26th International Congress on Modelling and Simulation, Adelaide, SA, Australia, 30 Nov–4 Dec 2025
mssanz.org.au/modsim2025

AI Large Language Models for control of uncrewed aerial systems

Simon Crase [a], **Simon Pywell** [b], **Kent Rosser** [a]

[a] *Defence Science and Technology Group, Edinburgh, South Australian, Australia,* [b] *Defence Science and Technology Group, Fishermans Bend, Victoria, Australia Email: simon.crase@defence.gov.au*

Abstract: The rapid progress and impressive capabilities of AI Large Language Models (LLMs) such as Chat-GPT present opportunities for enhanced capabilities in the field of Autonomy. We investigate the use of LLMs for natural language control of Uncrewed Aerial Systems (also referred to as UAS, UAVs or drones). We discuss the benefits LLMs can bring to UAS operations, implementation strategies, and the outcomes of UAS flight simulations using the current state-of-the-art LLMS. Furthermore, we identify an important role that simulation can play in the operational use of LLMs for UAS control.

Keywords: *Large Language Model, AI, Drone, UAV, UAS, Simulation*

1. INTRODUCTION

Large language models (LLMs) are a form of artificial intelligence (AI) that can understand, process and generate natural human language. Due to their applicability to a wide part of society and their consequential commercial potential, significant research is being applied to LLMs to accelerate their capabilities to an impressive level. This presents an opportunity for other domains, such as autonomy, to capitalise on these capabilities. In this context, LLMs can simplify the control a UAS, eliminating the need for extensive training in UAS piloting or software development for autonomous control.

2. APPLICATIONS FOR LLMS IN UAS OPERATIONS

To understand the potential value of LLMs in operation of UAS, consider the following scenario: The tasking of a long-range, fixed-wing UAS to fly 100km following the coastline to patrol for sharks at suburban beaches. This is a task that could easily be executed by a pilot in a small crewed aircraft with the pilot dynamically controlling the aircraft to follow the coastline. However, to do this with an autonomous UAS would require a human to manually generate a flight plan with hundreds of waypoints to accurately follow the coastline. Alternatively, a trained computer vision image classifier for detecting the coastline could be paired with an on-board camera, edge processing and control algorithms to control the platform to follow the coastline. This alternative requires significant skilled resources for development and model training, and the resultant capability can only do this one specific task. The use of LLMs could enable this task to be executed with minimal human effort. Regardless of whether the LLM generates hundreds of waypoints or processes live camera imagery (Han et al.) to dynamically control the platform, these could be achieved through tasking from a human using natural language and without software development skills.

Above is one example of tasking that could be given to a UAS via a LLM. Similarly, it could be following a road, the edge of a forest, a walking track, a car, searching an area for a lost hiker, identifying a bushfire outbreak, or identifying an intruder on private property. All of these tasks would take significant time and effort to develop as individual capabilities through traditional means. However, these could be quickly realised through the application of LLMs, and could even be applied mid-flight. LLMs bring agility and dynamism for dealing with unknown circumstances and capitalising on opportunities. In this paper, we focus on the use of LLMs for autonomous flight plan generation, execution and live UAS control from a ground control station as an initial demonstration, while our ongoing research aims to progress this to live sensor processing using LLMs and UAS control on-board the platform (at the edge).

3. IMPLEMENTATION AND TESTING

As LLMs have gained prominence in recent years, there have been several instances of using LLMs to control UAS in simulation (Chen et al., Tazir et al.). Our review revealed that most existing examples use middleware or specially designed interfaces for the LLM to control the UAS, instead of directly generating Python scripts as human software developers do to control a UAS. This approach simplifies the task for LLMs, which were previously unable to consistently produce functional Python code to control UAS directly. Early LLMs often called imaginary Python packages or functions while attempting to control UAS. Our testing of LLMs over a multi-year period show that with recent advances in the code generation capability of LLMs, LLMs can now generate UAS control code more reliably and there may not be a need for the middleware.

For our recent testing, we used ArduPilot SITL (Software in the Loop) UAS simulation environment, Ollama to select the LLMs, and Open Interpreter to enable LLMs to directly execute code in a terminal window. Open AI Chat-GPT-4o was used as an exemplar current state-of-the-art LLM. Open source LLMs tested include models from Meta, Microsoft, Mistral and Google.

Firstly, testing demonstrated that the LLMs were capable of installing ArduPilot SITL, appropriate python packages such as dronekit and pymavlink, and setting up the python and simulation environments. The LLMs were then given a command such as “use python to launch an ArduPilot quadcopter in SITL, climb to an altitude of 20m, fly 100m north, and then return to land”. More challenging requests that require more complicated calculations or use of context were also tested, e.g. “fly in a circle of radius 100m” or “fly 1km towards Sydney”.

In general, the LLMs were able to successfully generate code to conduct these tasks. The LLMs would produce a strategy for each stage of the flight and then generate code to connect to the UAS in the simulation, arm the UAS and put it in an appropriate flight mode, launch the UAS and climb to altitude, fly to waypoints calculated by the LLM, returning the UAS to its launch location and land. However, the LLMs could not *consistently* execute the requested UAS tasks on their first attempt (zero-shot learning). An example failure was a UAS that would launch and continuously to climb in altitude, while the LLM believed the UAS had flown north and completed all stages of the mission. Notably, when the LLMs observed these failed outcomes (either through their direct observation or being informed by the user), the LLMs were rapidly able to produce revised code to try to overcome these failures. This often led to success on subsequent attempts (few-shot learning).

These observations provide an insight into the current capabilities of LLMs for control of UAS. Their capabilities have progressed to the stage where they are able to execute complex tasks such as successfully executing UAS flights without the need for middleware or simplified interfaces. However, the LLMs are not able to consistently execute the requested UAS tasking at their first attempt. When tested, open source and smaller LLMs that could be run on a companion computer on-board a UAS were generally less capable than leading models, but their capabilities will improve in subsequent generations.

4. THE ROLE OF SIMULATION

One notable insight from the testing of LLMs for control of UAS is the valuable role that tightly coupled simulations like ArduPilot SITL can play. The difference between launching a UAS in SITL and a real UAS can be as simple as changing the port or network address commands are sent to. Hence, SITL can be run on a Ground Control Station laptop that is used to control a real UAS, allowing for rapid testing and verification of LLM-generated code outcomes in simulation before pushing it to the real UAS. This could even be done mid-flight as a verification step when using an LLM to set new UAS tasking.

The capabilities of LLMs are progressing rapidly but in their current state, they are not yet able to produce reliable control of UAS at their first attempt. LLM capabilities will continue to mature and they may achieve reliable control of UAS based on dynamic natural language tasking from human operators. However, until this is achieved, we believe our proposed approach of executing the tasking in simulation (to verify outcomes) before sending it to the real UAS is a valuable strategy.

5. CONCLUSION

In conclusion, we have identified that LLMs are of value for control of UAS due to the agility that natural language tasking enables for dealing with unknown circumstances and capitalising on opportunities identified during UAS missions. The coding capability of LLMs has matured such that middleware or translators for interfacing with UAS may no longer be needed. However, while the capabilities of LLMs mature to the point of reliable code generation, utilising a tightly coupled “simulation in the loop” approach is a means of ensuring mission success in dynamic circumstances.

REFERENCES

Hui Han, Siyuan Li, Jiaqi Chen, Yiwen Yuan, Yuling Wu, et al. (2025), Video-Bench: Human-Aligned Video Generation Benchmark. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Nashville, TN, USA, 2025, pp. 18858-18868, https://doi.org/10.1109/CVPR52734.2025.01757

Guojun Chen, Xiaojing Yu, Neiwen Ling, Lin Zhong (2024), TypeFly: Flying Drones with Large Language Model https://doi.org/10.48550/arXiv.2312.14950

Tazir, M. L., Mancas, M., & Dutoit, T. (2023). From words to flight: Integrating openai chatgpt with px4/gazebo for natural language-based drone control. In International Workshop on Computer Science and Engineering 2023.

