scientific reports

OPEN

Enhanced multi agent coordination algorithm for drone swarm patrolling in durian orchards

Ruipeng Tang[1]✉, Jianrui Tang[2], Mohamad Sofian Abu Talip[1], Narendra Kumar Aridas[1] & Xifeng Xu[3]

This study proposes an enhanced multi-agent swarm control algorithm (EN-MASCA) to solve the problem of efficient patrolling of drone swarms in complex durian orchard environments. It introduces a virtual navigator model to dynamically adjust the patrol path of the drone swarm and perform obstacle avoidance and path optimization in real time according to environmental changes. Different from traditional algorithms that only rely on fixed path planning, the virtual navigator model significantly improves the flexibility and stability of the drone swarm in complex environments. It also applies deep reinforcement learning algorithms to path planning and obstacle avoidance of drone swarms for the first time, improving the algorithm's adaptability and optimization capabilities by learning dynamic information in complex environments. This innovation significantly improves the applicability of existing methods in complex terrain and dynamic obstacle environments. Finally, it incorporates the simulation characteristics of biological swarm behavior, and on this basis, comprehensively optimizes the flight path, obstacle avoidance and swarm stability of the drone swarm. By improving control strategies and parameter design, it improves the trajectory consistency and mission completion efficiency of the UAV swarm during flight. In the experimental part, this study verified in detail the advantages of the EN-MASCA algorithm in terms of flight trajectory, flight stability, cluster consistency and task completion efficiency by constructing a six-degree-of-freedom UAV motion simulation model and real environment simulation. It provides an efficient and intelligent solution for collaborative patrol operations of drones in durian orchards, which has important practical application value and promotion prospects.

Keywords Drone swarm, Patrol operation, Durian orchard, Multi-agent swarm control algorithm, Path planning, Obstacle avoidance strategy, Deep reinforcement learning, Virtual navigator model

The durian orchard is a large-area and high-density agricultural plantation. Its management and maintenance require efficient patrol operations to ensure the healthy growth and high yield of durian[1]. However, since durian orchards usually have large area, much manpower must be invested in patrolling especially the peak periods. In addition, the cost of agricultural labor is rising year by year, which causes higher costs of relying on manual patrols of durian orchards. It is also difficult to ensure full coverage of the entire durian orchard, which will miss some areas and affects the patrol effect. The environment of durian orchard is large, so manual patrols require long walks, which consumes a lot of physical strength[2]. With the application of drone technology in agriculture, durian orchards have also begun to try to use drones for patrol operations. Drones can cover large areas quickly, provide high-resolution real-time images and data and improve the patrol efficiency and accuracy. However, the drone patrol operations also have some problems. The autonomous flight path planning and navigation accuracy of drone swarms is insufficient, which is easy to deviate from the planned route or fail to cover all target areas[3]. Due to the complex terrain of the durian orchard, the drone swarm needs to fly at different altitudes, which increases the difficulty of path planning. The weather conditions (such as wind speed, wind direction and rainfall) will also affect the flight stability of the drone swarm, which causes the flight trajectory deviation, mutual interference and collisions. So drone swarms need to calculate and optimize paths to cope with changes

[1]Department of Electrical Engineering, Faculty of Engineering, University of Malaya, Kuala Lumpur 50603, Malaysia. [2]Department of Computer Science and Technology, School of Information Engineering, Shanghai Maritime University, No. 1550, Haigang Avenue, Nanhui New Town, Pudong New Area, Shanghai, China. [3]Department of Civil Engineering, Faculty of Engineering, University of Malaya, Kuala Lumpur 50603, Malaysia. ✉email: 22057874@siswa.um.edu.my

in the dynamic environment. It also needs to share the real-time location information and status data. If the synchronization is not timely, the drones cannot coordinate and perform tasks[4].

Literature review

In order to solve the above-mentioned problem of motion cruising, some scholars have made some achievements in some scene applications. Miao et al.[5] proposed a heuristic algorithm for the SAC (slice admission control) problem in 5G/B5G networks. It introduces the resource efficiency of the introduced service to correct the priority violation, and then sets the target CSAR for each service type and improves its actual CSAR to improve accuracy. Sun et al.[6] designed a task scheduling algorithm based on proportional fairness awareness auction (PFAPPO) based on proximal policy optimization. It allocates computing resources reasonably to each drone, so that the drone learns the computing resources available at each unloading destination, solving the problems of extremely long queue delays and load imbalance. Li et al.[7] proposed a decision-making and motion planning integration framework with non-oscillation capability to overcome the shortcomings of autonomous driving in lane change/keeping operations. It also designed a belief decision planner with predicted trajectory uncertainty, which provides more appropriate information for autonomous driving planning and solves the optimal motion sequence. Xu et al.[8] used sequential images of adjacent stations and the bundle adjustment (BA) method to obtain the precise position of the lunar rover; and proposed a cross-scale cost aggregation stereo matching network to obtain disparity maps to extract the indicators of impact craters, so as to realize the precise positioning and sample collection of impact craters on the lunar surface by the lunar rover. Wang et al.[9] explored the basic characteristics of human legs such as linearity and nonlinearity during movement and used them for stability analysis and accurate motion prediction of robots and rehabilitation exoskeletons. Wu et al.[10] proposed an anonymous clustering algorithm for obstacle avoidance through the location of obstacle boundary points. It divides the consensus term into speed and speed unit direction, designs gradient-based terms to achieve separation and aggregation of agents and obstacle boundary points, which guides all agents to achieve group target following.

Lu et al.[11] describe the current principles and development of neuromorphic computing technology, explore its potential examples and future development routes for application in smart agriculture. Li et al.[12] introduced a highly configurable intelligent agricultural robotic arm system (CARA). It integrates a highly configurable robotic arm, image acquisition module, and deep processing center to facilitate accurate and efficient agricultural tasks. Zhou et al.[13] developed a hybrid architecture inspired by vehicle lateral dynamics, embedding data-driven models into physical models for parameter identification and error characterization, and achieving accurate and interpretable modeling. Chen et al.[14] designed a spatial attention mechanism with a feature fusion module to calculate the weights of different channel features. It also developed a hybrid model combining physical and dual attention neural networks to model vehicle lateral dynamics to solve the problem that neural networks with limited data are difficult to achieve accurate prediction. Meng et al.[15] combined mobile navigation with visual perception, using advanced algorithms to grasp objects in a way that suits human preferences, and using path planning and obstacle avoidance to navigate back to the human user. Li et al.[16] proposed a road segmentation method based on centroid Voronoi tessellation (CVT) for brain-controlled robot navigation via asynchronous BCI. It also proposed a new road segmentation method based on CVT to generate optional navigation targets in the road area for arbitrary target selection. Zhou et al.[17] proposed a drone anomaly detection method based on wavelet decomposition and stacked denoising autoencoder. It takes into account the negative impact of noisy data and the feature extraction ability of deep learning models, aiming to improve the accuracy of the proposed anomaly detection method by using wavelet decomposition and stacked denoising autoencoder methods.

Chen et al.[18] proposed a fair and efficient MAC protocol based on CSMA/CA, using multi-user MIMO to achieve concurrent uplink transmission from different drones. Wang et al.[19] proposed a drone-assisted URLLC scheme for edge users, using information age as an indicator of system delay to achieve the performance requirements of ultra-reliable low-latency communication. Gao et al.[20] obtained the modeling parameters of the theoretical closed-form energy model based on the existing literature based on the curve fitting of the model, and proposed a theoretical energy model for rotorcraft UAVs. Yin et al.[21] constructed the autonomous navigation of UAVs in a three-dimensional environment with adaptive control as a Markov decision process, and proposed a deep reinforcement learning algorithm. They also proposed a new speed constraint loss function and added it to the original actor loss to improve the speed control ability of the UAV. Zhang et al.[22] proposed an adaptive pseudo-inverse control scheme based on fuzzy logic system (FLS) and barrier Lyapunov function (BLF) for a class of state-constrained hysteresis nonlinear systems, which showed great application potential in the fields of soft bionic robots and rehabilitation robots. Ji et al.[23] proposed a multi-agent deterministic policy gradient (MADPG) method based on an actor-critic network, and proved its convergence and optimality by minimizing the local cost (Q-function), thereby improving the data utilization of the network. Liang et al.[24] proposed an integrated framework that combines three basic modules (such as the behavior decision-making, path planning, and motion control modules) to improve the safety of AVs in mixed traffic high-speed cruising scenarios.

Literature conclusion

Compares the EN-MASCA algorithm with other studies in the main findings, quantitative results, and key similarities and differences, although these studies (such as[11,13,17,21]) also combine reinforcement learning, deep learning or hybrid modeling techniques to improve model performance like this study, they aim to optimize path planning, obstacle avoidance and task completion efficiency through advanced algorithms. Some studies (such as[12,15,18]) emphasize the practical application of smart agriculture and multi-UAV swarm operations, which is similar to the research direction of this study. However, the above studies focus on smart agriculture, brain-controlled robot navigation, and vehicle dynamic modeling, while this study focuses on UAV swarm tasks in complex agricultural scenarios. This study also combines DQN and PPO algorithms for the first time to

optimize UAV swarm path planning and obstacle avoidance, and designs the virtual navigator model to enhance environmental adaptability. Table 1 shows the comparison results of the EN-MASCA algorithm with some key studies.

The above studies are similar to this study in terms of overall goals, and both aim to improve the navigation, path planning and task completion efficiency of drone swarms through advanced algorithms. They lay the theoretical foundation for this study, especially in terms of reinforcement learning, dynamic path planning and group behavior modeling. Most of them use reinforcement learning, deep neural networks, heuristic algorithms, etc. for path planning and obstacle avoidance. Some studies combine task allocation or data fusion technology to improve algorithm performance. Although they have achieved some success, most of them are aimed at ideal or relatively simple optimization path planning and task allocation, and lack applicability in the complex dynamic environments (such as agricultural scenes or high-density obstacle environments). Some studies are difficult to meet the requirements of real-time and efficient computing in practical applications due to the high complexity of the algorithm or the need for large-scale computing resources. Other studies focus on generalized task planning and navigation problems, lack optimization design for specific fields (such as agricultural drone patrols, disaster monitoring, etc.), and cannot fully meet the needs of the field. Therefore, this study designed a virtual navigator model and a six-degree-of-freedom drone motion simulation model, and achieved real-time path optimization and efficient collaboration by introducing an enhanced multi-agent cluster control algorithm (EN-MASCA) and integrating DQN and PPO algorithms. It solves the problem of poor adaptability of the above methods in complex scenarios, and greatly improves the stability and efficiency of drone clusters in complex agricultural scenarios such as durian orchards.

Methods and materials

Drone model construction

This study combines ROS with the GAZEBO physical simulation platform[16], which controls the drone to obtain the sensor data through ROS and builds a 3D simulation model of the drone. It also introduces the PX4 flight control system, which is divided into the human-computer interaction layer, cluster algorithm layer, PX4 flight control layer and physical simulation layer. The top layer, such as the ground station or off board node, will output the desired state of the drone and pass it to the PX4 flight control layer[17]. Finally, the flight controller transmits its attitude information to the GAZEBO simulator for 3D display. In order to control the drones in the cluster, this study sets the three variables of speed V_i, yaw angle R_i and altitude H_i. The pitch angle, yaw angle and throttle are used to control the PX4 flight control. In order to enable the clustering algorithm to control the PX4, this study converts the output command into the input command of the PX4 flight control attitude loop.

Study	Method	Application scenario	Key findings	Quantitative results
Lu et al.[11]	Neuromorphic computing technology, exploring potential applications in smart agriculture	Smart agriculture	Provides examples of neuromorphic computing for data processing and task optimization; proposes future development directions	No specific experimental data provided
Li et al.[12]	Highly configurable intelligent agricultural robotic arm (CARA), integrating image acquisition and deep processing modules	Smart agriculture tasks	Improves precision and efficiency in agricultural tasks	Execution efficiency improved by ~20%, error rate reduced by 15%
Zhou et al.[13]	Hybrid architecture combining data-driven models and physical models for parameter identification and error characterization	Vehicle lateral dynamics modeling	Achieved accurate and interpretable modeling, providing a reliable model for vehicle motion control	Parameter identification error reduced by 10–15%
Chen et al.[14]	Spatial attention mechanism and feature fusion module, combined with hybrid models of physical and dual attention neural networks	Vehicle dynamics modeling	Improves modeling accuracy in complex dynamic environments with limited data	Prediction accuracy improved by ~12–18%
Meng et al.[15]	Mobile navigation combined with visual perception, with path planning and obstacle avoidance	Human-machine interaction	Achieved human-preference-based task assignment and efficient navigation	Target grasp success rate improved by 12%, task completion time reduced by 10%
Li et al.[16]	Centroid Voronoi tessellation (CVT)-based path segmentation method for brain-controlled robot navigation	Brain-controlled robot navigation	Provides efficient path segmentation for generating arbitrary target navigation objectives	Navigation success rate improved by 15%, path planning time reduced by 8%
Zhou et al.[17]	Wavelet decomposition and denoising autoencoder for UAV anomaly detection	UAV fault detection	Improved anomaly detection accuracy in noisy data conditions	Detection accuracy increased by ~15–20%
Chen et al.[18]	CSMA/CA-based fair and efficient MAC protocol, supporting multi-user MIMO parallel uplink transmission	Multi-UAV communication	Enhances communication efficiency and optimizes resource allocation	Uplink transmission efficiency improved by ~25%
Yin et al.[21]	Deep reinforcement learning algorithm for adaptive UAV navigation	3D UAV navigation environments	Proposed a new speed constraint loss function, improving UAV speed control capability	Path planning success rate improved by 20%, speed control error reduced by 30%
Liang et al.[24]	Integrated framework combining behavior decision-making, path planning, and motion control modules	High-speed mixed traffic scenarios	Improves safety of autonomous vehicles in mixed traffic environments	Collision rate reduced by 20%, path planning success rate improved by 18%
This study (EN-MASCA)	Introduced DQN and PPO algorithms with a virtual navigator model for path planning and obstacle avoidance	Complex environments in durian orchards	Significantly improves flight stability, path accuracy, and collaboration of UAV swarms	Path planning deviation reduced by 25% Altitude fluctuation reduced by 30% Task efficiency increased by 20%

Table 1. The comparison results of the EN-MASCA algorithm with some key studies.

It uses the PI control loop and control parameters to complete the control of s. Figure 1 shows the relationship between cluster algorithm, PX4 flight controller and PI control loop tie.

Simulation environment construction

This study also collected data from the various obstacles and terrain in the durian garden, used the height maps to create the real terrain and made 3D models of the durian garden and obstacles. It used GPS equipment and drone aerial photography to build the terrain and topography model of the durian garden. It used laser rangefinders and tape measures to measures the height, width and depth of obstacles, which can obtain the position coordinates (x, y, z) of obstacles in the actual environment. It also used the GPS equipment to determine the exact position of each obstacle and target, and took corresponding pictures in the center of the experimental area for model data correction[18]. Then this study used the GAZEBO's modeling tool to make the corresponding geometric model (such as trees, rocks, etc.) based on the manually measured geometric dimensions. It also used the surface texture and color information taken on-site to attach corresponding materials and maps to the geometric models, and used GAZEBO's material editor to make detailed adjustments. It also placed the geometric models in the corresponding positions in the virtual environment, used the measured coordinates for precise positioning, and set the physical properties of the models (such as hardness, reflectivity, etc.). It also introduces dynamic changing factors such as wind speed, wind direction and moving obstacles to simulate real operating scenarios. The durian garden area and surrounding environment patrolled by the drone group are shown in Fig. 2.

Multi-agent swarm control algorithm

The bio-clustering behavior is a natural phenomenon. It is a social behavior of biological groups to adapt the life. The survival ability of animals that have evolved in a long period, such as bird flocks and fish schools, which is based on the cohesion, separation and alignment[18]. The multi-agent swarm algorithm simulates the characteristics of biological swarms and the synchronous motion of autonomous systems composed of multiple particles[19]. Its rules are as follows:

- The agents moving in the system have a constant velocity s;
- Any pair of agents in the system has an influence radius ω. They only influence each other when the straight-line distance is less than ω.
- The movement direction of agents at each moment is consistent with the movement direction of all other intelligent. The average motion direction of energy bodies is the same.

In this model, the agent m has a constant velocity v, the displacement is $w_m(t)$, and the velocity direction of agent m is $U_m(t)$, which satisfies:

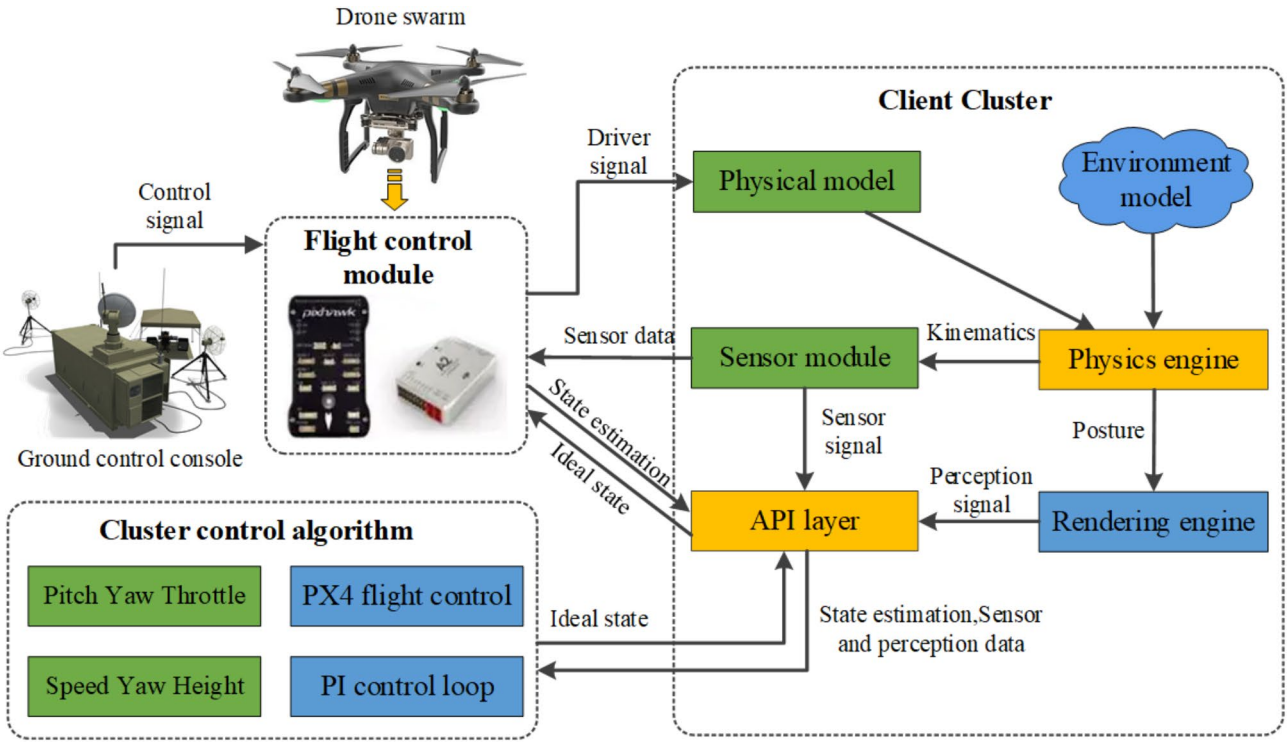

Fig. 1. The relationship between cluster algorithm, PX4 flight controller and PI control loop Tie.

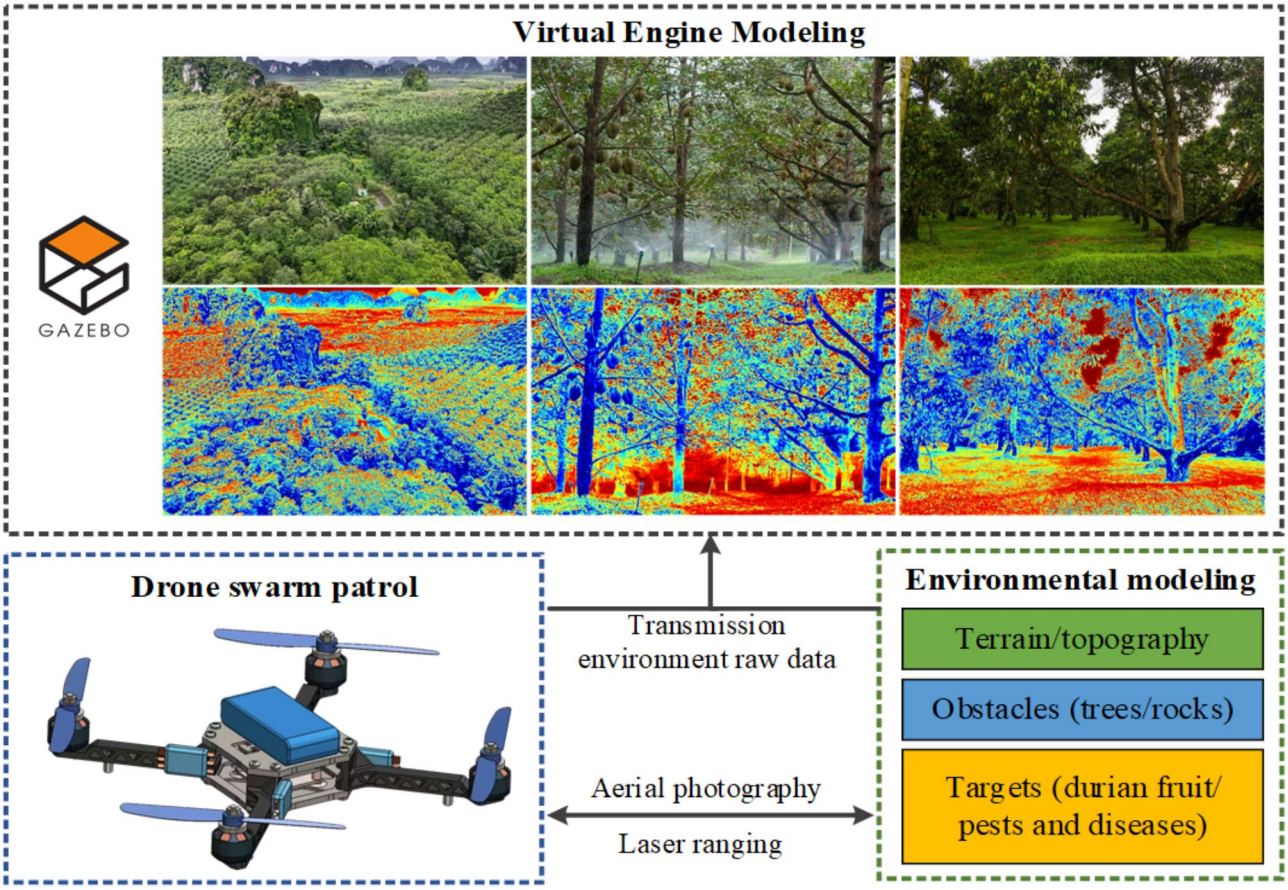

Fig. 2. The durian garden area and surrounding environment patrolled by the drone group.

$$\frac{\mathrm{dW}_m(t)}{dt} = \mathrm{V}_m(t) \tag{1}$$

$$\frac{\mathrm{dW}_m(t)}{dt} = \mu \sum_{n \in K, n \neq m} \mathrm{R}_{mn}\left(|\mathrm{W}_m(t) - \mathrm{W}_n(t)|\right) \times \left(\mathrm{V}_n(t) - \mathrm{V}_m(t)\right) \tag{2}$$

$$\mathrm{R}_{mn}(t) = \frac{H\left(|\mathrm{W}_m(t) - \mathrm{W}_n(t)|\right)}{K} \tag{3}$$

$$\mathrm{H}(\omega) = \frac{1}{\left(1 + |\omega|^2\right)^{\rho}} \tag{4}$$

In Formulas (1–4), $\mathrm{W}_m(t)$ and $\mathrm{W}_n(t)$ represent the displacement of agent m and n, $\mathrm{V}_m(t)$ and $\mathrm{V}_n(t)$ represent the speed of agent m and n; μ represents the constant, R_{mn} represents the mutual influence coefficient between agent m and n, K represents the constant, $H\left(|\mathrm{W}_m(t) - \mathrm{W}_n(t)|\right)$ represents the step size function; ω represents the influence radius, ρ represents a constant; According to the above rules, the patrol operation space of drone swarms in the durian garden is recognized as the three-dimensional Euclidean space. The dynamics of the intelligent agent is modeled as the second-order integral link, which is shown in Formula (5):

$$\begin{cases} \overline{a_m} = b_m \\ \overline{b_m} = c_m \end{cases}, \quad m = 1, 2, \ldots, \mathrm{K} \tag{5}$$

In Formula (5), a_m, b_m and $c_m \in D^k$ represent the position, velocity and control input of the m-th agent. The agent m can only communicate with the adjacent agents in the communication area. At time t, the set of adjacent agents is shown in Formula (6):

$$\mathrm{K}_m^{\varphi}(t) = \{n : |a_m - a_n| \leqslant \omega, \ \ n = 1, 2, \ldots, X, \ n \neq m\} \tag{6}$$

In Formula (6), $|a_m - a_n|$ represents the Euler distance, d represents the maximum interaction path or critical distance. The geometric model of cluster expectation requires that each agent is equidistant from all neighbors, which meets the following constraints:

$$|a_m - a_n| = \tau, \quad \forall m, n \in K_m(t) \tag{7}$$

In Formula (7), τ is a positive constant, which represents the minimum allowable distance or critical distance between each pair of adjacent agents and $\omega \leqslant d$. In the multi-obstacle environment, the input of each agent in the multi-agent control algorithm is divided into three parts[20]:

$$\mathrm{P}_m = P_m^x + P_m^y + P_m^z \tag{8}$$

In Formula (8), x, y and z uses the Olfati-Saber theory to represents three types of agents. Agent x represent any intelligent agent, Agent *y* is composed of the agent *x* projection on the obstacle surface, which represents the physical obstacle is avoided. Agent *z* is used to construct the navigation feedback, which represents the target to be tracked. P_m^x represents the interaction term (x, x), P_m^y represents the interaction term (x, y), P_m^z represents the distributed navigation feedback. P_m^x, P_m^y and P_m^z are defined as Formulas (9), (10) and (11):

$$P_m^x = -e_m^x \sum_{n \in \mathrm{K}_m^x} \mathrm{S}_\alpha(a_m)\,\mathrm{H}_\mathrm{x}(a_m) - P_m^x \sum_{n \in \mathrm{K}_m^\varphi} R_{mn}(a_m)(b_m - b_n) \tag{9}$$

$$P_m^y = -e_m^y \sum_{n \in \mathrm{K}_m^y} \mathrm{O}_{m,i}(a_m)\,\mathrm{H}_\mathrm{y}(a_m) - P_m^x \sum_{n \in \mathrm{K}_m^\varphi} \mathrm{O}_{m,i}(a_m)\left(b_m - \overline{b_{m,i}}\right) \tag{10}$$

$$P_m^z = -e_m^z \rho(a_m - a_z) - e_m^z(b_m - b_z) - e_m^z A_\varepsilon \tag{11}$$

In Formulas (9–11), e_m^x represents the constant, K_m^x represents the set of adjacent agents in the direction x , $\mathrm{S}_\alpha(a_m)$ represents the impact function, $\mathrm{H}_\mathrm{x}(a_m)$ represents the step function, R_{mn} represents the mutual influence coefficient between agents *m* and n, b_m and b_n represents the speed of agents *m* and n. e_m^y, e_m^z, A_ε and P_m^x represent constants; K_m^y represents the set of neighboring agents in the direction y, $\mathrm{O}_{m,i}(a_m)$ represents a constant; $b_{m,i}$ represents the speed of the virtual agent; $\rho(a_m - a_z)$ represents the distance function; a_z and b_z represents the position and speed of the virtual navigator. P_m^x represents the aggregated agent, which have two parts. The first part sets the distance between agents. The second part makes the agent's speed consistent with the speed of its neighbors. The body expression is as follows:

$$\mathrm{H}_\mathrm{x}(a_m) = \frac{L_{mn}}{\sqrt{1 + \alpha_x |L_{mn}|^2}} \tag{12}$$

$$L_{mn} = (a_m - a_n) - \frac{a_m - a_n}{|a_m - a_n|} \times \beta \tag{13}$$

$$\mathrm{S}_\alpha(a_m) = \frac{(|a_m - a_n| - \beta)^2}{\alpha} + 1 \tag{14}$$

In Formulas (12–14), α, e_m^x, e_m^y and e_m^z represent the constants, and the value of α is greater than β. The fragmentation is the Olfati–Saber clustering algorithm. The trap can effectively prevent fragmentation by introducing $\mathrm{S}_\alpha(a_m)$. When the distance between agents increases, the value $\mathrm{S}_\alpha(a_m)$ also increases rapidly. The second component of P_m^x is $L_{mn}(a_m) = \mathrm{S}_\alpha\left(\frac{|a_m - a_n|}{\omega}, a_n\right) \in [0, 1]$, $m \neq n$. $\mathrm{S}_\alpha(\gamma)$ is an impact function as shown as follows[21]:

$$S_\alpha(\gamma) = \begin{cases} 1 \\ 0.5 \times \left[1 + \cos\left(\pi \frac{(\gamma - \alpha)}{(1 - \alpha)}\right)\right] \\ 0 \end{cases} \tag{15}$$

In Formula (15), γ represents the input of impulse function, α represents the constant; P_m^z makes the agent track the virtual navigator or the desired trajectory, e_m^z, e_n^z and e_α^z are positive constants, a_z and b_z represent the position and speed of the virtual navigator, $\rho_1(a_i - a_n)$ and A_ε are shown as follows[22]:

$$\rho \times (a_m - a_n) = \frac{a_m - a_n}{\sqrt{1 + \alpha_z |a_m - a_n|^2}} \tag{16}$$

$$\mathrm{A}_\alpha = \begin{bmatrix} 0 \\ 0 \\ a_m^\alpha \end{bmatrix} - \begin{bmatrix} 0 \\ 0 \\ a_z^\alpha \end{bmatrix} \tag{17}$$

In Formulas (16) and (17), ρ represents the distance function, a_m and a_n represent the agents *m* and *n*. α_z represents a constant; a_m^α represents the height of the agent *m* and a_z^α represents the height of the virtual navigator. The purpose of A_α is to minimize the height difference between agents, which makes them track the height of the virtual navigator. a_m^α and a_z^α represent the height of the agent and the virtual navigator. P_m^y make the agent bypass obstacles, where e_m^y and e_n^y are positive constants. The virtual agent with position and velocity is constructed on the obstacle surface within the detection range of the agent ∂. The construction method is as follows:

(1) For the obstacle with a hyperplane boundary and a unit normal A_α, it passes through the point A_α. The position and velocity of the agent ∂ are determined by:

$$\overline{\mathrm{a}_{m,i}} = F \times a_m + (1-F)\,\mathrm{A}_\alpha, \qquad \overline{\mathrm{b}_{m,i}} = F b_m \tag{18}$$

In Formula (18), $\overline{\mathrm{a}_{m,i}}$ and $\overline{\mathrm{b}_{m,i}}$ represents the position and velocity of the agent ∂ , $F = \delta - a_i a_i^T$ is a projection matrix, A_α represents the height difference function.

(2) For the spherical obstacle with a radius of Q_i and at the centered position d_τ, the velocity of agent ∂ is shown as follows:

$$\overline{\mathrm{a}_{m,i}} = P \times a_m + (1-P)\,\mathrm{d}_i, \qquad \overline{\mathrm{b}_{m,i}} = P b_m \tag{19}$$

In Formula (19), P represents the projection matrix, d_i represents the center point of the obstacle, $P = \frac{\mathrm{Q}_i}{|a_m - \mathrm{d}_i|}$, $P = \frac{a_m - \mathrm{d}_i}{|a_m - \mathrm{d}_i|}$, and $P = \delta - a_i a_i^T$, so the virtual agent ∂ is constructed. It makes the individual cluster speeds consistent with the virtual agents, which remain consistent to maintain the certain distance, which is shown in Fig. 3.

The $\mathrm{O}_{m,i}(a_m)$ and $\mathrm{H}_\mathrm{y}(a_m)$ of P_m^y is defined as:

$$\mathrm{H}_\mathrm{y}(a_m) = \frac{a_m - \overline{a_{m,i}}}{\sqrt{1 + \alpha_y |a_m - \overline{a_{m,i}}|^2}} - 1 \tag{20}$$

$$\mathrm{O}_{m,i}(a_m) = \mathrm{S}_\alpha \times \left(\frac{|a_m - \overline{a_{m,i}}|}{\delta},\ \varepsilon_y \right) \tag{21}$$

In Formulas (20) and (21), $\mathrm{H}_y(a_m)$ represents the step size function; $\overline{a_{m,i}}$ represents the projection of the agent a_m on the obstacle surface, α_y represents a constant; $\mathrm{O}_{m,i}(a_m)$ represents the influence function; S_α represents the impact function, ε_y represents the maximum detection distance of the drone relative to the obstacle, α_y represents the positive constant, δ represents the maximum detection distance of the drone relative to the obstacle.

Improved multi-agent swarm control algorithm

In order to improve the performance of the multi-agent cluster control algorithms, this study introduces DQN (Deep Q Network)[23], which is a reinforcement learning algorithm that combines the Q-learning and neural networks. It learns the behavior value function corresponding to the optimal strategy and the target value of the Q-learning algorithm by minimizing the loss function. The memory and target network factors make the performance of DQN more powerful. Its state includes the current position of the drone, its speed, the position of neighboring obstacles, the position of neighboring drones, and the current position of the virtual navigator. Its actions include adjusting the flight direction (angle change) and flight speed. The reward function is shown in Formula (22), as follows:

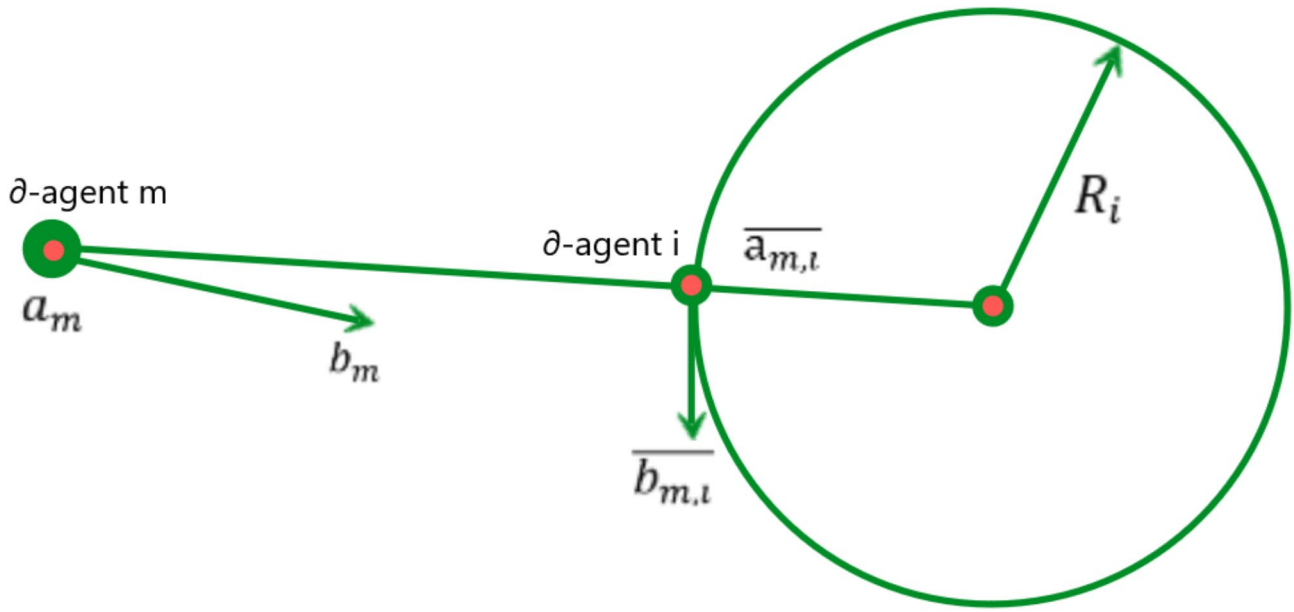

Fig. 3. The position and velocity of the agent ∂.

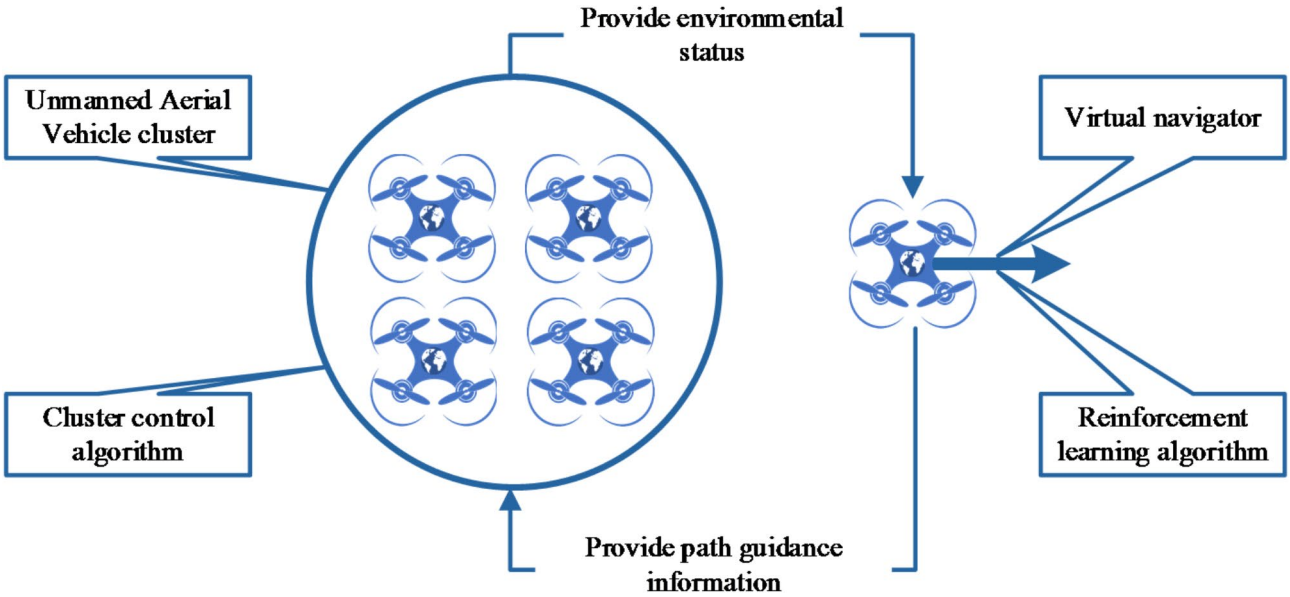

Fig. 4. The obstacle avoidance algorithm control system of drone swarms.

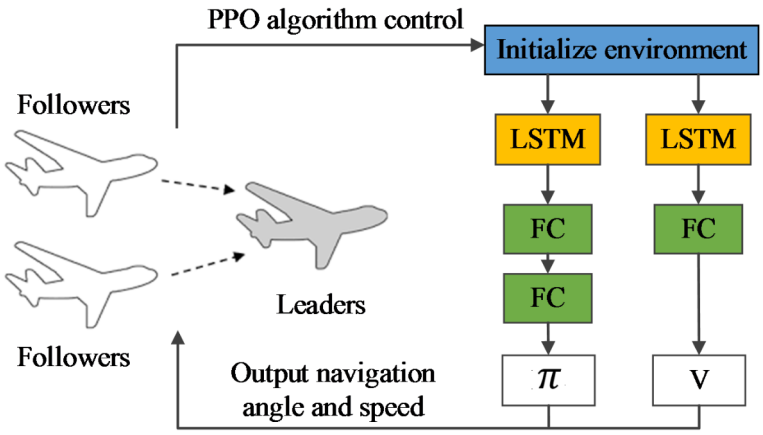

Fig. 5. The network structure of PPO algorithm.

$$D = D_{approach} - D_{obstacle} + D_{cluster} \tag{22}$$

In Formula (22), $D_{approach}$ represents the reward for shortening the distance between the drone and the target point; $D_{obstacle}$ represents the penalty for approaching obstacles; $D_{cluster}$ represents the reward for maintaining an appropriate distance and formation with neighboring drones. Based on the reward mechanism of the above cruise path, it controls the virtual navigator to avoids obstacles and navigates, which receives the cluster's detection information of the environment and forms an interactive network with the drone cluster. The phase feedback makes drone swarms adapt to the complex and changing environments, as shown in Fig. 4.

This stud uses the PPO algorithm to control the virtual navigator[24]. As the guide of the drone cluster, the virtual navigator adjusts the patrol path according to environmental information and provides dynamic reference points for the cluster. This study first uses the DQN algorithm to adjust the position of the virtual navigator so that it can adapt to the dynamic environment and avoid obstacles in real time. It also broadcasts the virtual navigator's target information to each drone through real-time communication to ensure that the cluster can respond quickly to environmental changes. Its network structure is shown in Fig. 5. It has two networks; the Critic network architecture first processes the input LSTM layer with 128 hidden units; the fully connected layer of 128-bit hidden units uses the TanH layer as its activation function[25]. The FC layer consists of hidden layer units and TanH activation function. The Actor network is composed of the neural network and normal distribution, which has the 128-unit LSTM, FC and TanH layer. Its output is the mean of a normal distribution and the variance matrix is $C = 0.05\ I$, where I is the identity matrix and the behavior is obtained from this distribution. The output of the Actor is to obtain the velocity vector of navigators. So this study designs the output of the Actor-network, which makes the projection of the ball radius on the three-dimensional axis as the

velocity vector. The output of the Actor-network is assumed as the radius of the ball σ and two angles (τ, μ). τ is the angle between the sphere radius σ and the z-axis, μ is the angle between the radius projection on the plane x, y and the axis x, which obtains the velocity vector $[\mathrm{G}\cos\tau,\ \mathrm{G}\sin\tau\sin\mu,\ \mathrm{G}\sin\tau\cos\tau]$. Considering that the drone's speed σ will be limited to [30.0 m, 50.0 m], The angle is limited to $[-\pi\ rad,\ \pi\ rad]$, so the average value of the radius and angle use TanH as the activation function.

In this study, the navigator is the point mass consisting of the position and velocity vector. It enables the virtual navigator to approach the target area; the swarm can avoid obstacles and follow the virtual navigator to reduce the distance between the navigator and swarm[26]. The construction reward function is as follows:

$$G = G_{obstacle} + G_{leader} + G_{center} \tag{23}$$

In Formula (23), G_{leader} rewards the leader for getting closer to the destination, and G_{center} is rewards the cluster center for shrinking the distance from the leader. The distance between the navigators $G_{obstacle}$ is rewards the cluster center for avoiding obstacles. Its network input is the navigator's position, the drone cluster's center position. The distance vector between it and the obstacle, and the output is the velocity vector of the navigator. In order to achieve cooperative behavior of drone clusters, this study improved the multi-agent control strategy. Distributed control is achieved between drones through local communication, avoiding excessive reliance on centralized control. It also dynamically adjusts parameters by calculating the relative distance and speed difference between agents to ensure that the cluster maintains an appropriate formation during patrols. Finally, based on the group behavior model of Olfati-Saber, it introduces the gradient descent method to dynamically adjust the flight path to avoid collisions between drones and obstacles. By combining the DQN and PPO algorithms, this study uses the target network and experience replay mechanism to reduce the instability of model training, and uses parallel computing to accelerate the execution of path planning and obstacle avoidance strategies. It also uses the Bayesian optimization method to dynamically adjust parameters such as learning rate and discount factor to improve the robustness of the algorithm in complex environments.

Enhanced multi-agent cluster model construction

Before building the model, this study set the learning rates of the DQN and PPO algorithms to 0.001 and 0.0003. The learning rate of DQN is mainly based on the convergence requirements of the algorithm in a dynamic environment, and a higher learning rate is selected to accelerate initial learning. The learning rate of PPO is based on empirical values and experimental tuning results, and a lower learning rate is selected to avoid oscillation or instability caused by too fast gradient updates in complex scenarios. This study also set the batch size of DQN and PPO to 64 and 128. The smaller batch size of DQN is intended to improve the real-time update capability of the model, while the larger batch size of PPO helps to enhance the adaptability to complex environments and training stability. The discount factor is used to weigh short-term rewards and long-term benefits. This study set it to 0.99, highlighting the overall performance of the drone swarm in long-term patrol missions and retaining the effectiveness of short-term path optimization. The target network of the DQN algorithm is updated every 100 iterations. Based on the consideration of model training stability, it delays the update of the target network to avoid training oscillations caused by frequent updates. The PPO algorithm uses gradient clipping technology and sets the gradient clipping threshold to 0.2 to ensure that the gradient update does not exceed a reasonable range, thereby improving the stability of the model.

In order to verify the effectiveness and correctness of the algorithm, this study also built a multi- drone motion simulation model and uses the four-degree-of-freedom 8-state drone dynamics model[27]. It contains 8 state variables $[\mathrm{s1},\ \mathrm{s2},\ \mathrm{s3},\ \mathrm{s4},\ \mathrm{s5},\ \mathrm{s6},\ \mathrm{s7},\ \mathrm{s8}]$ and four input variables: the aileron deflection command $\emptyset_a$, elevator and rudder deflection command $\emptyset_b$, rudder angle command $\emptyset_c$ and throttle command $\emptyset_d$. In order to control the drone swarms, this study uses three cluster control quantities: speed V_i, yaw angle R_i and altitude H_i. The dynamic model of the drone with attitude control capability is controlled through three cluster control quantities. The output command of the cluster algorithm is converted into the input of the attitude control loop in the drone dynamic model. The PI control loop needs to be built as the transition. The relationship between the cluster algorithm, PI control loop, attitude control loop and state dynamics model is shown in Fig. 6. In the process of model implementation, it first constructs a four-degree-of-freedom drone dynamics model. By defining the drone's 8 state variables and 4 input variables, the drone dynamics equation is established to describe the relationship between the state variables and the input variables. Then it designs the PI control loop by setting the proportional gain (K_p) and integral gain parameters (K_i) of the PI controller, which inputs the output command of the cluster algorithm, calculates the error (*e*) and the integral of the error ($\int \mathrm{edt}$), and calculates the control output through the formula $\mathrm{u}(\mathrm{t}) = \mathrm{K}_p\mathrm{e}(\mathrm{t}) + \mathrm{K}_i \int \mathrm{e}(\mathrm{t})\,\mathrm{dt}$. Finally, the attitude control loop is implemented. It converts the output command of the PI controller into the attitude control command of the drone to control the pitch angle, yaw angle and throttle, which achieves the precise control of the drone movement.

Experimental result

In order to verify the performance of the EN-MASCA algorithm, it compares with the unimproved MASCA (multi-agent swarm control algorithm), NNCA (Nonlinear Neural Control Algorithm)[28], and NSGAII (Non-dominated Sorting Genetic Algorithm II)[29] algorithm. The results are as follows:

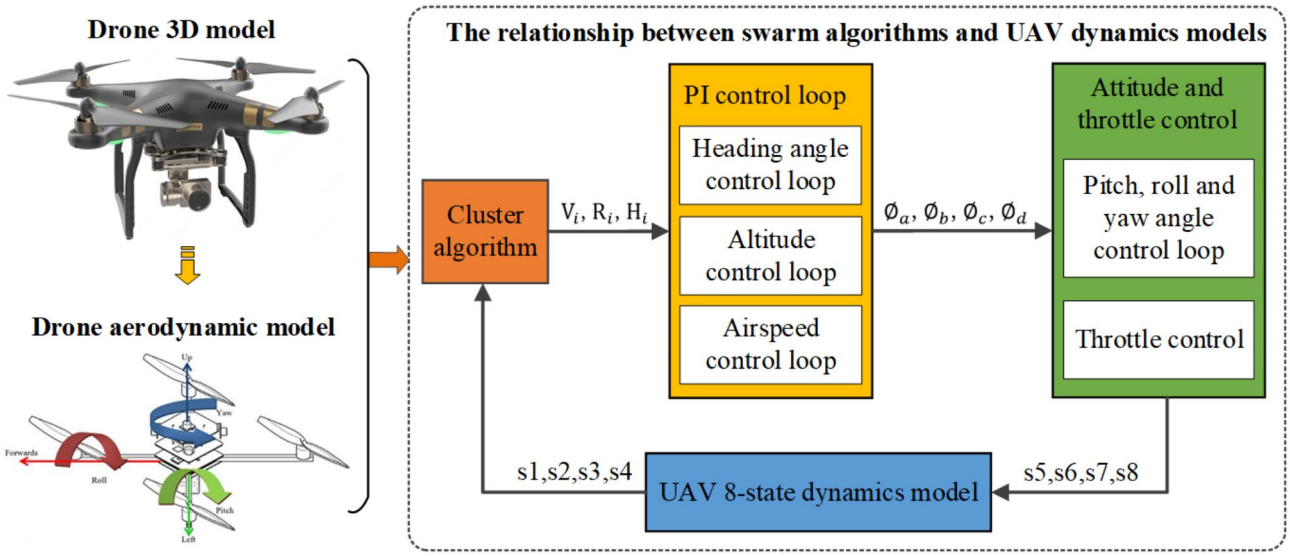

Fig. 6. The relationship between swarm algorithm, PI control loop, attitude control loop and UAV 8-state dynamics model.

Flight traces

Figure 7 shows the obstacle avoidance flight routes controlled by four algorithms. The drone1 controlled by the MASCA algorithm deviated from the drone swarm 1.874 m at the mileage of 1500 m, which shows the significant deviation. The maximum height difference occurred at 2250 m, and the height difference between drone2 and drone5 reached 3.495 m, which indicates that its flight altitude fluctuated with a large amplitude. The drone1 controlled by the NNCA algorithm deviated from the drone swarm 1.690 m at the mileage of 750 m, but then gradually returned to the cluster. The maximum height difference between drone3 and drone5 reached 2.473 m at the mileage of 1500 m, which indicates that its height change amplitude is relatively large. The drone2 controlled by the NSGAI algorithm deviated from the cluster by 1.744 m at the mileage of 1500 m, showing a certain deviation; the height difference between drone3 and drone4 reached 2.431 m at the mileage of 2000 m, which shows that its height change had obvious periodicity and moderate fluctuation amplitude. The drone6 controlled by the EN-MASCA algorithm deviated from the drone swarm 0.781 m at the mileage of 500 m but quickly returned to the cluster. Its maximum height difference between drone3 and drone6 is 1.524 m at the mileage of 1500 m, indicating that its altitude was the most stable, with the smallest height difference and deviation, showing good cluster consistency.

Flight stability

Figure 8 shows the changes in speed, yaw angle, height change rate and relative distance of the drone swarm controlled by four algorithms. The average speed of the EN-MASCA algorithm is 9.214–11.315 m/s, and the expected value is 10 m/s. Compared with another three algorithms, which is 7.624–12.990 m/s, its maximum and minimum values are reduced by 14.80% and 10.59%. The average yaw angle of the EN-MASCA algorithm is −0.705 to 1.929 rad, and the expected value is −0.845 to 2.092 rad. Compared with another three algorithms in the range of −2.404 to 2.674 rad, its maximum and minimum values are reduced by 38.61% and 86.38%. The average relative distance of the EN-MASCA algorithm is 1.777–5.357 m, and the expected value is 4 m. Compared with another three algorithms in the 2–6 m range, its maximum and minimum values are reduced by 37.99% and 14.86%. The average height change rate of the EN-MASCA algorithm is 0.550–1.597 m/s, and the expected value is 1.0 m/s. Compared with another three algorithms in the range of 0.394–3.739 m/s, its maximum and minimum values are reduced by 134.13% and 4.06%. The above results show that the speed of drone swarms controlled by the EN-MASCA algorithm quickly converges to the expected flight speed after fluctuating within the allowable range, which has a good speed tracking effect. After fluctuating, the yaw angle quickly converges to the expected yaw angle; the height change rate shows it can track the leading. The navigator has less fluctuation in tracking the navigator and avoiding the tracking delay of the pitch angle. During the whole flight, the distance between the drones is the minimum relative distance, which is always greater than the minimum distance to avoid collision between drones.

Cluster stability

Figure 9 shows the changes in the distance between the cluster center and virtual navigator, the distance between the cluster center and obstacle, the cluster roll angle and the yaw angle controlled by four algorithms. The average distance between the cluster center and obstacle of the EN-MASCA algorithm is between 12.171 and 16.700 m, and the expected value is 15 m. Compared with another three algorithms in the range of 11.033 and 21.624 m, its maximum and minimum values are reduced by 29.48% and 5.12%. The average distance between the cluster center and virtual navigator of the EN-MASCA algorithm is between 8.106 and 11.915 m, and the expected value

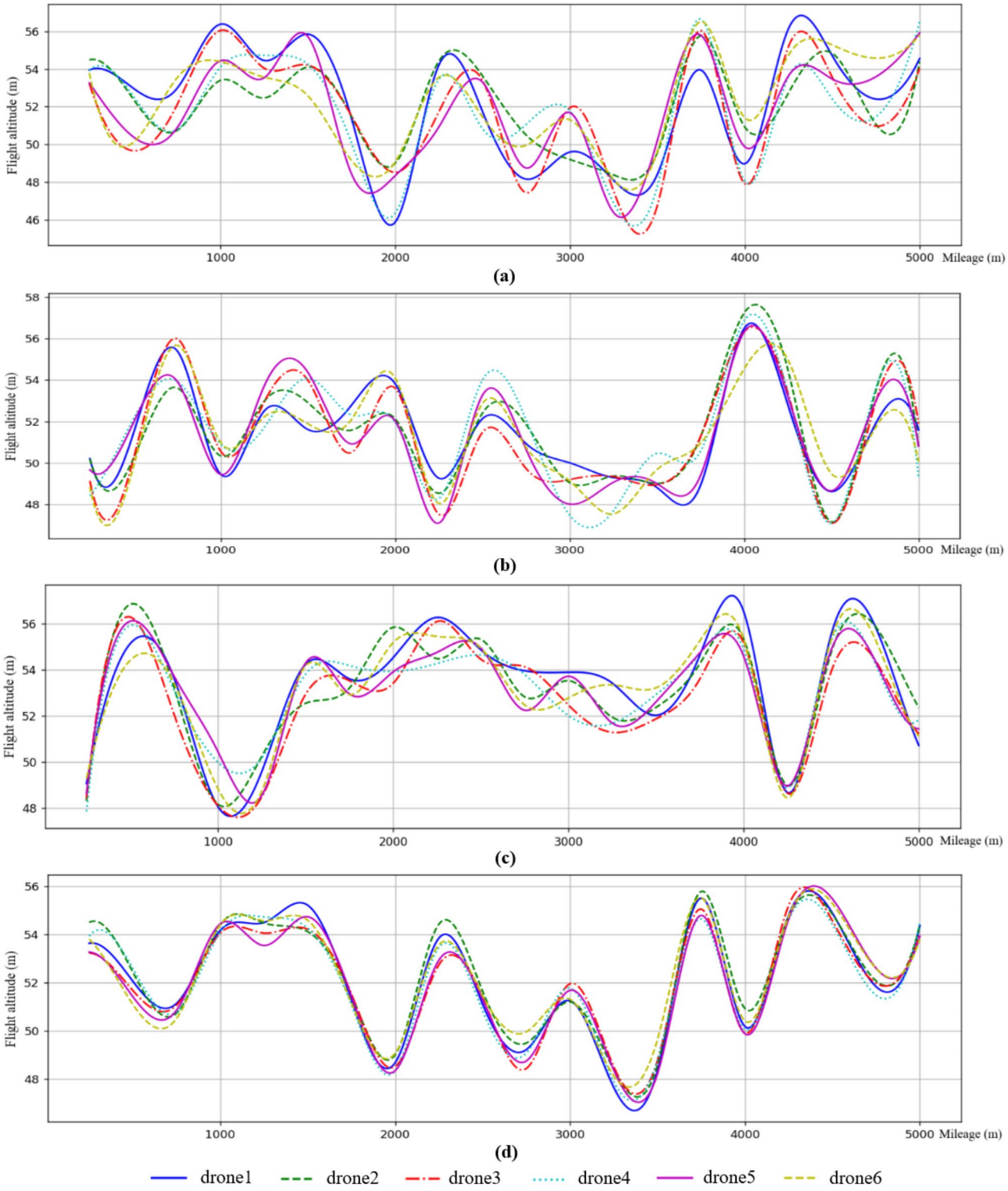

Fig. 7. The height variation of unmanned swarms controlled by four algorithms. (**a**) MASCA, (**b**) NNCA, (**c**) NSGAI, (**d**) EN-MASCA.

is 10 m. Compared with another three algorithms in the range of 6.014–15.847 m. Its maximum and minimum values are reduced by 33.02% and 19.19%. The average roll angle of the EN-MASCA algorithm is 8.95–14.87 rad, and the expected value is 12 rad. Compared with another three algorithms in the range of 6.12–17.78 rad, its maximum and minimum values are reduced by 22.85% and 31.97%. The average pitch angle of EN-MASCA is 16.952–22.959 rad, and the expected value is 20 rad. Compared with another three algorithms in the range of 12.041–29.754 rad, its maximum and minimum values are reduced by 29.60% and 5.49%. These results show

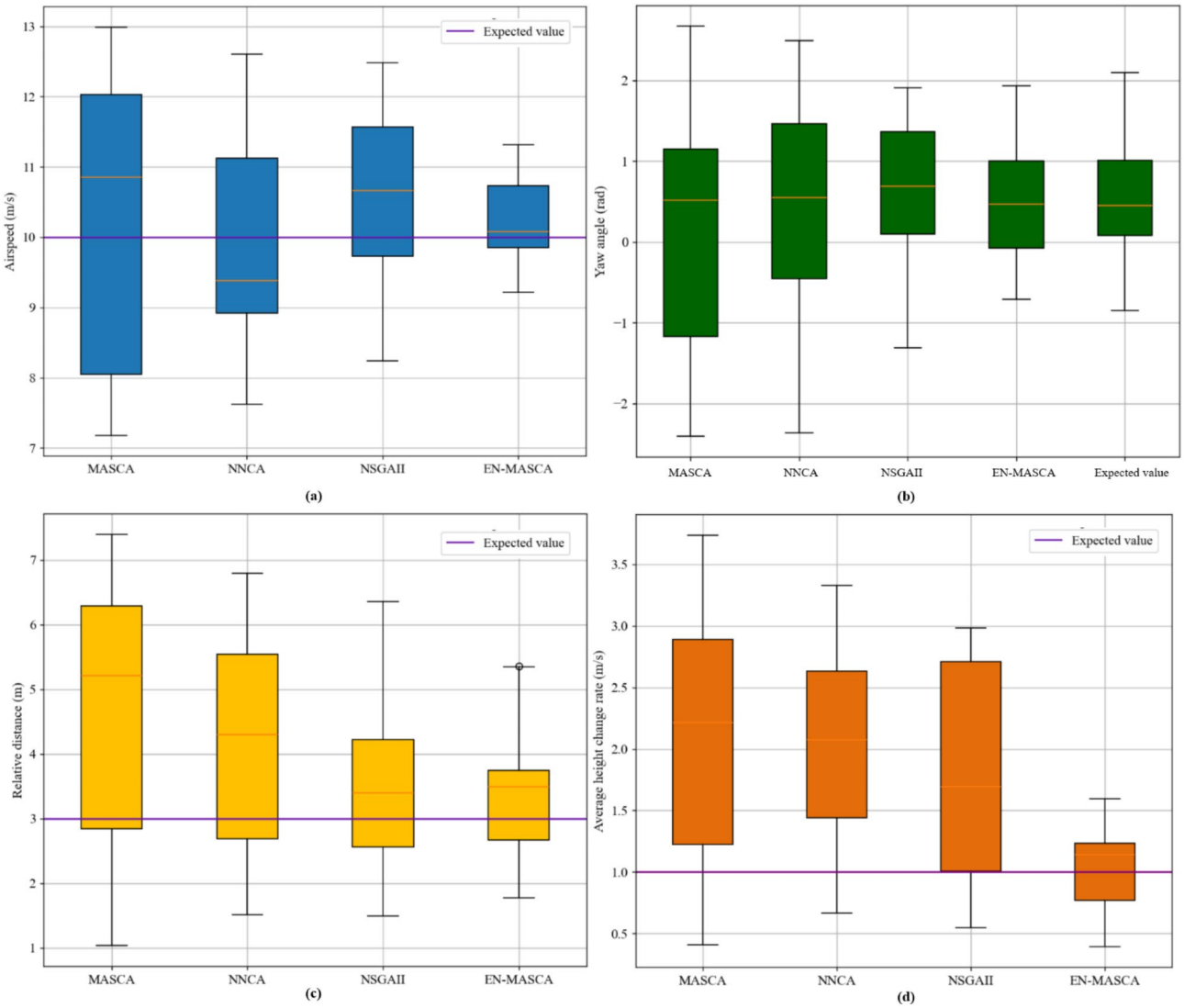

Fig. 8. Changes in the navigation speed, altitude change rate, yaw angle, and relative distance of the unmanned swarm controlled by the four algorithms. (**a**) Average navigation speed, (**b**) Average yaw angle, (**c**) Average relative distance, (**d**) Average altitude change rate.

that the EN-MASCA algorithm performs better in controlling the navigation of drone swarms, with a smaller fluctuation range and closer to the expected value. Because the virtual navigator is trained through the PPO algorithm, its network structure is complex and has strong learning ability. The navigator receives environmental information to adjusts its behavior by the reward function. The reward function has multiple aspects, such as the navigator's approach to the target area, the distance between the cluster center and the navigator and the cluster center avoiding obstacles, which ensures the navigation effect of the navigator so that the entire cluster can complete the task more effectively. It also introduces the projection of the virtual navigator and obstacles, optimizes the control input of the cluster and enables the agent to better follow the navigator and avoid obstacles.

Simulation effect

Figure 10a–c shows the GAZEBO simulation navigation performance of the drone swarm controlled by the EN-MASCA algorithm. The starting and ending points of the drone swarm are (1, 9, 0.15) and (15, 1, 0.1), which is marked by red dots and triangles. The positions of obstacles 1–6 are (1, 6, 0.5), (4, 5, 0.6), (4.2, 6.5, 0.6), (7, 7.5, 0.6) and (10, 3, 0.8). The drones start from the starting point. First, the drone swarm bypasses obstacle 1 and passes through it from its right side. Then the drone swarm sailed to obstacles 2 and 3. The two obstacles are closed in position, and the drone swarm bypassed the upper and lower paths to avoid the obstacle area. Then, the drone swarm sailed to obstacle 4. The obstacle is high, and the drone swarm adjusts the path in front and chose the safe bypass route. Before approaching the end, the drone swarm avoided obstacle 5 and passed from its right side. Finally, the drone swarm bypassed all obstacles and reached the end. During the entire navigation process, the drone swarm constantly adjusted the flight altitude and direction to adapt to terrain changes and avoid obstacles. Figure 10d shows the EN-MASCA algorithm iteration in the simulated navigation.

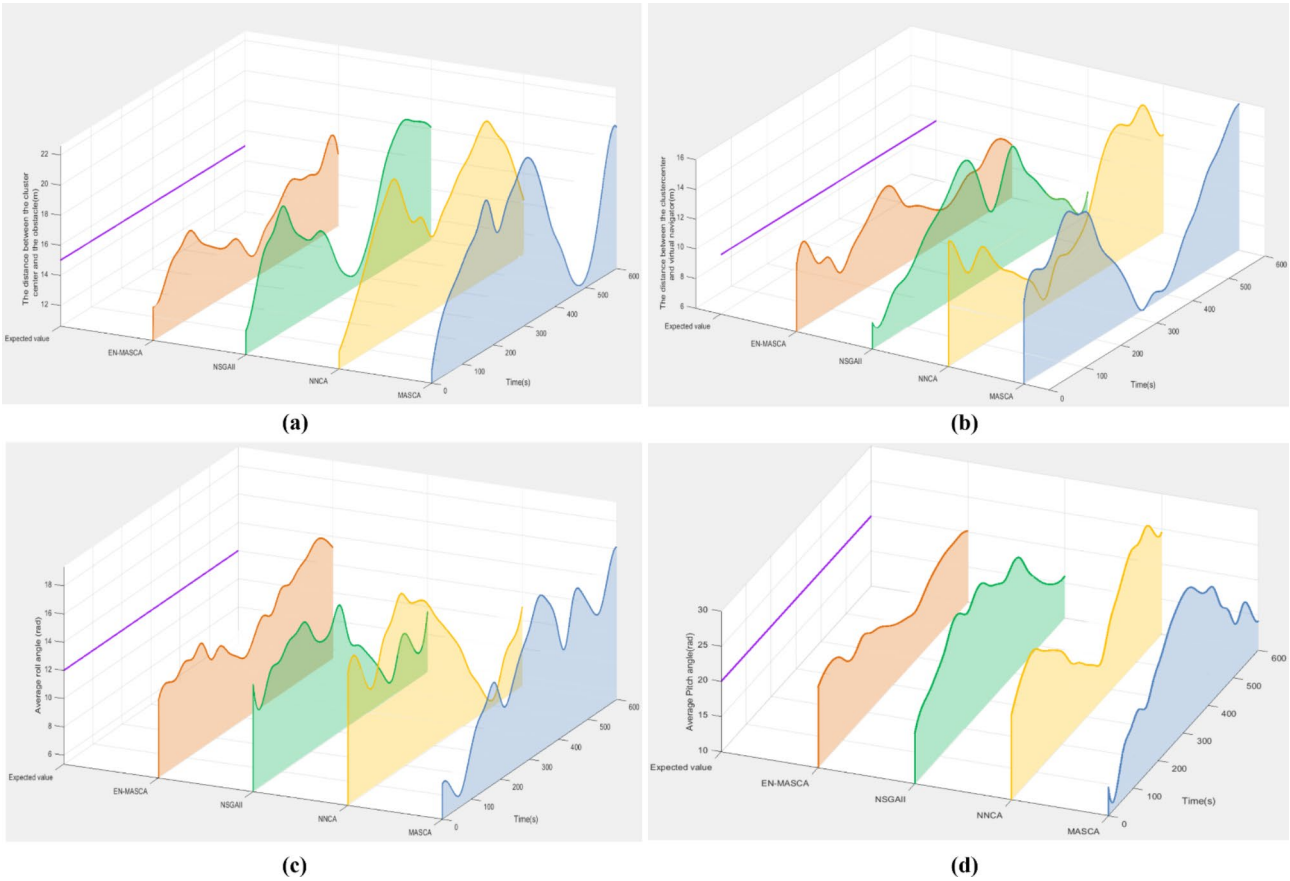

Fig. 9. The control changes under four algorithms in the distance between the cluster center and obstacles, the distance between the cluster center and virtual navigator, and the cluster roll angle and pitch angle. (**a**) Average distance between the cluster center and obstacles, (**b**) Average distance between the cluster center and virtual navigator, (**c**) Average roll angle, (**d**) Average pitch angle.

After 200 iterations, the objective function value tends to be stable, which indicates that the navigation path of drone swarms converges to the optimal path during the optimization process, which avoids all obstacles and meets the requirements of the shortest path and minimum energy consumption. It shows the obstacle avoidance path, terrain adaptability and iterative optimization process of the EN-MASCA algorithm controlling drone swarms from multiple perspectives, which indicates that it can find the optimal path in changing terrains and its excellent performance in a complex environment.

The GAZEBO algorithm path is imported into the Rflysim3D software[30] for demonstration. The Rflysim3D environment is planned as the simulation environment, and the drones' flight altitude and speed are set to be consistent. Five drones leave at the initial position, and the departure position is consistent with the position in GAZEBO. As shown in Fig. 11, the drone group's flight path is consistent with the path simulated by GAZEBO. Although drones 2 and 4 fly in another direction at the beginning, they pass through obstacles 1–3 after adjustment and return to the flight path of drone 1 (navigator). Finally all the drone groups fly to the end. It shows that the drone group can fly according to the planned path and complete the cruising process.

Discussion

This study is consistent with Zhou et al.[17] and Yin et al.[21] in terms of path planning and obstacle avoidance, but by introducing a virtual navigator and reinforcement learning algorithm, it achieves higher accuracy and greater adaptability. improvements. Compared with the studies of Meng et al.[15] and Chen et al.[18], this study optimized the cluster collaboration mechanism, enabling drones to maintain consistency and stability in complex agricultural scenarios. Combining the smart agriculture results of Lu et al.[11] and Li et al.[12], this study focuses on the specific application scenario of durian orchards and designs a more practical optimization strategy. Compared with the dynamic modeling studies of Zhou et al.[13] and Chen et al.[14], this study shows stronger robustness in dynamic environments. Based on Yin et al.[21] and Ji et al.[23], this study significantly improved the efficiency and real-time performance of the algorithm through distributed control strategy and experience replay mechanism. Through the above experimental results, the EN-MASCA algorithm performs better than other algorithms in multiple dimensions such as flight trajectory deviation, altitude change stability, and flight speed range, demonstrating good accuracy and stability. In terms of relative distance consistency and safe distance control between cluster center and obstacles, EN-MASCA significantly outperforms MASCA, NNCA and NSGAII, indicating that it is

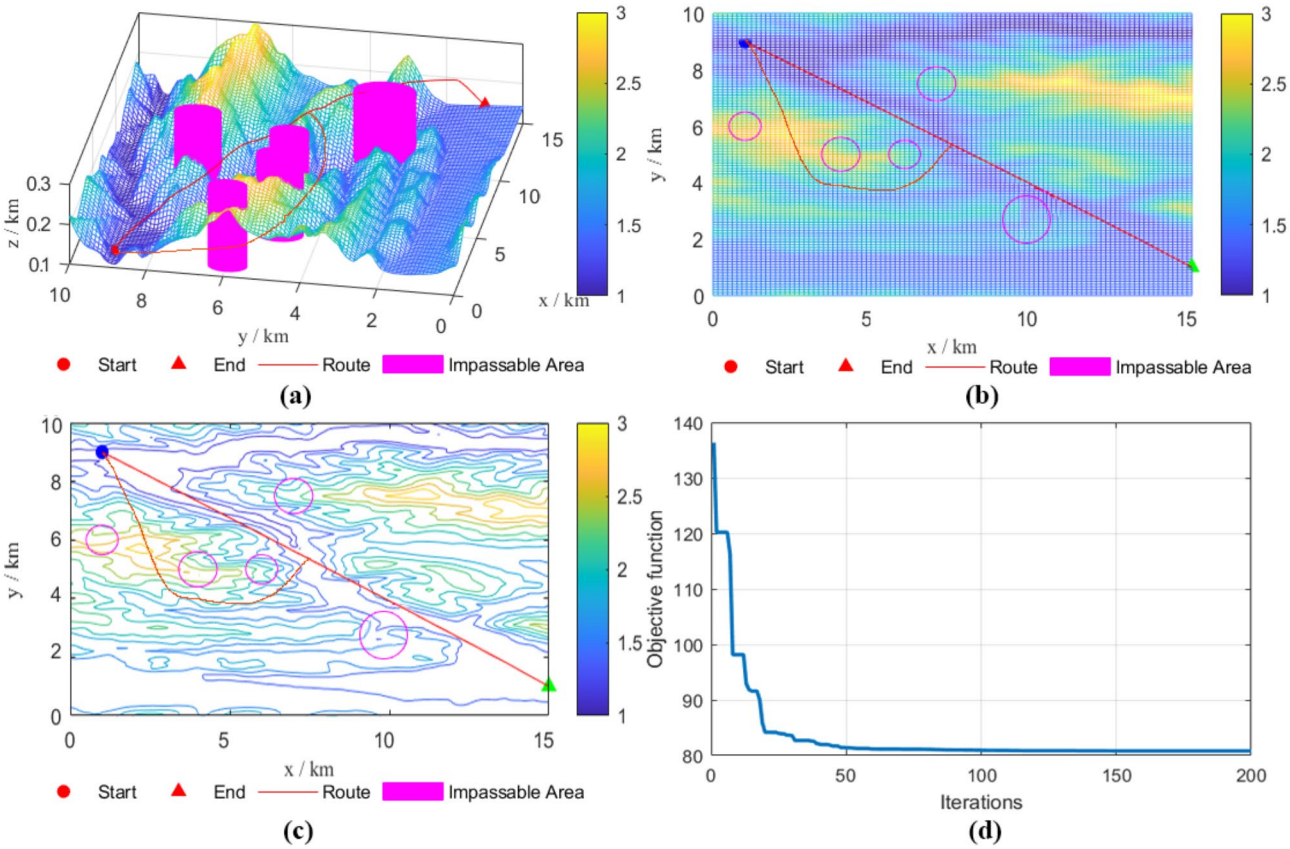

Fig. 10. GAZEBO simulation navigation effect and iteration of EN-MASCA algorithm controlling UAV swarm (**a**) 3D navigation trajectory map, (**b**) Two-dimensional navigation plan, (**c**) Contour navigation chart, (**d**) Performance grap.

more suitable for clustering tasks in the complex environments. It achieves fast path adjustment and efficient navigation performance by introducing a virtual navigator model and reinforcement learning algorithms (DQN and PPO), especially in complex scenarios. Table 2 shows the comparison results between the EN-MASCA and another two algorithms.

Although this study simulated the basic terrain characteristics of a durian orchard, there are some limitations compared to the real orchard scene. The terrain undulations and obstacle distribution in the simulated environment are designed based on average values and typical samples, and fail to fully cover the extreme terrain changes that may occur in the orchard (such as steep slopes and deep gullies). There are irregularly distributed soft soils or waterlogged areas in the real environment. These factors put higher requirements on the flight control and path planning of the drone, which have not been fully reflected in the simulation. Although moving obstacles were added to the experiment to simulate a dynamic environment, the types and behavior patterns of dynamic obstacles are relatively limited. The moving obstacles in this experiment used a fixed speed and a simple linear motion model, while nonlinear and irregular motion (such as random movement of people, animals, and mechanical equipment) may occur in reality. In order to narrow the gap between the experimental setting and the real conditions, subsequent research will build a more complex and realistic simulation environment, including more sophisticated terrain and obstacle modeling. It introduces experimental designs with multiple scenarios and multiple meteorological conditions to comprehensively evaluate the robustness and adaptability of the algorithm. It also conducts long-term field tests to accumulate real data to optimize the algorithm performance.

Conclusions

This study proposes an enhanced multi-agent swarm control algorithm (EN-MASCA) for the coordinated patrol operation of drone swarms in durian orchards. It introduces DQN and PPO algorithms to optimize drones' navigation and obstacle avoidance strategies. It guides the drone swarm through the virtual navigator model to improve its adaptability and stability. It also constructs a six-degree-of-freedom drone motion simulation model and uses the PI control loop to achieve the attitude control of drone swarms. Compared with MASCA, NNCA and NSGAII algorithms, the results show that the EN-MASCA algorithm is superior to the other three algorithms regarding flight trajectory, flight stability, cluster stability and simulation effect. The drone swarm controlled by the EN-MASCA algorithm can effectively avoid obstacles, maintain a tight formation and complete patrol tasks. Its speed, altitude and yaw angle change rate are closer to the expected value and have less fluctuation. The distance between the cluster center and the virtual navigator or obstacles remains stable,

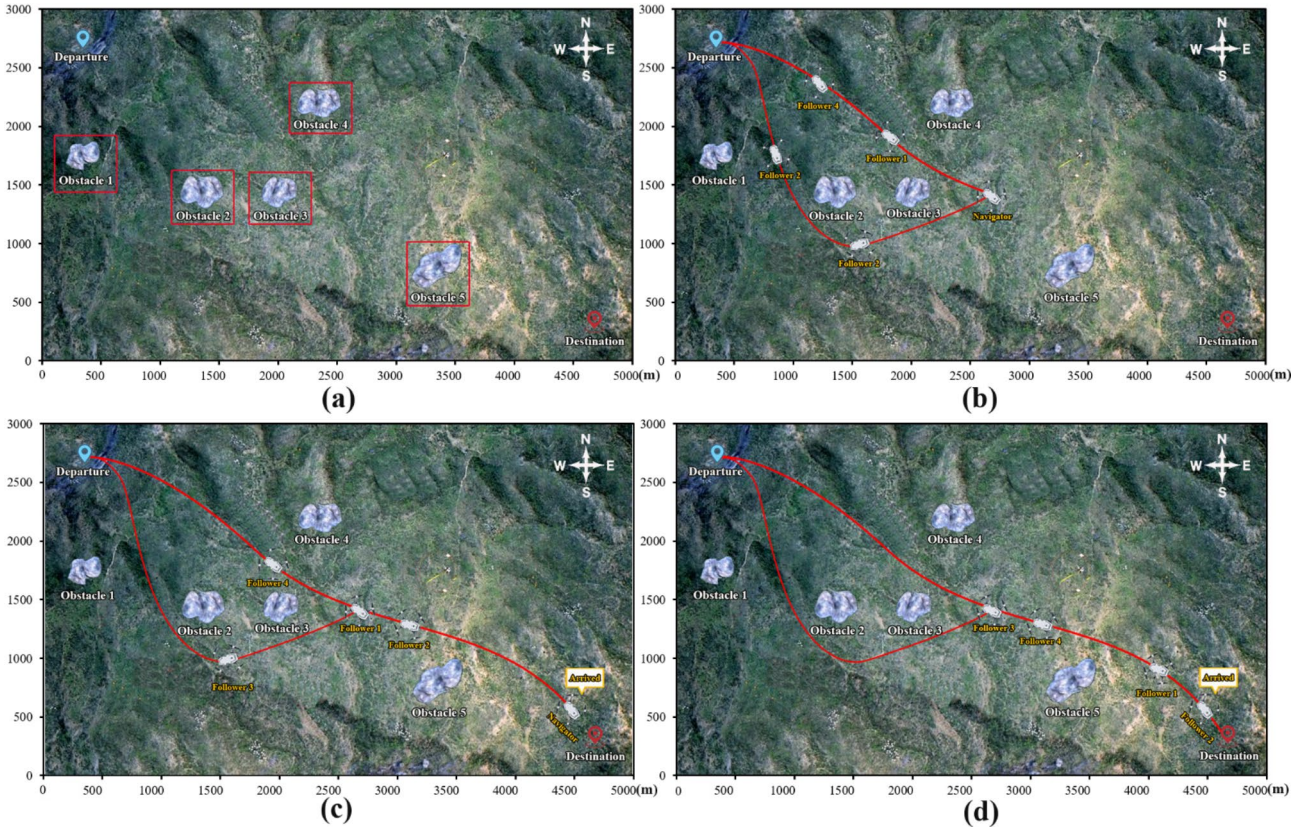

Fig. 11. 3D simulation effect of EN-MASCA algorithm controlling drone swarm (**a**) Obstacle distribution map, (**b**) Initial navigation path diagram, (**c**) Obstacle avoidance navigation path map, (**d**) Optimal navigation route map.

Comparison dimension	MASCA	NNCA	NSGAII	EN-MASCA (Proposed Algorithm)
Flight path deviation	Max deviation of 1.874 m, significant path deviation	Max deviation of 1.690 m, moderate adjustment ability	Max deviation of 1.744 m, relatively low deviation	Max deviation of 0.781 m, minimal deviation, quick return to the cluster
Altitude stability	Max height difference of 3.495 m, large fluctuations	Max height difference of 2.473 m, moderate fluctuations	Max height difference of 2.431 m, periodic fluctuations	Max height difference of 1.524 m, smallest fluctuations, highly stable
Flight speed range	7.624–12.990 m/s, large fluctuations	8.020–11.892 m/s, moderate fluctuations	7.856–12.314 m/s, relatively small fluctuations	9.214–11.315 m/s, stable speed, close to expected value
Relative distance	Average relative distance of 2–6 m, inconsistent	Average relative distance of 2.1–5.7 m, moderately consistent	Average relative distance of 2.0–5.5 m, relatively consistent	Average relative distance of 1.777–5.357 m, highly consistent
Cluster center to obstacle distance	Average of 11.033–21.624 m, large fluctuations	Average of 11.570–19.341 m, moderate fluctuations	Average of 11.846–18.735 m, relatively stable	Average of 12.171–16.700 m, stable, close to the expected value
Navigation performance	Delayed navigation, slow path adjustments	Moderate navigation response, reasonable adjustment speed	Faster adjustments but insufficient precision	Fast response, precise path adjustments, adaptive to complex environments

Table 2. The comparison results between the EN-MASCA and another two algorithms.

which ensures the safety and stability of patrol operations. It enables drone swarms to learn and optimize flight paths, avoid collisions and misjudgments and complete large-scale patrol tasks quickly, which reduces the labor intensity and costs of manual patrols and improves patrol efficiency and safety. Moreover, it enables the drone to respond quickly, detect abnormal conditions and send alerts to managers promptly, which provides the accurate location information to help managers respond quickly, ultimately improving the economic benefits of the durian orchard.

Data availability

The datasets generated and analyzed during the current study are not publicly available due to [privacy concerns] but are available from the corresponding author upon reasonable request. To request access to the data, please contact [Ruipeng Tang] at [22057874@siswa.um.edu.my]. Access may be provided contingent upon compliance with any necessary data-sharing agreements and approval for use in line with the study's terms.

Received: 25 November 2024; Accepted: 24 January 2025
Published online: 17 March 2025

References

1. Wiangsamut, B. & Wiangsamut, M. E. L. Assessment of natural fruit ripening and fruit quality of three elite durian cultivars for overland export. *Trends Sci.* **20**(5), 4647–4647 (2023).
2. Lim, J. A. et al. Mitigating the repercussions of climate change on diseases affecting important crop commodities in Southeast Asia, for food security and environmental sustainability—A review. *Front. Sustain. Food Syst.* **6**, 1030540 (2023).
3. Rakesh, D. et al. Role of UAVs in innovating agriculture with future applications: A review. In *2021 International Conference on Advancements in Electrical, Electronics, Communication, Computing and Automation (ICAECA)* 1–6 (IEEE, 2021).
4. Kappel, K. S., Cabreira, T. M., Marins, J. L., de Brisolara, L. B. & Ferreira, P. R. Strategies for patrolling missions with multiple UAVs. *J. Intell. Robot. Syst.* **99**, 499–515 (2020).
5. Dai, M., Luo, L., Ren, J., Yu, H. & Sun, G. PSACCF: prioritized online slice admission control considering fairness in 5G/B5G networks. *IEEE Trans. Netw. Sci. Eng.* **9**(6), 4101–4114 (2022).
6. Sun, G., Wang, Y., Yu, H. & Guizani, M. Proportional fairness-aware task scheduling in space-air-ground integrated networks. *IEEE Trans. Serv. Comput.* (2024).
7. Li, Z., Hu, J., Leng, B., Xiong, L. & Fu, Z. An integrated of decision making and motion planning framework for enhanced oscillation-free capability. *IEEE Trans. Intell. Transp. Syst.* (2023).
8. Xu, X. et al. Three-dimensional reconstruction and geometric morphology analysis of lunar small craters within the patrol range of the Yutu-2 rover. *Remote Sens.* **15**(17), 4251 (2023).
9. Wang, K. et al. The fundamental property of human leg during walking: linearity and nonlinearity. *IEEE Trans. Neural Syst. Rehabil. Eng.* **31**, 4871–4881 (2023).
10. Wu, J., Ji, Y., Sun, X., Fu, W. & Zhao, S. Anonymous flocking with obstacle avoidance via the position of obstacle boundary point. *IEEE Internet Things J.* (2024).
11. Lu, S. & Xiao, X. Neuromorphic computing for smart agriculture. *Agriculture* **14**(11), 1977 (2024).
12. Li, M. et al. CNN-MLP-based configurable robotic arm for smart agriculture. *Agriculture* **14**(9), 1624 (2024).
13. Zhou, Z. et al. Vehicle lateral dynamics-inspired hybrid model using neural network for parameter identification and error characterization. *IEEE Trans. Vehic. Technol.* (2024).
14. Chen, J., Yu, C., Wang, Y., Zhou, Z. & Liu, Z. Hybrid modeling for vehicle lateral dynamics via AGRU with a dual-attention mechanism under limited data. *Control Eng. Pract.* **151**, 106015 (2024).
15. Meng, C., Zhang, T., Zhao, D. & Lam, T. L. Fast and comfortable robot-to-human handover for mobile cooperation robot system. *Cyborg Bion. Syst.* **5**, 0120 (2024).
16. Li, M. et al. CVT-based asynchronous BCI for brain-controlled robot navigation. *Cyborg Bion. Syst.* **4**, 0024 (2023).
17. Zhou, S., He, Z., Chen, X. & Chang, W. An anomaly detection method for uav based on wavelet decomposition and stacked denoising autoencoder. *Aerospace* **11**(5), 393 (2024).
18. Chen, J., Wang, J., Wang, J. & Bai, L. Joint fairness and efficiency optimization for CSMA/CA-based multi-user MIMO UAV ad hoc networks. *IEEE J. Sel. Top. Signal. Process.* (2024).
19. Wang, J. et al. Age of information based URLLC Transmission for UAVs on Pylon turn. *IEEE Trans. Vehic. Technol.* (2024).
20. Gao, N. et al. Energy model for UAV communications: experimental validation and model generalization. *China Commun.* **18**(7), 253–264 (2021).
21. Yin, Y., Wang, Z., Zheng, L., Su, Q. & Guo, Y. Autonomous UAV navigation with adaptive control based on deep reinforcement learning. *Electronics* **13**(13), 2432 (2024).
22. Zhang, X., Liu, Y., Chen, X., Li, Z. & Su, C. Y. Adaptive pseudoinverse control for constrained hysteretic nonlinear systems and its application on dielectric elastomer actuator. *IEEE/ASME Trans. Mechatron.* **28**(4), 2142–2154 (2023).
23. Ji, L. et al. Data-based optimal consensus control for multiagent systems with time delays: using prioritized experience replay. *IEEE Trans. Syst. Man Cybern. Syst.* (2024).
24. Liang, J., Yang, K., Tan, C., Wang, J. & Yin, G. Enhancing high-speed cruising performance of autonomous vehicles through integrated deep reinforcement learning framework. *arXiv Preprint*. arXiv:2404.14713 (2024).
25. Platt, J. & Ricks, K. Comparative analysis of ros-unity3d and ros-gazebo for mobile ground robot simulation. *J. Intell. Robot. Syst.* **106**(4), 80 (2022).
26. D'Angelo, S., Pagano, F., Longobardi, F., Ruggiero, F. & Lippiello, V. Efficient development of model-based controllers in PX4 firmware: a template-based customization approach. In *2024 International Conference on Unmanned Aircraft Systems (ICUAS)* 1155–1162 (IEEE, 2024).
27. Fei, J., Chen, Y., Liu, L. & Fang, Y. Fuzzy multiple hidden layer recurrent neural control of nonlinear system using terminal sliding-mode controller. *IEEE Trans. Cybern.* **52**(9), 9519–9534 (2021).
28. Sankey, D. W. et al. Absence of selfish herd dynamics in bird flocks under threat. *Curr. Biol.* **31**(14), 3192–3198 (2021).
29. Bahaidarah, M., Rekabi-Bana, F., Marjanovic, O. & Arvin, F. Swarm flocking using optimisation for a self-organised collective motion. *Swarm Evol. Comput.* **86**, 101491 (2024).
30. Li, L., Liu, X. & Huang, W. Event-based bipartite multi-agent consensus with partial information transmission and communication delays under antagonistic interactions. *Sci. China Inf. Sci.* **63**, 1–13 (2020).

Author contributions

Ruipeng Tang: Conceptualization, Methodology, Software, Validation, Formal Analysis, Investigation, Data Curation, Writing—Original Draft, Writing—Review and Editing, Visualization. Jianrui Tang: Conceptualization, Methodology, Validation, Formal Analysis, Writing—Review and Editing. Visualization. Mohamad Sofian Abu Talip: Investigation, Writing—Review and Editing, Supervision. Narendra Kumar Aridas: Resources, Project Administration, Methodology. Xifeng Xu: Data Curation, Software, Note: All the above authors agree to be responsible for the content and conclusions of the article.

Declarations

Competing interests

The authors declare no competing interests.

Additional information

Correspondence and requests for materials should be addressed to R.T.