Master Thesis

Swarm Drones Control Platform

A modular and scalable platform for swarm drone control

Nicolas Leboucher

October 14, 2024

Supervisors

Dr Xiao Xiao and Dr Marc Teyssier Professor at IFT

Contents

Introduction

We present a novel platform for controlling swarms of Crazyfly drones that utilizes more user-friendly technology compared to the existing Python CFlib controller. This approach demonstrates the feasibility and potential of safely operating drones around humans in indoor environments. By leveraging accessible interfaces, our platform aims to make more accessible Human-Drone technologies using widely used web technologies.

0.1 State of the Art

In recent years, swarm drone technology has advanced significantly, leading to breakthroughs in human-drone interaction and multi-robot systems. While several research efforts have focused on the safety, control, and interaction between drones and humans, there remains a gap in the availability of robust platforms for developers to control drone swarms indoors around humans. This section will explore existing works that demonstrate safe operation of drones around humans, systems that enable real-time drone control based on human input, and the identified gaps in the current technological landscape.

0.1.1 Safe Drone Flight Around Humans

Safety is a primary concern when operating drones in close proximity to humans, especially in confined spaces such as indoor environments. Several research efforts have focused on ensuring that drones can operate around humans without posing significant risks.

One such project is Drone & Me, which investigates how drones can safely navigate around humans by recognizing human poses and responding accordingly [1]. In addition, Ju et al. (2017) proposed real-time trajectory adjustments for drones to ensure safe flight paths when operating around humans indoors. These systems emphasize the importance of autonomous collision avoidance and the use of protective hardware to mitigate risks [2].

0.1.2 Controlling Drones with Human Input

Beyond safety, there has been significant progress in enabling human control over drones through various input methods, such as body gestures, hand movements, or wearable technology. The FlyJacket platform, for instance, allows a user to intuitively control a drone by wearing an exoskeleton that translates upper body movements into flight commands [3]. This system demonstrates that drones can be controlled naturally and in real-time based on human gestures, but it is limited to controlling a single drone at a time.

In a more general approach, Cai et al. (2019) explored how drones can follow human postures and gestures using computer vision, allowing for more flexible interactions with drones [4]. These systems highlight the potential for intuitive drone control but lack the scalability needed to control swarms of drones effectively.

0.1.3 Technological Platforms for Swarm Drones

Several robust platforms have been developed to enable scalable swarm drone systems. Among these, Crazyflie is a widely used platform for research on nano-quadrotor swarms. It provides the infrastructure for advanced coordination and trajectory planning of multiple drones in a shared airspace. The platform's compatibility with tools like ROS (Robot Operating System) makes it a flexible tool for swarm applications [5].

Another well-known platform is Kilobot, developed by Harvard's Wyss Institute. Though primarily a ground-based swarm robotics platform, Kilobot's decentralized control algorithms provide valuable insights into swarm intelligence and distributed behaviors, which are applicable to aerial drones as well [6]. ModQuad further builds on swarm modularity by allowing multiple drones to self-assemble mid-air, forming complex structures that can adapt to various tasks [7].

The CrazyChoir platform demonstrates how multiple Crazyflie drones can perform synchronized movements in an artistic or musical context, emphasizing scalability in both control and safety [8]. These platforms provide strong technological foundations for swarm robotics, yet none fully integrate human interaction into swarm control in indoor environments.

CrazySwarm [9] represents the current state of the art for large-scale swarm control in research environments, particularly in fields like swarm robotics, distributed systems, and human-swarm interaction. It leverages ROS for modularity and scalability, and its integration with real-time motion capture systems provides a sophisticated platform for studying collective behaviors. Conversely, CFLib Swarm focuses on providing accessible, low-level control of Crazyflie drones for smaller-scale applications, mainly for educational purposes and basic experimentation.

CrazySwarm provides a more advanced, scalable solution for handling large swarms in complex environments. It overcomes the limitations of CFLib [10] Swarm, such as lack of real-time positioning and high-level control, making it the preferred platform for research in swarm robotics. [cite on cflib lack of real-time positioning]

We want to note here that ROS adds a layer of abstraction and infrastructure that provides many advantages but also introduces overhead. CFLib avoids this overhead by interacting directly with the drones in a more lightweight package. [cite this comparaison]

0.1.4 Human Pose Detection for Drone Control

Human-drone interaction has advanced with technologies that allow drones to react to human movements. One notable platform is FlyJacket, a wearable exoskeleton that translates upper body movements into drone control. FlyJacket emphasizes intuitive and immersive control, but it is focused on single-drone systems [3].

Beyond specialized wearables, recent advances in computer vision have made it possible to detect human poses and gestures without specialized hardware. Intel RealSense, combined with MediaPipe, allows real-time human pose estimation from RGB and depth cameras. This technology can detect skeletal movements and translate them into commands for drone control [11]. The system's accuracy and low latency make it highly applicable to environments where humans and drones must coexist.

In addition, Cai et al. (2019) proposed a monocular vision system for real-time posture recognition, where drones respond dynamically to human gestures in indoor environments. This system allows for direct interaction between humans and drones but does not yet support multi-drone systems [4].

Leap Motion offers another avenue for gesture-based control by detecting hand and finger movements. While Leap Motion has been successfully integrated with single drones, there is still potential to expand this to swarm systems for more complex indoor tasks [12].

The paper titled "Drone control using Kinect-based posture detection" [11] presents a compelling approach to enhancing human-drone interaction by leveraging RGB-D (Red, Green, Blue-Depth) data from the Kinect sensor. The authors demonstrate how Kinect can effectively capture real-time human poses and positions, enabling drones to respond to user gestures with a high degree of accuracy. By utilizing the depth data, the system can determine the spatial relationship between the human operator and the drone, allowing for intuitive control commands based on body movements.

This approach significantly reduces the need for additional hardware or complex controllers, as it translates natural human actions directly into drone maneuvers. The paper emphasizes that such a system can enhance the usability and accessibility of drones, particularly in indoor environments where GPS signals may be unreliable. Furthermore, the authors highlight the potential for combining this technology with other methods, such as those explored in [11] and [3], to create a more robust human-drone interaction framework that addresses current limitations in the field.

0.1.5 Gaps in Human-Swarm Interaction Platforms

Despite advancements in both safety mechanisms and control systems, there remains a significant gap in providing a unified platform that enables developers to control swarms of drones indoors while safely interacting with humans. Current platforms like Crazyflie and ModQuad provide scalable solutions for drone swarms but lack the necessary integration with human pose recognition and safety protocols for operating indoors around people [5, 7]. Furthermore, while platforms like FlyJacket enable human control, they focus on single-drone control, leaving a gap in swarming capabilities.

Thus, a key challenge remains in the development of a platform that can:

Enable real-time control of multiple drones based on human input (e.g., gestures, body movements). That includes real-time posture and position as all platforms do not seem to include. [cite on cflib lack of real-time positioning] Ensure safe flight paths in indoor environments.

Be accessible to developers for customization and open-source integration. Python is accessible but we want to be able to use these drones in unity or a web service to not require client-drone connection.

0.2 Contribution 1: Swarm Control

0.2.1 Introduction

In this section, we present our first contribution, which is the drone control kernel. We present the software architecture that we have developed to control drones in a swarm. We also present the results of the evaluation of the drone control kernel.

We were provided a synchronous control driven API we have worked around it to handle practial asynchronous calls. We show here this work.

0.2.2 Drone Control Kernel

Introduction, Motivation and Objectives

Currently, Crazyflie drones are controlled using their Python library, cflib, which provides developers with a robust toolkit for sending commands, receiving sensor data, and managing flight tasks. While this library offers powerful capabilities, it limits the flexibility of using the Python ecosystem, as direct access to the drones requires interacting through this specific interface. As a result, broader accessibility for developers or users unfamiliar with Python is restricted, making it challenging for people from diverse technical backgrounds to engage with drone control.

To address this limitation and open drone control to a wider audience, we propose a web-based control platform. By developing a web interface, we aim to abstract the complexities of the underlying Python-based system and make the control of drone swarms more accessible to a larger range of developers, including those familiar with other technologies. This approach democratizes drone control, allowing users to interact with Crazyflie drones through web-based APIs and interfaces, facilitating easier integration into various applications, including human-drone interactions in indoor environments.

Instantaneous API Calls and Background Processing: Asynchronous APIs enhance the functionality of drone control applications by handling requests in the background. This approach maintains application functionality while keeping resources free to process new requests. Especially in drone operations where real-time responsiveness is crucial, this capability ensures efficient handling of multiple tasks simultaneously.

Adaptability to Connectivity and Execution Time: Asynchronous APIs are particularly beneficial in environments with variable connectivity or where requests have longer execution times. They allow for the processing of complex operations without causing delays in the application's responsiveness, crucial for real-time drone control where delay can have significant consequences.

Event-Driven Communication: Asynchronous APIs, enable more intelligent communication between internal and external services. This is especially important in drone operations where multiple events or commands may need to be handled in real time, ensuring smoother and more efficient workflows

Support for Various Protocols: These APIs support a range of messaging protocols and transports such as WebSockets, GraphQL and UDP subscriptions, which are essential for the dynamic and varied requirements of drone control. This flexibility allows for fast and more accessible and versatile drone operation management.

Architecture

The drone control kernel is designed to provide a robust and flexible control system for managing drone swarms. The architecture of the drone control kernel is shown in Figure 1. The main components of the system are: Web API, which provides the web-based interface for controlling the drones; Socket, which allows for lower level client drone server interface with the drones; and the Drone Control Kernel, which manages the communication with the drones using Radio. The UDP server runs in parallel with the FastAPI application, thanks to $asyncio.create\_task()$. [13] This allows the UDP server to handle datagram messages asynchronously while the FastAPI server responds to HTTP requests.

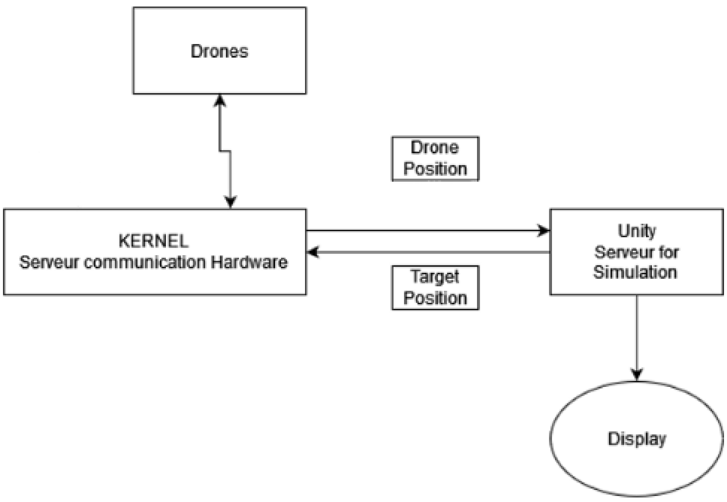

Figure 1: Drone Control Kernel, showing the main components of the system.

0.2.3 Evaluation

Each function in the UDP server is a coroutine, which allows for asynchronous execution. The UDP server listens for incoming messages from the drones and processes them accordingly. The FastAPI application provides the web-based interface for controlling the drones. It exposes various endpoints for sending commands to the drones, such as takeoff, land, and move. The FastAPI application communicates with the UDP server to send commands to the drones and receive responses. The Socket component provides a lower-level interface for communicating with the drones using the Crazyflie Python library. It handles the connection to the drones and sends commands to them using the cflib API. The Drone Control Kernel manages the communication with the drones using the Radio. It sends commands to the drones and receives responses from them using the Radio component (Crazy Radio PA).

Also a ThreeJS interface is provided to visualize the drones in 3D space. This interface allows users to see the drones' positions and orientations in real-time, providing a visual representation of the swarm's behavior. The ThreeJS interface communicates with the FastAPI application to receive updates on the drones' positions and orientations.

0.2.4 Results

0.2.5 Conclusion

0.3 Contribution 2: Object Detection

0.3.1 Introduction

In this brief section we describe the workings of the object detection system. It was developed by [Etienne Pommel] for this use case. The system is based on the mediapipe library [14], which is a state-of-the-art object detection algorithm that is capable of detecting objects in real-time. The system is able to detect objects with high accuracy in image and can be used to track objects as they move around the environment. To the standard RGB image, the system adds depth. RGB-D IMAGE

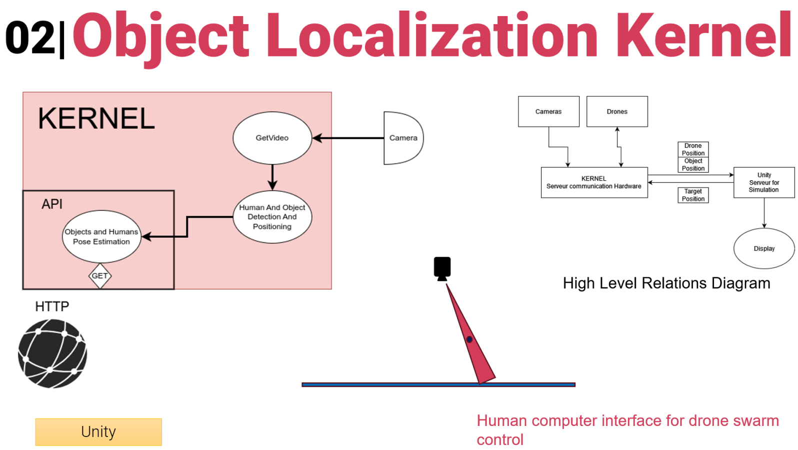

Figure 2: Drone Control Kernel, showing the main components of the system.

0.4 Contribution 3: Platform : Human-Swarm Interaction

ADD IMAGE SHOWING Parallel access to services

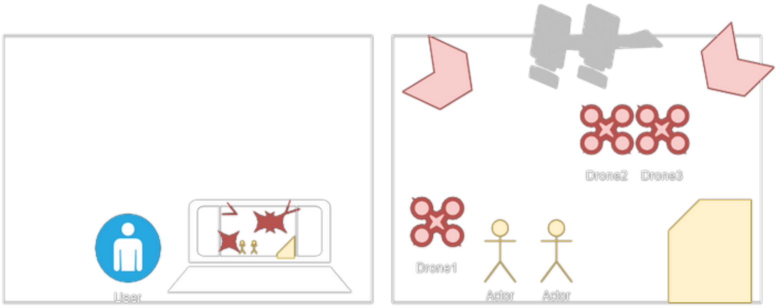

Figure 3: Our setup for controlling drones indoors around humans.

Conclusion (Phase 3)

~1 page

0.4.1 Introduction

In this section, we present our third contribution, which is the platform for human-swarm interaction. We present the setup and the software architecture that we have developed to control drones indoors around humans. We also present the results of the evaluation of the platform.

0.4.2 Collision Avoidance

One of the main challenges in controlling drones indoors around humans is ensuring that the drones can avoid collisions with people and objects. To address this challenge, we have implemented a collision avoidance system that uses depth sensors to detect obstacles in the environment. The system continuously monitors the surroundings and adjusts the drones' trajectories to avoid collisions. it is based on the original boids algorithm. [15]. The boids algorithm is a simple and elegant model for simulating the flocking behavior of birds. It is based on three simple rules: separation, alignment, and cohesion. The separation rule ensures that the drones maintain a safe distance from each other, the alignment rule ensures that the drones move in the same direction, and the cohesion rule ensures that the drones stay together as a group. By following these rules, the drones can navigate the environment safely and efficiently.

We have added that the alignment rule is modified to take into account the position of the human operator so the drones interact more with the human. This modification allows the drones to follow the human operator while maintaining a safe distance from them. The cohesion rule is also modified to ensure that the drones stay within a predefined area around the human operators. This modification allows the drones to move around the human operator without straying too far from them.

Conclusion (Phase 3)

~1 page

This section should briefly conclude the document. It should sum-up the project, the context, and the evaluations.

References

Here are the references in citation order.

[1] Jessica R. Cauchard and et al. 'Drone & Me: An Exploration Into Natural Human-Drone Interaction'. In: Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies (2015) (cited on page 2).

[2] X. Ju et al. 'Safe human-drone interaction in indoor environments'. In: International Conference on Human-Robot Interac 2017 (cited on page 2).

[3] Chien Nguyen, Pascal Fua, et al. 'FlyJacket: An upper body soft exoskeleton for immersive drone control'. In: IEEE International Conference on Robotics and Automation (ICRA). 2018 (cited on pages 2–4).

[4] L. Cai and et al. 'Human-drone interaction based on posture recognition with monocular vision'. In: Journal of Intelligent & Robotic Systems (2019) (cited on pages 2, 3).

[5] J. A. Preiss et al. 'CrazySwarm: A large nano-quadcopter swarm'. In: 2017 IEEE International Conference on Robotics and (2017) (cited on pages 2, 4).

[6] M. Rubenstein et al. 'Kilobot: A low-cost scalable robot system for collective behaviors'. In: 2012 IEEE International Confere (2012) (cited on page 2).

[7] David Saldana et al. 'ModQuad: The Flying Modular Structure that Self-Assembles in Midair'. en. In: 2018 IEEE International Conference on Robotics and Automation (ICRA). Brisbane, QLD: IEEE, May 2018, pp. 691–698. DOI: 10.1109/ICRA.2018.8461014. (Visited on 10/10/2024) (cited on pages 2, 4).

[8] J. L. Sanchez-Lopez et al. 'CrazyChoir: Artistic performance with multiple Crazyflie drones'. In: IEEE International Conference on Robotics and Automation (ICRA). 2022 (cited on page 2).

[9] James A. Preiss et al. 'Crazyswarm: A large nano-quadcopter swarm'. In: 2017 IEEE International Conference on Robotics IEEE. 2017, pp. 3299–3304 (cited on page 3).

[10] Bitcraze. CFLib - Python library for Crazyflie. https://www.bitcraze.io/documentation/repository/crazyflie-lib-python/master/. Accessed: 2024-10-10. 2020 (cited on page 3).

[11] Mediapipe Dev Team. 'Real-time pose estimation using Intel RealSense and MediaPipe for human-drone interaction'. In: (2020) (cited on pages 3, 4).

[12] Amine Tayebi. 'Gesture-based drone control with Leap Motion and Arduino'. In: International Journal of Advanced Resea (2020) (cited on page 3).

[13] Python Software Foundation. 'asyncio: Asynchronous I/O, event loop, coroutines and tasks'. In: (2022). Accessed: 2024-10-10 (cited on page 6).

[14] Google Research. MediaPipe: A Framework for Building Perception Pipelines. https://google.github.io/mediapipe/. Accessed: 2024-10-10. 2020 (cited on page 8).

[15] Craig W. Reynolds. 'Flocks, Herds, and Schools: A Distributed Behavioral Model'. In: Computer Graphics 21.4 (1987), pp. 25–34 (cited on page 9).