Hindawi
International Journal of Aerospace Engineering
Volume 2021, Article ID 8819618, 13 pages
https://doi.org/10.1155/2021/8819618

Research Article

Three-Dimensional Autonomous Obstacle Avoidance Algorithm for UAV Based on Circular Arc Trajectory

Jiandong Guo,[1] **Chenyu Liang**,[2] **Kang Wang**,[2] **Biao Sang**,[2] **and Yulin Wu**[2]

[1]*Key Laboratory of Unmanned Aerial Vehicle Technology of the Ministry of Industry and Information Technology, Nanjing University of Aeronautics and Astronautics, Nanjing 210016, China*
[2]*College of Automation Engineering, Nanjing University of Aeronautics and Astronautics, Nanjing 210016, China*

Correspondence should be addressed to Jiandong Guo; bh4rbc@163.com

Received 14 August 2020; Revised 22 February 2021; Accepted 3 April 2021; Published 22 April 2021

Academic Editor: Nicolas Avdelidis

This paper proposes an innovative and efficient three-dimensional (3D) autonomous obstacle algorithm for unmanned aerial vehicles (UAVs) which works by generating circular arc trajectories to avoid obstacles. Firstly, information on irregular obstacles is obtained by an onboard detection system; this information is then transformed into standard convex bodies, which are used to generate circular arc avoidance trajectories, and the obstacle avoidance problem is turned into a trajectory tracking strategy. Then, on the basis of the geometric relationship between a UAV and obstacle modeling, the working mechanism of the avoidance algorithm is developed. The rules of obstacle detection, avoidance direction, and the criterion of avoidance success are defined for different obstacle types. Finally, numerical simulations of different obstacle scenarios show that the proposed algorithm can avoid static and dynamic obstacles effectively and can implement obstacle avoidance missions for UAVs well.

1. Introduction

UAVs are widely used in military and civilian fields because of their outstanding advantages, such as flexibility, light weight, strong mobility, and good concealment [1]. Nowadays, the growth and applications of UAV technologies not only change the development direction of many industries but also bring huge market and economic benefits [2]. However, with the complication of flight missions and usage scenarios, higher requirements are placed on the autonomous flight capabilities and intelligence of UAVs, and autonomous obstacle avoidance technology for UAVs has gradually become one of the most important technologies [3].

The real-time autonomous obstacle avoidance problem studied in this article refers to the avoidance of various collisions during flight through the target path by avoidance actions, which are calculated according to the obstacle information based on known data or data coming from an onboard detection system in real time, as shown in Figure 1. Meanwhile, the difficulty of this problem lies in the irregularity, dynamics, and complexity of obstacles and three-dimensional space.

Sense and avoid (S&A) is a major pattern for obstacle threat avoidance. The aircraft's obstacle avoidance methods can usually be divided into following two categories:

(1) The method of resolving the new trajectory based on path planning: the main idea of this method is to transform the obstacle avoidance problem into a path planning problem [4, 5]. With the advances of research in this field, many significantly improved path planning algorithms have been proposed, including the Voronoi diagram (VD) [6] based on graph theory, artificial potential field (APF) [7, 8] based on field theory, the RRT [9] algorithm based on sampling theory, the A ∗ [10] algorithm based on heuristic information, and other algorithms based on swarm intelligence optimization theories [11–13].

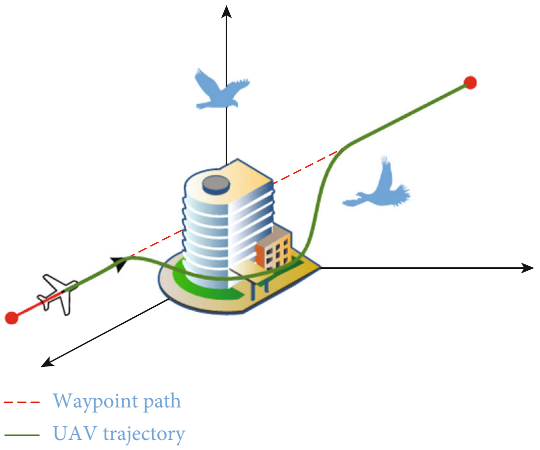


Figure 1: Obstacle avoidance problem.

While these algorithms have advantages, they also have certain limitations. VD has fast speed in its searching trajectory with high safety, but it is difficult to apply this method to a complex or high-dimensional dynamic environment. APF is simple in principle and has high computing efficiency, but it can easily fall into the local optimum with high time cost. The RRT algorithm does not require preprocessing of the state space and has a wide search range in the task space with high coverage and fast convergence, but the searching path is often not optimal. The A ∗ algorithm is simple and efficient, but it can easily fall into an infinite loop and has many vertices in the trajectory

The swarm intelligence algorithm is used to simulate the evolutionary strategy of a population in the natural environment or the characteristics of survival of a biological group, and uses multiple individuals in the group to perform an iterative search in the task space to find the optimal solution, such as the genetic algorithm (GA) [11, 14], the ant colony algorithm (ACO) [12, 15], and the differential evolution algorithm (DE) [13, 16]. Additionally, this kind of method has low computational complexity, flexible application, strong search ability, and good robustness; however, it requires huge computing resources and high-quality equipment.

Refs. [5, 17] present comparative cases of these path planning algorithms for different scenarios and obstacle layouts in terms of time consuming and optimal solutions, and the simulation results show that the designed path planning algorithms often conflict with each other in terms of optimality and convergence.

(2) The second method of resolving the guidance law based on geometrical relationships: this method calculates the guidance law of obstacle avoidance action according to the relative distance [18], speed [18, 19], angle [20], and other key information [21] between the aircraft and collisions. Compared to the obstacle avoidance methods for path planning mentioned above, the key factor for this type of method is the modification of the UAV's trajectory in real time to avoid collisions along its path with the help of feedback datum from onboard sensors taken from the environment, which often is divided into two steps, namely, detection and avoidance. The drone's exploration capabilities depend on the onboard sensors, which are commonly divided into cooperative and noncooperative groups. Cooperative technologies are those that can receive data from the obstacles which can then broadcast its information. Among the leading cooperative technologies are Traffic Alert and Collision Avoidance System (TCAS) [22] and Automatic Dependent Surveillance Broadcast (ADS-B) [23]. The relative distance, bearing, altitude, latitude, longitude, etc. can be obtained from the threat. However, these technologies have not yet shown enough maturity for utilization in every drone. Noncooperative technologies depend on measuring instruments, such as infrared cameras [21], photoelectric radars [18], laser rangefinders [24], and acoustooptic processors [25]. Each sensor has different limitations, and no single sensor can provide all the necessary information for obstacles. The optical and acoustic sensors are most suitable for angular measurements, while radar and laser are best for ranging. Therefore, research has been devoted to the field of sensing and detection, including improved sensor sensitivity [26, 27].

Many obstacle avoidance algorithms have been proposed according to the geometrical relationships between aircrafts and collisions. Sasongko et al. [19] proposed a method which calculated a set of avoidance waypoints to avoid collisions based on an obstacle model and an aircraft's speed vector. Zheng et al. [18] established fuzzy rules to obtain avoidance maneuver options according to the aircraft's forward speed and the distance between the vehicle and obstacles. Al-Kaff et al. [21] proposed an avoidance method through tracking along the safe flight boundary consisting of a set of obstacle feature points, which were obtained by determining the coordinate relationship between the aircraft and obstacles in real time. Overall, these obstacle avoidance methods based on geometrical relationships are easy to implement and test, but they are difficult to apply in actual complex flight environments.

This work is aimed at solving the problem of real-time autonomous obstacle avoidance for UAVs in three-dimensional space, and an innovative obstacle avoidance algorithm based on a circular arc trajectory is proposed according to the geometric relationship between UAVs and obstacles through obstacle modeling. This paper is organized as follows: Section II describes the obstacle avoidance problem and obstacle modeling. Section III analyzes the working mechanism of the novel algorithm, which we have named the Circular arc Trajectory Geometric Avoidance (CTGA) algorithm. Section IV presents the simulation results for static simple, complex, and dynamic obstacles, respectively, and analyzed the control accuracy and performance. Section V gives the conclusions and future work.

2. Mathematical Description of Obstacle Avoidance Problem

2.1. Modeling of Obstacles. When a UAV is performing a flight mission, there are many potential obstacles that may exist on the flight path, such as buildings, mountains, and trees. Most of these obstacles are often irregular and difficult to handle directly, and the obstacle data detected by onboard sensors is often incomplete. Paying too much attention to the details of obstacle shapes will greatly increase the difficulty and calculation of the obstacle avoidance algorithm design and reduce the obstacle avoidance efficiency. Therefore, when the detection system of the aircraft captures the collision threat information, a series of sampling points are obtained. According to these sampling points, an obstacle circle is fitted by some fitting algorithm, such as the least square method, but it has a large calculation amount and is not suitable for real-time obstacle avoidance for UAVs. We can compute the radius R and center $[o_x, o_y]^T$ of an obstacle circle using the following:

$$R = \frac{1}{2} \max \bigcup_{i=1}^{m-1} \bigcup_{j=i+1}^{m} \sqrt{\left(x_i - x_j\right)^2 + \left(y_i - y_j\right)^2}, \tag{1}$$

$$\begin{bmatrix} o_x \\ o_y \end{bmatrix} = \frac{1}{2} \begin{bmatrix} x_{i,\text{idx}} - x_{j,\text{idx}} \\ y_{i,\text{idx}} - y_{j,\text{idx}} \end{bmatrix}, \tag{2}$$

where $(x_i, y_i)(i = 1, 2, \cdots, m)$ is the convex hull of those sampling points detected by the onboard system, idx is the index of equation (1).

In this way, an obstacle circle is fitted which can cover the whole or key parts of the obstacle, it not only simplifies the description of obstacle, but it also facilitates the design of obstacle avoidance algorithm according to obstacle modeling.

It is possible that in some cases, the UAV may encounter obstacle obstructions that are larger than the distance between the aircraft and an obstacle. In other cases, the detection system may not be able to capture the whole size or key parts of the obstacle, due to its detection limitation. To deal with such situations, the UAV can continuously detect obstacle information during the avoidance process, and multiple obstacle circles are used to characterize the large obstacle. Examples showing the determination of an obstacle circle based on detection sampling points can be seen in Figures 2 and 3.

In three-dimensional space, it is reasonable that obstacles can be modeled with one or more standard convex bodies according to the obstacle detection system, such as a hemisphere, a cylinder, a cone, and a circular table. Their approximate expressions can be unified as follows [28]:

$$\Gamma = \left(\frac{x - x_0}{a}\right)^{2p} + \left(\frac{y - y_0}{b}\right)^{2q} + \left(\frac{z - z_0}{c}\right)^{2r}, \tag{3}$$

where (x, y, z) is an arbitrary point on the obstacle model, (x_0, y_0, z_0) represents the coordinate of the center point of the obstacle model, the parameters a, b, c and p, q, r are constants and determine the shape and size of the obstacle model. Figure 4 shows the various types of typical convex bodies obtained from different detection data combinations. $\Gamma < 1$, $\Gamma = 1$, and $\Gamma > 1$ express the inside, surface, and outside of the obstacle model, respectively.

According to these obstacle models, when a UAV encounters obstacles in the preplanned flight path, simple obstacles can be modeled by a single standard convex body and complex obstacles can be represented by a combination of multiple convex bodies, as depicted in Figure 5. Hence, circular arc trajectories can be designed, wherein the autonomous obstacle avoidance problem is transformed into a trajectory tracking strategy.

2.2. Obstacle Avoidance Problem Description. In this article, both the surface and the inside of an obstacle model are regarded as no-fly zones or threat zones, i.e., $\Gamma > 1$. The problem of autonomous obstacle avoidance for a UAV can be described as how to keep the UAV always outside the no-fly zone or threat zone during the flight mission. Assume that the mission space is S; S_{obs} represents the threatened area of the obstacle, i.e., $\Gamma \leq 1$, and S_{free} represents the flyable safety area, i.e., $\Gamma > 1$. The relationship between the zones is shown as follows:

$$\begin{cases} S = S_{\text{obs}} \cup S_{\text{free}}, \\ S_{\text{obs}} \cap S_{\text{free}} = \varnothing. \end{cases} \tag{4}$$

When the UAV detects an obstacle obstructing the target flight path, it must be able to adjust the flight path to avoid the obstacle. The CTGA algorithm can generate the circular avoidance trajectory in real time, and tracks the trajectory to achieve safe flight. So the flight path for the UAV in the space can be expressed as follows:

$$P = P_{\text{target}} + \bigcup_{i=1}^{n} P^i_{\text{avoid}}, P^i_{\text{avoid}} = \{\Gamma > 1\}, \tag{5}$$

where P_{target} is the target flight path, P^i_{avoid} is the circular avoidance trajectory for the obstacle at stage i, and n is the number of obstacles.

3. Circular Arc Trajectory Geometric Avoidance (CTGA) Algorithm

3.1. CTGA Principle and Procedure. The main idea of the CTGA algorithm is transforming the obstacle avoidance problem into path planning and path following, and it is based on the geometric relationship between the aircraft and the obstacle. The algorithm assumes that the position and velocity of the obstacles are given and accessible to the aircraft, then it designs a circular arc avoidance trajectory according to the obstacle model (simple or complex, static or dynamic). Therefore, the UAV avoids the obstacle by tracking the circular arc avoidance trajectory.

As illustrated in Figure 6, first, consider that a UAV is flying while tracking a predefined flight path; then, the UAV detects the collision threat based on the information of the

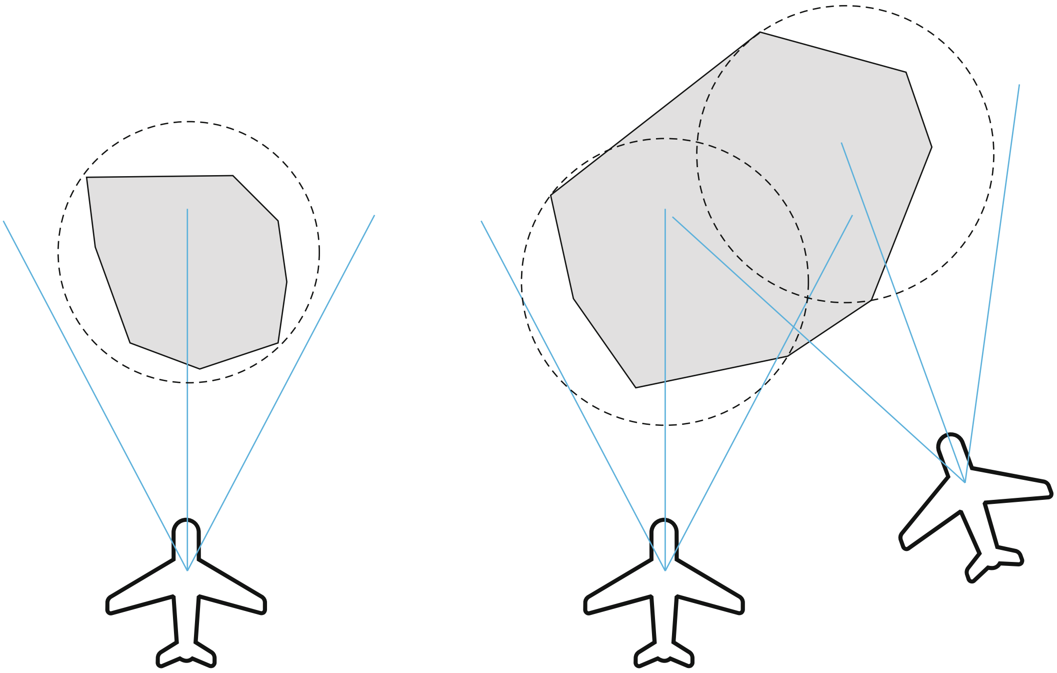
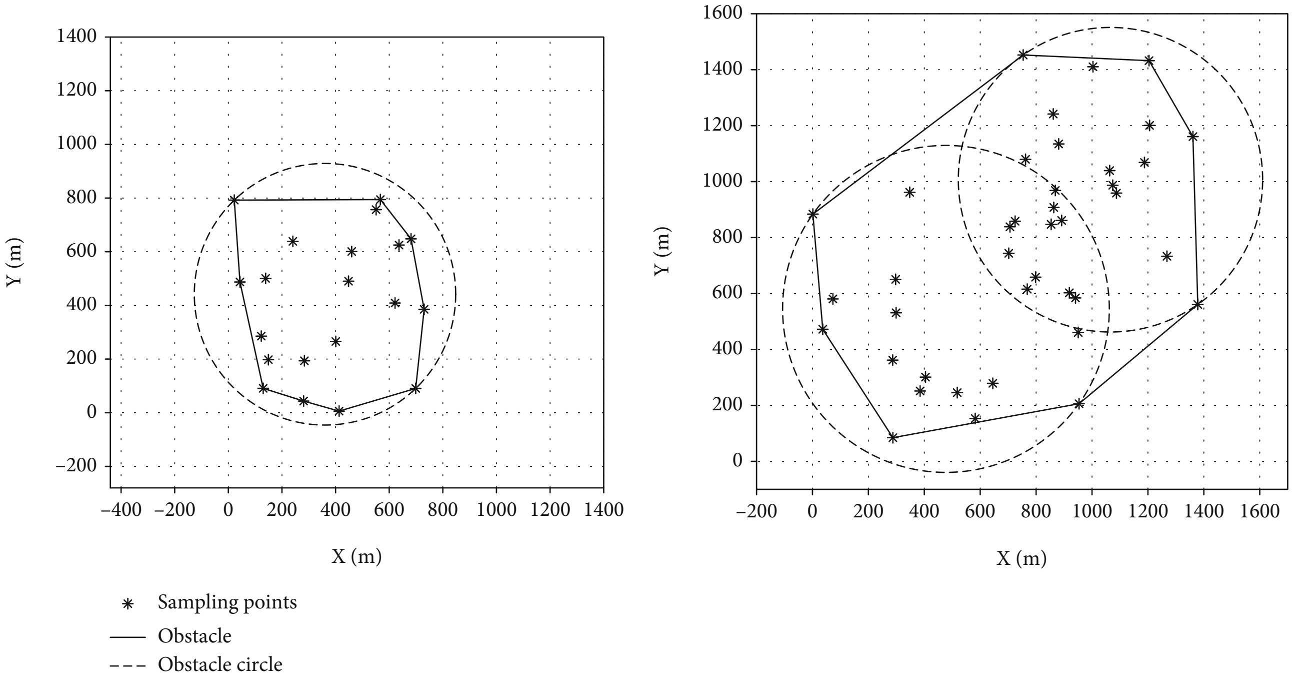


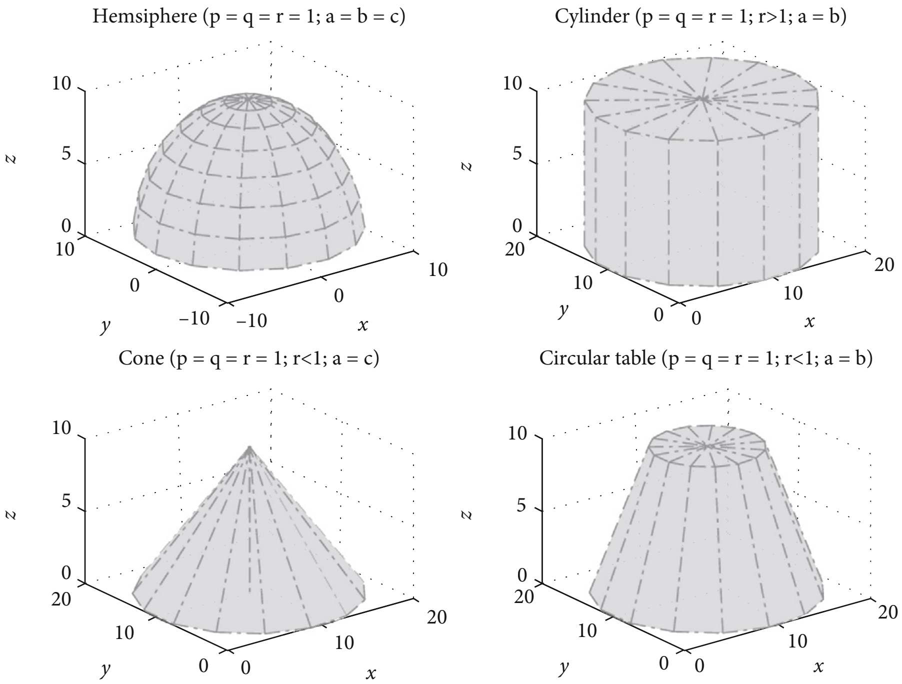
Figure 2: Obstacle detection.



(a) Simple obstacle

(b) Complex obstacle

Figure 3: Obstacle circle fitting.


Hemsiphere (p = q = r = 1; a = b = c)

Cylinder (p = q = r = 1; r>1; a = b)

Cone (p = q = r = 1; r<1; a = c)

Circular table (p = q = r = 1; r<1; a = b)

Figure 4: Typical convex body obstacle model.

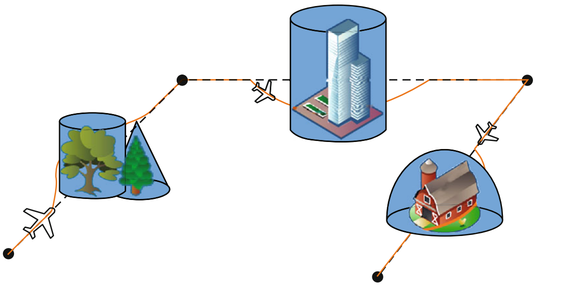


Figure 5: Illustration of UAV avoiding obstacles.

detected parts at point T_1, and an obstacle model enclosing the whole or main part of the obstacle can be constructed, with the hemisphere model being an example for this case. Therefore, a circular arc avoidance template trajectory as a reference path is generated in real time according to the obstacle model, whose radius corresponds to the obstacle model at the current altitude of the aircraft, and constantly changes with the aircraft altitude; the real-time avoidance trajectory can be expressed as follows:

$$\begin{cases} (x(t)-o_x)^2+(y(t)-o_y)^2=R(t)^2, & R(t)\infty H(t), \\ z(t)=h_1+d\cos\lambda\tan\gamma, \end{cases} \tag{6}$$

where H represents the real-time altitude of the UAV, γ is the flight path angle, d is the distance from the UAV to WP_1 under horizontal projection, and λ is the angle between the segment made by the UAV's center and WP_1, and the target path under horizontal projection.

Furthermore, in order to ensure that the generated circular arc trajectory can bypass the obstacles with a feasible path for the aircraft, a safe distance R_{safe} is added to the radius, which should be determined by the flight performance of the drones, so the radius of the circular arc trajectory is $(R(t)+R_{\mathrm{safe}})$. In addition, limited by the minimum turning radius $R_{\min}$ of the aircraft, the radius of the circular arc trajectory is as follows:

$$R(t)=\max\left(R_{\min}, R(t)+R_{\mathrm{safe}}\right), \tag{7}$$

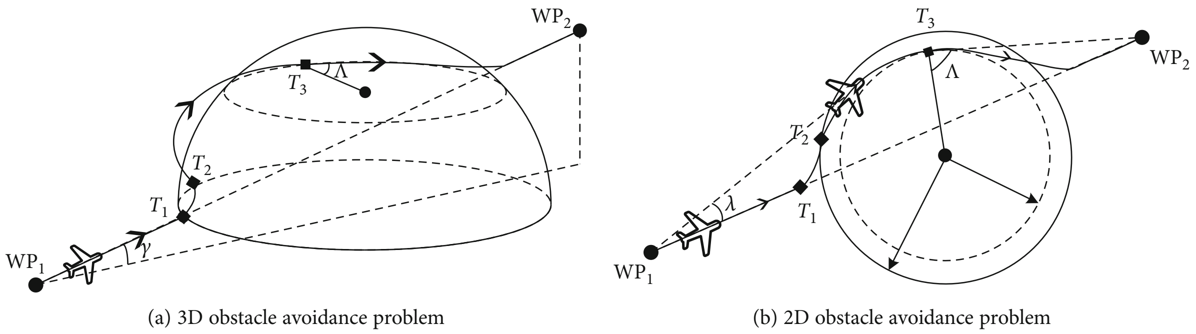
where the expression of $R_{\min}$ is as follows:

$$R_{\min}=\frac{V_n^2}{g\tan\left(\phi_{\max}\right)}. \tag{8}$$

Here, the value of the minimum turning radius is determined by the ground speed and the maximum roll angle of the UAV.

Immediately, the guidance system of the UAV begins an avoidance maneuver with the minimum turn radius possible of reaching the template trajectory tangentially at point T_2 (it should be pointed out that in order to reduce the risks involved with using the minimum turning radius during the maneuver avoidance flight, the strategy of the turn should be considered in advance, so that the turning radius can be slightly increased in actual flight), and tracks the circular arc template trajectory. Finally, when the UAV avoids the collision successfully at point T_3, it will return to its original navigation course immediately.

The CTGA algorithm includes three major processes: obstacle detection, avoidance direction selection, and



(a) 3D obstacle avoidance problem

(b) 2D obstacle avoidance problem

Figure 6: UAV obstacle avoidance with CTGA.

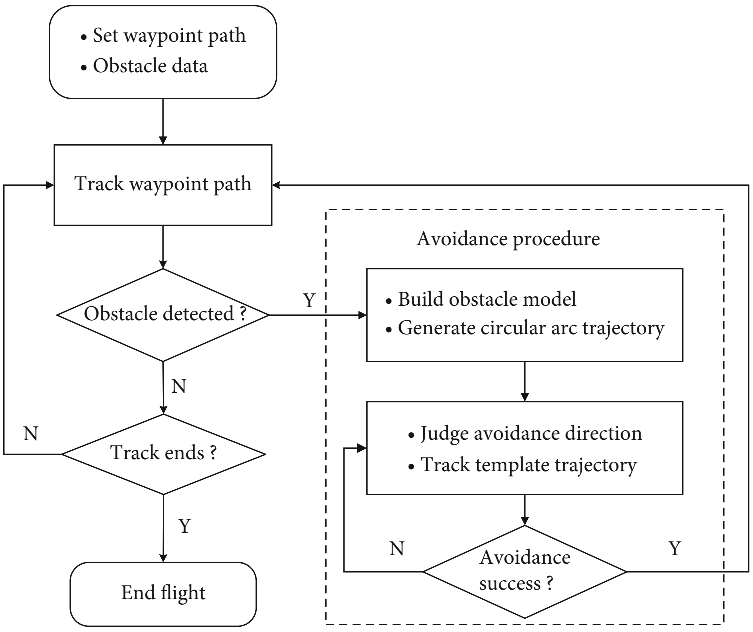


Figure 7: UAV guidance procedure.

avoidance success criteria determination. The detailed obstacle avoidance process, in conjunction to following the predefined waypoint path, is described in Figure 7.

3.2. Obstacle Detection and Avoidance Determination. Defining a detection radius R_{det} is important to allow an aircraft enough time and distance to maneuver away from the obstacles without colliding, which is determined by the vehicle onboard sensor.

The UAV detects a static obstacle during the flight, where $\overrightarrow{PO}$ is the position vector of the UAV and the obstacle, and Δ is the angle of the speed vector $\vec{V}_n$ and $\overrightarrow{PO}$, as shown in Figure 8. When the distance between the UAV and the obstacle is less than or equal to the detection radius

$$\left|\overrightarrow{PO}\right| \leq R_{\text{det}} + R_{\text{obs}}. \tag{9}$$

If the UAV is considered to be within the range of the

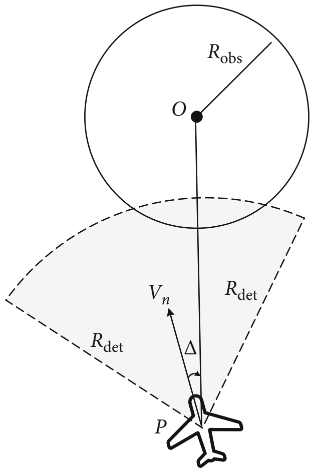


Figure 8: Obstacle detection and avoidance.

obstacle, then further checks should be made, and the judgment condition that the UAV needs to enter the avoidance state is as follows:

$$\varphi(t, V_n, \Delta) = \begin{cases} 1, & \left|\overrightarrow{\mathrm{PO}}\right| \sin \Delta < R_{\mathrm{obs}}, \\ 0, & \left|\overrightarrow{\mathrm{PO}}\right| \sin \Delta \geq R_{\mathrm{obs}}, \end{cases} \tag{10}$$

where R_{obs} is the radius of the corresponding obstacle model at the current altitude of the UAV. When equation (10) is equal to 1, the UAV should avoid the obstacle immediately.

However, equations (9) and (10) cannot be applied to dynamic obstacles. For dynamic obstacles, we need to be able to predict whether the aircraft will collide after flying for a certain time Δt at the current speed after the obstacles are detected. The rule for determining whether dynamic obstacles need to be avoided is as follows:

$$\begin{cases} \left|\overrightarrow{\mathrm{PO}}_t\right| = R_{\mathrm{det}} + R_{\mathrm{obs}}, & \text{at time } t, \\ \left|\overrightarrow{\mathrm{PO}}_{t+\Delta t}\right| < R_{\mathrm{obs}}, & \exists \Delta t > 0. \end{cases} \tag{11}$$

To some extent, the performance of the onboard sensor determines the obstacle avoidance capability of the vehicle, which is also the prerequisite and basis for obstacle avoidance. Therefore, the UAV is equipped with a suitable sensing detection system, one that is compatible with the performance of the UAV, the specific flying environment, and the characteristics of the obstacle.

3.3. Avoidance Direction Selection. The guidance system generates the circular arc avoidance trajectory in real time after detecting the obstacle threat, and it will reduce the avoidance route by choosing a reasonable avoidance direction. If the aircraft's velocity vector points to the right of the obstacle center, then it will fly along the circular arc template trajectory in a counterclockwise direction. Otherwise, it flies clockwise. The direction angle Δ can be obtained as follows:

$$\Delta = \psi_{\mathrm{obs}} - \chi, \quad \in [-\pi, \pi], \tag{12}$$

where ψ_{obs} is the angle from the connection of the UAV and obstacle center to the geography north, and χ is the course angle of the aircraft. If $\Delta > 0$, the UAV tracks a circular arc avoidance trajectory in a clockwise direction; otherwise, it tracks counterclockwise.

However, in the case of situations with complex or large obstacles which cannot be captured completely at the first time, we can use multiple standard convex bodies to model the complex obstacle. The circular arc avoidance trajectories are composed of different convex bodies; at the same time, the judging of the direction is always in process during the obstacle avoidance flights. As shown in Figure 9, when the UAV detects the obstacle again, the avoidance direction is affected by the previous obstacle. So the above rules do not apply to complex obstacle avoidance. In this case, we only need to judge the first obstacle avoidance direction as the direction in all complex obstacle avoidance processes.

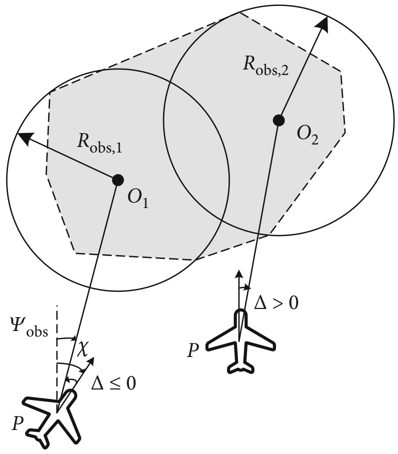


Figure 9: Avoidance direction selection.

3.4. Avoidance Success Criteria Determination. Avoidance success can be defined as keeping the UAV always outside the collision threat during the avoidance flight. The UAV follows the circular arc template trajectory until it is able to continue towards the original path in a straight line and without passing through the obstacle. We can compute the angle Λ (see in Figure 6) to determine whether avoidance is successful, which is the angle between position vector $\overrightarrow{\mathrm{PO}}$ and the vector of the current position of the UAV to the next waypoint. If $\Lambda > 90°$, the UAV avoids the obstacle successfully; otherwise, the UAV continues to track the template avoidance trajectory.

Similarly, a complex obstacle consists of n convex bodies. If $i < n$, the criteria requires that the angle $\Lambda > 90°$ or the distance between the UAV and the next obstacle is less than or equal to the detection radius. If $i = n$, the criteria only requires that $\Lambda > 90°$.

4. Simulation Results and Analysis

4.1. Simulation Assumptions. In order to verify the effectiveness of the proposed three-dimensional autonomous obstacle avoidance algorithm, the novel algorithm is performed in Matlab/Simulink. Here are some of the assumptions:

(1) In this section, we use a high-order, nonlinear numerical model of the Cessna 172 fixed-wing aircraft to test the effectiveness of the algorithm, which is constructed in Simulink. The overall aerodynamic parameters for the aircraft model are described in [29], and some flight performance constraints of the aircraft are defined in this paper. The maximum roll angle is $\phi_{\mathrm{max}} = \pm 30°$, and the maximum pitch angle is $\theta_{\mathrm{max}} = \pm 20°$. Suppose the initial position of the UAV in the inertial coordinate systems is (0, 0, -200)m and the cruising speed is $V_n = 50$ m/s. Unless otherwise specified, the predefined state of the UAV in all simulation tests is the same

```
Input: waypoint path $W = \{wp_1, \cdots, wp_n\}$, UAV position $P = (p_n, p_e, p_d)$, obstacles information, center $O$, radius $R_{\text{obs}}$, and speed $v$.
if state = 1 then
    UAV tracks the waypoints path
end if
for $i \leftarrow 1$ to $n$ do
    $O \leftarrow O + vt$
    $|\overrightarrow{\mathrm{PO}}| \leftarrow \sqrt{(p_n - o_n)^2 + (p_e - o_e)^2}$
    $\psi_{\text{obs}} \leftarrow \arctan((o_e - p_e)/(o_n - p_n))$
    $\Delta \leftarrow \psi_{\text{obs}} - \chi$
    if $|\overrightarrow{\mathrm{PO}}| \leq R_{\text{det}}$ & $R_{\text{det}} \sin \Delta < R_{\text{obs}}$ then
        state = 2
        if $\Delta > 0$ then
            UAV tracks the circular path in clockwise
        else
            UAV tracks the circular path in clockwise
        end if
    $\Lambda \leftarrow \arctan((wp_{\text{next},e} - p_e)/(wp_{\text{next},n} - p_n)) - \psi_{\text{obs}}$
        if $\Lambda > pi/2$ or $|\overrightarrow{\mathrm{PO}_{i+1}}| \leq R_{\text{det}}$
            state $\leftarrow$ 1
            $i \leftarrow i + 1$
        end if
    end if
end for
return UAV maneuver operations
```

Algorithm 1: The CTGA algorithm.

(2) This article mainly proposes an autonomous obstacle avoidance algorithm, and it does not require specific onboard sensors. Therefore, any sensor that can be installed on a UAV, which can acquire accurate information such as the position and speed data of obstacles is applicable. Meanwhile, it does not consider uncertain factors that affect the measurement accuracy of sensors or change the flight environment, such as wind speed and electromagnetic interference. In the simulation tests, the parameters such as the position and velocity of the obstacles are assumed and predefined, and the process of sense and detection is ignored

(3) The flight speed of the UAV model in this paper is limited to 40-80 m/s, and the overall maneuverability of the UAV is also limited, as well as its maximum attitude. Therefore, there is a certain restriction on the moving speed for the dynamic obstacles

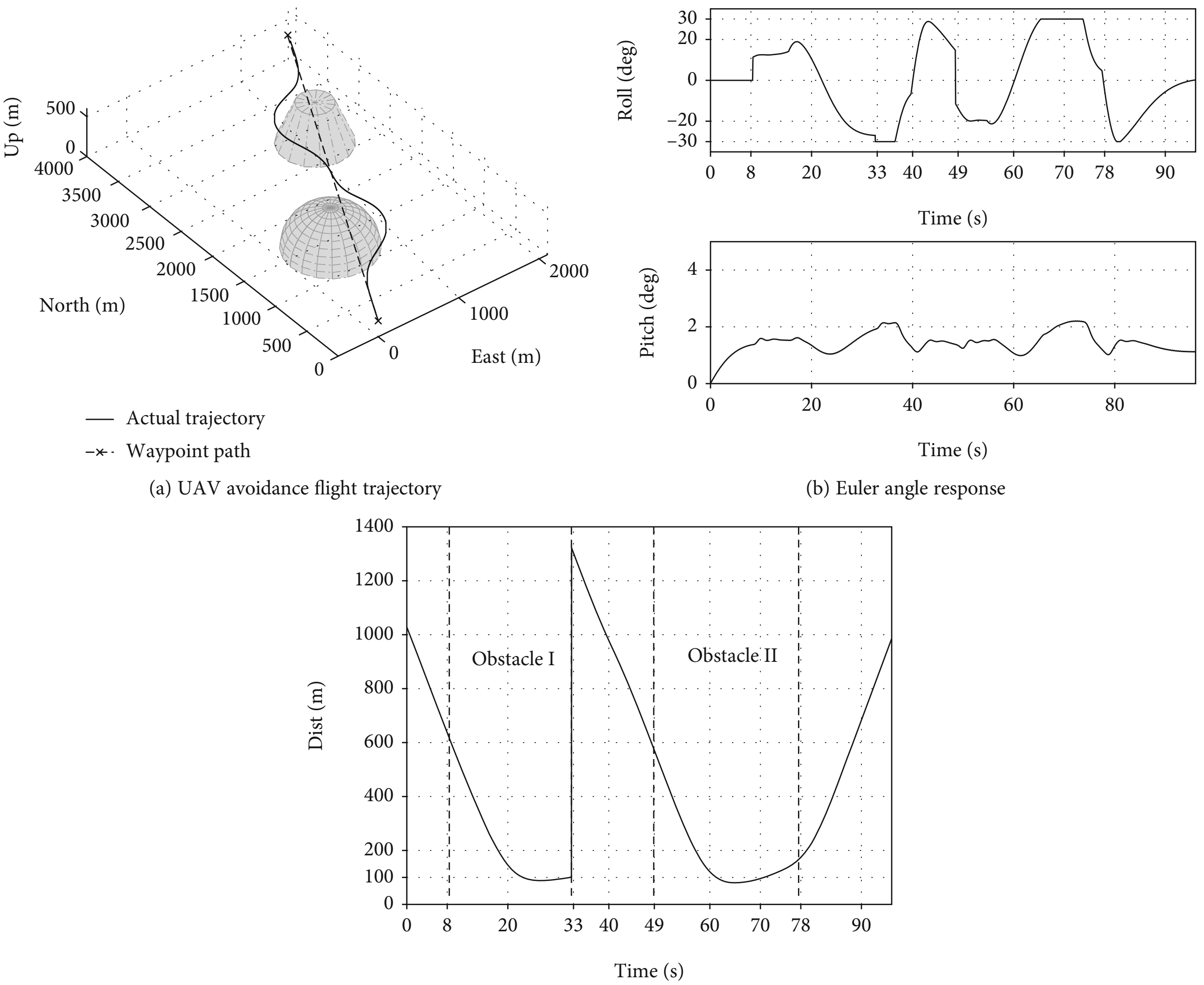
4.2. Obstacle Avoidance Simulations. The CTGA algorithm 1 is carried out in some cases when the UAV goes through potential collisions in the predefined flight path with simple, complex, and dynamic obstacles, respectively. In all simulation tests, the aircraft is operated to follow a waypoint flight path, and the coordinates of the predefined waypoint path are (0, 0, -200)m to (4000, 2000, -300)m in a north-east-Down coordinate, with the safe distance set at $R_{\text{safe}} = 100$ m.

4.2.1. Simple Obstacle Avoidance. For the first simulation, between the waypoints, the aircraft will face some simple obstacles which need to be avoided. For a simple obstacle, we only use a single convex body to model. The flight trajectory and flight status of the UAV during the entire avoidance process are shown in Figure 10.

It can be seen from Figure 10 that the UAV detects the obstacle threat twice and avoids obstacles successfully during the whole flight. First, the UAV detects obstacle I at 8 s; the CTGA algorithm generates the circular arc template avoidance trajectory real time, and determines the avoidance direction as counterclockwise according to the relationship between the flight direction and the obstacle position. Then, the UAV avoids obstacle I successfully at 33 s, and immediately returns to the predefined waypoint path with maximum roll angle and continues to complete the flight mission. Then, the UAV detects and avoids obstacle II at 49-78 s. Figure 10(c) shows that during the avoidance flight, the UAV has always been outside of the obstacles and has maintained a safe distance of about 100 m away from the surface of the obstacles. The simulation results fully confirm that the algorithm can effectively avoid simple obstacles.

4.2.2. Complex Obstacle Avoidance. For complex obstacles in the flight mission, as has been discussed previously, we can use multiple standard convex bodies to model complex obstacles. The flight trajectory, Euler angle response, and relative distance are depicted in Figure 11.

It can be seen from the figures that the algorithm performs a multistage procedure to generate three circular trajectories due to the obstacle size and detection system limitation, and the UAV tracks the avoidance trajectories to avoid the complex obstacle successfully. The UAV first detects obstacle I and performs avoidance maneuvers at 13 s, and after passing the first avoidance process, the UAV

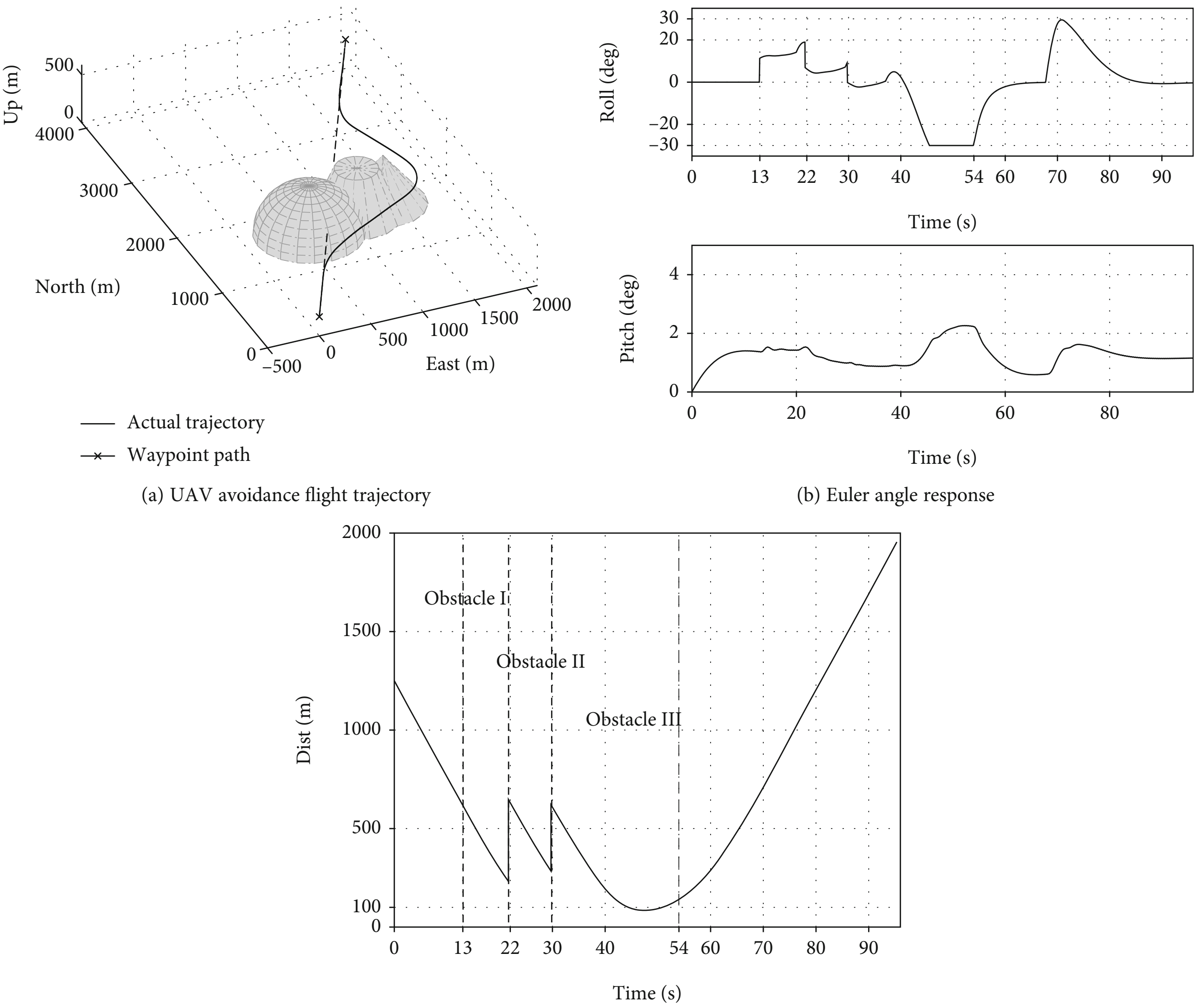


(a) UAV avoidance flight trajectory

(b) Euler angle response

(c) The relative distance between the UAV and the surface of the obstacle

Figure 10: UAV flight status of simple obstacle avoidance simulation.

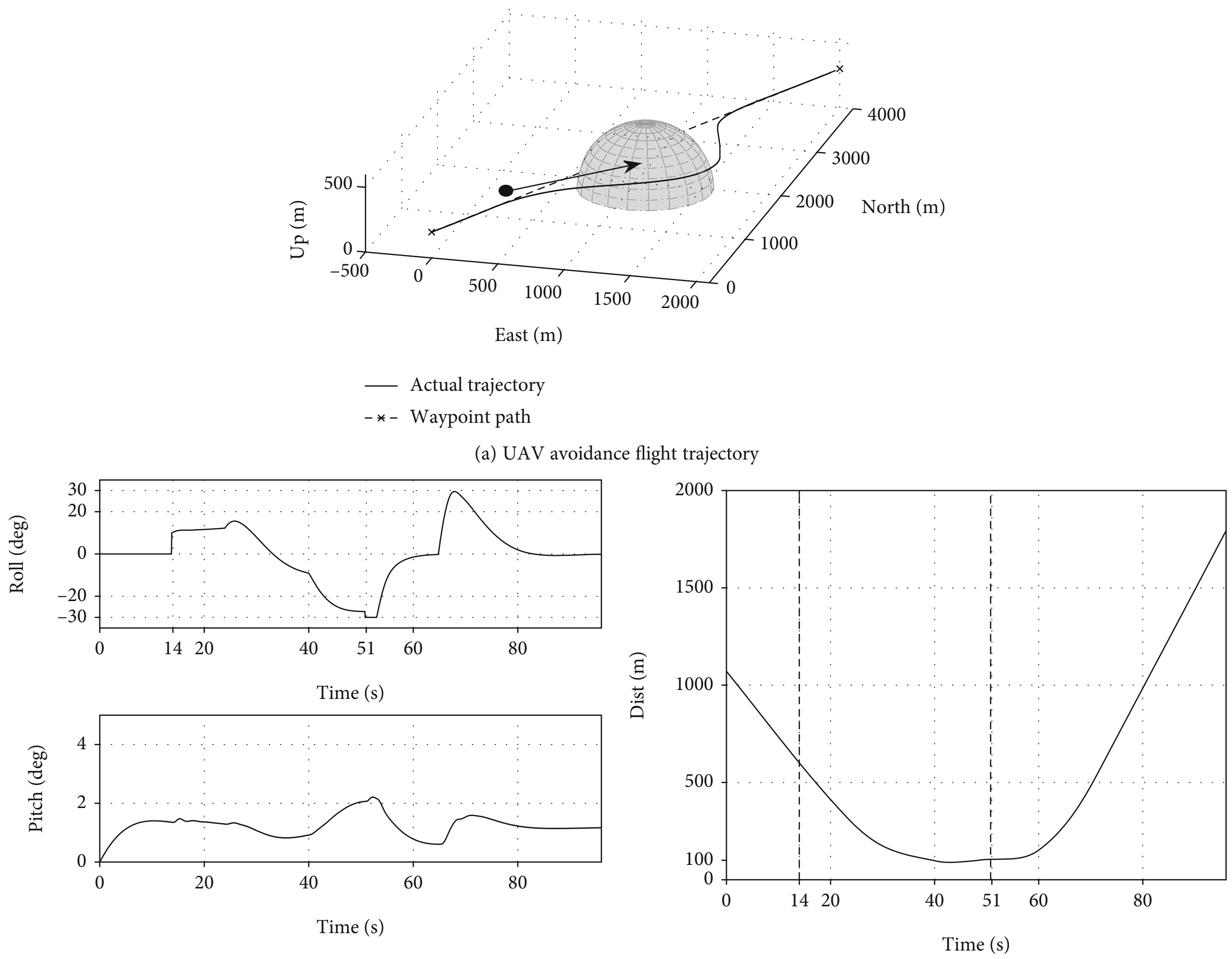


(a) UAV avoidance flight trajectory

(b) Euler angle response

(c) The relative distance between the UAV and the surface of obstacle

Figure 11: UAV flight status of complex obstacle avoidance simulation.



(a) UAV avoidance flight trajectory

(b) Euler angle response

(c) The relative distance between the UAV and the surface of the obstacle

Figure 12: UAV flight status of the dynamic obstacle avoidance simulation.

detects and avoids obstacle II at 22 s. Similarly, obstacle III is detected at 30 s; then, the UAV avoids the last obstacle and returns back to the waypoint path. As shown in Figure 11(c), during the avoidance flight, the UAV has always been outside of the obstacles and about 100 m away from the surface of obstacles, so it can be confirmed that the CTGA algorithm can also effectively avoid complex obstacles through a multiple obstacle model.

4.2.3. Dynamic Obstacle Avoidance. For the third simulation, in order to verify if the CTGA algorithm is also effective for a dynamic obstacle with a certain speed, we use a moving obstacle in the flight space, with an initial position of the obstacle center point at 1500 m and 300 m, and a moving speed of 20 m/s in the east, 10 m/s in the north, and stopping after 40 s. The real-time flight trajectory, Euler angle response, and relative distance between the UAV and the moving obstacle are shown in Figure 12.

It can be seen from the figures that the UAV detects the dynamic obstacle at 14 s, because the moving direction of the dynamic obstacle is towards the UAV. The whole avoidance process lasts 37 s, and avoidance finally ends at 51 s. The UAV has always been outside of the obstacles and about 100 m away from the surface of the obstacles, so it can be confirmed that the CTGA algorithm can also effectively avoid dynamic obstacles.

In summary, the CTGA algorithm provides an efficient obstacle avoidance method for UAVs. It is not only simple and convenient, but it can also avoid static and dynamic obstacles.

5. Conclusion

An innovative autonomous obstacle avoidance algorithm for UAVs in 3D space based on a circular arc trajectory has been proposed and discussed.

(1) The method of the obstacle model is simple according to obstacle data by an onboard detection system, which uses one or more standard convex bodies to enclose the detected parts of an obstacle.

Consequently, the circular arc avoidance trajectory is designed through the obstacle model. It not only transforms an obstacle avoidance problem into a path following problem, but it also simplifies the design of an obstacle avoidance algorithm

(2) The proposed CTGA algorithm can avoid static and dynamic obstacles by simply referring to the geometric relationship between a UAV and an obstacle, even when the obstacle geometry is not available and cannot be obtained directly; a multistaged circular trajectory correction procedure can be employed. This algorithm not only satisfies the maneuverability constraints of a UAV, but it also requires less calculation, which has led to a good applicability in practical engineering

The presented work is focused on the design of an obstacle avoidance algorithm for a UAV, and it has been verified in a simulation environment. Our future work will do more extensive and complex simulations to analyze the characteristics and performance of the proposed algorithm and apply the algorithm to actual flight tests.

Data Availability

The data used to support the findings of this study are included within the article.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by the Fundamental Research Funds for the Central Universities (grant number 56XAA17075).

References

[1] S. K. Chaturvedi, R. Sekhar, S. Banerjee, and H. Kamal, “Comparative review study of military and civilian unmanned aerial vehicles (UAVs),” *INCAS Bulletin*, vol. 11, no. 3, pp. 183–198, 2019.

[2] Y. L. Kang and Y. Xi, “Development situation, trend and countermeasure of consumer-level UAV market in China,” *ITM Web of Conferences*, vol. 11, 2017.

[3] Y. Lan, L. Wang, and Y. Zhang, “Application and prospect on obstacle avoidance technology for agricultural UAV,” *Transactions of the Chinese Society of Agricultural Engineering*, vol. 34, no. 9, pp. 104–113, 2018.

[4] L. Yang, J. Qi, J. Xiao, and X. Yong, “A literature review of UAV 3D path planning,” in *Proceeding of the 11th World Congress on Intelligent Control and Automation*, pp. 2376–2381, Shenyang, China, 2014.

[5] D. Agarwal and P. S. Bharti, “A review on comparative analysis of path planning and collision avoidance algorithms,” *International Journal of Mechanical and Mechatronics Engineering*, vol. 12, no. 6, pp. 608–624, 2018.

[6] H. Tong, W. W. chao, H. C. qiang, and X. Y. bo, “Path planning of UAV based on Voronoi diagram and DPSO,” *Procedia Engineering*, vol. 29, pp. 4198–4203, 2012.

[7] 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*, pp. 144–147, Hangzhou, China, 2013.

[8] Y. B. Chen, G. C. Luo, Y. S. Mei, J. Q. Yu, and X. L. Su, “UAV path planning using artificial potential field method updated by optimal control theory,” *International Journal of Systems Science*, vol. 47, no. 6, pp. 1407–1420, 2016.

[9] A. Kaur and M. S. Prasad, “Path planning of multiple unmanned aerial vehicles based on RRT algorithm,” in *Advances in Interdisciplinary Engineering*, pp. 725–732, Springer, Singapore, 2019.

[10] C. Zammit and E. V. Kampen, “Comparison between A* and RRT algorithms for UAV path planning,” in *2018 AIAA Guidance, Navigation, and Control Conference*, pp. 1–23, Kissimmee, Florida, 2018.

[11] S. A. Gautam and N. Verma, “Path planning for unmanned aerial vehicle based on genetic algorithm & artificial neural network in 3D,” in *2014 International Conference on Data Mining and Intelligent Computing (ICDMIC)*, pp. 1–5, Delhi, India, 2014.

[12] J. Chen, F. Ye, and T. Jiang, “Path planning under obstacle avoidance constraints based on ant colony optimization algorithm,” in *2017 IEEE 17th International Conference on Communication Technology (ICCT)*, pp. 1434–1438, Chengdu, China, 2017.

[13] K. Y. Kok and P. Rajendran, “Differential-evolution control parameter optimization for unmanned aerial vehicle path planning,” *PLoS ONE*, vol. 11, no. 3, article e0150558, 2016.

[14] J. . Silva Arantes, M. . Silva Arantes, C. F. Motta Toledo, O. T. Júnior, and B. C. Williams, “Heuristic and genetic algorithm approaches for UAV path planning under critical situation,” *International Journal on Artificial Intelligence Tools*, vol. 26, no. 1, 2017.

[15] S. Perez-Carabaza, E. Besada-Portas, J. A. Lopez-Orozco, and J. M. de la Cruz, “Ant colony optimization for multi-UAV minimum time search in uncertain domains,” *Applied Soft Computing*, vol. 62, pp. 789–806, 2018.

[16] X. Zhang and H. Duan, “An improved constrained differential evolution algorithm for unmanned aerial vehicle global route planning,” *Applied Soft Computing*, vol. 26, pp. 270–284, 2015.

[17] M. Radmanesh, M. Kumar, P. H. Guentert, and M. Sarim, “Overview of path-planning and obstacle avoidance algorithms for UAVs: a comparative study,” *Unmanned Systems*, vol. 6, no. 2, pp. 95–118, 2018.

[18] J. Zheng, B. Liu, Z. Meng, and Y. Zhou, “Integrated real time obstacle avoidance algorithm based on fuzzy logic and L1 control algorithm for unmanned helicopter,” in *2018 Chinese Control and Decision Conference (CCDC)*, pp. 1865–1870, Shenyang, China, 2018.

[19] R. A. Sasongko, S. S. Rawikara, and H. J. Tampubolon, “UAV obstacle avoidance algorithm based on ellipsoid geometry,” *Journal of Intelligent & Robotic Systems*, vol. 88, no. 2-4, pp. 567–581, 2017.

[20] Z. Lin, L. Castano, E. Mortimer, and H. Xu, “Fast 3D collision avoidance algorithm for fixed wing UAS,” *Journal of Intelligent & Robotic Systems*, vol. 97, no. 3-4, pp. 577–604, 2020.

[21] A. al-Kaff, F. García, D. Martín, A. de la Escalera, and J. Armingol, "Obstacle detection and avoidance system based on monocular camera and size expansion algorithm for UAVs," *Sensors*, vol. 17, no. 5, pp. 1061–1082, 2017.

[22] C. Essen and D. Giannakopoulou, "Analyzing the next generation airborne collision avoidance system," in *International Conference on Tools and Algorithms for the Construction and Analysis of Systems*, pp. 620–635, Springer, Berlin, Heidelberg, 2014.

[23] M. Huerta, "Integration of civil unmanned aircraft systems in the national airspace system roadmap," *Federal Aviation Administration*, vol. 19, 2013.

[24] S. Hrabar, "An evaluation of stereo and laser-based range sensing for rotorcraft unmanned aerial vehicle obstacle avoidance," *Journal of Field Robotics*, vol. 29, no. 2, pp. 215–239, 2012.

[25] M. Joachim, V. R. Alberto, and W. Iris, "Safe & sound: a robust collision avoidance layer for aerial robots based on acoustic sensors," in *2014 IEEE/ION Position, Location and Navigation Symposium - PLANS 2014*, pp. 1197–1202, Monterey, CA, USA, 2014.

[26] C. Luo, S. I. McClean, G. Parr, L. Teacy, and R. De Nardi, "UAV position estimation and collision avoidance using the extended Kalman filter," *IEEE Transactions on Vehicular Technology*, vol. 62, no. 6, pp. 2749–2762, 2013.

[27] J. Haugen and L. Imsland, "Monitoring moving objects using aerial mobile sensors," *IEEE Transactions on Control Systems Technology*, vol. 24, no. 2, pp. 475–486, 2015.

[28] H. L. Wang, W. T. Lyu, P. Yao, X. Liang, and C. Liu, "Three-dimensional path planning for unmanned aerial vehicle based on interfered fluid dynamical system," *Chinese journal of aeronautics*, vol. 28, no. 1, pp. 229–239, 2015.

[29] X. L. Yang, M. L. Alvarez, and T. Bruggemann, "A 3D collision avoidance strategy for UAVs in a non-cooperative environment," *Journal of Intelligent & Robotic Systems*, vol. 70, no. 1-4, pp. 315–327, 2013.