ICRA2026 Late Breaking Results Poster presented at
2026 IEEE International Conference on Robotics and Automation (ICRA 2026)
June 1-5, 2026. Vienna, Austria

A Large Language Model-Based Mission Manager for Autonomous UAV Control

1st Miloš Cihlář *CEITEC – Central European Institute of Technology*
Brno University of Technology
Brno, Czech Republic
Milos.Cihlar@vut.cz

***Abstract*— Autonomous unmanned aerial vehicles (UAVs) are traditionally controlled using behavior trees or state machines, which provide deterministic execution but limited adaptability in dynamic environments. Extending these conventional systems to handle new tasks requires manual specification of additional nodes or transitions, creating a scalability challenge as mission complexity increases. This work introduces a high-level mission manager leveraging local Large Language Models (LLMs) for autonomous UAV control. The system allows operators to issue high-level commands in natural language, which the LLM interprets and decomposes into sequences of ROS 2 actions, such as takeoff, navigation, object localization, and landing, without mission-specific programming. The LLM does not directly control the UAV but selects from a constrained set of tools mapped to ROS 2 actions or services. Real-time robot state is injected into the model context, ensuring that decisions are based on actual system status and environment perception.**

I. Introduction

The proposed system translates natural language commands into executable UAV actions through a layered architecture based on Large Language Models (LLMs). The LLM Manager is implemented as a ROS 2 C++ node performing an iterative think–act loop. In each cycle, a State Aggregator collects live telemetry, including position, battery level, altitude, and landing state. This data, together with conversation history (compacted mission plans) and tool definitions, is provided to the LLM.

The system runs on a large local model hosted on our server, Qwen 72B, enabling low-latency reasoning without cloud dependency. Smaller models can also be deployed locally on devices such as NVIDIA Jetson for basic missions, though with longer inference times. Deployment and testing in real-world conditions will be performed in future work. The LLM can read arbitrary ROS 2 topics and call any available services or actions. It does not require explicit mapping of data to actions, as it automatically identifies relevant inputs and continues execution until the mission is completed or aborted.

For safe and reliable control, the system relies on well-defined UAV functions—takeoff, offboard, go to pose, go through poses, and land—which form the foundation for all higher-level tasks. These core actions are frequently used by the UAV, simplifying LLM planning and ensuring predictable behavior.

II. Simulation Environment

Validation and development were performed in Gazebo with PX4 SITL to replicate real-world UAV missions without hardware risk. The scenario, inspired by the European Rover Challenge 2025, features an outdoor environment containing three experimental probes that the UAV must locate and identify. The UAV platform is a general-purpose F450 quadrotor equipped with cameras and altimeters. YOLO-based multi-object detection and ArUco markers were employed for probe identification and precision landing (see Figure 1).

Fig. 1: Gazebo simulation environment showing the UAV and three experimental probes for localization and precision landing.

III. System Architecture

The system consists of three main components: the LLM Manager, the Tool Registry, and the State Aggregator (see Figure 2). The LLM Manager processes natural language commands and produces either text responses or structured tool calls. It operates in iterative cycles, continuously planning a sequence of actions to complete the assigned mission. During each cycle, the LLM considers current telemetry, perception data, and system state, then decides which tool calls to execute. The system runs in a loop, allowing it to remain reactive: if a problem arises during the execution of a tool call, the LLM can detect the failure and adjust the plan dynamically rather than strictly following a pre-defined sequence. A mission ends when the LLM issues no further tool calls, at which point the task is marked as completed or failed.

The Tool Registry maps LLM outputs to ROS 2 actions and services, each encapsulating a specific UAV capability

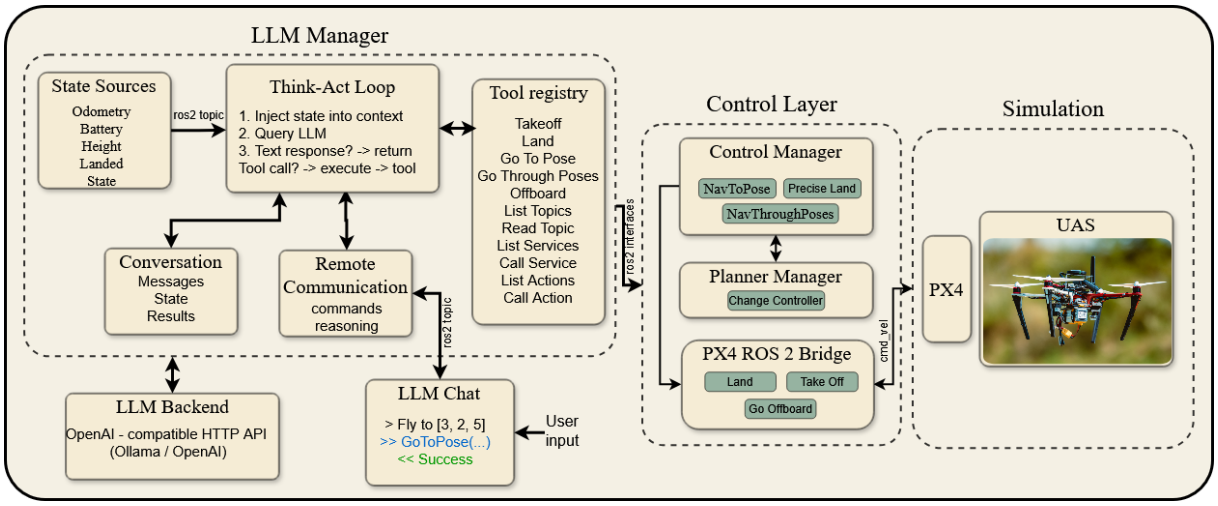

Fig. 2: High-level system architecture showing the LLM Manager, Tool Registry, and State Aggregator. The LLM interprets commands, plans tool execution, and adapts to real-time telemetry and perception data.

such as takeoff, offboard, go to pose, go through poses, or land. This mapping helps the LLM understand the available system capabilities, as core actions are used most frequently, while still allowing for more complex or custom sequences. The State Aggregator collects telemetry and perception data and injects it into the LLM context, enabling decisions to reflect the actual system state.

A chat interface allows the operator to view LLM responses and provide input at any time. The system also supports remote communication via ROS 2 topics, enabling dynamic supervision or external instruction during mission execution. The combination of iterative reasoning, tool-based execution, and interactive communication allows flexible and adaptive control of UAV missions.

IV. Experimental Validation

Four mission scenarios of increasing complexity were defined to evaluate the LLM-based mission manager:

- M1: Simple takeoff and landing.
- M2: Fly to a specified waypoint and land.
- M3: Sequential waypoint navigation through multiple points and landing.
- M4: Perception-driven mission: locate three experimental probes and perform ArUco-based precision landing.

Missions differ in the level of instruction provided. Simple missions (M1 and M2) contain all explicit steps required to complete the task, allowing the UAV to execute them without planning. In contrast, complex missions (M3 and M4) require the LLM to autonomously infer intermediate actions, generate a multi-step plan, and coordinate execution. This approach enables generalization beyond pre-defined sequences, removing the need for explicit mapping of every possible step, unlike traditional behavior trees or finite state machines. The LLM can dynamically adapt its plan if the UAV encounters unexpected conditions, demonstrating reactive and flexible decision-making.

V. Results

Across 20 test missions covering all four complexity levels, the system successfully completed 18 missions, demonstrating robust generalization and accurate selection of tool calls. Perception-driven tasks and mission-specific objectives were executed effectively, with only minor failures. In two cases, the UAV was unable to generalize the problem and execute actions in the correct order, such as attempting trajectory execution without prior takeoff or offboard activation. These issues suggest that improvements in the system description provided to the LLM or enhancements to failure-handling instructions could further increase reliability.

Performance metrics, summarized in Table I, include mission duration, number of LLM calls, and number of tool calls. These results indicate predictable and bounded system behavior, confirming that the LLM-based approach can handle both simple and complex missions while maintaining reactive and adaptive control.

TABLE I: Mission performance metrics

Mission	Success	LLM Calls	Tool Calls	Duration (s)
M1	5/5	3	2	50
M2	5/5	5	4	65
M3	4/5	5.4	3.6	89
M4	4/5	7.75	5.8	81

VI. Discussion

The results show that the LLM-based mission manager can reliably execute both simple and complex UAV missions. While simple missions follow explicit instructions, complex missions demonstrate the system's ability to infer intermediate steps and adapt plans based on real-time telemetry and perception. Minor failures highlight the need for improved guidance on prerequisites and failure handling. Overall, the approach offers greater flexibility and scalability than traditional behavior trees or state machines, enabling reactive, adaptive, and generalizable autonomous UAV behavior.