A collision avoidance strategy for multirrotor UAVs based on artificial potential fields

Stefan van der Veeken
Faculty of Applied Engineering,
University of Antwerp
Antwerp, Belgium
Stefan.vanderVeeken@student.uantwerpen.be

Jamie Wubben
Carlos T. Calafate
Juan-Carlos Cano
Pietro Manzoni
jwubben@disca.upv.es
calafate@disca.upv.es
jucano@disca.upv.es
pmanzoni@disca.upv.es
Universitat Politècnica de València
Valencia, Spain

Johann Marquez-Barja
imec - University of Antwerp
Antwerp, Belgium
Johann.Marquez-Barja@uantwerpen.be

ABSTRACT

Much research is being done to use unmanned aerial vehicles (UAVs) in numerous different commercial and military applications. Additionally, the trend towards autonomous flying is growing steadily, and this is becoming more important as autonomous flight eliminates the need for a human pilot. Autonomous flying requires UAVs to have the ability to navigate in urban or challenging environments without causing collisions or endangering humans. To achieve this objective, a safe and reliable collision avoidance system (CAS) needs to be used. In particular, a CAS needs to successfully sense and detect a possible collision with an object to efficiently avoid the obstacle. Multiple different techniques exist to implement a CAS. In this paper we propose a collision avoidance protocol which is based on magnetic attraction and repulsion forces.

CCS CONCEPTS

• **Applied computing** → **Avionics**; • **Networks** → **Cyber-physical networks**.

KEYWORDS

UAV, sense and avoid, artificial potential fields, ArduSim

ACM Reference Format:
Stefan van der Veeken, Jamie Wubben, Carlos T. Calafate, Juan-Carlos Cano, Pietro Manzoni, and Johann Marquez-Barja. 2021. A collision avoidance strategy for multirrotor UAVs based on artificial potential fields. In *Proceedings of ACM Conference (Conference'17).* ACM, New York, NY, USA, 8 pages.

1 INTRODUCTION

Unmanned aerial vehicles (UAVs) have become more pervasive in our everyday lives over the recent years. This is due to the fact that UAVs have great potential to be used in a wide variety of military and commercial applications, and in other sectors (e.g. agriculture, search-and-rescue, surveillance and public safety [4]). The key benefits of UAVs over manned aerial vehicles are their low-cost, their ability to operate in hard to reach areas, and the inherent feature of being able to operate without the need of a human pilot within the aircraft, and thus not putting lives in danger[13]. Ideally, UAVs should be able to fly fully autonomously in urban and other challenging environments. On top of that, UAVs should also be able to fly in the integrated airspace without colliding with other aerial vehicles, and both the national regulation of each country, and EU Regulations 2019/947 and 2019/945 (EASA), should be taken into account in the UAV flight control and path planning [5].

Research on automated flight control for UAVs also becomes more challenging due to the more demanding requirements associated to the growing range of mission scenarios and flight conditions[10]. Dynamic environments such as cities pose a challenge for UAVs, or even UAV swarms, to operate without causing collisions with static obstacles (e.g. buildings) or dynamic objects (e.g. other UAVs). To face these challenges, collision detection and avoidance systems need to be implemented to ensure safe and reliable autonomous operation of a UAV. Robustness and fault tolerance are very important for these mechanisms, since a crash can have catastrophic consequences and pose a potential danger to humans. However, many solutions exist for collision sensing, detection, and avoidance. Typically, UAVs have a number of on-board sensors to accomplish the situational awareness and autonomous decision making. Crucial for autonomous operation is obstacle detection, collision avoidance, and path planning [13]. A generalized overview of a collision detection and collision avoidance system is provided in Figure 1 [13].

The first step in a collision avoidance system is the perception (or detection) of an obstacle, which is achieved by using sensors. Different types of sensors (i.e. active sensors and passive sensors) are used to perceive the environment and detect stationary or moving obstacles. A collision avoidance approach is used upon detection of a possible collision to determine a new course of action for the UAV, or to calculate a new route. Four main approaches can be defined:

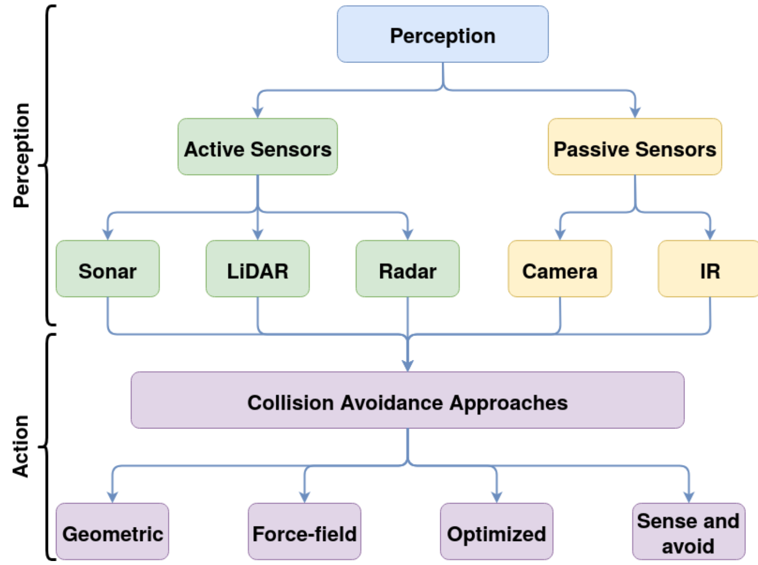


Figure 1: Generalized Collision Avoidance Modules [13].

- **Geometric approaches**: in these approaches the location, position and velocity information of the UAV and the obstacle are used to calculate a new route.
- **Force-field approaches**: by using artificial potential field (APF) methods, the UAV and obstacles are treated as charged particles it these approaches and, due to the attraction/repulsion forces, a new trajectory is planned [1].
- **Optimized approaches**: take into account the pre-known static obstacles, and they calculate the most optimal route to avoid them. However, these methods are not applicable to mobile obstacles.
- **Sense and avoid approaches**: make decisions at run-time for obstacle avoidance based on obstacle detection.

A UAV implements one or more of these approaches to solve a conflict. The ability to deal with such conflicts is crucial for the autonomous flying of UAVs to avoid crashing and causing any damage or harm to living beings.

In this paper, we propose a force-field method using APF to avoid collisions between multirotor UAVs. We present a formal description of our solution, and then validate the protocol using two different scenarios. Additionally, we will also determine the impact of increasing or decreasing safety levels through parameter tuning. Obtained results show the effectiveness of our solution at avoiding crashes while introducing small delays to the different UAV missions.

The remainder of this paper is organized as follows: Section 2 will discuss other related works. Section 3 will go into more detail about the solution we propose. Next, section 4 will describe the methods used, and the experiments performed to validate the solution. Afterwards, the results of these experiments will be discussed in section 5. This paper will be concluded in section 6 with some future works.

2 RELATED WORKS

Collision detection and avoidance is crucial for the autonomous flight of UAVs. Different strategies and approaches can be used, and each of these approaches has its unique features, as presented before. Therefore, a combination of strategies could also be used. However, constraints (i.e. computational power) should always be taken into account. According to [13] most of the collision avoidance strategies belong to one of the following categories:

2.1 Geometric approach

Geometric approaches calculate a new path given the geometrical information (e.g. velocity, location, heading) of the UAV itself and the obstacle involved. Once a possible collision is detected, a new trajectory will be calculated with the geometrical information from both parties to avoid the collision with the least possible deviation.

In [8] an Automatic Dependent Surveillance Broadcast (ADS-B) system is used to detect other airborne vehicles, and they propose a geometric approach to avoid collisions.

2.2 Force-field methods

Force-field or artificial potential field methods use the concept of charged particles to perform collision avoidance. Repulsion and attraction forces are used to respectively repel a UAV from an obstacle, or attract it to another object. It works well in static environments, but its application to dynamic environments is complex since it needs the geometric information and the knowledge of the motion of the robots and the obstacles [13].

In [11] authors propose an optimization algorithm based on artificial potential fields. They successfully implemented this method to resolve the problem of unreachable target and to take other UAVs into consideration as obstacles.

In [12] the authors suggest the use of an improved APF method to control a multi UAV cluster. They use the concept of a virtual core for cluster control. An attractive disturbance component is proposed together with the backtracking-filling method to solve the local minimum problem in the APF. They use the well-known k-means method to integrate and find the optimal attractive force between UAVs.

Liu and Zhao [6] propose the use of virtual waypoints to solve the local minimum problem. If a UAV is becomes trapped by dwelling around a local minimum, a virtual waypoint is calculated. The waypoint exerts extra forces, so the UAV can escape the local minimum. The virtual waypoint is cancelled after the UAV has escaped, so the original path planning can continue.

2.3 Optimised escape trajectories

In optimisation-based methods, collision avoidance will be looked at as a problem or equation that needs optimisation. Geographical information, and the position and size of the obstacles, is used in the optimisation calculations [1]. Considering the limited computational power of UAVs, the high complexity of these calculations pose a problem. To address this issue, several optimisation methods have been developed such as ant-inspired algorithms, genetic algorithms, Bayesian optimisation, gradient descent based methods, particle swarm optimisation, greedy methods, and local approximations [13].

Pérez-Carabaza et al.[9] propose a minimal time search (MTS) algorithm based on ant colonies to find an optimised collision-free trajectory for a UAV while ensuring continuous communication with the Ground Control Station.

2.4 Sense and avoid methods

In sense and avoid methods, UAVs are equipped with different sensors to detect obstacles and swiftly avoid them. The main focus of these methods is to reduce the computational power needed in UAVs to shorten response times. Therefore, sense and avoid methods are suitable for dynamic environments [13].

In [7], the authors propose an online 2D LIDAR approach to detect obstacles. Their method is able to distinguish between static and dynamic obstacles as well as tracking dynamic objects. The algorithm is proven to be more efficient regarding required memory and computation power needed.

3 PROPOSED SOLUTION

The solution we propose is based on the force fields method, which relies on the principles of magnetics to define an artificial potential field (APF). In particular, it is based on the interaction between attraction and repulsion forces. The attraction forces will be used to draw the UAV towards a specific target, while the repulsion forces will push the UAV in the opposite direction of the force, and will be used to avoid obstacles and other UAVs flying in close proximity. The resultant vector that determines the speed and direction of the UAV is generated by adding the attraction and repulsion forces together. In this paper the forces will only be applied on the x-axis and the y-axis, meaning that UAVs maintain their z-axis values fixed to avoid ground obstacles, and also to avoid surpassing the legal flight altitude limits. Hence, a UAV will not evade an obstacle by adjusting its flight altitude.

The mission of a UAV consists of one or more waypoints. Each of these waypoints will consecutively act as an attraction force for the UAV, and thus will pull the UAV towards each of them as the mission progresses. The UAVs themselves will act as repulsion forces for other UAVs, and thus they will repel each other to avoid collisions. The mission of a UAV consists of a list of waypoints, which contains the longitude and latitude of a destination. Hence, a GPS module must be installed on the UAV so that it can determine its position. The GPS error has a radius of approximately 5 meters, an issue which has to be taken into account when analysing the results.

The UAV is controlled using a mobility direction vector that consists of multiple vectors added together. Firstly, the vector pointing towards the next waypoint destination will be calculated. We call this the attraction vector. The size of the attraction force is dependent on the distance between the UAV and the waypoint itself. The attraction vector assumes the maximum speed of the UAV at startup. For the purpose of this paper, we assumed the maximum speed of a UAV to be 10 m/s (typical for most missions using multicopters). Once the distance between a UAV and its destination reaches a certain threshold value, the attraction force will start to decrease, and thus the UAV's speed will become lower, as depicted in figure 2. A UAV will not actively break when reaching its target; instead, it will break the inertia by slowing down. This is done to avoid the UAV overshooting its target. If a UAV flies past its target, it will slow down and change direction towards the destination again. If the UAV keeps overshooting its target, this will become an endless loop, and the destination will never be reached.

The UAV is controlled through vectors in the x, y, and z planes. These vectors have values in meter per second. The environment of the UAV consists of attraction and repulsion forces, which are mapped onto these speed vectors. The vectors are represented in polar form, with r being the magnitude of the vector, and θ being the angle. The vectors can be added together to create a new vector. The vector is converted back to its Cartesian form to use in the flight controller commands. To map the forces to a vector, firstly the magnitude is calculated, and secondly the angle θ is calculated using the position of the UAV and the waypoint. The magnitude of the attraction force of the waypoints on the UAVs is calculated using the following formula:

$$r_{attr} = \begin{cases} v_{max} & \text{if } d_0 < d \\ \frac{v_{max}}{2} & \text{if } d_s < d < d_0 \\ \frac{v_{max}}{d_s - d} & \text{if } d < d_s \end{cases} \tag{1}$$

Parameter d_0, in this case, is the distance until the UAV will fly at full speed. If the distance to the waypoint is smaller than this value, the speed will be halved until d_s, which is the slowdown distance. After reaching this distance, the attraction force will decrease inversely proportional to the distance. After testing, we decided on the following values for these parameters: $d_0 = 58$ meters and $d_s = 15$ meters, which results in the mapping depicted in figure 2.

Other UAVs flying nearby are considered as moving obstacles, and therefore generate a repulsion force to avoid collisions. For UAVs to gain awareness of each other, each UAV broadcasts beacons at a constant rate. These beacons include an identification, a timestamp, the most recent location, and the speed of the sender. Any

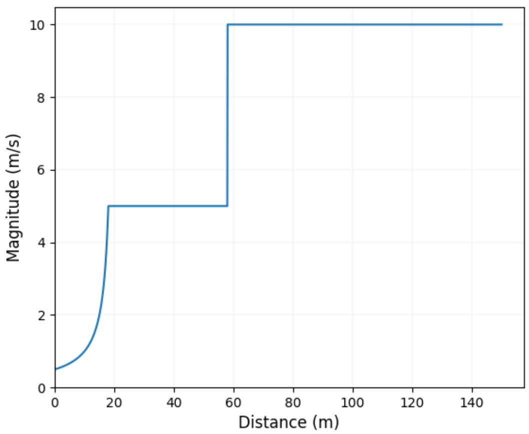


Figure 2: Attraction magnitude vs. distance.

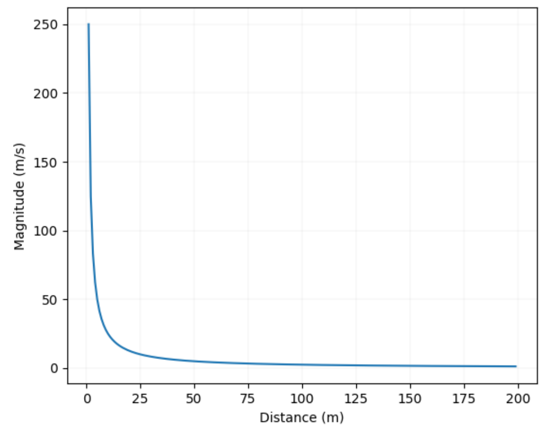


Figure 3: Theoretical repulsion magnitude vs. distance.

UAV, upon receiving such beacon, calculates the distance between the two UAVs with this information. Depending on the distance and the speed of the UAV, the message can either be ignored (large distances), or a repulsion vector is calculated (short distances). The parameter *timeToReact* is introduced to make sure that the amount of seconds a UAV has to react to other UAVs will be the same, despite one UAV having greater or lower speed than the other. The *timeToReact* is calculated in the following simple way:

$$timeToReact = \frac{d}{r} \tag{2}$$

With d being the distance between the UAVs, and r being the magnitude of the vector of the incoming UAV. The repulsion vector is inversely related to the distance between the UAVs, and it is calculated with the following formula:

$$r_{rep} = v_{max}/(\alpha \cdot d) \tag{3}$$

In this equation v_{max} is the maximum velocity of the UAV, α is a small scaling factor, and d is the distance between the UAVs. The repulsion will be small at great distances, but will become significant at smaller ranges, as depicted in figure 3. In this figure $\alpha = 0.04$.

According to equation 3, a distance of 25 meters will generate a repulsion vector with a magnitude of 10 m/s with $\alpha = 0.04$. As the maximum speed is set to 10 m/s, the maximum repulsion will already be achieved at 25 meters. For every UAV in close proximity, a repulsion vector is calculated. The sum of these separate vectors will result in the complete repulsion vector. If the magnitude of a vector is greater than v_{max} after adding them, it will be limited back to this number to make sure the UAV does not exceed the maximum speed.

The final mobility direction vector that decides the heading and speed of the UAV is determined by adding the attraction and repulsion vectors together. This is the vector that controls the UAV, and leads it away from obstacles and towards a desired target. In the following section the experimental setup used to validate our solution will be explained.

4 EXPERIMENTAL SETUP

The ArduSim tool [2] is used to develop the protocol and to perform the experiments detailed in this paper. ArduSim is a multi-UAV real-time simulator/emulator developed by the Computer Networks Group (GRC) of the Technical University of Valencia (UPV), Spain. It is an open-source software published under the Apache Licence 2.0, and it is freely available online [3]. ArduSim has some key characteristics that make it suitable as a development and testing environment. In this section a limited overview will be given. For more details please refer to [2].

ArduSim's key characteristics include:

- Communications API: ArduSim provides and easy to use API to get access to commonly used commands for UAVs (e.g. setting the speed, or moving the UAV to specific GPS coordinates). The API provides feedback to the developer regarding whether the execution of the command succeeded or failed.

- UAV-to-UAV communication: ArduSim relies on the IEEE 802.11a standard to provide communication between multiple UAVs, and between a UAV and the ground station. Three channel models are implemented with varying degrees of accuracy. If the protocol is being used on real UAVs, the User Datagram Protocol (UDP) is used to send and receive messages. Broadcast transmissions are simulated in case the protocol is only being tested in simulation.

- Scalability: ArduSim is designed as a multi-UAV simulator, and therefore it was developed bearing in mind scalability. ArduSim can support up to 100 UAVs in near real-time, and about 256 UAVs in soft real-time on a high-end PC (Intel Core i7-7700, 32GB DDR4 RAM). The number of supported UAVs can vary strongly depending on the actual hardware used.

- Protocol deployment: ArduSim was designed to ease the deployment of protocols on real UAVs. ArduSim simulates

Table 1: Algorithm parameter values per test.

Test case	timeToReact	Safety level
#1	15 (s)	Standard
#2	5 (s)	Low
#3	25 (s)	High

the communication between UAVs close to reality, and utilizes the same protocols. Switching between simulation or deploying the protocol on a real UAV is achieved by merely changing an execution parameter.

To validate our proposal, two different scenarios were executed. The first scenario involves two UAVs that are flying towards each other on a straight, vertical line over a distance of 500 meters. The second scenario involves one UAV taking over the other at a greater speed on the same straight, vertical trajectory. To accomplish the overtaking in this second scenario, the maximum speed of one of the UAVs was set to 5 m/s, while the maximum speed of the other UAV was kept at 10 m/s. A sensitivity analysis was performed for the $timeToReact$ parameter in these two scenarios to determine the most optimal tradeoff between time overhead and safety; for our study, we assume that distances between UAVs lower than 15 meters do not meet the minimum safety requirements expected. The different test cases defined, and the corresponding parameter values, can be found in table 1.

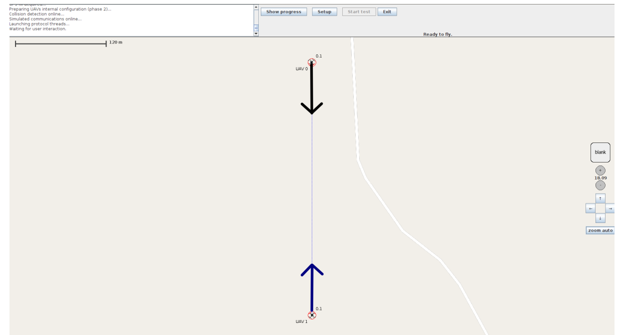
5 RESULTS

In this section we proceed to validate our contribution by devising a set of experiments that allow to assess its correct behavior, and determine the time and distance overheads associated with our solution.

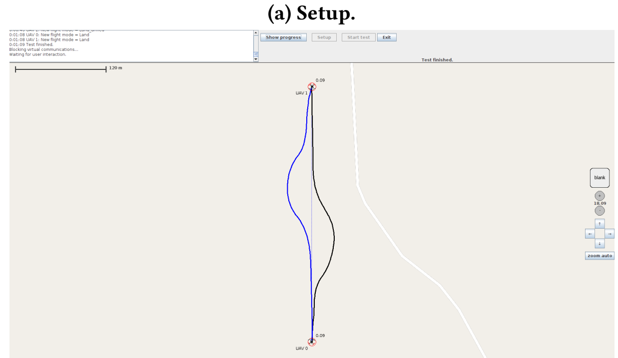
The simulation experiments are performed with UAVs flying at a fixed value for the z-axis (10 m/s). The data is gathered from the moment the UAVs are already airborne up until the moment they have safely landed on the ground. Firstly, a scenario where the UAVs fly towards each other will be discussed; afterwards, an overtaking scenario will be investigated. To investigate the effectiveness of the protocol, a comparison is made between the UAVs following the planned path without any collision avoidance protocol, and with the proposed protocol activated. The time to reach the destination from the moment UAVs are airborne up until the UAVs land, are displayed in table 2.

Table 2: Time to reach destination and land without collision avoidance for the two reference scenarios.

Scenario	UAV id	Total time
A (Head on)	0	64.25 s
	1	64.35 s
B (Overtake)	0	92.74 s
	1	86.75 s



(a) Setup.



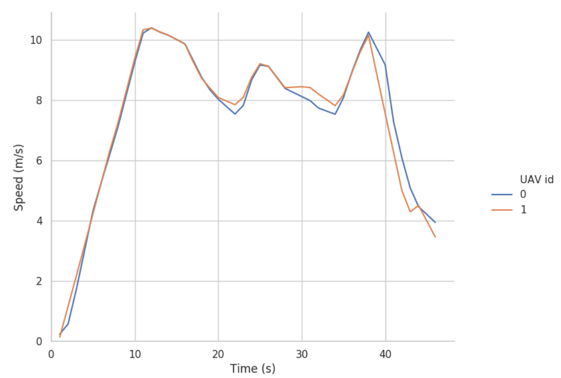
(b) Trajectory.

Figure 4: Scenario A: initial mission direction and actual trajectories.

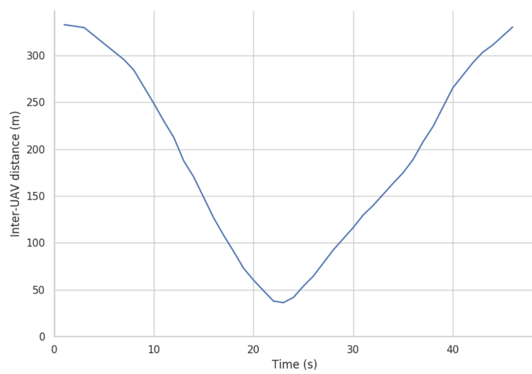
5.1 Scenario A: same trajectory, opposite directions

In the first scenario, the UAVs fly directly towards each other on the same trajectory. The initial setup of the experiment and the trajectory the UAVs have followed is visually displayed in figure 4. As seen in figure 4a, the UAVs are on a vertical trajectory and facing each other, on a collision course. In 4b we can see that the UAVs neatly deviate from the planned path to avoid a collision, and continue towards their target. The parameters for this first test can be found in table 1. In figure 5a the progression of UAV speed throughout time is displayed for both UAVs. It highlights that the tight interactions between both UAVs lead to similar changes in their speed pattern. In figure 5b the distance between the UAVs throughout time is displayed. Since $timeToReact$ was set to 15 seconds, the UAVs will start interacting within a 150 meters range, according to formula 2. From these figures, we can see that the UAVs reach their maximum velocity (10 m/s) after 10 seconds, and that the speed starts going down when they get within range; at that time they start their maneuver to avoid a collision. An overview of the most important statistics about this experiment can be found in table 3. This table includes the total time it took for the UAV to fly towards its target and land, the minimum distance that was kept between the UAVs, and the additional flight time required due to the collision avoidance maneuver.

For the first test case (standard safety levels), the minimum distance that was kept between the UAVs during this experiment was



(a) Speed vs. Time.



(b) Distance vs. Time.

Figure 5: Scenario A performance comparison for test case #1.

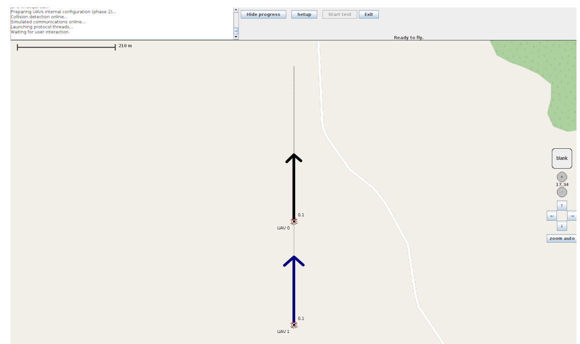
36.21 meters, and it only took 4.4-4.5 seconds longer for the UAVs to complete their mission and avoid a collision.

In the second test case (low safety levels) the *timeToReact* value is set to 5. This experiment only took about one second longer than the reference scenario (high efficiency), but the minimum distance between the UAVs was of only 8.02 meters, a value considered too low. Notice that the GPS module on the UAVs has an error radius of about 5 meters. Taking this into consideration, the minimum distance of 8.02 meters in this scenario is not enough to guarantee a collision-free trajectory and the desired safety levels.

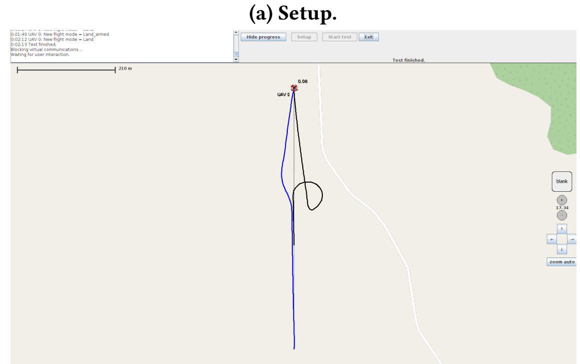
In the third and last test case (high safety levels), the *timeToReact* parameter is set to 25. According to table 3, the excess time for this test is on average of 20 seconds, and the minimum distance between the UAVs is 121.27 meters. This minimum distance is more than enough to guarantee a collision trajectory. However, considering that the collision avoidance maneuver extended the flight duration of each UAV by about 20 seconds, we can conclude that setting the $timeToReact = 25$ is not the optimal solution if time optimization is a priority.

Table 3: Experimental results: scenario A.

Test case	UAV id	Total time (s)	Min. distance (m)	Excess time (s)
#1	0	68.76	36.21	4.51
	1	68.76	36.3	4.40
#2	0	65.25	8.02	1.00
	1	65.36	8.02	1.00
#3	0	85.56	121.27	21.31
	1	83.66	121.27	19.30



(a) Setup.



(b) Trajectory.

Figure 6: Scenario B: initial mission direction and actual trajectories.

5.2 Scenario B: overtaking

The second scenario involves an overtake by a faster UAV on a slower UAV that is ahead. The scenario can be seen in figure 6. In the initial setup (see Figure 6a), we can observe that UAV with id 0 is in front, and that it will be overtaken by the UAV 1. The actual trajectory the UAVs have taken is shown in figure 6b. The faster UAV performs the overtaking action by going around the other UAV through the left. The slower UAV moves to the opposite side and performs a small loop to make room for the faster UAV to perform the overtake, resuming the movement towards its target immediately afterward. The parameters used in this second set of experiments are the same as in the first scenario (see table 1). In figure 7a the differences in terms of velocity between the two UAVs

Table 4: Experimental results: scenario B.

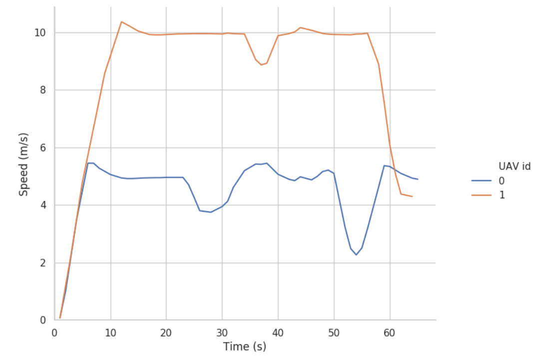
Test case	UAV id	Total time (s)	Min. distance (m)	Excess time (s)
#1	0	132.46	59.79	39.71
	1	87.63	59.79	0.88
#2	0	105.06	17.58	12.31
	1	87.14	15.77	0.40
#3	0	159.36	113.00	66.62
	1	87.44	112.58	0.69

is clearly visible. In particular, for t=24 seconds, UAV 1's speed is affected, and it starts the collision avoidance maneuver while UAV 0 is still advancing at maximum speed. This is due to the value of the $timeToReact$ parameter. Since UAV 0 has a velocity which is twice as fast as UAV 1's, from equation 2 it can be deducted that $d = timeToReact * r$, and thus UAV 0 will take action to avoid a collision before UAV 1 does. Figure 7b shows the distance between both UAVs throughout time. The same pattern can be seen as in the first scenario. The distance between the UAVs shrinks and then it grows again once the overtaking has been done. In table 4 the full experimental results, including the three test cases, can be seen.

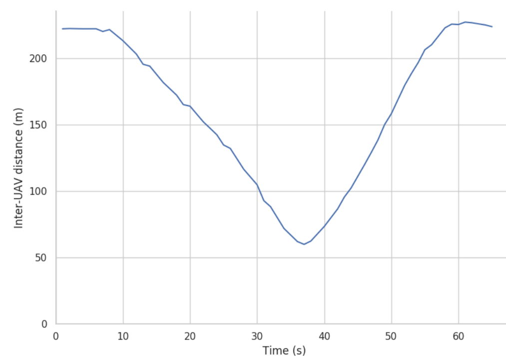
In the first test case (standard safety levels), the minimum distance is 59.79 meters, which is more than enough to guarantee a collision free path. On the other hand, the total time it took UAV 0 to reach its destination has increased by 39.7 seconds. Contrarily, UAV 1's time increased by less than a second, which is an excellent result.

In the second test case (low safety levels), the minimum distance assumed a value of 15.77 m which is just enough to ensure a collision free path taking the GPS error into consideration. The excess time for UAV 0 is now 12.3 second, a better result than the first test case. For UAV 1 the excess time decreased further to a mere 395 ms.

In the third and last test case (high safety levels), the minimum distance between the UAVs was 112.58 meters. This is obviously more than enough to guarantee that there are no collisions. UAV 0 performed the mission in 159 seconds, which is 66.6 seconds longer than the reference value, and can be considered a very high overhead. UAV 1 only took 690 ms longer to complete its mission.



(a) Speed vs. time.



(b) Distance vs. time.

Figure 7: Scenario B performance comparison for test case #1.

6 CONCLUSION AND FUTURE WORK

In future scenarios where UAVs become more pervasive, potentially fulfilling different missions by sharing the same aerial space, having efficient collision avoidance mechanisms becomes a necessity to enable their autonomous navigation. This is especially applicable to urban environments, where many of these flying devices are expected to coexist.

In this paper we proposed a collision avoidance strategy based on an artificial potential field method. Our solution uses attraction forces to pull the UAV towards a desired target (e.g. waypoints), and repulsion forces to push the UAV away from dynamic obstacles, in our case other UAVs. The protocol was validated for two typical scenarios where different UAVs share similar ongoing/returning routes. The results obtained are quite promising, and show that the protocol works correctly in these scenarios by fully avoiding collisions. Additionally, we showed how the $timeToReact$ parameter can be adjusted to achieve different trade-offs between safety and mission time overhead. In particular, we show that a small value for the $timeToReact$ parameter (low safety levels) may result in a collision in some cases as the UAVs might be too slow to react. On the contrary, large values for the $timeToReact$ parameter result in evasion maneuver that are extremely safe, but have a long execution time, which is undesirable when time optimisation is a priority.

As future work we intend to optimise the protocol and test it in different scenarios. Extended validations should also be performed to investigate the influence of other protocol parameters such as distance thresholds, or the impact of external factors such as wind conditions. This way, the protocol can become more versatile, and it can be made suitable for different scenarios. The protocol has been developed and tested for dynamic obstacles (e.g. UAVs), but it could also be expanded to be used for static obstacles. However, a different obstacle detection strategy would be required to achieve this characteristic, as it is clearly impossible to equip all static obstacles with the needed equipment to send beacons.

ACKNOWLEDGMENTS

This work was partially supported by the Erasmus+ Program, and by the "Ministerio de Ciencia, Innovación y Universidades, Programa Estatal de Investigación, Desarrollo e Innovación Orientada a los Retos de la Sociedad, Proyectos I+D+I 2018", Spain, under Grant RTI2018-096384-B-I00.

REFERENCES

[1] B. M. Albaker and N. A. Rahim. 2009. A survey of collision avoidance approaches for unmanned aerial vehicles. *International Conference for Technical Postgraduates 2009, TECHPOS 2009*. https://doi.org/10.1109/TECHPOS.2009.5412074

[2] Francisco Fabra, Carlos T. Calafate, Juan Carlos Cano, and Pietro Manzoni. 2018. ArduSim: Accurate and real-time multicopter simulation. *Simulation Modelling Practice and Theory* 87 (2018). https://doi.org/10.1016/j.simpat.2018.06.009

[3] GRCDev. 2020. ArduSim: Accurate and real-time multi-UAV simulation. https://github.com/GRCDEV/ArduSim. Accessed: September 10, 2021.

[4] S. Hayat, E. Yanmaz, and R. Muzaffar. 2016. Survey on Unmanned Aerial Vehicle Networks for Civil Applications: A Communications Viewpoint. *IEEE Communications Surveys Tutorials* 18, 4 (Fourthquarter 2016), 2624–2661. https://doi.org/10.1109/COMST.2016.2560343

[5] Yazdi I. Jenie, Erik Jan Van Kampen, Joost Ellerbroek, and Jacco M. Hoekstra. 2017. Taxonomy of Conflict Detection and Resolution Approaches for Unmanned Aerial Vehicle in an Integrated Airspace. *IEEE Transactions on Intelligent Transportation Systems* 18 (2017). Issue 3. https://doi.org/10.1109/TITS.2016.2580219

[6] Yuecheng Liu and Yongjia Zhao. 2017. A virtual-waypoint based artificial potential field method for UAV path planning. *CGNCC 2016 - 2016 IEEE Chinese Guidance, Navigation and Control Conference*, 949–953. https://doi.org/10.1109/CGNCC.2016.7828913

[7] Wang Min, Voos Holger, and Su Daobilige. 2018. 2018 15th International Conference on Control, Automation, Robotics and Vision (ICARCV). (2018).

[8] Jung-Woo Park, Hyon-Dong Oh, and Min-Jea Tahk. 2008. UAV Collision Avoidance Based on Geometric Approach.

[9] Sara Pérez-Carabaza, Jürgen Scherer, Bernhard Rinner, José A. López-Orozco, and Eva Besada-Portas. 2019. UAV trajectory optimization for Minimum Time Search with communication constraints and collision avoidance. *Engineering Applications of Artificial Intelligence* 85 (10 2019), 357–371. https://doi.org/10.1016/j.engappai.2019.06.002

[10] Hassan Shraim, Ali Awada, and Rafic Youness. 2018. A survey on quadrotors: Configurations, modeling and identification, control, collision avoidance, fault diagnosis and tolerant control. Issue 7. https://doi.org/10.1109/MAES.2018.160246

[11] Jiayi Sun, Jun Tang, and Songyang Lao. 2017. Collision Avoidance for Cooperative UAVs with Optimized Artificial Potential Field Algorithm. *IEEE Access* 5 (8 2017), 18382–18390. https://doi.org/10.1109/ACCESS.2017.2746752

[12] Enming Wu, Yidong Sun, Jianyu Huang, Chao Zhang, and Zhiheng Li. 2020. Multi UAV Cluster Control Method Based on Virtual Core in Improved Artificial Potential Field. *IEEE Access* 8 (2020), 131647–131661. https://doi.org/10.1109/ACCESS.2020.3009972

[13] Jawad N. Yasin, Sherif A.S. Mohamed, Mohammad Hashem Haghbayan, Jukka Heikkonen, Hannu Tenhunen, and Juha Plosila. 2020. Unmanned Aerial Vehicles (UAVs): Collision Avoidance Systems and Approaches. *IEEE Access* 8 (2020). https://doi.org/10.1109/ACCESS.2020.3000064