Hindawi
Mathematical Problems in Engineering
Volume 2021, Article ID 3798990, 9 pages
https://doi.org/10.1155/2021/3798990

Research Article

Research on an Obstacle Avoidance Method for UAV

Zhenqi He[1,2] and Lu Yao[3]

[1]*School of Artificial Intelligence, Xi'an Aeronautical Polytechnic Institute, Xi'an 710089, China*
[2]*UAV Intelligent Control Technology Innovation Team, Xi'an Aeronautical Polytechnic Institute, Xi'an 710089, China*
[3]*School of Aeronautical Manufacturing Engineering, Xi'an Aeronautical Polytechnic Institute, Xi'an 710089, China*

Correspondence should be addressed to Zhenqi He; 158936808@qq.com

Received 25 June 2021; Accepted 1 September 2021; Published 10 September 2021

Academic Editor: Fabio Bovenga

With the continuous development of UAV technology, UAV has been widely used in various industries. In the flight process of UAV, UAV often changes the given path because of obstacles (including static nonliving body and moving living body). According to the properties of obstacles and the characteristics of UAV, standard Kalman filter is used for nonmaneuvering targets, and sigma point Kalman filter is used for maneuvering targets. In the aspect of obstacle avoidance, the minimum search method is used to get the initial population of local programming. Then, the improved genetic algorithm is run. Combined with the predicted obstacle features, the local planning path can be obtained. Finally, the local planning path and global planning path are combined to generate the planning path with new obstacles. At the end of the paper, the obstacle avoidance strategies of static and moving obstacles are simulated. The simulation results show that this method has fast convergence speed and good feasibility and can flexibly deal with the obstacle avoidance and local path planning of various new obstacles.

1. Introduction

In recent years, with the continuous development of UAV and computer technology, UAV technology is widely used in power, agriculture, film, and other industries. In high temperature, cold, dangerous, and other environments [1], UAV technology can help people work. With the continuous expansion of the application field of UAV, the performance requirements of UAV are higher and higher, especially for the key technologies of UAV, such as endurance, shooting, and obstacle avoidance [1, 2].The obstacle avoidance technology of UAV is related to the safety of UAV, so it is a hot issue in the research of UAV.

At present, many research studies on UAV obstacle avoidance are to obtain the two-dimensional position information of UAV height through UAV airborne radar and construct an artificial potential field model to guide UAV to quickly track targets and avoid air obstacles. Similar to this obstacle avoidance method, it is more suitable for obstacle avoidance of stationary objects and does not predict dynamic objects [1, 2]. In the subsequent path planning, the optimal path cannot be obtained. In this study, the obstacles are classified, the standard Kalman filter prediction algorithm is used for nonmaneuvering targets, and the sigma point Kalman filter method is used for maneuvering targets. Better results can be obtained in path planning.

There are two common methods for obstacle avoidance planning of UAV [3]; one is to adjust the UAV flight speed to avoid obstacles according to the relative position and speed between UAV and target; the other is to change the UAV heading to avoid obstacles. Considering that the outdoor environment is mostly static obstacles, it is difficult to meet the obstacle avoidance requirements only by adjusting the UAV speed, so the second method is considered.

In this paper, according to the characteristics of obstacles, obstacles are divided into nonmaneuvering targets and maneuvering targets. The standard Kalman filter is used to predict the maneuvering target, and the sigma point Kalman filter is used to predict the maneuvering target. Through the study of these two parts, we can effectively predict the characteristics and location of obstacles. Then, for the predicted obstacles, the minimum search method is used to obtain the initial population of local programming and run the improved genetic algorithm. Combined with the predicted obstacle characteristics, the local planning path can be obtained. Finally, the planning path with new obstacles is generated by combining the local planning path and the global planning path. The simulation results show that this method has fast convergence speed and good feasibility and can flexibly deal with obstacle avoidance and local path planning of various new obstacles.

2. Decision of Obstacle Avoidance Algorithm

The sensors used for real-time obstacle avoidance in UAV are mainly lidar, and many scholars have done a lot of research on the use of lidar [1, 4]. However, the main starting point of these research results is to avoid obstacles, which is not fully based on the attributes of obstacles and the characteristics of UAV, so as to organically combine safety obstacle avoidance and path planning. In this paper, based on the information obtained by UAV, combined with the characteristics of obstacles, UAV, and preplanned global path, local planning path is generated near the global planning path on the basis of safe obstacle avoidance, so as to achieve the unity of global and local path planning [5].

The obstacle avoidance decision algorithm is shown in Figure 1. Target recognition is based on the information obtained by lidar. If obstacles are found beyond the safe distance, obstacle avoidance will not be initiated. On the contrary, obstacle avoidance is based on the results of target recognition.

3. Prediction of Nonmaneuvering Target and Maneuvering Target

3.1. Standard Kalman Filter Algorithm. The filtering process of the standard Kalman filter is calculated in a recursive way of "prediction-revision." First, the prediction value is calculated, and then, the observed value is modified according to the new information and Kalman gain (weighting term). The predicted value can be obtained from the filtered value, and the filtered value can be obtained from the predicted value. The interaction between the filtering and the prediction does not require any observation data to be stored, and it can be processed in real time. The system structure block diagram is shown in Figure 2.

Assume that the state equation and measurement equation of the system are, respectively,

$$\begin{aligned} X_k &= A_{k/k-1}X_{k-1} + B_k W_k, \\ Z_k &= C_k X_k + V_k, \end{aligned} \tag{1}$$

where X_k is the n-dimensional state vector at time k. That is to say, the estimated vector $A_{k/k-1}$ is $n * n$-dimensional state transition matrix. X_{k-1} is the n-dimensional state vector at time k-1. B_k is the dynamic noise matrix $n * n$. W_k is n-dimensional dynamic noise. Z_k is the observation vector at time K. C_k is k-time measurement matrix $n * n$. The m-dimensional observation noise at time V_k is k. And, dynamic noise W and observation noise K are uncorrelated zero mean white noise sequences. That is to say, for all k and j, $EW_k = 0, EV_k = 0$. Then,

$$\operatorname{Cov}\left(W_k, W_j\right) = EW_k {W_j}^{\mathrm{T}} = Q_k \delta_{kj}, \tag{2}$$

$$\operatorname{Cov}\left(V_k, V_j\right) = EV_k {V_j}^{\mathrm{T}} = R_k \delta_{kj}, \tag{3}$$

$$\operatorname{Cov}\left(W_k, V_j\right) = EW_k {V_j}^{\mathrm{T}} = 0, \tag{4}$$

where δ_{kj} is Kronecker symbolic function,

$$\delta_{kj} = \begin{cases} 0, & k \neq j, \\ 1, & k = j, \end{cases} \tag{5}$$

and Q_k is the variance of dynamic noise. In the Kalman filter, it is required to be a known nonnegative fixed array. R_k is the observation noise variance matrix. In the Kalman filter, it is required to be a known nonnegative fixed array. That is to say, there is noise in each measurement component.

Using the minimum variance to deduce the Kalman filter, we can obtain the following.

Forecast estimate $\overline{X}_{k/k-1}$ is

$$\overline{X}_{k/k-1} = A_{k/k-1}\overline{X}_{k-1}. \tag{6}$$

Filter estimation $\overline{X}_k$ is

$$\overline{X}_k = \overline{X}_{k/k-1} + K_k\left(Z_k - C_k \overline{X}_{k/k-1}\right), \tag{7}$$

where K_k is the best gain matrix.

Measured prediction $Z_{k/k-1}$ is

$$Z_{k/k-1} = C_k \overline{X}_{k/k-1}. \tag{8}$$

New information prediction ε_k is

$$\varepsilon_k = \overline{Z}_k - Z_{k/k-1}. \tag{9}$$

Prediction error covariance matrix $P_{k/k-1}$ is

$$P_{k/k-1} = A_{k/k-1}P_{k-1}A_{k/k-1}^{\mathrm{T}} + B_{k-1}Q_{k-1}B_{k-1}^{\mathrm{T}}. \tag{10}$$

Filtering error covariance matrix P_k is

$$P_k = \left(I - K_k C_k\right)P_{k/k-1}. \tag{11}$$

Optimal gain matrix K_k is

$$K_k = P_{k/k-1}C_k^{\mathrm{T}}\left(C_k P_{k/k-1} C_k^{\mathrm{T}} + R_k\right)^{-1}. \tag{12}$$

3.2. Prediction of Uniform Linear Motion with Standard Kalman. In this paper, the target mathematical model $\begin{cases} x = ut + x_0 \\ y = vt + y_0 \end{cases}$ of uniform linear motion is predicted, where u and v are the velocity component of the target in x and y coordinate axes, which are constant parameters. x_0 and y_0 are the displacement of the target at the initial time, and they are also constant parameters:

$$X_k = AX_{k-1} + BW_k, \tag{13}$$

where the state variable X_k is

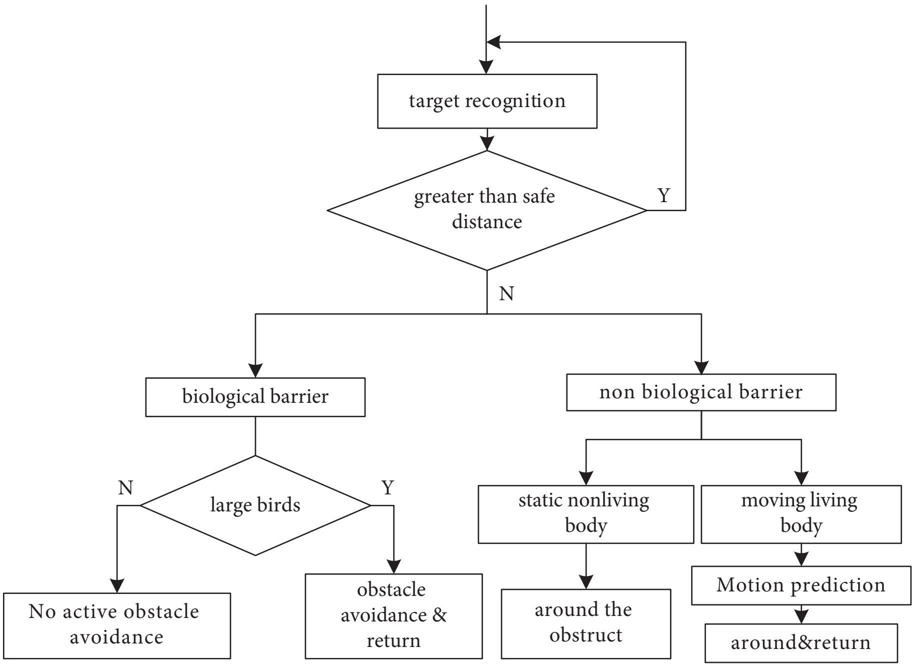

Figure 1: Structure diagram of UAV obstacle avoidance planning.

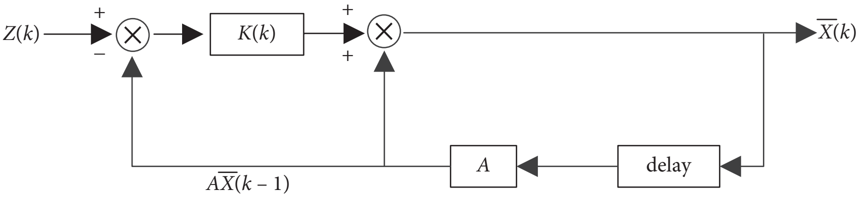

Figure 2: Standard Kalman filter system block diagram.

$$X_k = [\, x_k \;\; \hat{x} \;\; y_k \;\; \hat{y} \,]. \tag{14}$$

The state transition matrix A and dynamic noise matrix B are

$$A = \begin{bmatrix} I_2 & TI_2 \\ 0_2 & I_2 \end{bmatrix},$$
$$B = \begin{bmatrix} \left(\dfrac{T^2}{2}\right) I_2 \\ TI_2 \end{bmatrix}. \tag{15}$$

Sampling time $T = 1$ s. W_k is the mean value, its value is zero, and the dynamic variance is the dynamic noise of Q. The value of dynamic variance Q is as follows:

$$Q = \begin{bmatrix} 5 & 0 \\ 0 & 5 \end{bmatrix}. \tag{16}$$

The observation equation of the system is as follows:

$$Z_k = C_k X_k + V_k, \tag{17}$$

where C is the observation matrix,

$$C = [\, 1 \;\; 0 \;\; 1 \;\; 0 \,], \tag{18}$$

and V_k is the mean value, its value is 0, and the dynamic variance is the dynamic noise of R. The value of dynamic variance R is as follows:

$$R = \begin{bmatrix} 5 & 0 \\ 0 & 5 \end{bmatrix}. \tag{19}$$

The initial state X_0 and the initial covariance matrix P_0 are as follows:

$$X_0 = [\, 0 \;\; 40 \;\; 0 \;\; 20 \,]^{\mathrm{T}},$$
$$P_0 = \begin{bmatrix} 5 & 0 & 0 & 0 \\ 0 & 5 & 0 & 0 \\ 0 & 0 & 5 & 0 \\ 0 & 0 & 0 & 5 \end{bmatrix}. \tag{20}$$

The prediction results of nonmotorized obstacles using the standard Kalman are shown in Figures 3 and 4. The

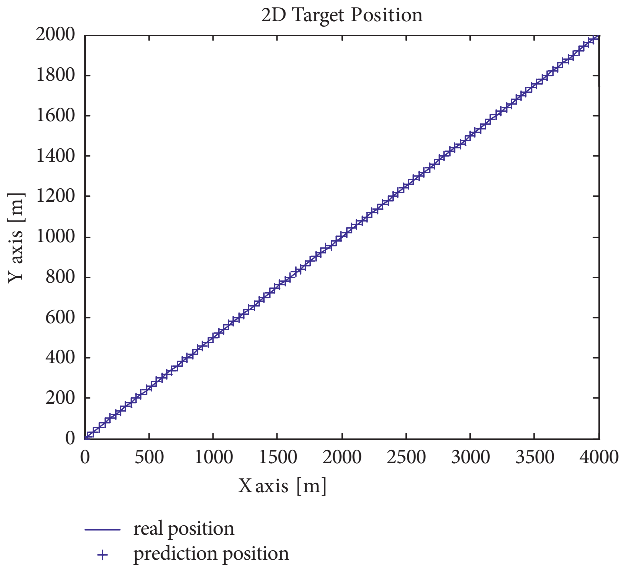

Figure 3: Actual and predicted positions.

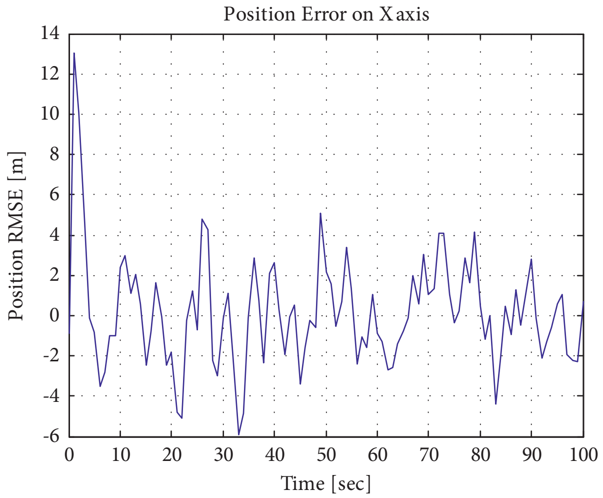

Figure 4: Prediction error.

simulation results show that the prediction effect is satisfactory.

The standard Kalman filter can achieve good results in predicting the state of nonmaneuvering target. For maneuvering target, its motion is unknown, so the standard Kalman filter cannot accurately describe its motion state. In this paper, the sigma point Kalman filter is used to realize the state estimation of the maneuvering target. Its core idea is unscented transformation, which propagates a finite number of sigma points by using nonlinear state and observation equations. Then, weighted sum is used to get the mean and variance of the state probability distribution.

3.3. Sigma Point Kalman Filtering Algorithm. UKF, the sigma point Kalman filter algorithm, is based on UT transformation and adopts the Kalman filter framework. The specific sampling form is deterministic sampling. The number of discrete points (called sigma points) sampled by UKF is small, and the specific number depends on the selected sampling strategy. The most commonly used is $2n+1$ sigma point symmetric sampling.

According to the state estimates $\hat{x}_{k-1}$ and P_{k-1} at time $k-1$, the realization process of obtaining the state estimates $\hat{x}_{k-1}$ and P_{k-1} at time k is given.

First, according to $\hat{x}_{k-1}$ and P_{k-1}, the sigma point χ^{i}_{k-1}, $i = 1, \ldots, 2n$, is constructed according to the sigma point sampling formulas (5)–(31):

$$\chi_i = \begin{cases} \overline{x}, & i = 0, \\ \overline{x} + \left(\sqrt{(n+\lambda)P}\right)_i, & i = 1, \ldots, n, \\ \overline{x} + \left(\sqrt{(n+\lambda)P}\right)_i & i = n+1, \ldots, 2n. \end{cases} \tag{21}$$

Then, predict the sigma point and mean:

$$\chi^{i}_{k|k-1} = f_k\left(\chi^{i}_{k-1}\right), \tag{22}$$

$$\hat{x} = \sum_{i=0}^{2n} W_i^{(m)} \chi^{i}_{k|k-1}, \tag{23}$$

$$P_{k|k-1} = \sum_{i=0}^{2n} W_i^{(c)} \left[\chi^{i}_{k|k-1} - \hat{x}_{k|k-1}\right]\left[\chi^{i}_{k|k-1} - \hat{x}_{k|k-1}\right]^{\mathrm{T}} + Q_{w,k}, \tag{24}$$

Correction (measurement update):

$$Y^i_{k|k-1} = g\left(\chi^i_{k-1}\right), \tag{25}$$

$$\hat{y} = \sum_{i=0}^{2n} W_i^{(m)} Y^i_{k|k-1}, \tag{26}$$

$$P_{y,k} = \sum_{i=0}^{2n} W_i^{(c)} \left[Y^i_{k|k-1} - \hat{y}_{k|k-1}\right]\left[Y^i_{k|k-1} - \hat{y}_{k|k-1}\right]^{\mathrm{T}} + R_{v,k}, \tag{27}$$

$$P_{xy,k} = \sum_{i=0}^{2n} W_i^{(c)} \left[\chi^i_{k|k-1} - \hat{x}_{k|k-1}\right]\left[Y^i_{k|k-1} - \hat{y}_{k|k-1}\right]^{\mathrm{T}}, \tag{28}$$

$$K_k = P_{xy,k} P^{-1}_{y,k}, \tag{29}$$

$$\hat{x}_k = \hat{x}_{k|k-1} + K_k\left(y_k - \hat{y}_{k|k-1}\right), \tag{30}$$

$$P_k = P_{k|k-1} - K_k P_{y,k} K_k^{\mathrm{T}}. \tag{31}$$

3.4. Maneuvering Target Prediction Simulation. In order to verify the prediction of the movement of obstacles with mobility by the sigma point Kalman filter method, this paper constructs the state equation and observation equation of obstacle movement:

$$\begin{aligned} x &= 0.2x + 0.01x^2 + 8\cos(1.2 * (k+1)) + \mathrm{sqrt}(10) * \mathrm{randn}, \\ y &= x^2 + 0.1\mathrm{randn}. \end{aligned} \tag{32}$$

Take the initial state of x as $x = 1$, the estimated value of the initial state is $x = 0.5$, the process state covariance Qis $Q = 10$, the measurement noise covariance R is $R = 0.01$, and the initial estimated variance P is $P = 1000$. The prediction of maneuverable obstacle motion using the sigma point Kalman filter is shown in Figures 5 and 6. From the simulation results, it can be seen that the method has achieved good prediction effect on the mobility movement.

4. Obstacle Avoidance Operator in Local Planning Based on the Minimal Search Method

For local path planning, fast obstacle avoidance is the primary issue under the condition of safety assurance, so it is necessary to consider how to improve the planning speed. Perform local planning between the abovementioned starting point and ending point. First of all, it is necessary to find the initial population that can be used for fast local planning, so this paper designs an obstacle avoidance operator.

Obstacle avoidance operator: by performing obstacle avoidance operations on the starting point and ending point of the path through the obstacle, a new local initial path can be obtained. Then, the local path is planned through genetic algorithm and then returned to the originally planned path. If the new obstacle is a triangle ABC, extrapolate the intersection of the global planning path and the obstacle to a safe distance and get the starting and ending points of the local path as P and Q. As shown in Figure 7. The ABC three-point coordinates can be substituted into PQ line equation, and the point with one end (such as point a) can be determined by symbols, and R point is generated near point a, where r-point coordinate is one of $(x_a + \Delta x,\ y_a + \Delta y)$, $(x_a - \Delta x,\ y_a + \Delta y)$, $(x_a + \Delta x,\ y_a - \Delta y)$, and $(x_a - \Delta x,\ y_a - \Delta y)$. If neither PR line nor QR line segment intersects with triangle ABC, the r-point coordinate is accepted as a point in the path. Such a problem can be reduced to the solution of the minimum value. The most common algorithm is the steepest descent method. The steepest descent method searches according to the inverse direction of the gradient of the multivariate function. It still needs to find the partial derivative and select a step size. The convergence rate

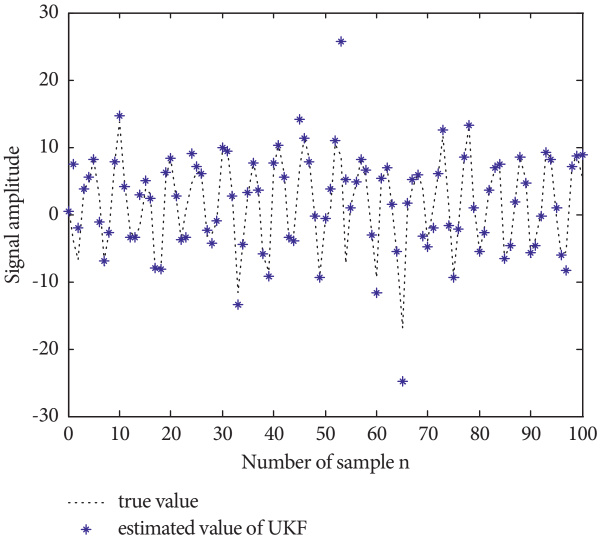

Figure 5: The true value and UKF estimated value.

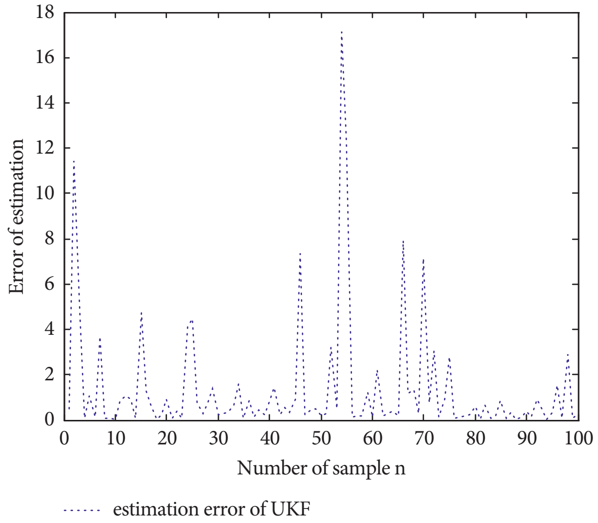

Figure 6: UKF estimation error.

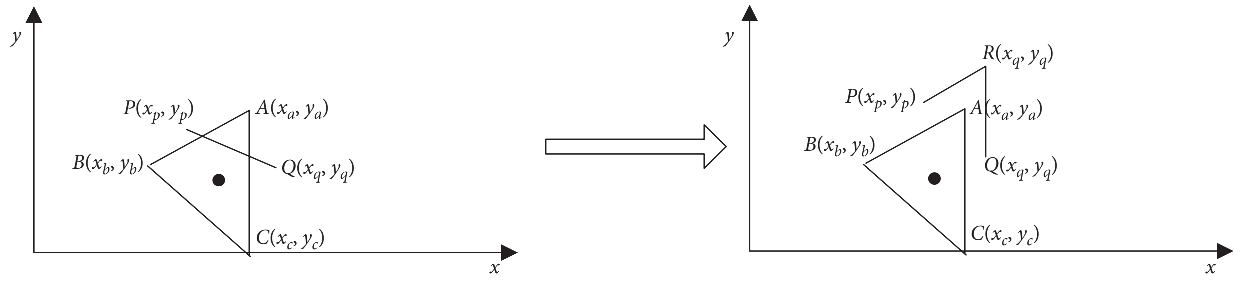

Figure 7: Diagram of the obstacle avoidance operator.

of this method is linear and becomes very slow after the first few steps.

In this paper, a minimum search method is proposed to find the minimum extremum of R point. The minimum search method is a simplified model, which can solve the problem of n-dimensional space in practice. In order to illustrate the problem conveniently, take two-dimensional space as an example.

In this paper, the symmetry axis of conic is used to estimate the minimum extremum value of R point. First of all, the problem can be attributed to a function of one variable $f(x)$. We consider the minimal value problem of $f(x_a - \Delta x)$, $f(x_a)$, and $f(x_a + \Delta x)$ on a diagonal. Since these three points can determine a conic curve, the minimum value R point can be estimated by using the known $f(x_a - \Delta x)$, $f(x_a)$, and $f(x_a + \Delta x)$.

Suppose the quadratic function is $h(x) = ax^2 + bx + c$, and it coincides with $f(x)$ at three known points; according to the undetermined coefficient method, the equations are obtained:

$$\begin{cases} f(x_a + \Delta x) = a(x_a - \Delta x)^2 + b(x_a - \Delta x) + c, \\ f(x_a) = a{x_a}^2 + bx_a + c, \\ f(x_a + \Delta x) = a(x_a + \Delta x)^2 + b(x_a + \Delta x) + c. \end{cases} \tag{33}$$

For the symmetric axis of $h(x)$ at $-b/(2a)$, the equation system can be solved:

$$-\frac{b}{2a} = x_a + \frac{f(x_a - \Delta x) - f(x_a + \Delta x)}{2[f(x_a - \Delta x) - 2f(x_a) + f(x_a + \Delta x)]}\Delta x. \tag{34}$$

Because the axis of symmetry of $h(x)$ is near x_a and Δx is small enough, when $f(x_0 - \Delta x) - 2f(x_0) + f(x_0 + \Delta x) > 0$, $h(x)$ is a concave function and the minimum value is taken at $-b/(2a)$.

From equation (34), the estimation point R of the ultimate minimum value can be obtained. The search process of R point is shown in Figure 7.

The P and Q points are reserved as the starting and ending points, and the initial population is generated near the PRQ so that the local optimal path can be quickly searched.

5. Local Planning Path Simulation

As mentioned above, the global planning path is shown in Figure 8(a). In the simulation, the obstacles are represented by multilateral type and segmented by triangles. The starting point of the path is (0, 0), and the target point of the path is (20, 0). If new obstacles appear on the planned path, the type of new obstacles needs to be determined through target identification. Here, we mainly discuss the case that the new obstacle is an inanimate body.

Firstly, the obstacle avoidance operator subroutine is run to obtain the initial population of local planning. Then, run the improved genetic algorithm to get the local planning path. Finally, the local planning path is combined with the global planning path to generate the planning path with new obstacles. The simulation is programmed in Matlab platform, and genetic operation is realized by genetic algorithm toolbox function of Sheffield University. The initial cross rate was 0.9, the initial variation rate was 0.2, and the genetic algebra was 1000. The simulated annealing process is set to 100 times and the simulated annealing temperature is set to 1000 by using binary number coding.

5.1. The New Obstacles Are Inanimate Objects at Rest.

If the above targets are identified, the new obstacles are inanimate objects at rest (mountains, buildings, etc.), as shown in Figure 8(b). The obstacle coordinates are (11.3, −3.1), (14.2, −2), and (12.1, −0.7). The wind speed is $v_{\text{wind}y} = 0.1\,\text{m/s}$.

Figure 8 shows the combination of local path planning and global path planning on the basis of global path planning. As shown in Figure 8(a), if new obstacles are found on the planned path, local planning is carried out by the improved genetic algorithm mentioned above. Combining with the original optimal path, the path planning results of the combination of local path planning and global path planning are obtained. Most of the paths maintain the global path planning results. The new obstacle part is local path planning, as shown in Figure 8(b). The path points include (0, 0), (4, −1.8), (10.2, −1.4), (12.2, −0.7), (14.2, −1.2) (16.1, −1), and (20, 0). As can be seen from Figure 8(b), the UAV bypasses the obstacle in the downstream direction. Figure 9 shows the error curve of local programming. The genetic algebra converges at about 200. The simulation results show that the convergence speed of local programming is fast.

If there are large static inanimate obstacles on the original global planning path, as shown in Figure 10(a), the coordinates of the new obstacles are (4.8, −0.7), (7.1, −3.2), and (10.2, −2), and the wind speed is $v_{wy} = 0.1\,\text{m/s}$. The simulation results are shown in Figure 10(a). The

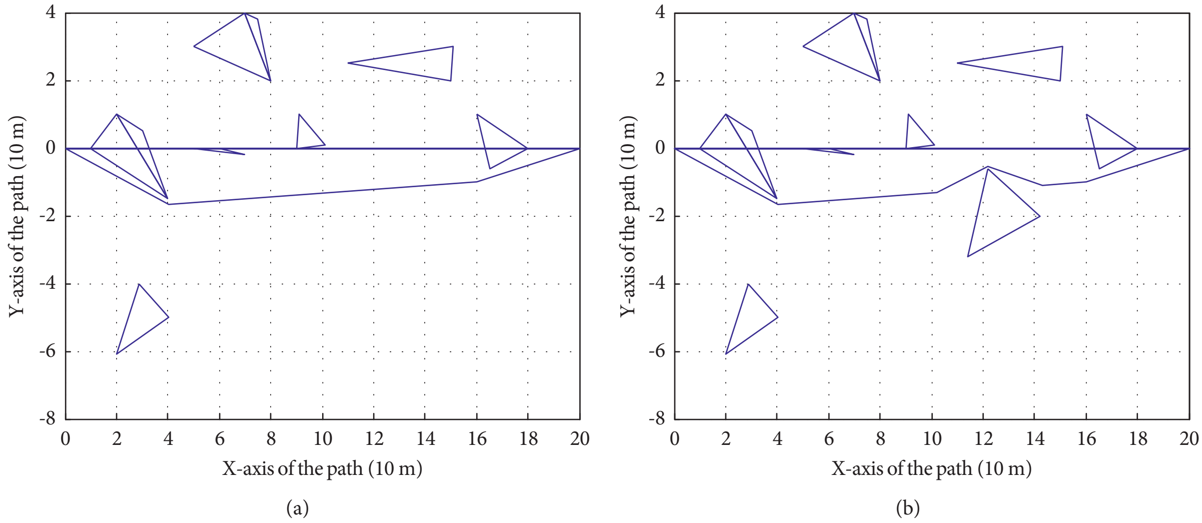

Figure 8: Path planning combining local path planning with global path planning. (a) Global path planning. (b) Local path planning based on global planning.

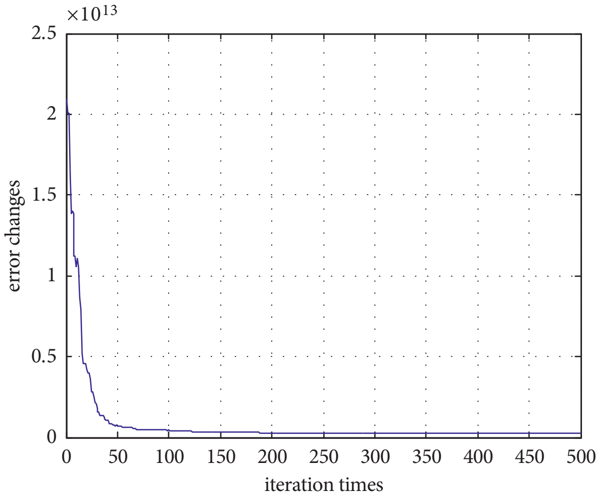

Figure 9: Local path planning error curve.

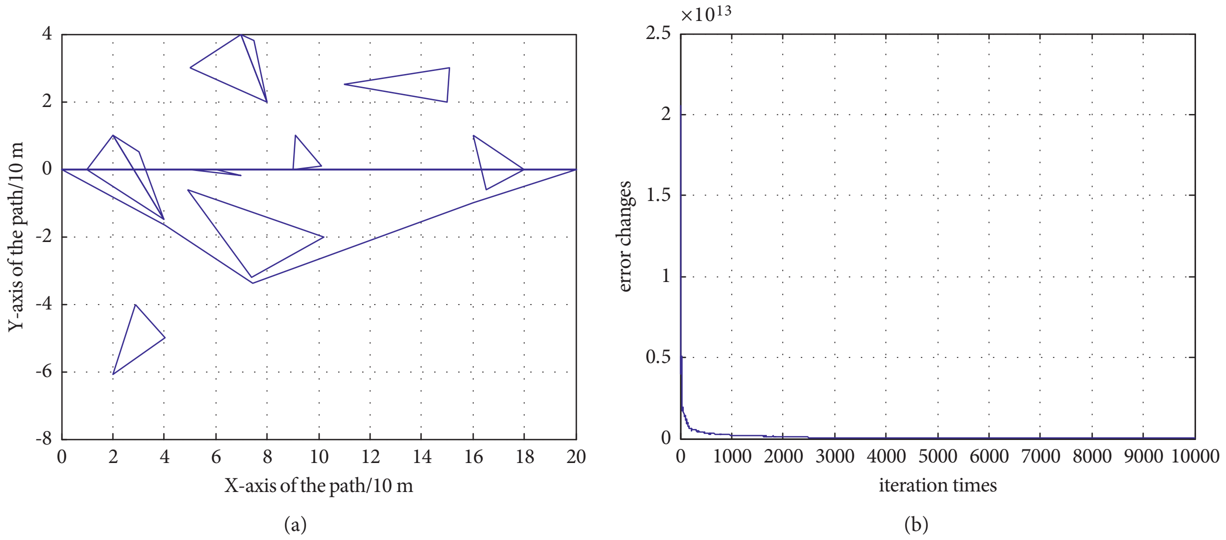

Figure 10: Path planning combining local path planning with global path planning for large obstacles. (a) Local path planning based on global planning; (b) local path planning error curve.

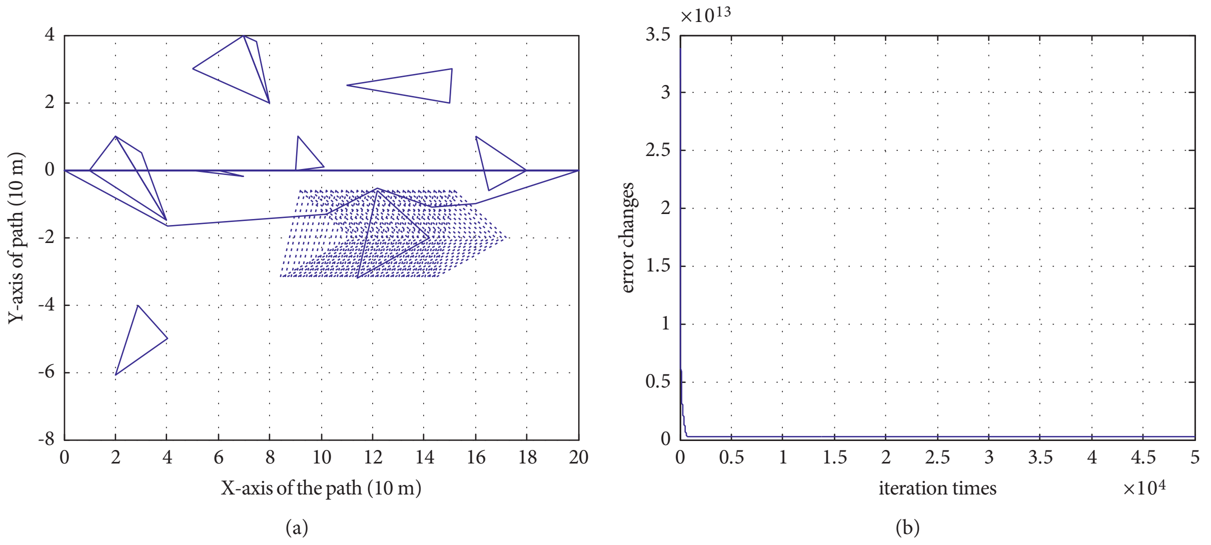

Figure 11: Path planning of uniform moving obstacles with local and global combination. (a) Local path planning based on global planning; (b) local path planning error curve.

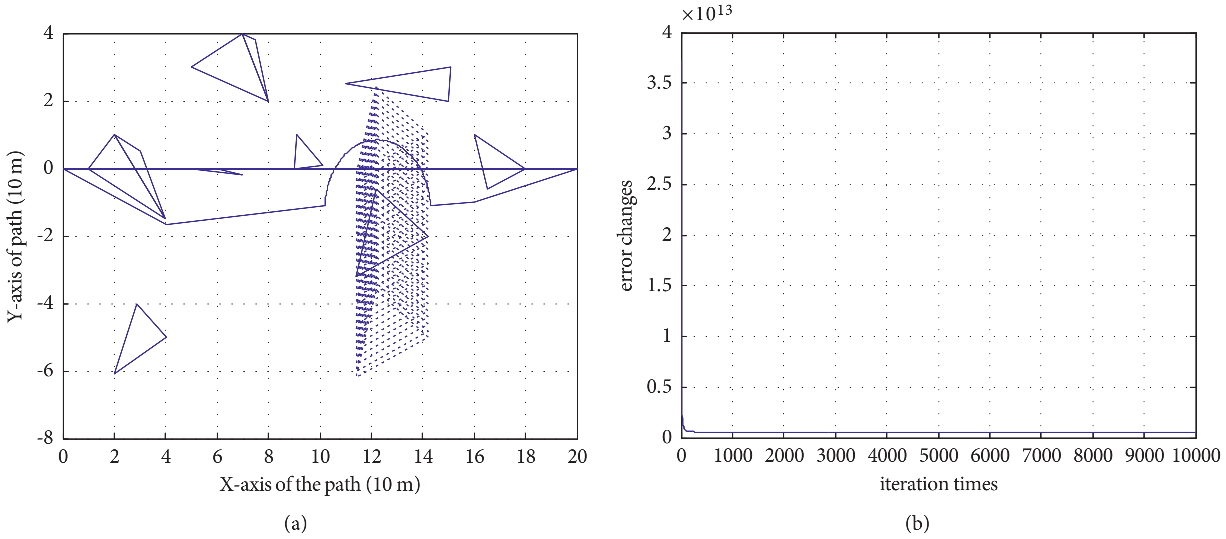

Figure 12: Path planning of local and global combination for accelerating moving obstacles. (a) Local path planning based on global planning; (b) local path planning error curve.

combination path of global and local planning is (0, 0), (4, −1.8) (7.2, −3.4), (16.1, −1), and (20, 0).

From the simulation results, the local path part is not planned along the wind speed direction because if it is planned along the wind speed direction, the UAV will enter the obstacle dense area, so it is difficult to realize the local path planning. Figure 10(b) shows the error curve of local programming. The genetic algebra converges about 600 times. The simulation results show that the convergence speed of local programming is fast.

5.2. The Obstacle Is a Moving Nonliving Body (Aircraft). On the path of global path planning shown in Figure 7(a), after target recognition, new obstacles of moving inanimate body appear.

If the new obstacle moves at a constant speed, as shown by the dotted line in Figure 11(a), the new obstacle is represented as a triangle. Among the new obstacles, $v_x = -0.2\,\mathrm{m/s}$ and $v_y = 0$. The wind speed is $v_{\mathrm{wind}y} = 0.1\,\mathrm{m/s}$. The motion of the obstacle can be estimated by the Kalman filter.

In Figure 11(a), due to the low speed of the obstacle and the small cross-sectional area of the obstacle, the local planning path completes the local motion planning along the short direction of the obstacle detour path and returns to the global planning path after the local path planning.

Figure 11(b) shows the error curve of local programming, and the genetic algebra converges around 150. The simulation results show that the local programming converges fast.

If the new obstacles are uniformly accelerated, as shown by the dotted line in Figure 12(a), the new obstacles are represented as triangles. Among new obstacles, $v_x = 0$, $v_y = 0.2\mathrm{m/s}$, and $a_{y0} = 0.2\,\mathrm{m/s^2}$. The wind speed is $v_{\mathrm{wind}y} = 0.1\,\mathrm{m/s}$. The motion of the obstacle can be estimated by the Kalman filter.

It can be seen from the simulation results in Figure 12(a) (the planned path is represented by a thick line) that the local planned path is divided into two parts: bypassing obstacles and returning. Due to the acceleration of the new obstacles, the part around the obstacles and the return part are both curves. The part around the obstacles starts the obstacle avoidance planning in advance to ensure the safety. The return part quickly returns to the global planning path, achieving the unity of local planning and global planning. Figure 12(b) shows the error curve of local planning. The genetic algebra converges around 400. The simulation results show that the convergence speed of local planning is fast.

6. Conclusions

When UAV flies on the planned path, new obstacles appear on the planned path. In different cases, the local path planning method will be determined according to the UAV obstacle avoidance decision. On the basis of obstacle target recognition, such as fully considering the type characteristics of obstacles, the influence of wind speed, and the influence of yaw angle change rate, the Kalman filter, prediction, and its improved algorithm are used to estimate the target motion parameters [6–8]. The local approximate obstacle avoidance path is planned by the obstacle avoidance operator of minimum search method, and the local initial population is generated. On this basis, the improved genetic algorithm is used to plan the local obstacle avoidance path. The starting point and ending point of local planning are all on the global planning path, which achieves the unity of global path planning and local path planning from the method and implementation. The simulation results show that the method has fast convergence speed and good feasibility and can flexibly deal with the obstacle avoidance and local path planning of various new obstacles.

Data Availability

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

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

The authors acknowledge the National Natural Science Foundation of China (Grant 61502391), Yulin Science and Technology Project in China (Grant 2019-116-13), 13th Five-Year Plan of Shaanxi Province in 2020 (Grant GH20Y649), School Level Scientific Research Projects (Grant 33/115040393), and UAV Intelligent Control Technology Innovation Team (Grant 33/115040441).

References

[1] F. Huang, P. Wu, and X. Li, “Distributed flocking control of quad-rotor UAVs with obstacle avoidance under the parallel-triggered scheme,” *International Journal of Control, Automation and Systems*, vol. 19, no. 3, pp. 1375–1383, 2021.

[2] J. Sandino, F. Vanegas, and F. Maire, “UAV framework for autonomous onboard navigation and people/object detection in cluttered indoor environments,” *Remaote sensing*, vol. 20, no. 12, pp. 1–31, 2020.

[3] S. Back, G. Cho, and J. Oh, “Autonomous UAV trail navigation with obstacle avoidance using deep neural networks,” *IEEE robotics and automation letters*, vol. 100, no. 3, pp. 1195–1211, 2020.

[4] B. Lindqvist, S. S. Mansouri, S. Agha-mohammadi, and J. Ali-akbar, “Nonlinear MPC for collision avoidance and control of UAVs with dynamic obstacles,” *IEEE robotics and automation letters*, vol. 5, no. 4, pp. 6001–6008, 2021.

[5] L. Salt, D. Howard, G. Indiveri, and Y. Sandamirskaya, “Parameter optimization and learning in a spiking neural network for UAV obstacle avoidance targeting neuromorphic processors,” *IEEE Transactions On Neural Networks And Learning Systems*, vol. 31, no. 9, pp. 3305–3318, 2020.

[6] X. Dai, Y. Mao, T. Huang, N. Qin, D. Huang, and Y. Li, “Automatic obstacle avoidance of quadrotor UAV via CNN-based learning,” *Neurocomputing*, vol. 402, no. 8, pp. 346–358, 2020.

[7] M. Lv, W. Yu, J. Cao, and S. Baldi, “A separation-based methodology to consensus tracking of switched high-order nonlinear multiagent systems,” *IEEE transactions on neural networks and learning systems*, vol. 78, pp. 1–13, 2021.

[8] M. Lv, W. Yu, J. Cao, and S. Baldi, “Consensus in high-power multiagent systems with mixed unknown control directions via hybrid nussbaum-based control,” *IEEE Transactions on Cybernetics*, vol. 99, pp. 1–13, 2020.