the UAV. The OBC is in charge of running the calculations of the CAS strategy and managing the communication and the protocol for starting the mission. The OBC is connected to the FCU through a serial module. Figure 2 shows the quadrotor with the hardware, and Table I shows a summary of the UAV information.

B. Middleware

The middleware used for managing the high-level controller is ROS 2. The robot operating framework (ROS) framework is an open-source tool used for robotic applications that helps to develop tasks in real-time. In this case, we use ROS 2 Humble version [18] for Ubuntu 22.04 and Python as the programming language. ROS is based mainly on the concept of nodes that represent the recurrent task, and the topic is the means of communication between the nodes. One of the advantages is that different nodes can run in parallel and share information using the topics even through the WLAN network. Using this advantage, we employ a router in the field that generates the WLAN network and runs the different nodes in the devices, e.g., UAVs and GCS, sharing information through the topics.

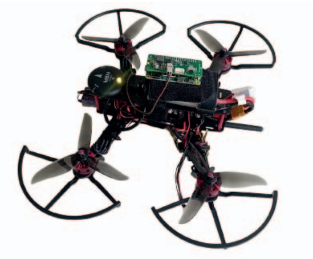

Fig. 2. Quadrotor QAV25 from Holybro with a Raspberry Pi zero 2w and a serial communication module.

TABLE I
UAV FEATURES.

Items	QAV250 UAV
OBC hardware	Raspberry Pi Zero w2 with Pi Connect Lite module. Single-core Processor up to 1 GHz, 512 MB RAM
OBC firmware	Dronekit and ROS 2
Battery	Lipo 1300 mAh
Size	250 mm
Flight time	7 min
Total weight	619 g

C. UAV position and obstacle definition

The position of the UAV is estimated by the FCU using a combination of the GPS and IMU. The UAV's altitude is fixed at a specific altitude, which allows the use of the MAPOFs in the 2D frame. The obstacle is assumed to be known only when the UAV is close to it and defined as a GPS coordinate, i.e., the obstacle from the UAV's point of view includes the GPS error. Internally, the UAV position and the obstacle position are transformed into meters and passed to the MAPOFs strategy (1), (8).

D. High-level controller

The OBC is in charge of managing the high-level controller. For this purpose, the serial connection between the FCU and OBC uses a Mavlink protocol (supported by Ardupilot) and the Dronekit package in Python. This UAV system allows the Ardupilot to change the FCU's flight mode to "guided," which allows automatic tasks. In the present case, the "to go" task command is used to apply the artificial force $F_{\sigma(t)}$ generated by the CAS strategy (1), (8). The command receives a GPS position as a parameter that indicates a position to be reached. Since the CAS returns a vector referring to the current position of the UAV, this is transformed into global coordinates and used as input to the command. That allows the UAV to go in the direction of the target position while avoiding obstacles.

IV. MODELING AND SIMULATION

At first, we provide simulation results to assess the performance of the proposed CAS using the MAPOFs strategy (1), (8). This is followed by experimental test runs in Section V.

Since the FCU has a low-level controller [19], we use a homological second-order system as UAV dynamics for simulation purposes [20], i.e.,

$$\ddot{\xi} = \frac{u_{\sigma(t)}}{m}, \tag{9}$$

where m is the mass of the UAV and $u_{\sigma(t)}$ is defined as

$$u_{\sigma(t)} = F_{\sigma(t)}(\xi) - k_d\dot{\xi}. \tag{10}$$

The term $F_{\sigma(t)} \in \mathbb{R}^2$ denotes the artificial force from (1) applied in each moment and governed by the switching signal in (8). The constant $k_d \in \mathbb{R}_{>0}$ is a positive gain used to add dissipation in the velocity to the control.

To test the MAPOFs-based CAS strategy, we use (9) and (10) considering three stationary obstacles in the position $\zeta_1 = [6.0, 0.0]^\top$, $\zeta_2 = [12.5, 3.5]^\top$ and $\zeta_3 = [18.0, -1.0]^\top$. The goal is placed in the position $\eta = [24.0, 2.0]^\top$. The detection radius is $r_d = 4$ m, and the security radius is set to $r_m = 2$ m. The parameters of the APOFs are selected as $k_\eta = 1$, $k_g = 2$, $k_\zeta = 2$ with $m = 0.619$ kg and $k_d = 3$. In this case, the temporary targets g_1^* and g_2^* are placed at a distance of 3 m from the obstacle. We test the proposed MAPOFs strategy using three initial conditions in $\xi(t_0) = [-2.0, -3 - i\frac{8}{1.7}]$ for $i \in \{0, 1, 2\}$. The results of the simulations for the three initial conditions are shown in Figure 3.

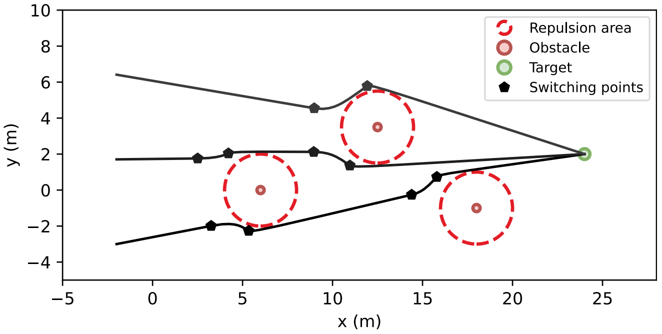

Fig. 3. Results for simulation: the CAS strategy is simulated for three obstacles and three initial conditions.

V. EXPERIMENTAL RESULTS

Two scenarios are used to test the CAS with the MAPOFs strategy (1), (8). The first one uses two fixed obstacles, and the second uses another UAV as an obstacle. The experiments are developed on a soccer field, and a fixed GPS point is selected as the global origin of the frame and an altitude of 2 m. The Haversine formula [21] is used to show the UAV and obstacle distances in meters, considering the origin as a GPS reference position and the radius of the earth as 6378.137 km. In this case, the distances are not superior to 100 m, i.e., the previous approximation gives a good conversion from GPS position to meters.

Additionally, to avoid buffer saturation of the MAVlink protocol between the OBC and the FCU, we select an input control frequency of 20 Hz. At higher frequencies, the control works with delay.

A. Test 1

The first test consists of two obstacles defined by two fixed points in GPS position. When the UAV approaches the obstacle at a distance less than the detection range $r_d = 4$ m, the obstacle position is visible. The obstacles are placed in $\zeta_1 = [5.0, 0.0]^\top$ and $\zeta_2 = [11.3, -1.5]^\top$. Figure 4 shows the experimental setup with the UAV, obstacles, and the GCS. The UAV takes off and flies to the initial position $\xi(t_0) = [18.75, -2.25]^\top$. The objective of the UAV is to reach the position $\eta = [-1.42, 1.18]^\top$. The parameters of the artificial forces are selected as $k_\eta = 1$, $k_g = 2$ and $k_\zeta = 2$. We consider a security radius of $r_d = 2$ m. The temporary points g_1^* and g_2^* are placed at 6 m from the obstacle. When the UAV is placed in the initial position, the GCS sends a command to start the mission. This communication is under ROS 2 and the WiFi network. The results of the experiments are shown in Figure 5. The position of the UAV is obtained from the flight controller, which estimates the position using GPS and IMU. To better assess the CAS performance observed in the experiment, we simulate the same scenario based on the high-level model and programming in Python. The obtained results from the simulation are also displayed in Figure 5 and are used as an ”ideal” reference for evaluating the CAS performance in Section VI.

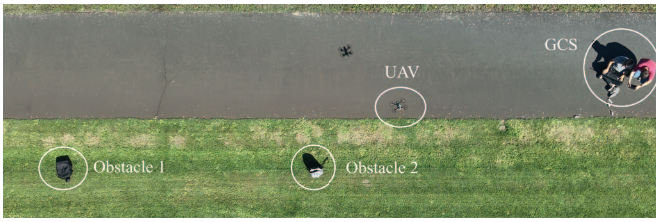

Fig. 4. Setup for experiment 1: the obstacles are placed in the defined GPS positions, and the GCS starts the automatic flight.

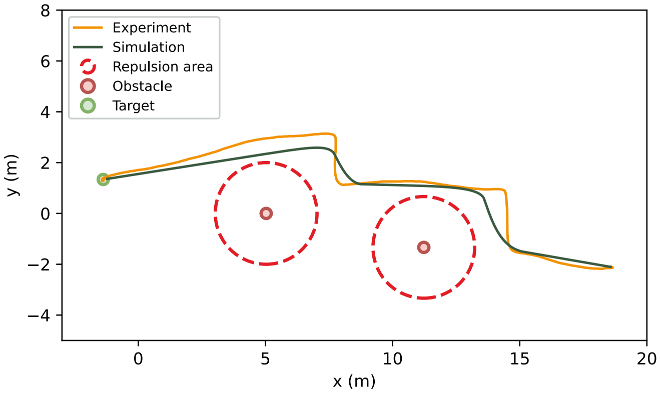

Fig. 5. Results for experiment 1: Comparison of the CAS for the experiment and the simulation for two fixed obstacles.

B. Test 2

The second test corresponds to two UAVs flying in parallel but going in different directions, emulating a scenario when two UAVs fly close to each other. In this case, only one of the UAVs is using the CAS strategy. The detection and security area are defined as $r_d = 6$ m and $r_m = 3$ m respectively. In this case, the detection radius is increased with respect to the previous experiment to consider a possible undesired performance in the low-level control of the second UAV. The initial positions of the UAVs are $\xi_1(t_0) = [0.0, -15]^\top$ and $\xi_2(t_0) = [3.2, -3.2]^\top$. The target positions are $\eta_1 = [3.75, -1.0]^\top$ and $\eta_2 = [-16.2, 2.4]^\top$. For the artificial forces, the same parameters as in test 1 are selected. In this case, the temporary points are placed at 8 m from the obstacle.

To control the two UAVs, the GCS runs a ROS 2 node that publishes a topic on the current action. Since the two UAVs are connected to the same network, they can access this topic and perform the actions according to the instructions. After the take-off, the UAVs reach the initial position, and the command to start the mission is sent. UAV 2 publishes its GPS position on a topic, and UAV 1 reads the topic. If the distance between the two UAVs is less than the detection radius, UAV 1 considers the other UAV position as an obstacle and applies the MAPOFs-based CAS strategy (1), (8). The experimental results are shown in Figure 6.

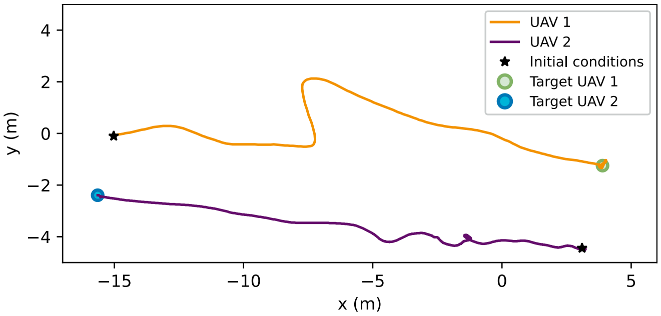

Fig. 6. Results for experiment 2: Two UAVs flying in parallel.

VI. PERFORMANCE ASSESSMENT

A. Simulations

As shown in Figure 3, the MAPOFs-based CAS strategy (1), (8) successfully guides the UAV in the correct direction to avoid obstacles. The black points represent the zone when a change of APOF occurs, and the red dotted lines are the repulsion areas around the obstacle. It is important to note that the UAV only detects obstacles within its detection range r_d. Thus, space partitioning is calculated only when an obstacle enters the detection area. Additionally, the CAS strategy and space partitioning are designed to consider a single obstacle at a time. Therefore, if multiple obstacles are detected simultaneously, only the closest obstacle is considered.

The distance covered along the shortest path from the initial condition at $\xi(t_0) = [-2.0, 3 - \frac{8}{1.7}]$ to the target position while avoiding obstacles, is 26.55 m, compared to 27.10 m when using the CAS strategy. This demonstrates that the trajectory generated by the MAPOFs (1), (8) closely approximates the optimal path, while also offering the advantages of fast response, low computational cost, and with no need for any a-priori obstacle information.

B. Experiments

To analyze the results of the experiments, it is necessary to consider the error in the GPS measurement [22]. In this case, the GPS accuracy is 2.0 m CEP. Since experiments are conducted at a short distance, the results are affected. Hence, we considered that the position of the UAVs is not directly a GPS position but an estimation of the FCU firmware using the IMU measurements that lead to better precision in the position estimation [23].

In Figure 5, the trajectory taken by the UAV shows a similar behavior to the simulation. However, a slight deviation with respect to the simulation in the change from the attraction force to the avoidance force is noticed. This is due mainly to two factors. The first is the position estimation of the UAV. Since the position of the obstacle is relative to the position of the UAV, this influences the temporary targets during the flight, changing the avoidance forces and, consequently, generating a deviation in the expected trajectory. The second factor is the rather simple dynamics (9) used in the simulations. Since the experiments are developed in an outdoor environment, perturbations, such as wind gusts, can affect the low-level control of the UAV and disturb a state that is not considered in the simple model.

In test 2 (Figure 6), it is noticed that the trajectory of UAV 1 goes in the opposite direction to the target position. This behavior is a consequence of the movement of the other UAV 2. The avoidance force directs the UAV 1 towards the temporary target. However, UAV 2 moves in the opposite direction, and the temporary target moves with the UAV 2. This behavior is observed until UAV 1 is in the attraction area to reach the target position.

Throughout all experiments, calculating the MAPOFs (1), (8) in the Raspberry Pi takes under 10 milliseconds. Still, due to a flight controller restriction, the control commands are sent at a rate of 50 milliseconds. This communication rate might have to be further reduced in cases where higher speeds are needed.

VII. CONCLUSIONS

Several simulations and outdoor experimental tests of a novel collision avoidance strategy based on MAPOFs, introduced by the authors in [15], have been presented. The MAPOFs strategy avoids the chattering common in the standard APOF approach. The results in simulations show that the UAV avoids the obstacles at the same time that it reaches the target position for different initial conditions. The MAPOFs strategy achieves a route very close to the shortest one to reach the target. Moreover, a platform for the experimental implementation of the strategy is presented, and the results of two outdoor experiments are discussed, the first with fixed obstacles and the second with a second moving UAV as an obstacle. The experiments show that in both settings, the UAV successfully avoids the obstacles and does not suffer from chattering.

Given that position estimations and environmental perturbations affect the experiments, analyzing the robustness of the strategy is an essential direction for future research. Additionally, incorporating multiple UAVs as mobile obstacles within the CAS design presents a promising avenue for improvement.

REFERENCES

[1] S. A. H. Mohsan, N. Q. H. Othman, Y. Li, M. H. Alsharif, and M. A. Khan, "Unmanned aerial vehicles (UAVs): Practical aspects, applications, open challenges, security issues, and future trends," *Intelligent Service Robotics*, vol. 16, no. 1, pp. 109–137, 2023.

[2] M. Banafaa, Ö. Pepeoğlu, I. Shayea, A. Alhammadi, Z. Shamsan, M. A. Razaz, M. Alsagabi, and S. Al-Sowayan, "A comprehensive survey on 5g-and-beyond networks with UAVs: Applications, emerging technologies, regulatory aspects, research trends and challenges," *IEEE Access*, 2024.

[3] M. Alhafnawi, H. A. B. Salameh, A. Masadeh, H. Al-Obiedollah, M. Ayyash, R. El-Khazali, and H. Elgala, "A survey of indoor and outdoor UAV-based target tracking systems: Current status, challenges, technologies, and future directions," *IEEE Access*, vol. 11, pp. 68324–68339, 2023.

[4] M. R. Rezaee, N. A. W. A. Hamid, M. Hussin, and Z. A. Zukarnain, "Comprehensive review of drones collision avoidance schemes: Challenges and open issues," *IEEE Transactions on Intelligent Transportation Systems*, 2024.

[5] S. S. Ge and Y. J. Cui, "New potential functions for mobile robot path planning," *IEEE Transactions on robotics and automation*, vol. 16, no. 5, pp. 615–620, 2000.

[6] T. Urakubo, K. Okuma, and Y. Tada, "Feedback control of a two wheeled mobile robot with obstacle avoidance using potential functions," in *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)(IEEE Cat. No. 04CH37566)*, vol. 3, pp. 2428–2433, IEEE, 2004.

[7] L. Garcia-Delgado, A. Dzul, V. Santibáñez, and M. Llama, "Quadrotors formation based on potential functions with obstacle avoidance," *IET Control Theory & Applications*, vol. 6, no. 12, pp. 1787–1802, 2012.

[8] X. Chen and J. Zhang, "The three-dimension path planning of UAV based on improved artificial potential field in dynamic environment," in *2013 5th International Conference on Intelligent Human-Machine Systems and Cybernetics*, vol. 2, pp. 144–147, IEEE, 2013.

[9] Z. Pan, C. Zhang, Y. Xia, H. Xiong, and X. Shao, "An improved artificial potential field method for path planning and formation control of the multi-UAV systems," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 69, no. 3, pp. 1129–1133, 2021.

[10] E. Restrepo, I. Sarras, A. Loria, and J. Marzat, "3D UAV navigation with moving-obstacle avoidance using barrier lyapunov functions," *IFAC-PapersOnLine*, vol. 52, no. 12, pp. 49–54, 2019.

[11] H. Zhu and J. Alonso-Mora, "Chance-constrained collision avoidance for mavs in dynamic environments," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 776–783, 2019.

[12] S. Han, J. Kwon, and K.-S. Kim, "Position uncertainty-integrated potential function for collision avoidance systems based on model predictive control," *IEEE Transactions on Intelligent Vehicles*, 2024.

[13] S. P. Bharati, Y. Wu, Y. Sui, C. Padgett, and G. Wang, "Real-time obstacle detection and tracking for sense-and-avoid mechanism in UAVs," *IEEE Transactions on Intelligent Vehicles*, vol. 3, no. 2, pp. 185–197, 2018.

[14] Y. Nagao and K. Uchiyama, "Formation flight of fixed-wing UAVs using artificial potential field," in *29th Congress of the International Council of the Aerospace Sciences*, pp. 1–8, 2014.

[15] O. F. Archila, A. V. VandeWouwer, and J. Schiffer, "A multiple artificial potential functions approach for collision avoidance in UAV systems," *Submitted to IEEE Intelligent Transportation Systems Transactions*, 2024. https://doi.org/10.48550/arXiv.2503.22830.

[16] T. Weerakoon, K. Ishii, and A. A. F. Nassiraei, "An artificial potential field based mobile robot navigation method to prevent from deadlock," *Journal of Artificial Intelligence and Soft Computing Research*, vol. 5, no. 3, pp. 189–203, 2015.

[17] A. S. Morse, "Supervisory control of families of linear set-point controllers-part i. exact matching," *IEEE transactions on Automatic Control*, vol. 41, no. 10, pp. 1413–1431, 1996.

[18] S. Macenski, T. Foote, B. Gerkey, C. Lalancette, and W. Woodall, "Robot operating system 2: Design, architecture, and uses in the wild," *Science Robotics*, vol. 7, no. 66, p. eabm6074, 2022.

[19] S. Baldi, D. Sun, X. Xia, G. Zhou, and D. Liu, "Ardupilot-based adaptive autopilot: Architecture and software-in-the-loop experiments," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 58, no. 5, pp. 4473–4485, 2022.

[20] W. Ren and R. W. Beard, *Distributed consensus in multi-vehicle cooperative control*, vol. 27. Springer, 2008.

[21] C. C. Robusto, "The cosine-haversine formula," *The American Mathematical Monthly*, vol. 64, no. 1, pp. 38–40, 1957.

[22] S. Zhu, F.-H. Massmann, Y. Yu, and C. Reigber, "Satellite antenna phase center offsets and scale errors in gps solutions," *Journal of Geodesy*, vol. 76, pp. 668–672, 2003.

[23] A. Khosravian, J. Trumpf, R. Mahony, and T. Hamel, "Recursive attitude estimation in the presence of multi-rate and multi-delay vector measurements," in *2015 American Control Conference (ACC)*, pp. 3199–3205, IEEE, 2015.