

Special Topic: Embodied Intelligence

# Formation control and path planning of multi-robot systems via large language models

Dong XUE<sup>1\*</sup>, Xuanjie ZHOU<sup>1</sup>, Ming WANG<sup>1</sup> & Fangzhou LIU<sup>2\*</sup><sup>1</sup>*Key Laboratory of Smart Manufacturing in Energy Chemical Process, Ministry of Education, East China University of Science and Technology, Shanghai 200237, China*<sup>2</sup>*Research Institute of Intelligent Control and Systems, School of Astronautics, Harbin Institute of Technology, Harbin 150001, China*

Received 1 August 2024/Revised 26 November 2024/Accepted 8 January 2025/Published online 24 April 2025

**Abstract** Existing path planning and coordination control methods for multi-robot systems (MRS) typically rely on predefined rules and rudimentary algorithms. However, these methods often struggle to adapt flexibly to complex environments and to adjust motion targets appropriately. To address this challenge, this study presents a large language model (LLM)-assisted framework. By integrating textual descriptions of complex motion constraints, robot information, and local environmental data as inputs, LLMs generate motion objectives and translate them into executable control commands for the robots, thereby achieving coordinated control and path planning. This framework facilitates the generation, maintenance, and reshaping of formations in MRSs during path planning, applicable to both obstacle-free and obstacle-avoidance environments. Simulation results demonstrate that LLM-based control strategies enhance the autonomy, adaptability, flexibility, and robustness of MRS by processing complex information, making intelligent decisions, adapting to environmental changes, and handling disturbances and uncertainties.

**Keywords** multi-robot systems, formation control, path planning, obstacle avoidance, large language models

**Citation** Xue D, Zhou X J, Wang M, et al. Formation control and path planning of multi-robot systems via large language models. *Sci China Inf Sci*, 2025, 68(5): 150205, <https://doi.org/10.1007/s11432-024-4290-4>

## 1 Introduction

Over the past few decades, research in multi-robot systems (MRSs) has made significant strides. Compared to single-robot systems, MRSs offer higher robustness, flexibility, and redundancy, making them widely applicable in various fields such as agriculture [1], search and rescue [2], and exploration and patrol [3]. These notable advantages have driven researchers to develop more efficient and reliable control architectures to execute complex tasks [4].

Formation control and path planning are crucial fields in the coordination and cooperation research of MRSs. Common formation control strategies include the virtual structure method [5], leader-follower strategy [6], and behavior-based methods [7]. While these methods perform well in static environments, they still face challenges in dynamic and unknown environments. Decentralized control methods utilize local decision-making based on sensor information from each robot [8], improving flexibility but often lacking in control accuracy and robustness. Centralized control methods, where a leader coordinates the movements of other robots [9], offer higher control precision but place a heavy burden on the leader.

In the field of path planning, existing methods such as deep Q-learning network (DQN) algorithms [10], improved particle swarm optimization (IPSO) [11], and adaptive particle swarm optimization [12] have made progress in complex environments. However, these methods still have limitations when dealing with dynamic environments. In obstacle-rich environments, although traditional obstacle avoidance techniques [13–15] are effective, they still lack flexibility and robustness. Therefore, existing methods continue to face challenges in navigating through obstacle-laden environments and handling special fault disturbances.

With the advancement of large language models (LLMs), LLMs can integrate local environmental data to generate global decision-making plans, thereby improving task efficiency and reliability in MRSs,

\* Corresponding author (email: [dong.xue@ecust.edu.cn](mailto:dong.xue@ecust.edu.cn), [fangzhou.liu@hit.edu.cn](mailto:fangzhou.liu@hit.edu.cn))

while also supporting natural language interactions. LLM-based methods have introduced new solutions for MRS control. For instance, some studies have used LLMs to analyze shared information for robot convergence [16], while others have demonstrated the high success rate and flexibility of pre-trained LLMs in multi-robot collaboration [17]. Additionally, there have been comparisons between centralized and decentralized LLM planning frameworks [18], and investigations into the limitations of LLMs in multi-robot path planning [19]. Furthermore, recent advancements highlighted the application of LLMs in various aspects of robotics, such as grounding language in robotic affordances [20] and developing dynamic strategies for tasks like cloth flattening [21]. Another study emphasizes enhancing robot manipulation through human-robot collaboration [22]. However, the application and development of LLMs in system decision control are still immature, with issues such as resource and time consumption due to repetitive LLM decisions and a lack of concrete studies and solutions for communication delays. Moreover, there is a lack of research on formation control and dynamic path planning in MRSs using LLMs.

In this article, we aim to study LLM-based decision-making for leader-follower control strategies in MRSs within unknown environments, focusing on their application in formation control, obstacle avoidance, and dynamic path planning. During the experiments, various types of information and control constraints are provided to LLMs, enabling them to make appropriate decisions at different stages to achieve dynamic path-planning tasks. Throughout the process, the robots exchange information and execute instructions via an interactive network. The leader analyzes and makes decisions, while the followers execute the instructions, thereby achieving formation adjustment and movement. The main contributions of this study can be concluded as follows:

- Introducing LLM-based decision-making methods into MRSs, replacing mathematical model constraints with text descriptions to achieve efficient formation control and path planning;
- Designing a nested loop motion control framework based on LLM decision-making methods to reduce decision frequency and handle obstacles and system faults;
- Proposing a method of addressing LLM-predicted communication delays by delaying the update time of the leader to ensure consistency and resolve asynchronous control issues.

The rest of this article is organized as follows. Section 2 covers the unmanned aerial vehicle (UAV) dynamics model, the intra-system interaction network, and the LLM basic foundation. Section 3 discusses the problems involved in actual robot path-planning tasks. Section 4 outlines the processes and solutions of the proposed method to the problems mentioned in the previous section. Sections 5 and 6 provide simulation results and conclusion, respectively.

## 2 Preliminaries

This section will introduce the basic theories related to the research technologies. It begins with an overview of graph theory and interaction structures, followed by the dynamics model of quadrotor UAV, and concludes with the fundamentals of LLMs.

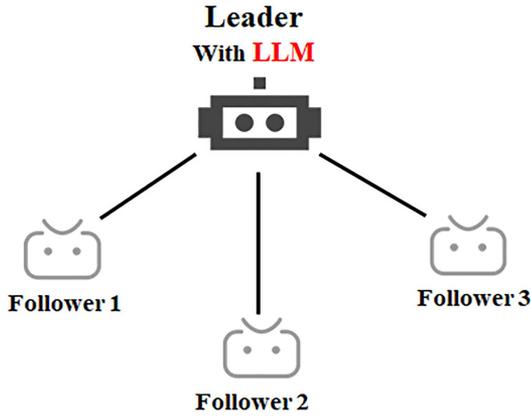
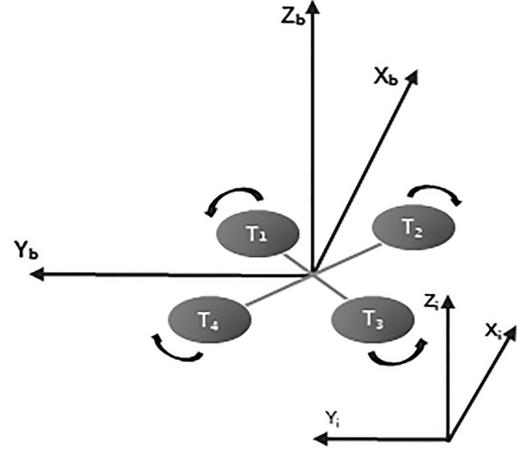
### 2.1 Graph theory

Graph theory is an effective tool for analyzing group relationships and modeling the interaction topology of MRSs [23]. An undirected graph  $G = (\mathcal{V}, \mathcal{E})$  consists of a set of nodes  $\mathcal{V}$  and a set of edges  $\mathcal{E} \subseteq \{i, j : i, j \in \mathcal{V}\}$ . For a node  $i \in \mathcal{V}$ , the set of neighboring nodes is defined as  $\mathcal{N}_i := \{j \in \mathcal{V} : i, j \in \mathcal{E}\}$ . In this article, we only consider simple undirected graphs without self-loops, with each robot defined as a node.

In the interaction network architecture of the MRS under study, we employ an undirected and partially connected network topology for formation control (as depicted in Figure 1). This architecture utilizes a leader-follower strategy, where one robot is designated as the leader and is integrated with LLMs. The leader aggregates local information from the followers along with perceptual data to construct a comprehensive internal representation of the system. Leveraging LLMs, the leader performs analysis and decision-making, subsequently communicating the derived instructions to the followers. The followers then execute the actions based on the instructions received from the leader.

### 2.2 Quadrotor UAV model

With the widespread deployment of UAVs in various real-world applications, the demand for their control performance is increasing [24]. This study focuses on small quadrotor UAVs, treating them as the primary


**Figure 1** (Color online) Interaction network structure.

**Figure 2** “X” quadrotor UAV structure.

subjects of a multi-agent system for in-depth exploration. In this subsection, the paper will provide a detailed explanation of the body structure and dynamic model of the UAV involved in the experiment.

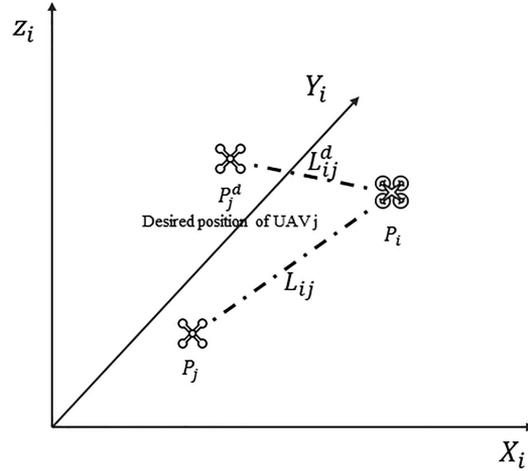
First, a common “X” configuration is chosen as the frame structure for the small quadrotor UAV in this study (Figure 2). The main structure consists of two pairs of intersecting straight arms of equal length, with the control center and sensing equipment typically mounted at the center point. Four propellers are fixed on the fuselage in four directions, with two pairs of propellers rotating in opposite directions to counteract torque generated by rotation. By controlling these four propellers to generate different lift forces, the UAV achieves six motion modes: pitch, roll, yaw, forward/backward, left/right, and vertical movements, collectively referred to as translational modes.

The study employs mathematical modeling to describe the dynamics of a small quadrotor UAV using a physics-based approach. The body coordinate system  $B = x_b, y_b, z_b$  and the earth coordinate system  $\mathcal{I} = x_i, y_i, z_i$  are specified based on the right-hand rule.

With the above considerations, the position-attitude dynamics model of our UAV in this study is

$$\begin{cases} \begin{bmatrix} \dot{v}_x \\ \dot{v}_y \\ \dot{v}_z \end{bmatrix} = \frac{f}{m} \begin{bmatrix} \cos \psi \sin \theta \cos \varphi + \sin \psi \sin \varphi \\ \sin \psi \sin \theta \cos \varphi - \cos \psi \sin \varphi \\ \cos \varphi \cos \theta \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ g \end{bmatrix}, \\ \begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} = J^{-1} \left\{ \begin{bmatrix} \tau_x \\ \tau_y \\ \tau_z \end{bmatrix} + \left( \begin{bmatrix} J_{xx} & 0 & 0 \\ 0 & J_{yy} & 0 \\ 0 & 0 & J_{zz} \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix} \right) \times \begin{bmatrix} p \\ q \\ r \end{bmatrix} + \begin{bmatrix} -J_0 q \Omega \\ J_0 p \Omega \\ 0 \end{bmatrix} \right\}, \end{cases} \quad (1)$$

where the yaw angle  $\psi$  is around the  $z_i$  axis, the pitch angle  $\theta$  is around the  $y_b$  axis, and the roll angle  $\varphi$  is around the  $x_b$  axis of the quadrotor UAV. The control input  $f$  represents the resultant force generated by the four motors, while the angular velocity of the UAV in its body coordinate system  $S_d$  is denoted by  $\omega^b = [p \ q \ r]^T$ . The inertia tensor matrix  $J = \text{diag}(J_{xx}, J_{yy}, J_{zz})$  encapsulates the principal moments of inertia  $J_{xx}$ ,  $J_{yy}$ , and  $J_{zz}$ , corresponding to the moments of inertia about the  $x$ -,  $y$ -, and  $z$ -axes, respectively. Moreover, the moment of inertia of the motor combined with the propeller along the motor shaft is denoted by  $J_0$ . The motor speed  $\omega$  indicates the rotational velocity of each motor, and the parameter  $\Omega$  is defined as  $\Omega = \omega_1 - \omega_2 + \omega_3 - \omega_4$ . Finally, the torque generated by the lift force on the UAV is expressed as  $\tau^b = [\tau_x^b \ \tau_y^b \ \tau_z^b]^T$ .



**Figure 3** Leader-follower formation control constraints.

The power distribution matrix  $U$  is defined as

$$U = \begin{bmatrix} f \\ \tau_x \\ \tau_y \\ \tau_z \end{bmatrix} = \begin{bmatrix} C_T & C_T & C_T & C_T \\ \frac{C_T d}{\sqrt{2}} & -\frac{C_T d}{\sqrt{2}} & -\frac{C_T d}{\sqrt{2}} & \frac{C_T d}{\sqrt{2}} \\ -\frac{C_T d}{\sqrt{2}} & -\frac{C_T d}{\sqrt{2}} & \frac{C_T d}{\sqrt{2}} & \frac{C_T d}{\sqrt{2}} \\ -C_M & C_M & -C_M & C_M \end{bmatrix} \begin{bmatrix} \omega_1^2 \\ \omega_2^2 \\ \omega_3^2 \\ \omega_4^2 \end{bmatrix}, \quad (2)$$

where  $C_T$  represents the thrust coefficient, which defines the relationship between the propeller speed and the generated lift force.  $C_M$  is the moment coefficient, which indicates the relationship between the propeller speed and the torque produced about the motor axis. The parameter  $d$  denotes the length of the UAV arm, which is the distance from the center of the UAV to each motor.

In the context of a homogeneous MRS, each quadrotor  $i \in \{1, 2, \dots, n\} \subset \mathbb{N}$  is modeled as a rigid sphere. The nonlinear discrete-time state equation of each UAV is

$$x_i^{k+1} = F_i(x_i^k, u_i^k), \quad (3)$$

where  $x_i^k = [p_i^k, v_i^k, \phi_i^k, \theta_i^k, \psi_i^k]^T \in \mathcal{X}_i \subset \mathbb{R}^{n_x}$  represents the state of the UAV at time  $k$ , with  $p_i^k$  denoting the position,  $v_i^k$  representing the three-dimensional velocity, and  $\phi_i^k$ ,  $\theta_i^k$ , and  $\psi_i^k$  referring to the roll, pitch, and yaw angles, respectively. The control input  $u_i^k = [f_i^k, \tau_{i,x}^k, \tau_{i,y}^k, \tau_{i,z}^k]^T$  represents the power distribution values from (2) at time  $k$ , and serves as the input to the model.

### 3 Problem description

In this section, we first elaborate on the mathematical models of MRS and the necessary introduction of control technologies. The underlying challenges arising from formation control and path planning problems in MRS are then explored.

#### 3.1 Formation control and obstacle avoidance

When the target point and the initial state of the UAVs are known, control algorithms must be pre-designed to achieve formation control and path planning in the local environment. Crucial issues include formation stability, path safety, and control robustness. For instance, constraints are often incorporated into the pre-designed algorithms to stabilize the UAVs, maintain the stability of the formation structure, and prevent internal collisions.

As illustrated in Figure 3, the leader-follower scheme is a widely applied constraint and control strategy [25]. Here,  $L_{ij}$  represents the actual distance between UAV  $i$  and UAV  $j$ , while  $L_{ij}^d$  represents the

desired distance needed to maintain formation stability. The control task consists of determining the linear and angular velocities between each pair of UAVs to eliminate the formation error within the whole team, which leads to

$$\lim_{t \rightarrow \infty} (L_{ij} - L_{ij}^d) = 0, \quad \forall i, j \in \{1, 2, \dots, n\}. \quad (4)$$

After completing the formation, when planning the path between the current formation and the target point, obstacles between them must be considered. Common obstacle avoidance constraints include collision avoidance between UAVs and between UAVs and obstacles. When the avoidance strategy involves adjusting the formation structure, collision avoidance inside the UAV formation must also be considered and thus needs to be satisfied

$$\begin{cases} \|p_i^k - p_j^k\| > r_i + r_j, & \forall j \in \{1, 2, \dots, n\}, j \neq i, \\ \|p_i^k - p_o^k\| > r_i + d, & \forall j \in \{1, 2, \dots, n\}, j \neq i, \end{cases} \quad (5)$$

where  $r$  is the maximum radius of the UAV body, and  $d$  is the maximum radius of obstacles.

Implementing flight control under these complex constraints is a challenging task. This study aims to introduce LLM decision-making, using textual constraints instead of mathematical ones to reduce the difficulty of solving formation and obstacle avoidance problems in path planning.

### 3.2 Asynchronous control under communication delays

In real-world scenarios, there are often communication delays in the exchange of information between UAVs. This means there is a lack of synchronization between the information and movement updates of the leader and followers. The leader must wait for a period to receive delayed information from the followers before making decisions. Similarly, the leader updates its target and movements first, while followers start executing new decision actions after a delay upon receiving the new signals from the leader. These communication delays affect the responsiveness of UAVs, resulting in asynchronous control in this team (Figure 4). To investigate the impact of asynchronous control caused by communication delays on the proposed method, we model these delays as Poisson random variables and incorporate communication delays into the interaction network [26], which leads to

$$\delta_{ij}(t) = C_{\text{delay}} \cdot \text{Pois}(\lambda_{ij}(t)), \quad \lambda_{ij}(t) = \sum_{j \in \mathcal{N}} \|x_i^k - x_j^k\|, \quad (6)$$

where  $\delta_{ij}(t)$  represents the communication delay between UAVs, modeled as a scaled Poisson random variable. The constant  $C_{\text{delay}}$  scales the magnitude of the delay, with larger values indicating longer delays. The parameter  $\lambda_{ij}(t)$  is the rate of the Poisson distribution and is calculated as the sum of Euclidean distances between UAV  $i$  (with its current position denoted by  $x_i^k$ ) and all interacting UAVs  $j$  in the set  $\mathcal{N}$ . For the leader UAV,  $\mathcal{N}$  includes all follower UAVs, reflecting the communication from the leader to its followers. Conversely, for a follower UAV,  $\mathcal{N}$  consists solely of the leader, as followers only receive information from the leader. To simulate the stochastic nature of real-world communication delays, we sample from the Poisson distribution defined by  $\lambda_{ij}(t)$  to estimate the actual delay. This approach ensures that each calculated delay captures inherent randomness and variability.

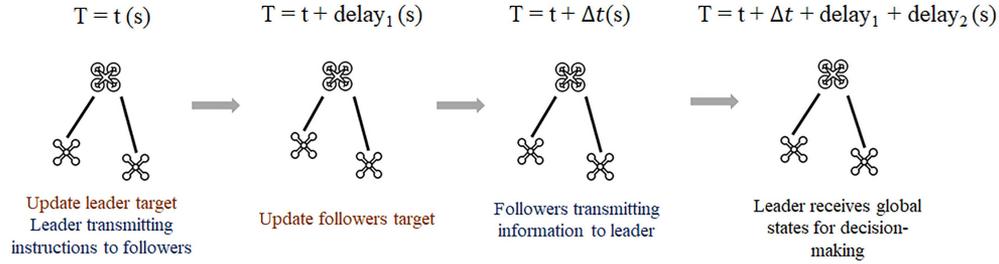
The subsequent responsibility of LLMs is to select appropriate control strategies and generate corresponding decisions to mitigate the uncertainty introduced by communication delays and enhance the stability of the formation.

### 3.3 Unexpected faults

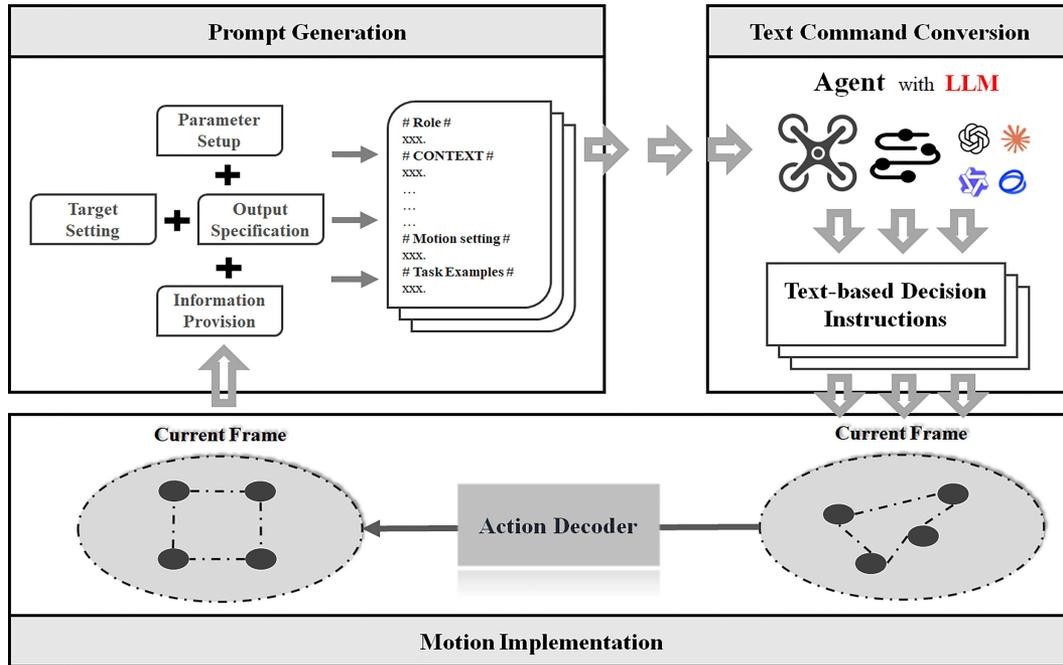
When UAVs plan a path in an environment, they may encounter various degrees of disturbances and failures. These can be categorized into two main scenarios.

- Mild disturbances. UAVs may experience slight external disturbances, such as changes in wind or small-scale airflow fluctuations, causing deviations from their planned paths. If not managed, these disturbances can affect the stability of the entire formation.
- Severe failures. In more critical cases, a UAV may lose control or communication due to significant external disturbances or mechanical failures, making it unable to continue its role in the formation. The remaining UAVs must quickly respond and reorganize to maintain coordinated movement.

This evaluation will also verify the robustness and intelligence of the method under unforeseen circumstances.



**Figure 4** (Color online) Asynchronous control under signal delay. The term  $\text{delay}_1$  refers to the time required for the leader to transmit a signal to the followers, while  $\text{delay}_2$  represents the time required for the followers to send a signal back to the leader. At a global time instant  $t$ , after the followers have moved for a specified duration  $\Delta t$ , the current state information is transmitted to the leader. It is crucial to note that this  $\Delta t$  represents the time interval for updating the state information, rather than a communication delay. The leader will then utilize the updated information from the followers, which may be further affected by  $\text{delay}_1$  and  $\text{delay}_2$ .



**Figure 5** (Color online) LLM-based control framework.

## 4 Methodology and analysis

This section mainly introduces the specific processes of the LLM-based control method and the measures to address the related issues presented in Section 3.

### 4.1 Overview framework

Figure 5 illustrates the overall framework of the proposed method, which consists of three main modules. (1) Prompt generation. This module generates textual prompts tailored to specific motion tasks by integrating various pieces of information and constraints. (2) Text command conversion. This process converts the new target decisions output by LLMs into control commands for the dynamics model of the robot. (3) Motion implementation. Robots execute flight movements and local information detection based on the new targets. The following sections will provide a detailed explanation of each component.

### 4.2 Prompt generation

To generate effective prompts, all relevant information and constraints must be organized into a unified textual description. This description primarily includes four parts: (1) the physical and motion parameters of the robots; (2) the motion goals and strategies for the next phase; (3) the local environment and

information about the robots team; (4) the specified format requirements for the model output. This prompt will serve as input for LLMs, enabling them to incorporate their reasoning to generate the motion goals of the next phase, thus making decisions for different movement phases.

To address the issues presented in Section 3, it is crucial to fully leverage and validate the decision-making capabilities of LLMs. LLMs are responsible for various aspects of motion decision-making, including formation control, obstacle avoidance, and fault management. A single prompt cannot entirely satisfy the diverse requirements of formation maintenance, obstacle avoidance, and fault handling. Hence, prompt engineering must be further optimized. By incorporating motion phase detection, event-driven adaptive prompt selection can be achieved, enabling the LLM to perform dynamic path planning under different conditions. Specifically, the system evaluates the overall state in real time based on the actual events occurring during the motion of robots and selects the most appropriate prompt for the LLM according to the analysis of the movement phase of the team. For instance, the requirements for maintaining formation stability differ from those for obstacle avoidance, and obstacle avoidance strategies may vary depending on environmental conditions. These variations necessitate carefully designed prompts to guide LLMs in generating high-quality motion strategies. Detailed prompts can be found in Appendix A, and related algorithms are provided in Algorithms B.1 of Appendix B.

### 4.3 Text command conversion

The decisions output by LLMs integrated into the leader are textual data that cannot be directly used. Therefore, they need to be converted into inputs for the dynamic model. Here, we provide a method to make textual decisions feasible.

In the scheme, we first use the prompt mentioned in the previous section to select the form of the output content. Depending on the different movement strategies specified by the user, there are two categories. The first approach involves having LLMs assign a predicted coordinate to each robot individually and output the values for all robots simultaneously. The second approach involves first generating a predicted coordinate for the next formation center and then generating predicted coordinates for the individual robots based on the relative structure of the formation. These two methods can be applied to both formation correction and maintenance. Next, the textual information of the formation nodes at the next moment is transmitted to each robot. The robots can use motion encoders to convert the coordinate information into inputs for the dynamic model, thus achieving motion control.

As known from Subsection 2.2, the control inputs for each UAV include thrust  $f$  and torques in three directions  $\tau_x, \tau_y, \tau_z$ . Combined with (2), we use a cascade proportional-integral-derivative (PID) control. The difference between the target coordinates at the next moment provided by the textual decision instructions and the current coordinates serves as the PID input, generating the dynamic model inputs for the discrete-time model. The specific PID implementation process is as follows.

The cascade PID control of the UAV is divided into an inner loop and an outer loop. The outer loop PID generates the reference acceleration command for the UAV, while the inner loop PID provides the combined external torque command. The output of the outer loop position control PID is defined as

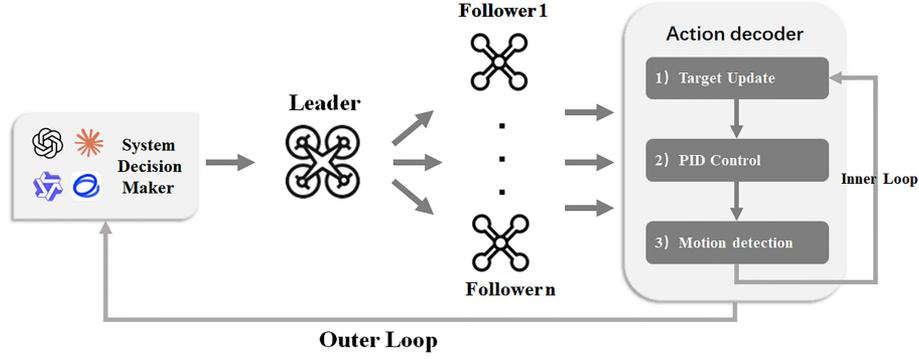
$$\begin{cases} u_x = \text{PID}_x.\text{out}(x, x_d, \dot{x}, \dot{x}_d), \\ u_y = \text{PID}_y.\text{out}(y, y_d, \dot{y}, \dot{y}_d), \\ u_z = \text{PID}_z.\text{out}(z, z_d, \dot{z}, \dot{z}_d). \end{cases} \quad (7)$$

According to (1), one can obtain

$$\begin{bmatrix} u_x \\ u_y \\ u_z \end{bmatrix} = \frac{f}{m} \begin{bmatrix} \cos \psi \sin \theta \cos \phi + \sin \psi \sin \phi \\ \sin \psi \sin \theta \cos \phi - \cos \psi \sin \phi \\ \cos \phi \cos \theta - mg \end{bmatrix}, \quad (8)$$

where  $f = m\sqrt{u_x^2 + u_y^2 + (u_z + g)^2}$ .

Simultaneously, using the reference yaw angle (set manually), the reference roll angle is calculated, and



**Figure 6** (Color online) Motion implementation flow.

the reference pitch angle is determined as

$$\begin{cases} \phi_d = \arcsin \left[ (u_x \sin \psi_d - u_y \cos \psi_d) \frac{m}{f} \right], \\ \theta_d = \arcsin \left[ \frac{u_x m - U_1 \sin \psi_d \sin \phi_d}{f \cos \psi_d \cos \phi_d} \right]. \end{cases} \quad (9)$$

Next, the attitude control operates independently of position, where the PID control output directly corresponds to the combined external torque of the UAV in three directions

$$\begin{cases} \tau_x = \text{PID}_{\text{roll}}.out(\phi, \phi_d, \dot{\phi}, \dot{\phi}_d), \\ \tau_y = \text{PID}_{\text{pitch}}.out(\theta, \theta_d, \dot{\theta}, \dot{\theta}_d), \\ \tau_z = \text{PID}_{\text{yaw}}.out(\psi, \psi_d, \dot{\psi}, \dot{\psi}_d). \end{cases} \quad (10)$$

Combining (8) and (10), the control input  $u_i^k = [f, \tau_x, \tau_y, \tau_z]$  for the dynamics equation is obtained.

#### 4.4 Motion implementation

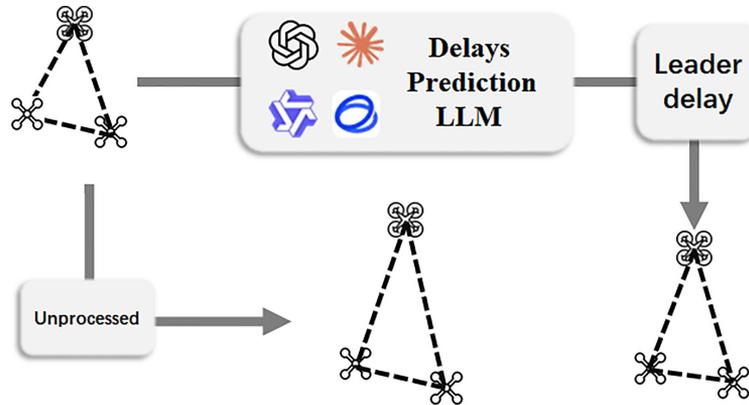
After LLMs generate the next phase targets for the robots, each robot will update its respective target and adjust its movement accordingly. This movement will utilize discrete-time control at a specified frequency to ensure motion toward the target points. It is important to note that each execution and computation by LLMs requires a certain amount of resources and time. To ensure that the entire motion process is completed within a reasonable control range, this study proposes a nested loop control process to reduce resource consumption associated with LLMs, as illustrated in Figure 6. Related algorithms can be found in Algorithms B.2 and B.3 of Appendix B.

In the outer loop, LLMs act as the decision-maker, responsible for planning the formation structure and movement targets for the current robots. In each cycle, the decision-maker generates decision information and transmits it through the interaction network between robots, guiding the movement of the entire system for that cycle. When the movement target is achieved or an unexpected situation arises, the leader collects signals from the followers and feeds them back to the decision-maker, thus transitioning to the next cycle.

In the inner loop, each robot moves towards its designated target according to personalized instructions and a specified control frequency. Due to the time cost associated with using LLMs, multiple inner loops can be executed within a single outer loop. The number of inner loop executions depends on the distance between the current position and the decision target. This approach effectively enhances the movement efficiency of robots.

Additionally, a motion detection phase is introduced at the intersection of the inner and outer loops. This phase performs detection after each discrete-time step of motion control to assess the completion of movement, allowing for an early transition to the next cycle when targets are met. It also handles local environment perception, stopping the loop upon detecting obstacles to facilitate early decision-making for obstacle prediction and avoidance, ensuring efficient and safe movement.

When dealing with specific obstacle avoidance and fault issues, the feedback from the motion detection phase will participate in prompt selection and generation. This will enable the dynamic path planning task of the entire system to be completed in a feedback-based manner using LLM decision-making.



**Figure 7** (Color online) Delay prediction control.

#### 4.5 Delays predict

In multi-robot control, communication delays often impact the stability of the formation. As discussed in Subsection 3.3, asynchronous target updates can lead to inconsistencies in robot velocities, thereby affecting the stability of the formation. Additionally, the inaccuracies introduced by delayed updates pose a risk of collisions between the formation and obstacles.

LLMs can act as intelligent cores, understanding the process and mechanism to predict communication delays between the leader and followers. Leveraging this capability, LLMs can output new decisions to adjust the timing of leader target updates appropriately. Specifically, after LLMs make decisions and issue commands to the leader, the target of the leader should not be updated immediately. Instead, by understanding the delay logic through LLMs and using the predictive information, the delay between the leader and the followers can be estimated. The target of the leader update can then be controlled to occur after the predicted delay.

To implement the above strategy, a prediction module has been designed and introduced. This module uses the mathematical delay model described in (6) as the basis for prediction logic, providing an appropriate delay time for updating the target of the leader. After this delay, the leader can update its target, ensuring consistency and stability within the formation. As shown in Figure 7, this approach helps prevent the update of the target of the leader from disrupting the existing structure of the formation. It is important to note that when parameters in the equations change, the corresponding prompts must be adjusted to ensure that LLMs correctly understand the requirements for delay prediction and can provide accurate predictions.

Furthermore, compared to previous methods that relied on high-performance LLMs for decision-making based on complex information, delay prediction no longer depends on multiple information sources and complex constraints. Using a smaller model can enhance system performance to some extent. The hybrid use of models can balance performance and efficiency, leading to more efficient task completion.

However, for severe and unpredictable communication delays, such as situations where unexpected network issues cause the delay prediction model to fail, the system must be capable of adaptation and adjustment. Although the current model relies on predefined delay logic and parameters, in unpredictable scenarios, redundancy mechanisms or fallback strategies need to be implemented. For instance, the system can include real-time monitoring modules to detect anomalies of communication quality and delay, dynamically adjusting the operational logic of the model. If the leader does not receive new information after a period exceeding the combined predictable delay and communication interval, small-scale LLMs can be deployed on followers. These local LLMs can take over decision-making to temporarily update targets, ensuring the continuity of movement when new information is not received as expected. This resilient design ensures that even if the communication delay model fails, the system can maintain basic operational and control capabilities, supporting the successful execution of tasks.

**Table 1** Physical parameters.

Parameter type	Value
Quality: $m$	0.8
Acceleration of gravity: $g$	9.8
Moment of inertia: $J_{xx}, J_{yy}, J_{zz}$	4.212e-3, 4.212e-3, 8.255e-3
Arm length: $d$	0.12
Lift coefficient of propeller: $C_T$	2.168e-6
Torque coefficient of propeller: $C_M$	2.136e-8
Moment of inertia of the motor and propeller: $J_0$	1.01e-5
Max speed: $v$	4
Max force: $f$	10
Radius of obstacle: $r$	0.1-0.5
Range of perception: $p$	3
pid_x: $kp, ki, kd$	0.5, 0, 120
pid_y: $kp, ki, kd$	0.5, 0, 120
pid_z: $kp, ki, kd$	0.7, 0, 200
pid_phi: $kp, ki, kd$	0.5, 0, 20
pid_theta: $kp, ki, kd$	0.5, 0, 120
pid_psi: $kp, ki, kd$	0.1, 0, 10

## 5 Experiment

This section mainly introduces the experimental setup and simulates the movement of an MRS composed of UAVs in a realistic environment, as well as the simulation results under LLM-based decision-making.

### 5.1 Experimental setup

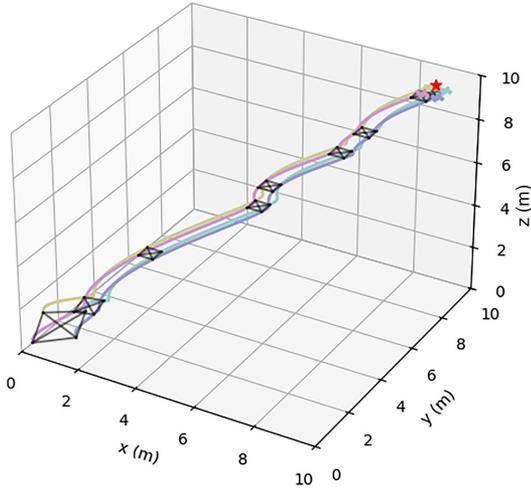
To validate the performance of the proposed LLM-assisted method in formation control and path planning, we have established an unmanned vehicle control experimental platform using Python 3.10 on an Ubuntu 20.04 system. This platform is designed to support research on cluster control and cooperative actions of UAVs.

In terms of the experimental environment configuration, Ubuntu 20.04 was chosen as the operating system, running on a machine equipped with 8 Intel(R) Xeon(R) Platinum 8369B processors and 32 GB of memory. Additionally, LLM API was introduced to facilitate the use of LLM as a primary tool in the experiments. Following the setup of the experimental environment, it is necessary to define the physical model parameters and control parameters for the entire experiment. These parameters will directly impact the motion behavior and control effectiveness of the UAVs. Table 1 lists the physical parameters used in this experiment.

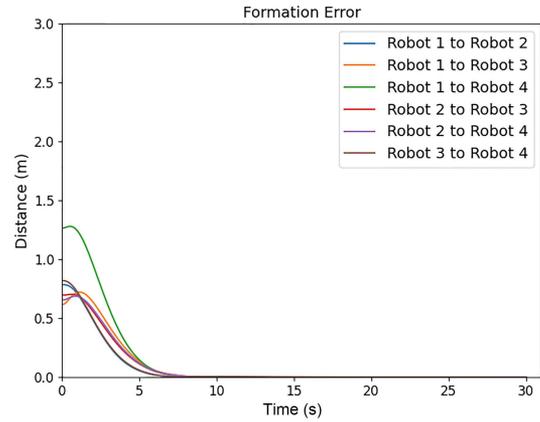
After completing the basic experimental setup, we first define the motion of UAVs. In a 3D space, a specific area far from the final formation target point is selected, within which four UAVs with initial velocities and angles set to zero are discretely distributed. In the initial stage, the UAVs receive the formation target and move to the model-generated formation node coordinates, completing the initial formation. Once the formation of the system stabilizes, each segment of the motion target is progressively planned, moving towards the final target point of the formation. When the UAVs are within a certain range of the target point, the dynamic path planning task is considered complete, and the UAVs stop moving. Throughout the motion phase, the UAVs' trajectories, attitude changes, and task execution are continuously recorded for analysis and evaluation.

### 5.2 Path planning experiment in UAVs formation

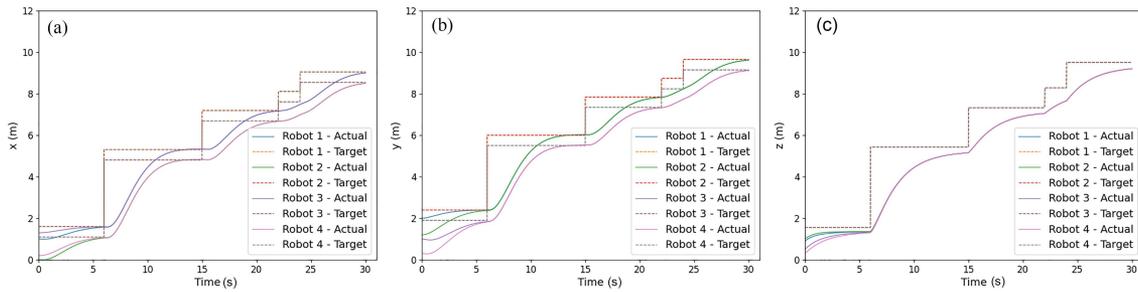
In traditional path planning problems for UAV formations, three main phases are typically involved: forming the preset target formation, maintaining the formation during movement, and dynamically planning the movement path. Traditional methods often rely on predefined rules and algorithms to achieve specific formation and movement strategies. While effective for UAVs designed for specific goals and scenarios, these methods are less adaptable in dynamic environments that require more diverse formation shapes and movement requirements. This also necessitates more complex pre-configuration and constraints.



**Figure 8** (Color online) Flight trajectory of the UAVs. The different colors represent the trajectories of different UAVs, and the red point represents the formation target.



**Figure 9** (Color online) Formation error. The deviation between the actual distance and the expected distance between UAVs.

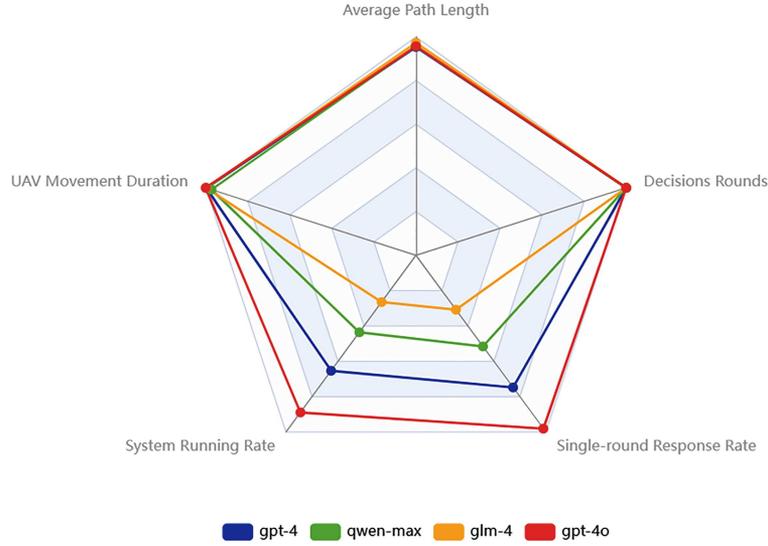


**Figure 10** (Color online) Position change of the UAVs in the  $x$  (a),  $y$  (b), and  $z$  (c) directions.

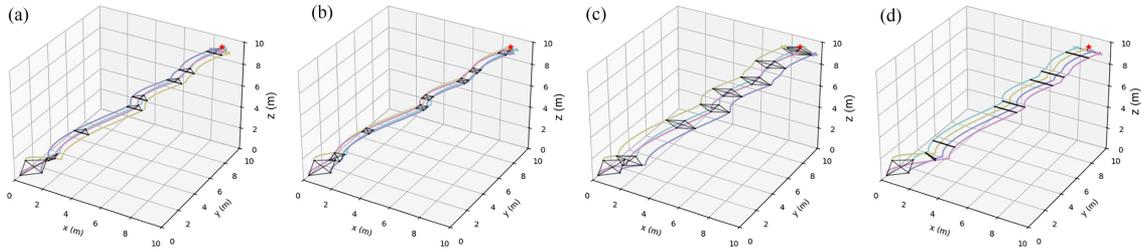
In contrast, the LLM-based formation method proposed in this study addresses these challenges by leveraging prompt engineering, as described in Subsection 4.2. By providing formation requirements and examples through prompts, LLMs can allocate initial formation nodes based on the current information of the UAVs. Specific node allocation strategies can also be defined via prompts, transforming complex mathematical constraints into simpler, direct textual expressions, thereby offering more flexibility in user-LLM interactions. However, it is important to note that due to the inherent uncertainty in LLM decision-making, identical prompts may yield different initial formation node coordinates or different update coordinates during dynamic path planning. This variability does not affect overall movement but introduces some numerical differences.

According to the entire motion trajectory of MRSs shown in Figure 8, all UAVs are assigned to their formation nodes and move toward these nodes for formation initialization at the beginning of the motion. Once the formation structure stabilizes, LLMs generate the next target points for each time step to ensure that the UAV swarm progresses towards the final formation target, thereby achieving dynamic path planning. The formation movement is completed when the UAVs reach the target point. The motion control strategy proposed in this study differs from traditional methods that generate paths and track trajectories. Instead, it relies more on dynamically updating target points and combining PID control. While PID control may incur some speed loss, it provides flexibility, allowing for more agile adjustments of targets and movement trajectories within a limited motion space. An example of the dialogue flow based on the LLM decisions can be found in Appendix C.

Moreover, Figure 9 demonstrates the stability of the formation within the system, with the UAVs maintaining formation accuracy after initialization and during subsequent movements. Meanwhile, Figure 10 presents the updates of target coordinates and position changes for each UAV in three directions during the movement of the system. It can be observed that the LLM-based decision-making method successfully meets the movement requirements and process, maintaining formation stability and reliably



**Figure 11** (Color online) Comparison of multiple models under five metrics. (1) Average path length: average path length of 4 UAVs. (2) Decisions rounds: number of LLM decision rounds required to complete formation and dynamic path planning. (3) Single-round response rate: the response rate of LLMs for a single round. (4) System running rate: workload of the entire system per unit time. (5) UAV movement duration: total duration of UAVs performing the mission.



**Figure 12** (Color online) Four UAVs formed a triangle (a), square (b), diamond (c), and line (d) formation structure.

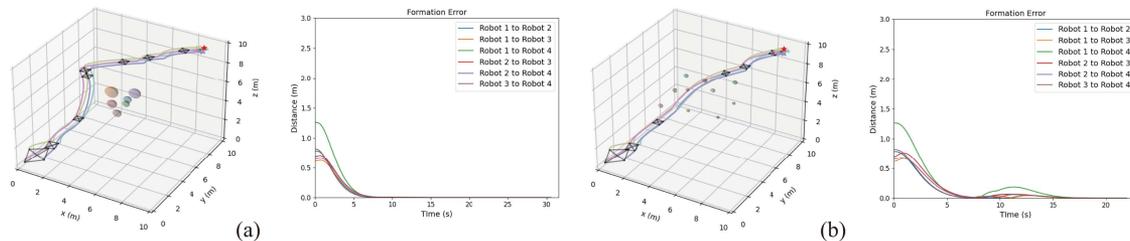
guiding the group to the target point.

In addition, we also considered the differences among various LLMs in solving specific problems. We selected several representative LLMs and compared their performance by calling the respective APIs. As shown in Figure 11, the performance metrics of LLMs under the existing prompt control strategy did not show significant differences that would affect specific performance. However, there were differences in their response speed and total runtime, with GPT-4o notably standing out for its shorter response time while still providing accurate results and achieving nearly identical movement decisions. Consequently, we will choose GPT-4o as the primary LLM API for future experiments in this study. Regarding formation control, it is worth noting that LLMs have a unique advantage in generating formations of different shapes. As depicted in Figure 12, various formation shapes and movement trajectories can be created by simply modifying the prompts without setting specific formation structures in the program. This flexibility allows the method to accommodate different formation requirements. As mentioned earlier, since LLM decisions are based on prompts, a lack of strong constraints in the prompts may lead to variability in the generated formation node coordinates across different runs. This uncertainty might result in slight positional differences, though not in shape, even for the same formation shape requirements.

### 5.3 Dynamic path planning and obstacle avoidance experiment

When dynamic path planning introduces obstacles under local perception, solving the avoidance problem involves more complex factors. The size and distribution of obstacles, as well as the chosen avoidance strategies, significantly impact the performance and behavior of the entire system. This subsection focuses on evaluating two obstacle avoidance strategies, exploring their effectiveness and safety in environments with different types of obstacle distributions.

Traditional obstacle avoidance methods based on local environment perception, although intricate and



**Figure 13** (Color online) Two types of obstacle avoidance motions in different environments. (a) Detour to avoid concentrated obstacles; (b) adjust formation to pass through scattered obstacles.

**Table 2** UAVs movement path length (m).

Experiment type	UAV1	UAV2	UAV3	UAV4
Experiment (a)	17.237	18.116	17.823	18.653
Experiment (b)	14.643	15.498	15.137	15.904

**Table 3** Minimum distance from the center of the UAV to the surface of the obstacle (m).

Experiment type	UAV1	UAV2	UAV3	UAV4
Experiment (a)	0.881	1.221	1.305	1.573
Experiment (b)	0.685	0.563	0.339	0.422

effective, often apply a single strategy across various scenarios. This study aims to investigate whether an LLM-based approach can more flexibly implement different avoidance strategies suited to varying obstacle environments. To address this, we designed targeted experiments to test LLM-based avoidance strategies, validating their ability to select appropriate methods in diverse situations, thereby achieving superior overall performance.

Figure 13 highlights the differences between the two avoidance methods when dealing with different types of obstacles. In terms of formation stability, the detour strategy (a) effectively maintains the stability of the entire formation and reduces internal collisions. In contrast, the formation adjustment strategy (b) allows the formation to alter its structure to reach the target point, resulting in slightly reduced stability. However, after avoiding obstacles, this strategy can reestablish the formation, demonstrating robustness despite a slight decrease in stability. Regarding path cost (Table 2), the formation adjustment strategy (b) is advantageous compared to the detour strategy (a) as it can navigate through the entire obstacle field without adding excessive distance. Overall, each avoidance method has its strengths and weaknesses. LLM-based intelligent systems can potentially select different strategies based on specific needs, enabling UAVs to adapt to more complex environments while minimizing overall costs.

It is also important to note that since UAVs perceive only the local environment, decision accuracy is improved by reassessing and selecting the next safe point for avoidance whenever an obstacle is detected. By combining this approach with PID control, UAVs can achieve flexibility in motion through adaptive decision-making. This method enhances the overall robustness of the system, making the avoidance process safer and more cost-effective. Finally, Table 3 presents the minimum distances between UAVs and obstacles in various experiments, demonstrating that all distances remain within a safe range.

To validate the high success rate of the proposed method, an additional experiment was conducted. This experiment involved repeating the three previous experiments with the same configuration 20 times, observing and recording three key metrics: collision avoidance rate, obstacle avoidance rate, and model response success rate. These metrics indicate whether a drone collides during formation flight, whether a drone collides with obstacles during obstacle avoidance, and whether the model successfully outputs decisions in each round to ensure stable control of the drone system. The specific experimental results are shown in Table 4. Due to the inherent randomness of large models, the results are not 100% successful. Additionally, the understanding and decision-making of the model may vary depending on the motion strategy. Optimizing the motion strategy can help improve the overall success rate of the system. The relatively high success rate indicates that the LLM-based approach can effectively serve as the “brain” of a multi-agent system.

The randomness and hallucination issues of the model are, to some extent, limited by the performance of the model and the quality of the prompt design. Given the current limitations in hardware and

**Table 4** Multi-round experiment success rate. Experiments (a), (b), and (c) represent the previous formation experiment, the detour-obstacle avoidance experiment, and the adjust-formation-obstacle avoidance experiment, respectively.

Experiment type	Collision avoidance rate (%)	Obstacle avoidance rate (%)	Response success rate (%)
Experiment (a)	100	–	100
Experiment (b)	100	100	100
Experiment (c)	100	80	95

resources, it is anticipated that by optimizing the prompt design and utilizing a more powerful model, the success rate can be further improved, making the overall motion control decisions more precise and efficient.

#### 5.4 Communication delay experiment

In this subsection, based on the negative impacts of delays described in Subsection 3.2, we introduce the communication delay model outlined in Subsection 3.2 into the experiments of Subsection 5.2. The aim is to design a corresponding experiment to verify and discuss the performance of the proposed method under communication delays and to evaluate the effectiveness of the delay prediction technique discussed in Subsection 4.5.

The results shown in Figure 14 reveal the motion trajectories and formation stability under varying degrees of delay prediction. In the first set of formation experiments, by comparing Figures 14(a)–(c), it is evident that without delay prediction, the leader integrated with LLMs updates its target and moves first after LLMs makes a decision. This action affects the followers, causing their movement updates to lag due to the delay, thereby disrupting the formation structure. However, by predicting delays, we can defer the target of the leader update, achieving temporal consistency in movement updates and maintaining formation stability. Under ideal conditions, the formation can remain stable while moving toward the target point.

Similarly, the benefits of delay prediction are apparent in the subsequent obstacle avoidance experiments. When an obstacle is detected, the system does not immediately update its state but waits to gather more information before making a decision. This approach aids in better route planning and reduces the risk of collisions caused by the leader advancing while neglecting the followers. As shown in Table 5, without delay prediction, communication delays, and untimely updates increase the risk of collisions. Conversely, delay prediction can mitigate these risks to some extent.

Overall, delay prediction significantly enhances the stability of formation movement and reduces the risk of collisions during obstacle avoidance. These experimental results highlight the potential application of LLM in managing asynchronous control and communication delays, significantly improving the stability and reliability of UAVs in complex environments. Furthermore, this suggests that if UAVs possess a certain level of proactive intelligence, they may not solely rely on predefined delay prediction methods for asynchronous control, thereby enhancing system stability and reducing computational burden.

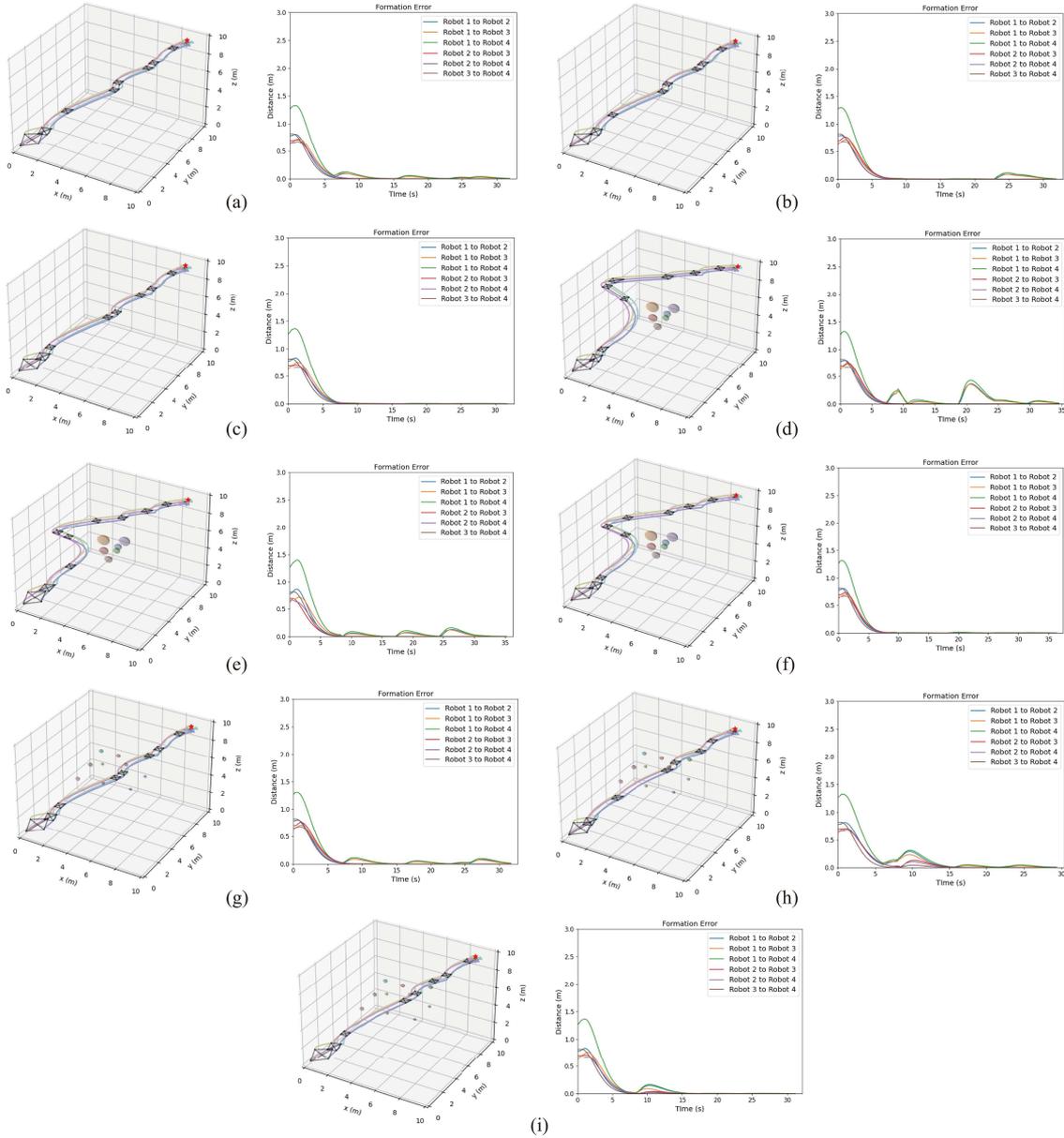
#### 5.5 Formation reorganization under special faults experiment

This subsection examines how to ensure the completion of formation and target point tracking tasks when UAVs experience collective propulsion failures or individual UAVs completely depart from the formation due to mechanical faults. Experimental results are shown in Figure 15.

In Figure 15(a), from seconds 8 to 10, mechanical failures caused by the overload of four propellers on the UAVs resulted in the disruption of the stable structure of the formation. Despite this, the system managed to reach the target for the current round and continuously updated the target in subsequent movements.

Figure 15(b) introduces a wider range of random input disturbances, simulating a scenario where, at second 9, a follower UAV experiences a fault and departs from the formation. The remaining UAVs, upon receiving current information from LLMs and making decisions, were instructed to reconfigure into a triangular formation and steadily approach the target point.

These results indicate that for LLM-based control systems, when specific UAV information is lost or a UAV significantly deviates from the formation, the leader can issue commands solely to the remaining followers. These commands guide the remaining UAVs to reconfigure the formation structure while maintaining stability and safety as they continue toward the target point, thereby completing the remaining tasks. This indicates that UAVs integrated with LLM possess a certain level of robustness,

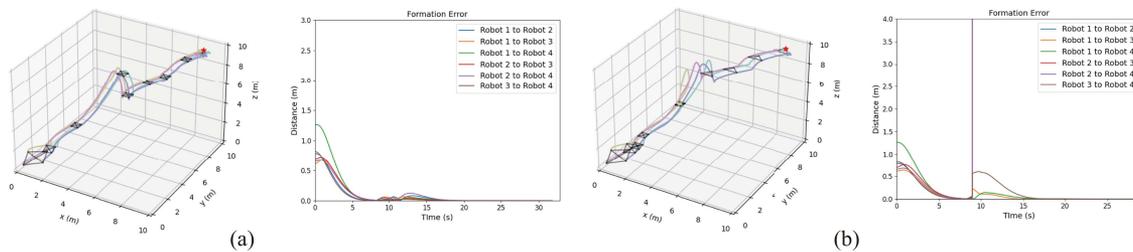


**Figure 14** (Color online) Motion trajectories and formation stability in various experimental scenarios under delayed environments. Formation (a) without, (b) with, and (c) with ideal delay prediction. Detour for obstacle avoidance (d) without, (e) with, and (f) with ideal delay prediction. Formation adjustment for obstacle avoidance (g) without, (h) with, and (i) with ideal delay prediction.

**Table 5** Minimum distance from the center of the UAV to the surface of the obstacle ( $C_{\text{delay}} = 0.08$ ) (m).

Experiment type	UAV1	UAV2	UAV3	UAV4
Experiment (d)	0.694	0.988	1.153	1.376
Experiment (e)	0.730	1.000	1.083	1.344
Experiment (f)	0.628	0.946	1.069	1.310
Experiment (g)	$\leq 0$	0.145	0.293	0.152
Experiment (h)	0.292	0.521	0.551	0.572
Experiment (i)	0.312	0.543	0.562	0.588

enabling timely adjustments to ensure task completion under special circumstances. Such systems can be effectively applied in challenging and hazardous environments to perform tracking tasks efficiently.



**Figure 15** (Color online) Fault motion under dynamic input disturbance. (a)  $f$  (0–1 N); (b)  $f$  (0–1.5 N).

## 5.6 Results and discussion

In the experiments presented in this paper, the method based on LLMs demonstrated significant flexibility and real-time adaptability, especially when faced with various environmental and target changes. Compared to traditional methods like reinforcement learning (RL) and model predictive control (MPC), the LLM method effectively reduces computational overhead and can adapt to environmental changes without requiring frequent adjustments or recalculations. This instant inference capability based on the language model allows the LLM method to quickly respond and make decisions in dynamic environments, without the need for extensive training as in RL, or real-time optimization and model updates as in MPC. This reasoning approach based on the language model enables LLM to avoid the computational bottlenecks and real-time issues common in traditional methods.

Furthermore, LLMs possess the ability to process and generate natural language, which provides a unique advantage in understanding and responding to human instructions and interacting with human users. The rich knowledge embedded in LLMs, derived from their training on vast amounts of text data, can be leveraged in decision-making processes, enhancing their effectiveness across various tasks. This knowledge-rich foundation allows LLMs to operate with a lower demand for task-specific training, thereby saving significant computational resources and time compared to methods like RL and MPC.

The flexibility and generality of LLMs make them suitable for a wide range of tasks, allowing them to adapt their behavior according to the context and specific requirements of different scenarios. This adaptability is particularly advantageous in dynamic and uncertain environments, where the ability to adjust strategies on-the-fly is crucial. Additionally, the scalability of LLMs means that their capabilities can be expanded by incorporating more data and computational power, making them well-suited to tackle increasingly complex tasks and environments.

Although RL and MPC methods perform well in certain scenarios, they typically require recalculating or adjusting strategies when faced with dynamic environments and changing targets, which is often associated with high computational costs, thereby limiting real-time responsiveness. In contrast, the LLM method can quickly adapt to environmental changes through fine-tuning of text-based strategies or event-driven adjustments, significantly reducing computational costs and improving real-time performance. This adaptability makes the LLM method particularly effective in dynamic and uncertain environments, especially in resource-constrained and time-sensitive scenarios, demonstrating greater flexibility and efficiency.

Although the advantages of LLM have not been comprehensively evaluated in this paper, particularly in terms of computational costs and energy efficiency, future research will further validate the performance of the LLM method in dynamic environments through experiments and provide a detailed comparison with traditional methods (such as RL and MPC). We anticipate that the LLM method will demonstrate substantial advantages when facing complex and uncertain multi-robot collaboration tasks.

## 6 Conclusion and future work

This study proposes a decision control technique based on LLMs to address formation and path planning challenges in MRSs. The method involves designing prompts tailored for different phases of motion, enabling LLMs to generate stage-specific goals. This guidance directs the entire MRS toward the final formation target, achieving both formation movement and path planning tasks. Simulation experiments have demonstrated the effectiveness of the method in completing these tasks while exhibiting intelligent obstacle-avoidance capabilities. Furthermore, LLMs can perform primary tasks even in environments

with communication delays and can reconfigure formations to safely guide remaining units in the event of unexpected faults, highlighting their role in enhancing the robustness and flexibility of MRSs.

Nevertheless, the study primarily considers static obstacles, revealing a slight deficiency in simulating dynamic obstacle avoidance for real-world scenarios. Additionally, the application of multimodal data fusion in specific scenarios has not been explored, particularly regarding its potential to reduce hallucinations and provide optimal strategies, which could further improve performance by enhancing data quality and domain-specific training. Future work should focus on dynamic obstacle avoidance and the application of multimodal language models in this field, as well as exploring feasible solutions to reduce hallucination issues.

**Acknowledgements** This work was supported in part by National Key Research and Development Program of China (Grant No. 2023YFB3307800), National Natural Science Foundation of China (Grant Nos. 62173147, 62373123), Major Science and Technology Project of Xinjiang (Grant No. 2022A01006-4), and Fundamental Research Funds for the Central Universities (Grant No. 222202517006).

**Supporting information** Appendixes A–C. The supporting information is available online at [info.scichina.com](http://info.scichina.com) and [link.springer.com](http://link.springer.com). The supporting materials are published as submitted, without typesetting or editing. The responsibility for scientific accuracy and content remains entirely with the authors.

## References

- 1 Ju C, Kim J, Seol J, et al. A review on multirobot systems in agriculture. *Comput Electron Agr*, 2022, 202: 107336
- 2 Liu Y, Nejat G. Multirobot cooperative learning for semiautonomous control in urban search and rescue applications. *J Field Robot*, 2016, 33: 512–536
- 3 Dai X, Jiang L, Zhao Y. Cooperative exploration based on supervisory control of multi-robot systems. *Appl Intell*, 2016, 45: 18–29
- 4 Tsiu L, Markus E D. A survey of formation control for multiple mobile robotic systems. *Int J Mech Eng Robot Res*, 2020, 9: 1515–1520
- 5 Sun Z, Xia Y. Receding horizon tracking control of unicycle-type robots based on virtual structure. *Int J Robust Nonlinear Control*, 2016, 26: 3900–3918
- 6 Xiao H, Chen C L P. Leader-follower consensus multi-robot formation control using neurodynamic-optimization-based nonlinear model predictive control. *IEEE Access*, 2019, 7: 43581–43590
- 7 Lee G, Chwa D. Decentralized behavior-based formation control of multiple robots considering obstacle avoidance. *Intel Serv Robot*, 2018, 11: 127–138
- 8 Khaledyan M, de Queiroz M. A formation maneuvering controller for multiple non-holonomic robotic vehicles. *Robotica*, 2019, 37: 189–211
- 9 Wei H, Lv Q, Duo N, et al. Consensus algorithms based multi-robot formation control under noise and time delay conditions. *Appl Sci*, 2019, 9: 1004
- 10 Yang Y, Juntao L, Lingling P. Multi-robot path planning based on a deep reinforcement learning DQN algorithm. *CAAI Trans Intel Tech*, 2020, 5: 177–183
- 11 Das P K, Jena P K. Multi-robot path planning using improved particle swarm optimization algorithm through novel evolutionary operators. *Appl Soft Comput*, 2020, 92: 106312
- 12 Tang B, Xiang K, Pang M, et al. Multi-robot path planning using an improved self-adaptive particle swarm optimization. *Int J Adv Robot Syst*, 2020, 17: 1729881420936154
- 13 Dai Y, Lee S G. Formation control of mobile robots with obstacle avoidance based on GOACM using onboard sensors. *Int J Control Autom Syst*, 2014, 12: 1077–1089
- 14 Bai C, Yan P, Pan W, et al. Learning-based multi-robot formation control with obstacle avoidance. *IEEE Trans Intell Transp Syst*, 2021, 23: 11811–11822
- 15 Fujimori A, Kubota H, Shibata N, et al. Leader-follower formation control with obstacle avoidance using sonar-equipped mobile robots. *Proc Institution Mech Engineers Part I-J Syst Control Eng*, 2014, 228: 303–315
- 16 Chen H B, Ji W K, Xu L F, et al. Multi-agent consensus seeking via large language models. 2023. [ArXiv:2310.20151](https://arxiv.org/abs/2310.20151)
- 17 Zhao M D, Jain S Y, Song S R. RoCo: dialectic multi-robot collaboration with large language models. 2023. [ArXiv:2307.04738](https://arxiv.org/abs/2307.04738)
- 18 Chen W Z, Koenig S, Dilkina B. Why solving multi-agent path finding with large language model has not succeeded yet. 2024. [ArXiv:2401.03630](https://arxiv.org/abs/2401.03630)
- 19 Chen Y C, Arkin J, Zhang Y, et al. Scalable multi-robot collaboration with large language models: centralized or decentralized systems? 2023. [ArXiv:2309.15943](https://arxiv.org/abs/2309.15943)
- 20 Ahn M, Brohan A, Brown N, et al. Do as I can, not as I say: grounding language in robotic affordances. 2022. [ArXiv:2204.01691](https://arxiv.org/abs/2204.01691)
- 21 Fu T, Li C, Liu J, et al. FlingFlow: LLM-driven dynamic strategies for efficient cloth flattening. *IEEE Robot Autom Lett*, 2024, 9: 8714–8721
- 22 Liu H, Zhu Y, Kato K, et al. Enhancing the LLM-based robot manipulation through human-robot collaboration. *IEEE Robot Autom Lett*, 2024, 9: 6904–6911
- 23 Oh K K, Park M C, Ahn H S. A survey of multi-agent formation control. *Automatica*, 2015, 53: 424–440
- 24 Hassanalian M, Abdelkefi A. Classifications, applications, and design challenges of drones: a review. *Prog Aerospace Sci*, 2017, 91: 99–131
- 25 Wang P K C. Navigation strategies for multiple autonomous mobile robots moving in formation. *J Robot Syst*, 1991, 8: 177–195
- 26 Ballotta L, Talak R. Safe distributed control of multi-robot systems with communication delays. 2024. [ArXiv:2402.09382](https://arxiv.org/abs/2402.09382)