Research and Implementation of Multi-Agent UAV System Simulation Platform Based on JADE

Zhichao Zheng(✉), Yonggang Li, and Ying Wang

School of Communication and Information Engineering,
Chonqing University of Posts and Telecommunications,
Chongqing 400065, People's Republic of China
996179491@qq.com

Abstract. The unmanned aerial vehicle (UAV) system under the information condition is a complex adaptive system. The position, speed, efficiency value and other attributes of each unit in the UAV system are changing at high speed all the time. The dynamic equation of a single UAV's motion trajectory and operation is non-linear. The traditional modeling method is difficult to accurately reproduce the complex behavior of the system, so it is very important to put forward a modeling method suitable for the UAV system to reflect the contingency and emergence of the complex system. Therefore, a visualized multi-agent UAV system simulation platform is designed and implemented, regards UAV as an independent agent and the Java agent development framework (JADE) was used to simplify the development of multi-agent. Considering that the agent also needs a motion and obstacle avoidance model, a set of rules system is defined for the agent and an obstacle avoidance algorithm based on fuzzy control is implemented. Judging from the test results, the autonomous obstacle avoidance function can already be realized when the formation is merged, and the trajectory can be planned automatically when scanning the area. At the same time, this paper also provides a reference model for modeling and simulation for task reliability assessment.

Keywords: UAV · Agent · Avoidance algorithm · Task reliability

1 Introduction

The concept of agent is derived from distributed artificial intelligence. People call the computing entities that reside in an environment with the characteristics of autonomy, flexibility, sociality, and responsiveness as agents [1]. With

This work is supported by the National Defense Pre-Research Quick Support Foundation of China (no. 80911010302).

H. Gao et al. (Eds.): ChinaCom 2021, LNICST 433, pp. 115–128, 2022.
https://doi.org/10.1007/978-3-030-99200-2_10

the development of science and technology, agent has attracted much attention and is widely used in various fields [2]. The structure of agent is mainly divided into deliberate agent, reactive agent, Belief-Desire-Intention model (BDI) model and hybrid agent, deliberate agent is also called thinking agent [3]. The reactive agent [4] has no representation and reasoning behavior of the environment, but only works through stimulus response behavior. In the 1990s, the BDI model [5] was proposed by Rao and Georgeff. It was derived from the three words Belief, Desire, and Intention. There are two agent-based modeling theories in existing research: Distributed Artificial Intelligent (DAI) theory and Complex Adaptive System (CAS) theory [6]. The goal of DAI research is to establish a huge complex system composed of multiple subsystems, and each subsystem cooperates with each other to solve specific problems. CAS is a system in which adaptive subjects interact, evolve together, and emerge layer by layer, mainly reflected in the interaction and interactivity between agent and agent. This paper adopts the form of CAS, which can highlight the autonomy, adaptability and intelligence of agent. JADE is a software framework [7] that implements an efficient agent platform and supports the development of multi-agent systems. Literature [8] explains the principle of JADE, introduces the basic method of designing a performance test platform based on JADE, and provides a technical approach to realize the performance test platform. Literature [9] discusses mobile agents and their application in a heterogeneous multi-agent system in the form of a group of collaborative drones and UAV autonomous robots. Literature [10] proposes a multi-agent-based extended travel support system model and the method proposed in literature [11] involves the use of a multi-agent system, which solves the problem of using JADE to calculate the optimal multi-objective control. These literatures focus on the control of agent. However, visualization is also important for multi-agent systems

In this paper, a visualized multi-agent UAV system simulation platform is designed and implemented. In the form of CAS, the program model of agent is constructed with the help of object-oriented programming method to realize the multi-agent system. In order to realize the formation flight and area scanning functions of the agent, the rule system of the agent in the equipment system environment must be determined first, and the movement model and obstacle avoidance model must be realized. Secondly, according to the model, UAV formation and obstacle avoidance functions are realized, and a more efficient area scanning function is also realized. Finally, the GUI is used to display the task reliability curve.

2 Rule System

Rule system is an important part of UAV agent. According to different UAV agents, different rule system is established. Agent performs probability matching according to the weights in the rule base. Firstly, appropriate rules are selected and executed according to task objectives, internal states and environmental parameters. Finally, the rule system is evaluated and modified according to the execution results

The rules in the rule base form corresponding effector instructions according to sensor perception results and internal state monitoring. This paper mainly implements the agent motion model and agent obstacle avoidance model in UAVs. The structure of the agent's rule system is shown in Fig. 1.

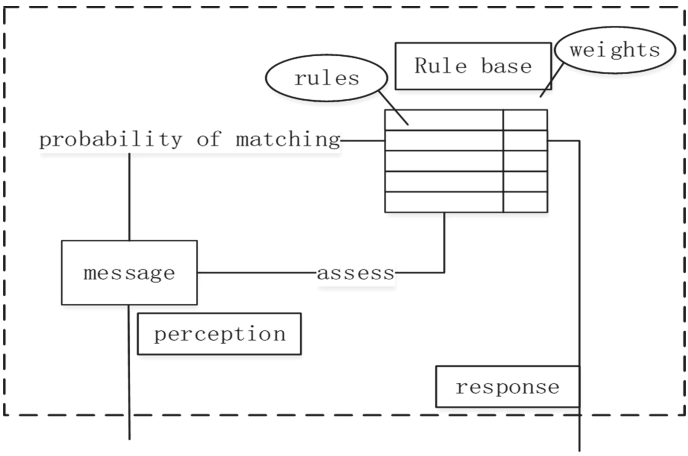


Fig. 1. Agent's rule system

2.1 Agent Mobility Models

Agent moves in order to accomplish tasks, such as agent requires to reach a specified destination, and the follower dynamically adjusts its position according to the position of the leader. Therefore, it is important to discuss the movement of the agent, and to understand how the agent dynamically adjusts its position to reach the desired position. As shown in Fig. 2, an UAV (agent) receives the command from the superior at the point of the global coordinate system $A(x1, y1)$ and moves quickly to the target point $P(x2, y2)$ at the initial speed $\overrightarrow{v}_{\alpha 1}$, which is the direction of movement and has a driving force $\overrightarrow{F}$ to point to the target point.

Then, the kinematics model of UAV moving in the specified direction is:

$$v_x = \overrightarrow{v}_{\alpha 1} \cos \alpha \tag{1}$$

$$v_y = \overrightarrow{v}_{\alpha 1} \sin \alpha \tag{2}$$

where, v_x is the velocity component of the initial velocity $\overrightarrow{v}_{\alpha 1}$ along the x axis, and v_y is the velocity component of the initial velocity $\overrightarrow{v}_{\alpha 1}$ along the y axis. When a UAV receives an order from a superior and needs to reach a certain destination, in order to reach the combat area with the fastest speed, it needs a force pointing to the target point. If the fixed target point and the coordinates

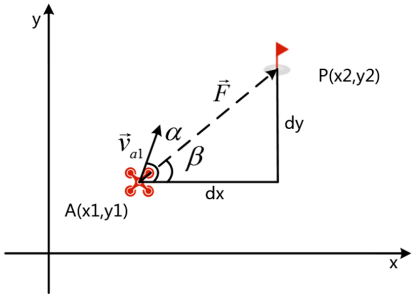


Fig. 2. Agent flies to the fixed mission area

of the UAV are known, then the direction is β and the driving force is, and its dynamic equation is:

$$\tan\beta = \frac{y2 - y1}{x2 - x1} = \frac{dy}{dx} \tag{3}$$

In Eq. 3, the direction of driving force β can be expressed as:

$$\beta = \tan^{-1}\frac{dy}{dx} \tag{4}$$

Given that the weight of the UAV is m kg, the acceleration of the UAV under the action of the driving force $\overrightarrow{F}$ can be expressed as:

$$\overrightarrow{a} = \frac{\overrightarrow{F}}{m} \tag{5}$$

The acceleration along the x and y axes can be expressed as:

$$\overrightarrow{a}_x = \overrightarrow{a}\cos\beta \tag{6}$$

$$\overrightarrow{a}_y = \overrightarrow{a}\sin\beta \tag{7}$$

Then the ideal position of UAV at the next moment can be expressed as:

$$x = x1 + \overrightarrow{v}_x \Delta t + \frac{1}{2}\overrightarrow{a}_x \Delta t^2 \tag{8}$$

In Eq. 8, Δt is the sampling period. The ideal position (x, y) that the agent needs to move in the next step can be obtained through the above formula. By constantly updating the position and initial speed, the UAV can reach the destination at the fastest speed and complete the task.

As shown in Fig. 3, the followers fly to the leaders at different time, and the leaders constantly communicates with the followers to exchange data. The leader sends real-time location information to the followers, and the followers parse the location information sent by the leaders into a fixed target movement, then the followers update the location and speed in real-time through internal calculations. Finally, the task of moving the followers to the moving target is completed.

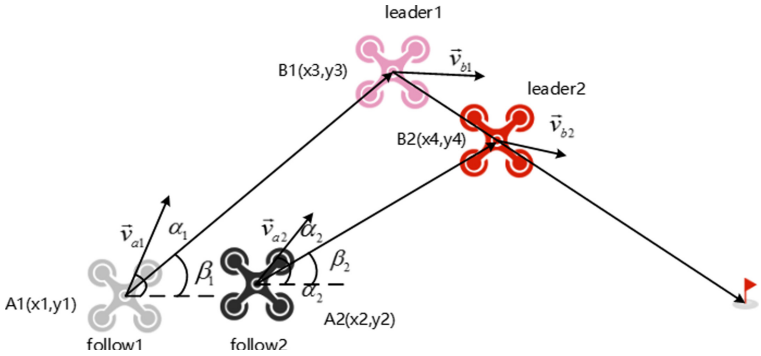


Fig. 3. The follower flies to the leader at different times

2.2 Agent Obstacle Avoidance Algorithm

Agent obstacle avoidance models are divided into two types, one is between agent and static obstacles, the other is between agent and moving objects. The essence of the collision avoidance algorithm lies in the agent can safely reach the designated destination and complete the task in the simulation process without overlapping with the non-self-dangerous area. The current UAV obstacle avoidance algorithm [14] has high time complexity and mostly target static obstacles. This paper proposes an improved speed collision avoidance algorithm based on literature [15,16], which is shown in Fig. 4.

In order to ensure that the agent can complete the task safely, based on the agent's mobile model, the speed of the agent is continuously modified through the collision avoidance algorithm so that the extension direction of the speed does not intersect with the velocity danger zone (VDZ), that means to select suitable speed at intervals for the agent which is outside the VDZ caused by the obstacle. If the direction of the agent's speed is outside the VDZ, the speed is directly used to point to the agent's target position, and the agent can safely move to the target point, that is, the agent chooses a speed that will not collide in the future to move. But the speed obstacle avoidance algorithm, each time it traverses all agents within the communication distance and obtains the best speed will make the algorithm more complicated. In order to satisfy that each agent does not collide with static obstacles or dynamic obstacles in the process of moving, and complete tasks in its own safe area, a speed collision avoidance algorithm based on fuzzy control is proposed. As shown in Fig. 5, in a two-dimensional plane, there is a static obstacle A, whose reference point position is $P_a(x1, y1)$, and whose radius is R_a. In addition, there is an Agent B whose reference point position is $P_b(x2, y2)$, radius is R_b and initial velocity is $\overrightarrow{v}_b$.

Assuming that agent slowly approaches the static obstacle so that agent B has exactly an intersection point with obstacle A, which can be expressed as:

$$\sqrt{(x1 - x2)^2 + (y1 - y2)^2} = R_a + R_b \tag{9}$$

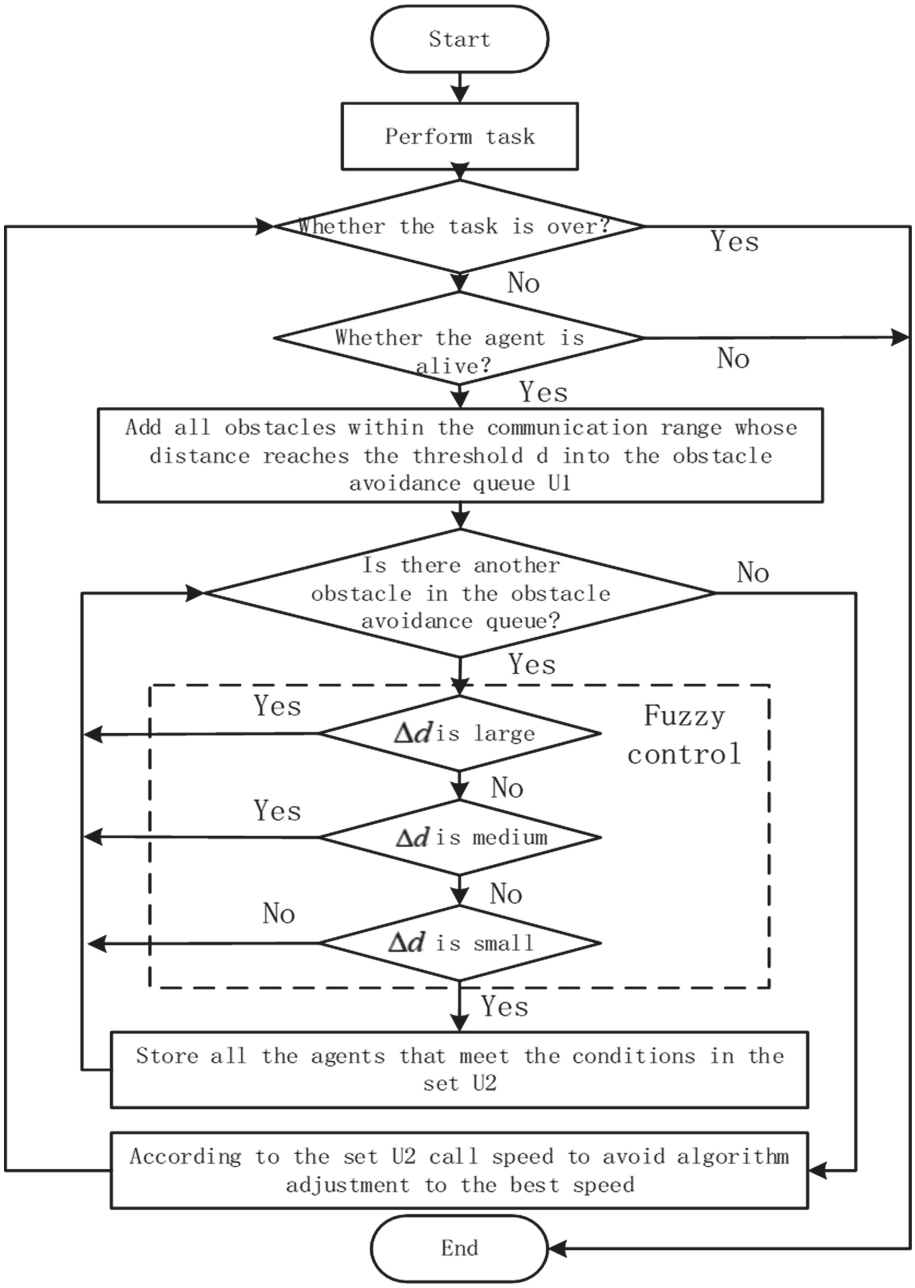


Fig. 4. Speed collision avoidance algorithm based on fuzzy control

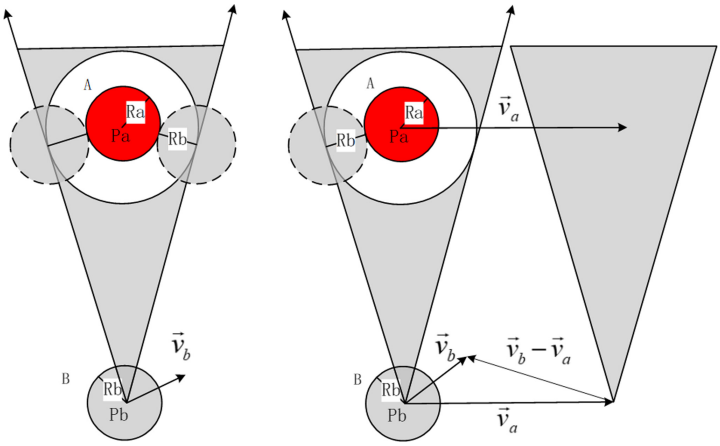


Fig. 5. Agent collides with a 'static' or 'dynamic' object

Set up the equations of agent B and obstacle A:

$$\begin{cases} (x - x1)^2 + (y - y1)^2 = R_a^2 \\ (x - x2)^2 + (y - y2)^2 = R_b^2 \end{cases} \tag{10}$$

By finding the number n of solutions of the equation system, the following conclusions can be obtained.

$$n = \begin{cases} 0, \Delta < 0 \\ 1, \Delta = 0 \\ 2, \Delta > 0 \end{cases} \tag{11}$$

In Eq. 11, when $\Delta = 0$, the agent and the obstacle just have an intersection point, and there is no collision. When $\Delta > 0$, the agent has collided with the obstacle. When $\Delta < 0$, the agent does not collide with the obstacle.

According to Eq. 11, to make the agent avoid obstacles successfully, it only needs to satisfy $n \leq 1$ at all times, that is, $\Delta \leq 0$. As shown in Fig. 5, taking the center P_b of agent B as the starting point to make a tangent to obstacle A, and two rays will be generated. When the direction of the agent's speed is between the two rays, then at some point in the future, agent B will collide with obstacle A. This possible collision speed is expressed as VDZ, that is, at some point in the future, the agent will collide with the obstacle. The number of solutions will be greater than 1. At this time, agent B needs to continuously detect obstacles within the communication range, and adjust the speed autonomously and coordinately, so that agent B can successfully avoid obstacles. When the value of Δd is large, no action is performed and it just continues to detect. When the value of Δd is medium, some simple actions should be performed, such as ignition of a small thrust engine or a warning message to other agents. When the value of Δd is small, add A to B's obstacle avoidance set U2, use the traditional speed

obstacle avoidance algorithm to select the best speed, and set the speed of B as the best speed. Among them, Δd is the relative position between A and B in a period of time step Δt, which can be expressed by the following formula:

$$\Delta d = \min_{0 \le t_s \le \Delta t} \|(pb - pa) + (\overrightarrow{v}_b - \overrightarrow{v}_a) \times t_s\| \tag{12}$$

In Eq. 12, Δd as a variable of the distance between A and B, the values of Δd are small, medium, and large after fuzzification. For these three values, as shown in Table 1.

Table 1. Fuzzy control rules for obstacle avoidance.

Input variable Δd	Output variable Δd	Whether a collision is detected	Thread priority
Δd(large)	Long	Not detected	Low priority
Δd(medium)	Medium	Not detected	Medium priority
Δd(small)	Short	Detected	High priority

In the multi-agent UAV system simulation platform, agents and agents couple with each other, form formations, and have a superior-subordinate relationship. In addition to avoiding collisions between agents and the environment, obstacle avoidance functions are also required between agents and agents. As shown in Fig. 5, it contains two moving entities, which can be regarded as two agents or a situation of an agent and a dynamic obstacle. Since both A and B are moving entities, obstacle A is selected as the reference object, and the relative speed $\overrightarrow{V}$ of agent B relative to obstacle A is obtained.

$$\overrightarrow{V} = \overrightarrow{V}_b - \overrightarrow{V}_a \tag{13}$$

The obstacle avoidance set U2 of the current agent is updated in real time, and then the traditional obstacle avoidance algorithm is called to obtain the best obstacle avoidance speed, and finally the agent moves to the ideal position. As shown in Fig. 6, there will be a VDZ whether it is an agent, a static obstacle or a dynamic obstacle. If the agent wants to avoid collisions and reach the specified position, it needs to avoid the speed danger zone, choose the speed direction outside the speed danger zone, and continuously update the speed in each sampling period to make it reach the target safely point.

3 Simulation Results

3.1 UAVs Formation Simulation Results

As shown in Fig. 7, when the agent has a multi-component cluster, each formation will select a leader through negotiation. Under the leadership of the leader, when the agents reach the mission area, if the leaders can communicate with

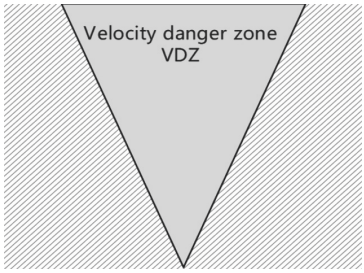


Fig. 6. Collision zone and safety zone

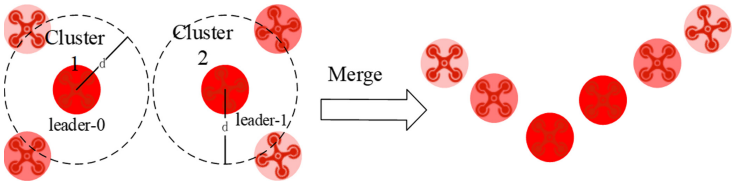


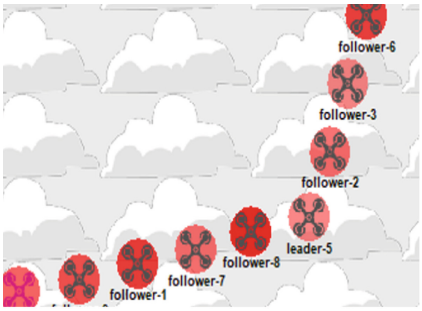
Fig. 7. Two groups of UAVs formation

each other, they will reselect the leader to form a new formation. In the end, the task of all agents is to reach the task area and merge into a cluster.

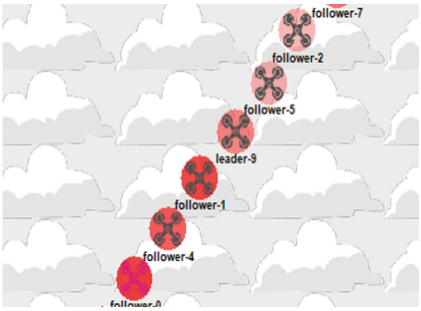
UAVs can fly in formation through clustering and fusion. As shown in Fig. 8, a total of 10 UVAs are created, and the distance between UAVs is 6 m. The initial formation of the UAVs formation is Fig. 8(a) and the angle in the formation is set to 30°. Figure 8(b) is formation 1, the formation angle is 180° and the distance between UAVs is 6 m. Figure 8(c) is formation 2, the formation angle is 45° and the distance between UAVs is 90 m. Figure 8(d) is formation 3, the formation angle is 120° and the distance between UAVs is 6 m. The test results show that the UAVs formation adjustment function is normal and can meet expectations.

3.2 Collision Avoidance Test Simulation Results

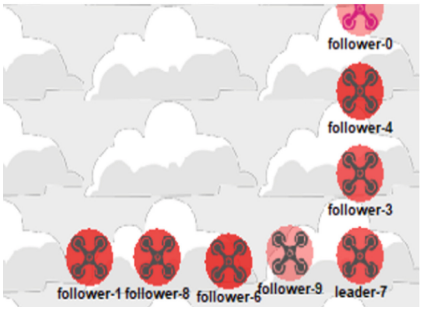
Through the agent obstacle avoidance model, the UAV can avoid obstacles with static and dynamic objects. As shown in Fig. 9, by creating 10 UAVs and setting the flight mode of the drones to formation flying, it is used to test whether the collision avoidance model between UAVs is correct. As shown in Fig. 9(a), when UAV follower-2 goes to its desired position in the formation, there is fol-lower-8 in the trajectory, Fig. 9(b) shows the position where the follower-2 successfully bypassed the follower-8, and the two did not collide. Follower-8 avoids the trajectory of follower-2 by adjusting its position. The test results show that the collision avoidance algorithm proposed in this paper is suitable for this system and can meet expectations.



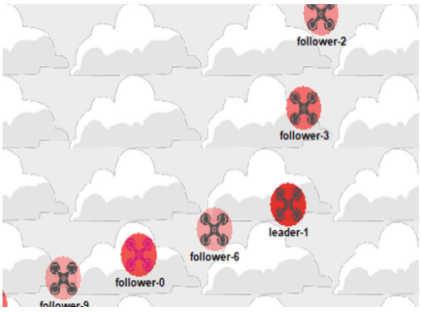
(a) Initial formation



(b) Formation 1

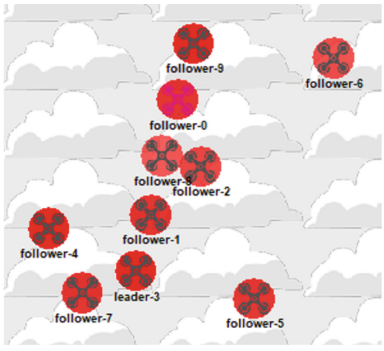


(c) Formation 2

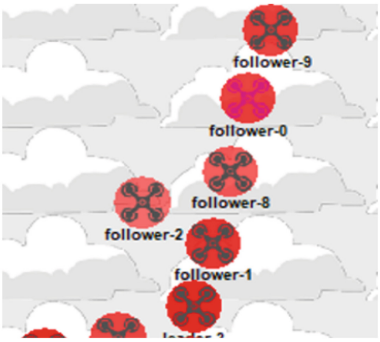


(d) Formation 3

Fig. 8. UAVs formation flying



(a) Follower-2 and follower-8 will collide



(b) Successful obstacle avoidance

Fig. 9. Collision avoidance test

3.3 Dynamic Task Reliability Assessment

Simulation Scenario Hypothesis and Task Reliability Definition. The scene is set as $32 \times 32\,\mathrm{km}^2$, and the maximum number of UAVs is 32. Each UAV will perform the straight line cruise scanning task. In the simulation scenario,

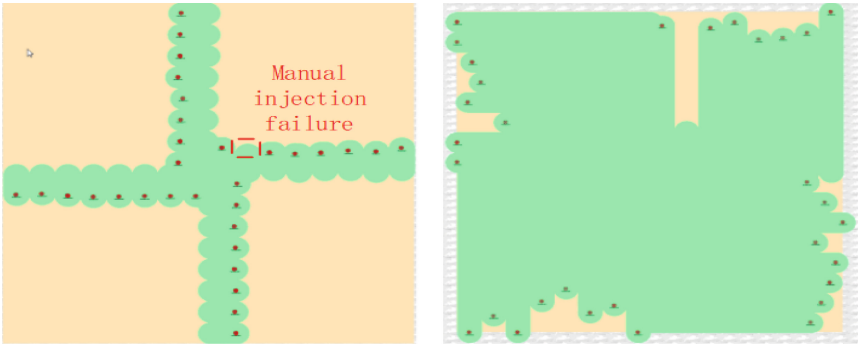
the number of aircraft is taken as the constraint condition, and the constraint interval is 1 to 32 aircraft. Under this constraint, the maximum flying speed of the UAV is 15 m/s, and it can cover the scanning area with a diameter of 2 km. The same number of UAVs are simulated to complete the assigned tasks, and the less time-consuming the UAVs are, the higher their reliability will be.

In the large-area detection task, the task completion degree is defined as the task reliability with the goal of completing the scanning task, which can be expressed as:

$$R = \frac{S_h}{S_{all}} \tag{14}$$

In Eq. 14, S_h is the scanned area, and S_all is the total area of the scanned area. In this paper, R > 82% indicates that the scan task is completed, but due to the existence of equipment failure, the task may not be completed during the execution of the algorithm.

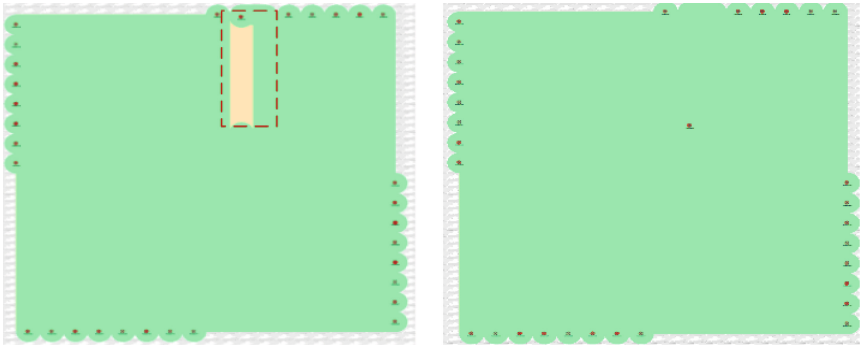
Comparison Between the Traditional Recovery Mode and Intelligent Recovery Mode. In the traditional recovery mode, the UAV scans a given area. In the initial creation stage, the UAV receives the area scanning task from the superior, and after parsing the task information, it starts the area scanning task immediately after reaching the designated starting point of the task. As shown in Fig. 10(a), with the simulation progresses, a fault is manually injected to randomly strike a UAV which is forced to go offline. Because one of the UAVs has broken down, the scanning task assigned to it cannot be completed in time. The other 31 UAVs continue to perform straight-line cruise scanning tasks, as shown in Fig. 10(b).



(a) Manual injection failure (b) UAVs perform scanning tasks

Fig. 10. Traditional recovery mode

After 31 UAVs have completed their straight-line cruise tasks, the superior checks whether the scanning task is completed. If there is an area which is not



(a) Neighboring UAVs complete remaining task (b) Task completed

Fig. 11. Traditional recovery mode

accessible to scan, the superior will re-issue the task and let the neighboring UAV continue to complete the task, as shown in Fig. 11(a).

In the intelligent recovery mode, after fault injection, because the intelligent recovery UAV is highly autonomous, it can sense its surroundings in real time, monitor neighbor UAV, communicate with other UAV, collect various information, and finally make a decision, and decide what to do next on the cruise task. As shown in Fig. 12, the current UAV detects the downline of the neighbor UAV, and the UAV make decisions through consultation, fill the inner cruise path, and move the uncompleted task area to the edge.

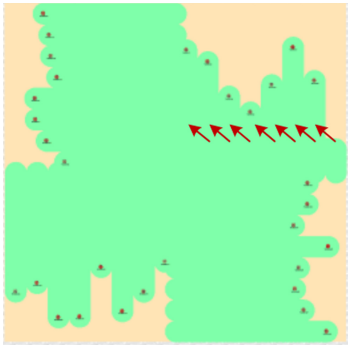


Fig. 12. Autonomous negotiation movement

Finally, the task reliability curves of the traditional recovery mode and the intelligent recovery mode can be obtained. In Fig. 13, the x-axis represents the system time, which is calculated from the creation stage, and the y-axis represents the task reliability R. In the intelligent recovery mode, when the system

time reaches about 1600 s, the UAVs have completed the task of regional scanning.

It can be seen that after 1300 s, in the intelligent recovery mode, the task reliability is higher than the traditional recovery mode, and the task can be completed faster, so the intelligent recovery algorithm is better than the traditional algorithm.

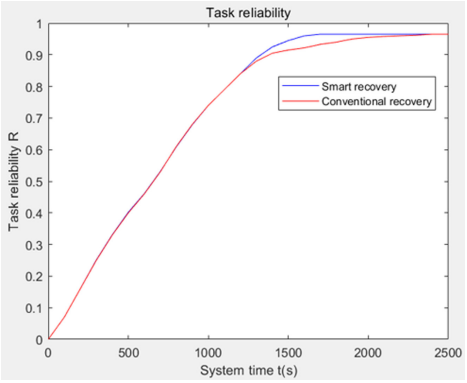


Fig. 13. Task reliability

4 Conclusion

As a complex system, the multi-agent UAV system has many uncertainties. This paper first established motion model and obstacle avoidance model of the UAV, and simulated the functions of UAV movement, obstacle avoidance and area scanning. Then, the user can view the real-time task status through the interactive interface. Finally, the test verification shows that each functional module of the visual multi-agent UAV system simulation platform based on JADE can run nor-mally, meet various performance indicators, and realize the reliability evaluation of dynamic tasks. It can be seen from the test results that the multi-agent UAV system simulation platform can realize functions such as autonomous obstacle avoidance, formation and formation integration, and automatic negotiation between UAVs. Meanwhile, it also provides a reference model for modeling and simulation for task reliability evaluation.

References

1. Jennings, N.: On agent-based software engineering. Artif. Intell. **117**(2), 277–296 (2000)

2. Liu, J.: Operation command decision modeling and simulation research based on agent technique. In: 2019 International Conference on Information Technology and Computer Application (ITCA), pp. 203–206. IEEE, Guangzhou (2019)
3. Jennings, N.R.: On agent-based software engineering. Artif. Intell. **2**(5), 99–110 (2016)
4. Rauff, J.V.: Multi-Agent Systems: An Introduction to Distributed Artificial Intelligence. Addison-Wesley, Boston (1999)
5. Rao, A., Georgeff, M.P.: Decision procedures for BDI logics. J. Logic Comput. **8**(3), 293–343 (1998)
6. Zhou Kaibo, W., Xiaokang, G.M., et al.: Neural-adaptive finite-time formation tracking control of multiple nonholonomic agents with a time-varying target. IEEE Access **8**, 62943–62953 (2020)
7. Bellifemine, F., Caire, G., Greenwood, D.: Developing Multi-Agent Systems with JADE, pp. 1–286. Wiley, Hoboken (2007)
8. Baiquan, X.: Design of platform for performance testing based on JADE. In: 2014 Sixth International Conference on Measuring Technology and Mechatronics Automation, pp. 251–254. IEEE, Zhangjiajie (2014)
9. Obdržálek, Z.: Mobile agents in multi-agent UAV/UGV system. In: 2017 International Conference on Military Technologies (ICMT), pp. 753–759. IEEE, Brno (2017)
10. Sreeja, M.U., Kovoor, B.C.: Multi agent based extended travel support system using JADE framework. In: 2017 International Conference on Energy, Communication, Data Analytics and Soft Computing (ICECDS), pp. 3561–3564. IEEE, Brno (2017)
11. Estrada, D.F., Lee, K.Y.: Multi-agent system implementation in JADE environment for power plant control. In: 2013 IEEE Power and Energy Society General Meeting, pp. 1–5. IEEE, Brno (2017)
12. Abeywickrama, H.V., Jayawickrama, B.A., He, Y., Dutkiewicz, E.: Algorithm for energy efficient inter-UAV collision avoidance. In: 17th International Symposium on Communications and Information Technologies (ISCIT), pp. 1–5. IEEE, Cairns (2017)
13. Yoo, C., Cho, A., Park, B., Kang, Y., Shim, S., Lee, I.: Collision avoidance of Smart UAV in multiple intruders. In: 12th International Conference on Control, Automation and Systems, pp. 443–447. IEEE, Jeju (2012)
14. Ling, L., Niu, Y., Zhu, H.: Lyapunov method-based collision avoidance for UAVs. In: The 27th Chinese Control and Decision Conference (2015 CCDC), pp. 4716–4720. IEEE, Qingdao (2015)
15. Van den Berg, J., Lin, M., Manocha, D.: Reciprocal velocity obstacles for real-time multi-agent navigation. In: 2008 IEEE International Conference on Robotics and Automation, pp. 19–23. IEEE, Pasadena (2008)
16. Rezaee, H., Abdollahi, F., Menhaj, M.B.: Model-free fuzzy leader-follower formation control of fixed wing UAVs. In: 13th Iranian Conference on Fuzzy Systems (IFSC), pp. 1–5. IEEE, Qazvin (2013)