Journal of Intelligent & Robotic Systems (2020) 99:909–931
https://doi.org/10.1007/s10846-020-01151-x

Dynamic Path Planning of the UAV Avoiding Static and Moving Obstacles

Xia Chen[1] · Miaoyan Zhao[1] · Liyuan Yin[1]

Received: 27 June 2019 / Accepted: 6 January 2020 / Published online: 23 April 2020

Abstract
This paper introduces a dynamic path planning method for the UAV that can avoid both static and moving obstacles. The condition with sudden threats can better reflect the real situation of the UAV in the real environment. First of all, the A* algorithm is adopted to generate an optimal path in a known environment in this method. Then, in the situation of static sudden threats, a series of candidate paths are generated by the principle of cubic spline second-order continuity. In order to make the static sudden threat at the center of a cluster of candidate paths, they need to be adjusted. After that, this path cluster completely surrounds the sudden threat and has symmetry about the sudden threat. When encountering a sudden threat of movement, factors such as the speed, acceleration and certain parameters of the movement obstacle or the UAV are considered, and a correlation model of the dynamic sudden threat is established. Finally, the total cost function is established to select the optimal obstacle avoidance path, and the total cost function contains four sub-cost functions, they are static security cost function, smoothness cost function, consistency cost function and dynamic security cost function. The simulation results demonstrate the effectiveness of the proposed method.

Keywords Dynamic path planning · Local path planning · Obstacle avoidance · Cubic spline · Cost function

1 Introduction

The path planning for an unmanned aerial vehicle (UAV) has been the focus of significant interest from academia, industry, agriculture and the military all along [1–4]. Path planning is especially important as the core technology of UAVs, it can be divided into two parts: global planning and local planning [5–7]. Global planning, it can also be seen as static path planning, is to plan the path when all environmental information is known; and if the environment is unknown or partially unknown, the local planning method is adopted for path planning. And local planning can also be regarded as dynamic path planning. [8–10]. With the continuous development of technology and the increasingly complex operational environment, global planning has gradually failed to meet the complex environment demand [11–13].

In recent years, many scholars and experts have studied dynamic path planning of UAVs in the complex environment. For instance, the bug based algorithms has the simplest behaviour for obstacle avoidance, such as Bug1, Bug2 [14] and Tangent Bug [15]. This kind of algorithms makes UAVs follow the obstacle edge until it is avoided. These algorithms are easy to implement and can operate in the controlled environment. However, they have a serious problem that can cause local minima and block the movement of the UAV [16]. The dynamic window approach is an online obstacle avoidance strategy [17]. In the grid map, this strategy converts the generated motion trajectory into a value function to calculate the speed, and sends it to the bottom layer. Whereas, the algorithm lacks the effective mechanisms for convergen, making it impossibly achieve the requirement to reach the target location under certain circumstances [18].

Grid-based approaches and potential field approaches also have been widely used in solving the dynamic path planning problem of the UAV. For instance, a novel hierarchical path planning method is presented in the mobile

✉ Miaoyan Zhao
137660746@qq.com

Xia Chen
xiachen1108@163.com

Liyuan Yin
2992868494@qq.com

[1] Shenyang Aerospace University, Shenyang, China

robot navigation problem [19]. This type of method has all future path information known after planner execution and before vehicle motion. However, as the complexity of the environmental situation increases, the amount of calculations of the method are also multiplied and no longer applicable. Based on the sparse A* search algorithm for path planning and the improved artificial potential field, a method of dynamic trajectory planning for UAV is proposed [20]. Nevertheless, due to the defect of the potential field method, when the potential field is balanced, the UAV is easily trapped.

Another common method for path planning is the discrete optimization approaches. In the past, a great deal of literatures in discrete optimization simulation is based on the probabilistic search techniques of simulated annealing [21] and evolutionary algorithms [22]. The technology of optimized development algorithm obtained by hybrid techniques has been widely adopted. This approach combines multiple algorithms into a single optimization strategy [23]. In state-of-the-art techniques, an enhanced discrete particle swarm optimization path planning for UAV vision-based surface inspection is presented [24]. This method can provide a safe and smooth path for autonomous vehicles or UAVs. However, it can only treat static obstacles. Moving obstacles are not considered in this method. Similarly, a path planning method is considered based on the parallel neural network structure and dynamically adjustable step size strategy. The proposed algorithm not only guarantees the safety of the UAV to evade the threat, but also improves the convergence speed of the algorithm. That method shows an effective obstacle avoidance method [28]. Although this method has many merits, the path smoothness is unsatisfactory and cannot be applied to the actual situation.

Nevertheless, in the environment with a sudden threat, there are many issues associated with path planning for the UAV in practical applications. For example: Firstly, the algorithm is required to be reasonable to avoid sudden threats. Secondly, the system should be designed to reduce the amount of computation and improve the time-consuming algorithm. Thirdly, consideration must be given to issues of the smoothness and continuity of a path, and local dynamics or global constraints [25–27].

In this paper, we mainly focused on a local planning after the global planning. Firstly, the A* algorithm is used for global planning to obtain the optimal obstacle avoidance path. Then, the problem when encountering a sudden threat can be solved by utilizing cubic spline second-order continuity combined. This method can quickly obtain a cluster of candidate paths, and help to reduce the calculation time to achieve real-time. In certain cases, starting and ending points of candidate paths can be obtained by finding the intersections between them, and this candidate path cluster has symmetric properties. When the candidate paths need to be adjusted, some informations are used as the basis. The basis for adjusting candidate paths is that the location information for sudden threats and the starting and ending points of the candidate path. In this paper, the sub-cost functions are established, i.e. the static and dynamic security cost function, the smoothness cost function and the consistency cost function. These four cost functions are weighted and combined into a novel total cost function to evaluate all candidate paths. The candidate path corresponding to the minimum value of the function is the optimal path.

In this article, the contributions can be summarized as follows: This paper utilizes the principle of the discrete optimization method to plan the path in the dynamic environment. It provides a theoretical basis for the UAV to avoid both dynamic threats and static threats. Regardless of the global planning or the local planning, the smoothness of the path is taken into account, and the obtained path is smooth and without spikes. The candidate paths are obtained by the cubic spline second-order continuity principle and can surround static threats. When discovering a dynamic sudden threat, a related model needs to be established for avoidance. Multiple sub-cost functions are set up to get the total cost function by weighting, the weights can be adjusted according to specific needs to obtain the required path. Moreover, this method can help to reduce the calculation time to achieve real-time, it also can avoid unnecessary calculations.

The work of this article is arranged as follows: Section 2 describes the generation principle of the candidate path cluster, its characteristics under certain circumstances, and the adjustment principle. The selection rule of the candidate path is indicated in Section 3. Section 4 introduces the contents of the overall structure of the framework and global path planning method. And Section 5 introduces the simulation results with different environments to verify that the method is effective; Finally, the conclusion is shown in Section 6.

2 Generation of Candidate Paths Under Sudden Threats

When the UAV is flying along the global optimal path, if it encounters a sudden threat, the UAV cannot continue to flight along the original planned path, and needs to avoid local sudden threats. For the convenience of calculation, both the sudden threat and the known obstacles in this paper are set to be round. Because of the cubic spline method has the advantages of short time-consuming and high efficiency in generating candidate paths, this paper uses the cubic spline method to generate candidate paths. The corresponding cubic spline curve equation is established as follows [30]:

$$
\begin{aligned}
y &= a(x - x^T_{start})^3 + b(x - x^T_{start})^2 + c(x - x^T_{start}) \\
&\quad + y^T_{start},\ x \in (x^T_{start}, x^T_{end}) \\
c &= 0; \\
a &= \frac{c\Delta x + 2(y^T_{start} - y_{mid})}{\Delta x^3}; \\
b &= \frac{3(y_{mid} - y^T_{start}) - 2c\Delta x}{\Delta x^2}
\end{aligned} \tag{1}
$$

y^T_{start}, y^T_{end}, x^T_{start} and x^T_{end} are the starting point coordinate $O(x^T_{start}, y^T_{start})$ and the ending point coordinate $A(x^T_{end}, y^T_{end})$ in the planned candidate path, respectively. x_{mid} is the central abscissa of the sudden threat, $\Delta x = x_{mid} - x^T_{start}$. Parameters a, b and c are cubic spline parameters. Since $c = 0$, different a and b correspond to different candidate paths. There is only one uncertainty variable y_{mid} in a and b, so setting N different y_{mid} values will result in N different sets of parameters a and b, so N different candidate paths will also be generated. As shown in Fig. 1, the blue path is all candidate paths, the red circle is a sudden threat, and the purple point is the extreme point y_{mid} of the candidate path. To facilitate giving the expression of y_{mid} denotes it as:

$$y_{mid} = y^T_{start} + \omega \tag{2}$$

The range of values ω and $\Delta\omega$ are determined according to the actual situation. In Fig. 1a, in order to generate more candidate paths in a smaller range, ω takes a value range of [-2, 2], and the step size is $\Delta\omega$, i.e. ω takes a value every 0.5 , and ω takes values respectively [-2,-1.5,-1,-0.5,0,0.5,1,1.5,2], get a total of 9 candidate paths. In Fig. 1b, the value of ω is in the range of [-4, 4],the step size is still $\Delta\omega = 0.5$, and the values of ω is [-4,-3.5,-3,-2.5,-2,-1.5,-1,-0.5,0,0.5,1,1.5,2,2.5,3,3.5,4] respectively. A total of 17 candidate paths were obtained.

The symmetry of the candidate paths are generated by (1) and (2) that is analyzed as follows: From (1) and (2), the values of ω and $-\omega$ are substituted respectively:

$$
\begin{aligned}
y_1 &= \frac{2(y^T_{start} - y^T_{start} - \omega)}{\Delta x^3}(x - x^T_{start})^3 \\
&\quad + \frac{3(y^T_{start} + \omega - y^T_{start})}{\Delta x^2}(x - x^T_{start})^2 + y^T_{start}
\end{aligned} \tag{3}
$$

$$
\begin{aligned}
y_2 &= \frac{2(y^T_{start} - y^T_{start} + \omega)}{\Delta x^3}(x - x^T_{start})^3 \\
&\quad + \frac{3(y^T_{start} - \omega - y^T_{start})}{\Delta x^2}(x - x^T_{start})^2 + y^T_{start}
\end{aligned} \tag{4}
$$

Then the following formula can be obtained: $y_1 - y^T_{start} = -(y_2 - y^T_{start})$

Therefore, the candidate paths are generated that is symmetric about sudden threats. The common point coordinates of many candidate paths are further analyzed below: when $y_1 \neq y_2$, two candidate paths can be generated, namely

$$
\begin{aligned}
y_1 &= \frac{2(y^T_{start} - y^T_{start} - \omega_1)}{\Delta x^3}(x - x^T_{start})^3 \\
&\quad + \frac{3(y^T_{start} + \omega_1 - y^T_{start})}{\Delta x^2}(x - x^T_{start})^2 + y^T_{start}
\end{aligned} \tag{5}
$$

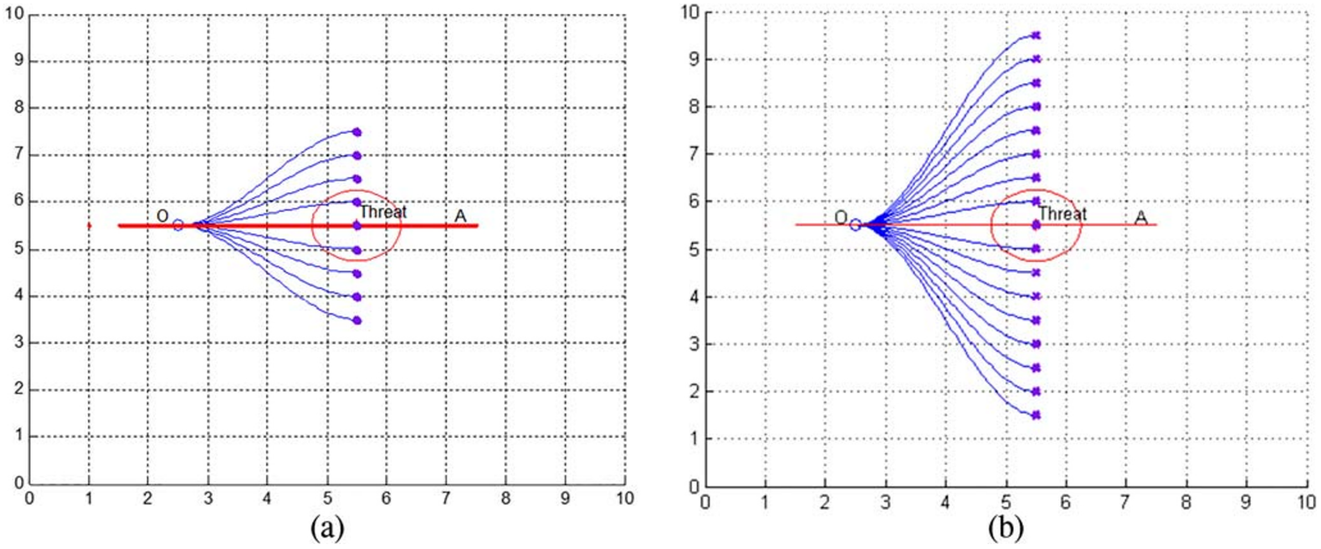


Fig. 1 Candidate path map in the range of (x^T_{start}, x_{mid})

$$y_2 = \frac{2(y^T_{start} - y^T_{start} - \omega_2)}{\Delta x^3}(x - x^T_{start})^3 + \frac{3(y^T_{start} + \omega_2 - y^T_{start})}{\Delta x^2} \times (x - x^T_{start})^2 + y^T_{start} \qquad (6)$$

Substituting $x = x^T_{start}$ into y_1 and y_2 respectively, $y_1 = y_2 = y^T_{start}$ can be obtained, i.e., both y_1 and y_2 pass the starting point $O(x^T_{start}, y^T_{start})$, after calculation, $y_1 = y_2 = y^T_{start}$ is acquired when $x = 1.5\Delta x + x^T_{start}$ is substituted into y_1 and y_2 respectively, i.e., both y_1 and y_2 pass the point coordinate $(x = 1.5\Delta x + x^T_{start}, y^T_{start})$, this point is called $A(x^T_{end}, y^T_{end})$, this point is also the end point coordinate of the candidate path, as shown in Fig. 2a and b, that is, all candidate paths pass through point A. i.e., $A(x^T_{end}, y^T_{end})$.

It should be noted that in Fig. 2a and b, the center line of the candidate path, i.e., the red line OA in Fig. 2a and b, coincides with the horizontal axis of the sudden threat. Since y_{mid} is selected in the vertical axis of the sudden threat, and the value is symmetric about the center line of the horizontal axis of the sudden threat, i.e. the candidate path is symmetric about the center line. As long as $|\ \omega\ | \geq r$, the generated candidate path must surround the sudden threat, where r is the sudden threat radius.

However, in actual situations, the center line of the sudden threat is not necessarily parallel to the horizontal axis of the Cartesian coordinate system, i.e., there is a non-zero angle α between the center line and the horizontal axis, as shown in Fig. 3a and b, $\alpha = arctan(2/3)$, ω is set in the same way as Fig. 1a and b.

It can be seen from Figs. 3 and 4 that generating the candidate path center line is not the center line of the sudden threat, so that the candidate path does not easily surround the sudden threat, as shown in Fig. 4a, even if the candidate path surrounds the sudden threat, most candidate paths are often worthless candidate paths, and the selected evasive path is often not the optimal path, as shown in Fig. 4b.

In order to ensure that the angle between the candidate path center line and the center line of the sudden threat is zero, the rotation of Figs. 3 and 4 is required, and the rotation angle is $\alpha = arctan(2/3)$. As shown in Figs. 5 and 6, the candidate paths rotate around point O, making the candidate path center line coincide with the center line of the sudden threat, and adjusted coordinate formula of the candidate paths is as follows:

$$\begin{cases} x_0(t) = (x - x^T_{start}) \times \cos\alpha - (y(t) - y^T_{start}) \times \sin\alpha + x^T_{start} \\ y_0(t) = (x - x^T_{start}) \times \sin\alpha + (y(t) - y^T_{start}) \times \cos\alpha + y^T_{start} \end{cases} \qquad (7)$$

Because the candidate paths are symmetric about the sudden threat, and both are assigned to $A(1.5\Delta x + x^T_{start}, y^T_{start})$ and $O(x^T_{start}, y^T_{start})$. After the candidate paths are rotated ,there are still symmetry about the sudden threat, and the sudden threat is at the center of the candidate paths.

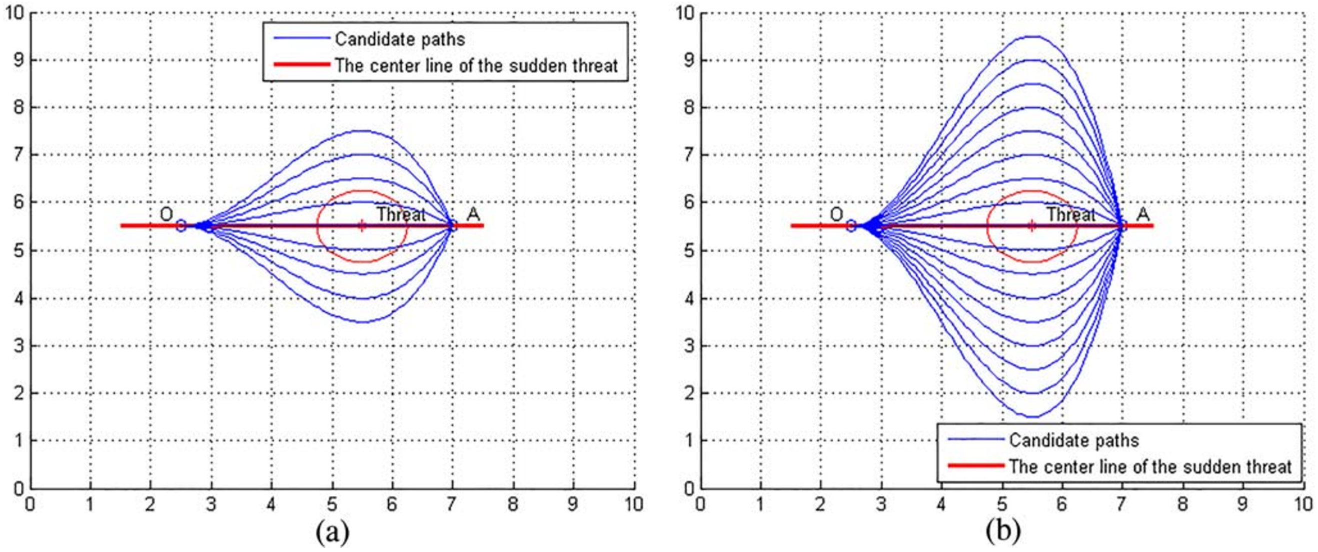


Fig. 2 Candidate path map in the range of (x^T_{start}, x^T_{end})

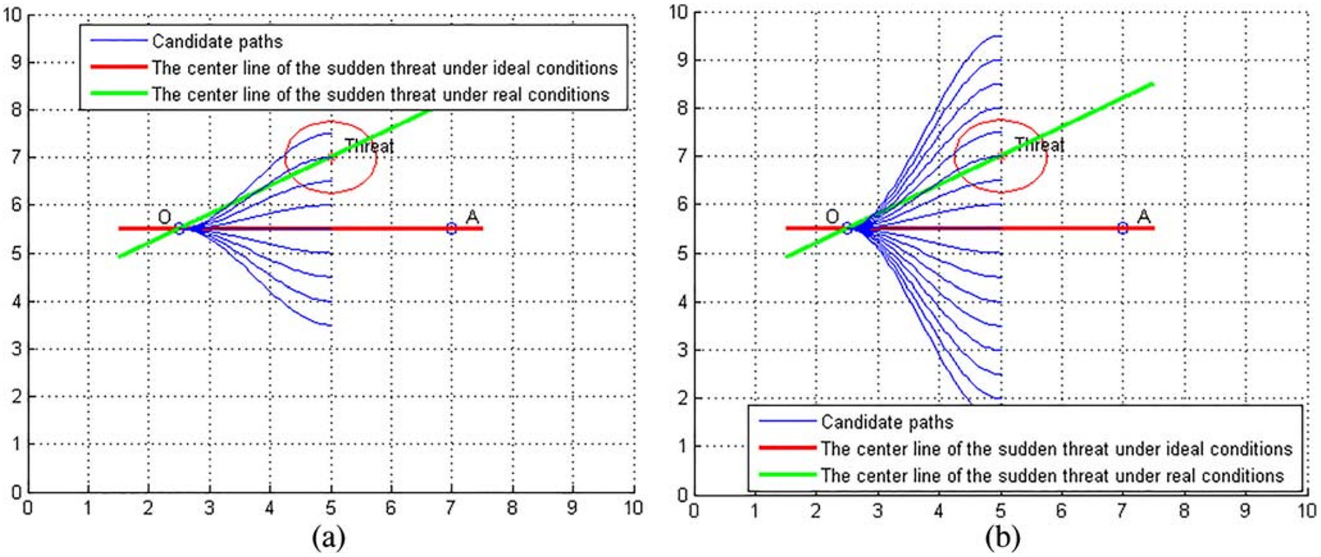


Fig. 3 Semi-closed candidate path map when $\alpha \neq 0$

3 Choice of Optimal Path Under Sudden Threat

There is a reasonable selection by setting the cost function among many candidate paths. The total cost function is set in this paper considering four factors, namely static security cost function, smoothness cost function, consistency cost function and dynamic security cost function. Considered comprehensively, the goal is to choose a better path to avoid sudden threats.

3.1 Static Security Cost Function

Safety is the primary factor in the flight of the UAV. The main factor affecting the safety of the UAV is the detected obstacles and sudden threats. By calculating the minimum distance $d(n)$ between the center of the sudden threat and the corresponding candidate paths, and comparing with the radius r of the sudden threat, the collision occurs only when the minimum distance is smaller than the radius of the sudden threat, as shown in Fig. 7.

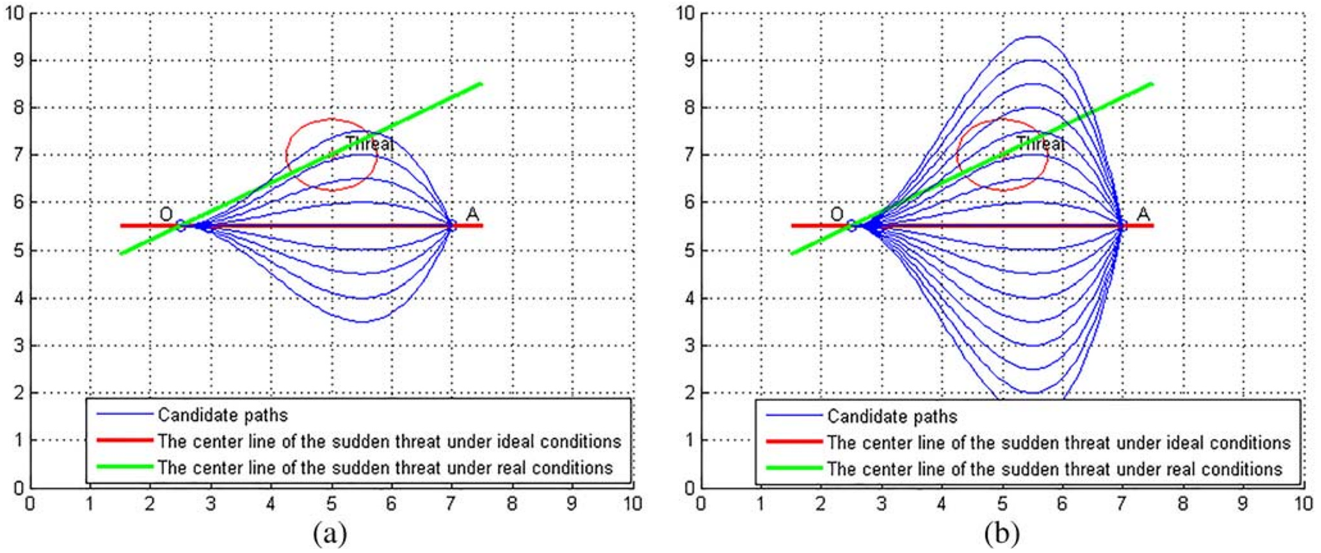


Fig. 4 Closed candidate path map when $\alpha \neq 0$

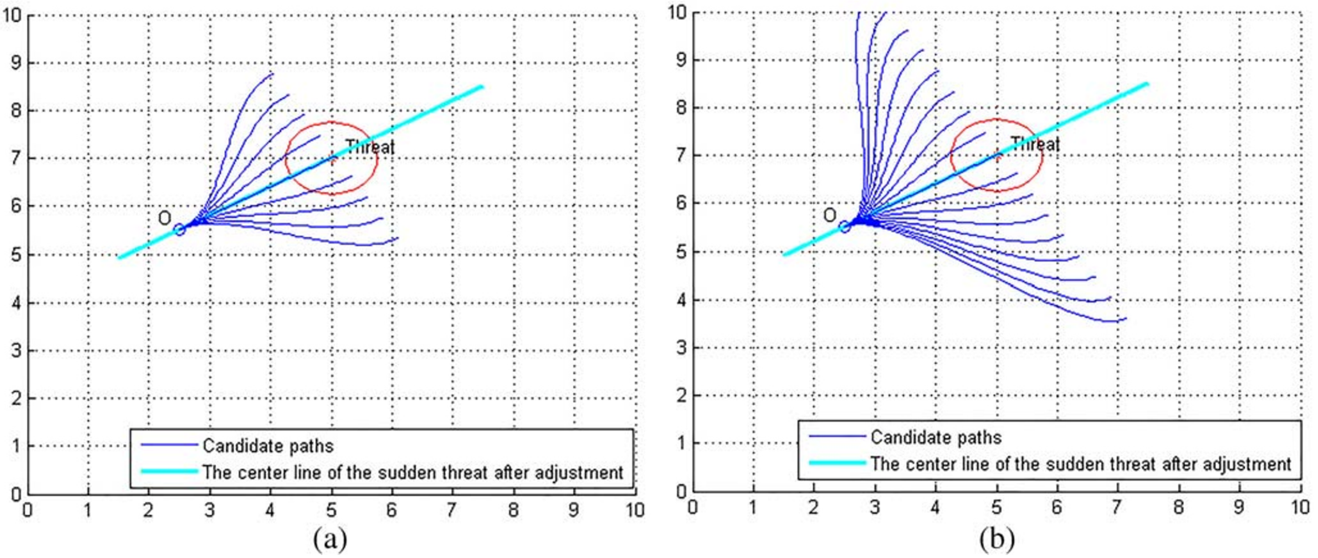


Fig. 5 Semi-closed candidate path map when $\alpha = 0$

However, the candidate paths also need to avoid the known obstacles by calculating the minimum distances $t(n)$ between the center of all known obstacles and all corresponding candidate paths, and comparing them with the radius r_0 of the known obstacles, only when the minimum distance is bigger than the radius of the known obstacles, collision will not happen, as shown in Fig. 7. In the second candidate path l_2 and the third candidate path l_3, $d(n) < r$ are established, so the second and third candidate paths collide with the sudden threat, and in the first candidate path l_1 and the fourth candidate path l_4, $d(n) > r$ are established. Therefore, the first and fourth candidate paths do not collide with the sudden threat, if only the sudden threat factor is considered, both candidate paths are feasible paths. However, the first candidate path l_1 satisfies $t(n) < r_0$, so l_1 collides with a known obstacle and does not meet the security requirement, so only the fourth candidate path l_4 is a feasible path. I.e., the candidate path is a feasible path only when both the $d(n) > r$ and $t(n) > r_0$ conditions are satisfied. n is the number of the candidate paths.

Considering the above two aspects, the result of the screening can be represented by setting the collision detection function R. If a candidate path collides with a obstacle or a sudden threat, $R = 1$, or $R = 0$. i represents the sequence number of the candidate paths, and $R[i]$ represents the R function value of the i-th candidate path. The a-d points in Fig. 8 represent candidate paths in the four cases, they are called Case 1-4 respectively,in

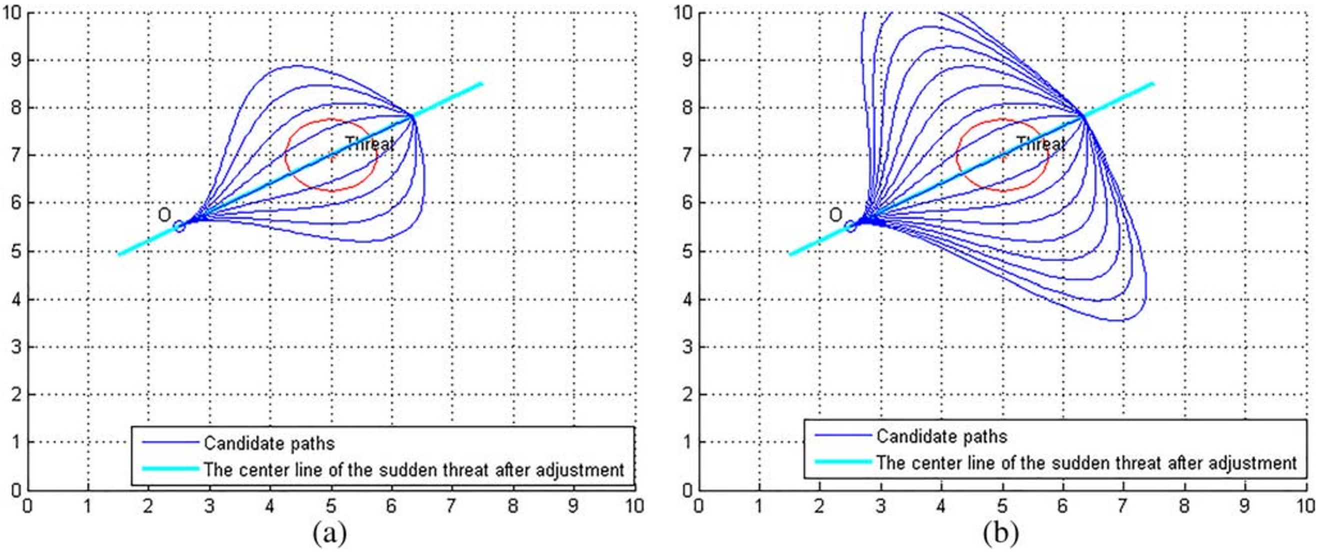


Fig. 6 Closed candidate path map when $\alpha = 0$

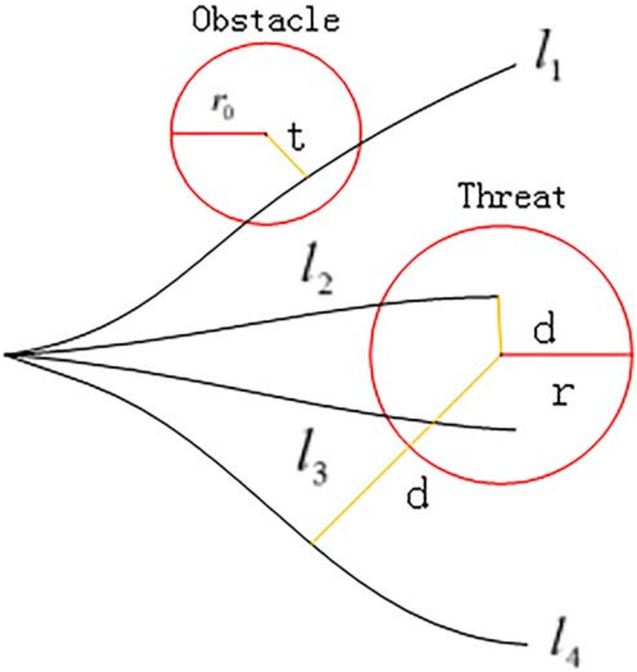


Fig. 7 Security cost function selection path diagram

this paper. And a-d in Fig. 9 represent the corresponding values of the four collision detection functions R of Fig. 8 a–d, respectively. For example, in Fig. 8a, there are 9 candidate paths from top to bottom, respectively, and the corresponding values of the collision detection function R are 0, 0, 0, 1, 1, 1, 1, 1, 1 respectively, as shown in Fig. 9a, and other cases are equally available.

In order to make the collision detection function R more closely expresses the security level. The collision detection function R is convoluted by the discrete Gaussian convolution method to achieve the purpose. It has more effective detection of the collision detection function. If only collisions are considered, several paths with the same effect will appear. For instance, the third candidate path sorted from top to bottom in Fig. 8d is similar to the fourth candidate path case. However, the third distance is closer to the obstacle and its curvature is greater, so that the possibility of hitting the obstacle is higher. Therefore, it may be an inappropriate choice to regard it as the optimal path only by the criterion of the function R. Therefore, in this paper R function is convolved by adopting the discrete

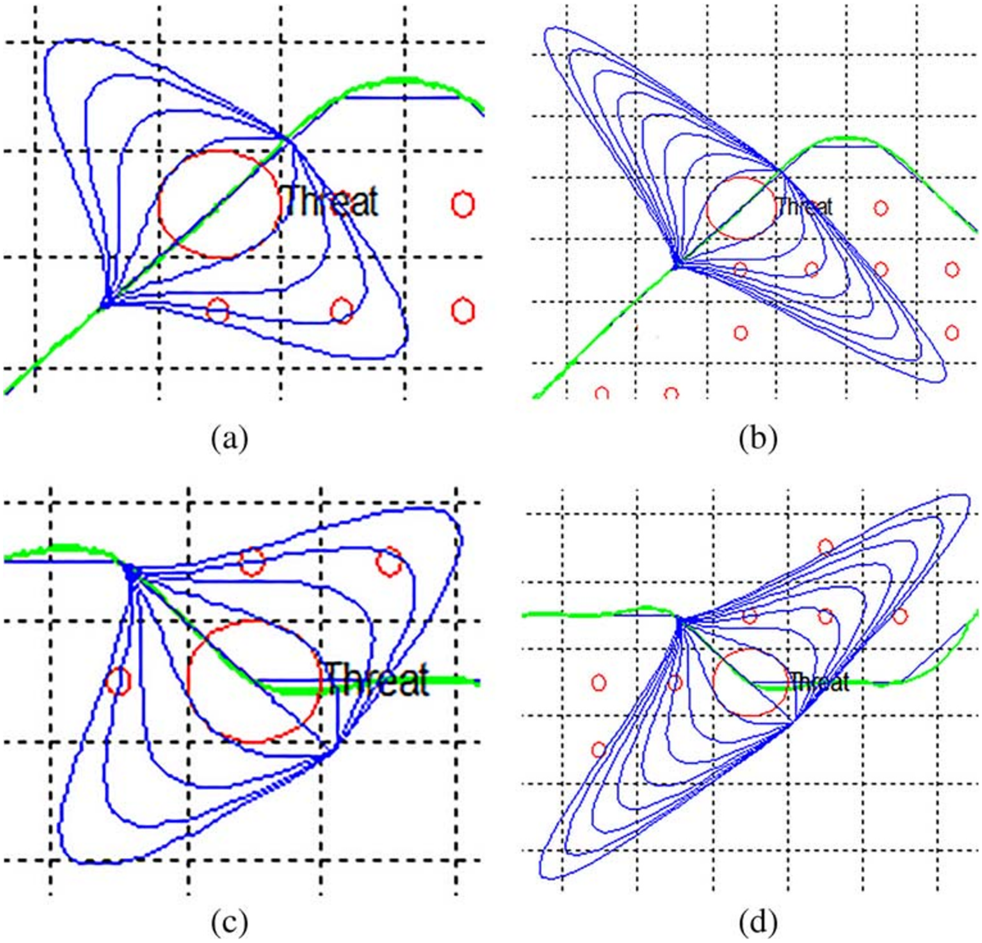
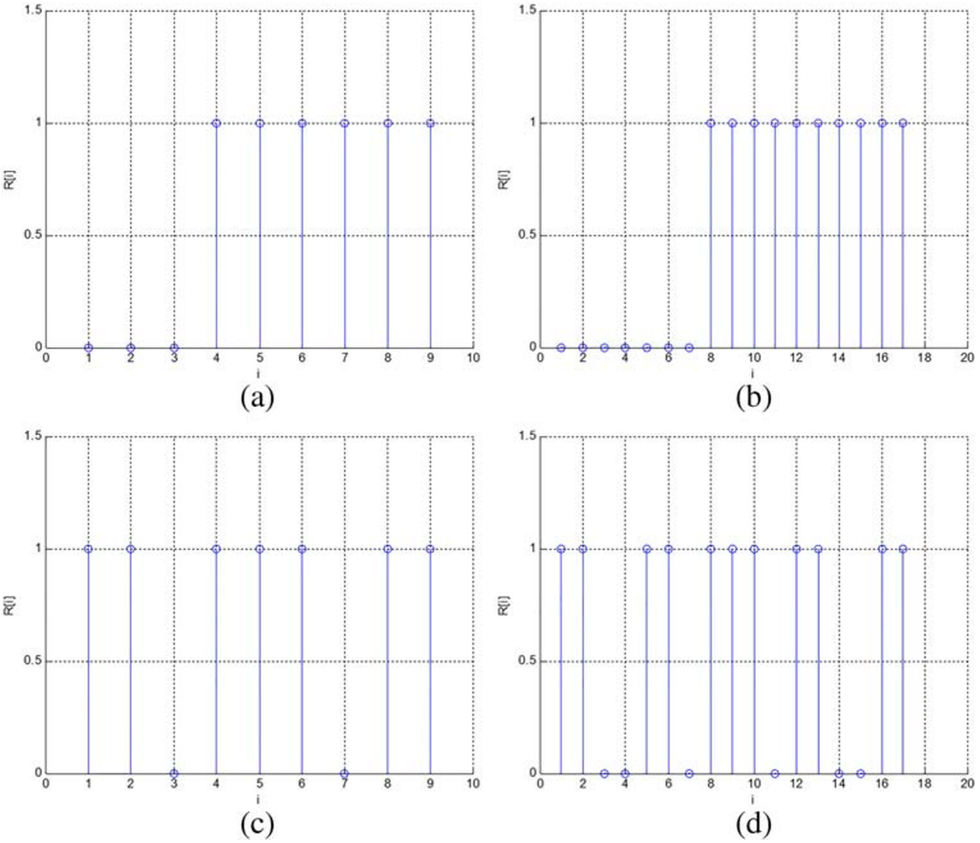


Fig. 8 Candidate path map in four cases

Fig. 9 Collision detection function R value



Gaussian convolution method, the following expression is obtained:

$$f_{safe}(i) = \sum_{k=-N}^{N} g_i[k]R[k+i] \tag{8}$$

In this formula, $f_{safe}(i)$ is the security cost function of the i-th candidate path, and the number of candidate paths is $2N + 1$. $g_i[k]$ is a discrete Gaussian function whose expression is as follows:

$$g_i[k] = \frac{1}{\sqrt{2\pi\sigma}} e^{-\frac{(k-i)^2}{2\sigma^2}} \tag{9}$$

Where σ is the standard deviation of collision risk, determining the effective range of collision detection. Figure 10 is a figure of the security cost function for the four cases of Fig. 9.

In Fig. 10, $\sigma = 0.5$, and Fig. 10a–d correspond to the four cases of Fig. 9a–d, respectively. It can be seen from Fig. 10a that the value of the second candidate path is the smallest. Nevertheless, the second candidate path is clearly not the best choice in Fig. 8a. Comparing with the collision detection function, although the effect of the security cost function is better than the former, it is inevitable that the effect will be unsatisfactory sometimes. In order to further improve the total cost function, other factors must be considered.

3.2 Smoothness Cost Function

The smoothness cost function of the UAV is further considered below. A smooth path allows the UAV to fly stably, so smoothness is also a factor that must be considered. The path's smoothness is related to its curvature directly. In this paper, the square of the curvature is used to integrate on the path as the smoothness cost function $f_{smooth}(i)$. The expression is as follows:

$$f_{smooth}(i) = \int K_i^2(x)dx \tag{10}$$

Where $K_i(x)$ is the curvature of the i-th candidate path at the x position, and the upper and lower limits of the integral are the starting and ending points of the candidate path.

The smoothness cost function represents the degree of bending of the candidate path. The higher the degree of

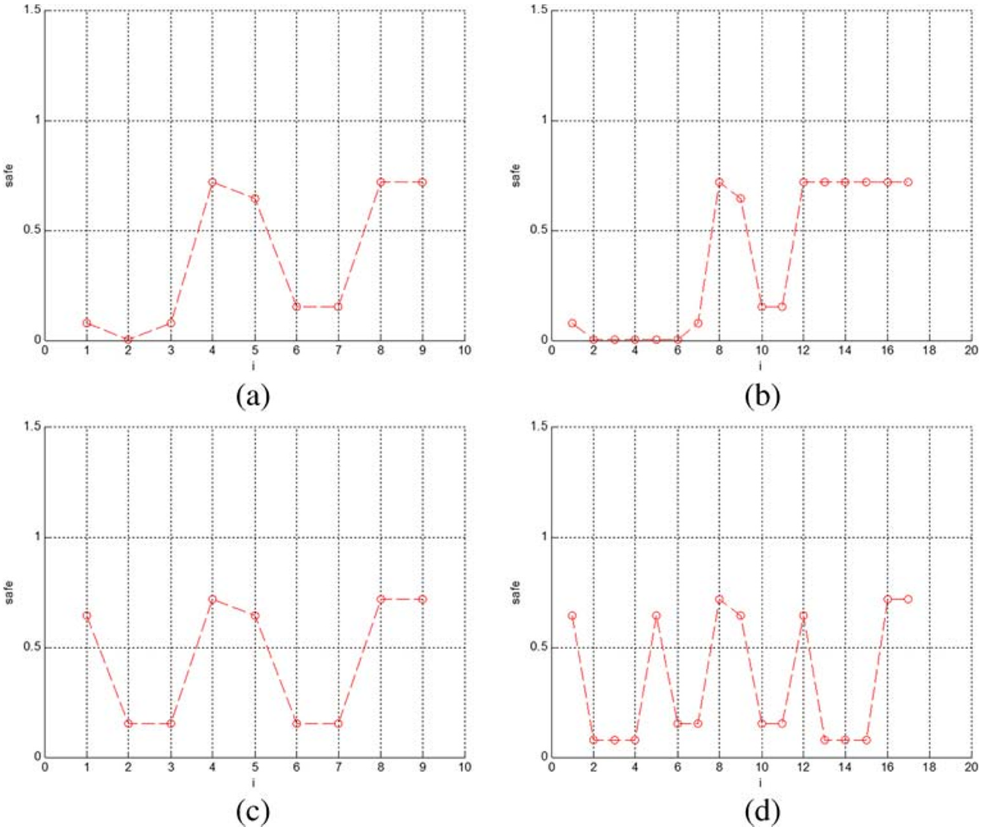


Fig. 10 Static security cost function value diagram

bending, the larger of the angle of the UAV that needs to be turned. On the contrary, the smoothness is better.

In addition, it is also necessary to consider the maximum deflection angle that the UAV can withstand, which is a factor that must be considered due to its own factors.

The constraint expression of deflection angle is as follows: $\theta_i \leq \theta_{\max},\ i = 1, 2, \cdots, N_h$. Where:

$$\theta_i = \arccos\left[\frac{(x_i - x_{i-1}, y_i - y_{i-1})}{\| x_i - x_{i-1}, y_i - y_{i-1} \|} \cdot \frac{(x_{N_h} - x_i, y_{N_h} - y_i)^T}{\| x_{N_h} - x_i, y_{N_h} - y_i \|}\right] \tag{11}$$

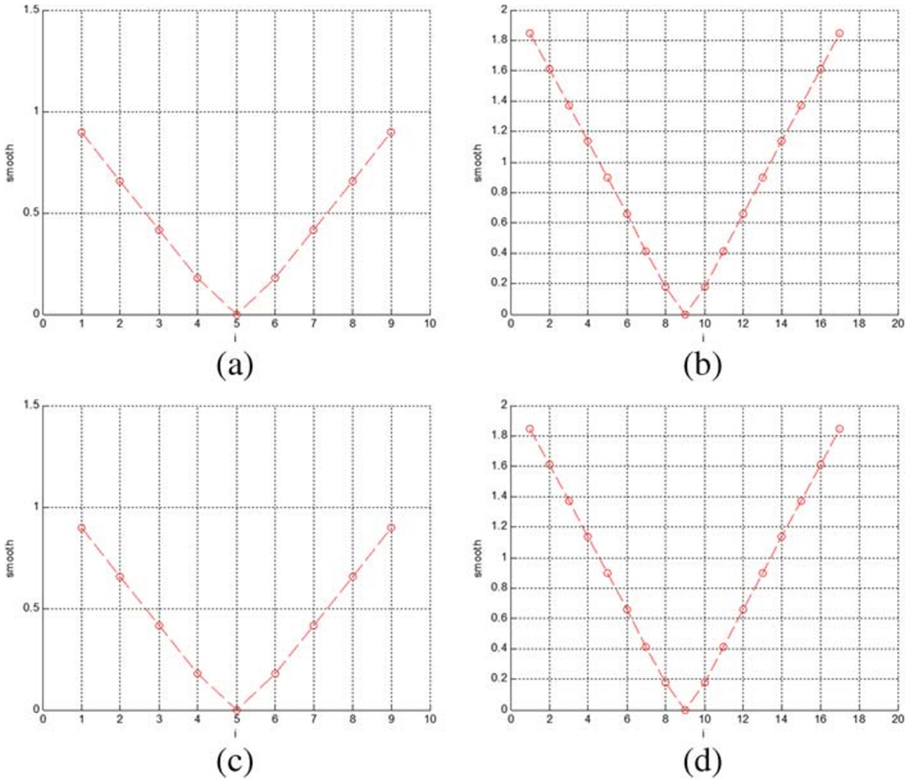
From Fig. 11, it can be seen that Fig. 11a and c correspond to the case of Fig. 8a and c, and both of the fifth candidate paths are minimum; Fig. 11b and d correspond to the cases of Fig. 8b and d, both of which are the ninth candidate path is the smallest. However, neither the fifth candidate path nor the ninth candidate path is the best choice, the reason is that the two paths do not meet the security requirements. Therefore, it is necessary to consider the cost function of both security and smoothness comprehensively.

Figure 12 is a cost function diagram that considers both security and smoothness. Figure 12a corresponds to the case of Fig. 8a, and that is compared with the second candidate path (Fig. 10a) that only considers security and the fifth (Fig. 11a) that only considers smoothness. Figure 12a considers the influence of the above two factors, and the second candidate path is still selected as the optimal path. Figure 12b corresponds to the case of Fig. 8b, it is compared with the fourth candidate path (Fig. 10b) obtained by only considering security and the ninth (Fig. 11b) only considering smoothness. The influence of the above two factors is considered by Fig. 12b. Then the sixth candidate path is the best path. In other cases, the same situations are available. According to the above comparison, it can be concluded that the path effect obtained by considering the two aspects is more ideal than considering only the safety or smoothness.

3.3 Consistency Cost Function

Consistency is a characteristic that emphasizes the connection between the candidate paths and the original path. If there is a sudden change between the candidate paths and the original path, a sharp turn will happen. Not only it will affect the smoothness but it will also cause unexpected things like security issue. The consistency cost function mainly considers two aspects: on the one hand, the Euclidean distance between the candidate paths and the location of the original obstacle avoidance path, on the

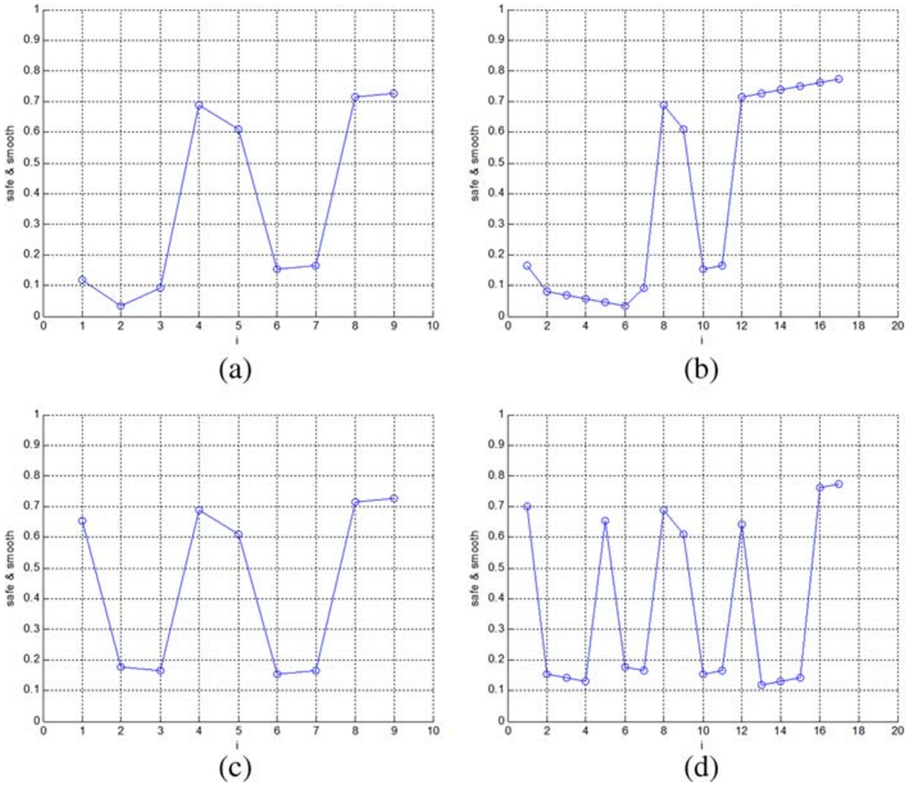
Fig. 11 Smoothness cost function value map



(a) (b)

(c) (d)

Fig. 12 Cost function value map for comprehensive security and smoothness



(a) (b)

(c) (d)

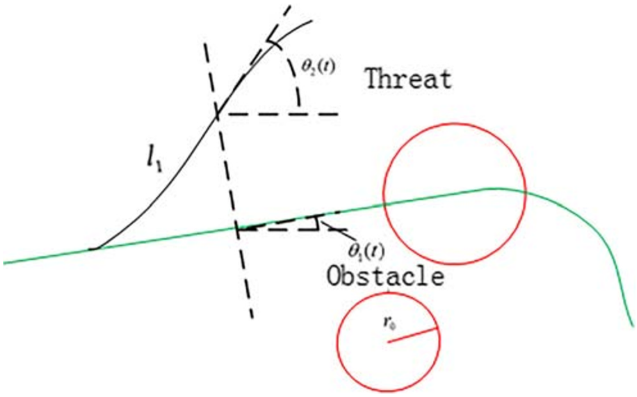


Fig. 13 Schematic diagram of coherent cost function parameters

other hand, the absolute value of the difference about the deflection angles of the two paths, its expression is:

$$f_{coherence}(i) = \frac{\lambda_1}{x_{start}^T - x_{mid}} \int_{x_{mid}}^{x_{start}^T} \times \sqrt{(x_0(t)-x)^2 + (y_0(t) - y_{yuan}(x))^2} dx + \lambda_2 \Delta\theta_i \quad (12)$$

In this formula, $\Delta\theta_i(t) = \frac{\|\theta_1(t)-\theta_2(t)\|}{\|x_{start}^T - x_{mid}/0.05\|}$, $\lambda_1 + \lambda_2 = 1$, $\theta_1(t)$ is the deflection angle between the original path and the x-axis, $\theta_2(t)$ is the deflection angle between the tangent of the candidate path and the x-axis ; as shown in Fig. 13, $y_{yuan}(x)$ is expressed as the ordinate of the original obstacle avoidance path that is the green path. And the black path represents a candidate path. The consistency cost function value graph is Fig. 14.

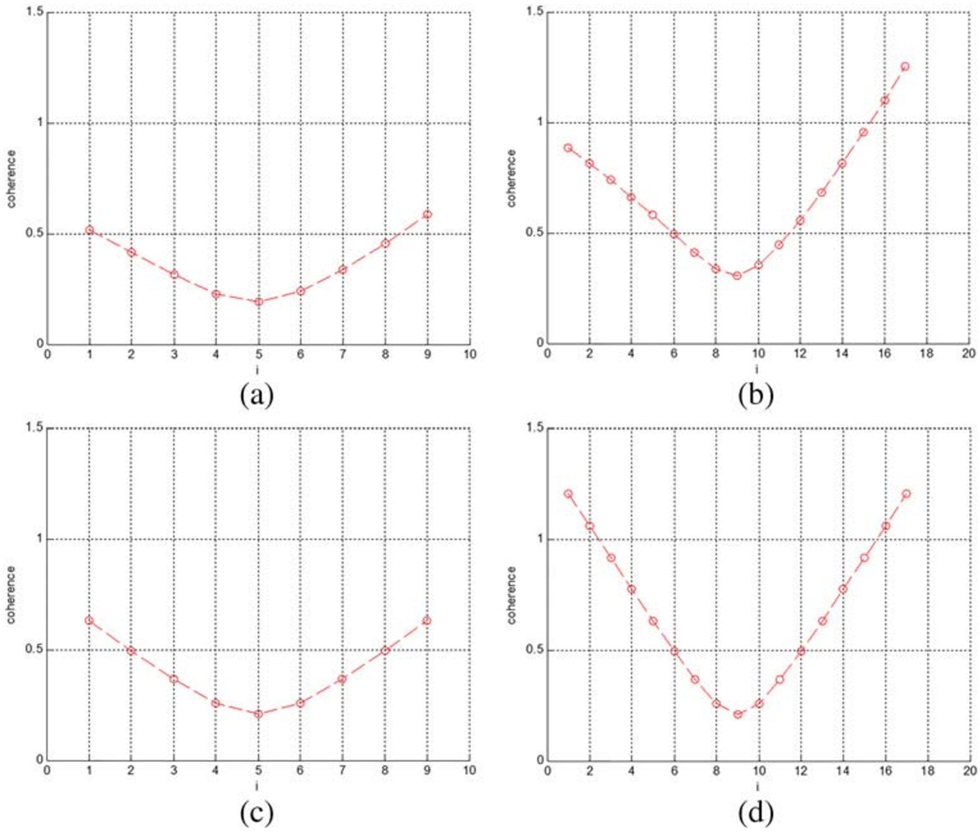
After analyzing the above three cost functions, comprehensive consideration of security, smoothness and consistency is obtained, and Fig. 15a-d corresponds to the case of Fig.8a–d.

It can be seen from Fig. 15a that the most ideal candidate path is the third candidate path, which is better than the result of combination of security and smoothness, because the third candidate path is more stable and safer with the requirements than the second candidate path. Similarly, the seventh item in Fig. 15b is the best candidate path, the seventh item in Fig. 15c is the best candidate path, and the eleventh in Fig. 15d is the best candidate path. They all get better results.

3.4 Dynamic Security Cost Function

The path obtained by weighting with the above three cost functions is only applicable to static or non-moving obstacles and sudden threats. The obtained path is not suitable for environments with the dynamic sudden threat.

Fig. 14 Consistency cost function value graph



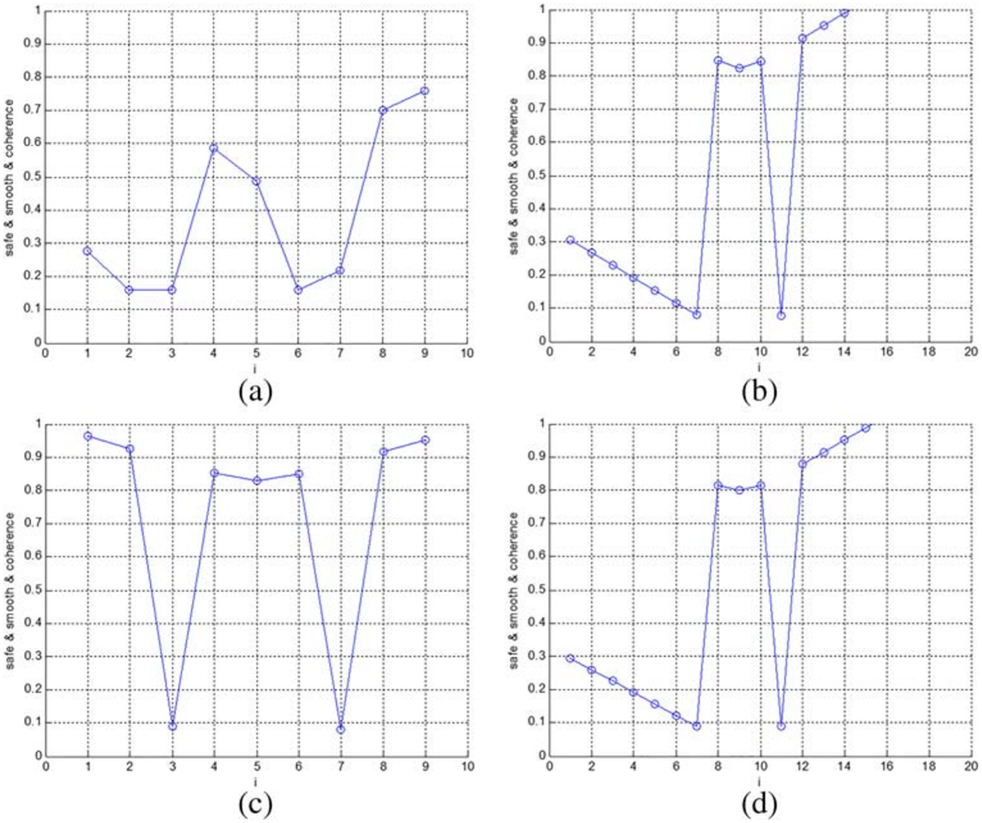


Fig. 15 Cost function graph for integrated security, smoothness and consistency

Once a dynamic threat is encountered, obtaining the best path will no longer be the optimal choice. The dynamic sudden threat can be detected by devices such as sensors, but all motion trajectories of dynamic sudden threats cannot be obtained, so real-time monitoring is required [7]. The algorithm runs for a short time, the dynamic sudden threat during this period can be seen as a moving obstacle for uniform linear motion. Therefore, the dynamic sudden threat is set to have a constant velocity. And it is set to be a plane circle along the trajectory of the road, i.e., the moving direction is the same as the tangential direction of the original obstacle avoidance path. Its physical meaning can be understood as the following situation.

If our UAV encounters a search of an enemy mobile UAV, the enemy UAV can be assumed as a circle with a certain radius that moves in a direction along. As shown in Fig. 16, it moves from point P_0 to point P_1.

It can be seen from Fig. 16 when the dynamic sudden threat moves from point P_0 to point P_1, it may collide with the candidate path, and the point which the collision may occur is the collision point.

To simplify the problem, the selection of collision points is divided into two cases. When the moving obstacle moves on the left side of the UAV, the intersection points of the bottom movement trajectory of the movement obstacle near the side of the UAV and the candidate paths are set as the collision points. And the red line indicates the movement trajectory in Fig. 17a and c, the pink points indicate the collision points. When the moving obstacle moves on the right side of the UAV, the top movement trajectory of the moving obstacle near the side of the UAV is intersects with the candidate paths, the intersection points are set as the collision points as shown in Fig. 17b and d, the red

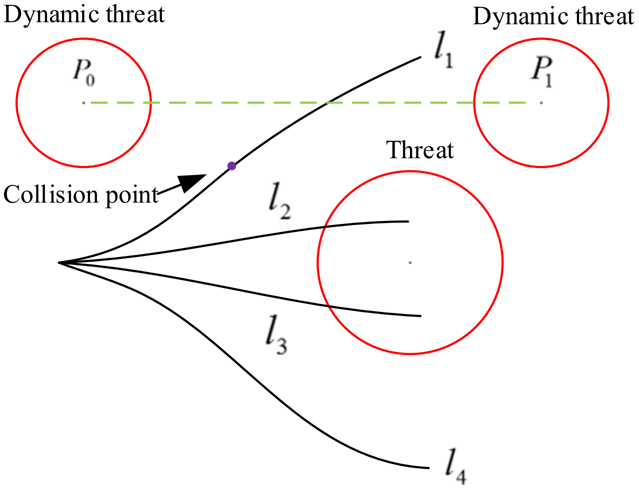
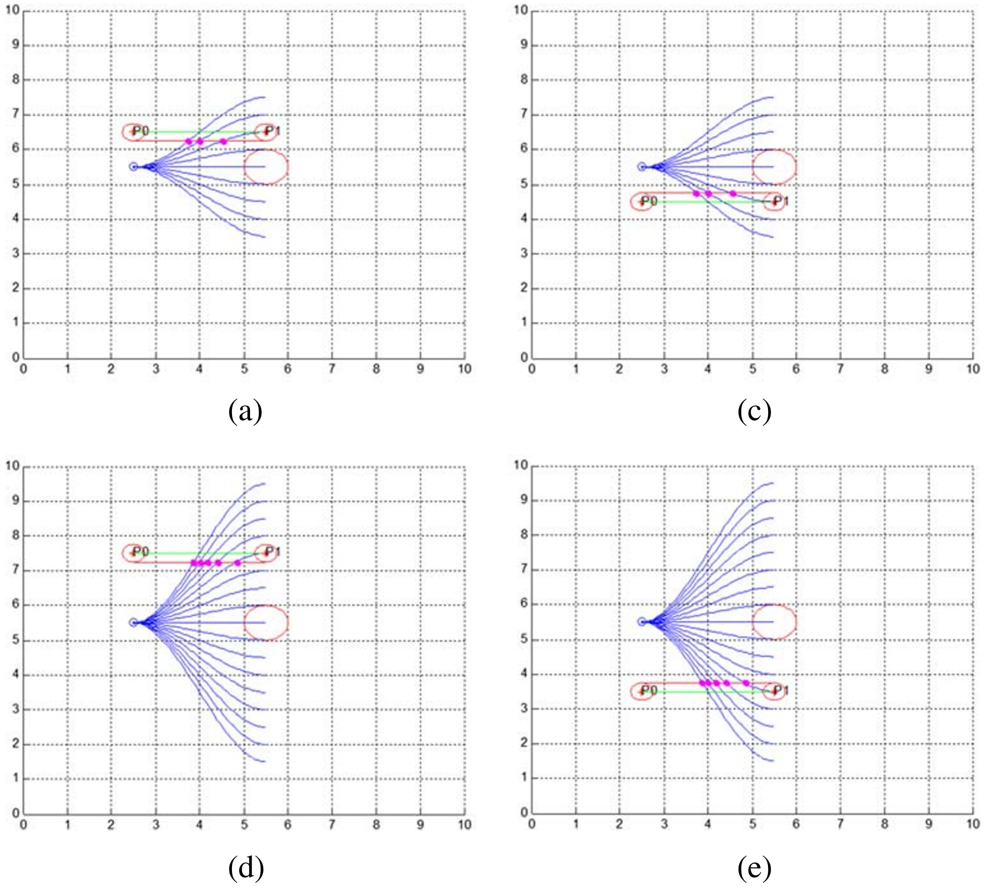


Fig. 16 Dynamic burst threat Mobility Diagram



(a)

(c)

(d)

(e)

Fig. 17 Schematic diagrams of four dynamic sudden threat movements

line indicates the movement trajectory, the points of pink represent the collision points.

The four cases of Fig. 17 are combined with the four cases of Fig. 8 to obtain eight cases as shown in Fig. 18a–h. In Fig. 18, blue paths are the candidate paths, green path is the original obstacle avoidance path, small hollow circles are known obstacles.

It can be seen in Fig. 15a–d that the optimal paths are obtained as r_3 and r_7 by the security cost function, the smoothness cost function and the consistency cost function without considering the dynamic sudden threat, but some optimal paths are collided with dynamic sudden threats. In order to avoid collisions ,necessary factors such as the speed and acceleration of the UAV must be considered. If a dynamic sudden threat flies along a path, it is likely to collide with the UAV at the point of collision. Therefore, the factors of the speed and acceleration need to be considered.

The acceleration of the UAVs candidate path can be expressed as:

$$a(r_i) = 2(S_c - S_g)v_0^2/(\Delta s(r_i) - S_c)^2 \tag{13}$$

In this formula, S_c represents the sum of the radius of the dynamic sudden threat and the overall radius of the UAV, S_g is the distance between the UAV and the dynamic sudden threat; $\Delta s(r_i)$ is the arc between the starting point and the collision point in the candidate path; v_0 is the speed of the UAV, which is set to 8m/s in this paper.

$$S_g = \begin{cases} S_0, \ S_0 \geq \Delta s(r_i) \\ \Delta s(r_i), \ others \end{cases} \tag{14}$$

In this formula, S_0 is the distance between the UAV and the dynamic threat.

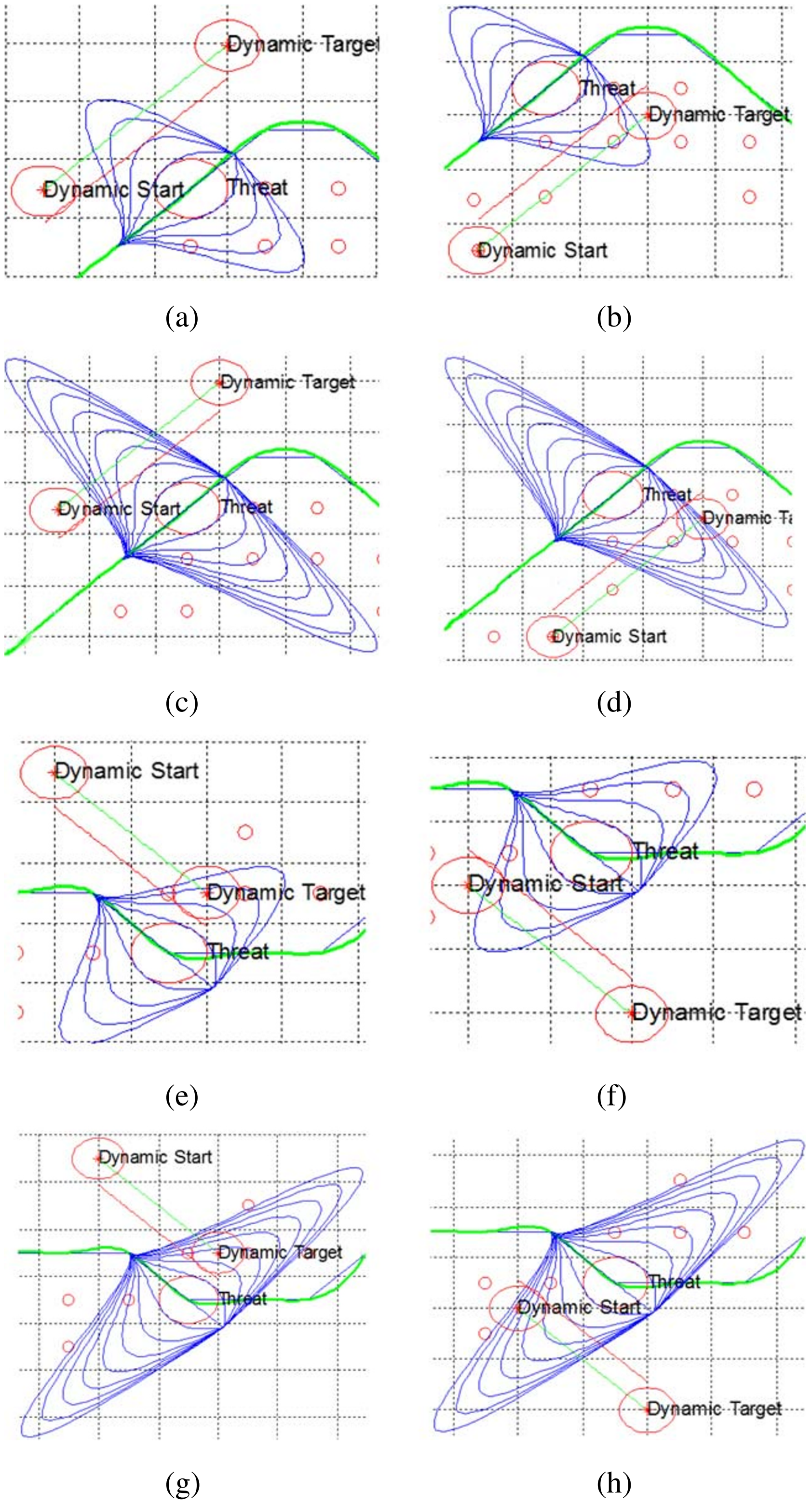
The speed of the UAV is constrained in reality, such as the influence of the curvature of the path, the impact of safety, and the maximum speed limit that the UAV can reach. The final speed is expressed as follows:

$$v_{lim}(r_i) = \min[v_{sm}(r_i), v_{sa}(r_i), v_{sign}] \tag{15}$$

In this formula, $v_{sm}(r_i)$ represents the speed of the UAV's candidate path that is affected by its curvature, which can be expressed as:

$$v_{sm}(r_i) = \sqrt{\frac{|\ a_l\ |_{\max}}{\max\ k(r_i)}} \tag{16}$$

Fig. 18 Candidate path Graph in eight cases

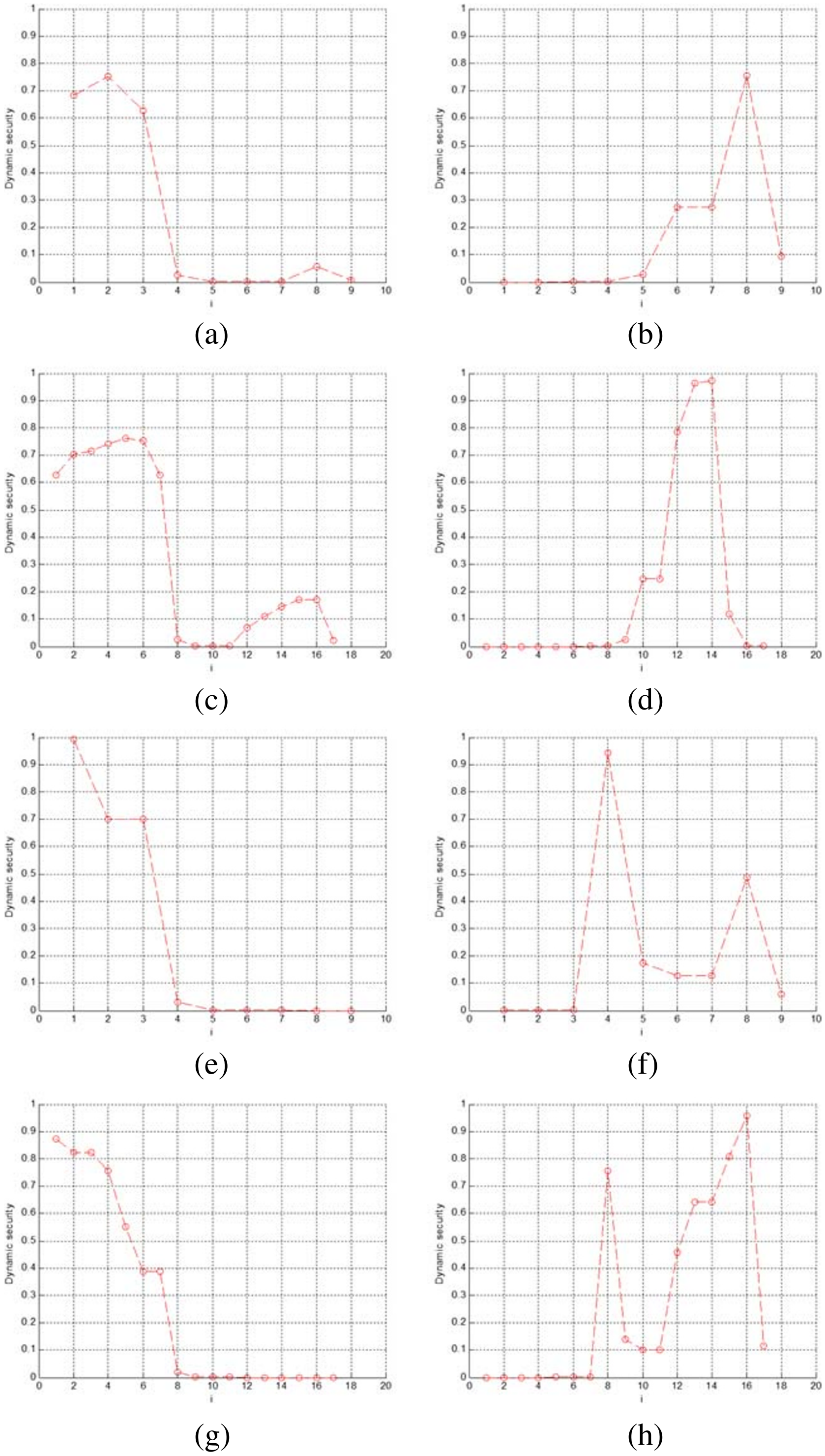


$v_{sa}(r_i)$indicates the speed of the UAVs candidate path under the influence of security factors. Its expression is as follows:

$$v_{sa}(r_i) = v_k - k_{safe} f^2_{safe}(r_i) v_k \qquad (17)$$

In this formula, $| \ a_l \ |_{\max}$ is the lateral acceleration, it is set to 5000, max $k(r_i)$ is the maximum curvature of the candidate path, k_{safe} is the safety gain, in this paper it is set to 0.9; v_k is the reference speed of the path, it is set to 50.

Fig. 19 Dynamic security cost function value graph in eight cases



(a) (b)

(c) (d)

(e) (f)

(g) (h)

Velocity has certain limits, so acceleration has limitations as velocity too, the restrictions are as follows:

$$a(r_i) \leq \frac{0.5(v_{lim}^2(r_i) - v_0^2)}{(\Delta s(r_i) - S_g)} \tag{18}$$

In order to make the acceleration in the continuous planning step similar, the Gaussian filtering method is used to process the acceleration.

In the dynamic security cost function, acceleration is a very important factor, so the established cost function is related to acceleration. The dynamic security cost function can be expressed as:

$$f_{dynamic}(r_i, a(r_i)) = \mid a(r_i) \mid \Delta s(r_i) - S_g \mid a(r_i) \mid \tag{19}$$

The function graph is obtained by using the dynamic security cost function that is shown in Fig. 19. It can be seen from the Fig. 19a that the value from $r_1 \sim r_3$ is much higher than other candidate paths, and it also shows that the three candidate paths are easy to collide with the dynamic sudden threat. The function value of r_6 is the smallest, so r_6 is the optimal path when only the dynamic security cost function is considered. However, we can know from Fig. 18a that r_6 is not the optimal path for this plan, so other cost functions should be added together, as the same with the other case. Thus, the total cost function is introduced.

3.5 Total Cost Function

The four cost functions are obtained now, and a total cost function is designed. The candidate path corresponding to the minimum value of the function is the optimal path. The total cost function of the design can be expressed as:

$$\begin{aligned} f_z(r_i, a(r_i)) = {} & \beta_1 f_{safe}(r_i) + \beta_2 f_{smooth}(r_i) \\ & + \beta_3 f_{coherence}(r_i) + \beta_4 f_{dynamic}(r_i, a(r_i)) \end{aligned} \tag{20}$$

In this formula, $\beta_1 + \beta_2 + \beta_3 + \beta_4 = 1$, β_1, β_2, β_3 and β_4 are the weights of the static security cost function, the smoothness cost function, the consistency cost function, and the dynamic security cost function, respectively. If the total cost function is considered to focus on security, the following conditions should be met:

$$\beta_1 > \max\{\beta_2, \beta_3, \beta_4\} \tag{21}$$

Other situations are equally available.

4 Overall Structure of the Framework and Global Path Planning Method

4.1 Overall Structure of the Framework

The overall process of this paper is shown in Fig. 20. Step1: First, the environment information is set and the location of the starting point, target point and obstacles are got. Step2: The global path planning for the UAV bases on A* algorithm. Using the A* algorithm to generate a path from the starting point to the target point, you can completely avoid the obstacles and then smooth it. Step3: When a sudden threat is detected, a series of candidate paths are generated according to the cubic spline curve equation. Then, the candidate paths are adjusted as required. Step4: The target total cost function is set to select the optimal path.

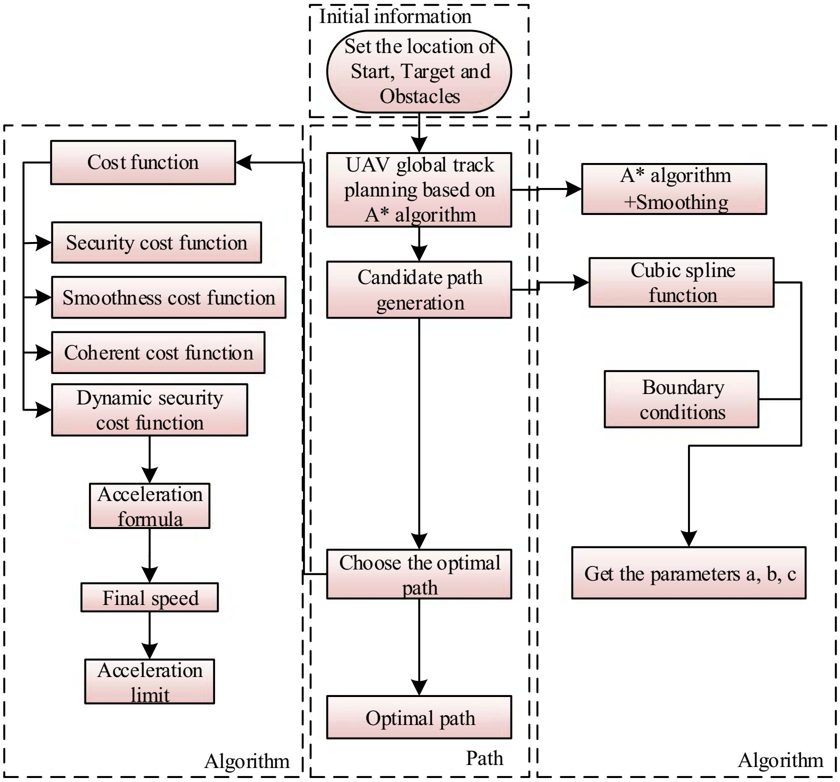


Fig. 20 Overall flow chart

4.2 Global Path Planning Method

In an environment with known obstacles, an optimal path is generated for avoiding obstacles. This paper uses A* algorithm to plan the path, the A* algorithm flow chart is shown in Fig.21.

The A* algorithm adopts the idea of heuristic search and evaluates each search path in the state space to get the best path, and then searches from this path to the target [31]. Its cost function is expressed as follows $F = G + H$, F is the evaluation function, G is the shortest path value from the starting position to the current location of the node, and H is the shortest path of the currently located node to the target point [32]. The A* algorithm can obtain discrete points in the optimal path, and these points can ensure that they are not in obstacle interior or not contacted with the obstacle boundary. Then these points are connected into a polyline to get an optimal path that completely avoids the obstacle. At last, the path is smoothed so that it has no vertices.

5 Experimental Results and Analysis

5.1 Comprehensive Analysis

In this paper, two cases are considered. According to the A* algorithm, two optimal trajectories are obtained, as shown in Fig. 22a and b, then the two paths are smoothed to obtain two smooth optimal paths, as shown in Fig. 22c and d.

After the UAV meets the sudden threat, the local path planning is needed, in this paper, two kinds of 9 candidate paths and 17 candidate paths are considered respectively. They are combined to get the four cases of a-d in Fig. 23.

After obtaining the total cost function, the best path that meets the requirements is obtained by giving the appropriate weights. The chosen values for the weights in this paper are 0.65, 0.15, 0.1, and 0.1, respectively. Different weight combinations will appear in different focuses. Through the total cost function, 8 kinds of cost function value (Fig. 24) and final effect (Fig. 25) are obtained. Figure 25a–h correspond to Fig. 24a–h, and the

Fig. 21 A* algorithm flow chart

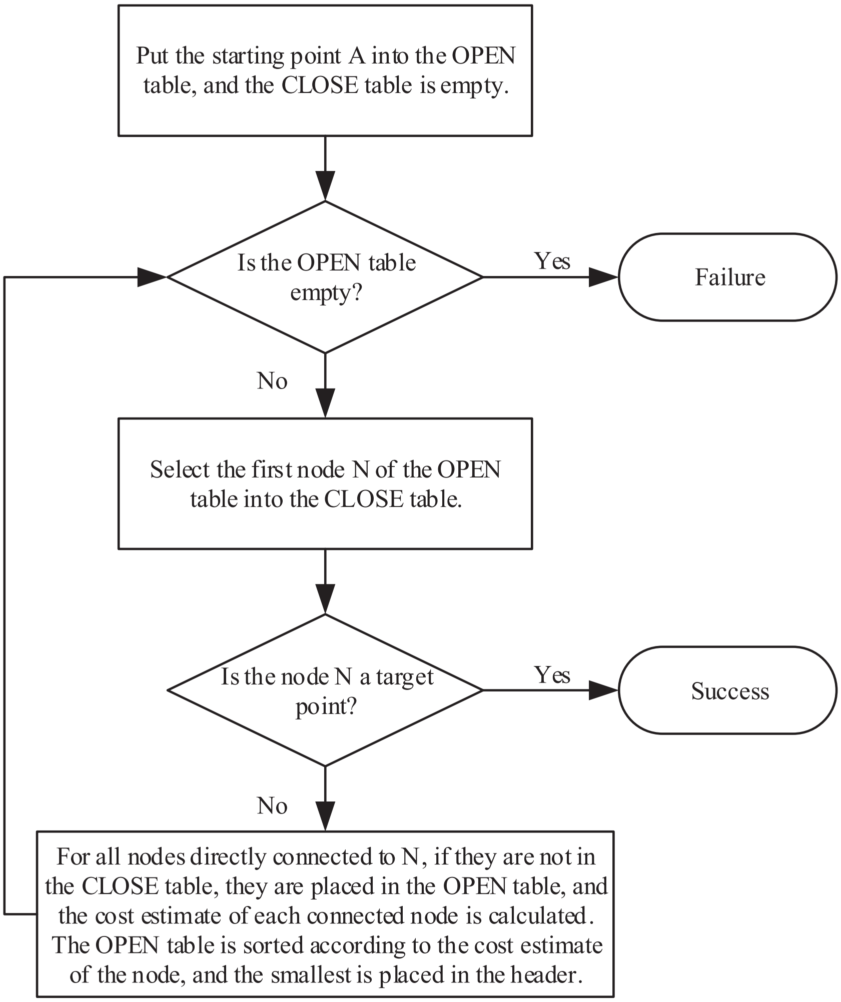
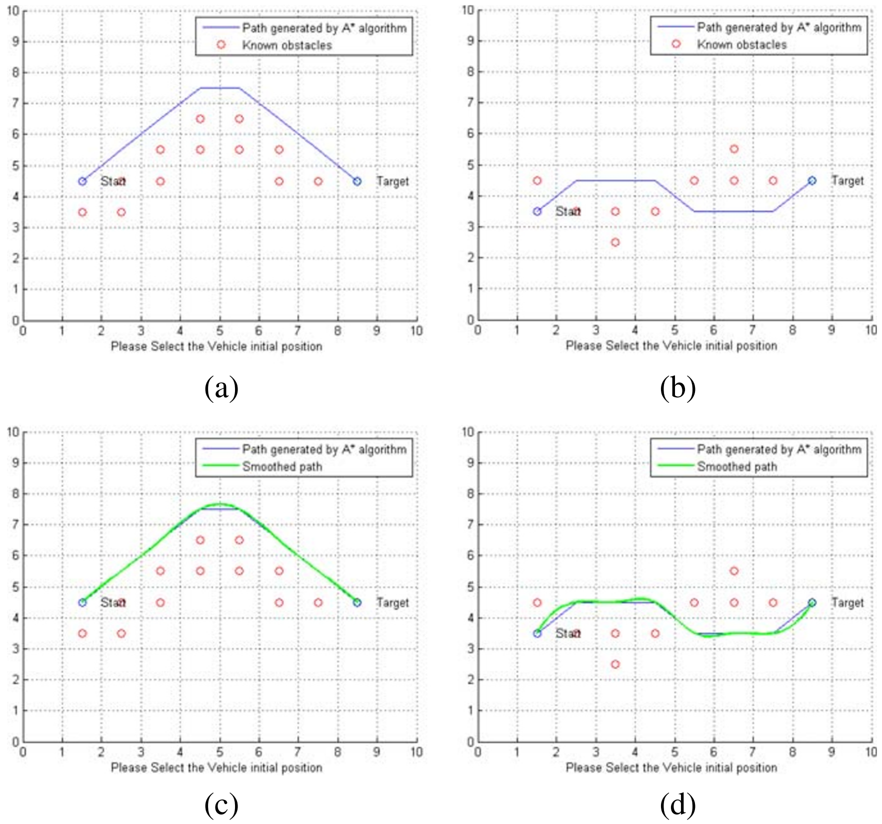

Put the starting point A into the OPEN table, and the CLOSE table is empty.
Is the OPEN table empty?
Yes
Failure
No
Select the first node N of the OPEN table into the CLOSE table.
Is the node N a target point?
Yes
Success
No
For all nodes directly connected to N, if they are not in the CLOSE table, they are placed in the OPEN table, and the cost estimate of each connected node is calculated. The OPEN table is sorted according to the cost estimate of the node, and the smallest is placed in the header.

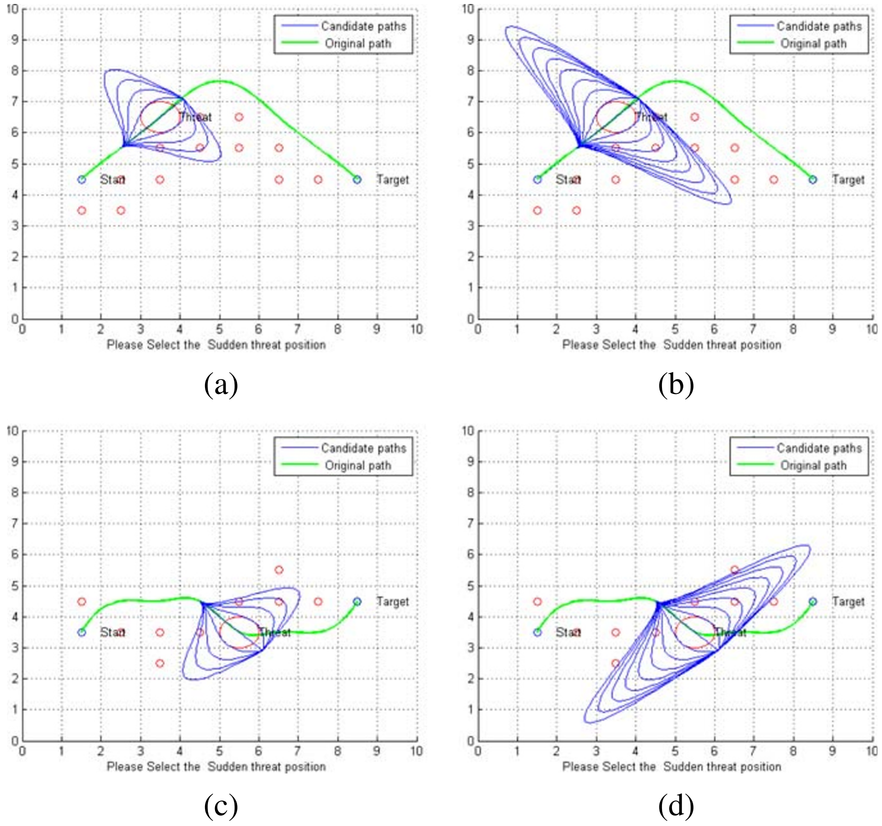
Fig. 22 Path planning diagram based on A* algorithm



(a) (b)

(c) (d)

Fig. 23 Four-case candidate path generation graph



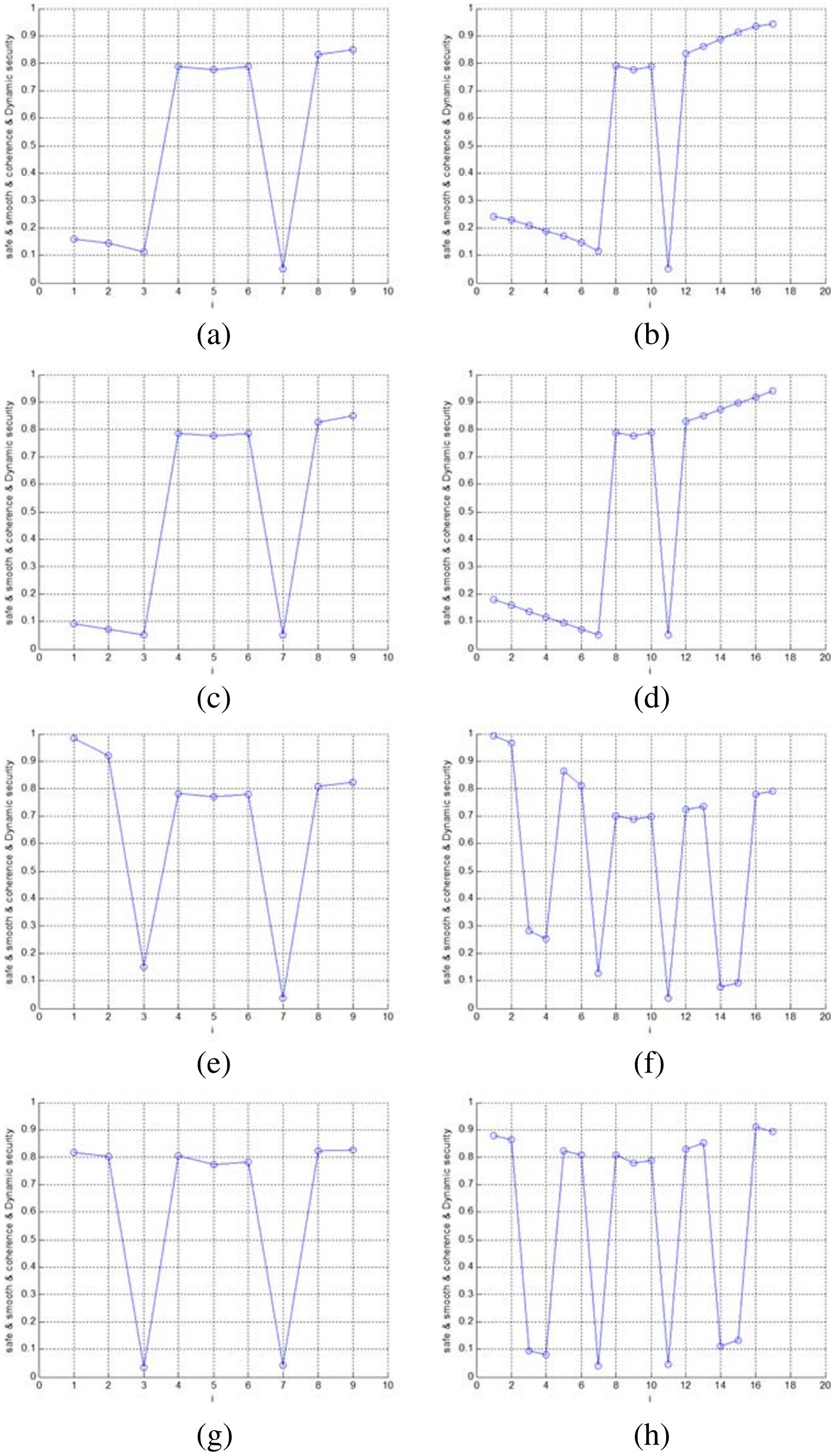
(a) (b)

(c) (d)

candidate paths corresponding to the minimum value of the total cost function value is selected as the best path, i.e., the path is marked with red in Fig. 25. Both Fig. 25a and c are case 1, but the difference is the location of the dynamic threat, so the optimal path chosen is different. Other situations are similar.

A comparative analysis of dynamic threats and no dynamic threats will be conducted in Table 1.

Fig. 24 Total cost function graph



(a)

(b)

(c)

(d)

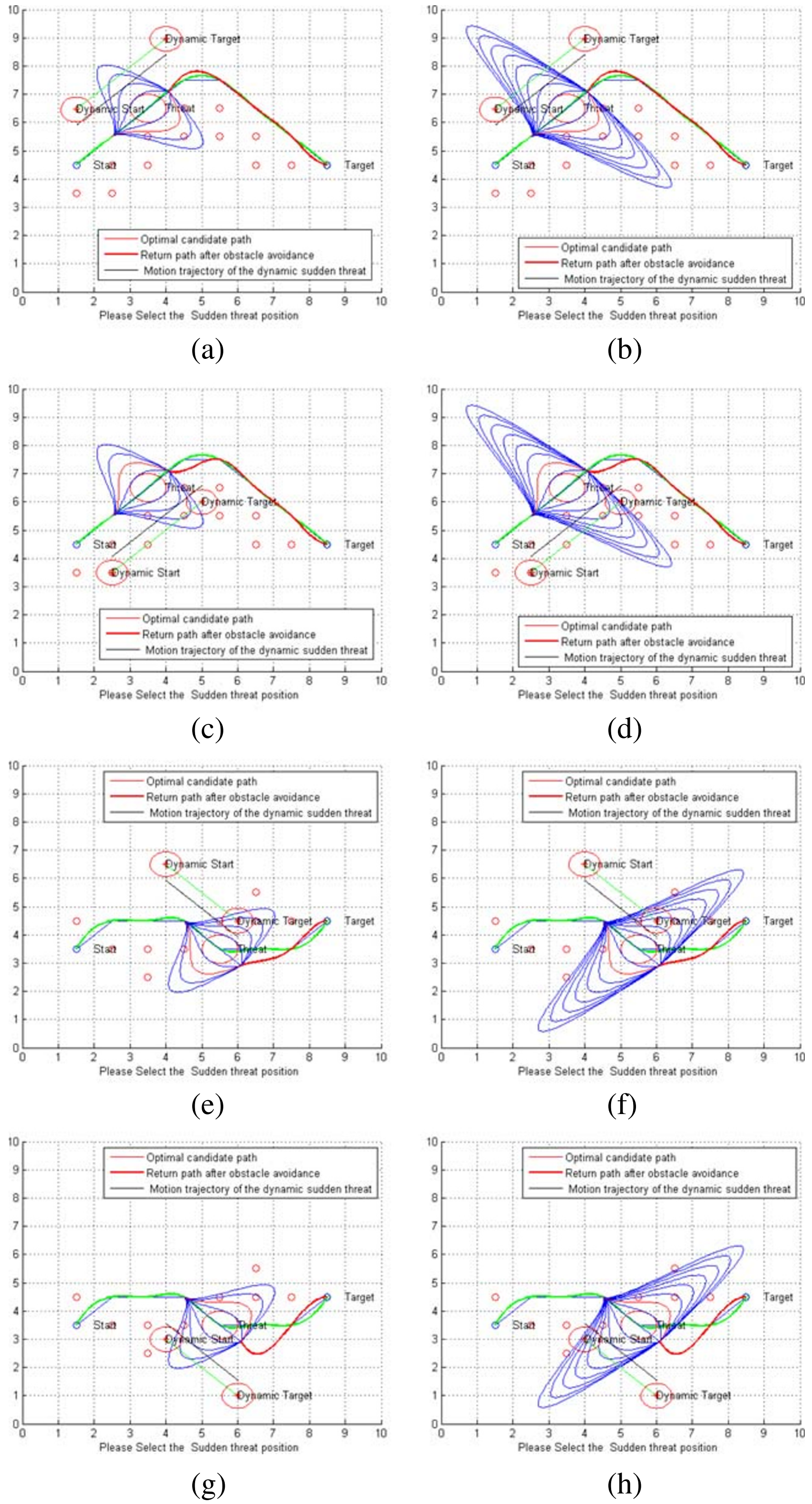
(e)

(f)

(g)

(h)

Fig. 25 Final effect graph



(a)

(b)

(c)

(d)

(e)

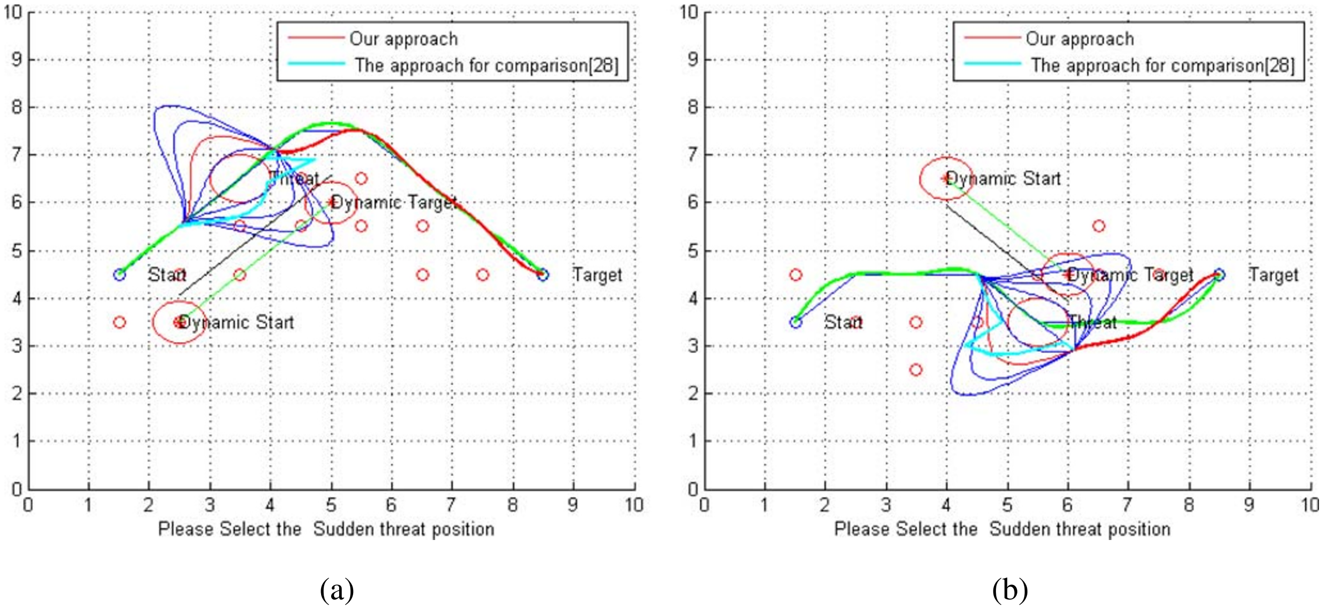
(f)

(g)

(h)

Table 1 Dynamic sudden threat comparison table

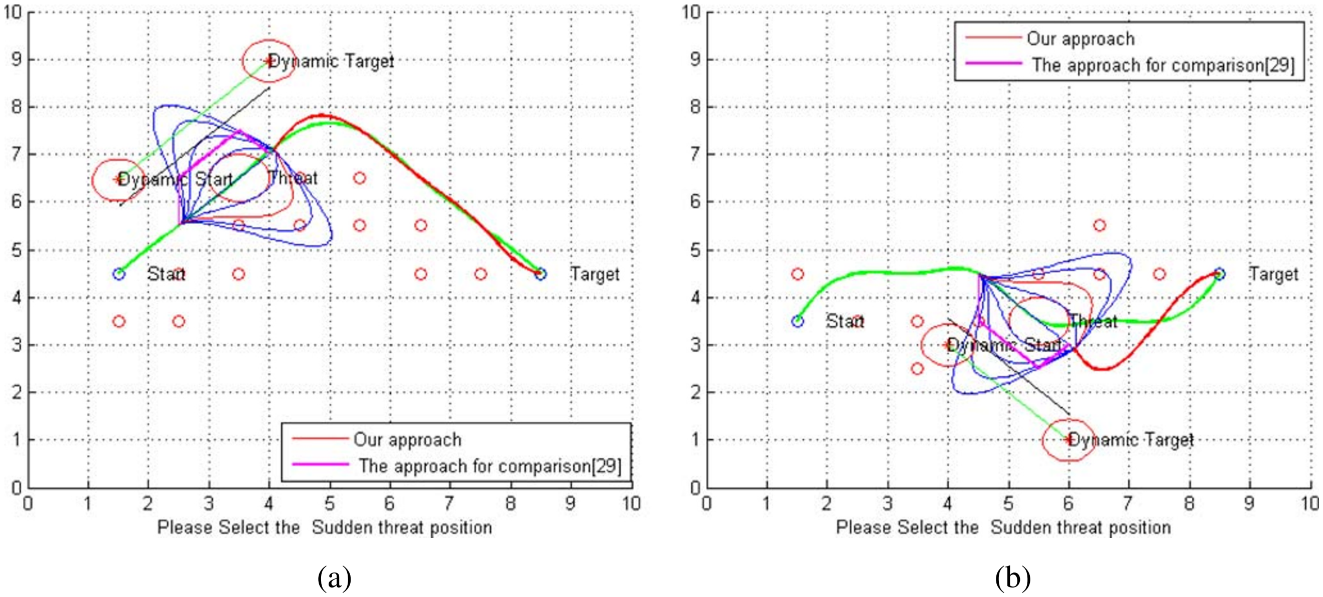
$i - th\ candidate\ path$	$Case1$	$Case2$	$Case3$	$Case4$
No dynamic threats	3rd	7th	7th	11th
Dynamic threat (left)	7th	11th	7th	11th
Dynamic threat (right)	3rd	7th	3rd	7th



(a)

(b)

Fig. 26 The method proposed by us is compared with the method



(a)

(b)

Fig. 27 The method proposed by us is compared with the method

It can be seen from the table that after adding a dynamic sudden threat, some of the optimal paths change. After the change, it can satisfy the safe and smooth path to avoid the dynamic sudden threat and meet the requirements of the path.

5.2 Running Time

In this paper, the experimental hardware is Intel(R) Core(TM) i5-3470 CPU @ 3.20GHz, 4.00GB memory. The path generation is obtained by Matlab 2013 simulation. It is known from experiments, the time to generate the candidate paths is about 0.465809s, the selection time is about 1s, and the sum of the time is about 1.5s. Of course, different hardware and other conditions will produce different runtimes.

5.3 Comparison with the Existing Techniques

To demonstrate the effectiveness of the proposed method, we compare our approach with other two recent obstacle avoidance methods[28, 29]. In the same situation, the two methods simultaneously evade sudden threats. In this paper, two representative conditions of Fig. 25c and e are selected for comparative simulation with [28]. Figure 25a and g are adopted to compare with the approach[29]. Through these two examples, a general conclusion can be obtained.

In Figs. 26 and 27, the blue-green path is the approach [28]. Similarly, the pink path is the approach[29]. The red path in the candidate path cluster is our approach. By comparison, we can see that the generated path by our method is smoother and can avoid both static and dynamic threats. Moreover, the simulation effect of the other methods show that the distance between their obstacle avoidance path and the sudden threat is closer than our obstacle avoidance path, so that they hardly meet the security requirements. Our approach is compared with the other two approaches, as shown in Figs. 26 and 27.

6 Conclusion

This paper proposes a dynamic path planning method for the UAV under sudden threats. Firstly, the A* algorithm is used to plan the optimal path for obstacle avoidance in the environment of known obstacle information. Then, when a sudden threat is encountered, the corresponding parameters are set according to the cubic spline second-order continuity, and multiple candidate paths are generated. The symmetry of candidate paths and the coordinates of common points of many candidate paths are analyzed. Static security, smoothness, consistency and dynamic security cost functions are further established to optimize candidate paths. Eventually, the total cost function is established, and the optimal path of the total cost function is given. The path planning from the starting point to the ending point is realized. In the listed cases, the experimental results show an obtained path that meets all requirements. Additionally, this method has the characteristics of short time-consuming and strong real-time. Since this article only considers the case where the dynamic threat and the static threat are both circular, there are no situations involving other threats. In order to make the UAV adapt to the operational environment better, how to consider the path planning of multiple sudden threats coexist that will be an issue that needs further research in the future.

References

1. Ollero, A., Kondak, K.: 10 years in the cooperation of unmanned aerial system. In: 2012 IEEE/RSJ International Conference on Intelligence Robots and Systems, p. 5450C5451 (2012)
2. Jiang, B., Bishop, A.N., Anderson, B.D., Drake, S.P.: Optimal path planning and sensor placement for mobile target detection. Automatica **60**, 127C139 (2015)
3. Guruprasad, K., Ghose, D.: Deploy and search strategy for multi-agent systems using Voronoi partitions. In: 4th International Symposium on Voronoi Diagrams in Science and Engineering, p. 91C100 (2007)
4. Hayat, S., Yanmaz, E., Muzaffar, R.: Survey on unmanned aerial vehicle networks for civil applications: a communications viewpoint. IEEE Communications Surveys and Tutorials **18**(4), 2624C2661 (2016)
5. Sung, I., Nielsen, P.: Zoning a service area of unmanned aerial vehicles for package delivery services. J. Intell. Robot. Syst. (2019)
6. Babel, L.: Coordinated target assignment and UAV path planning with timing constraints. J. Intell. Robot. Syst. **94**, 857 (2019)
7. Yao, W., Lu, H., Zeng, Z.: Distributed static and dynamic circumnavigation control with arbitrary spacings for a heterogeneous multi-robot system. J. Intell. Robot. Syst. **94**, 883 (2019)
8. Lei, T., Zhang, Y., Lu, J.: The application of UAV remote sensing in mapping of damaged buildings after earthquakes. International Conference on Digital Image Processing, 10806 (2018)
9. Rossi, M., Brunelli, D.: Autonomous gas detection and mapping with unmanned aerial vehicles. IEEE Trans. Instrum. Meas. **65**(4), 765C775 (2016)
10. Primatesta, S., Rizzo, A., Cour-Harbo, L.A.: A ground risk map for unmanned aircraft in urban environments. J. Intell. Robot. Syst. (2019)
11. Mairaj, A., Baba, A.I., Javaid, A.Y.: Application specific drone simulators: recent advances and challenges. Simul. Model. Pract. Theory, 100–117 (2019)
12. Haartsen, Y., Aalmoes, R., Cheung, Y.: Simulation of unmanned aerial vehicles in the determination of accident locations. In: ICUAS 2016, International Conference on Unmanned Aircraft Systems, p. 993C1002 (2016)
13. Wang, M., Liu, J.N.K.: Fuzzy logic based robot path planning in unknown environment. In: The IEEE International Conference on Machine Learning and Cybernetics, p. 813C 818. IEEE (2005)
14. Lumelsky, V.J., Stepanov, A.A.: Path-planning strategies for a point mobile automaton moving amidst unknown obstacles of arbitrary shape. Algorithmica **2**, 403C430 (1987)

15. Kamon, I., Rivlin, E., Rimon, E.: New range-sensor based globally convergent navigation algorithm for mobile robots. In: IEEE International Conference on Robotics and Automation, p. 429C435 (1996)
16. Molinos, E.J., Llamazares, N., Oca?a, M.: Dynamic window based approaches for avoiding obstacles in moving. Robotics and Autonomous Systems **118**, 112C1 (2019)
17. Fox, D., Burgard, W., Thrun, S.: Dynamic window approach to collision avoidance. IEEE Robot. Autom. Mag. **4**(1), 23–33 (1997)
18. Ogren, P., Leonard, N.E.: A convergent dynamic window approach to obstacle avoidance. IEEE Trans. Robot. **21**(2), 188–195 (2005)
19. Zuo, L., Guo, Q., Xu, X., Fu, H.: A hierarchical path planning approach based on a and least-squares policy iteration for mobile robots. Neurocomputing **170**(c), 257C266 (2015)
20. Yuan, Y., Xing-she, Z., Kai-long, Z.: Dynamic trajectory planning for unmanned aerial vehicle based on sparse A* search and improved artificial potential fiel. Control Theory and Applications **27**(07), 953–959 (2010)
21. Haddock, J., Mittenthal, J.: Simulation optimization using simulated annealing. Comput. Ind. Eng. **22**(4), 387–395 (1992)
22. Pierreval, H., Tautou, L.: Using evolutionary algorithms and simulation for the optimization of manufacturing systems. IIE Transactions (Institute of Industrial Engineers) **29**(3), 181–189 (1997)
23. Alireza Feyzbakhsh, S., Matsui, M.: Adam-Eve-like genetic algorithm: a methodology for optimal design of a simple flexible. Comput. Ind. Eng., 233–258 (1999)
24. Phung, M.D., Cong, H.Q., Dinh, T.H., Ha, Q.: Enhanced discrete particle swarm optimization path planning for UAV vision-based surface inspection. Autom. Constr. **81**, 25C33 (2017)
25. Yang, K., Sukkarieh, S.: 3D smooth path planning for a UAV in cluttered natural environments. In: IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 794–800. IEEE (2008)
26. Jayasinghe, J.A.S., Athauda, M.B.G.D.A.: Smooth trajectory generation algorithm for an unmanned aerial vehicle (UAV) under dynamic constraints: using a quadratic Bzier curve for collision avoidance. In: 2016 Manufacturing and Industrial Engineering Symposium: Innovative Applications for Industry, pp. 1–6. MIES (2016)
27. Zhou, S., Zhu, G., Li, H., Wang, Y., Liu, X.: Real-time route planning for UAV based on weather threat. In: 2011 International Conference on Remote Sensing, Environment and Transportation Engineering, vol. 2011, pp. 2342–2345. RSETE (2011)
28. Xia, C., Yudi, A.: Application of improved neural network in 3D path planning of UAVs. Electron. Opt. Control. **25**(9), 7–11 (2018)
29. Xia, C., Yudi, A., Hongli, L.: Research on three-dimensional path planning of UAV based on improved ant colony algorithm. Tactical Missile Technology **02**, 59–66 (2019)
30. Wang, H., Kearney, J., Atkinson, K.: Arc-length parameterized spline curves for real-time simulation. In: 5th International Conference on Curves and Surfaces, pp. 387–396 (2002)
31. Maneev, V.V., Syryamkin, M.V.: Optimizing the A* search algorithm for mobile robotic devices. Materials Science and Engineering **516**(1) (2019)
32. Qi, Z., Aqun, Z.: A multipath seeking algorithm based on a * algorithm. J. Electron. Inf. Technol. **35**(04), 952–957 (2010)

Xia Chen She received a Ph.D., Decision Science from the Department of Systems Engineering at Northeastern University in 2006. From January 2006 to December 2009, she was a postdoctoral researcher at the Department of Information Science and Engineering at Northeastern University. She is engaged in research and education at Shenyang Aerospace University. She is currently as a professor at the School of Automation, Shenyang Aerospace University. Her research interests include multi-UAV automatic control and decision-making, multi-UAV mission planning and path planning, and multi-UAV control and countermeasure technologies.

Miaoyan Zhao She has obtained a master's degree in 2020 from the School of Automation, Shenyang Aerospace University. Shenyang, China. Her research focus on UAV path planning.

Liyuan Yin he has obtained a master's degree in 2020 from the School of Automation, Shenyang Aerospace University. Shenyang, China. His research focus on. multi-agent system consistency.