A Novel LLM-Based AI Agent for Autonomous Drone Swarm Control Through Natural Language Prompts

Khalid Mohamed
Department of Informatics
Osmania University
Email: mohamedkhalid149@hotmail.com

***Abstract*—This paper introduces a novel approach to autonomous drone swarm control using a Large Language Model (LLM)-based AI agent that interprets and executes user objectives conveyed through natural language prompts. The AI agent autonomously coordinates multiple drones, translating human commands into precise drone actions, enabling complex missions such as surveillance, multi-waypoint navigation, and autonomous return. By leveraging the inherent scalability and adaptability of the system, the agent dynamically manages real-time decision-making and control of the drone swarm, ensuring seamless task execution. The communication framework integrates a robust API for reliable control, allowing the agent to interpret, plan, and carry out missions with minimal human intervention. Performance is evaluated based on task completion accuracy, where successful mission outcomes are defined by the fulfillment of the user's objectives. Experimental results show that the system achieves high levels of task accuracy, demonstrating its potential for applications in disaster response, aerial monitoring, and automated multi-drone missions.**

***Index Terms*—LLM, AI Agents, drone, smarm robotics, autonomous systems, natural language interface, task completion accuracy**

I. Introduction

The growing need for autonomous drone swarms across industries such as disaster response, surveillance, and agriculture is evident due to their ability to perform complex tasks efficiently and without direct human intervention. In disaster response, for example, UAV swarms can be deployed for search-and-rescue operations, offering real-time data from remote areas and helping emergency teams make faster decisions [1]. In agriculture, UAV swarms can monitor large fields, automate crop inspections, and optimize resource usage. Similarly, surveillance missions benefit from UAV swarms' ability to cover large areas simultaneously, maintaining continuous situational awareness [2].

Managing UAV swarms, however, presents several challenges. Real-time decision-making becomes more complex as the number of UAVs increases. Effective coordination is required to avoid UAV collisions, ensure coverage, and respond to dynamic environmental changes [3]. UAV swarms must also efficiently allocate tasks and share resources, ensuring that each drone executes its tasks in sync with others. As noted by [4], UAVs need to manage both global and local objectives—maintaining formation and addressing specific tasks simultaneously. The need for dynamic task reallocation and decentralized coordination further complicates the challenge, as UAVs must adapt to new circumstances, such as obstacles or resource limitations, while working cohesively as a swarm [5].

Artificial Intelligence (AI) plays a critical role in enhancing the capabilities of UAV swarms by enabling real-time decision-making, coordination, and collaboration in multi-agent environments. Swarm robotics, in particular, leverages AI techniques like reinforcement learning and multi-agent collaboration to improve task execution efficiency. For instance, [3] discusses a task and resource dynamic assignment algorithm that enables UAVs to adaptively assign tasks, ensuring real-time coordination and optimal task allocation. Similarly, evolutionary computation combined with swarm intelligence helps UAV swarms maintain communication and coordinate complex tasks, even in uncertain environments [6]. These AI-driven frameworks allow UAV swarms to make independent decisions while staying aligned with the mission's global objectives.

Recently, the integration of Large Language Models (LLMs) has introduced new possibilities for controlling UAV swarms through natural language commands. LLMs, such as those utilized in the cloud-edge-end architectures, have improved multi-agent strategy generation and task control by mapping natural language commands to complex robot sequences, thereby bridging the gap between human intent and machine execution [7]. LLMs also enable real-time task allocation and adaptive behavior in multi-agent environments by processing human directives and converting them into executable tasks for UAVs [8]. While LLMs face challenges in multi-agent coordination, particularly in tasks like flocking [9], their integration into swarm robotics is a significant step toward making UAV systems more intuitive and responsive to human inputs.

In current UAV control systems, several limitations hinder their effectiveness in dynamic environments. Traditional methods rely heavily on manual control, where human operators must continuously guide drones, limiting scalability for large swarms and complex missions. This approach can be labor-intensive, prone to human error, and unsuitable for critical time-sensitive scenarios such as disaster response, where real-time decision-making is essential. Research demonstrates that manual control methods limit task execution flexibility and scalability, especially when using traditional controllers that require constant human supervision [10]. Manual operation constraints also lead to challenges in adapting to dynamic environments where drones must respond to real-time events [11].

To address these challenges, there is a growing demand for more autonomous systems capable of real-time task generation. Studies highlight the need for modular and scalable data acquisition systems that enable drones to perform tasks independently while adapting to unpredictable conditions [12]. The implementation of scalable frameworks can significantly enhance real-time coordination and execution, as demonstrated by methods that utilize UAV clusters for real-time reconstruction and task management [4]. By transitioning from manual controls to AI-based autonomous systems, UAVs can better adapt to real-world scenarios and improve mission efficiency.

This paper introduces a novel LLM-based AI agent designed to control a drone swarm through natural language prompts. By leveraging the capabilities of Large Language Models (LLMs), the proposed system bridges the gap between human intent and autonomous drone operations. The agent interprets complex user commands and translates them into actionable tasks for the drone swarm, allowing for real-time task allocation and execution.

Key features of the system include its ability to dynamically allocate tasks across the swarm in real-time, ensuring efficient coordination between drones. The agent operates autonomously, making decisions based on both the user's input and environmental conditions, which enhances the overall flexibility and scalability of the system. Additionally, the system boasts high task accuracy, meaning that objectives specified in the natural language prompts are fulfilled reliably. This approach eliminates the need for manual drone control, enabling more intuitive and scalable operation of drone swarms across a variety of complex missions.

The rest of the paper is organized as follows: Section II provides a review of related work, focusing on advancements in drone swarm control, AI-based autonomous systems, and the role of natural language processing in robotic systems. Section III presents the system architecture of our LLM-based AI agent for drone swarm control, detailing its components and how it processes natural language commands into executable tasks. Section IV outlines the evaluation methodology used to assess the performance and accuracy of the proposed system in various scenarios. Section V presents the results of our experiments, highlighting the effectiveness of the system in real-time task allocation and autonomous decision-making. Section VI presents challenges and future directions for this work, while Section VII concludes the paper.

II. Related Work

Existing approaches for controlling drone swarms range from manual, semi-autonomous, to fully autonomous systems. Manual control involves direct human oversight, where operators issue commands for specific tasks. Although this approach ensures accuracy and direct control, it is not scalable when dealing with large swarms due to the cognitive load on operators [13]. On the other hand, semi-autonomous systems allow operators to supervise UAVs while the drones carry out pre-programmed tasks, but this approach still lacks flexibility and adaptability to dynamic environments [14].Manual control of large-scale UAV swarms poses significant challenges due to the high cognitive load placed on human operators, particularly when managing large numbers of drones. More intuitive human interfaces are needed to effectively handle the complexity of data generated by the swarm, reducing the operational burden on operators [15].

Autonomous systems, on the other hand, introduce centralized and decentralized control methodologies for efficient swarm coordination. Centralized control models rely on a central node to allocate tasks and synchronize actions, as demonstrated by [16], where a centralized task assignment approach enhances search efficiency during rescue missions. However, decentralized control has emerged as a preferred solution due to its ability to improve scalability and reduce system vulnerabilities. UAVs under decentralized control can make local decisions and communicate through mesh networks or peer-to-peer topologies, increasing system resilience [17]. Decentralized models enable swarms to operate autonomously without constant input from a central controller, as highlighted by [18], where AI-driven decision-making fosters collaborative autonomy in UAV swarms. In contrast to centralized approaches, decentralized strategies better support complex, real-time tasks in dynamic environments.

AI has become integral to enhancing the control of UAV swarms, particularly through the use of reinforcement learning (RL) and multi-agent systems. Traditional machine learning approaches initially focused on rule-based systems and supervised learning for drone path planning and obstacle avoidance. However, more recent advancements have introduced reinforcement learning and multi-agent systems to enable drones to make independent decisions in real-time. For example, multi-agent reinforcement learning (MARL) has been applied to improve the coordination and fault tolerance of UAV swarms in dynamic environments, allowing drones to adjust their behavior based on their individual health status or environmental changes [19]. These systems also optimize energy use and maximize communication efficiency, making them more suitable for complex missions [20].

Reinforcement learning techniques, such as Proximal Policy Optimization (PPO) and Deep Deterministic Policy Gradient (DDPG), have been applied to real-time UAV swarm coordination, significantly improving UAVs' ability to maintain multi-hop connectivity and autonomous navigation in uncertain environments. Studies like [21] explore cooperative multi-agent reinforcement learning, which enables drones to collaborate and coordinate effectively in tasks like search-and-rescue and surveillance. Moreover, federated reinforcement learning allows UAV swarms to operate autonomously without continuous communication with a central controller, reducing communication delays and improving scalability [22]. Additionally, AI-driven approaches like topology control optimize the communication network and improve overall mission success by enhancing the UAV swarm's ability to adapt to dynamically changing conditions [23].

AI has also been integrated into UAV swarm systems through advanced machine learning techniques like neural networks and bio-inspired optimization algorithms. These AI approaches have enhanced the decision-making capabilities of UAV swarms, enabling them to adapt to dynamic environments. For example, bio-inspired algorithms such as the BOLD (Bio-Inspired Optimized Leader Election) algorithm have been implemented to improve decision-making in cluster head elections within UAV swarms, resulting in a 15% increase in network lifetime compared to traditional particle swarm optimization (PSO) methods [24]. Similarly, the use of edge machine learning enables local decision-making for real-time coordination and visual computing, allowing UAVs to operate autonomously without the need for continuous centralized control [25].

Additionally, extreme learning machines and multi-agent deep reinforcement learning (DRL) have been used for real-time trajectory control and situation awareness within UAV swarms. These AI-driven techniques facilitate energy-efficient decision-making and improve overall coordination in complex, data-rich environments like search-and-rescue operations and surveillance [26]. This integration of machine learning in UAV systems has paved the way for UAVs to make informed decisions autonomously, ensuring robust coordination even under challenging conditions.

Large Language Models (LLMs) have recently made notable strides in robotics, particularly by introducing natural language interfaces for autonomous systems. These models significantly reduce the gap between human intent and machine execution by allowing users to communicate with robots using everyday language, eliminating the need for specialized technical knowledge. For instance, the InCoRo framework integrates LLM controllers into classical robotic feedback loops, enabling zero-shot generalization and real-time adaptation to dynamic environments [27].Similarly, DriveGPT4, a multimodal LLM-based system, has shown advancements in end-to-end autonomous driving by processing video inputs and providing reasoning capabilities for vehicle control signals [28]

However, most of these advancements have focused on individual robotic systems and have not yet been extended to more complex, multi-agent environments like drone swarms. While LLMs have been applied to autonomous systems for task assignment [29] and autonomous driving systems through multimodal capabilities [28], they lack the level of dynamic coordination required for real-time, multi-agent control. Current challenges, such as LLMs' limitations in processing spatial relationships for collaborative tasks like multi-agent flocking, further highlight the need for innovation [9].

Our research addresses these limitations by using LLMs to develop AI agents capable of controlling drone swarms through natural language prompts. This approach significantly advances the state-of-the-art by enabling real-time task allocation, decision-making, and coordination across multiple agents, overcoming the limitations of prior systems. By extending LLM capabilities to multi-agent operations, our work introduces a new level of flexibility and scalability in swarm control, making it a major step forward in applying AI to real-world drone applications.

III. System Architecture and Design

This research introduces a novel architecture for controlling a swarm of simulated drones using an LLM-based AI agent capable of interpreting user commands and executing complex drone missions autonomously. The system is entirely simulation-based, with each drone represented by an individual instance running *ArduPilot* firmware. The architecture is designed to ensure seamless interaction between the user, the AI agent, and the drone swarm, with real-time monitoring capabilities provided by *MavProxy*, a ground control station (GCS). The overall system architecture is depicted in Figure 1, showcasing the AI agent for Drone Swarm.

A. LLM-Based AI Agent for Drone Control

At the core of the system is an AI agent capable of interpreting high-level, natural language commands provided by the user and autonomously translating them into a series of drone actions. By processing user commands, the AI agent defines mission parameters, such as specifying waypoints, initiating takeoff, or commanding drones to return to base. The AI agent operates by leveraging a sequence of predefined tools that enable drone control actions and maintains internal states to manage ongoing conversations and mission contexts.

1) Thought-Action-Observation Cycle: The AI agent operates using a thought-action-observation cycle, which enables it to:

- **Thought:** The AI agent processes the user's input to determine appropriate drone operations, such as launching a surveillance mission. During this stage, the agent utilizes its **system prompt**—a predefined set of instructions that guide its interpretation of user commands and ensure its responses are consistent with the mission goals. It also refers to the **chat history**, which stores previous user interactions, to provide continuity and context when generating commands.
- **Action:** Based on the understanding of the task, the AI agent initiates the drone operations by issuing commands like takeoff, navigating to waypoints, or switching flight modes (e.g., GUIDED or AUTO). The agent leverages predefined **tools**, which encapsulate specific drone control functionalities such as arm, disarm, takeoff, and go-to-location. These tools interact with the DroneAPI, enabling the agent to execute drone operations autonomously.
- **Observation:** While the agent does not provide real-time feedback directly, mission progress is monitored through telemetry data supplied by *MavProxy*. This data helps validate task completion and allows the system to log

mission success or failure. The agent maintains an **agent scratchpad**, a temporary memory space where it records intermediate steps and decisions during its thought-action cycle. This helps the AI agent track the execution of commands and refine subsequent actions if necessary.

2) Memory and Tools for Contextual Control: The AI agent utilizes multiple forms of memory to handle context and mission continuity effectively. The **chat history** stores past interactions between the user and the AI, enabling it to maintain contextual understanding over long interactions. This allows the agent to build on previous commands, ensuring that multi-step missions are executed seamlessly.

The **system prompt** establishes the AI's behavior and ensures it interprets user commands in alignment with predefined drone control objectives. By maintaining a coherent system prompt, the AI agent ensures consistent interpretation of user instructions.

Additionally, the **agent scratchpad** records intermediate steps during task execution, helping the agent monitor the progression of actions. For instance, when switching a drone's mode or navigating to a waypoint, the scratchpad stores the sequence of commands issued, providing a clear execution path that can be referenced throughout the mission.

Finally, **tools** serve as specialized functions that enable the AI agent to directly control the drone. These tools allow the agent to interact with the drones by arming them, controlling their navigation, managing flight modes, and more. The tools are the operational extensions of the AI agent's decisions, enabling it to execute actions in the physical (or simulated) environment.

The AI agent's ability to translate user objectives into structured drone commands, along with its decision-making capabilities, enables it to manage and control multiple drones, simulating real-world mission dynamics.

B. Simulated Drone Swarm Using ArduPilot Firmware

Each drone in the system operates as a simulated instance running *ArduPilot*, an open-source firmware widely used in UAV applications. *ArduPilot* provides a realistic simulation for tasks like waypoint navigation, takeoff, and landing, ensuring that the drone behaves similarly to how it would in real-world operations.

The *ArduPilot* firmware offers several key features:

- **Flight Mode Management:** The system supports various flight modes such as AUTO (for autonomous missions) and GUIDED (for real-time control). The AI agent issues commands to switch flight modes during different mission phases.
- **Safety Mechanisms:** The system ensures that all safety checks, such as pre-arm validations and mission readiness assessments, are performed before executing any operations.

C. MavProxy for Drone Communication and Monitoring

MavProxy serves as the ground control station (GCS) for the drone swarm and facilitates communication between the AI agent and the drones using the *MavLink* protocol. It handles telemetry feedback, allowing for real-time monitoring of the drone swarm during missions.

MavProxy plays a crucial role by:

- **Telemetry Feedback:** The system receives updates on drone positions, flight modes, and other parameters, which are essential for monitoring mission progress.
- **Command and Control:** High-level commands issued by the AI agent, such as mode switching or navigation, are transmitted to the drones through *MavProxy*.

D. Drone Control API

A *Drone Control API* acts as an intermediary between the AI agent and the drone simulations. The API abstracts the complexity of drone communication, providing high-level functions for managing the drone swarm. These include:

- **Mission Uploading and Execution:** The API allows the AI agent to upload waypoint missions and commands the drones to execute them when switched to AUTO mode.
- **Flight Mode Control:** The API enables the AI agent to switch drone flight modes as required by the mission.
- **Status Retrieval:** The API provides real-time access to each drone's status, such as location, altitude, and battery level.

E. Mission Management and Drone Swarm Coordination

The AI agent manages multi-drone operations by dividing tasks across the swarm. Each drone can be assigned specific waypoints, enabling coordinated missions such as wide-area surveillance. The system supports:

- **Multi-Drone Tasking:** The AI agent autonomously divides tasks among the swarm, ensuring optimal coverage of the mission area.
- **Autonomous Mission Execution:** Once the mission is uploaded, the drones operate independently, with only minimal intervention from the AI agent.

F. Drone Status Monitoring

To ensure that drones are functioning properly and ready for missions, the system includes a real-time status monitoring tool. The "get drone status" tool is integrated into the system, providing detailed feedback on the operational state of each drone. The tool retrieves data such as latitude, longitude, altitude, battery level, armed status, ground speed, and current flight mode. A sample response for the "get drone status" tool is as follows:

A sample drone status is as follows:

- Latitude: -35.362048
- Longitude: 149.164489
- Altitude: 30 meters
- Battery Level: 85%
- Armable: Yes
- Armed: Yes
- Ground Speed: 5.2 m/s
- Flight Mode: GUIDED

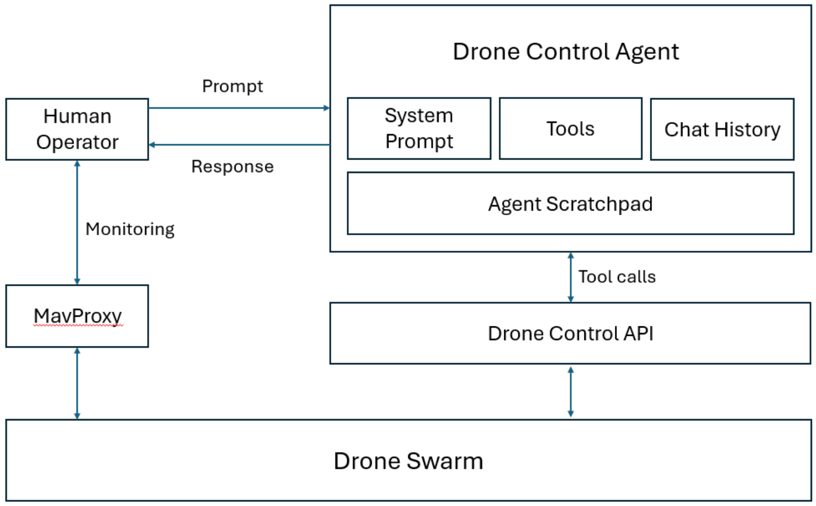

Fig. 1. Swarm Control System Architecture

The drone is flying at a stable altitude of 30 meters, is fully operational, and armed in GUIDED mode. The current mission is ongoing, and the drone is ready to accept further commands.

G. Simulation Environment

The system operates entirely in a simulated environment, where each drone is represented by an individual *ArduPilot* simulation instance. This approach offers the following advantages:

- **Risk-Free Testing:** The simulation environment eliminates the risks associated with real-world drone operations, such as hardware failure or accidents.
- **Scalability:** Multiple drones can be simulated without the need for physical hardware, enabling extensive testing of swarm control scenarios.
- **Cost-Effective Development:** The simulation-based approach reduces the need for expensive UAV hardware, making the system more accessible for development and experimentation.

This comprehensive architecture demonstrates a flexible and scalable method for managing drone swarms through AI-driven autonomous operations. The integration of *ArduPilot*, *MavProxy*, and a custom-built *Drone Control API* provides a robust framework for simulating real-world UAV swarm control scenarios.

H. Operational Workflow

The overall operational workflow involves the user providing high-level tasks through natural language prompts. The AI agent interprets these tasks and breaks them down into actionable commands for the drones. For example, a user command like ”patrol the perimeter” is processed by the AI agent, which identifies key actions such as takeoff, navigation to specific points around the perimeter, and landing. These actions are then executed by the drones, with the mission progress observed through MAVProxy. The flexibility of this architecture allows the AI agent to handle both simple and complex drone operations autonomously.

I. System Limitations

While the system successfully manages multiple drones and executes missions based on user input, it is important to note that the current implementation is entirely within a simulated environment. Real-time mission feedback is limited to what MAVProxy displays, and the AI agent does not currently handle dynamic changes or contingencies during mission execution. Future iterations of this system could integrate real-time feedback loops to make the AI agent more reactive to in-flight events.

IV. Evaluation Methodology

The evaluation of the AI agent’s ability to control a drone swarm was conducted with three different swarm sizes: 5 drones, 15 drones, and 30 drones. For each swarm size, the agent was tasked with sequentially executing 100 user prompts, one at a time. After each prompt was processed, the results were manually verified to assess whether the AI agent accurately executed the task and produced the expected outcome. This verification process ensured that any deviation from the expected behavior was captured and logged. If the task was executed correctly, the result was marked as “passed”; if the outcome differed from expectations, the result was marked as “failed.”

A. Evaluation Setup

Each evaluation followed this process:

1) **Sequential Prompt Execution:** A series of 100 user prompts was provided to the AI agent, one after the other. This allowed the AI agent to retain and use chat history as it processed each subsequent prompt. This structure was intentional to evaluate how well the agent leveraged its memory and maintained context over time.

2) **Swarm Sizes:** The evaluation was performed across three different swarm sizes to test the scalability of the AI agent:
   - **5 drones:** A small swarm setup to assess the basic functionality and control capabilities of the AI agent.
   - **15 drones:** A medium-sized swarm to test the system's ability to manage more complex operations and handle greater amounts of information.
   - **30 drones:** A large-scale swarm used to push the boundaries of the system's scalability and performance under higher demand.
3) **Manual Verification:** After each prompt, the mission execution was verified by manually observing the drone's performance through telemetry data provided by the simulation (via MAVProxy). For each prompt, the result was classified as either "passed" or "failed," based on whether the expected behavior was achieved.
4) **Effect of Chat History:** The sequential nature of the prompt evaluation allowed the AI agent to maintain context from previous commands. This enabled us to measure whether the agent's performance improved or deteriorated over time, particularly in scenarios where complex tasks were dependent on earlier commands.

B. Sample Queries for Evaluation

Table 1 shows different sample user prompts, the corresponding Agent responses, the results of those prompts, and whether the task was executed correctly or not.

The table highlights complex, context-dependent queries that the AI agent must accurately interpret and execute. These queries test the agent's ability to manage multiple drones, understand group dynamics, and carry out multi-step tasks while adapting to the swarm's changing state.

C. Performance Metrics

The effectiveness of the AI agent was measured using two key performance metrics:

Task Success Rate is defined as the proportion of tasks successfully completed according to the user's request. Each task was manually verified, and if the result matched the expected behavior, it was marked as successful. The formula is:

$$\text{Task Success Rate} = \frac{\text{Number of Successful Tasks}}{\text{Total Number of Tasks}} \times 100$$

For example, if out of 100 tasks, 92 were executed correctly, the Task Success Rate would be 92%.

Error Rate is the proportion of tasks where the AI agent failed to interpret or execute the user prompt accurately. Failures may stem from misinterpretation of the command or errors in task execution. The formula is:

$$\text{Error Rate} = \frac{\text{Number of Failed Tasks}}{\text{Total Number of Tasks}} \times 100$$

For instance, if 8 tasks out of 100 failed due to errors, the Error Rate would be 8%.

D. Scalability and Extensibility

The evaluation also assessed the system's scalability as the number of drones increased. The Task Success Rate and Error Rate were recorded for swarm sizes of 5, 15, and 30 drones. These results indicated the system's scalability in handling more drones without significant performance degradation.

The system is designed for extensibility, allowing for future upgrades. This could include adding more sophisticated AI agents, expanding drone operations, or enabling multi-swarm coordination. The core system remains flexible enough to accommodate these enhancements while maintaining its prompt-based drone control capabilities.

V. Results and Discussion

The evaluation of the AI agent across different swarm sizes—5 drones, 15 drones, and 30 drones—showed outstanding performance, with a **Task Success Rate of 100%** and an **Error Rate of 0%** for each configuration.

A. Results Table

Table 2 presents the Task Success Rate and Error Rate for drone swarms of varying sizes, demonstrating the system's performance across different configurations.

B. Discussion

The AI agent's performance highlights several critical aspects of its scalability and reliability:

1) Scalability and Performance Consistency: The agent's ability to handle drone swarms of varying sizes without any performance degradation is a key takeaway from the evaluation. Even with the complexity of managing 30 drones, the system maintained a **100% success rate**. This demonstrates that the AI agent can effectively manage increasing complexity and workload as the number of drones in the swarm grows, making it suitable for larger-scale applications.

2) Chat History and Context Management: By retaining and utilizing chat history, the AI agent could handle sequential tasks with dependencies, ensuring the correct execution of each task even as the series of prompts progressed. The flawless completion of 100 tasks in each swarm size setup suggests that the agent effectively leverages its memory and maintains context, allowing it to adapt and perform long-term tasks with multi-step dependencies.

3) Absence of Errors: The system's **0% error rate** across all three configurations reflects its reliability in understanding and executing user commands. Each task, whether simple or complex, was executed without deviation from the expected outcome. This shows that the integration between the AI agent and the underlying simulation, including MAVProxy for telemetry, is both stable and efficient, allowing for precise mission execution.

4) Impact of Task Complexity: The evaluation included a wide variety of tasks, ranging from simple commands like "arm and takeoff" to more complex ones involving group coordination and conditional decisions based on drone status, such as battery percentage or proximity. Despite the variety,

Query	Agent Response	Result	Comments
Arm and takeoff all drones of the swarm.	All drones are being armed and will take off to the specified altitude.	The entire swarm successfully arms and takes off, hovering at the designated altitude.	Task executed correctly.
Send the drone with the highest battery percentage to scout the warehouse and come back to its current location. Do this in AUTO mode.	The drone with the highest battery percentage is being sent to scout the warehouse. It will return to its original position after the mission.	The drone with the highest battery flies to the warehouse, scouts the area, and returns to its original location.	Task executed correctly.
Switch all drones of the swarm to guided mode.	Switching all drones to GUIDED mode.	All drones successfully switch to GUIDED mode, ready to receive real-time commands.	Task executed correctly.
Let's say the base is at the center of an imaginary square of length 100m. I want a drone at each corner of this square.	Four drones are being deployed to the corners of the square centered at the base.	Four drones navigate to the respective corners of the 100m square and hover at the specified coordinates.	Task executed correctly.
Send the nearest drone to outpost 1 to scout outposts 1, 2, and 3. Send the nearest drone to outpost 4 to scout outposts 4, 5, and 6. Do this in AUTO mode.	Sending the nearest drones to their assigned outposts. One will scout outposts 1, 2, and 3, and the other will scout outposts 4, 5, and 6.	Two drones are dispatched. The nearest drone to outposts 1–3 scouts those outposts, and the other drone scouts outposts 4–6 as requested.	Task executed correctly.
Consider drones 1 and 2 as group 1. Consider drones 3 and 4 as group 2.	Drones 1 and 2 are assigned to Group 1. Drones 3 and 4 are assigned to Group 2.	Drones are grouped successfully as per the instructions.	Task executed correctly.
Send group 1 to outpost 1, then send group 2 to the warehouse.	Group 1 is being sent to outpost 1, and Group 2 will proceed to the warehouse.	Group 1 navigates to outpost 1, and Group 2 proceeds to the warehouse as instructed.	Task executed correctly.

TABLE I
SAMPLE QUERIES AND RESPONSES

Swarm Size	Task Success Rate (%)	Error Rate (%)
5 drones	100	0
15 drones	100	0
30 drones	100	0

TABLE II
TASK SUCCESS RATE AND ERROR RATE FOR DIFFERENT SWARM SIZES

the agent handled each task correctly, confirming its flexibility and capability to manage intricate, multi-drone operations. This suggests that the system is well-suited for tasks that require dynamic decision-making, such as allocating drones to specific waypoints based on real-time conditions.

5) Future Scalability: Although the evaluation tested swarm sizes up to **30 drones**, the system architecture appears capable of scaling even further with appropriate infrastructure. The seamless management of larger swarms opens up opportunities for its application in broader operations, such as search-and-rescue missions or disaster management, where a high number of drones may need to be coordinated simultaneously.

6) Potential Improvements and Limitations: While the current setup performed exceptionally well in a simulated environment, further testing with real-world conditions, such as obstacles, weather variability, or hardware limitations, could reveal areas for improvement. Introducing dynamic factors in future evaluations could help assess the system's fault tolerance and its capacity for real-time adjustments in more unpredictable scenarios.

C. Conclusion of Results

The results demonstrate that the AI agent is a reliable and scalable solution for drone swarm management in simulated environments. The system's flawless task execution across varying swarm sizes shows its potential for real-world applications, although future evaluations with dynamic, real-world conditions are recommended to assess its full capabilities.

VI. CHALLENGES AND FUTURE DIRECTIONS

While the AI agent-based drone swarm control system demonstrated high accuracy and reliability in executing tasks, several challenges were identified during the development and evaluation phases.

A. Scalability Limitations

Although the system performed well with up to 30 drones, scaling to larger swarms could introduce complexities in communication and coordination. The current architecture relies on sequential task execution and limited use of memory. In the future, more advanced communication protocols and distributed decision-making algorithms could be integrated to enhance scalability, allowing the system to manage hundreds of drones effectively.

B. Real-time Feedback

One of the major limitations is the lack of real-time feedback from the AI agent regarding the ongoing mission. Currently, the system depends on *MavProxy* for monitoring drone telemetry, while the AI agent itself does not actively monitor or adapt to the real-time state of the drones. Future iterations could integrate a more interactive agent that provides dynamic feedback during mission execution, potentially improving responsiveness and adaptability in complex scenarios.

C. Limited Error Handling

While the error rate was recorded as 0% in the current evaluation, this is partially due to the simplified simulation environment. In real-world applications, unforeseen errors such as hardware malfunctions, communication failures, or dynamic obstacles are likely to occur. Future work should focus on improving the agent's ability to detect, diagnose, and recover from such errors autonomously, without human intervention.

D. Contextual Understanding

The system performed well with sequential prompt execution, maintaining chat history and responding accurately to complex queries. However, as the complexity of user inputs increases, particularly with ambiguous or contextually rich instructions, the agent may struggle to correctly interpret user intentions. Enhancing the agent's natural language understanding, possibly through more advanced large language models (LLMs) or additional fine-tuning, could improve its performance in handling intricate and ambiguous user prompts.

E. Real-world Implementation

The current system was evaluated in a simulated environment, where external factors like weather, terrain, and real-world communication latencies were not considered. Transitioning the system from simulation to real-world drone operations presents additional challenges, such as ensuring stable connections between drones and control systems, dealing with unexpected obstacles, and managing regulatory requirements. Future work should aim to pilot this system in real-world environments, gradually improving its robustness to real-time variables.

F. Future Directions

Future research will focus on addressing these challenges by exploring new control methodologies, improving error handling mechanisms, and scaling the system to larger swarms. Additionally, incorporating real-time monitoring and feedback mechanisms directly into the AI agent, as well as transitioning from simulation to real-world implementations, will be crucial steps for expanding the practical applications of this technology. Moreover, the use of reinforcement learning or hybrid AI techniques could further enhance the system's decision-making capabilities and improve performance in unpredictable environments.

VII. Conclusion

In this paper, we presented a Novel AI agent, powered by GPT-4o, for autonomous control of drone swarms through natural language prompts. The system achieved a 100% task success rate across all tested swarm sizes, demonstrating both its precision and scalability. Through its ability to interpret complex, multi-step commands and coordinate a variety of drone operations, the agent proved highly effective in managing drone swarms in simulated environments. This research highlights the significant potential of integrating LLM-based agents in real-time drone operations, offering a robust framework for future advancements in swarm management. The architecture also provides a foundation for expanding this technology into more complex missions, larger swarms, and broader real-world applications, making it a valuable contribution to the field of autonomous drone control.

References

[1] F. A. De Alcantara Andrade, A. Reinier Hovenburg, L. Netto De Lima, C. Dahlin Rodin, T. A. Johansen, R. Storvold, C. A. Moraes Correia, and D. Barreto Haddad, "Autonomous Unmanned Aerial Vehicles in Search and Rescue Missions Using Real-Time Cooperative Model Predictive Control," *Sensors*, vol. 19, no. 19, p. 4067, Sep. 2019.

[2] A. Ryan, J. Tisdale, M. Godwin, D. Coatta, D. Nguyen, S. Spry, R. Sengupta, and J. K. Hedrick, "Decentralized Control of Unmanned Aerial Vehicle Collaborative Sensing Missions," in *2007 American Control Conference*. New York, NY, USA: IEEE, Jul. 2007, pp. 4672–4677.

[3] X. Fu, P. Feng, and X. Gao, "Swarm UAVs Task and Resource Dynamic Assignment Algorithm Based on Task Sequence Mechanism," *IEEE Access*, vol. 7, pp. 41 090–41 100, 2019.

[4] Q. Dong and Z. Liu, "Formation control for unmanned aerial vehicle swarm with disturbances: A mission-driven control scheme," *Optimal Control Applications and Methods*, vol. 44, no. 3, pp. 1441–1462, May 2023.

[5] J. Wang, S. Duan, S. Ju, S. Lu, and Y. Jin, "Evolutionary Task Allocation and Cooperative Control of Unmanned Aerial Vehicles in Air Combat Applications," *Robotics*, vol. 11, no. 6, p. 124, Nov. 2022.

[6] G. Leu and J. Tang, "Survivable Networks via UAV Swarms Guided by Decentralized Real-Time Evolutionary Computation," in *2019 IEEE Congress on Evolutionary Computation (CEC)*. Wellington, New Zealand: IEEE, Jun. 2019, pp. 1945–1952.

[7] Z. Luan, Y. Lai, R. Huang, Y. Yan, J. Wang, J. Lu, and B. Chen, "Hierarchical Large Language Models in Cloud-Edge-End Architecture for Heterogeneous Robot Cluster Control," in *Proceedings of the 2023 2nd International Symposium on Computing and Artificial Intelligence*. Shanghai China: ACM, Oct. 2023, pp. 102–105.

[8] J. Brawer, K. Bishop, B. Hayes, and A. Roncone, "Towards A Natural Language Interface for Flexible Multi-Agent Task Assignment," 2023.

[9] P. Li, V. Menon, B. Gudiguntla, D. Ting, and L. Zhou, "Challenges Faced by Large Language Models in Solving Multi-Agent Flocking," 2024.

[10] K.-S. Park, S.-H. Kim, G. E. Guerra Padilla, K.-J. Kim, and K.-H. Yu, "Operational Performance Evaluation of Remote Controllers for Manual Control of UAV," *Journal of Institute of Control, Robotics and Systems*, vol. 24, no. 4, pp. 315–320, Apr. 2018.

[11] M. Abdelkader, M. Mabrok, and A. Koubaa, "OCTUNE: Optimal Control Tuning Using Real-Time Data with Algorithm and Experimental Results," *Sensors*, vol. 22, no. 23, p. 9240, Nov. 2022.

[12] P. Zarsky, J. Hnidka, and D. Rozehnal, "Real-Time UAV Data Acqusition System," in *2023 International Conference on Military Technologies (ICMT)*. Brno, Czech Republic: IEEE, May 2023, pp. 1–6.

[13] S. S. Abdi and D. A. Paley, "Safe Operations of an Aerial Swarm via a Cobot Human Swarm Interface," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. London, United Kingdom: IEEE, May 2023, pp. 1701–1707.

[14] Shiyan, "Approach to conception and modeling for distributed hierarchical control for autonomous drone swarm," *Advances in Machine Learning & Artificial Intelligence*, vol. 5, no. 1, pp. 01–08, Feb. 2024.

[15] B. Williamson, E. Taranta, Y. Moolenaar, and J. LaViola Jr., "Command and Control of a Large Scale Swarm Using Natural Human Interfaces," *Field Robotics*, vol. 3, no. 1, pp. 301–322, Jan. 2023.

[16] Y. Chen, Y. Zhang, G. Zhang, and Y. Gu, "A Drone Swarm-Based Wildfire Search and Rescue Method with Autonomous Behavior Modeling and Centralized Task Assignment," in *2024 IEEE 4th International Conference on Power, Electronics and Computer Applications (ICPECA)*. Shenyang, China: IEEE, Jan. 2024, pp. 341–345.

[17] H.-P. Chang, L.-Y. Lin, and K.-Y. Lian, "Autonomous Swarm Flight Control of Quadrotors Based on Boids Model & RTK-GPS Positioning," in *2023 International Automatic Control Conference (CACS)*. Penghu, Taiwan: IEEE, Oct. 2023, pp. 1–6.

[18] J. A. Ricardo, L. Giacomossi, J. F. S. Trentin, J. F. B. Brancalion, M. R. O. A. Maximo, and D. A. Santos, "Cooperative Threat Engagement Using Drone Swarms," *IEEE Access*, vol. 11, pp. 9529–9546, 2023.

[19] Z. Jiang, T. Song, B. Yang, and G. Song, "Fault-Tolerant Control for Multi-UAV Exploration System via Reinforcement Learning Algorithm," *Aerospace*, vol. 11, no. 5, p. 372, May 2024.

[20] J. Liu, H. Luo, R. Ruby, H. Wang, H. Tao, and K. Wu, "UAV-based Reliable Optical Wireless Communication via Cooperative Multi-agent Reinforcement Learning Approach," in *2023 IEEE 29th International*

Conference on Parallel and Distributed Systems (ICPADS). Ocean Flower Island, China: IEEE, Dec. 2023, pp. 856–863.

[21] M.-A. Blais and M. A. Akhloufi, "Proximity-Based Reward Systems for Multi-Agent Reinforcement Learning," in *2024 International Conference on Image Processing and Robotics (ICIPRoB)*. Colombo, Sri Lanka: IEEE, Mar. 2024, pp. 1–6.

[22] W. Lee, "Federated Reinforcement Learning-Based UAV Swarm System for Aerial Remote Sensing," *Wireless Communications and Mobile Computing*, vol. 2022, pp. 1–15, Apr. 2022.

[23] T. Yoo, S. Lee, K. Yoo, and H. Kim, "Reinforcement Learning Based Topology Control for UAV Networks," *Sensors*, vol. 23, no. 2, p. 921, Jan. 2023.

[24] R. Ganesan, X. M. Raajini, A. Nayyar, P. Sanjeevikumar, E. Hossain, and A. H. Ertas, "BOLD: Bio-Inspired Optimized Leader Election for Multiple Drones," *Sensors*, vol. 20, no. 11, p. 3134, Jun. 2020.

[25] C. Toma, M. Popa, B. Iancu, M. Doinea, A. Pascu, and F. Ioan-Dutescu, "Edge Machine Learning for the Automated Decision and Visual Computing of the Robots, IoT Embedded Devices or UAV-Drones," *Electronics*, vol. 11, no. 21, p. 3507, Oct. 2022.

[26] W. Fan, K. Luo, S. Yu, Z. Zhou, and X. Chen, "AoI-driven Fresh Situation Awareness by UAV Swarm: Collaborative DRL-based Energy-Efficient Trajectory Control and Data Processing," in *2020 IEEE/CIC International Conference on Communications in China (ICCC)*. Chongqing, China: IEEE, Aug. 2020, pp. 841–846.

[27] J. Y. Zhu, C. G. Cano, D. V. Bermudez, and M. Drozdzal, "InCoRo: In-Context Learning for Robotics Control with Feedback Loops," 2024.

[28] Z. Xu, Y. Zhang, E. Xie, Z. Zhao, Y. Guo, K.-Y. K. Wong, Z. Li, and H. Zhao, "DriveGPT4: Interpretable End-to-End Autonomous Driving Via Large Language Model," *IEEE Robotics and Automation Letters*, vol. 9, no. 10, pp. 8186–8193, Oct. 2024.

[29] J. Song, Z. Zhou, J. Liu, C. Fang, Z. Shu, and L. Ma, "Self-Refined Large Language Model as Automated Reward Function Designer for Deep Reinforcement Learning in Robotics," 2023.