

Research on path planning of multi-rotor UAV based on improved artificial potential field method

Zhengqing Liu^{*}, Xinhua Wang¹, and Kangyi Li

Nanjing University of Aeronautics and Astronautics, Jiangsu Nanjing 211100, China

Abstract. UAV needs sensor to fly in an environment with obstacles. However, UAV may not be able to move forward when it encounters a large obstacle, or UAV will be in a dangerous state when the sensor fails briefly which disturbed by the environment factors. In order to solve these problems, the following methods are proposed in this paper. Aiming at the first problem, this paper proposes an improved APF method for path planning, and verified by simulation experiments that this method can find the optimal path. Aiming at the second problem, this paper proposes a solution to expand the range of obstacles and dynamically change the distance in the APF repulsion function. It is verified that the UAV can fly safely within the short time of the sensor problem by simulation experiments. In conclusion, this paper has an important reference value for the application of UAV online dynamic path planning in engineering.

1 Introduction

With the rapid advancement of science and technology, the technology of unmanned aerial vehicle(UAV) has developed rapidly. The obstacle avoidance sensors used on UAV have higher accuracy, longer detection distances and stronger anti-interference performance. Subsequently, many scholars proposed a lot advanced path planning algorithms, there are traditional methods such as A* algorithm[1-3], APF method[4-5], and Ant Colony algorithm[5-8], as well as intelligent path planning methods based on Neural Network[9] and Genetic algorithms[10] etc.

However, there may be some problems as follow that are not considered in the above methods in engineering applications.

(1)When the UAV encounters a striped obstacle (such as a wall), the traditional APF method may not work properly. The addition of lateral disturbance may temporarily leave the local optimal state, but it may cause the UAV path to oscillate, which is not conducive to flying;

(2)In the case of interference from the external environment, the sensor may have problems such as decreased accuracy and shorter detection range. When the global information is unknown and the sensor is not working properly, the traditional path

^{*} Corresponding author: lzqlys@126.com

planning method will put the UAV into a dangerous state. Therefore, this paper mainly studies these two issues.

2 Traditional path planning algorithm

2.1 Artificial potential field method

The APF method is a virtual force method, which equates the movement of a UAV in an environment to the movement of a charged ball in a force field environment with both positive and negative charges. The APF method has strong adaptability to unknown environments and relatively simple calculations. Therefore, it is widely used because it meets the requirements that UAV can avoid obstacle real-time[4].

The repulsion field function of the traditional APF method is as follows:

$$U_{rep}(q) = \begin{cases} \frac{1}{2}\eta\left(\frac{1}{\rho(q, q_{obs})} - \frac{1}{\rho_0}\right)^2 & \rho(q, q_{obs}) \leq \rho_0 \\ 0 & \rho(q, q_{obs}) > \rho_0 \end{cases} \quad (1)$$

$\rho(q, q_{obs})$ is the distance between the UAV and the obstacle, and ρ_0 is the radius of influence of the obstacle. If $\rho(q, q_{obs}) > \rho_0$, it means that the UAV is in a safe area and there is no need to avoid collision.

The repulsion can be understood as the gradient function of the repulsion field:

$$F_{rep}(q) = -\nabla U_{rep}(q) = \begin{cases} \eta\left(\frac{1}{\rho(q, q_{obs})} - \frac{1}{\rho_0}\right)\frac{1}{\rho^2(q, q_{obs})}\nabla\rho(q, q_{obs}), \rho(q, q_{obs}) \leq \rho_0 \\ 0 & , \rho(q, q_{obs}) > \rho_0 \end{cases} \quad (2)$$

Similarly, the gravitational field function and gravity are:

$$U_{att}(q) = \frac{1}{2}\xi\rho^2(q, q_{obs}) \quad (3)$$

$$F_{att}(q) = -\nabla U_{att}(q) = \xi\rho(q, q_{obs}) \quad (4)$$

In the end, the direction of the UAV's movement is the combined force of the obstacle's repulsion and the target's gravity.

2.2 Bug2 algorithm

The principle of the Bug algorithm[11] is similar to the movement strategy of insect crawling. When the sensor does not detect the obstacle, the UAV moves to the target in a straight line. When encountering the obstacle, the UAV keeps a certain distance from the obstacle and moves along the obstacle. This algorithm is relatively simple and not affected by the global position information and the shape of obstacles, but the planned path is longer and inefficient.

Bug2 is a bit simpler than the Bug1 algorithm, there are still two main actions: motion-to-goal and boundary-following, as shown in Fig.1, the UAV moves along the edge of the obstacle. If it encounters the line $\Gamma(X_{po\ int}, X_{goal})$ between $X_{po\ int}$ and X_{goal} again, it will continue along $\overline{X_{po\ int} X_{goal}}$ until it encounters the obstacle again or reaches the target.

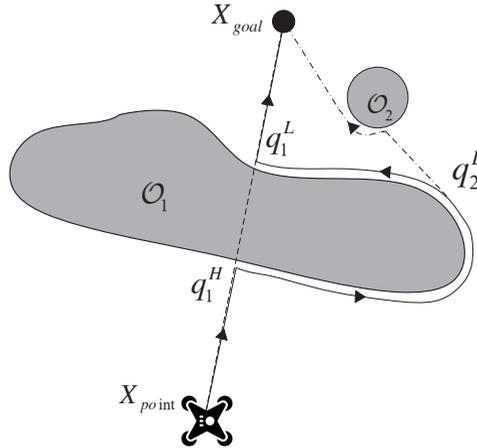


Fig. 1. Bug2 and APF method path planning principle analysis diagram.

3 Improved path planning algorithm

The UAV may encounter obstacles with special shapes during the flight, such as the bar-shaped obstacles shown in Fig.1. If the traditional path planning method such as APF or bug2 algorithm is used alone, it may not be able to bypass the obstacles or fly extra distances. Next, we compare and analyse the advantages and disadvantages of the above two.

q_1^H is the point where the UAV first encounters the obstacle, we can call it hit point, so q_1^L , q_2^L must be the leave point. When the UAV is at the point q_1^H , the UAV is very likely to fall into the local optimal position and cannot move forward, if the traditional APF method is used for path planning, If lateral disturbance is applied to the UAV, then this method will plan an oscillation path, which is not conducive to actual flight. The solid line is the path planned by the Bug2 algorithm, and the curve segment $\widehat{q_2^L X_{goal}}$ is the path planned by the APF method. Obviously, the path $\widehat{q_1^H q_2^L \sim q_2^L X_{goal}}$ planned by Improved APF method is the best path.

This paper proposes an improved APF algorithm that combines the advantages of traditional APF and Bug2 algorithms. The specific obstacle avoidance process is shown in Table1.

Table1. Path planning process.

Algorithm: Improved APF method
Input: A point UAV with a tactile sensor Output: A path to goal point
while True do repeat from X_{point} move toward X_{goal} . until X_{goal} is reached or an obstacle is encountered. if X_{goal} is reached then

```

        exit
    end if
    if an obstacle is encountered then
        attempt to use APF method to avoid obstacle
        if step into a local optimal state or overtime then
            switch to Bug2 algorithm
        end if
    end if
end while
    
```

4 Obstacle avoidance method when the sensor fails briefly

In the normal flight of the UAV, the sensor can detect obstacles in advance, and make obstacle avoidance actions to re-plan the path after entering the obstacle's influence range. However, the sensor is likely to be affected by the external environment in actual engineering applications, the performance of the sensor will deteriorate or temporarily fail. The safest strategy at this time is to slow down or hover, but in order to increase the cruising distance, the UAV should be kept in motion state as long as possible.

This paper judges the sensor fault condition as: $\chi_B^{\max} < \rho$, where χ_B^{\max} is the maximum detection distance when the sensor fails, ρ is the radius of the obstacle's influence range. Generally, $\chi_G^{\max} > \rho_B$, χ_G^{\max} is the maximum detection distance when the sensor works normally.

The flight of a UAV can be regarded as the process of tracking a fixed or dynamic target. Of course, the UAV is in the most dangerous state when the sensor fails briefly when tracking a dynamic target. In this case, the position of the target point is updating, and the magnitude and direction of the target's attractive force on the UAV are changing. We can assume that the UAV's flight trajectory is a straight line in a short time and only consider the situation that the UAV is in the expanded influence range of the obstacle.

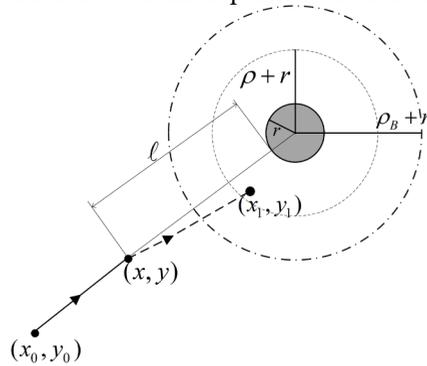


Fig. 2. Schematic diagram of UAVs movement near the obstacle.

Based on the data normally acquired by the sensor at the last moment, we can infer the relative position relationship between the obstacle and X_{po_int} , from which we can calculate the position of the obstacle in the current position coordinate system, we record it as (x', y') , which remains unchanged (see Fig.1). Before the sensor returns to normal, we can get the distance between the UAV and the obstacle by using the real-time position coordinates (x, y) of UAV and the coordinates (x', y') of the obstacle. From this distance, we can judge whether UAV is within the expanded influence range of the obstacle.

If we get

$$\ell - I \leq \rho_B \tag{5}$$

ℓ is the distance between the UAV and the obstacle detected at the last moment of the sensor failure; $I = v\Delta t + \frac{1}{2}a\Delta t^2$ is the distance the UAV moved forward in time Δt ; $\rho_B + r$ is the radius after the expansion of the obstacle's influence range, and r is the radius of the obstacle.

We believe that UAV has entered the influence range of obstacle, and need to re-plan a new path when inequality expression (5) holds. Now, the distance between the UAV and the obstacle in the repulsion function expression satisfies $\rho(q, q_{obs}) = \ell - I$, and keep $\rho(q, q_{obs})$ is changing before the sensor returns to the normal state,

$$\rho(q, q_{obs}) = \begin{cases} \rho_B * (1 - \frac{t}{2}) & 0 \leq t < 2 \\ \frac{\rho_B}{t} & t \geq 2 \end{cases} \tag{6}$$

Equation(6) shows the longer flight time in the influence range of the obstacle, the smaller $\rho(q, q_{obs})$ will be, thereby the repulsion will increase and "drives" the UAV out.

5 Simulation

We use simulation experiments to verify the effectiveness and feasibility of the algorithm proposed in this paper. The simulation environment is MATLAB (R2017b).

5.1 Simulation experiment with large obstacle

When encountering a large obstacle, if the UAV tracks a dynamic target point, it may get rid of the obstacle and move on, but this possibility is very small when tracking a fixed target point, so this paper only considers the situation that tracking fixed target point when encountering a large obstacle.

In the motion environment, MATLAB is used for three simulations: the APF method is used alone, the Bug2 algorithm is used alone, and Improved APF method to perform path planning simulation. The simulation parameters are shown in Table2 and Table3. And the starting point of the UAV is (5, 5), the end point is (80,100), there is a bar-shaped obstacle (black part) between the start point and the end point(see Fig.3).

Table 2. APF simulation parameters.

Parameter Name	Parameter Value
UAV step	10
Target gravity coefficient λ	1.2
Obstacle repulsion gain coefficient η	6.0

Table 3. Bug2 simulation parameters.

Parameter Name	Parameter Value
UAV step	10
Distance from obstacles	5.0/m

According to Fig.3(a), it can be seen that the APF method is used alone, the UAV cannot bypass the obstacle because the obstacle is "large" enough, even if a lateral

disturbance is applied. According to Fig.3(b), if we use the Bug2 algorithm alone, although UAV can eventually reach the target point, it has gone the extra distance and the path is not the best. According to Fig.3(c), the UAV flies along the wall after encountering a huge obstacle, when the obstacle is bypassed, the UAV finally uses the APF's potential field strategy to plan the best path and waste the shortest time.

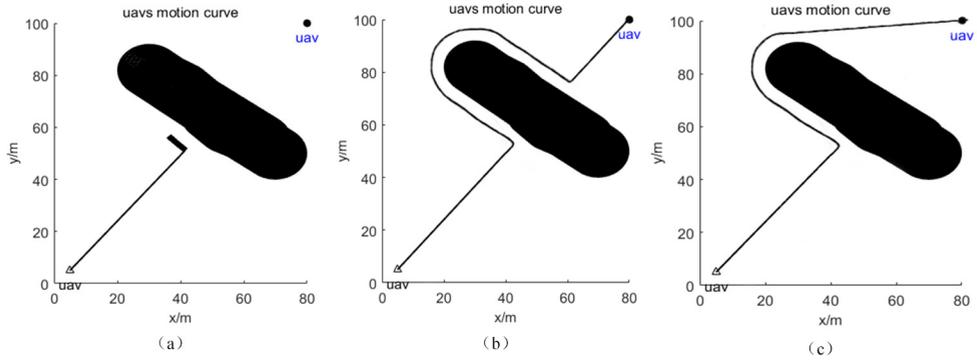


Fig. 3. Path planning comparison simulation diagram.

We counted the performance of various algorithms, as shown in Table4.

Table 4. Performance comparison of three path planning methods.

Performance	APF	Bug2	Improved APF
Average time	--	21.04	19.22
Shortest path length	--	191.802	184.361

5.2 Simulation experiment with fixed target point

The simulation parameters are shown in Table5.

Table 5. Fixed target point simulation parameter setting.

Parameter Name	Parameter Value
Normal maximum detection distance χ_G^{\max}	6.0/m
Abnormal maximum detection distance χ_B^{\max}	2.0/m
Obstacle radius r	1.0/m
Radius of obstacle action ρ	3.0/m
Radius of obstacle action after expansion ρ_B	4.0/m
target gravity coefficient λ	1.0
obstacle repulsion gain coefficient η	5.5
UAV step	10

We set when UAV is 4.5m away from the No.2 obstacle, the sensor fails and the sensor detection distance is shortened to 2m. From Fig.5(a), the shortest distance from UAV to the obstacle is 2.446m, the sensor cannot detect the obstacle, so UAV keeps flying straight and

does not avoid the obstacle. Therefore, the UAV is in a dangerous state during actual flight. This method is not desirable.

During adopting the improved obstacle avoidance method, the UAV normally flies at a distance of 4.5m from the No.2 obstacle, which is not yet in the expanded influence area, after advancing 0.5m, the UAV enters the expanded influence range of obstacle, and jumps within and outside the obstacle's influence area, which affected by gravitational and repulsive forces from target point. From Fig.5(b), the minimum distance between UAV and the obstacle is 3.418m, which is greater than the normal radius of the obstacle's influence range, the UAV is safe, and successfully avoids the obstacle, as shown in Fig.4(b).

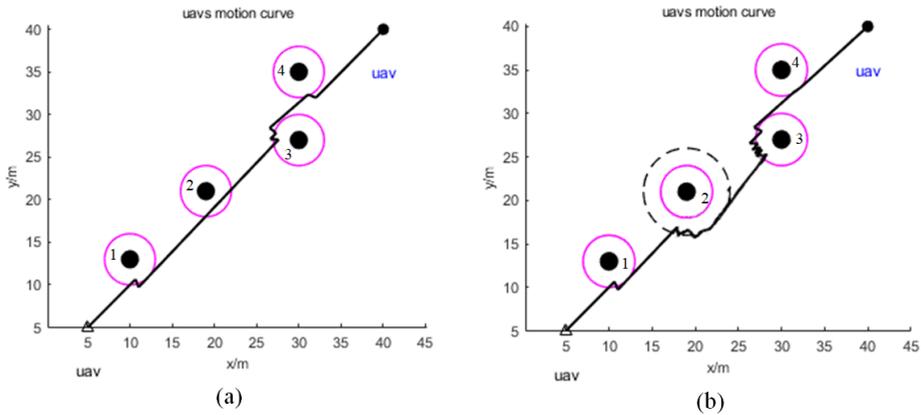


Fig. 4. Fixed target point simulation comparison diagram.

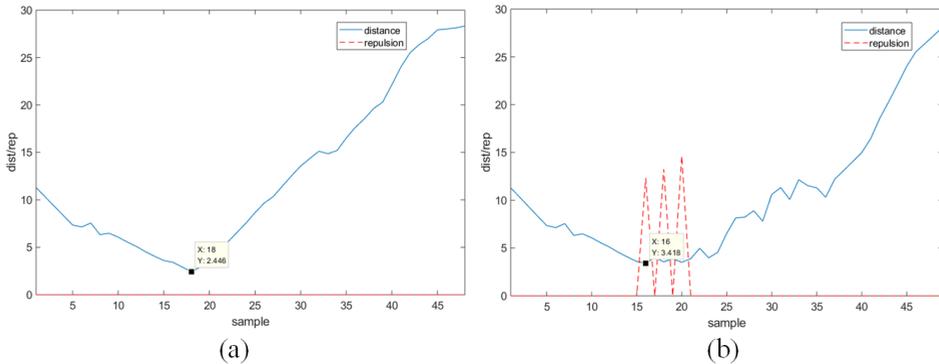


Fig. 5. Comparison diagram of repulsion and distance between UAV and No. 2 obstacle.

5.3 Simulation experiment with dynamic target point

The simulation parameters are the same as those in Table5. The UAV follows a fixed target in a linear motion before encountering the No.3 obstacle, and tracks the dynamic target near the No.3 obstacle.

We set when UAV is 4.5m away from the No.3 obstacle, the sensor fails and the sensor detection distance is shortened to 2m. As shown in Fig.7(a), the shortest distance between UAV and the obstacle is 2.325m, which exceeds the detection range of the sensor, so the obstacle avoidance is not triggered, but the UAV is in a dangerous state at this time. As

shown in Fig.7(b), after using the improved obstacle avoidance method, the UAV will be affected by the obstacle's repulsion after it enters the expanded influence area of the obstacle, and the repulsion will increase with time. At the same time, the position change of the target point will cause the change of magnitude and direction. From Fig6(b), the shortest distance between UAV and the obstacle is 3.563m, which is larger than the normal radius of the obstacle's influence range, and the UAV finally "escapes" the obstacle safely, as shown in Fig.6(b).

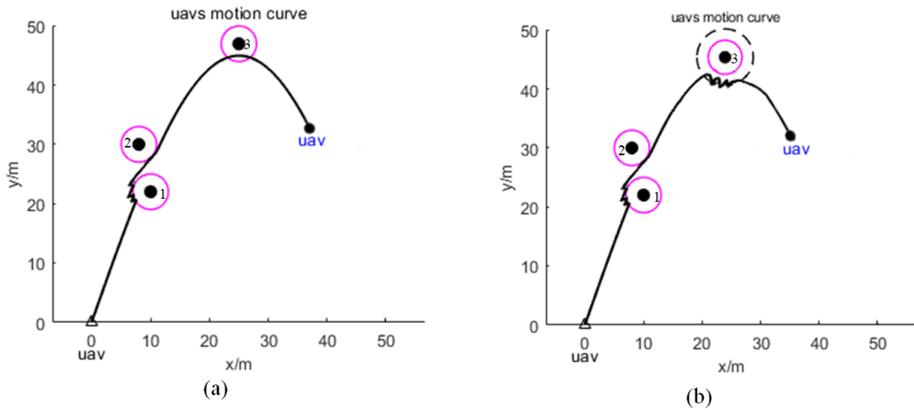


Fig. 6. Dynamic target point simulation comparison diagram.

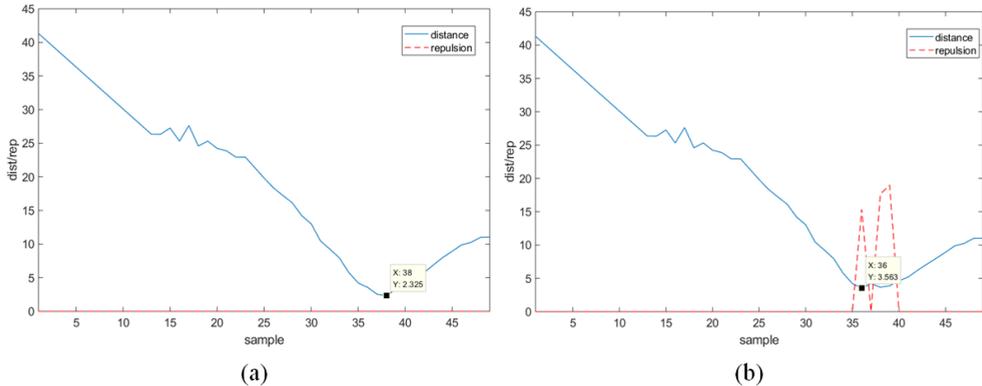


Fig. 7. Comparison diagram of repulsion and distance between UAV and No.3 obstacle.

6 Summary

This paper mainly studies the path planning of multi-rotor UAV when the sensor is affected by the environment or encountering a large obstacle when tracking fixed or dynamic target point. When encountering a large obstacle, the Bug2 algorithm has the feature of "walking along the wall" and it can reach the target point without being surrounded by obstacles. Therefore, the improved APF path planning algorithm is used to solve the problem that the target blocked by obstacles is not reachable. If the detection distance affected by the environment becomes shorter or the obstacle cannot be detected in time, we can expand the influence range of the obstacle, dynamically change the distance between the UAV and the obstacle in the repulsion function, and combine the prior distance to perform avoidance

actions in advance, which can effectively ensure the safety of the UAV. The validity and feasibility of the algorithm are verified by the above simulation experiment, which has important reference value for the promotion of practical engineering application.

This paper does not consider the obstacle avoidance problem of multi aircraft flight and the complex situation of more fixed obstacles and dynamic obstacles. In the future, it needs to verify the feasibility of the algorithm in the more complex situation.

References

1. Chen GU. Application of improved A* algorithm in robot path planning[J]. *Electronic Design Engineering*, **295**: 303-320(2014)
2. Zhan, Weiwei, Wang, Wei, Chen, Nengcheng. Path planning strategies for UAV based on improved A* algorithm[J]. *geomatics & information science of wuhan university*, **40**(3):315-320(2015)
3. Xu Z , Hu J , Ma Y , et al. A Study on Path Planning Algorithms of UAV Collision Avoidance[J]. *Xibei Gongye Daxue Xuebao/Journal of Northwestern Polytechnical University*, **37**(1):100-106(2019).
4. Zhang C . Path Planning for Robot based on Chaotic Artificial Potential Field Method[J]. *Science Technology & Engineering*, **317**(1):012056(2018).
5. Song J , Zhao M , Liu Y , et al. Multi-Rotor UAVs Path Planning Method based on Improved Artificial Potential Field Method[C]. *2019 Chinese Control Conference (CCC)*. (2019).
6. Dreoj J , Siarry P . Continuous interacting ant colony algorithm based on dense heterarchy[J]. *Future Generation Computer Systems*, **20**(5):p.841-856(2004).
7. Liao W H , Kao Y , Fan C M . Data aggregation in wireless sensor networks using ant colony algorithm[J]. *Journal of network and computer applications*, **31**(4):p.387-401(2008).
8. Cho D H , Jang D S , Choi H L . Heterogeneous, Multiple Depot Multi-UAV Path Planning for Remote Sensing Tasks[C]. *Aiaa Information Systems-aiaa Infotech @ Aerospace*. (2018).
9. Li H , Yang S X , Seto M L . Neural-Network-Based Path Planning for a Multirobot System With Moving Obstacles[J]. *IEEE Transactions on Systems Man & Cybernetics Part C*, **39**(4):410-419(2009).
10. Bhaduri A . A mobile robot path planning using Genetic Artificial Immune Network algorithm[C]. *World Congress on Nature & Biologically Inspired Computing. IEEE*, (2010).
11. K. Taylor and S. M. LaValle, "I-Bug: An intensity-based bug algorithm," *2009 IEEE International Conference on Robotics and Automation*, pp. 3981-3986(2009).