A Fixed Wing UAV Path Planning Algorithm Based On Genetic Algorithm and Dubins Curve Theory

Peng Yan¹, Zhuo Yan², Hongxing Zheng¹ and Jifeng Guo^{1,a}

¹*School of Astronautics, Harbin Institute of Technology, Harbin 150001, China*

²*Aerospace Propulsion Test Technology Institute, Xi'an 710000, China*

Abstract. Path planning is an essential problem in the autonomy research of UAVs. This paper proposes a new path planning algorithm for fixed wing UAVs based on genetic algorithm (GA) and Dubins curve theory. Path planning is treated as a global optimization problem under certain constraints, including the velocity vectors of initial and goal points and the minimum turning radius of UAVs. Dubins curve theory is utilized to satisfy the velocity vector constraints. GA is utilized to generate the shortest threats avoidance path in a 2D environment. A new encoding scheme is proposed, taking into account initial circles, goal circles, threats circles and the positional relationship between the path and these circles. The 2D Dubins path is converted to Dubins airplane path by adding a flight-path angle to it. The algorithm was tested in a complex flight environment and the planned path was tracked by a 6DOF Simulink model of a fixed wing UAV. Results show that the algorithm can generate shortest threats avoidance path in a complicated 3D environment and meanwhile satisfy constraints mentioned above.

1 Introduction

Path planning plays an important role in enhancing the ability of autonomous flight of unmanned aerial vehicles (UAVs). In general, there are two different approaches to construct this path: 1) on-line path planning and 2) off-line path planning [1]. On-line path planning is real-time local planning and it requires real-time access to environmental information. Off-line path planning is global planning based on the known environment information, and its goal is to generate an optimal flyable path from an initial point to a goal point under certain constraints. This paper gives an algorithm to solve the global path planning problem of fixed wing UAVs.

There already exist different kinds of path planning algorithms. The most widely used algorithms are graph searching algorithms such as A* and Dijkstra, which are applied to grids and visibility graph [2]. In addition, sampling based algorithms like the Probabilistic Roadmap Method (PRM) and the Rapidly-exploring Random Tree (RRT) are commonly used in path planning problems [3]. The Potential Fields method, called also Artificial Potential Fields, is usually used with other algorithms due to the good results [2, 4]. In [5], a two-level path planning algorithm was proposed, which deals map uncertainty and utilizes Voronoi diagrams and A* to search the global path and Potential Fields to plan local path. Besides, heuristic-based algorithms are widely used in path planning problems due

^a Corresponding author : guojifeng@hit.edu.cn

to their high robustness compared other existing search methods under a relatively high number of constraints [4]. In [6], Li et al. proposed a Genetic-PSO (Particle Swarm Optimization) algorithm to optimize the flight paths for a group of multiple agricultural UAVs. In [7], Liu et al. presented a 3D path planning algorithm based on adaptive sensitivity decision operator combined with PSO technique.

Compared with other heuristic-based algorithms, GA is commonly used in path planning problems. For example, Li and Huang et al. [8] proposed a hierarchical genetic algorithm with optimized search region (OSR-HGA) to solve detours. Lin et al. [9] proposed a hybrid solution for UAV multi-target path planning, which integrates GA with DP (Dynamic Programming) for UAV to guarantee the safety requirement and feasibility for path planning. Sonmez et al. [10] used GA to find the optimal flyable path for the UAVs in a 3D environment.

Each aforementioned method has its own strength over others in certain aspects. However, for fixed wing UAVs, the motion constraints are not taken into account, which means they usually cannot be used in practical situations. The first constraint is the maximum curvature of the UAV under certain speed, and this limits the minimum turning radius of the UAV. The second constraint is the velocity vectors of initial and goal points, and this determines what kind of direction the UAV leaves the initial point and reaches the goal point. To meet these constraints, the Dubins curve theory [11] is utilized.

Since Dubins found that the paths of minimum length and bounded curvature can be easily constructed, the results have been widely used in path planning problems [12-14]. In [15], Shanmugavel et al. used CLC paths for path planning to solve the problem of the simultaneous arrival of UAVs and the shortest path of each UAV was found out from the available set of CLC paths. In [16], Yang et al. proved that the shortest path consists of no more than five parts, which is either an arc or a straight line under the assumption that there is only one circular obstacle. In [17, 18], Savla et al. proposed methods to solve the traveling salesperson problems (TSP) for Dubins' vehicle. In [19], Gao et al. described an algorithm that can find the shortest path derived from the theory of the Dubins curve during maneuvers and improves the performance of the UAV to follow the waypoints. The Dubins curve is also used in 3D environment. In [20], Wilburn et al. proposed a complete algorithm for producing a Dubins path in 3D environment through a set of 2 poses. In [21], Beard et al. proposed a method that extended the Dubins car model into a Dubins airplane model by adding a flight-path angle to the Dubins car path. The Dubins airplane path between two waypoints consists of helical and straight-line paths. However, these methods are used in simple environments and are difficult to obtain results in complicated environments which have more than one threat. To solve the problem of UAVs path planning in complicated environments, this paper proposes a new algorithm based on GA and Dubins curve theory.

This paper treats path planning as a global optimization problem with mission, environment and UAVs physical constraints. It has three main contributions. First, the minimum turning radius of UAVs is considered. Second, the velocity vector constraints of initial and goal points are considered. Third, the proposed algorithm can be used in a complicated 3D environment and obtain the shortest threats avoidance path. Simulation results demonstrate that the algorithm is capable of generating high quality path.

2 Problem formulation and Dubins paths

This section briefly describes the UAVs path planning problem to be solved. Path planning is defined as generating the shortest path from initial point to goal point under certain constraints, including the minimum turning radius of UAVs and the velocity vector constrains of initial and goal points.

2.1 Assumptions

Some assumptions are used to simplify the problem:

- Threats are treated as circles in a 2D environment and cylinders in a 3D environment and the radius of circles and cylinders denote the scope of threats.

- The UAV is assumed to fly at a constant speed and the influence of the wind is ignored.
- The velocity vector constraints are only acting on a horizon plane and have no influence on the vertical direction.

The flight environment in a horizon plane can be represented in Figure 1, where the red dotted circles represent threats, the small circle and the star point represent initial point and goal point, respectively, $P_s(x_s, y_s, \vec{v}_s)$ and $P_f(x_f, y_f, \vec{v}_f)$ denote the initial configuration and goal configuration in a 2D environment, respectively.

Figure 1. Flight environment in a horizontal plane.

2.2 The minimum turning radius of the UAV

The minimum turning radius of the UAV depends on the maximum roll angle of the UAV and the speed of the UAV, which can be derived from equations

$$\begin{aligned} F_{lift} \sin \gamma_{max} &= m \frac{v^2}{r_{min}} \\ F_{lift} \cos \gamma_{max} &= mg \end{aligned} \quad (1)$$

where F_{lift} represents the lift force on the UAV, γ_{max} represents the maximum roll angle of the UAV, m represents the mass of the UAV, v represents the speed of the UAV, r_{min} represents the minimum turning radius of the UAV, g represents the acceleration of gravity.

From equation (1), the minimum turning radius r_{min} can be calculated by equation

$$r_{min} = \frac{v^2}{g \tan \gamma_{max}} \quad (2)$$

2.3 Dubins paths between two configurations

The Dubins curve theory shows that the shortest path between two configurations in a 2D plane is the concatenation of circular arcs and their tangent lines. Given a certain configuration in a 2D plane, the UAV can turn right or left. Thus a certain configuration has two tangent circles as shown in Figure 2. One turns right and is marked as R , another turns left and is marked as L .

Figure 2. Tangent circles.

In Figure 2, P and $\vec{v}$ represent the position and the velocity vector of the UAV, respectively. ϕ represents the position of the UAV in the tangent circles and can be calculated by equation

$$\phi = \text{mod}(\theta \pm \frac{\pi}{2} + 2\pi, 2\pi) \quad (3)$$

where θ represents the orientation of the velocity $\vec{v}$. The choice of a positive sign is associated with a right turn and the choice of a negative sign is associated with the left turn.

The centers of the tangent circles can be calculated once given the configuration of the UAV. The equations of the circles are:

$$\begin{aligned} x_{cs} &= x_s - r_s \cos \phi_s \\ y_{cs} &= y_s - r_s \sin \phi_s \\ x_{cf} &= x_f - r_f \cos \phi_f \\ y_{cf} &= y_f - r_f \sin \phi_f \end{aligned} \quad (4)$$

where (x_{cs}, y_{cs}) and (x_{cf}, y_{cf}) are the coordinates of centers of C_s and C_f , respectively. The angles ϕ_s and ϕ_f are the positions of the UAV in the initial circle C_s and the goal circle C_f , respectively. r_s and r_f are the radius of the circles C_s and C_f , respectively.

The initial configuration and goal configuration both have two circles. There are four paths between these four circles: RSR, RSL, LSR and LSL [13, 21]. They form the Dubins paths between two configurations.

2.4 Implementation steps

To generate a flyable and shortest path in a 3D environment, our work is divided into two steps:

(1) Generate the shortest path and meanwhile satisfy the velocity vector constraints in a 2D plane. This step generates the shortest Dubins path in a complicated environment by using GA. The path in a 2D environment consists of circular and straight-line paths.

(2) Convert the 2D path to the 3D path by adding a flight-path angle to the 2D path. The path in a 3D environment consists of helical and straight-line paths.

3 GA for Dubins path planning in a 2D environment

This section presents how to use GA to generate the shortest threats avoidance Dubins path in a complicated 2D environment. The main stages in GA include selection, crossover, and mutation. In addition, to determine whether the path intersects threats, the detection operator is proposed. The path

is denoted by a chromosome. The chromosome encoding method is taken detailed consideration to make better use of GA search capability.

3.1 Chromosome encoding scheme

The chromosome encoding scheme has a great effect on the design of genetic operators, such as crossover and mutation. We propose a new binary encoding scheme which takes into account the initial circles, goal circles, threats circles, and the positional relationship between the path and these circles.

Given the initial configuration $P_s(x_s, y_s, \vec{v}_s)$, goal configuration $P_f(x_f, y_f, \vec{v}_f)$ and the positions and scopes of threats, which are denoted by $T_i (i=1, 2, \dots, n)$ (where n is the total number of threats), the shortest path is at least tangent to $k (k \in [0, n])$ threats. Thus, the problem transfers into solving the shortest path (C_s, W, C_f) . Where W is a sequence of threats that are tangent to the path, C_s and C_f are initial and goal circles, respectively. The encoding scheme contains all situations in (C_s, W, C_f) . Take the positional relationship between the circle and the path into consideration, each threat has three encoding schemes: the threat is tangent to the right of the path, the threat is tangent to the left of the path, and the threat isn't tangent to the path. Because one encoding scheme can represent that the threat is not tangent to the path, the number of encoding schemes to represent n threats is at least $2n+1$. Due to the path is tangent to the C_s and C_f certainly, the initial circle C_s and the goal circle C_f need 4 encoding schemes. Thus, to represent n threats, the number of binary digit is $\text{ceil}(\log_2(2n+1))n$. To represent C_s and C_f , the number of binary digits is 2. A chromosome has $\text{ceil}(\log_2(2n+1))n + 2$ binary digits. A chromosome is shown in Figure 3.

Figure 3. Chromosome composed by C_s , threats and C_f .

3.2 Fitness function

In order to generate the shortest path, the fitness function is designed as

$$f = \frac{d}{L+C} \quad (5)$$

where L is the length of path, C is a binary variable which punishes f when the path intersects with threats, d is the Euclidean distance from the initial point (x_s, y_s) to the goal point (x_f, y_f) which is calculated by

$$d = \sqrt{(x_s - x_f)^2 + (y_s - y_f)^2} \quad (6)$$

C is defined as

$$\begin{cases} C=0 & \text{if path doesn't intersect with the threats} \\ C=10d & \text{if path intersects with the threats} \end{cases} \quad (7)$$

3.3 Genetic operator

Conventional genetic operators include selection, crossover and mutation. On the basis of these operators, detection operator is proposed to determine whether the path intersects with the threats.

Selection: Two parents are randomly selected based on their fitness by using the roulette wheel selection method.

Crossover: One-point crossover is used. Two crossover sites are chosen randomly and then the genes between the two sites in the two parent paths are exchanged.

Mutation: The parental chromosome is chosen according to the selection method. The mutation operation is done by randomly select a gene in the parental chromosome and replace it with an opposite gene with a certain probability.

Detection: The detection operator is used to determine whether the path generated by the above three operators intersects with the threats. If the path intersects with the threats, the variable C is assigned a value of $10d$. If not, the variable C is assigned a value of zero.

3.4 Control parameters

A number of parameters affect the performance of GA implementation. These parameters include population size and the probability of crossover and mutation. Usually, the population size ranges from 50 to 200. The larger the population size, the more diverse it will be. The probability of crossover determines how often the crossover operator is applied to an individual. The mutation operator is an effective way to maintain population diversity. Commonly the value of the crossover probability ranges from 0.60 to 1.00 and the value of the mutation probability ranges from 0.05 to 0.15 [9]. The GA terminates when the maximal number of iterations is reached.

4 The Dubins airplane path in a 3D environment

In the previous section, the Dubins path in a 2D environment is proposed. In this section, the 2D path is converted to the Dubins airplane path by adding a flight-path angle to the 2D path. Figure 4 depicts a UAV flying with velocity $\vec{V}$ and flight-path angle $\mathcal{G}$ in a vertical plane.

Figure 4. A UAV flies in a vertical plane.

The physical limit on the UAV in the vertical plane is represented by

$$|\mathcal{G}| \leq \bar{\mathcal{G}} \quad (8)$$

where $\bar{\mathcal{G}}$ is the maximum flight-path angle subject to the physical limit of UAV.

The 2D Dubins path consists of circular and straight-line paths. The circular paths are turned into the helical paths when the Dubins path is converted to the Dubins airplane path.

4.1 Minimum distance Dubins airplane path

This subsection describes how to concatenate straight-line and helix paths to produce the minimum-distance path between two configurations for the UAV. Initial and goal configurations are defined as $P_s(x_s, y_s, z_s, \vec{v}_s)$ and $P_f(x_f, y_f, z_f, \vec{v}_f)$ which are formed by adding altitude information to the 2D configurations. As described in[14, 21], there are three different cases for Dubins airplane path that depend on the altitude difference between the initial and goal configuration, the length of 2D Dubins path, and the flight-path angle limit $\bar{\mathcal{G}}$. The three cases are defined to be low altitude, medium altitude, and high altitude [14].

In this study, the low altitude is only taken into consideration. The altitude difference between the initial and goal configuration is said to be the low altitude if

$$|z_f - z_s| \leq L \tan(\bar{\mathcal{G}}) \quad (9)$$

where the term on the right is the maximum altitude difference that can be obtained by flying at the flight-path angle $\pm\bar{\mathcal{G}}$ for a distance L which is the length of a 2D Dubins path.

In the low altitude case, the altitude difference between the initial and goal configurations can be achieved by flying the 2D Dubins path with a flight-path angle satisfying constraint (8). Therefore, the optimal flight-path angle can be calculated by

$$\mathcal{G}^*=\arctan\left(\frac{z_f-z_s}{L}\right)$$

(10)

The complete path planning algorithm is summarized in Algorithm 1.

Algorithm 1 Fixed Wing UAV Path Planning Algorithm

Calculate the minimum turning radius using equation (2)
Calculate initial and goal circles using equation (4)
Encode initial circles, goal circles, threats circles
Initialize the population
while *iteration < the maximum number of iteration*
 for individual **in** population
 Construct Dubins path among circles
 Detect threats
 Calculate fitness
 Perform *selection* operation
 Perform *crossover* operation
 Perform *mutation* operation
 end
 iteration = iteration + 1
end
Select the best individual from the population as the shortest Dubins path
Calculate the optimal flight-path angle using equation (10)
Convert the Dubins path to Dubins airplane path by adding the optimal flight-path angle to the Dubins path.

5 Simulation and discussions

The new proposed algorithm was tested in a complex flight environment, which has ten threats. The planned path was tracked using a 6DOF Simulink model of a fixed wing UAV. The trajectory tracking algorithm is provided by [22]. The proposed algorithm was tested in MATLAB environment and ran on a server with a 1.8 GHz CPU and 4.0 GB RAM.

5.1 Parameter settings

The mission of the UAV is to depart from an initial point and arrive at a goal point under certain constraints which include the velocity vectors of initial and goal points in a horizontal plane, the maximum roll angle and the maximum flight-path angle. The flying space is a cube with side length 10000*m*. The positions and radius of the threats are randomly generated with the radius ranging from 800*m* to 1000*m*. The physical constraints of the UAV are listed in Table 1. The parameters of GA are listed in Table 2. The configurations of the UAV are shown in Table 3.

Table 1. UAV physical constraints.

Constraint	Value	Unit
$\gamma_{\max}$	20	deg
$\bar{\mathcal{G}}$	15	deg

Table 2. GA parameters.

Parameter	Value
Population size	200
Selection rate	0.9
Mutation rate	0.1
Maximum iterations	20

Table 3. Values of the UAV configurations.

Configuration	Value	Unit
Initial coordinates	(0, 0, 3000)	<i>m</i>
Goal coordinates	(10000, 10000, 500)	<i>m</i>
Initial velocity	(30, 40)	<i>m/s</i>
Goal velocity	(0, 50)	<i>m/s</i>

5.2 Results

The algorithm runs 10 times and the best results are shown in Figure 5.

Figure 5. Simulation results.

Figure 5 (a) shows the planned path and the actual path in 2D view. The planned path is tangent to two threat circles. The minimum turning radius is 700.9*m*. The velocity vector constraints of initial and goal points are satisfied by turning left at the initial and goal points at the minimum turning radius, respectively. The path length is 14894.5 *m*.

Figure 5 (b) shows the planned path and the actual path in 3D view. The planned path is from (0, 0, 3000) to (10000, 10000, 500), the altitude difference is -2500*m*. The optimal flight-path angle is -9.53 degrees which satisfies the UAV physical constraint. The length of the Dubins airplane path is 15102.8 *m*. The Dubins airplane path successfully avoids the threats and satisfies the constraints. The

actual path and the planned path are almost identical which indicates that the planned path is a flyable path.

Figure 5 (c) shows the fitness value for each iteration. The max fitness value of the population reaches the maximum value after fifth iteration. The mean fitness value continues to increase with the iteration, and it gradually approaches the maximum fitness, which indicates that the algorithm converges smoothly.

Figure 5 (d) shows the minimum path length for each iteration. The min path length reaches the minimum value after the fifth iteration.

5.3 Discussions

As demonstrated by the results, the proposed algorithm can generate the shortest threats avoidance path in a complicated 3D environment and satisfy the constraints mentioned above. However, there are three aspects worth noting. One is the randomness of the algorithm which makes the optimal result need to run the algorithm multiple times. Another is that there are two cases may result that the final path isn't the shortest path. First, for a non-cylindrical threat, in order to generate the path, the threat is treated as a large cylinder. Second, if the path is tangent to a threat whose radius is smaller than the minimum turning radius of the UAV, the radius of the threat will be increased to the minimum turning radius of the UAV and rerun the algorithm to calculate the path until the radius of the threat which is tangent to the path is greater or equal to the minimum turning radius of the UAV. The third aspect is that we only consider the low altitude situation in a 3D environment. For medium altitude and high altitude situations, the solution process is more complex and will be studied in the future work.

6 Conclusions

In this paper, a new path planning algorithm for fixed wing UAVs is proposed which generates the shortest threats avoidance path in a complicated 3D environment and meanwhile satisfies the constraints including velocity vectors of initial and goal points and the minimum turning radius of the UAV. First, we have a simple description of the problem to be solved and make some appropriate assumptions. Second, the Dubins curve theory is utilized to satisfy the velocity vector constraints. Then, in order to generate the shortest path, GA is used and a new encoding scheme is proposed which takes into account initial circles, goal circles, threats circles, and the positional relationship between the path and these circles. At last, the 2D Dubins path is converted to the Dubins airplane path by adding a flight-path angle to it. The simulation results show our algorithm is capable of generating high quality path in a complicated 3D environment.

References

1. H. Ergezer and M. K. Leblebicioğlu, *International Conference on Unmanned Aircraft Systems*, 79-88(2013)
2. O. Souissi, R. Benatitallah, D. Duvivier, and A. H. Artiba, *International Conference on Industrial Engineering and Systems Management*, 1-8(2013)
3. D. González, J. Pérez, V. Milanés, and F. Nashashibi, *IEEE Trans. Intell. Transp. Syst.* **17**, 1-11(2015)
4. T. T. Mac, C. Copot, D. T. Tran, and R. D. Keyser, *Robot. Auton. Syst.* **86**, 13-28(2016)
5. K. Ok, S. Ansari, B. Gallagher, and W. Sica, *IEEE International Conference on Robotics and Automation*, 4596-4601(2013)
6. X. Li, Y. Zhao, J. Zhang, and Y. Dong, *IEEE International Conference on TOOLS with Artificial Intelligence*, 691-697(2017)
7. Y. Liu, X. Zhang, X. Guan, and D. Delahaye, *Aerosp. Sci. Technol.* **58**, 92-102 (2016)
8. J. Li, Y. Huang, Z. Xu, J. Wang, and M. Chen, *IEEE International Conference on Control &*

- Automation*, 1033-1038(2017).
9. C. E. Lin and Y. M. Syu, *JoAAA*, **48**, 203-220(2016)
 10. A. Sonmez, E. Kocyigit and E. Kugu, *International Conference on Unmanned Aircraft Systems*, 50-55(2015)
 11. L. E. Dubins, *Am. J. Math.* **79**, 497-516(1957)
 12. X. Sun, C. Cai and X. Shen, *J. Intell. Robot. Syst.* **79**, 3-19(2015)
 13. I. Lugo-Cardenas, G. Flores, S. Salazar, and R. Lozano, *International Conference on Unmanned Aircraft Systems*, 339 - 346(2014)
 14. H. Chitsaz and S. M. Lavalley, *IEEE Conference on Decision & Control*, 2379-2384(2008)
 15. M. Shanmugavel, A. Tsourdos, R. Zbikowski, and B. White, *AIAA Guidance, Navigation, and Control Conference and Exhibit*, (2006)
 16. D. Yang, D. Li and H. Sun, *Math. Probl. Eng.* **2013**, 1-6(2013)
 17. K. Savla, E. Frazzoli and F. Bullo, *IEEE Trans. Autom. Control.* **53**, 1378-1391(2008)
 18. K. Savla, E. Frazzoli and F. Bullo, *American Control Conference*, 786-791(2005)
 19. X. Z. Gao, Z. X. Hou, X. F. Zhu, J. T. Zhang, and X. Q. Chen, *Acat. Polytech. Hun.* **10**, 221-239(2013)
 20. J. N. Wilburn, M. G. Perhinschi and B. K. Wilburn, *AIAA Guidance, Navigation, and Control (GNC) Conference*, 5232 (2013)
 21. M. Owen, R. W. Beard and T. W. McLain, *Handbook of Unmanned Aerial Vehicles*, 1677-1701(2015)
 22. D. R. Nelson, D. B. Barber, T. W. McLain, and R. W. Beard, *IEEE Trans. Robot.* **23**, 519-529(2007)