User Friendly Drone Control using LLM for Real-World Application

Team Fury GPT

Nikil Krishnakumar | Rammesh Adhav Saravanan | Sujeendra Ramesh

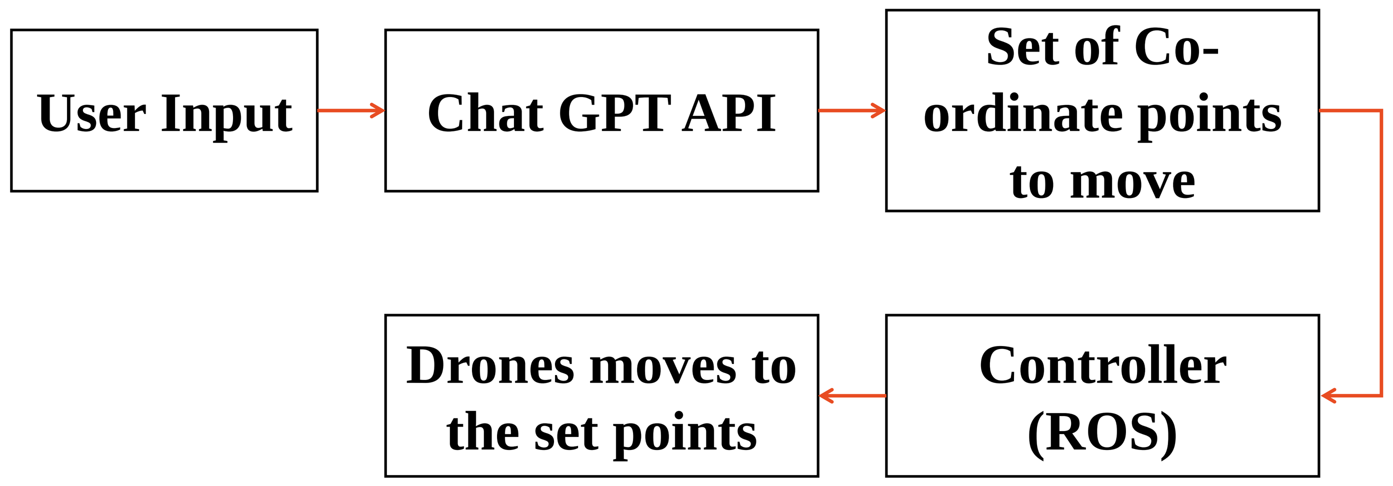
Introduction

Drones have found tremendous utility in various sectors such as agriculture, delivery, and surveillance. However, their **complex controls** present **challenges** for users from diverse backgrounds.

This project **aims** to simplify drone control by leveraging the large language model GPT-3.5. Through our approach, **users can effortlessly** operate drones without the need for **coding or manual control**. The model is made control task specific with chain of thought prompting, we enable users to set coordinates for drones based on tasks and environmental conditions.

This project is step forward to make drone control user-friendly to enhance accessibility of drone across industries.

Methodology



The flowchart illustrates the process of **converting** user inputs into a **set of coordinates** for the drone's trajectory using **GPT-3.5** model. These generated coordinates are then provided as input to the controller, which publishes **ROS** (Robot Operating System) topics to **maneuver** the drone according to the specified **set points**.

Chain of Thought prompting:

This provides example cases, guidelines, and output format for GPT 3.5 to generate the list of coordinates according to the task requested by the user.

References

[1] OpenAI. (2024). ChatGPT (GPT 3.5) [Large language model]. https://chat.openai.com/chat

[2] Oborne, Michael (2019). Mission Planner (Version 1.3.70) [Computer software]. Retrieved from https://ardupilot.org/planner/

[3] Thomas, D., Woodall, W., & Fernandez, E. (2014). Next-generation ROS: Building on DDS. In ROSCon Chicago 2014. Open Robotics. https://wiki.ros.org/noetic/Installation/Ubuntu

Results and Evaluation

We have developed a **user interface** that enables users to input tasks and view the set of **three-dimensional coordinates** generated by GPT-3.5 model as results in the response dialogue box upon selecting the **'generate response'** option. Subsequently, upon selecting **'execute controller,'** the generated set points are provided to the controller, which publishes the coordinates as ROS topics to govern the drone's movement. The ROS topic used for publish the co-ordinate is /drone/mavros/set-point-position/local and ROS topic to get the drone's position at each instance is /drone/mavros/local-position/pose. The next step's co-ordinates will be published after the pervious co-ordinates are successfully reached.

The **real-life drone successfully** completed the task by traversing a square-shaped path based on user input. Additionally, it can recognize and handle synonyms for specific tasks. Moreover, the drone is **capable of executing** various shapes such as rectangles, combination of tasks, and following provided random directions. It can also generate successful outputs when dimensions of requested shapes are provided in **complex forms** such as **area, perimeter, or in abbreviated formats.**

Figure -1
The User Interface for User Input, Response from GPT 3.5 and to access the previous operations

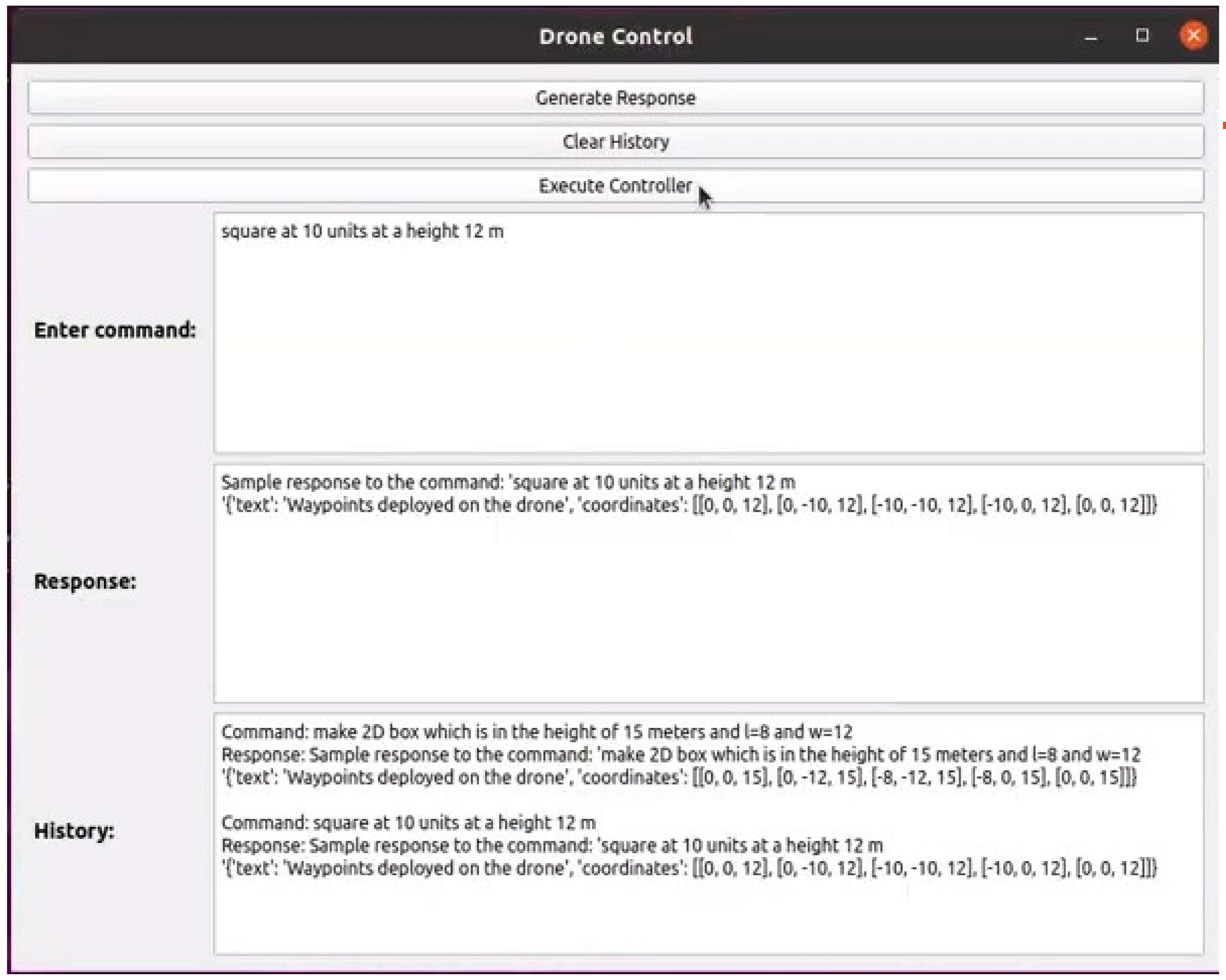


Figure - 2
The Real – life drone used for performing the specified operation

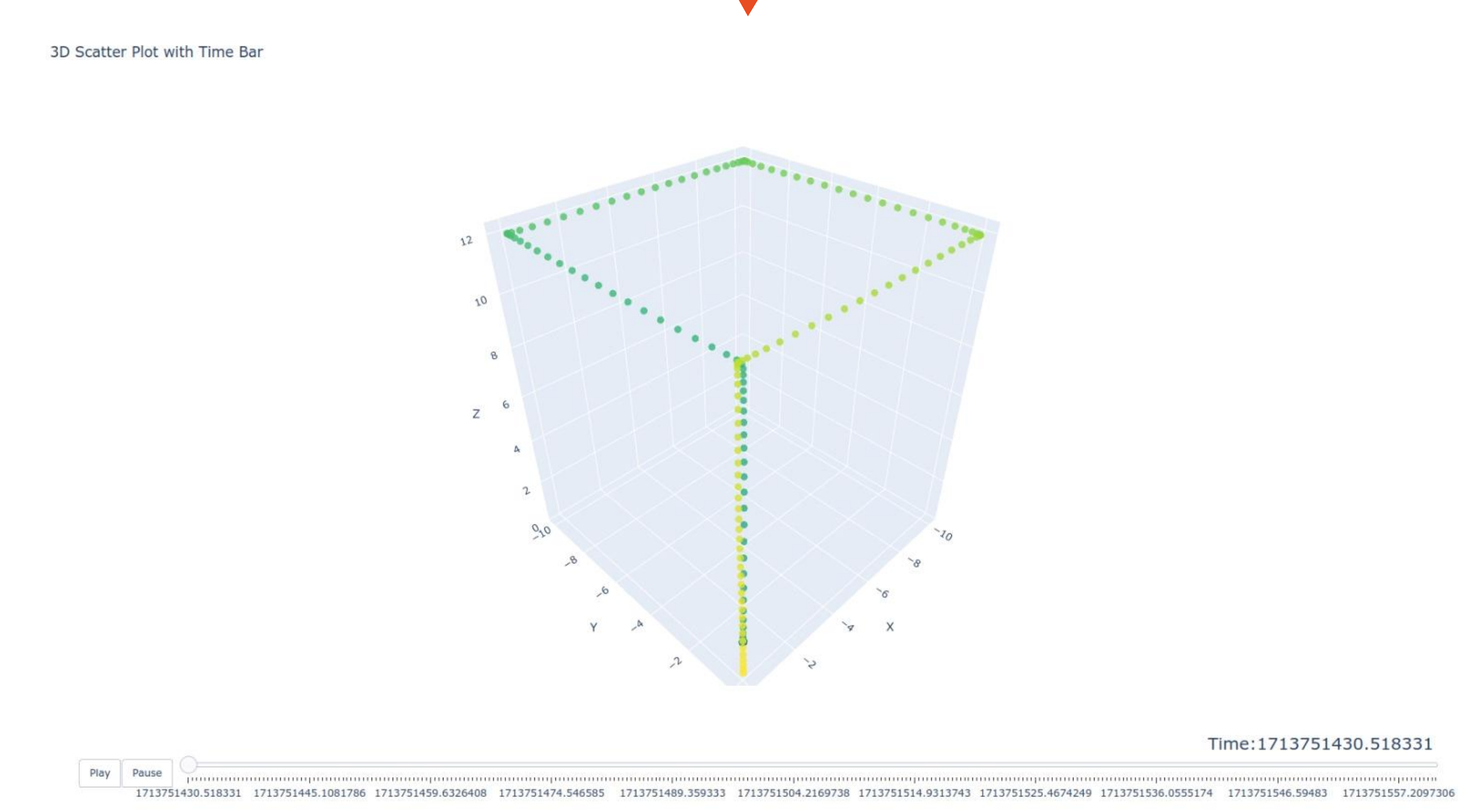


Figure -3
Shows the path travelled by the drone for user input – square at 10 units at a height 12 m

Future work

Our project's future work involves integrating vision capabilities into the drone using large language models (LLMs). For instance, if the task is to track a specific individual, such as someone wearing a red hat and white jacket, the drone will provide image and depth information to the LLM. Subsequently, the LLM will analyze the data to detect the person with the requested features and generate a list of coordinates for tracking purposes.