
Controlling robotic swarms using Large Language Models

Author:

Stoyan Ganchev
High School of Mathematics "Acad. Kiril Popov"
Plovdiv, Bulgaria
s.ganchev2002@gmail.com

Abstract

This paper presents an approach to utilizing Large Language Models (LLMs) for the control and coordination of robotic swarms. It explores how LLMs can improve the efficiency of swarm robotics in tasks such as decision-making, task translation, and collision avoidance. The study includes both simulations and real-world experiments, employing drones to demonstrate the practical application of LLMs in translating complex human instructions into coordinated actions by robotic swarms.

The project aims to develop a dedicated library for Espressif microcontrollers, enhancing their integration into swarm robotics. Their use with LLMs could be beneficial due to their low operational power.

The study contributes to the fields of multi-robot systems and LLMs, showcasing innovative ways to enhance the functionality and efficiency of robotic swarms.

Additionally, we explore the capabilities of prompt-tuned LLMs to engage in free-form dialogue, synthesize code, and employ task-specific prompting functions and closed-loop reasoning through dialogues, further broadening their application in swarm robotics.

1 Introduction

Swarm robotics[6] is an area of research that focuses on the study of multiple robots working together in a coordinated manner. These robots are inspired by the behaviours observed in natural groups, such as insects or birds. The goal is to understand how individual robots can interact with each other to achieve common objectives. This field has practical applications in fields like environmental monitoring, search and rescue missions, resource management and others.

In the field of swarm robotics, task management appears as a key problem. Typically, robotic swarms are engineered to execute a singular, pre-programmed task, which significantly restricts their efficacy in scenarios demanding a variety of activities. Moreover, it is further complicated by challenges such as decision-making, programming complexity, and collision avoidance. Each of these factors contributes to the complexity of operating a robotic swarm effectively. Additionally, human-robot communication presents difficulties, often impeding the seamless integration of human inputs into the robotic swarm's operations.

At the moment, multiple software toolkits designed for orchestrating swarm activities are being developed, mainly targeting microcontrollers such as Raspberry Pi and Nvidia[14]. These toolkits have limitations such as a restricted range of functions and complexity in programming. Furthermore, there is a noticeable lack of equivalent tools for Espressif microcontrollers, mainly due to their limited operational capabilities.

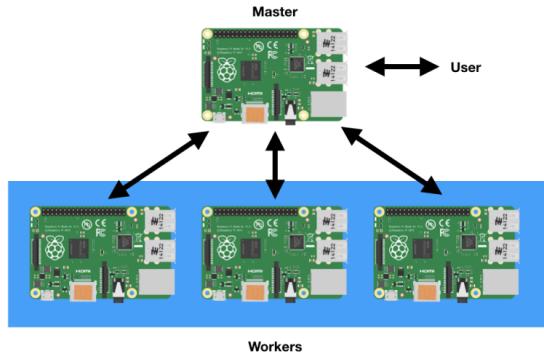


Figure 1:
On this figure we can see a schema of Rasberry Pi swarm.[4]

1.1 Hypothesis

Our hypothesis is that Large Language Models (LLMs) could be effectively used for improving various aspects of robotic swarm control, including task translation, decision-making and collision avoidance. Appendix: C. It is posited that LLMs, with their natural language processing abilities, can translate human inputs into instructions for robotic swarms, thereby improving the communication between humans and machines and simplifying swarm operation. This integration is expected to provide quicker and more accurate reactions to dynamic environments.

Moreover, we aim to create a library that is compatible with Espressif microcontrollers, which are widely used in IoT and embedded systems. Having this library would help with the integration of Espressif microcontrollers into swarm robotics and collective intelligence projects. Controlling swarms of Espressif microcontrollers with LLMs will be more practical and efficient, due to the low operational power of these microcontrollers. This would lead to faster advancements in both swarm robotics and Espressif microcontroller applications.

2 Background

This section shares insights on how Large Language Models improve robotics, make drones safer with trajectory checks, test risks using simulations, and connect devices with simple controls and communication methods. This information is important for the understanding of the project.

2.1 Large Language Models (LLMs)

Large Language Models are advanced natural language processing tools designed to handle a variety of complex tasks. These models are characterized by their ability to generate text, translate languages, synthesize code, and more. A key feature of LLMs is their interaction capabilities, particularly in dialogue-based applications, where they can generate responses and engage in conversations based on the input they receive.

2.1.1 Prompt tuning

Prompt tuning is a technique used in natural language processing (NLP) and machine learning to adapt large language models to specific tasks or improve their performance on certain types of inputs without the need for extensive retraining or fine-tuning the entire model. This approach leverages the idea that providing the right kind of prompt or context to a model can significantly influence its outputs, making it more useful for specific applications.

Task-Specific Prompts: By crafting specific prompts or instructions that lead into the text input, users can guide the model to generate responses that are more aligned with desired outcomes.

Zero-Shot Learning: In cases where no examples are provided, the prompt is designed to be sufficiently descriptive to enable the model to understand and perform the task based solely on its pre-existing knowledge and the context given.

2.1.2 LLMs for Robotics

LLMs for robotics is an emerging field that shows considerable promise. These advanced models excel in translating human language into specific robotic commands, streamlining the way humans interact with all kinds of robotic systems. This capability is pivotal in developing APIs for robot control, which greatly enhances the process of prompt engineering with LLMs. The integration of LLMs in this field signifies a leap in technological innovation and opens up new avenues for more intuitive and effective human-robot interactions.

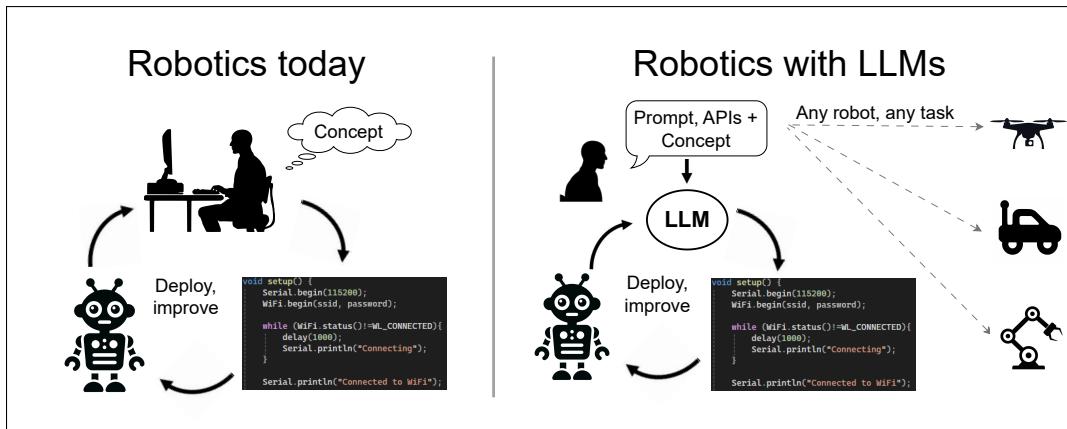


Figure 2:

Comparison between the way robots are usually programmed and the way they would be programmed using LLMs in the loop.

2.1.3 LLMs for decision making

LLMs are proving to be considerable tools in the field of decision-making, particularly in the context of swarm robotics. Their advanced natural language processing capabilities allow them to interpret, analyze, and make decisions based on large volumes of data they were trained on. This is very important in swarm robotics, where collective decision-making and coordination among multiple robots are essential. The ability of LLMs to process and understand natural language also makes them ideal for translating high-level human directives into actionable tasks for robot swarms. This synergy between human input and autonomous robotic decision-making, facilitated by LLMs, is vital for the effective operation and versatility of swarm robotics.

2.2 Trajectory optimization algorithms

Trajectory optimization filters are essential for the functionality and safety of drone swarms. Using a distributed drone swarm motion planning framework such as AMSwarm, these filters incorporate critical information about the drone system. This includes parameters like the maximum allowable speed, actuation limits of the drones, and their collision envelopes. By processing this information, the AMSwarm safety filter effectively computes feasible and collision-free trajectories. These trajectories are closely aligned with waypoints generated by the LLM, ensuring that the drone swarm's movements are both secure and precisely coordinated with the planned actions. This approach is key to balancing safety and efficiency in complex drone swarm operations. Appendix: B

2.3 Simulations

Simulations serve as tools, providing a virtual environment, similar to the physical world, where different scenarios can be tested. This not only makes the development process faster but also mitigates risks associated with real-world trials. The advantages of simulations allow us to fine-tune algorithms, optimize communication protocols, and analyse the collective behaviour of robotic swarms before deployment.

2.4 Microcontroller

Microcontrollers are compact integrated circuits with a central processing unit, memory and input/output peripherals, which have a main role in modern technology. Serving as the computational core of electronic devices, microcontrollers enable autonomous operation by executing pre-programmed tasks and responding to external inputs. In this project, we use Espressif microcontrollers, which can be seen in Appendix E.1.

2.5 Communication protocols

Communication protocols[11] are standardized frameworks that enable effective communication between electronic devices. Similar to how language conventions help people with different languages understand each other, these protocols create a shared language for devices. Well-defined rules and protocols dictate how data is structured, transmitted, and received, which ensures accurate and reliable information exchange. The communication protocols used in this project can be seen in Appendix A.

2.6 Positioning systems

Positioning systems determine the position of an object in space. They are very important in robotics, especially drone swarms. These systems allow the drones to determine their own and the location of the others, to make sure not to collide.

2.6.1 Global Positioning System (GPS)

In the context of drone navigation and control, the most recognized positioning system is the Global Positioning System. GPS operates by utilizing signals from satellites orbiting the Earth. These satellites continuously transmit signals received by GPS devices, including

those on drones. By processing these signals, GPS provides drones with essential data regarding their geographic location (latitude and longitude), altitude, and precise time. This information is vital for navigation and flight control, enabling drones to maintain accurate course and altitude and to synchronize operations in cases involving multiple drones. The GPS network is composed of multiple satellites, ensuring global coverage and consistent availability of positioning information.

3 Approach

This section presents the practical implementation and testing of the concepts proposed in the hypothesis. Our research contains solutions, focusing on the integration of LLMs with both simulated environments and actual drones. Additionally, we have developed an Espressif microcontroller library designed for the control of robotic swarms. Through testing and refinement, we aim to demonstrate the effectiveness of these solutions in real-world scenarios.

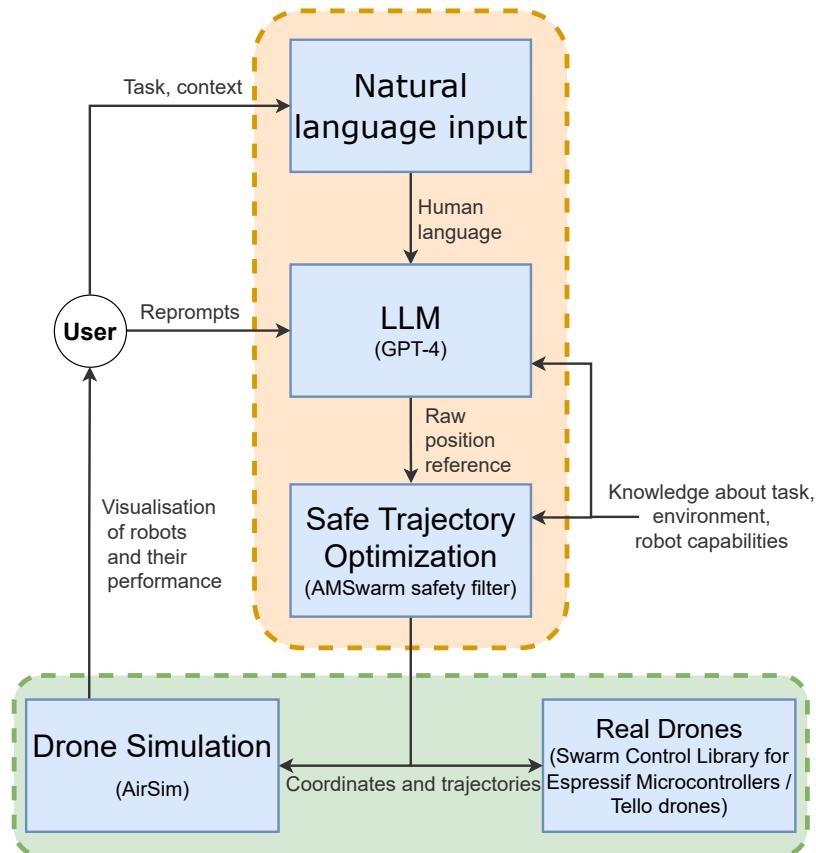


Figure 3: Block diagram of the project

3.1 Software

In our research, we use GPT-4 as the LLM of choice, leveraging its training on diverse data sources. This training equips GPT-4 with an understanding of various contexts and scenarios. Its ability to use wide-ranging data makes it good at decision-making. This ability is essential in our application, as it allows the prompt-tuned version of GPT-4 to analyze complex situations, weigh various factors, and make informed decisions. We strictly identify the limitations when tuning the model, which can be seen in Appendix D.

3.1.1 Simulation

AirSim simulation. In our research, we use AirSim as the simulation environment to test and validate our hypotheses. It provides a realistic and versatile platform for simulation, offering real-world physics and drone dynamics. Within this simulated environment, we have successfully created a scenario involving a group of five drones. These drones are programmed to execute a series of coordinated movements simultaneously. This simulation serves as proof of our theoretical models and also lays the groundwork for practical applications in swarm drone control and coordination.

GPT-4 capabilities assessment. In this experiment, we tailored the GPT-4 model to make strategies for wildfire suppression using drone swarms. The task involved creating plans that included a specified number of drones, water refilling location and area of the fire. This approach aimed to assess the model's ability to handle complex logistical challenges in a critical environmental context.

The modified version showcased its abilities by effortlessly formulating complex strategies, converting tasks into detailed diagrams, creating precise pseudocode for programming, and conducting in-depth analysis. Its ability to adapt to these tasks met our requirements and expectations.

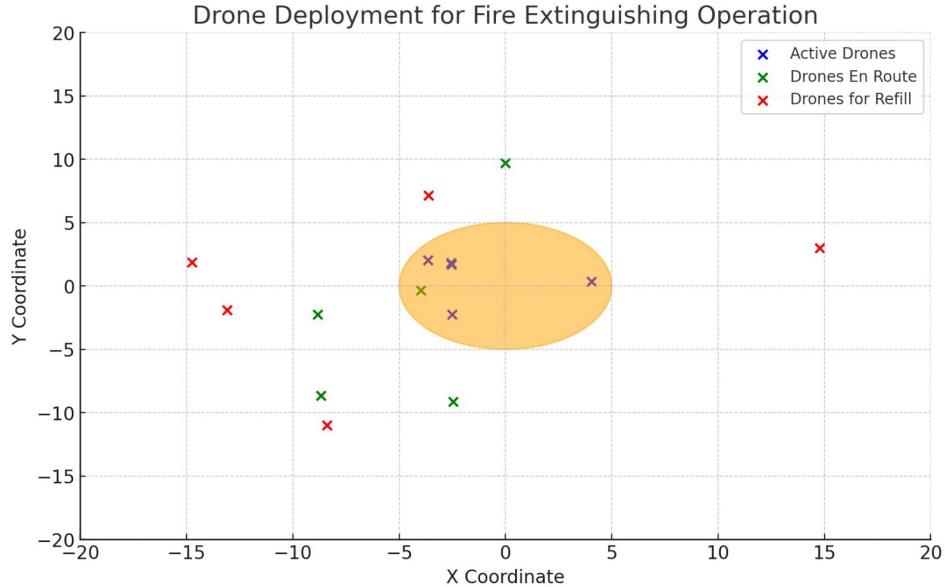


Figure 4:

On this figure we can see a schema made from GPT-4 that illustrates the completion of the given task.

GPT-4 control of AirSim simulation. In our research, we use a task-specific-tuned GPT-4 model to orchestrate a simulated swarm of drones within the AirSim environment. The initial phase of our experiment involves defining the number of drones to be controlled. Following this setup, we employed natural human language to give commands and directives to the swarm. This approach allowed for intuitive and flexible control, with GPT-4 not only interpreting our instructions but also actively making decisions while writing the control functions. This ensured that the tasks we articulated were executed correctly and efficiently. By translating these instructions into specific control actions for each drone in the swarm, the LLM demonstrated its ability to process and execute natural language commands in real-time. This capability was important in showcasing the potential of using advanced language models for dynamic and complex tasks, such as swarm robotics control.

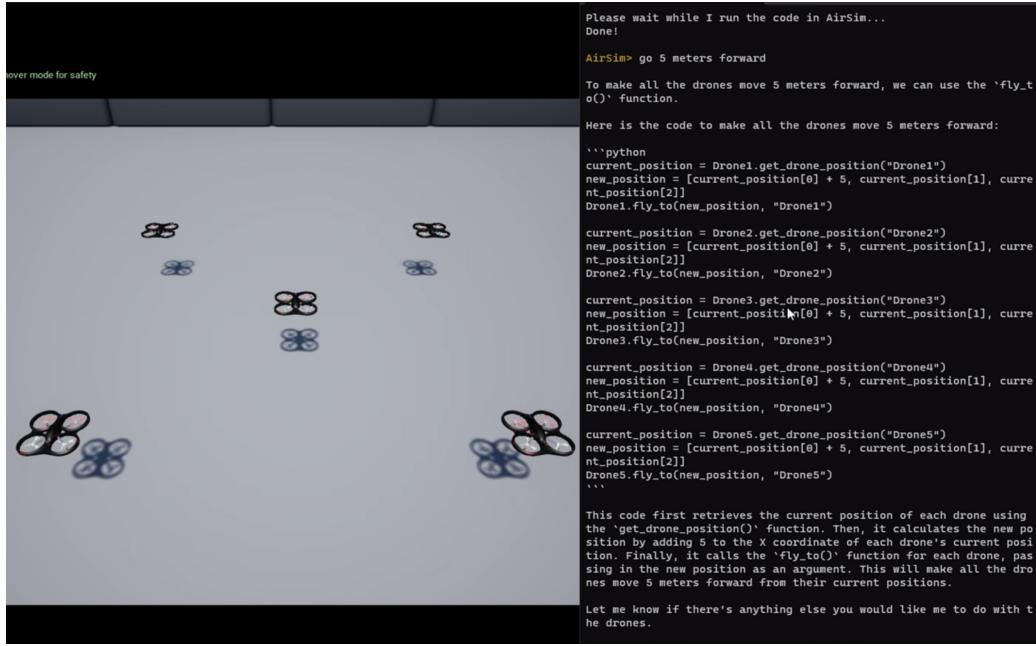


Figure 5:

On this figure we can see a photo of the AirSim simulation with five drones, which are simultaneously being controlled with commands, given in human language and translated to the robots by GPT-4.

3.1.2 Swarm Control Library for Espressif Microcontrollers

Espressif microcontrollers present an effective choice for swarm robotics applications, offering a balance of efficiency and compactness ideal for coordinating collective behaviours. Despite their numerous advantages, these devices encounter limitations in processing power when tasked with executing more sophisticated algorithms. However, in the context of using LLMs for swarm control, the primary role of Espressif microcontrollers shifts towards executing commands received from the LLM, rather than processing complex algorithms directly, which makes it very convenient for our needs.

We have started the creation of a specialized library designed to make the integration of LLM into a robotic swarm consisting of Espressif microcontrollers more seamless. This library is programmed using the Arduino language and is compatible with a range of Espressif microcontroller models.

We have developed two different methodologies of communication: MQTT and HTTP. Both of them require Wi-Fi. MQTT allows multiple devices to be controlled at the same time, as long as they are within 20-25 meters of each other. This type of communication is very fast, with a latency of only 10 to 20 milliseconds.

In contrast, the HTTP connection protocol tolerates communication between devices that are further apart, which allows for remote interaction. However, it's important to note that the farther apart the devices are, the more delay there will be.

This gives users the option to choose the communication methodology that best fits their needs.

3.2 Hardware experiments

GPT-4 control of swarm from Tello drones. Our experimentation involved employing a task-specific-tuned version of GPT-4 to orchestrate the control of a swarm consisting of two Tello drones. The model translates tasks presented in human language into executable functions, directly controlling the movements and actions of the drones, following waypoints and trajectories generated from the LLM. The implementation of GPT-4 in this context demonstrates a remarkable level of precision and efficiency, ensuring that the drones perform assigned tasks without any collisions. The success of this approach shows the potential of LLMs like GPT-4 in enhancing the responsiveness of robotic systems to human instructions.



Figure 6:

On this figure we can see a photo of the two Tello drones, which are being controlled by the GPT-4.

Swarm Control Library for Espressif Microcontrollers. Our library can interface with many different Espressif microcontrollers. The only requirement for compatibility is the existence of a WiFi module. Furthermore, the integration of sensors and actuators with these controllers will enable the creation of a comprehensive and operational swarm system.

We undertook numerous experiments using ESP32, ESP8266, and ESP12 devices. Our observations indicate that the library seamlessly works with multiple devices, thereby enabling fast inter-device communication. The latency was approximately 10 milliseconds, with a throughput of 500 messages per second. Furthermore, we have established a simple broker-client system, visually shown in the illustration below.

In light of our experimental findings, it is safe to say that the library functions in a robust manner.

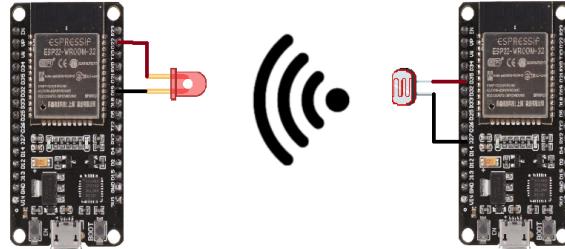


Figure 7:

On this figure we can see a schema of the esp broker-client system. When the light sensor of the broker does not detect light, it sends a message to all clients, to light up their connected LEDs. An implemented version of the schema can be seen in Appendix E, figure 15.

4 Related work

Swarm control is essential for autonomous robotics, IoT, and industrial automation systems. Despite the growing importance of swarm control in various domains, a notable gap exists in methods for controlling them in more straightforward methods.

4.1 ChatGPT for Robotics

The paper "ChatGPT for Robotics"[15] explores the integration of OpenAI's ChatGPT in various robotics applications. It presents a strategy combining prompt engineering and a robotics API library, enabling ChatGPT to adapt to diverse tasks, simulators and factors in robotics. The study evaluates the effectiveness of different prompt engineering techniques and ChatGPT's abilities to engage in closed-loop reasoning through dialogues. It covers applications in aerial navigation, manipulation and embodied agents, demonstrating ChatGPT's versatility in solving complex robotics tasks and interacting primarily through natural language instructions. In this work, we pursue a similar goal, but with a focus on language-based control of robotic swarms.

Imagine we are working with a household robot. The job of this robot is to clean the house. The objects available around are: vacuum cleaner, furniture cleaner, soft cloth.

The functions you can use are:

- `locate_object(object_name)`: Returns coordinates of the object.
- `go_to_location(coordinates)`: Moves robot to the given location. Returns nothing.
- `pick_object(object_name)`: Pick up the object. Returns nothing.
- `use_object(object_name)`: Uses the object. Returns nothing.

Can you use these functions to write code for cleaning the house?

```
def clean_house():
    # Step 1: Vacuum Cleaning
    vacuum_coordinates = locate_object("vacuum cleaner")
    go_to_location(vacuum_coordinates)
    pick_object("vacuum cleaner")
    use_object("vacuum cleaner")

    # Step 2: Furniture Cleaning
    furniture_cleaner_coordinates = locate_object("furniture cleaner")
    go_to_location(furniture_cleaner_coordinates)
    pick_object("furniture cleaner")
    use_object("furniture cleaner")

    # Step 3: Dusting with Soft Cloth
    soft_cloth_coordinates = locate_object("soft cloth")
    go_to_location(soft_cloth_coordinates)
    pick_object("soft cloth")
    use_object("soft cloth")
```

Figure 8:

Example of a good prompting practice when defining APIs for a robotics problem.

4.2 Eureka: Human-level reward design via coding large language models

In the realm of leveraging large language models (LLMs) for innovative applications, the "Eureka"[9] project stands as a noteworthy precedent. This research aims to utilize the capabilities of LLMs for human-level reward design, employing a methodical approach to encode complex tasks into comprehensible instructions that LLMs can interpret and act upon. The similarities between our work on controlling robotic swarms and the "Eureka" project are pronounced, particularly in our reliance on LLMs to bridge the gap between human intentions and machine executions. Both projects underscore the versatility of LLMs in translating natural language directives into precise, actionable outcomes, thereby enhancing the efficiency and effectiveness of human-machine collaboration.

4.3 PiSwarm

PiSwarm[7] is an innovative swarm robotics platform that illustrates advancements in autonomous collective behaviour. Orchestrated through a network of interconnected Raspberry Pi units, PiSwarm experiments with the combination of computational intelligence and robotic systems. This platform serves as a framework for exploring emergent phenomena, distributed algorithms, and decentralized decision-making, thereby contributing to the evolution of swarm robotics research and its potential applications in various fields. In contrast to "PiSwarm", this study focuses on the use of Espressif microcontrollers for robotic swarms, controlled by LLMs.

5 Conclusion

In conclusion, this research marks a significant stride in the field of swarm robotics, successfully bridging the gap between the advanced language processing abilities of LLMs and the operational efficiency of Espressif microcontrollers.

Through simulations in AirSim and practical experiments with real drones, the research conducts a detailed assessment of the GPT-4 model, prompt tuned for specific tasks, such as strategic planning, employing a wildfire extinguishing scenario as a specific test case. The experiments with real drones show the project applicability and the potential of LLMs to enhance decision-making and operational efficiency in complex scenarios. Furthermore, a dedicated library has been developed to facilitate the future integration of LLMs with robotic swarms using Espressif microcontrollers, utilising MQTT and HTTP communication protocols, which have low latency and enable long-distance data transfer, facilitating more effective swarm control.

As we continue to explore and enhance the capabilities of both LLMs and microcontroller-based swarms, the possibilities for their application in diverse fields such as environmental monitoring, search and rescue, and IoT systems are endless, promising a future where human-to-robot communication is more intuitive and effective.

5.1 Future work

Limit testing of LLMs for swarm coordination. Our exploration will focus on evaluating the capabilities of LLMs in managing swarm control across scalable scenarios. This will be achieved by testing in simulated environments and drone swarm deployments. This approach is designed to assess how LLMs perform and adapt in varying scales of operation, providing valuable insights into their effectiveness and scalability in real-world applications of swarm robotics.

Library improvements.[9] In the future, our research will focus on including Bluetooth and LoRa[12] communication to enhance swarm coordination. Additionally, we will incorporate GPS into our library and explore the use of advanced positioning algorithms in robotic swarms. This approach has the potential to advance swarm robotics for Espressif microcontrollers to higher levels of efficiency.

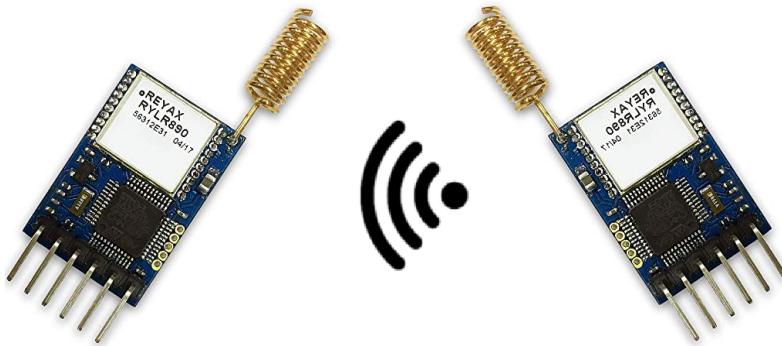


Figure 9:

On this figure we can see a schema of communication between two LoRa[12] modules.

References

- [1] Espressif esp32 microcontroller. URL https://nothans.com/wp-content/uploads/2021/03/HiLetgo_ESP32.jpg.
- [2] Http protocol architecture. URL <http://bytesofgigabytes.com/IMAGES/Networking/HTTPcommunication/http%20communication.png>.

- [3] Mqtt protocol architecture. URL <https://static.javatpoint.com/tutorial/computer-network/images/mqtt-protocol3.png>.
- [4] A raspberry pi cluster. URL <https://images.ctfassets.net/2lpsze4g694w/6qdj1mbRwW8lqB70xc2oHr/409d1e54e6848232e661ab1cf1934c0d/diagram.png>.
- [5] Vivek K. Adajania, Siqi Zhou, Arun Kumar Singh, and Angela P. Schoellig. Amswarm: An alternating minimization approach for safe motion planning of quadrotor swarms in cluttered environments, 2023.
- [6] Manuele Brambilla, Eliseo Ferrante, Mauro Birattari, and Marco Dorigo. Swarm robotics: a review from the swarm engineering perspective. *Swarm Intelligence*, 7(1):1–41, Mar 2013. ISSN 1935-3820. doi: 10.1007/s11721-012-0075-2.
- [7] James Hilder, Rebecca Naylor, Artjoms Rizihs, Daniel Franks, and Jon Timmis. The pi swarm: A low-cost platform for swarm robotics research and education. In Michael Mistry, Aleš Leonardis, Mark Witkowski, and Chris Melhuish, editors, *Advances in Autonomous Robotics Systems*, Lecture Notes in Computer Science, page 151–162, Cham, 2014. Springer International Publishing. ISBN 978-3-319-10401-0. doi: 10.1007/978-3-319-10401-0_14.
- [8] Aoran Jiao, Tanmay P. Patel, Sanjmi Khurana, Anna-Mariya Korol, Lukas Brunke, Vivek K. Adajania, Utku Culha, Siqi Zhou, and Angela P. Schoellig. Swarm-gpt: Combining large language models with safe motion planning for robot choreography design, 2023.
- [9] Yecheng Jason Ma, William Liang, Guanzhi Wang, De-An Huang, Osbert Bastani, Dinesh Jayaraman, Yuke Zhu, Linxi Fan, and Anima Anandkumar. Eureka: Human-level reward design via coding large language models. *arXiv preprint arXiv: Arxiv-2310.12931*, 2023.
- [10] Alexander Maier, Andrew Sharp, and Yuriy Vagapov. Comparative analysis and practical implementation of the esp32 microcontroller module for the internet of things. In *2017 Internet Technologies and Applications (ITA)*, pages 143–148, 2017. doi: 10.1109/ITECHA.2017.8101926.
- [11] Neven Nikolov. Research of mqtt, coap, http and xmpp iot communication protocols for embedded systems. In *2020 XXIX International Scientific Conference Electronics (ET)*, pages 1–4, 2020. doi: 10.1109/ET50336.2020.9238208.
- [12] Ruben M. Sandoval, Antonio-Javier Garcia-Sanchez, and Joan Garcia-Haro. Optimizing and updating lora communication parameters: A machine learning approach. *IEEE Transactions on Network and Service Management*, 16(3):884–895, 2019. doi: 10.1109/TNSM.2019.2927759.
- [13] Shital Shah, Ashish Kapoor, Debadeepa Dey, and Chris Lovett. Airsim: High-fidelity visual and physical simulation for autonomous vehicles. *Field and Service Robotics*, pages 621–635, November 2017. URL <https://www.microsoft.com/en-us/research/publication/airsim-high-fidelity-visual-physical-simulation-autonomous-vehicles/>.
- [14] John C. Shovic. *Raspberry Pi IoT Projects*. Apress, Berkeley, CA, 2016. ISBN 978-1-4842-1378-0. doi: 10.1007/978-1-4842-1377-3. URL <http://link.springer.com/10.1007/978-1-4842-1377-3>.
- [15] Sai Vemprala, Rogerio Bonatti, Arthur Bucker, and Ashish Kapoor. Chatgpt for robotics: Design principles and model abilities, 2023.
- Link to code and supplementary material: <https://github.com/StoyanGanchev/Controlling-robotic-swarms-using-Large-Language-Models.git>

A Communication protocols

A.1 MQTT protocol

MQTT[11] (Message Queuing Telemetry Transport) protocol is a communication framework designed to make exchanging information between devices straightforward. It operates on a publish-subscribe model, where devices can publish data to a central point called a broker, which then distributes that data to other devices that have subscribed to receive it. This efficient approach ensures that devices only get the data they need, reducing unnecessary traffic and conserving resources.

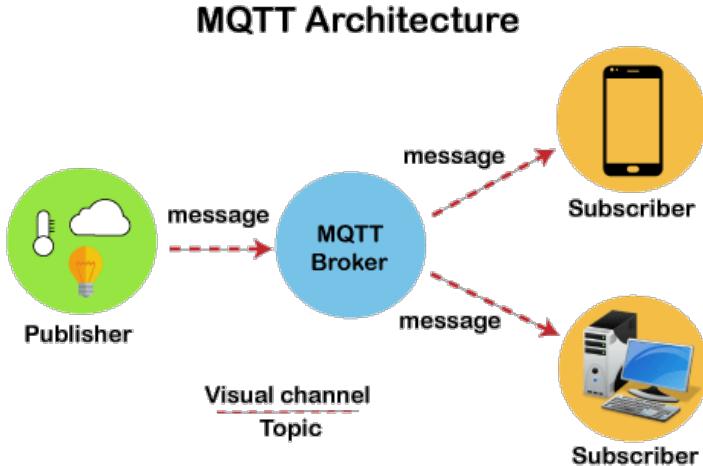


Figure 10:

On this figure we can see a schema of the MQTT Protocol for communication.[3]

A.2 HTTP protocol

HTTP[11] (Hypertext Transfer Protocol) is a communication method used between controllers. It allows them to share data and commands systematically. One controller sends a request, asking for specific information or an action. The receiving controller processes the request and replies with the needed information or confirmation. This exchange happens over networks, often the Internet.

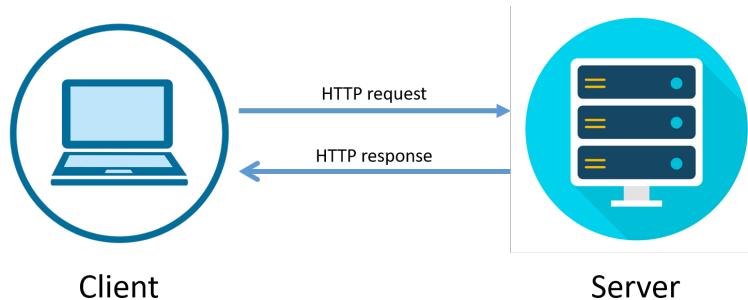


Figure 11:

On this figure we can see a schema of the HTTP Protocol for communication.[2]

B AMSwarm

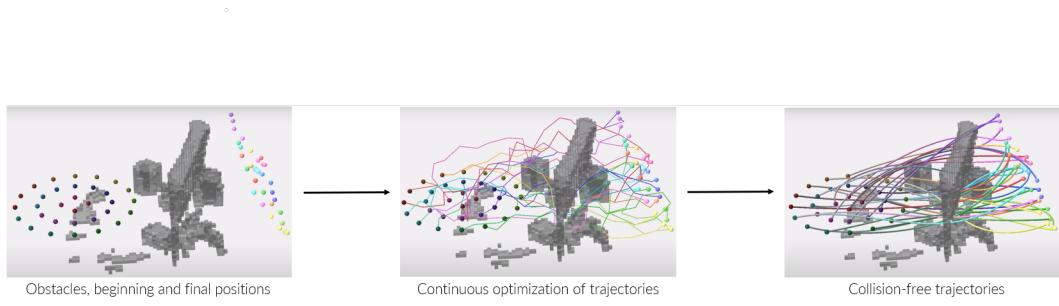


Figure 12:
On this figure we can see the sequential phases of the AMSwarm algorithm.

C Swarm control

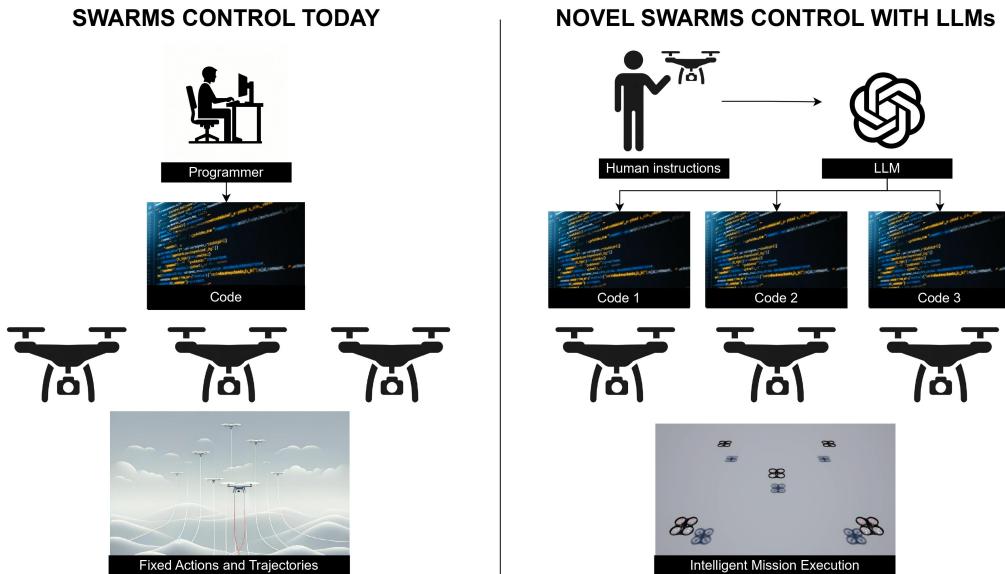


Figure 13:
On this figure we can see a comparison of the way swarms of robots are being controlled today and the way we are controlling them.

D Example of initial prompts for robotic tasks

In this section, we present the prompt used to fine-tune our AI model for drone control within the AirSim simulation. This prompt outlines available functions and execution steps, ensuring clarity on model capabilities. By contrasting correct and incorrect execution methods, we guide the model in avoiding common errors, therefore enhancing model performance for drone simulation tasks. This approach not only improves decision-making but also optimizes drone control in the virtual environment.

Guidelines for Commanding Drone Swarms with AirSim

This guide outlines the process for efficiently operating drone swarms within the AirSim simulator using Python.

Follow these instructions to ensure accurate and effective drone control:

Initial Setup:

- At the outset, determine the total number of drones.
- Initialize each drone with a unique identifier. For example:
Drone1 = AirSimWrapper("Drone1")

Command Functions:

Use the following specified functions for drone control:

- drone_id.takeoff(drone_id): Commands the drone to take off.
- drone_id.land(drone_id): Commands the drone to land.
- drone_id.fly_to([x, y, z], drone_id): Directs the drone to a specific coordinate.
- drone_id.fly_path(points, drone_id): Guides the drone along a predefined path.
- drone_id.set_yaw(yaw, drone_id): Sets the drone's yaw.
- drone_id.get_yaw(drone_id): Retrieves the drone's current yaw.
- Additional functions for position and yaw control as provided.

Execution Specificity:

- Execute the functions for each drone individually, without using loops or batch commands. Example:

Correct: Drone1.takeoff("Drone1"), Drone2.takeoff("Drone2")

Incorrect: Using loops or combined commands for multiple drones.

Command Clarity:

- Do not make assumptions. If a task requires clarification, ask explicitly. Example:
"Question - There are two spheres. Which one should the drone fly to ?"

Axis Conventions:

- Follow these conventions for movement commands:
Forward: Positive X axis.
Right: Positive Y axis.
Up: Positive Z axis.

Prohibited Actions:

- Avoid using functions not specified in the instructions.
- Do not attempt to shorten or optimize the code beyond the given guidelines.
- Do not use lists not created through the specified functions.

E Hardware components and experiments

E.1 Espressif microcontrollers

Espressif microcontrollers[10], exemplified by the ESP32, ESP8266 and ESP12 series, combine compact design with remarkable capabilities. Their small operational footprint is accompanied by a focus on energy efficiency, which is a key feature of their family. This guarantees efficient computational performance while consuming minimal power. These microcontrollers are ideal for applications that require extended autonomous functioning, making them particularly valuable for IoT, wearables, and battery-powered systems.

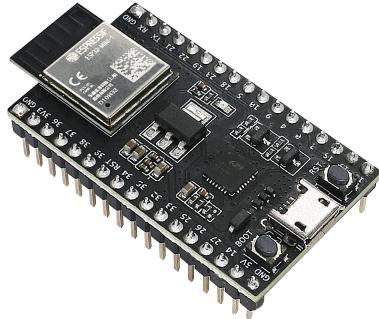


Figure 14:
On this figure we can see an Espressif ESP32 microcontroller.[1]

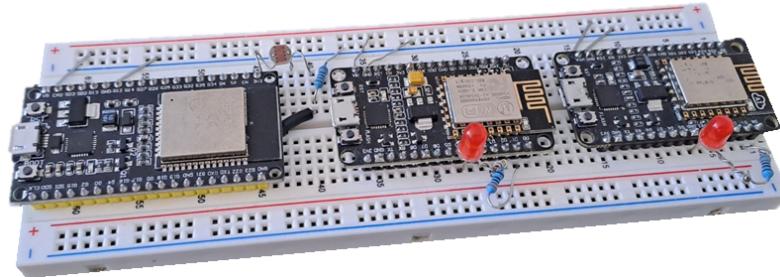


Figure 15:
On this picture we can see an implemented version of figure 7 with two clients.