Manned and Unmanned Aerial Vehicles Cooperative Combat Framework Based on Large Language Models

Hanyue Shi [1], Shaowei Li[2], Zian Huang[1], Shida Li[1], Ang Li[1]& Yaoming Zhou[1,*]

[1] School of Aeronautic Science and Engineering, Beihang University, Beijing, China
[2]AVIC Chengdu Aircraft Industrial (Group) Co., Ltd., Chengdu, China
*Corresponding author, E-mail: zhouyaoming@buaa.edu.cn

Abstract

The collaborative combat of manned and unmanned aerial vehicles is one of the principal directions for the future development of aerial combat systems, facing the challenge of excessive decision-making and operational burdens on pilots during UAV control. This paper proposes a novel framework for the collaborative combat of manned and unmanned aerial vehicles, incorporating large language model(LLM) into the process of aircraft pilots commanding and controlling multiple unmanned aerial vehicles (UAVs). The framework utilizes LLM for complex semantic understanding and monitoring of task instruction execution. It allows pilots to issue task instructions to UAVs using non-standard natural language. The received natural language task instructions are matched with the preloaded policy library of the designed task executor in UAVs, and an appropriate policy is selected for execution. During task execution, UAVs provide feedback on the task execution status to manned aircraft at key nodes, and continue task execution upon confirmation by manned aircraft until task completion or receipt of new task instructions. The framework is tested in typical beyond-visual-range combat scenarios of manned and unmanned aerial vehicle collaboration. It exhibits good human-machine interaction, robustness, trustworthiness, explainability, and effectively reducing the decision-making and operational burdens on pilots. The research findings of this paper can be widely applied to various task scenarios where humans and robots collaborate to accomplish tasks, providing a feasible technical route for the collaborative combat of manned and unmanned aerial vehicles.

Keywords: Manned and unmanned aerial vehicles cooperative; Large language model agent; Human-machine cooperation; Large language model control; Natural language control

1. Introduction

Military robots constitute vital components of modern defense systems and national emergency response frameworks. The realization of fully autonomous aerial unmanned combat aircraft as companions represents the ultimate objective pursued by nations globally. However, due to factors such as technological readiness, safety concerns, and ethical considerations, achieving this goal in the near term presents challenges. Presently, major airpower nations prioritize the coordinated operation of unmanned aerial vehicles (UAVs) with manned aircraft for aerial missions as a primary trajectory for advancing air combat capabilities in the future[1-4]. Substantial efforts have been devoted to this area of research and development.

Gangl et al. proposed a system framework for commanding multiple unmanned combat aerial vehicles from a single manned fighter aircraft, enabling the unmanned aircraft not only to possess perception, reasoning, decision-making, and action capabilities but also to effectively collaborate within the entire cooperative combat system based on the received cooperation information from friendly forces[5]. Johnson and Duran's collaborative system model centers on the team, utilizing the Observable, Predictable, and Directable (OPD) framework as guidance to help designers identify the needs for consistent team cooperation[6]. Chang et al. proposed a novel hybrid active collaboration framework for human-unmanned aerial vehicle transparent team collaboration based on collaborative design methods, decomposing complex tasks into a series of executable subtasks and employing deep reinforcement learning to train optimal policies and path planners for subtasks considering threats from enemy cluster behaviors[7]. This method can handle complex tasks and is relatively straightforward to apply in complex human-machine formations. In the literature, various human-machine interaction systems are proposed, including hybrid active interaction[8], adaptive autonomy[9], adjustable autonomy[10], cooperative control[11], etc. However, many of these are centered on autonomy, aiming to maximize the autonomy level of team members. The design of low interaction burden for manned and unmanned aircraft collaborative combat mechanisms remains a significant challenge[12].

The traditional approach of controlling unmanned aerial vehicles (UAVs) using ground stations is inadequate for coping with the complex, dynamic, and adversarial nature of the battlefield environment. For aerial missions with relatively low maneuverability requirements, placing UAV operators onboard aircraft enables manned-unmanned aerial vehicle collaboration. However, the addition of extra UAV operators results in increased weight and size of the aircraft, along with decreased maneuverability, particularly disadvantageous for fighter aircraft, especially carrier-based ones, when executing aerial combat missions. Fortunately, with advancements in unmanned aerial vehicle autonomous control and decision-making technologies, epitomized by artificial intelligence, the autonomy of UAVs continues to improve, enabling fighter aircraft pilots to simultaneously command and control both manned aircraft and UAV formations. The control paradigm for UAVs by fighter aircraft pilots evolves from full manual maneuver control to maneuver intent control and ultimately to task instruction issuance, allowing UAVs to autonomously execute tasks based on their own states and the battlefield situation. Notably, the application of large-scale language models such as Chat-GPT[13] plays a crucial role in alleviating the manipulation and decision-making burden on pilots when controlling UAVs.

Advanced large-scale language models are extensive network models pretrained on massive datasets using open-ended high-capacity architectures and self-supervised optimization techniques. They possess sophisticated semantic understanding capabilities and extensive knowledge reservoirs, enabling tasks such as text generation[13], emergency problem-solving[14], and code writing[15]. Intelligent agents designed based on LLM exhibit broad application potential in scenarios such as multimedia content generation, virtual character interaction, and robot planning and control. Within these intelligent agents, LLM serve as cognitive cores, collaborating with components such as monitoring and memory to accomplish tasks[16]. This paper introduces LLM into the interaction process between manned and unmanned aerial vehicles. An unmanned aerial vehicle intelligent agent based on LLM is designed. Furthermore, a framework for collaborative combat between manned and unmanned aerial vehicles with low interaction burden is constructed. Finally, tests are conducted in typical scenarios of manned and unmanned aerial vehicle collaborative combat missions.

The principal contributions of this work are enumerated below:

- Proposing a low interaction burden framework for collaborative combat between manned and unmanned aerial vehicles, incorporating LLM into the process of aircraft pilots commanding and controlling UAVs. This framework interprets natural language inputs from pilots, enabling fighter pilots to issue mission commands to UAVs in non-standard natural language. This significantly reduces the operational load on pilots for commanding and controlling UAVs, thereby enhancing the collaborative combat effectiveness between manned aircraft and multiple UAVs;

- Designed is a UAV mission executor incorporating preloaded policy library and other components. By selecting appropriate strategies, this executor autonomously controls UAV mission execution at the task level. This effectively reduces the cognitive and decision-making load on pilots;

- Developed is a critical collaborative combat status feedback mechanism where UAVs provide feedback to manned aircraft on mission execution status during operations. At key junctures such as weapon release, UAVs await authorization from manned aircraft before proceeding, thus mitigating conflicts arising from UAV autonomy with existing ethical, moral, and legal frameworks.

2. Collaborative Combat Simulation Environment

Multi-aircraft cooperative air combat, especially the collaborative combat between manned and unmanned aerial vehicles, is a highly complex aerial combat process that relies on coordinated cooperation to effectively engage and neutralize the enemy, which becomes particularly prominent in the context of manned and unmanned aerial vehicle collaborative combat. In order to validate the proposed framework for collaborative combat between manned and unmanned aerial vehicles based on LLM as described in subsequent sections, this paper constructs a high-fidelity simulation environment for manned and unmanned collaborative combat. The simulation environment includes models such as six-degree-of-freedom aircraft dynamics, medium-to-long-range air-to-air missile models, radar models, missile proximity warning equipment models, data link models, and flight state sensors. Specifically, the six-degree-of-freedom aircraft dynamics model is built based on the JSBSim library[17], and the medium-to-long-range air-to-air missile model comprises a five-degree-of-freedom missile dynamics model and guidance laws.

2.1 Six-Degree-of-Freedom Aircraft Dynamics Model

This study focuses on six-degree-of-freedom fixed-wing aircraft designed for beyond-visual-range combat missions. For the sake of convenience in testing and experimentation, the F-16 fighter aircraft dynamics model from the open-source JSBSim library is selected as the dynamic model for both manned and unmanned aircraft in this paper. The JSBSim library is an open-source aircraft dynamics model library developed in C++, which internally defines six-degree-of-freedom full kinematic equations for various aircraft, including the F-16 fighter aircraft, to simulate aircraft aerodynamic performance through a physics engine, providing high-fidelity simulation realism. Therefore, the JSBSim library is widely recognized and utilized in both academic and industrial domains. There exist differences in maneuvering capabilities between manned and unmanned aircraft. Due to platform constraints and cost considerations, the maneuvering capability of unmanned aircraft is generally inferior to that of manned aircraft. Therefore, restrictions are imposed on the maneuvering capability of unmanned aircraft in this paper, prohibiting the activation of engine thrust augmentation mode and limiting the maximum flight speed to 0.85 Mach. Additionally, under the condition of flight speed below 0.85 Mach and without engine thrust augmentation, the maneuvering performance of manned and unmanned aircraft is consistent. In this simulation environment, the control input variables for the aircraft dynamics model are [aileron deflection, elevator deflection, rudder deflection, throttle position], where all deflection angles are normalized inputs ranging from -1 to 1, and the throttle position ranges from 0.4 to 1.2, with the portion (1, 1.2] representing the activation of engine thrust augmentation mode.

2.2 Medium-to-Long-Range Air-to-Air Missile Model

In the simulation environment constructed in this paper, both the red and blue forces employ weapons of the same type. The weaponry exclusively comprises radar-guided medium-to-long-range air-to-air missiles. For computational simplification, it is assumed that the roll angle during missile flight is 0, and it is further assumed that any delays and errors associated with the execution of normal overload n_y and lateral overload n_z in the missile's implementation are negligible. Under these assumptions, the missile's dynamic equations are

$$\begin{cases} n_x = \dfrac{F_{\mathrm{T}} - F_{\mathrm{D}}}{mg} \\ F_{\mathrm{T}} = \text{-g} \cdot \mathrm{Isp} \cdot \dot{\mathrm{m}} \\ F_{\mathrm{D}} = \dfrac{1}{2} \cdot C_{\mathrm{D}} \cdot S \cdot \rho(z) \cdot v^2 \\ \dot{v} = \mathrm{g}\left(n_x - \sin(\theta)\right) \\ \dot{\varphi} = \dfrac{\mathrm{g}}{v} \cdot \dfrac{n_y}{\cos(\theta)} \\ \dot{\theta} = \dfrac{\mathrm{g}}{v} \cdot \left(n_z - \cos(\theta)\right) \end{cases} . \tag{1}$$

Where n_x, n_y, and n_z represent the overload in the three directions of the body axis system, measured in units of g (gravitational acceleration). n_y and n_z are determined by the guidance law. F_{T} and F_{D} denote the thrust of the rocket engine and the aerodynamic drag, respectively. Isp represents the specific impulse of the rocket engine. $\dot{\mathrm{m}}$ denotes the missile's mass loss rate due to burning fuel, as during missile flight, only the combustion of fuel results in mass loss. Therefore, $\dot{\mathrm{m}}$ also signifies the burning rate of rocket engine fuel. C_{D} is the aerodynamic drag coefficient of the missile. S represents the projected cross-sectional area of the missile in the direction of motion. $\rho(z)$ indicates the air density and z represents the altitude of missile flight. v denotes the absolute velocity of missile flight. v is a scalar quantity, calculated using the magnitude of the velocity vector $\boldsymbol{v}$ in the missile's flight velocity in the Earth coordinate system. φ denotes the heading angle of the missile. θ represents the pitch angle of the missile. $\dot{v}$ signifies the rate of change of the missile's flight speed magnitude. The calculation method for S is

$$S = \left(\frac{\mathrm{d}}{2}\right)^2 \cdot \pi + \mathrm{L}\left(\sin^2\left(\dot{\theta}\right) + \sin^2\left(\dot{\varphi}\right)\right). \tag{2}$$

Where d represents the diameter of the projectile and L denotes its length. Unless otherwise specified, the units of the above variables are in accordance with the International System of Units. φ is obtained by integrating the angular velocity $\dot{\varphi}$, and θ is obtained by integrating the angular velocity $\dot{\theta}$. The latest velocity v is calculated, and based on the azimuth angle φ and elevation angle θ, it is decomposed into the missile's velocity vector $\boldsymbol{v}$ in the Earth coordinate system. The missile employs proportional navigation guidance law in a lofted trajectory[18]. Throughout the missile's flight, the aircraft continuously provides target position and velocity information until the missile's onboard radar activates, searches for, and intercepts the target. In the event of the missile losing target information from its own aircraft, it extrapolates the target's potential position and velocity based on the last available target information before loss and calculates the required normal load factor accordingly. The missile utilizes a proximity fuze and automatically detonates upon nearing the target. Upon meeting termination conditions, the missile self-destructs, which include exceeding a finite flight time, flying below a minimum effective speed, or detonating.

2.3 Aircraft Avionics Equipment Model and Sensor Model

(1) Radar

In the simulation, the effective detection range of the onboard radar for a radar cross-section (RCS) 1m^2 target is 120km for manned aircraft and 96km for unmanned aircraft. The only difference between the two is in the detection range, while all other performance parameters remain identical. The azimuth and range detection errors of the radar are 0.1%, and the errors follow a Gaussian distribution. The radar scan center is set to the body-axis direction, with a maximum scan elevation angle of 80 degrees (±40 degrees) and a maximum scan azimuth angle of 120 degrees (±60 degrees). Radar maximum effective interception range calculation formula is[19]

$$R_{\max} = \sqrt[4]{\mathrm{RCS}} \times R \ . \tag{3}$$

Where R is the detection distance for a target with an RCS of 1m^2. When the target distance is less than the maximum effective detection range $R_{\max}$ and the target is within the radar scanning area, the radar intercepts the enemy aircraft. Upon interception, the radar records and returns the status information of the enemy aircraft. The information is returned in the form of a list, including the following details of the target aircraft: [unique identifier, friend or foe identification, model, azimuth, elevation, approach angle, longitude, latitude, altitude, velocity vector, acceleration vector].

(2) Missile Proximity Warning Equipment Model

The simulation environment assumes that when the enemy missile's onboard radar is activated and the aircraft is within its scanning range, the aircraft can receive warning information regarding the approaching missile. The warning information only includes the direction of the incoming missile relative to the aircraft.

(3) Data Link Model

The simulation environment incorporates a data link model for communication and situational information sharing among aircraft within the same coalition. The model is activated when the aircraft is equipped with compatible data link types and the inter-aircraft distance meets the requirements. It is assumed that data transmission via the data link is free from delays, packet loss, and bandwidth constraints.

(4) Sensor Model

A sensor model with Gaussian noise is employed to obtain the aircraft's position, velocity, acceleration, and attitude angles. It is assumed that the data collection frequency of each sensor matches the communication frequency of the data link.

3. Overall Design of Manned and Unmanned Aircraft Collaborative Combat Framework Based on LLM

3.1 Structure of Collaborative Combat Framework

The collaborative combat framework for manned and unmanned aircraft based on LLM consists of a manned aircraft platform, UAV platform, UAV LLM Mission Manager, and UAV Mission Executor, as depicted in the Figure 1. The manned aircraft platform and UAV platform form a combat formation to cooperate with each other to achieve mission objectives.

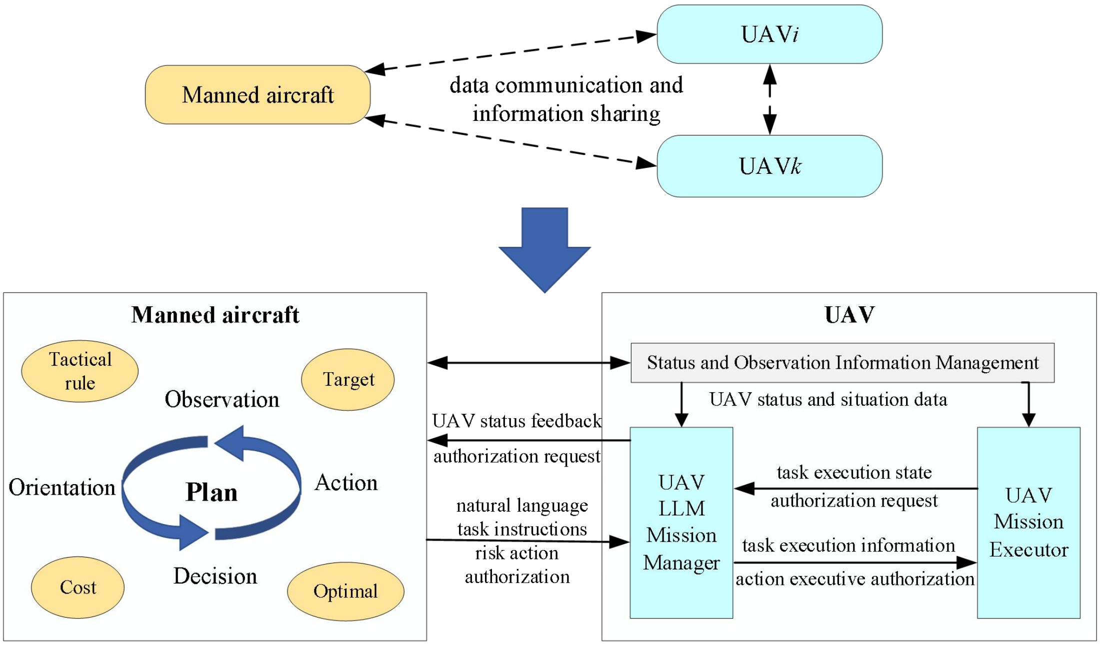

Figure 1 – Structure of manned and unmanned aircraft collaborative combat framework

The manned aircraft platform is operated and controlled by human pilots. It acquires the status and observational information of all UAVs within the formation via data links. Integrated with its own status and radar observation information, the manned aircraft platform plans the tasks to be executed by each UAV within the formation, generates natural language task instructions, and transmits them to the individual UAVs through data links. Additionally, the manned aircraft platform monitors the task execution status of each UAV and authorizes high-risk actions that the UAVs need to execute.

The UAV platform serves as the carrier for both the UAV LLM Mission Manager and the UAV Mission Executor. It executes control commands provided by the UAV Mission Executor, such as control surface deflections and engine throttle settings, to accomplish the tasks designated by the manned aircraft. Equipped with various weapons and avionics, the UAV platform is capable of conducting aerial combat missions involving detection, guidance, and attack, with a certain level of maneuverability to evade enemy threats. It possesses the capability to coordinate attacks with manned aircraft, facilitating missile guidance for other aircraft and engaging in data sharing with them via data links.

The UAV LLM Mission Manager serves as the brain of the UAV platform. Its primary role involves processing natural language task instructions received from the manned aircraft, matching them with the Policies preloaded in the UAV mission executor's policy library, and storing the task instructions along with the policy information in the Task Execution Information Recording Module. Authorization from the manned aircraft is sought by the UAV LLM Mission Manager before executing high-risk maneuvers.

The UAV Mission Executor acts as the executor of the UAV platform. Its main functions include providing executable preloaded policy information to the UAV LLM Mission Manager for matching with natural language task instructions and executing the Policies arranged by the mission manager to assist the manned aircraft in achieving mission objectives.

3.2 UAV LLM Mission Manager

The composition modules of the UAV LLM Mission Manager mainly include the Natural Language Processing Module, Task Queue Module and Task Execution Information Recording Module, as illustrated in the Figure 2 depicting their interrelationships.

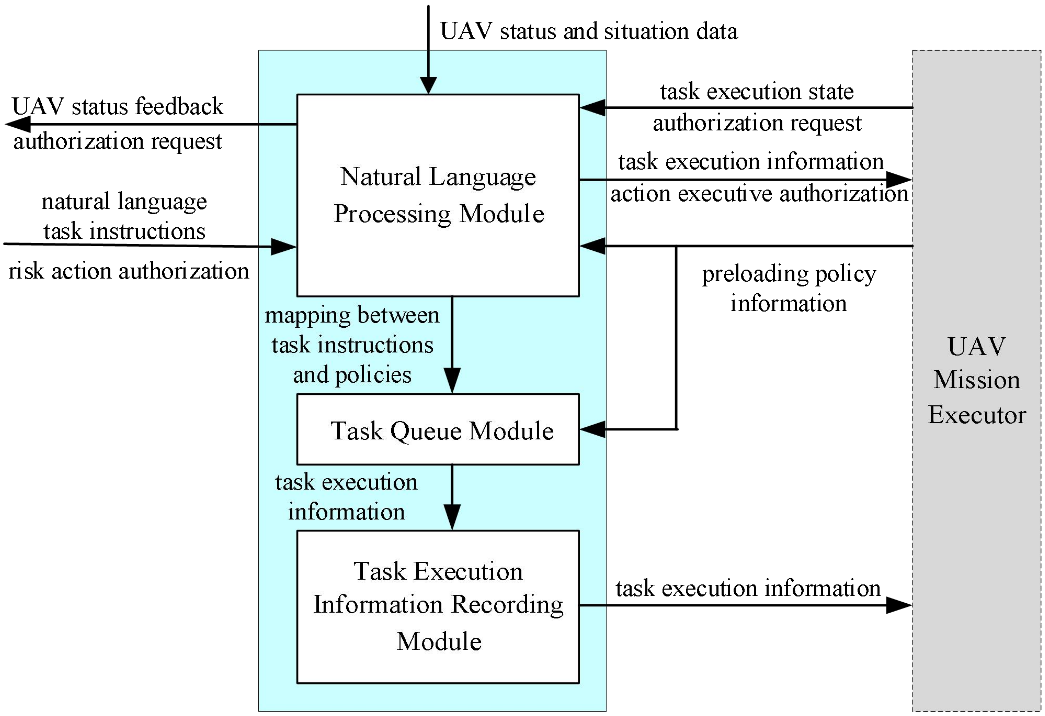

Figure 2 – Structure of the UAV LLM Mission Manager

3.2.1 Natural Language Processing Module

The Natural Language Processing Module serves as the core of the UAV LLM Mission Manager. It comprises submodules including the Natural Language Regularization Submodule, LLM Submodule, LLM Response Processing Submodule and Historical Dialogue Storage Submodule, and as illustrated in the Figure 3. Its primary functions include natural language task instruction recognition and determination of the execution status of tasks on the UAV platform.

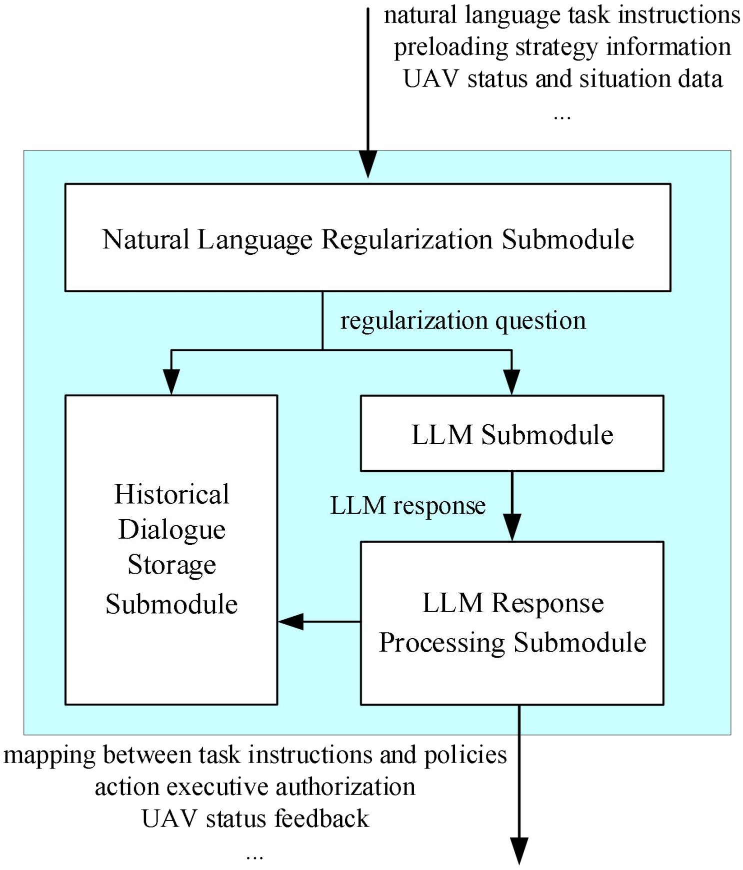

Figure 3 – Structure of the Natural Language Processing Module

(1) Natural Language Task Instruction Recognition

In the process of natural language task instruction recognition, a LLM deployed on the UAV platform parses the pilot's natural language task instructions. Subsequently, the task instruction content is compared with the policies in the Preloaded Policy Library of the task executor, and the number of the matching policy is returned for subsequent process recording and execution. The workflow is as follows: 1) Natural language task instructions received via data link are processed through the Natural Language Regularization Submodule to obtain standardized questions. The purpose is to enable the LLM to more accurately recognize and understand the questions and provide standardized answers; 2) The standardized questions are inputted into the LLM Submodule, and then the LLM's answers are awaited; 3) The LLM Response Processing Submodule processes the answers to obtain results that the program can use.

(2) UAV Platform Task Execution Status Determination

Through deploying a LLM on the UAV platform, the execution status of tasks on the UAV platform is determined. The LLM is used to compare the UAV's own flight status and situational information with the termination conditions of the subtasks defined by the policy to determine whether the policy subtask should be terminated and subsequent tasks executed. The process is similar to natural language task instruction recognition, but the difference lies in the combination of the UAV's flight status and situational information with the termination conditions of policy subtasks to generate the problem for determining the execution status of unmanned aerial vehicle platform tasks. Finally, the dialogue record of this Q&A session is stored in the Historical Dialogue Storage Submodule.

(3) Unmanned Aerial Vehicle Task Execution Status Feedback

Based on the results of determining the execution status of tasks on the UAV platform or the evaluation results of the Task Event Evaluator in the UAV Mission Executor, feedback is provided to the human operator. This feedback is given through changes in icons and text, such as variations, flashing, and color changes, to alert the human operator to changes in the execution status of UAV tasks.

3.2.2 Task Queue Module

The task queue is utilized to log the task instructions issued by the human operator to the UAV platform and their corresponding relationships with policies, adhering to the first-in, first-out (FIFO) principle. When the executing task instruction terminates, it is dequeued, and subsequent task instructions are then executed. If the task instruction is the last one, the default subsequent policy and task parameters are selected for execution. Information regarding the task instruction currently being executed is stored within the Task Execution Information Recording Module.

3.2.3 Task Execution Information Recording Module

The Task Execution Information Recording Module records information about the task instructions currently being executed, encompassing details such as natural language task instruction content, policy number, policy subtask execution number, policy execution duration, subtask execution duration, and other relevant information. It records task execution information of the UAV platform and provides benchmark information for task execution status determination. The information is stored in a dictionary structure for convenient storage and retrieval. The task instruction information stored in the Task Execution Information Recording Module is aligned with the executed task instructions in terms of both time and content. Hence, when the executed task instruction changes, the task instruction information stored in the Task Execution Information Recording Module also changes correspondingly.

3.3 UAV LLM Mission Manager

The task executor mainly consists of three components: Task Event Evaluator, Preloaded Policy Library, and Action Executor, as illustrated in the Figure 4 depicting their interrelationships. The function of the Task Event Evaluator is to assess the task execution status and determine whether feedback to the human operator is necessary. The Preloaded Policy Library comprises policies tailored to task scenarios and UAV performance, capable of controlling UAVs to complete various subtasks. The Action Executor is responsible for converting action numbers generated by policies into control quantities for UAV execution, including control surface deflections, throttle settings, weapon launch commands, and radar control commands.

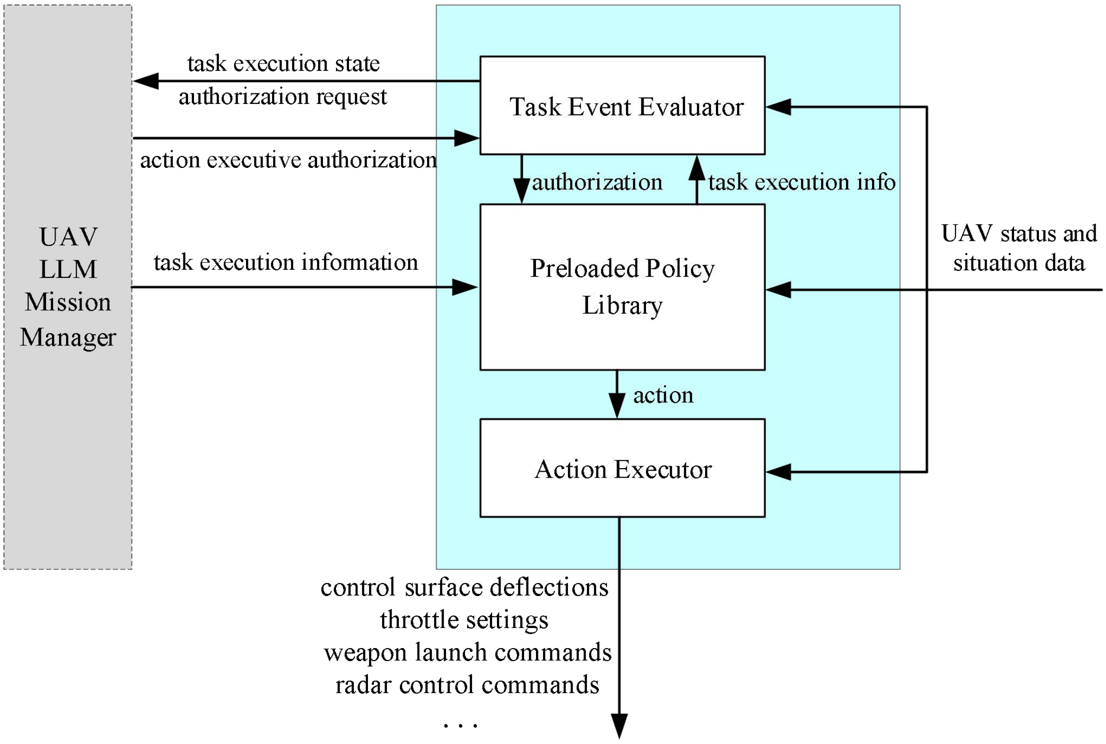

Figure 4 – Structure of UAV Mission Executor

3.3.1 Task Event Evaluator

The Task Event Evaluator processes and analyzes its own flight status and radar information to determine if predefined events have been triggered. Predetermined events primarily include objective events such as self-state events, situational awareness events, etc., including but not limited to target interception, target disappearance, weapon launch preparation, successful weapon launch, missile proximity warning, equipment failure, low altitude warning, etc. Each objective event corresponds to a unique code. Upon the triggering of an objective event, the corresponding unique code is transmitted to the UAV LLM Mission Manager for task execution status feedback. Particularly for high-risk events such as weapon launch and low-altitude dive, the mission manager needs to report to the human operator to obtain corresponding execution permissions before execution.

3.3.2 Preloaded Policy Library

The preloaded policy library stores the policy set loaded onto the UAV platform before the commencement of operations. The policies in the set are designed and constructed for specific UAV models. Different sets of policies are loaded for different numbers, models, configurations, and mission objectives of UAVs. The loaded policy set should meet the capability requirements of UAVs to cooperate with human operators in achieving operational objectives during task execution. The policy set in the Preloaded Policy Library is divided into two parts: a general flight task policy set and a dedicated task policy set. The policies in the general flight task policy set are universal policies that may be used for all airborne combat missions, such as formation flying with manned aircraft, disengaging from enemy targets and returning, etc. The dedicated task policy set comprises policies specifically needed by UAVs to achieve operational objectives, such as area search policies, line search policies, multi-aircraft cooperative search policies, etc., for ground search missions, and attack launch policies, cooperative guidance policies, missile warning maneuver policies, etc., for airborne combat missions. Policies are saved in the form of dictionaries in the preloaded policy library, with saved information including policy number, policy description, default subsequent policy, subtask information, and policy decision maker. The subtask information includes subtask number, subtask type, subtask content description, and subtask termination conditions. The decision maker makes action selections based on its own flight status, self-observed data, teammate-shared data, and the execution policy and subtask numbers. The selected action number is then transmitted to the Action Executor for execution.

3.3.3 Action Executor

The Action Executor controls the aircraft through the control quantities of the three aircraft channels, achieving designated maneuver actions. The maneuver controller consists of three nested layers from top to bottom, namely the tactical action number controller, the maneuver action number controller, and the meta-maneuver controller. Its structure is illustrated in the Figure 5. Through the design of the three-layer maneuver controller, all tactical actions of the UAV are considered as sequences of rolls, overloads, and speed increases/decreases, aligning with the operational habits of human pilots and airborne combat requirements, and possessing strong reliability, interpretability, and transferability. The top layer is the tactical action number controller, which computes the maneuver action number, target altitude, speed, overload, and target attitude parameters selected, which are input into the intermediate layer's maneuver action number controller to realize tactical actions such as missile evasion, turning towards enemy aircraft, and turning towards friendly aircraft. The intermediate layer, the maneuver action number controller, calculates the desired overload, desired roll angle, and desired speed based on the input parameters, realizing five types of maneuvers including level flight acceleration/deceleration, given gradient climb/descent, horizontal turns, and barrel rolls. The bottom layer, the meta-maneuver controller, uses PID control methods to track the desired overload, desired roll angle, and desired speed, controlling the deflection of control surfaces and throttle lever positions to manipulate the aircraft to the target attitude.

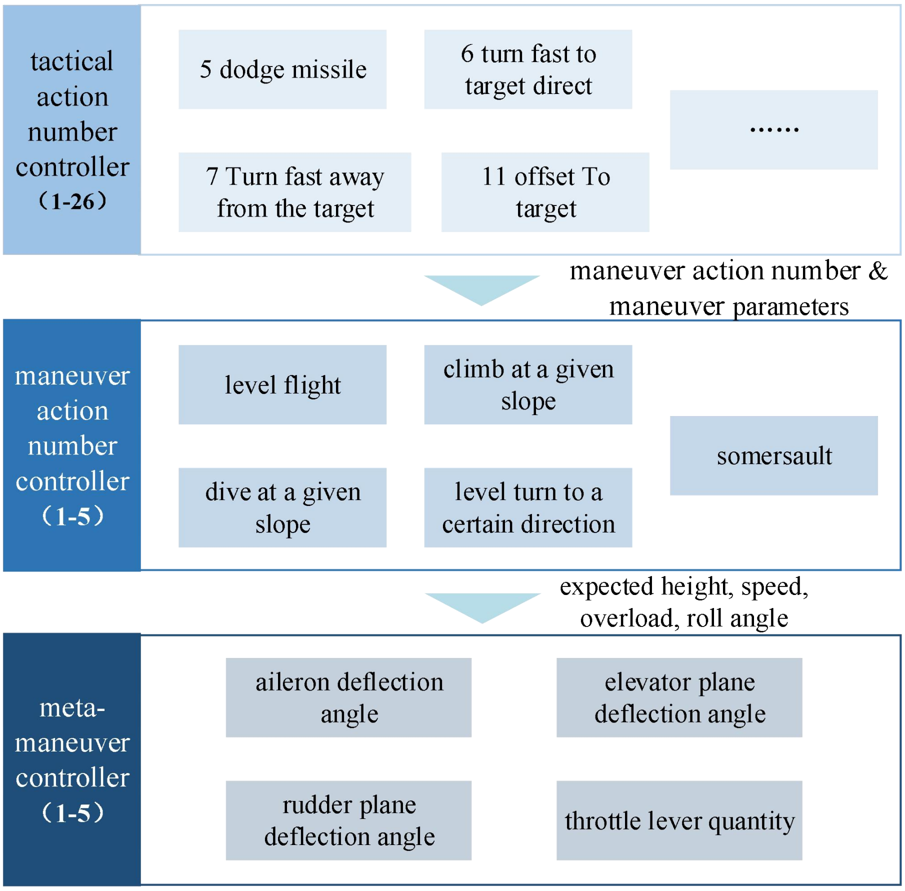

Figure 5 – Structure of Action Executor

4. Construction of Manned and Unmanned Aerial Vehicle Collaborative Combat Framework Based on LLM

In response to typical scenarios of manned-unmanned aerial combat cooperation, the manned and unmanned aerial vehicle collaborative combat framework proposed in Chapter 3 is applied to solve aerial combat missions.

4.1 Task Description

Task Background: The red side's early warning radar detected a formation of blue fighter jets intruding into the alert zone. The red side dispatched manned and unmanned combat aircraft, led by a manned aircraft, to intercept.

Task Objective: The operational objective of the red side is to ensure the survival of manned aircraft while shooting down or repelling all blue aircraft.

Task Area: The operational area is centered at 140 degrees east longitude and 60 degrees north latitude, with a radius of 120 kilometers, encompassing airspace ranging from 2 to 15 kilometers in altitude.

Composition of Forces: The red side's composition consists of one manned combat aircraft leading two unmanned combat aircraft. The blue side's composition consists of two manned combat aircraft. Manned combat aircraft on both red and blue sides share identical configurations and performance characteristics. Unmanned combat aircraft have inferior configurations and performance compared to manned aircraft. For ease of experimental scenario design, both manned and unmanned aircraft utilize the F-16 dynamics model. A comparison of performance indicators between manned and unmanned aircraft is presented in the Table 1.

Table 1 – Performance configuration for manned and unmanned aircraft

Performance	Manned Aircraft	UAV
Aircraft dynamics model	F-16 six-degree-of-freedom dynamics model (JSBSim library)	F-16 six-degree-of-freedom dynamics model (JSBSim library); Engine reheat mode activation not permitted; Maximum flight speed 0.85 Mach
Maximum permissible overload	6g	6g
Weapon configuration	AIM-120D * 8	AIM-120D * 4
Radar cross section (RCS)	circumference: $1m^2$	frontal: $0.01m^2$; other directions: $1m^2$
Effective detection range for $1m^2$ RCS target by radar	120km	100km
Data link communication distance	200km	200km
Missile approach warning capability	Yes	Yes

Initial Battlefield State: The initial state of the battlefield primarily encompasses the spawn positions of both sides and their relative dispositions. The red faction adopts an equilateral triangular formation with a manned aircraft at the rear and two unmanned aircraft at the front. The initial distances between the red faction's aircraft are randomly set between 15 kilometers and 45 kilometers. The blue faction's two manned aircraft are positioned in a tight formation with an approximate spacing of 4 kilometers. The initial distance between the blue and red factions is approximately 150 kilometers. Both sides start at an altitude of 8 kilometers, with initial attitudes set to straight and level flight.

Figure 6 – The initial positions of both the red and blue sides

4.2 Operational Capability Requirement Analysis

Based on the task scenario setup outlined in Section 4.1, the entire airborne combat mission process can be divided into cycles of planning, reconnaissance, and execution phases. Each phase can be further subdivided into several tasks. For each task, based on the disparity between the actual capabilities and required capabilities of each aircraft in the formation, aircraft are classified as supporters or non-supporters. Supporters are capable of independently completing the task, while non-supporters are unable to fulfill the task. The requirement analysis is illustrated in the Table 2.

Table 2 – Operational capability requirement analysis

Stage	Task	Capability Requirements	Manned Aircraft Support	UAV Support	Explanation
Planning	Target allocation	Analyze task requirements and plan; understand the status, capabilities, and dependencies of each aircraft in the formation	Yes	No	Manned aircraft pilots need to plan tasks for unmanned aircraft; unmanned aircraft cannot independently analyze and understand task requirements.
	Maneuver planning	Analyze task requirements and plan	Yes	No	Unmanned aircraft need policies set by manned aircraft pilots, including maneuver planning schemes.
Reconn-aissance	Target search	Search for enemy units using radar and other avionic equipment	Yes	Yes	Manned aircraft carry powerful active radars, while unmanned aircraft carry slightly weaker active radars.
	Target tracking	Ability to identify, locate, and track enemy targets	Yes	Yes	Both manned and unmanned aircraft have the capability to identify critical targets.
Attack	Attack instruction confirmation	Confirm attack targets; have the authority to launch missiles at targets	Yes	No	Manned aircraft have the authority to confirm targets and engage them; unmanned aircraft need authorization from manned aircraft before engaging targets.
	Missile launch	Equip air-to-air missiles and have the capability to launch them at targets	Yes	Yes	Both manned and unmanned aircraft can launch missiles at confirmed targets.
	guidance	Ability to transmit target information obtained to missiles via the aircraft's data link	Yes	Yes	Both manned and unmanned aircraft can guide launched missiles using fire control radar, as long as the radar is operational and can track targets and missiles.
	Cooperative guidance	Ability to transfer missile guidance authority to other aircraft in the formation; ability to	Yes	Yes	Both manned and unmanned aircraft can transfer missile guidance authority to each other using fire control radar,

		accept missile guidance authority from other aircraft in the formation			inter-aircraft data links, and aircraft-to-missile data links; this transfer occurs automatically when an aircraft is unable to maintain missile guidance capability.
Defense	Missile approach warning	Carry missile approach warning equipment	Yes	Yes	Both manned and unmanned aircraft are equipped with missile approach warning devices and can receive information about missiles locking onto them.

4.3 Detailed Design of Cooperative Combat Framework

This section provides a detailed design of the cooperative combat framework for manned and unmanned aircraft based on large language models applied to airborne combat missions. A test case of the proposed cooperative framework is presented. Considering the characteristics of typical manned and unmanned cooperative airborne combat mission scenarios, the cooperative combat framework, as illustrated in Figure XX, is elaborately designed for manned and unmanned aircraft to collaborate in airborne combat.

4.3.1 Detailed Design of Cooperative Combat Framework

(1) Task Event Evaluator

Design a Task Event Evaluator suitable for airborne combat missions. The task events involved include target detection, target disappearance, weapon launch preparation, successful weapon launch, missile approach warning, equipment malfunction warning, low altitude warning, policy subtask switching, and task instruction completion. The judgment criteria and unique codes for each objective event are shown in the Table 3.

Table 3 – Task event information

Unique Code	Event Name	Judgment Criteria
UAV_E_001	Target detection	Number of targets intercepted by radar > Number of targets intercepted by radar in the previous time step
UAV_E_002	Target disappearance	Number of targets intercepted by radar < Number of targets intercepted by radar in the previous time step
UAV_E_003	Weapon launch preparation	Weapon launch command = True
UAV_E_004	Weapon launch successful	Number of weapons held at the current time step < Number of weapons held in the previous time step
UAV_E_005	Missile approach warning	Number of warning alerts triggered by missile approach warning equipment > Number of alerts triggered in the previous time step
UAV_E_006	Equipment malfunction warning	Number of self-checking equipment malfunctions > 0
UAV_E_007	Low altitude warning	Aircraft altitude above ground < 2000m
UAV_E_008	Policy subtask switching	Trigger termination conditions for policy subtasks
UAV_E_009	Task instruction completion	Trigger termination conditions for policies

(2) Preloaded Policy Library

In response to the requirements of cooperative airborne combat missions involving manned and unmanned aircraft, as well as the performance and equipment configuration of unmanned aircraft, the preloaded policies include formation flying policy with manned aircraft, disengagement from enemy target policy, airborne target search policy, attack launch policy, missile guidance policy, relay guidance policy, missile warning maneuver policy, and collision avoidance policy. The execution of actions is carried out by the Action Executor designed in Section 3.3. To enhance the safety of unmanned aircraft mission execution and reduce reaction time, the missile warning maneuver policy and collision avoidance policy in the preloaded policies have higher priority. Upon meeting trigger conditions, they interrupt the current task immediately for execution. After the completion of policy execution, the previous task is automatically resumed without the need for intervention or control by manned aircraft.

Table 4 – Preload policies information

Policy Code	Policy Name	Policy Description	Default Successor Policy	Subtask Information
UAV_DM_001	Formation flying with manned aircraft	Form a flight formation with manned aircraft	None	See Table 5 for details.
UAV_DM_002	Disengagement from enemy target	Fly away from target X	Formation flying with manned aircraft	See Table 6 for details.
UAV_DM_003	Target search	Search for targets in the X direction	Missile launch	See Table 7 for details.
UAV_DM_004	Missile launch	Launch missiles towards target X	Missile guidance	See Table 8 for details.
UAV_DM_005	Missile guidance	Guide missiles towards target X	Airborne target search	See Table 9 for details.
UAV_DM_006	Relay guidance	Take over guidance of missiles launched by friendly aircraft towards target X	Airborne target search	See Table 10 for details.
UAV_DM_007	Missile warning maneuver	Take evasive maneuvers to evade enemy missiles	Previous task	Automatically triggered when missile warning exists. Aircraft turns away from the direction of incoming missiles and flies straight at maximum speed. End condition is when all warnings disappear.
UAV_DM_008	Collision avoidance	Adopt maneuvering policy to avoid friendly aircraft	Previous task	Automatically triggered when distance to friendly aircraft is within 1000 meters. Aircraft turns away from the direction of the friendly aircraft and flies straight. End condition is when the distance to all friendly aircraft is greater than 1000 meters.

Table 5 – Formation flying with manned aircraft policy subtask information

Subtask Number	Subtask Type	Subtask Type	Subtask Termination Condition
1	Climb	Climb to the same altitude as manned aircraft	Absolute error between aircraft altitude and manned aircraft altitude is less than 100m
2	Turn	Turn in the direction of manned aircraft	Absolute error between aircraft direction and manned aircraft direction is less than 5 degrees
3	Level flight	Fly at a speed not exceeding 0.85Ma to track the manned aircraft	None

Table 6 – Disengagement from enemy target policy subtask information

Subtask	Subtask	Subtask Type	Subtask Termination Condition

Number	Type		
1	Turn	Turn in the opposite direction of Target X	Absolute error between aircraft direction and opposite direction of Target X is less than 5 degrees
2	Level flight	Fly at 0.85Ma speed	Distance from enemy target exceeds 100km

Table 7 – Target search policy subtask information

Subtask Number	Subtask Type	Subtask Type	Subtask Termination Condition
1	Turn	Turn in the direction of Manned Aircraft X	Absolute error between aircraft direction and direction of Manned Aircraft X is less than 5 degrees
2	Climb	climb to 8km altitude	Absolute error between aircraft altitude and 8000m is less than 100m
3	Radar on	Turn on radar and continue scanning	Radar is set to ON
4	Level flight	Fly at 0.85Ma speed	New target discovered

Table 8 – Missile launch policy subtask information

Subtask Number	Subtask Type	Subtask Type	Subtask Termination Condition
1	Turn	Turn in the direction of Target X	Absolute error between aircraft direction and direction of Target X is less than 10 degrees
2	Radar on	Turn on radar, track, and lock onto target	Information about Target X is obtained
3	Missile launch prep	Request launch authorization from manned aircraft	Authorization for launch granted by manned aircraft driver
4	Missile launch	Launch a missile at Target X	Successful weapon launch

Table 9 – Missile guidance policy subtask information

Subtask Number	Subtask Type	Subtask Type	Subtask Termination Condition
1	Dive	Dive to 5000m altitude	Absolute error between aircraft altitude and 5000m is less than 100m
2	Level flight	Fly straight towards Target X	All missiles launched by this aircraft at Target X are either locked on or invalidated

Table 10 – Relay guidance policy subtask information

Subtask Number	Subtask Type	Subtask Type	Subtask Termination Condition
1	Receive guidance	Receive guidance authority for Target X from friendly aircraft	Successful receipt of missile guidance authority from friendly aircraft
2	Dive	Dive to 5000m altitude	Absolute error between aircraft altitude and 5000m is less than 100m
3	Level flight	Fly straight towards Target X	All missiles guided by this aircraft at Target X are either locked on or invalidated

4.3.2 Cooperative Air Combat UAV LLM Mission Manager

(1) Natural Language Task Instruction Recognition

Taking the natural language task instruction "Aircraft 1 search for targets in the direction of 30 degrees" as an example:

1) First, the UAV platform needs to determine whether it needs to execute the command. A standard format question is constructed by the Natural Language Regularization Submodule, in the format of: "This aircraft is Aircraft [Aircraft Number]. Does it need to execute the task command [Instruction Content]? If yes, return the number 1, otherwise return the number 0. !!! Only return numbers !!!", where Instruction Content is the natural language instruction received by the UAV. For example, "This aircraft is Aircraft 1. Does it need to execute the task command 'Aircraft 1

search for targets in the direction of 30 degrees'? If yes, return the number 1, otherwise return the number 0. !!! Only return numbers !!!".

2) The answer returned by the LLM in 1) is processed by the LLM Response Processing Submodule to obtain the judgment result. If it is 0, the subsequent steps are skipped; if it is 1, the subsequent steps are continued.

3) Extract the instruction content part of the task instruction, and combine it with the description information of each policy in the Preloaded Policy Library. Construct a standardized question using the Natural Language Regularization Submodule as follows: "[Instruction Content] is most matched with which task among the following: 1. [Policy Description 1]; 2. [Policy Description 2], ..., N. [Policy Description N], !!! Only return numbers !!!", for example "'Search for targets in the direction of 30 degrees' is most matched with which task among the following: 1. Form a flight formation with manned aircraft; 2. Search for targets in the X direction; …; N. Fly away from target X, !!! Only return numbers !!!".

4) The answer returned by the LLM in (3) is processed by the LLM Response Processing Submodule to obtain the matching relationship between the task instruction and the policy.

(2) UAV Platform Task Execution Status Judgment

Taking the task instruction "Search for targets in the direction of 30 degrees" that Aircraft 1 is about to execute as an example, assuming Aircraft 1 is currently maintaining a height of 5000 meters with the radar turned off and flying in the same direction as the manned aircraft:

1) Based on the execution policy number and subtask number recorded in the Task Execution Information Recording Module, retrieve the corresponding termination condition of the subtask from the Preloaded Policy Library. For example, when the aircraft executes subtask number 2, retrieve its termination condition from Table X as: "The absolute error between the aircraft's altitude and 8000 meters is less than 100 meters".

2) Based on the aircraft's own flight status and situational information, combined with the subtask termination condition extracted in 1), construct a standardized question structure via the Natural Language Regularization Submodule as follows: "The aircraft is in [Subtask Type]. At this time, the aircraft [Description of the degree to which flight status/situational information meets the subtask termination condition], and the termination condition of the task is [Subtask Termination Condition]. Does the [Subtask Type] task need to be terminated? If yes, return the number 1, otherwise return the number 0, !!! Only return numbers !!!". For example, "The aircraft is in [climbing] task. At this time, the aircraft [altitude 5000 meters, target altitude 8000 meters, absolute error 3000 meters], the termination condition is [the absolute error between the aircraft's altitude and 8000 meters is less than 100 meters], does the [climbing] task need to be terminated? If yes, return the number 1, otherwise return the number 0, !!! Only return numbers !!!".

3) The answer returned by the LLM in 1) is processed by the LLM Response Processing Submodule to obtain the judgment result. If it is 0, the subsequent steps continue to execute the subtask; if it is 1, the subtask is terminated, and the next subtask is continued to be executed. If the subtask is the last subtask in the sequence, switch to the subsequent policy for execution based on the Task Queue Module.

4) Save the question and answer to the Historical Dialogue Storage Submodule.

4.4 Red Team Manned Aircraft Pilot

In the selected mission scenario, the UAV in the red team formation needs to be planned, target allocated, and attack authorized by the manned aircraft pilot. The UAV autonomously executes the planning results of the manned aircraft. To highlight the characteristics of the manned and unmanned aerial vehicle collaborative combat framework proposed in this paper, a command-line interaction method is used to represent the pilot's target allocation and maneuver planning for both manned and unmanned aircraft. For manned aircraft, the following command format is used for control: [tactical action number, weapon launch command, radar command]. For unmanned aircraft, the following format is used for natural language instruction setting: "[Unmanned aircraft number] unmanned aircraft (to target [blue target number]) executes [policy description] task". The task event evaluator selected in Section XX and the feedback data of unmanned aircraft status are used as triggering conditions for command-line interaction. Task events, unmanned aircraft status feedback information, and observation data of aircraft in the formation are used as the basis for planning. During the command-line interaction process, the combat simulation program is paused, and after receiving the command-line input command, the simulation program continues to run. The influence of the time consumed by the command-line interaction process and the LLM reasoning on the combat process is ignored.

4.5 Blue Team Manned Aircraft Pilot Decision Model

The blue team manned aircraft pilot uses an expert system decision model, the main workflow of which is shown in the figure. Blue team aircraft maintain a height of 8000 meters and a speed of 1.2 Mach during combat if missile guidance is not being conducted, keeping the radar open to search for red team targets. When a red team target is detected, the aircraft turns toward the enemy target. After the red team target enters the missile range, missiles are immediately launched to attack the enemy target. The minimum launch interval for missiles targeting each enemy target is 40 seconds. When conducting missile guidance, blue team aircraft maintain flight at a height of 5000 meters and a speed of 0.85 Mach, keep the radar open, and face the target. After the missile proximity warning is triggered, the aircraft turns away from the direction of the incoming missile, accelerates, and flies straight. After the missile proximity warning disappears, the aircraft turns back towards the red team target to continue the attack and guidance.

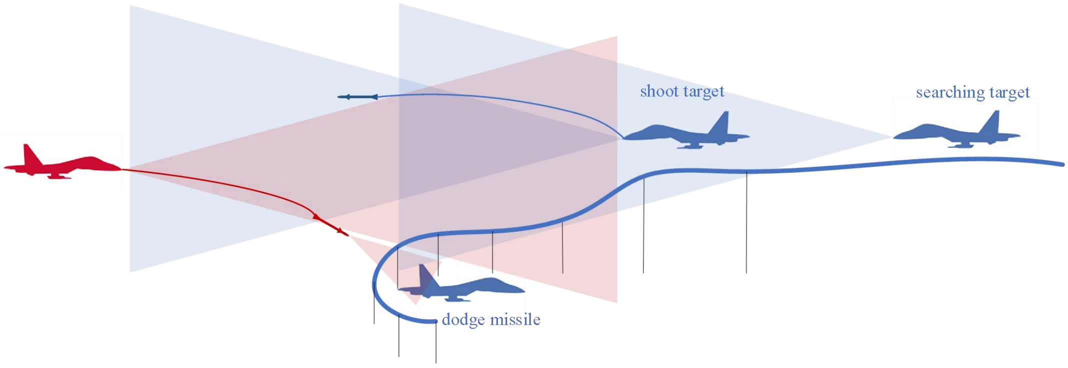

Figure 7 – Blue team manned aircraft pilot decision model operational diagram

5. Verification Experiment of Manned and Unmanned Aerial Vehicle Collaborative Combat Framework Based on LLM

5.1 Experimental Conditions

The experiment was conducted within the manned and unmanned aircraft collaborative combat simulation environment developed in Chapter One, with mission scenarios as outlined in Section 4.1.

The combat simulation environment runs on the Windows 10 operating system, configured with Python 3.7 compilation environment and related dependencies. Additionally, the JSBSim library, version 1.1.6, is installed. Tacview software, version 1.9.1, is required for replay data visualization.

Training and deployment of LLM are costly endeavors. Given the experimental nature of this study to validate the collaborative combat framework, open-source LLM were utilized through API calls. The LLM used in this study is the Spark3.5 Max version developed by the ANHUI USTC iFLYTEK Co Ltd of China. The model settings include a reply length limit of 2048 tokens, top-k=1, and temperature=0.1. Each UAV within the formation interacts through a separate chat box, numbered consistently with the aircraft identifier.

5.2 Experiment Result

This paper aims to validate the functionality of the framework for coordinated manned-unmanned aerial vehicle combat using natural language commands. It assesses the effectiveness of a system architecture and natural language command processing mechanism based on LLM. The Red Team formation consists of one manned aircraft and two UAVs. The manned aircraft conducts planning and control through command-line interaction, while UAVs are controlled via natural language commands inputted through command-line interfaces. This setup simulates the process where manned aircraft issue natural language task instructions to UAVs during coordinated combat scenarios. Initial positions and mission setups for both teams are detailed in Section 4.1.

At the start of the combat simulation, commands [2,0,1] are utilized via command-line interaction to instruct the Red Team's manned aircraft to activate radar and accelerate forward to conduct a flat search for targets. Additionally, natural language commands are set for "Aircraft 1 to search target at -20 degrees direction" and "Aircraft 2 to search target at 20 degrees direction" to control UAV 1 and UAV 2, respectively, to search targets 20 degrees to the left and right of the Red manned aircraft. This maneuver aims to create separation and cover spatial areas of the battlefield. The combat simulation scene at this stage is depicted in Figure 8, where the fan-shaped areas represent the effective detection zones of aircraft radars for targets with a radar cross-section (RCS) of 1 square meter. Due to the initial distances exceeding the effective detection range of both sides' aircraft, neither the Red nor Blue Team can detect each other.

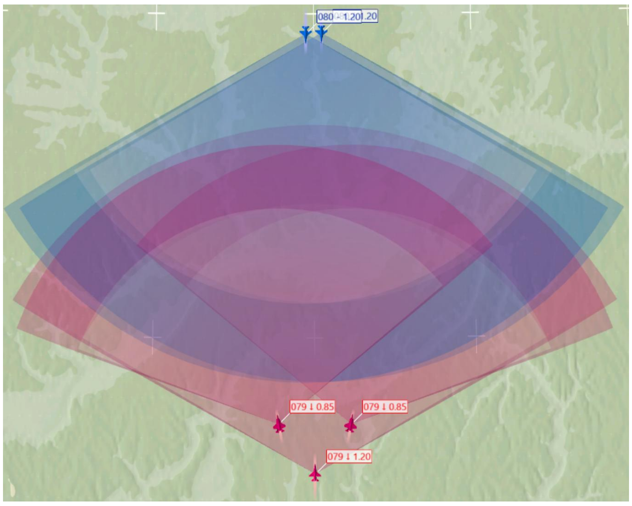

Figure 8 – Red UAVs were executing target search policy

As illustrated in Figure 9, both Red and Blue aircraft enter each other's radar detection range. The Red Team's manned aircraft detects two Blue targets. However, due to the low RCS of UAVs 1 and 2 (0.01 forward RCS), the Blue Team only detects the manned aircraft and not the two UAVs. At this stage, with a considerable distance between them, launching missiles poses a risk of evasion by the opposing side. Using command-line interaction with commands [2,0,1], the Red Team rapidly approaches the targets. Simultaneously, natural language commands are issued for "Aircraft 1 to launch missile at Target 1" and "Aircraft 2 to launch missile at Target 2". This directs Red UAVs to attack Blue targets and establish suppressive fire.

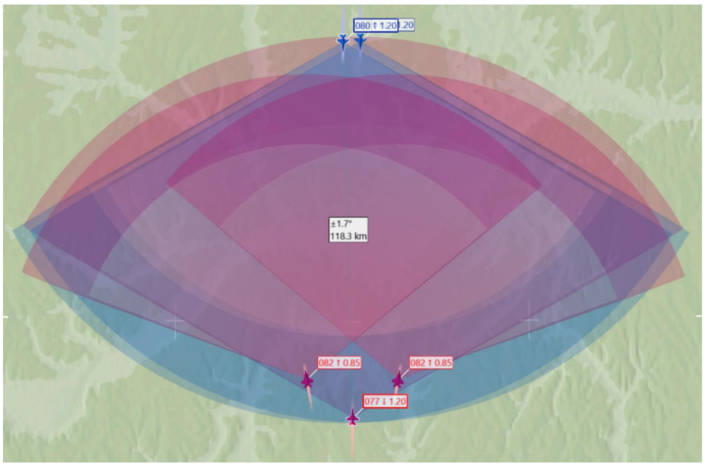

Figure 9 – Red manned aircraft detected the target

As shown in Figure 10, approximately 85 km from the targets, commands [2,1,1] are set via command-line interaction to instruct the Red manned aircraft to launch a missile at Blue Target 1. After the successful launch of this missile, commands [2,2,1] are issued to launch a missile at Blue Target 2. Upon successful execution of both missile launches, commands [4,0,1] initiate a dive to approximately 5000 meters for guidance. Before issuing weapon launch commands, authorization from the manned aircraft is sought. Authorization is granted for UAVs 1 and 2 to launch missiles at Blue Targets 1 and 2, respectively.

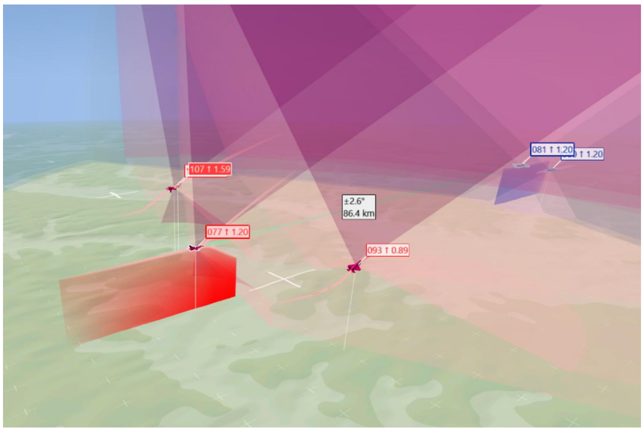

Figure 10 – Red UAVs were executing missile launch policy

Following the successful missile launches by Red UAVs, their policy execution regarding missile launch tasks concludes. The Red manned aircraft does not assign further tasks to the two UAVs, who proceed with default tasks of "missile guidance to target *i*", where *i* corresponds to the previously designated missile launch targets. As shown in Figure 11, the two UAVs commence guidance tasks and initiate descent.

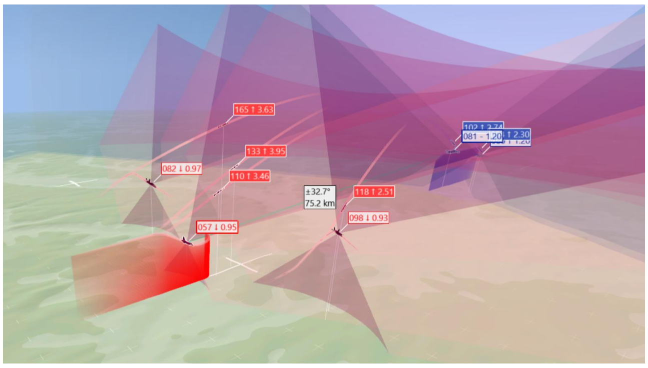

Figure 11 – Red UAVs were executing missile guidance policy

At approximately 60 km from the targets, command [7,0,1] directs the Red manned aircraft to maneuver away from the targets and turn in the opposite direction to evade potential incoming missiles from Blue. This maneuver aims to enhance aircraft survivability. To maintain threat and lethality against Blue targets, the manned aircraft transfers control of the missiles it has launched and guided to the UAVs before veering away from the targets. Natural language commands are issued for "Aircraft 1 to take over missile launched at Target 1 and guide it" and "Aircraft 2 to take over missile launched at Target 2 and guide it". This transfers guidance authority for the missiles previously launched at two Blue targets by the Red manned aircraft to the two UAVs, enabling the manned aircraft to turn away from the targets. As shown in Figure 12, the two UAVs of the Red Team execute relay guidance policies,.

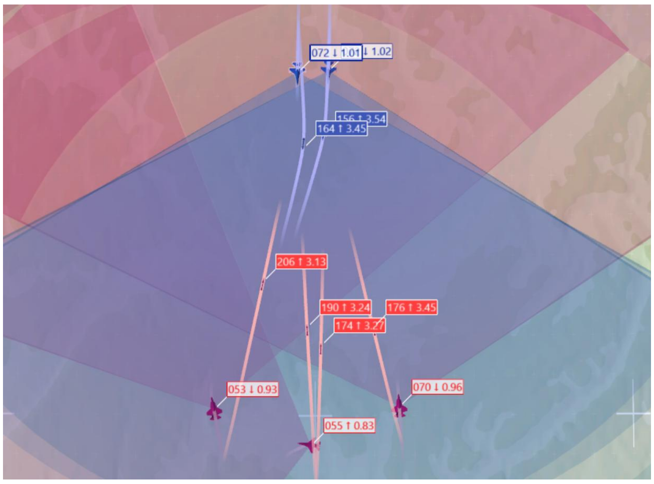

Figure 12 – Red UAVs were executing relay guidance policy

After turning away from the targets, the Red manned aircraft accelerates to exit the high-risk area. The relay guidance policies executed by Red UAVs lead to successful proximity to their respective targets, as shown in Figure 13. Onboard radar initiates search for Blue targets, and terminal guidance is conducted after intercepting Blue targets. The relay guidance policies for Red UAVs conclude.

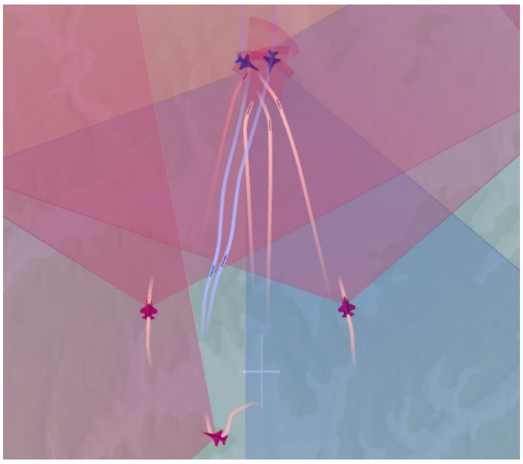

Figure 13 – Completion of the relay guidance policy execution by the red UAVs

Both Blue aircraft are successfully hit by Red missiles, as shown in Figure 14. Natural language commands are reissued for "Aircraft 1 to search target at -20 degrees direction" and "Aircraft 2 to search target at 20 degrees direction" to instruct UAVs 1 and 2 to continue searching for potential Blue targets. Red UAVs ascend to 8000 meters to continue their target search tasks.

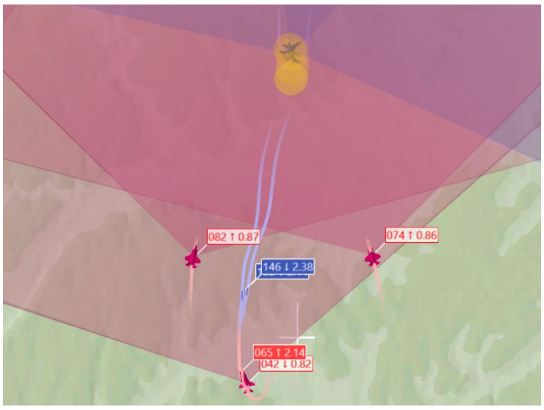

Figure 14 – Red Team UAVs executed target search policy after all Blue targets disappeared,

Throughout the experiment, various components and modules of the proposed manned-unmanned aerial vehicle collaborative combat framework were tested. All components and modules demonstrated the capability to fulfill their intended functionalities. The UAV large language model task manager responded promptly, stably, and reliably to natural language task commands issued by manned aircraft. The system effectively matched natural language task commands with appropriate executable policies, ensuring smooth and logical transitions between strategy sub-tasks. Timely and effective feedback on UAV task status was provided. The designed UAV mission executor operated normally, successfully controlling UAVs to execute mission commands.

6. Conclusion

- This paper presents a framework for manned-unmanned aerial vehicle collaborative combat based on large language models. The framework includes a UAV LLM Missile Manager and UAV Mission Executor, achieving an interaction mechanism for low operational and decision-making burdens in manned-unmanned aerial vehicle collaborative combat. This framework allows manned aircraft pilots to issue mission commands to UAVs using natural language, within the constraints of assigned aircraft mission permissions. By designing a Natural Language Regularization Module to construct regularization questions, uncertainties and irregularities in LLM responses are effectively reduced. This enhances the framework's human-machine interaction, credibility, and reliability. Additionally, a simulation environment for manned-unmanned aerial vehicle collaborative combat simulator was designed and constructed to experimentally validate the proposed framework in a typical missions;
- Designed is a UAV Mission Executor that includes Preloaded Policy Library and other components, providing policy information to the LLM mission manager and executing actions selected by the task manager's policies. The Action Executor performs actions outputted by policies to control UAVs in executing mission commands. During this process, pilots only need to supervise the UAV's mission execution status, without the necessity for direct real-time control of UAVs;
- Developed is a critical collaborative combat status feedback mechanism that allows human pilots to intervene and control various decision-making levels of UAVs to mitigate potential ethical and legal risks. During mission execution, UAVs provide feedback to manned aircraft on mission execution status, and high-risk actions such as weapon release require authorization from manned aircraft before execution.

7. Acknowledgement

This work was co-supported by the National Natural Science Foundation of China (No. 52272382), the Aeronautical Science Foundation of China (No. 20200017051001), and the Fundamental Research Funds for the Central Universities, China.

8. Contact Author Email Address

For any questions please contact:

zhouyaoming@buaa.edu.cn

9. Copyright Statement

References

[1] Team, Enterprise Capability Collaboration. *Air superiority 2030 flight plan*. US Air Force, 2016.

[2] Francis, Michael. The Role of Unmanned Air Vehicles in Advancing System-of-Systems (SoS) Technologies and Capabilities. *In AIAA International Air and Space Symposium and Exposition: The Next 100 Ye*ars, AIAA, 2003.

[3] Yang Wei. Development of future fighters. *Acta Aeronautica et Astronautica Sinica*, 41(6): 524377, 2020.

[4] Sun Cong. Development trend of future fighter: A review of evolution of winning mechanism in air combat. *Acta Aeronautica et Astronautica Sinica*, 42(8): 525826, 2021.

[5] Gangl, Stefan, Benjamin Lettl, and Axel Schulte. Management of multiple unmanned combat aerial vehicles from a single-seat fighter cockpit in manned-unmanned fighter missions. *AIAA Infotech@ Aerospace (I@ A) Conference*, Boston, 2013.

[6] Johnson M. *Coactive design: designing support for interdependence in human-robot teamwork*. Delft University of Technology, 2014.

[7] Wang Chang, Wu Lizhen, Yan Chao, Wang Zhichao, Long Han, Yu Chao. Coactive design of explainable agent-based task planning and deep reinforcement learning for human-UAVs teamwork. *Chinese Journal of Aeronautics,* Vol. 33, No. 11, pp 2930-2945, 2020.

[8] Bradshaw JM, Feltovich PJ, Jung H, et al. *Dimensions of adjustable autonomy and mixed-initiative interaction*. Springer, 2004.

[9] Djerk G. *ePartners for dynamic task allocation and coordination*. Delft University of Technology, 2012.

[10]Zhao Z, Wang C, Niu Y, et al. Adjustable autonomy for humanUAVs collaborative searching using fuzzy cognitive maps. *In 2019 2nd China symposium on cognitive computing and hybrid intelligence*, IEEE, 2019.

[11]Fong TW. *Collaborative Control: A Robot-centric model for vehicle teleoperation*. Carnegie Mellon University, 2001

[12]Wang Haifeng. Discussions on the development of high performance collaborative combat un-manned aerial vehicle. *Acta Aeronautica et Astronautica Sinica*, 45(20):530304, 2024.

[13]Achiam, Josh, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida et al. *Gpt-4 technical report*. arXiv preprint arXiv:2303.08774, 2023.

[14]Cobbe, Karl, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert et al. *Training verifiers to solve math word problems*. arXiv preprint arXiv:2110.14168, 2021.

[15]Li Jia, Ge Li, Chongyang Tao, Huangzhao Zhang, Fang Liu, and Zhi Jin. *Large Language Model-Aware In-Context Learning for Code Generation*. arXiv preprint arXiv:2310.09748, 2023.

[16]Weng Lilian. Llm-powered autonomous agents. lilianweng.github.io, 2023.

[17]Berndt, Jon. JSBSim: An open source flight dynamics model in C++. *In AIAA Modeling and Simulation Technologies Conference and Exhibit*, AIAA. 2004.

[18]Yang Xuerong, Li Xiaodong, Liang Jiahong. Study of a parabola-trajectory compound guidance law for air-to-air missile. *Systems Engineering and Electronics*, Vol. 31, No. 12, pp 2918-2923, 2009.

[19]Ji Jinzu, Huang Peilin, Ma Yunpeng, Zhang Junsheng. *Stealth Principle*. Beihang University Presss, 2018.