Real-time Path Planning Strategy for UAV Based on Improved Particle Swarm Optimization

Ze Cheng
School of Electrical Engineering and Automation, Tianjin University, Tianjin, China
Email: chengze@tju.edu.cn

Ergang Wang, Yixin Tang and Yucui Wang
School of Electrical Engineering and Automation, Tianjin University, Tianjin, China
Email: ergangtju@163.com

***Abstract*—Unmanned Aerial Vehicle (UAV) path planning is divided into off-line static path planning and real-time dynamic path planning. The former one is applied to the ideal situation that the terrain has been clear, and there is no unexpected situation in flight. Actually, however, the flight situation is very complex, we have to adopt real-time path planning based on off-line static path planning. To meet the demand of real-time dynamic path planning, this paper proposes a dynamic path planning strategy of adaptive chaotic particle swarm optimization (PSO) algorithm, which owns both good global and local search ability. The simulation shows that the path planning strategy this paper proposes, basically, meets the needs of real-time path planning. Moreover, it has better performance than other algorithm.**

***Index Terms*—Unmanned Aerial Vehicle (UAV), real-time path planning, improved PSO.**

I. INTRODUCTION

The primary task of UAV path planning is off-line static path planning, which serves as a reference trajectory tracking control to make UAV fly based on the flight path planned before. During the flight, the airborne sensor surveys obstacles and threats, it will start real-time dynamic track re-planning once found new obstacles or moving threats. It is a difficult point in the research of UAV path planning that how to effectively do a track re-planning to avoid danger during the flight. At present, as a kind of outstanding optimization algorithm, the standard PSO is applied to the research of UAV path planning with a good effect. Against shortcomings of PSO that it has a low convergence speed and it is likely to fall into the local optimum, many researchers have put forward improved measures. Reference [1] put forward the method of inertia weight. The inertia weight ω is a scale factor relevant to the previous velocity, and the updating equation of the velocity is as follows:

$$v_{id}^{k+1} = wv_{id}^{k} + c_1 rand_1^k \left(pbest_{id}^k - x_{id}^k \right) + c_2 rand_2^k \left(gbest_d^k - x_{id}^k \right) \tag{1}$$

The larger ω can strengthen the ability of global search, while the smaller ω can strengthen the ability of local search. The value of ω in the standard PSO can be seen as 1, as a result, it has a lack of local search ability in the late iteration. The experimental results show that PSO has a higher convergence speed when the value of ω is during the interval of [0.8, 1.2]. Reference [2] set the value of ω to decrease linearly from 0.9 to 0.4, making PSO explore in a larger area at the beginning and rapidly locate the approximate location of the optimal solution, as the value of ω gradually decreases, particles slow down to start the local search. This method accelerates the convergence rate and improves the performance of PSO algorithm. However, when the problem to be solved is complex, this method makes the PSO lack of global search ability in the late iteration, and it fails to find the optimal solution required.

More and more researchers have proposed other improvements for PSO algorithm, which are mainly as follows, Reference [3] proposed to change inertia weight and learning factor, reference [4] added chaos perturbation to PSO to improve the activity of particles, reference [5] proposed a kind of improved PSO based on the principle of avoiding disadvantages, reference [6] proposed a dynamic PSO with double variable subgroups. To cope with problems above, this paper proposes a hybrid PSO algorithm, which joins an adaptive strategy and a theory of chaos optimization search to increase the particle diversity, and it is combined with a variable structure optimization search theory.

To cope with the problem of suddenly moving obstacles in the real-time UAV path planning, we should predict the obstacles' movement path, based on which UAV can start real-time path regenerative planning. Where, the real-time path planning is divided into three steps as follows, data acquisition and data fusion, reconstruction of maps of flight environment, path re-planning. Kalman filtering algorithm is used to forecast the track of moving obstacles [7]. Through all above, it achieves real-time dynamic track re-planning of UAV online.

II. PATH PLANNING ALGORITHM

doi:10.4304/jcp.9.1.209-214

A. Improved PSO Algorithm

PSO algorithm is a kind of evolution computing technology put forward by *Kennedy* and *Eberhart* based on swarm intelligence in 1995, which is inspired by birds, fish swarm, and the pattern of the human society [8].

The current position of particle i is $x_i=(x_{i1}, x_{i2}, ..., x_{iD})$, and the current velocity is $v_i=(v_{i1}, v_{i2}, ..., v_{iD})$; the historical optimal position of particle i is $p_i=(p_{i1}, p_{i2}, ..., p_{iD})$, the historical optimal position of all particles is $p_g=(p_{g1}, p_{g2}, ..., p_{gD})$. Each particle flies in the solution space and updates its own position and velocity according to the optimal value of itself and groups, the specific updating formula is as follows:

$$v_{id}^{k+1} = wv_{id}^k + c_1 rand_1^k \left(pbest_{id}^k - x_{id}^k \right) + c_2 rand_2^k \left(gbest_d^k - x_{id}^k \right)$$
$$x_{id}(t+1) = x_{id}(t) + v_{id}(t+1) \qquad (2)$$

Although PSO algorithm owns such a powerful ability of optimization, it is likely to fall into local optimum and slow down the speed of convergence in the late iteration. In this paper, the adaptive strategy and chaos optimization search theory is added to the standard PSO algorithm.

The adaptive strategy is a method to dynamically adjust the inertia weight factor ω and the learning factor c_1 and c_2 according to the change of the search process. As is mentioned above, ω is the key parameter in PSO, which could adjust the local and global search ability of algorithm. Usually, we expect particles to have strong global search ability in the early evolutionary search, while strong local search ability in the late evolutionary search. In order to meet the above requirements, this paper adopts adaptive strategy to dynamically adjust the inertia weight factor as follows:

$$\omega = \omega_{min} + (\omega_{max} - \omega_{min}) \times \exp[-\lambda \times (\frac{t}{t_{max}})^2] \qquad (3)$$

Where, t is the current iteration times, t_{max} is the maximum iteration times, ω_{max} and ω_{min}, respectively, are the given maximum and minimum value of the inertia weight factor, $\lambda= 3$ is the factor to control the degree of change smoothing.

In the standard PSO, c_1 is a self-learning factor and c_2 is a social-learning factor, both of which are referred to set as 2. In the early evolutionary search, we require particles have strong self-learning ability and weak social-learning ability, which makes particles fly in the entire search space; while in the late evolutionary search, we require particles have weak self-learning ability and strong social-learning ability, which makes particles fly towards the global optimal solution. In order to get better search performance, the dynamic adjustment strategy of c_1 and c_2 should be as follows:

$$\begin{cases} c_1 = (c_{1\min} - c_{1\max}) \times \dfrac{t}{t_{\max}} + c_{1\max} \\ c_2 = (c_{2\max} - c_{2\min}) \times \dfrac{t}{t_{\max}} + c_{2\min} \end{cases} \qquad (4)$$

Where, c_{1max} and c_{1min}, respectively, are the maximum and the minimum of c_1, c_{2max} and c_{2min}, respectively, are the maximum and the minimum of c_2.

By means of dynamically adjusting the inertia weight factor ω and the learning factor c_1 and c_2, the global and local search ability of PSO algorithm is greatly improved. However, the problem that it is likely to fall into local optimum still exists, which is due to the lack of diversity of the particle swarm. This paper proposes to add chaos optimization search theory to the standard PSO algorithm, which adds chaos perturbation to particles to increase the diversity of the swarm. Chaos as a non-linear phenomenon is very common in the nature, the messy state shown in the process of change on the surface does not disguise its inherent regularity, precisely because of the randomness, ergodicity and regularity of the chaotic system, in which we can optimize the search. Logistic map is a typical chaotic system as follows:

$$z_{n+1} = \mu z_n (1 - z_n), n = 0,1,2,\cdots \qquad (5)$$

Where, μ is a control variable, the system is totally at chaotic state when $\mu=4$ and $0 \le z_0 \le 1$.

All above are improvements for the weakness of the standard PSO algorithm in the process of search. However, there is still a problem of the "curse of dimensionality" in the PSO algorithm, that is, the performance of the algorithm drops sharply with the increase in the number of problem solution space dimension.

Considering the characteristics of the PSO algorithm and the needs of track planning, an optimization search theory of variable structure is proposed in this paper to alleviate the curse of dimensionality. The core idea of this optimization search theory is as follows: the entire track line can be searched in sub-block at first, and then we can form a complete track line by combining those blocks. This theory exactly matches the idea of reducing the dimension in the high-dimensional particle optimization search, and it can effectively forbid the PSO algorithm to fall into the problem of the "curse of dimensionality".

By reducing the dimension of the particle optimization search, it can effectively avoid the "curse of dimensionality". However, in order to get the complete optimization search track line, they should share the information of the optimization search. As a result, this paper proposes an information sharing structure, chain structure.

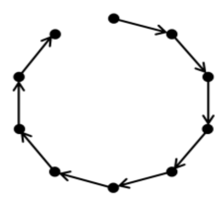

Figure 1. Chain tructure.

As is shown in the Fig.1, we decompose the high-dimensional particle into the low-dimensional minimum search particles which are equal to each other. We adopt a recursive mode. For the reason that the order of the decomposition of high-dimensional particle is according to the order of track directions movement, the search of decomposed low-dimensional particles can also be in order of track forward. The optimal solution searched by each low-dimensional particle will impact the search of the next adjacent particle. Finally, we will achieve the optimal solution for the entire track. For the reason that it costs little time for each search, this search method with chain structure allows real-time path planning.

B. Forecast of the Track of the Obstacle

Based on kalman filter [9] and predict the position of moving targets, UAVs realized online local track replanning in dynamic environments.

Concrete implementation method of Kalman filter is as follows. At first , process model of the system is used to predict the next state of the system, according to the state equation of the system, as shown in Equation (6), that is to say, we use the previous state of the system to predict the current state:

$$X(k \mid k-1) = AX(k-1 \mid k-1) + BU(k) \tag{6}$$

Where $X(k|k\text{-}1)$ is the predicted results using a previous state, $X(k\text{-}1|k\text{-}1)$ is the optimal result of a previous state, $U(k)$ is the control variable of the current state, A and B are system parameters. In the radar systems of detecting moving target, objects states in the *X-axis* and *Y-axis* direction can be expressed as

$$\begin{cases} X(k \mid k-1) = X(k-1 \mid k-1) + TV_x(k-1 \mid k-1) + (T^2/2)a_x(k-1 \mid k-1) \\ V_x(k \mid k-1) = V_x(k-1 \mid k-1) + Ta_x(k-1 \mid k-1) \\ Y(k \mid k-1) = Y(k-1 \mid k-1) + TV_y(k-1 \mid k-1) + (T^2/2)a_y(k-1 \mid k-1) \\ V_y(k \mid k-1) = V_y(k-1 \mid k-1) + Ta_y(k-1 \mid k-1) \end{cases} \tag{7}$$

Where, X, V_x and α_x and are the target position in *X-axis*, velocity and acceleration, Y, V_y and α_y are the target position in *Y-axis*, velocity and acceleration.

$X(k|k\text{-}1)$ Covariance needs to be updated after system status is updated, as is shown in Equation (8):

$$P(k \mid k-1) = AP(k-1 \mid k-1)A^T + Q \tag{8}$$

Where, $P(k|k\text{-}1)$ is the corresponding covariance to $X(k|k\text{-}1)$, $P(k\text{-}1|k\text{-}1)$ is the corresponding covariance to $X(k\text{-}1|k\text{-}1)$, Q is the noise covariance of process.

With the state of the prediction result, further measurement of the state is still needed:

$$Z(k \mid k-1) = HX(k \mid k-1) + V(k \mid k-1) \tag{9}$$

Where, $Z(k|k\text{-}1)$ is the measured values, H is the parameter of measurement system, $V(k|k\text{-}1)$ is the process and measurement noise.

With predicted values and measured values , we can obtain the optimal estimate value $X(k|k)$ of the current state, as shown in Equation (10).

$$X(k \mid k) = X(k \mid k-1) + Kg(k)(Z(k) - HX(k \mid k-1)) \tag{10}$$

Where, K_g is Kalman gain

$$Kg(k) = P(k \mid k-1)H^T / (HP(k \mid k-1)H^T + R) \tag{11}$$

Now the optimal estimate value $X(k|k)$ in k state is obtained. However, in order to make the Kalman filter run continuously and form autoregressive process, we also need to update the covariance of $X(k|k)$ in k state:

$$P(k \mid k) = (I - Kg(k)H)P(k \mid k-1) \tag{12}$$

Where I is the identity matrix. When the system is in $k+1$ state, $P(k|k)$ is $P(k\text{-}1|k\text{-}1)$ in the formula (8), so the algorithm can operate continuously. In this paper, we use the Kalman filter algorithm to track and predict a moving target, where the input of Kalman filter is the position and speed of the moving object detected by the sensor, the output is the position information of the moving object at next time.

III. REAL-TIME DYNAMIC PATH PLANNING STRACTEGY

During the UAV cruise flight, if the airborne sensor detects moving obstacles in front of the UAV, the autonomous control system of UAV will call the real-time dynamic path planning strategy. Specific planning steps are as follows:

A. Forecast of the Track of the Obstacle

The airborne sensor could get the current position and velocity information of obstacles, and then, it could obtain the optimal estimation of the obstacle position at the next moment through the process of the Kalman filter algorithm. As a result, we could get the prediction of the obstacle position.

B. Track Search

In this paper, the improved PSO algorithm and chain structure is used to search. The specific algorithm is as follows:

- *Settings of the Particle Swarm*

The size of the particle swarm is set as m, the dimension of the particle is n, which is equal to the number of track control points. We divide the X-axis direction between the starting and ending into $n+1$ equal portions, so that the horizontal axis of the n track control points locate in these bisectors. Now we just do the random search for the n control points in the Y-axis coordinate. In order to search with chain structure, the minimum search particle dimension is set as N, these N track control points are the numerical points required to be planned for each time in chain structure.

- *Initialization of the Particle Swarm*

Compared to the global static path planning, the starting point of real-time dynamic path planning is set as the current UAV location, while the end point does not change. At the beginning of the algorithm, it initializes the position and velocity of the N track control points from the starting point. As the search algorithm is carried out, it will save the optimal results if the former particle of the chain structure gets an optimal solution. During the initialization of the latter particle of the chain structure, the starting point of search in the next section is set as the last point of the control points searched before, the same for the initialization of the next consecutive N-track control points.

- *Calculation and Preservation of the Particle Fitness Value*

At first, we could calculate the threat cost function value for each particle [10], from which we could synthesize to get the fitness value of each particle. Then, we could preserve the individual optimum searched by each particle and the global optimum searched by the particle swarm. In the calculation process of the threat cost value, considering the movement characteristics of the obstacle and the dynamic process of the threat scope, the distance between the two adjacent track control points is designed to be equal to the flight distance of the UAV in unit time. Thus, in the calculation of the threat cost, the effect of the current moving obstacle on each control point needs to be considered.

- *Position and Velocity Update of the Particles*

Based on the traditional update of the velocity and position, the chaotic disturbance is added to the optimization process. Finally, those update particles with higher values are saved.

- *Update the individual extreme value of each particle and global extreme value of the particle swarm.*
- *Termination of the Judgment*

In the track search process of each N particles, it needs to judge whether or not that the number of iterations is maximum, or the evaluation function is optimal, according to which it choose to continue or terminate the operation of the track search. If the termination condition is not satisfied, it needs to follow the chain link to jump to the 2nd step of the algorithm. Then, the search for the next N track control points begins. If the termination condition is satisfied, it ends the search process. Then, track control points are fitted to a curve with B-spline curve fit. The fitted curve is projected onto a digital map of the security zone, which is regarded as the reference track for the next UAV flight.

In order to make the track planning more efficient, this paper proposes an obstacle avoidance strategy to accelerate the arithmetic operations. We can abstract the impact of the moving obstacles on the UAV flight safety to the problem of space objects encounter. For the reason that the planning algorithm is implemented in the two-dimensional plane, according to the relationship between the UAV flight direction and the moving obstacle movement direction, obstacle avoidance strategy is divided into four cases, as is shown in figures from Fig.2 to Fig.5.

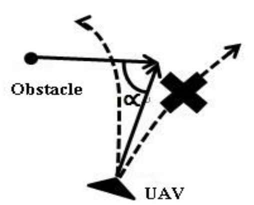

Figure 2. Collision in the 1st cas

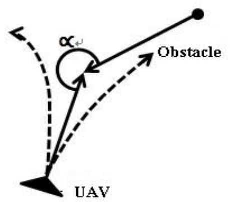

Figure 3. Collision in the 2nd case.

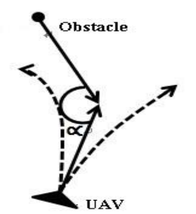

Figure 4. Collision in the 3rd case

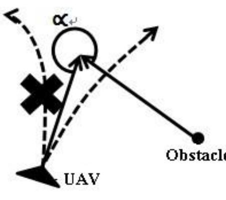

Figure 5. Collision in the 4th case.

Considering the different angle between the UAV flight direction and the moving obstacle movement direction, there are four cases to be analyzed as follows: $0°\leq\alpha\leq 90°$, $90°\leq\alpha\leq 180°$, $180°\leq\alpha\leq 270°$ and $270°\leq\alpha\leq 360°$. Where, the direction of the UAV flight is set as the positive direction, the track direction marked "×" is not be considered, the solid line represents the original path of obstacles and UAV, the dashed line represents the changed path of the UAV to avoid the obstacle.

IV. SIMULATIONS

When the UAV detects a moving obstacle ahead, the first thing to do is to start prediction tracking system to predict moving objects trajectory and model to load it into the digital map. When the Kalman filter is used for the moving object trajectory prediction, sampling frequency is 1 second, and parameters designing is given as (13).

$$P_0=\begin{bmatrix}0.1 & 0\\ 0 & 0.1\end{bmatrix},\quad A=\begin{bmatrix}1 & 1\\ 0 & 1\end{bmatrix},\quad B=\begin{bmatrix}0.5\\ 1\end{bmatrix},$$
$$Q=\begin{bmatrix}0.01 & 0\\ 0 & 0.01\end{bmatrix},\quad H=\begin{bmatrix}1 & 0\end{bmatrix} \qquad (13)$$

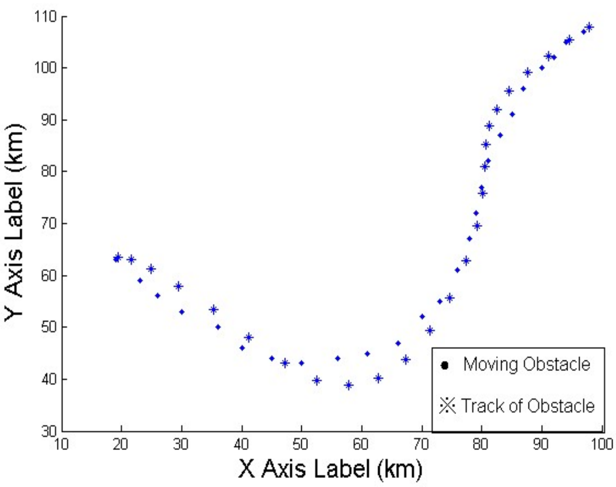

Figure 6. Object motion tracking results in two-dimensional plane

In order to verify the accuracy of the Kalman filter algorithm, we have a prediction and tracking research on the irregular motion objects in two-dimensional plane. The prediction results of motion trajectory are shown in Figure 6, where "· "is actual position of the object," * "is the object predicted position.

Assuming that the size of the simulate digital map is 100*100km^2, wherein there are 8 peaks shaped terrain or obstacles, their coordinates in the two-dimensional map, respectively, are as follows: (23, 70), (25, 20), (45, 55), (75, 70), (75, 25), (47, 25), (55, 80) and (85, 45), the threat radiuses, respectively, are as follows: 12, 12, 11, 11, 9, 7, 8, 7. We will picture the movement trajectory on a digital map for the moving objects predicted, the UAV will start a real-time dynamic path planning strategy for the flight path re-planning.

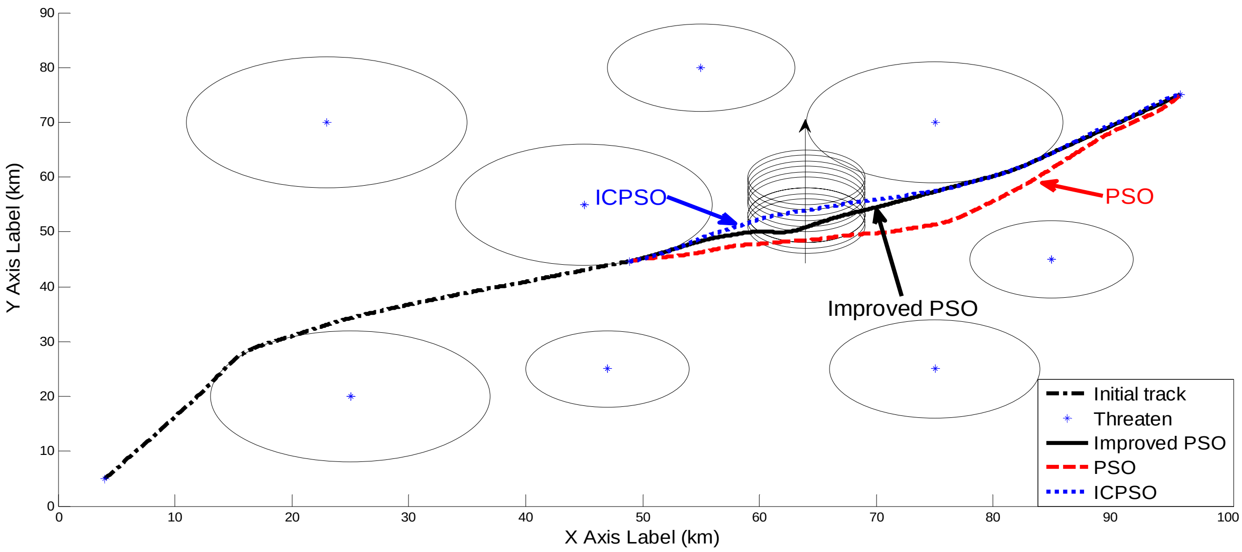

Figure 7.Real-time dynamic path planning in 2-D map, collision in the 4th case.

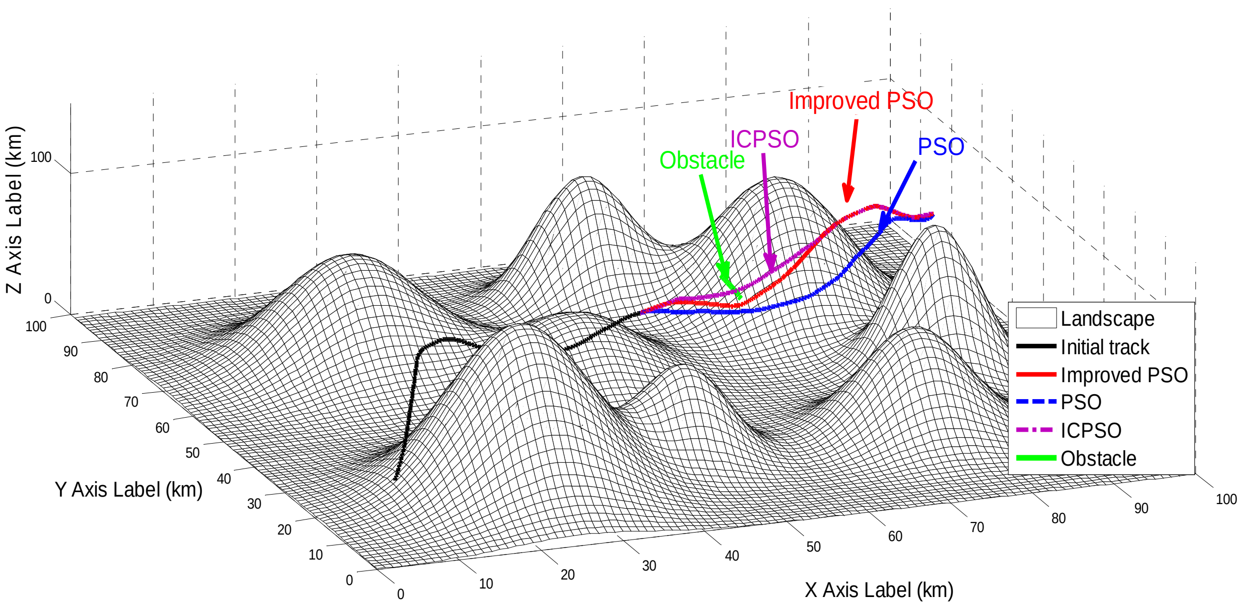

Figure 8. Real-time dynamic path planning in 3-D map, collision in the 4th case.

Path planning results corresponding to the collision in the 4th case are shown in Fig.7 and Fig.8, the trajectory of the obstacle represents the predicted position and the threat radius of the moving obstacle for each sampling time, the direction of the arrow indicates the direction of its trajectory. The path planned by the improved PSO and the obstacle avoidance strategy commendably avoids obstacles. While not only does the standard PSO algorithm hardly plan new path curves, and the risk of collision exists. Although the immune PSO algorithm without obstacle avoidance strategy makes the planning trail safe, the track line is non-optimal with longer range.

In summary, in the simulation study of the UAV path planning strategy, it makes the track planning more effective to use an improved PSO with variable structure strategy.

V. Conclusion

This paper proposed a kind of adaptive chaos PSO, which is successfully applied to UAV path planning in the practical flight environment. Before the start of path planning, the terrain environment is pretreated, including the extraction of terrain contours and the fitting of the terrain with an oval. Then, the variable structure optimization search strategy based on improved PSO, which successfully completed the UAV real-time path planning. Moreover, our comparison of the improved PSO and other PSO algorithms shows, with statistical significance, that our implementation of the real-time dynamic path planning strategy produces superior trajectories to the standard PSO and immune PSO algorithm.

References

[1] Shi Y, Eberhart R C. A modified particle swarm optimizer [R]. IEEE International Conference of Evolutionary Computation, Anchorage, Alaska, May 1998.

[2] Shi Y, Eberhart R C. Empirical study of particle swarm optimization [A]. Proceeding of Congress on Evolutionary Computation [C]. : Piscataway, NJ: IEEE Service Center, 1999. 1945- 1949.

[3] ZHANG Jinhua. Modified adaptive PSO algorithm based on cloud theory [J]. Computer Engineering and Applications, 2012, 48(5):29-31.

[4] ZHAO Zhi-gang, CHANG Cheng, Adaptive Chaos Particle Swarm Optimization Algorithm [J], Computer Engineering, 2011, 37(15): 128-130.

[5] WANG Xing-bo,LI Ben-wei,YANG Xin-yi, Research on particle swarm optimizing algorithm based on seek advantage and avoid disadvantage principle [J], Application Research of Computers, 2012,29(3):933-936.

[6] WANG Wei,LI Mei-yi,PENG Xia-dan, Dynamic Particle Swarm Optimization Algorithm Based on Two-layer Alterable Sub-population [J], Journal of Chinese Computer Systems, 2012,33(1):145-150.

[7] Ashraf Elnagar. Prediction of Moving Objects in Dynamic Environments Using Kalman Filters[C]. Proceeding of 2001 IEEE International Symposium on Computational Intelligent in Robotics and Automation, 2001:414-419

[8] Beekman M, Ratnieks F L W. Long-rang Foraging by the Honey-bee, Apis Mellifera L[J].Functional Ecologicy, 2000, (14): 490-496.

[9]]Ashraf Elnagar. Prediction of Moving Objects in Dynamic Environments Using Kalman Filters[C]. Proceeding of 2001 IEEE International Symposium on Computational Intelligent in Robotics and Automation, 2001:414-419

[10] Ghirmai, T. Gaussian particle filtering for tracking maneuvering targets[C]. SoutheastCon, 2007. Proceedings. IEEE, 2007, 439-443.