Strategic use of drone swarms in aerial firefighting: Technologies applied to prevention, detection and monitoring

End-of-degree project

Academic year 2024-2025

Bachelor's Degree in Aeronautical Management

By

Joan Segura Santori

Supervised by

Ender Çetin

School of Engineering

Sabadell, June 2025

El abajo firmante, Ender Çetin, director del Trabajo de Fin de Grado, profesor de la Escuela de Ingeniería de la UAB,

CERTIFICA:

Que el trabajo al que corresponde la presente memoria ha sido realizado bajo su dirección por

Joan Segura Santori

Y para que conste firma la presente en Sabadell, 26 de junio de 2025

CETIN ENDER - Y6196182L

Digitally signed by CETIN ENDER - Y6196182L
Date: 2025.06.26 14:14:11 +02'00'

Firmado: Ender Çetin

SUMMARY SHEET – BACHELOR'S DEGREE FINAL PROJECT OF THE SCHOOL OF ENGINEERING

Title of the final project:

Strategic Use of drone swarms in aerial firefighting: Technologies applied to prevention, detection and monitoring.

Ús estratègic d'eixams de drons per a l'extinció aèria d'incendis: Tecnologies aplicades a la prevenció, detecció i monitoratge.

Uso estratégico de enjambres de drones para la lucha aérea contra incendios: Tecnologías aplicadas a la prevención, detección y monitorización.

Author: Joan Segura Santori

Date: June 2025

Mentor: Ender Çetin

Degree: Bachelor's Degree in Aeronautical Management

Keywords:

- English: Drones, Drone Swarms, AirSim, wildfires, wildfire simulation, monitoring, behaviour patterns, Rule-Based Algortihms (RBA).
- Castellano: Drones, Enjambres de Drones, AirSim, incendios forestales, simulación de incendios, monitorización, patrones de comportamiento, Algoritmos basados en reglas (RBA).
- Català: Drons, Eixams de Drons, AirSim, incendis forestals, simulació d'incendis forestals, monitorització, patrons de comportament, Algorismes basats en regles (RBA).

Summary:

- English: The aim of this paper is to explore the capabilities of drone technology in the monitorization and wildfire extinction To do this, the analysis relies in tools such as drone swarms, simulation engines and rule-based algorithms (RBA) to find the most efficient and optimal approach in a given wildfire situation.
- Castellano: El objetivo de este trabajo es explorar las capacidades de la tecnología de drones en la monitorización y extinción de incendios forestales. Para ello, el análisis se basa en herramientas tales como los enjambres de drones, software de simulación y algoritmos basados en reglas (RBA) para encontrar el enfoque más eficiente y óptimo en cada situación específica de incendio forestal.
- Català: L'objectiu d'aquest treball és explorar les capacitats de la tecnologia de drones en la monitorització i l'extinció d'incendis forestals. Per això, l'anàlisi es basa en eines com els eixams de drones, el motors de simulació i algorismes basats en regles (RBA) per trobar l'enfocament més eficient i òptim en cada situació específica d'incendi forestal.

Abstract

Wildfires are an increasing threat to ecosystems, infrastructure, and even human lives, particularly in regions like Catalonia, where climatic and geographical conditions support their propagation. Despite technological advances in wildfire suppression, there are still critical limitations, like the high operational cost of aerial resources and the lack of real-time information during emergency situations. In response to these challenges, this paper explores the integration of autonomous drone swarms into existing firefighting strategies.

The main objective is to design, implement, and validate a set of decentralised operations for autonomous agents (drones) using rule-based algorithms (RBA). These aim to enhance wildfire detection, monitoring, and prevention. The proposed solution follows a behaviour-based architecture in which each agent acts independently, yet collectively contributes to the mission's success.

A simulation environment based on Unreal Engine and AirSim has been developed to test the behaviours under simulated wildfire conditions. Python is used to implement all control logic and operations coordination. The results show that the proposed system is capable of detecting and monitoring multiple fire sources autonomously, and adapting to different scenarios. Additionally, the cost-effectiveness and flexibility of drones makes them a valuable aid for conventional aerial means.

This work intends to make a contribution to the future integration of UAV (Unmanned Aerial Vehicle) or drone technology in wildfire operations by offering a scalable, autonomous, and data-driven alternative to the traditional response strategies.

Resum

Els incendis forestals representen una amenaça creixent per als ecosistemes, les infraestructures i la vida humana, especialment en regions com Catalunya, on les condicions climàtiques i geogràfiques en faciliten la propagació. Tot i els avenços tecnològics en l’extinció d’incendis, persisteixen limitacions crítiques—com ara el cost operatiu elevat dels recursos aeris i la manca d'informació en temps real durant les emergències. En resposta a aquests reptes, aquest treball explora la integració d’eixams de drons autònoms en les estratègies actuals de lluita contra incendis.

L’objectiu principal és dissenyar, implementar i validar un conjunt d’operacions descentralitzades per a agents aeris autònoms mitjançant algorismes basats en regles (RBA). Aquestes operacions tenen com a finalitat millorar la detecció, monitoratge i prevenció d’incendis forestals. La solució proposada segueix una arquitectura basada en comportaments, on cada agent actua de forma independent però col·lectivament contribueix a l’èxit de la missió.

S'ha desenvolupat un entorn de simulació amb Unreal Engine i AirSim per provar els comportaments en condicions simulades d'incendi. Tota la lògica de control i coordinació de les operacions s'ha implementat en Python. Els resultats demostren que el sistema proposat és capaç de detectar i monitorar múltiples focus d'incendi de manera autònoma i adaptar-se a diferents escenaris. A més, el baix cost i la flexibilitat dels drons els posicionen com un complement valuós als recursos aeris convencionals.

Aquest treball contribueix a la futura integració de la tecnologia UAV (Unmanned Aerial Vehicle) en operacions contra incendis, oferint una alternativa escalable, autònoma i basada en dades a les estratègies de resposta tradicionals.

Resumen

Los incendios forestales representan una amenaza creciente para los ecosistemas, las infraestructuras y la vida humana, especialmente en regiones como Cataluña, donde las condiciones climáticas y geográficas favorecen su propagación. A pesar de los avances tecnológicos en la extinción de incendios, persisten limitaciones críticas—como el elevado coste operativo de los recursos aéreos y la falta de información en tiempo real durante las emergencias. En respuesta a estos desafíos, este trabajo explora la integración de enjambres de drones autónomos en las estrategias actuales de lucha contra incendios.

El objetivo principal es diseñar, implementar y validar un conjunto de operaciones descentralizadas para agentes aéreos autónomos (drones) mediante algoritmos basados en reglas (RBA). Estas operaciones están orientadas a mejorar la detección, la monitorización y la prevención de incendios forestales. La solución propuesta sigue una arquitectura basada en comportamientos, en la que cada agente actúa de forma independiente, pero contribuye colectivamente al éxito de la misión.

Se ha desarrollado un entorno de simulación con Unreal Engine y AirSim para probar los comportamientos en condiciones simuladas de incendio. Toda la lógica de control y coordinación de las operaciones ha sido implementada en Python. Los resultados demuestran que el sistema propuesto es capaz de detectar y monitorizar múltiples focos de incendio de manera autónoma y adaptarse a distintos escenarios. Además, el bajo coste y la flexibilidad de los drones los posicionan como un complemento valioso a los recursos aéreos tradicionales.

Este trabajo pretende contribuir a la futura integración de la tecnología UAV (Unmanned Aerial Vehicle) o drones en operaciones contra incendios, ofreciendo una alternativa escalable, autónoma y basada en datos a las estrategias de respuesta convencionales.

Keywords

Drone swarm, wildfire detection, Agent-driven operation, Rule-Based Algortihms (RBA), aerial firefighting, autonomous agents, AirSim simulation, multi-agent coordination, monitoring, behaviour-based robotics, Python.

Paraules clau

Eixam de drons, detecció d'incendis forestals, operacions gestionades per agents autònoms, algoritmes basats en regles (RBA), extinció aèria d'incendis, agents autònoms, simulació AirSim, coordinació multiagent, monitoratge d'incendis, robòtica basada en comportaments, programació en Python.

Palabras clave

Enjambre de drones, detección de incendios forestales, operaciones gestionadas por agentes autónomos, algoritmos basados en reglas (RBA), extinción aérea de incendios, agentes autónomos, simulación en AirSim, coordinación multiagente, monitorización de incendios, robótica basada en comportamientos, programación en Python.

Table of contents

List of figures

1. Introduction

Every year, wildfires destroy property and landscapes, and increasingly impact society. Catalonia is especially affected by this issue due to its specific characteristics, dry vegetation, big plain extensions in the central depression and mostly warm temperatures all year, plus the wind factor whose intensity is variable but present all year along as well.

Over the past decade, Catalonia has experienced significant fluctuations in wildfire activity, influenced by climatic conditions, land management practices, and preventive measures.

In 2024, Catalonia experienced one of its mildest wildfire seasons in 15 years. Between June and August, approximately 500 hectares burned, the lowest since 2020. This success was largely attributed to early summer rains and proactive prevention strategies. However, this data is unusual, as the historic data trends show that forest fires are a major problem.

In the past decades, Catalonia has experienced significant wildfires, with variations in the number of fires and the area burned over the years. The annual average is between 400 and 600 wildfires per year and the burned area ranges from 1,000 to 3,000 hectares annually, with peaks in critical years.

The most severe recent years are:

1994: Over 60,000 hectares burned. This fire can be seen in Figure 1.1 as a big red circle in the centre area, and in more detail in Figure 1.2 showing the vast area ravaged by the wildfire.

2012: Around 10,000 hectares were affected by fires in Alt Empordà.

2022: 5,800 hectares burned, one of the worst recent years.

The current trend shows fewer fires but with an increased latent risk due to more frequent droughts and heat waves. This shows which shows a positive evolution at fire prevention, but wildfires still happen and emergency services have to respond.

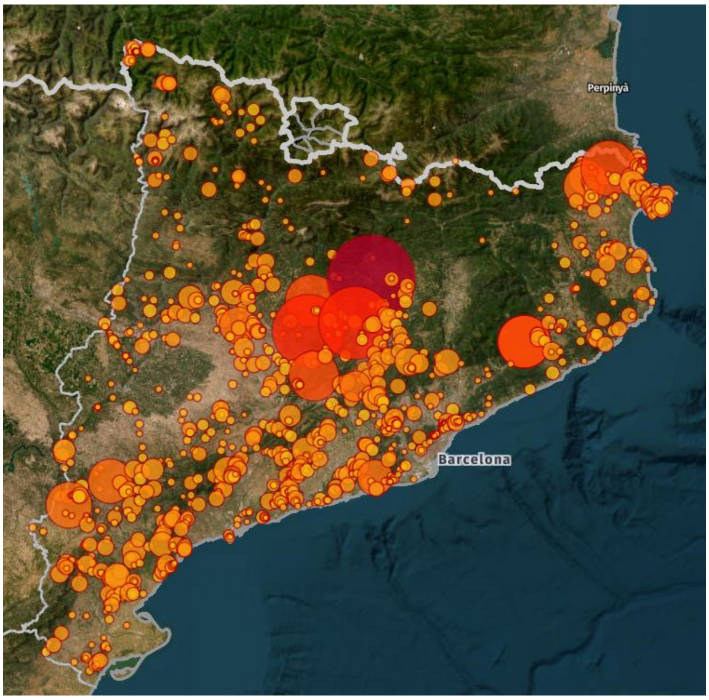

Figure 1.1 *Main Wildfires in Catalonia in the Last 40 Years (by Burnt Area) ICGC. Source: [1].*

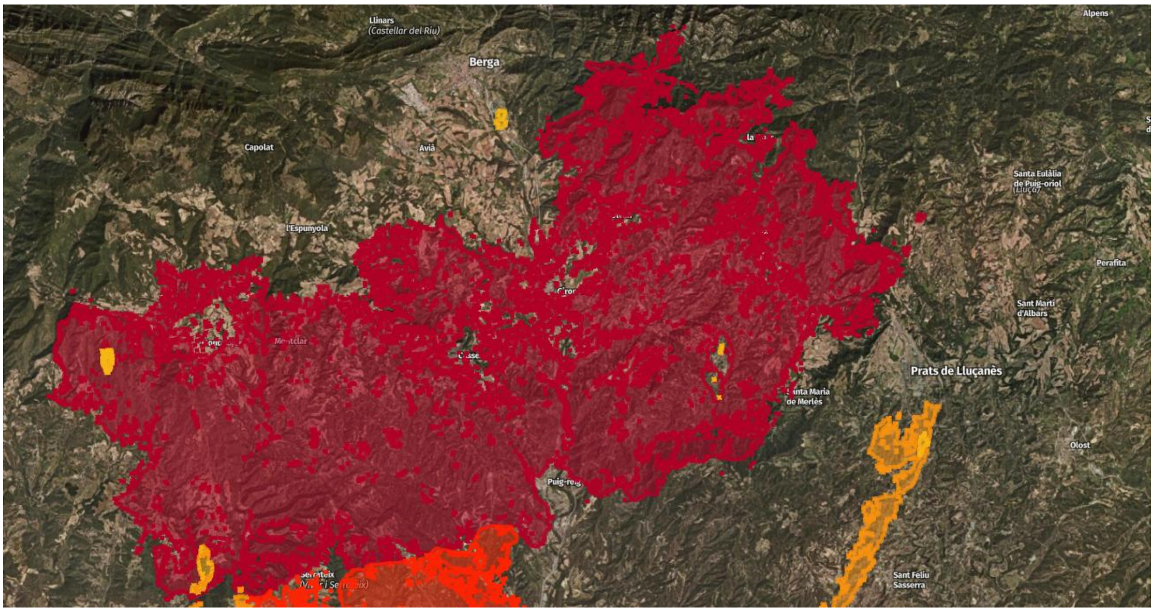

Figure 1.2 *Casserres Wildfire 07/04/1994 – 227.393 km2 (Affected area). ICGC. Source: [1].*

What can be seen in Figure 1.2 is unusual, but it is also the combination of different aggravating factors at a specific moment in time that multiplies the impact of the fire. When something like this happens the result is what this image shows.

I am using this this specific fire as an example, because one of the pilots I met in my professional career participated in the extinction operations. According to testimony, the impact could have been significantly reduced if elements such as drones had supported the monitoring of the fire, because a serious problem was the uncontrolled spread of the fire. Secondary fires were appearing due to wind and vegetation with no one being able to locate and report them until they became uncontainable.

When these events took place, the technology related to drones had not yet been developed. Let us consider how much vegetation, homes, and even lives could have been spared and what an exceptional work could have been done in extinguishing that fire.

The motivation for this work, therefore, comes from the fact that on the day that something like this happens again, we will be able to respond in the most precise and effective way possible, thus avoiding a repetition of the events of 1994 in Casserres.

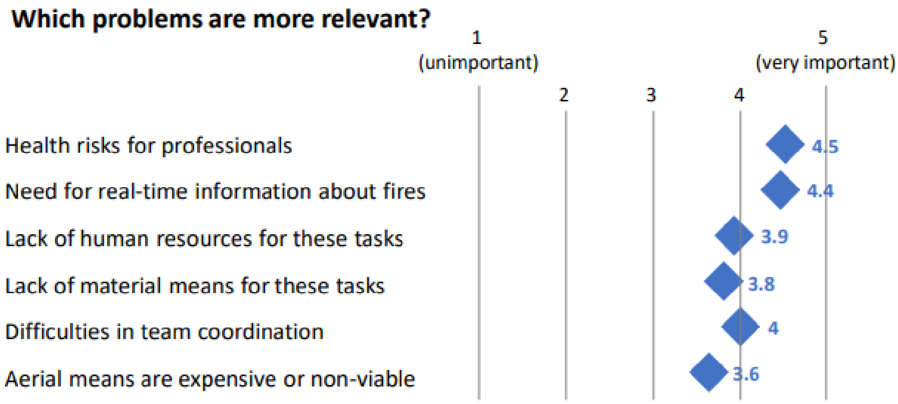

Figure 1.3 *Relevance of problems in extinguishing tasks. Source [2].*

Finally, to prove the real need for drones, we can refer to Figure 1.3, a graph extracted from a study that analyses the current problems in forest firefighting missions and the potential of robotic technologies to address them. The authors surveyed firefighters to identify the main difficulties they face in their work and possible technological solutions.

Some of the primary concerns are the lack of real-time information about fire development and challenges in team coordination. Therefore, it is obvious that drones are an excellent tool to mitigate these problems, given their versatility, manoeuvrability and speed. They could assist in providing real-time fire information by tracking fire progress and gathering information from a satellite point of view. This impacts directly and positively in the second major problem, that is, coordination between teams. If there is an accurate and timely knowledge of the state of the fire, the response strategy much more precise, easier and safer to deploy. It removes the risks of guess working and decision-making based on uncertainty.

1.1. Motivation

While beginning this Bachelor’s degree, I thought that it was mainly related to commercial aviation. As I progressed, I learned that it covers a much wider field of aviation applications. One of these is the utilisation of aerial means in emergency services. Thanks to this, I discovered the potential application of this knowledge in firefighting operations.

Parallel to my growing interest, I had the opportunity to work in a company named Titan Firefighting, based in the Sabadell Airport (LELL). The company's mission is to support terrestrial means from air-to-ground water and retardant drops and fire monitoring, making use of the Air Tractor AT-802 aircraft. I served as a Ground Operator. During my time there, I had the opportunity to speak with various experienced pilots who had years of service and had operated in countless wildfire missions. In one of these conversations, the topic of drones came up and how they could assist very effectively in firefighting efforts. We also discussed how, to this day, drones are not yet integrated into wildfire suppression services in the autonomous community of Catalonia.

I think that developing an approach on how to integrate this technology into MAER (Fire brigade air resources) would be pioneering work that could be taken to a real-world application and provide a very important service to society and the territory.

1.2. Goals

The objective of this work is to develop the implementation process for the use of drones within the existing current strategies and resources, study their impact on the terrain, understand how the use of algorithmic solutions can take their efficiency to another level, and finally, develop a simulation in which these topics can be visualised. To achieve this, I have set the following key points that lead up to the completion of this paper.

1. Outline the sequence of events from the moment a fire breaks out, through the activation and deployment of emergency services, to the actions they take upon arrival at the scene. Then, analyse the optimal integration of the drone swarm within this timeline and define the tasks and actions it should undertake to monitor the fire in real time and reduce the operational workload.
2. Include multimedia files showcasing the impact of a wildfire and providing insight into how it spreads. Additionally, incorporate knowledge acquired from firsthand testimonies of pilots and firefighters describing their experiences on the ground exploring how drones could have supported the operation in that specific case and potentially altered the outcome.
3. Gather and present the different technologies and initiatives related to the use of drones in the fire emergency field. Assess the state of development and application of algorithmic solutions and contribute to the development and understanding of these by working on a simulation that replicates the behaviours of such agents. The objective is to analyse whether these could potentially be applied in real-world scenarios.

1.3. Timeline

The following Gantt chart describes the timeline and steps of this project:

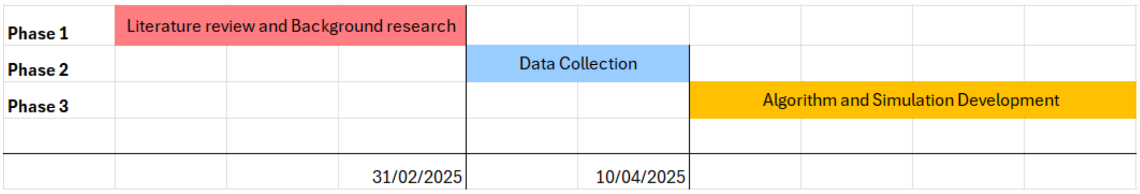

Figure 1.4 a *Gantt chart*

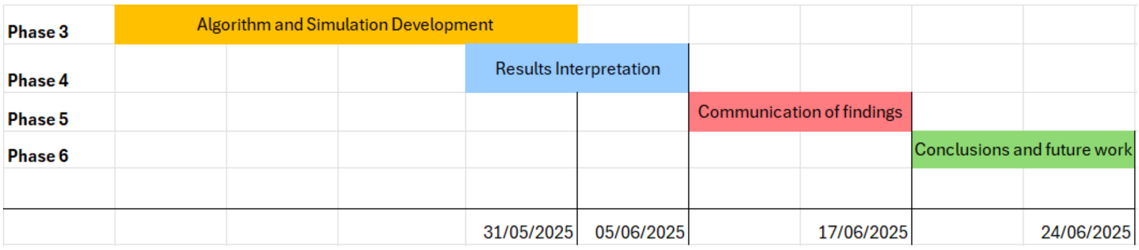

Figure 1.4 b *Gantt chart. Source: Author*

1.4. Content & Methodology

To achieve the objectives for the development of this work, I will follow the methodology detailed in the following chart with the detail of the sequence of events.

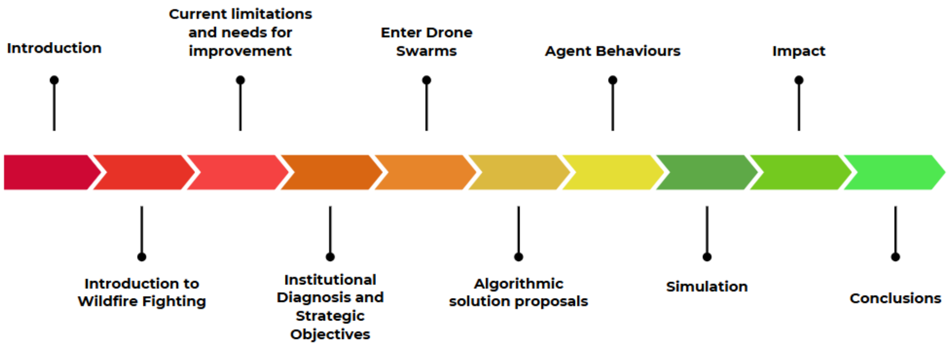

Figure 1.5 *Chronology of development and event sequence. Source: Author*

1.5. Users

The target audience of this study are the following groups:

- **Firefighters and emergency services personnel**: This paper aims to provide tools and insights that can help firefighters and other emergency responders in managing wildfires more efficiently by integrating drone swarms and RL for real-time monitoring and decision-making support.
- **Researchers and academics in the fields of AI, wildfires, and robotics**: This work is directed at those interested in applying algorithms to autonomous systems, especially in high-stakes scenarios like wildfire suppression, providing a foundation for future research and development in these areas.
- **Developers and engineers in drone technology**: It also targets professionals working in drone development and deployment, offering them a perspective on how Rule-Based Algortihms and swarm robotics can optimize the use of drones in emergency situations.
- **Policy Makers and authorities in disaster management**: This project provides valuable insights into the potential of drone technology for improving disaster response and management strategies, helping policymakers make decisions on integrating such technologies into their emergency plans.
- **Technology enthusiasts and peers**: The project serves as an educational resource for students and technology followers, particularly those with an interest in drones, AI, and disaster management, offering them an introduction to the practical applications of RBA in critical real-world situations.

1.6. Risks

This paper depends on several factors, such as the ability to abstract information from external sources, availability of reliable data and the use of programming tools to emulate real-life scenarios. Therefore, the risks identified are detailed below:

- **Technical Feasibility**: AI-driven drone coordination, satellite data may face hardware or software limitations, affecting real-world deployment.
- **Operational Constraints**: Integration of autonomous drones with manned aircraft could pose regulatory and safety challenges, such as data governance, hacking and hazardous situations.
- **Data Availability**: Limited access to real-time wildfire data may impact the accuracy of simulations.
- **Environmental Factors**: The unpredictability of fire behaviour may challenge the calculated predictions and response planning, which also means precise predictive RBA could be hard to implement.

2. Fundamental principles of wildfires and spread dynamics

2.1. Context

Extreme wildfires are an increasing global threat. Climate change is leading to drier and hotter summers, thereby extending the wildfire season. For instance, the wildfire prevention campaign activated in the autonomous community of Catalonia in 2024 was extended by 15 days compared to previous periods, running from 1 June to 30 September.

Catalonia is particularly vulnerable to this new reality, as it has an extensive forested area that is increasingly abandoned and stressed due to a lack of rainfall and rising temperatures. The high density of trees (Figure 2.1) and undergrowth (Figure 2.2) increases the amount of available fuel, making wildfires more intense and fast-spreading.

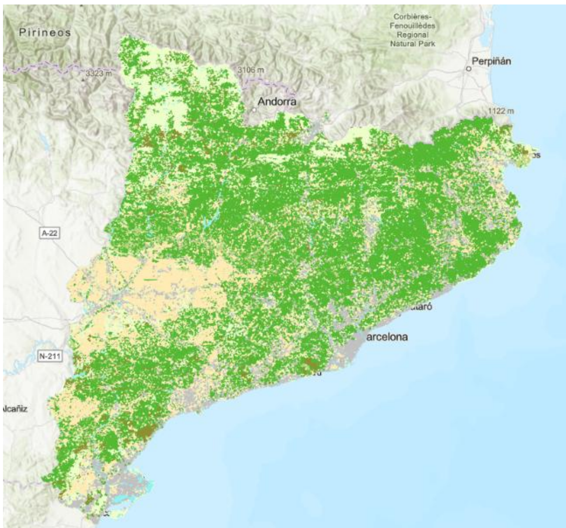

Figure 2.1 *Forest Map of Catalonia. Source [3]*

Figure 2.2 *Forest undergrowth. Source [4]*

2.2. Wildfires: Basic concepts

Wildfires are uncontrolled fires that spread across vegetated areas affecting forests grasslands and other natural zones. They can be ignited by natural causes such as lightning or by human activities, including negligence and arson. Their behaviour depends on three main factors:

1. Fuel, the quantity, type of vegetation, its distribution and moisture level.
2. Weather conditions including wind, temperature, humidity and precipitation.
3. Topography, the slope and characteristics of the terrain.

Irrespective of the cause of ignition, wildfires require three key components to ignite: fuel, heat and oxygen. These are commonly referred to as *the "fire triangle"*, according to Professor Joseph Roise (North Carolina State University) in one of his publications of reference in the topic (refer to [1]).

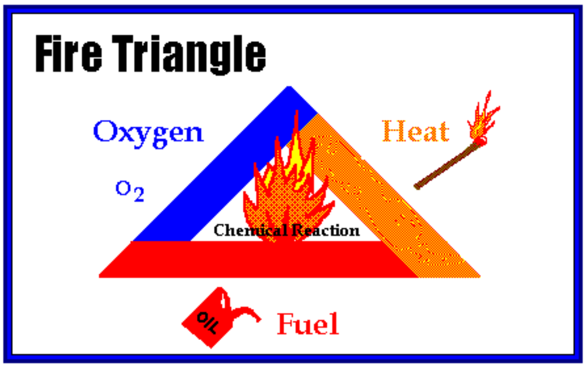

Figure 2.3 *Fire Triangle. Source [5]*

Although wildfires can happen at any time of the year, the risk increases when an area experiences increasingly dry and warm conditions, such as drought in the warmest seasons (see the comparison of Figure 2.4 and Figure 2.5).

Drought conditions are sometimes preceded by a period of above-average rainfall, which favours and accelerated growth of plants and trees. This vegetation, however, usually wilts and becomes dried out when drought conditions occur, reducing moisture content and becoming fuel for wildfires (Figure 2.3).

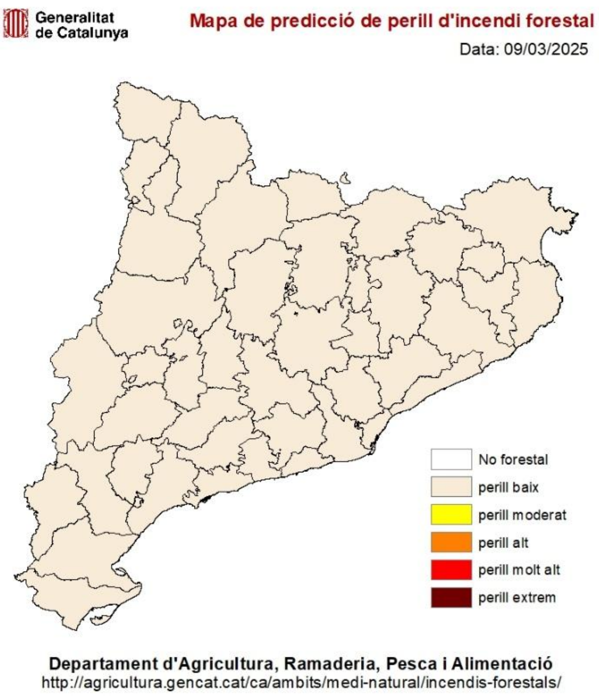

Figure 2.4 *Wildfire risk prediction on 09/03/2025. Source [6]*

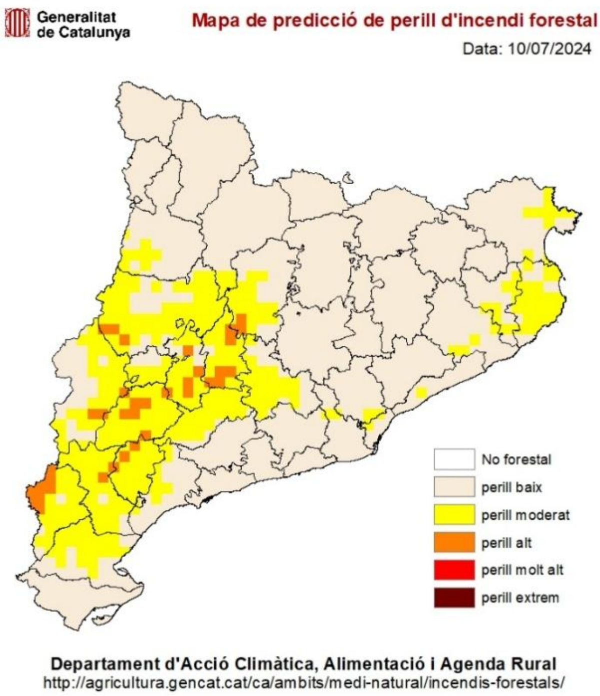

Figure 2.5 *Wildfire risk prediction on 10/07/2024. Source [6]*

As previously detailed, wildfires spread on the basis three factors: vegetation type, weather conditions, and topography. In order to fully comprehend the subsequent sections of this study, it is essential to have a clear understanding of the principles and mechanisms governing wildfire propagation, as the technology we seek to implement is fundamentally based on this phenomenon.

With this in mind: wildfires tend to spread in three distinct ways, depending on the combination of the factors mentioned above:

- **Surface fires** burn low vegetation and travel along the ground.
- **Crown fires** jump from treetop to treetop, moving rapidly and intensely.
- **Spot fires** occur when embers are carried by the wind and ignite new fires far from the original blaze.

The modelling capabilities of the Fire Dynamo tool [2] utilises agent-based modelling to simulate wildfires by i) analysing individual elements such as trees and shrubs, ii) integrating satellite remote sensing data to assess vegetation flammability and iii) considering factors such as topography, fuel availability, and weather conditions. As a result, below there is a clear example of fire propagation:

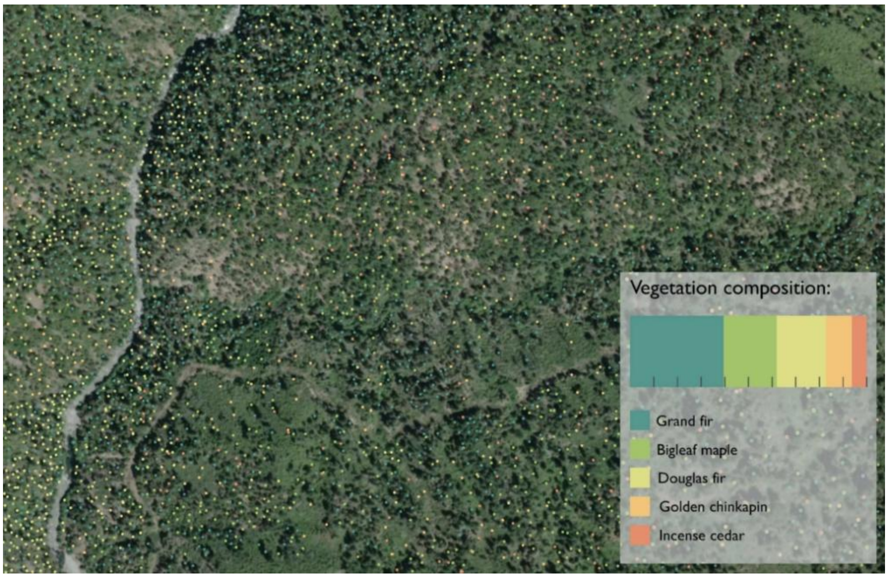

Figure 2.6 *Vegetation map based on satellite imagery. Source [7]*

This first image is a satellite photograph of a forested area. It provides a realistic, overhead view of the terrain and vegetation. The image shows a mix of dense forest cover with varying shades of green, suggesting different types of vegetation or stages of growth.

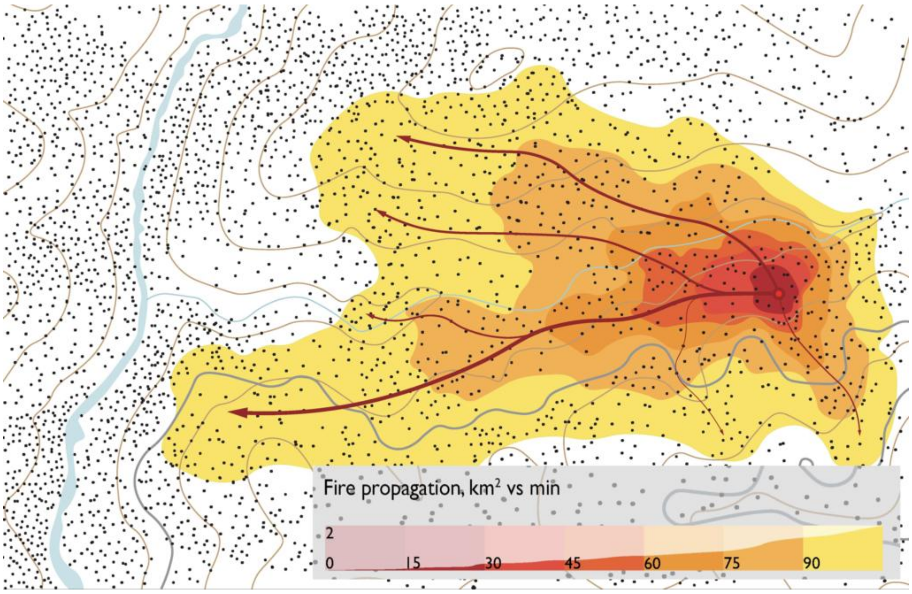

Figure 2.7 *Estimations of fire progression using agent-based simulations. Source [7]*

On the other hand, this image presents a simplified, schematic view of fire propagation over time. It utilizes a colour gradient, ranging from light yellow to dark red, to visually represent the spread of fire in square kilometres (km^2) over a period of 90 minutes.

The fire's path is indicated by arrows, suggesting direction and progression, also over a topographical map.

The above simulation (Figure 2.7) shows fast fire spread towards the upper-left quadrant between the 30 and 45-minute marks, and the photograph (Figure 2.6) reveals a dense stand of Douglas fir (a species often considered highly flammable) in that same general direction. It can also be seen where the simulation indicates a slower rate of spread in an area, and the photograph shows a clearing with less vegetation or a higher concentration of Bigleaf maple (which might be less flammable) in that same area.

Additionally, by comparing the topographical features represented by contour lines in the simulation to the visual terrain in the photograph, we can see how the simulation model incorporates the influence of terrain. The simulation shows the fire spreading faster uphill, and the photograph confirms, indeed, an upward slope in that direction.

Finally, the simulation considers wind coming from southwest or west-southwest influencing the fire rapid progression.

3. Drone swarms: An overview

3.1. Introduction

In the recent years, similar initiatives and proposals have been developed and some innovations resemble the approach presented in this project. For this reason, this section will provide a state-of-the-art review with the aim of identifying existing technological solutions, relevant contributions and contextualising the proposed system within the autonomous drone field. Accordingly, below there are the related topics covered:

- The cooperation of autonomous agents within a single mission.
- The current state of the field regarding the use of autonomous drone agents for large-scale solutions.
- The use of drones in wildfire monitoring and suppression tasks.

3.2. Cooperation of autonomous agents

Autonomous drone operations have evolved from single, remote-controlled platforms to coordinated swarms that work together in complex operations. *"These multi-UAV (Unmanned Aerial Vehicle) systems rely on distributed coordination, allowing each drone to contribute to a collective mission such as surveillance, search-and-rescue, or environmental monitoring without requiring centralized control"* [24].

The cooperation between autonomous agents is founded on several core principles that ensure their effectiveness in complex, dynamic environments. One of the most crucial elements is **decentralisation**. By enabling each agent to make decisions independently, the system acquires reliability, scalability, and flexibility: qualities that are essential for missions involving large-scale or unpredictable scenarios.

Another key factor is the **balance between heterogeneity and homogeneity**. While agents may be different in their capabilities (heterogeneity), they must also share enough standardised features (homogeneity) to work together efficiently. Managing this balance is essential to ensure both the scalability and reusability of the system across different applications and operational contexts. This feature basically defines the adaptability of these cooperative agents to different scenarios. In a real-life scenario, there may be a variety of models (with different capabilities) to be integrated in a single swarm, so it is essential that they are capable to integrate without disruption.

To enable smooth cooperation, agents should implement **standardised interfaces** understood as a "common language". These interfaces allow different agents, regardless of their specific functions or origins, to interact seamlessly. For example, this standardisation allows to combine simpler functionalities into more complex behaviours (functionality aggregation), and support dynamic collaboration as missions evolve (collaboration support).

Effective cooperation also depends on **communication and visibility**. Agents must be able to identify and understand the roles of their collaborators, using mechanisms such as **leader/follower** formation. **Concurrency and resource management** are essential for multi-agent operations, through a model that defines roles such as leader, collaborator, and shared collaboration objects. In this manner, *"[...] drones are able to perform multiple tasks simultaneously, distribute workload, and share resources."* [25]

3.3. Autonomous drone agents

Unmanned Aerial Vehicles (UAVs), also known as drones, are aerial systems typically operated remotely by a human. They can also be designed with autonomous capabilities, making them very useful for carrying out tasks beyond the visual line of sight.

"A drone swarm is a unified system made up of multiple drones working together to accomplish a mission. However, it is not just a simple grouping of individual drones. It also involves various structural elements such as data links between the drones, a mission network, and a ground control platform, as illustrated in Figure 3.1" Source: (Zhang et al., 2022, p. 2) [26]

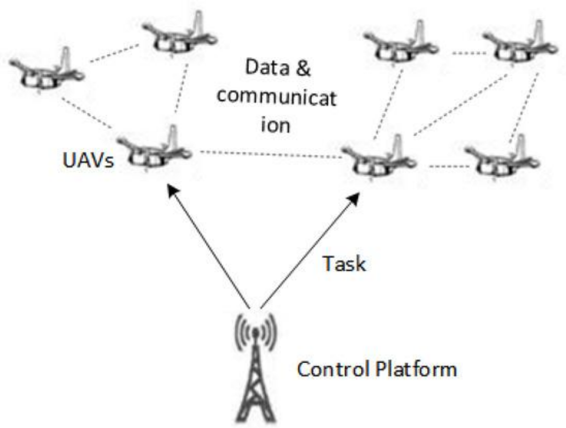

Figure 3.1 *Composition of drone swarm. Source [8]*

There are several factors involved in the ability of drones to navigate autonomously such as path planning, trajectory generation and behaviour design. The result of this integration is a system capable of providing large scale scenarios solutions and completing complex missions with minimal human intervention.

Autonomy in drones can be categorised into three levels:

1. Sensorimotor autonomy allows the drone to execute basic commands such as predefined tasks or following preprogrammed paths.

2. Reactive autonomy enables obstacle avoidance, safe distance maintenance, and coordination with other drones.
3. Cognitive autonomy involves environment mapping, route planning, and learning from experience. This paper will focus on the first level.

The ability to deploy drones at a large scale, provides many advantages across various sectors. For example, in agriculture, they enable continuous monitoring of crop growth using specialised cameras and cloud-based data analysis [14]. For rescue and emergency operations, they can enhance security by monitoring areas beyond the reach of traditional surveillance systems, contribute to disaster mitigation through the inspection of damaged buildings or inaccessible areas even support telecommunications by deploying temporary communication networks in remote or disaster-affected locations [21].

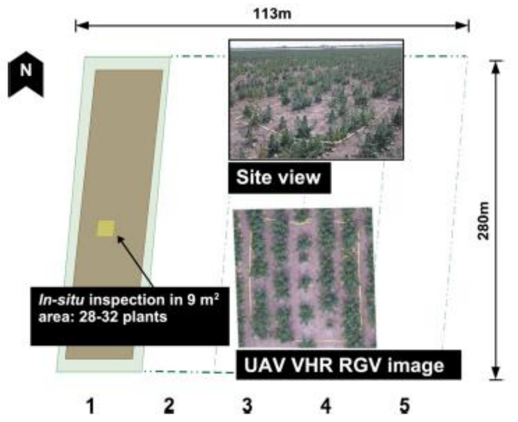

Figure 3.2 *Small scale experimental setup for counting crops in an area of 9 m2. Source [9]*

Figure 3.3 *Drone flying over affected area during the DANA floodings in Valencia. Source [10]*

These agents not only provide quality visual information of a large area but can also enable automated fire perimeter mapping and other tasks that are crucial for effective resource deployment and strategic firefighting. This paper will focus on these applications.

3.4. Drones applied to wildfire solutions

The rapid development of UAV solutions has led to innovative systems designed to overcome traditional limitations in wildfire detection and monitoring. Conventional techniques—such as satellite imaging, static sensor networks, aircrafts, etc.— are negatively affected by latency, insufficient resolution, and poor area coverage of wildfires.

To address these drawbacks, some state-of-the-art solutions have been developed. One of the most remarkable advancements are the drones equipped with thermal sensors and cameras, combined with artificial intelligence models capable of recognising signs of fire. These drones can detect and locate a fire autonomously and very quickly, even within the first few minutes after it starts. In addition to detecting fire, they also send real-time environmental data, essential for decision-making. In one such implementation, fire was detected within 1–5 minutes of ignition (Figure 3.4), and the AI model achieved an accuracy of 99.46% [20].

Figure 3.4 *Demonstration of the fire tracking algorithm where the particles are initially detected within the FOV. Source [11]*

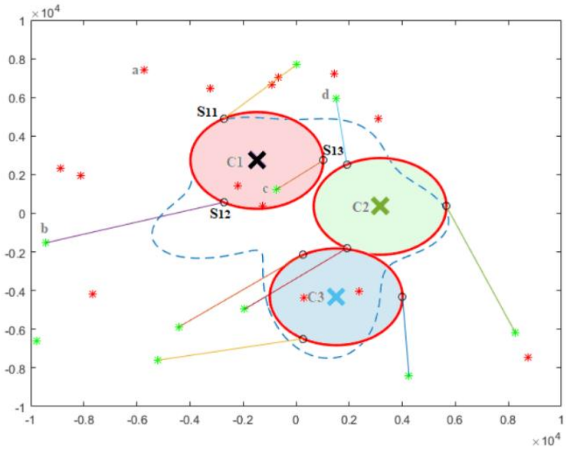

Figure 3.5 *Fire monitoring in remote areas. Source [11]*

Another domain of research focuses on decentralised multi-UAV coordination using leader-follower coordination models. This approach enables the efficient partitioning of large areas without relying on centralised control or constant communication between drones. Drones equipped with heterogeneous sensors execute in collaboration perimeter mapping and dynamic path planning. Such configurations ensure full coverage of the fire zone, reduce latency in data acquisition, and ability to adapt to changing fire conditions [21]. Figure 3.5 provides a visual example of how drones can be grouped effectively to cover as much area as possible. The main (leader) drone assigns positions to the other drones by considering how long it would take each one to reach the target locations and how much battery they have left. Once in place, the drones fly in a circular pattern to capture images of the area.

Complementing these advances, fully autonomous UAVs using onboard computing have also shown effectiveness in real-time surveillance. These platforms integrate AI algorithms capable of fire detection, object classification, and anomaly detection on the spot. *"Combined with thermal and visual sensors, they can provide early warnings in under 30 seconds during flight operations across risk zones, with coverage capabilities surpassing static watchtowers and detection accuracy superior to satellite imaging"* [22].

Finally, in terms of active intervention, there have been some trials with drone-carried fire-extinguishing payloads. For example, fire-extinguishing balls and small thermally activated suppressant devices have been mounted on drones capable of autonomous navigation [23]. While initial tests (Figure 3.6) demonstrated limited efficacy in large-scale structural fires, results indicated potential usefulness in controlling short grass wildfires under defined conditions.

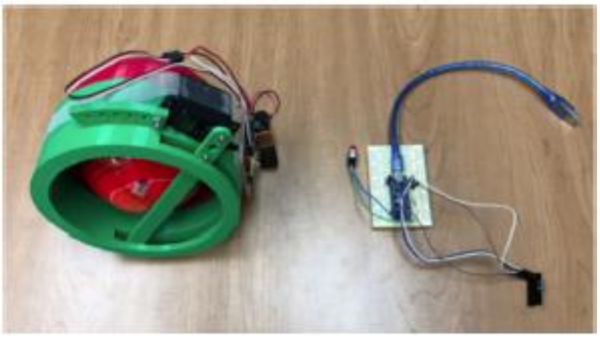

Figure 3.6 *Prototype of a ball-release mechanism (left) and remote starter circuit (right), both attached to the UAV. Source: [12]*

4. Current means: limitations and elements for improvement

4.1. Introduction

Once there is a clear understanding of the wildfire essentials and propagation fundamentals, it is now essential to assess the current state of wildfire suppression. Given that this analysis aims to introduce drone swarms to fight fires, it is necessary to identify the shortcomings and needs of the existing wildfire management models.

To achieve this, this section will focus on gathering and analysing information published by the Spanish Ministry of Agriculture, Fisheries and Food, as well as other public sources, including research studies in this field. The objective is to present the current state and key challenges to the proposed to our proposed technological solutions, providing a clear framework for their application.

4.2. Aerial firefighting initiatives and costs in Spain

Given that drones are an aerial resource, it is necessary to examine the use of aerial firefighting assets in Spain, considering their activation periods and measuring their use in terms of flight hours. To do so, we can refer to the following graph, which illustrates the relationship between the flight hours of aircraft operated by the General State Administration and the total number of interventions related to wildfires.

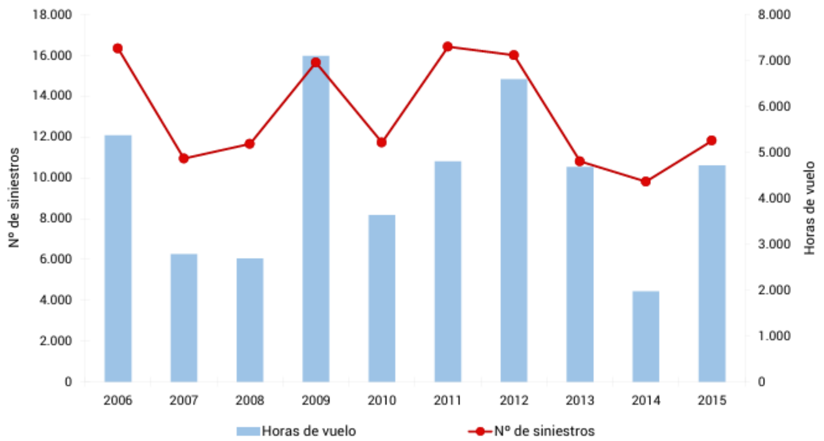

Figure 4.1 *Relationship Between light Hours of Aircraft Operated by the General State Administration and the Total Number of Incidents, 2006-2015. Source [13]*

Aerial resources are highly valuable and essential, particularly when a wildfire cannot be contained by ground units alone. In such cases, aerial assets provide crucial support, both in terms of fire suppression and the coordination of firefighting operations. This is evidenced by the figure above, which illustrates a general trend whereby an increase in the number of wildfire incidents is accompanied by a corresponding rise in flight hours, reflecting the intensified aerial effort required to combat these emergencies effectively.

However, it is very important to pay attention to the following extract from [3]:
"This technical advantage (aerial means introduced in 1991) has led to a progressive increase since then, reaching such significant levels that Spain is now one of the leading countries in the world in terms of aerial resources used per surface area. However, despite its potential, its use is hindered by high operating costs, as a result of which the economic situation of the administrations has been a decisive factor in the increase or reduction of their assets."

Currently in Catalonia, the firefighting brigades ("Bombers de la Generalitat") [17] use the AS350B3 helicopter. This helicopter is used both for extinguishing (water discharges) and coordination tasks. The operational cost calculated by ACC (Aircraft Cost Calculator) [4] is broken down as follows:
"Based on 300 annual owner-operated hours and EUR6 - per - GAL fuel cost, the airbus H125/Eurocopter AS350B3e has total variable costs of EUR265,832, total fixed costs of EUR441,174, and an annual budget of EUR707,006. This breaks down to EUR2,357 per hour." [4].

The high operational cost of these assets limits their use to situations of extreme necessity and hinders their deployment on a regular basis. In contrast, the operational cost of drones is significantly lower, allowing them to be deployed even in response to minor incidents, becoming a more cost-effective alternative.

4.3. Institutional objectives aligned with this paper

The Spanish's Ministry for Ecological Transition and the Demographic Challenge publication *"Strategic Guidelines for the Management of Forest Fires in Spain, 2022"* [5] establishes several key objectives to enhance wildfire prevention and suppression strategies. Some of these objectives align closely with the integration of autonomous drone swarms, namely:

- *"3.3.1 Objective 4: Adapting Firefighting Capabilities to New Scenarios and Enhancing Coordination; Objectives – Specific Objectives".* Objective no. 4, page 10.
  Climate change is leading to increasingly severe wildfire conditions, often beyond conventional existing capacities. The Ministry highlights the need of strategic planning, decision-making improvements, and proactive operational strategies to tackle this issue. Use of drone swarms and its advantages such as collected information, experience, etc. would help to stay one step ahead of the fires.

- *"Optimisation of Aerial Resources".* Point no. 5, page 10.
  Spain possesses a considerable fleet of aerial firefighting assets, yet their maintenance and operational costs and maintenance requirements are high. The Ministry emphasises the need for

an efficient allocation of these resources based on risk, financial feasibility, and operational effectiveness. Additionally, it emphasizes the need to update and adapt this fleet to new technologies in the long-term. Drone swarms offer a cost-effective complement to traditional aerial assets, enabling continuous monitoring without the financial and logistical burdens associated with manned aircraft. Their rapid deployment and adaptability make them particularly valuable for early-stage fire detection and continuous situational assessment.

- *"Enhancing Information Exchange Among Agencies".* Point no. 9 page 11.
  "It is advisable to enhance information exchange between administrations regarding wildfire occurrences, the allocation of firefighting resources, and the tracking of their positions, as a means to strengthen operational safety and improve intervention efficiency…"
  Drone swarms, equipped with real-time data transmission capabilities, facilitate seamless information exchange between firefighting teams, command centres, and government agencies.

- *"Understanding Fire Behaviour and Prediction".* Point no. 12, page 11.
  "*The interpretation of wildfire behaviour and the ability to predict its evolution are essential for working safely and efficiently, as well as for planning and adapting firefighting and control tactics*."
  "[…] *It is necessary to continue integrating new technologies into firefighting operations to enhance effectiveness and response capabilities […]*"
  Using algorithms enable drone swarms to analyse environmental variables, model fire spread patterns, and suggest optimal suppression strategies. This predictive capability enhances both immediate tactical responses and long-term strategic planning for wildfire mitigation.

- *"Objective 5: Advancing Knowledge of Wildfires and Their Impact on Ecosystems; Objectives – Specific Objectives".* Objective no. 5, page 12.
  Wildfires are complex events influenced by multiple factors associated with environmental changes. The Ministry stresses the importance of research, innovation, development, and knowledge dissemination to improve wildfire understanding and developing new management tools.

- *"Technological Enhancements in Fire Management"*. Point no. 6, page 13.
  "*In the technological field, it is essential to continue investing in improvements in certain areas, such as fire behaviour simulation, risk indices associated with forest variables, fleet tracking, airborne elements, the analysis of large historical data sets (Big Data) ...*"
  Drone swarms can enhance these areas, they should be seen as a real option to address the related requirements in the future.

5. Design and development

5.1. Introduction to the project approach

So far, this paper has covered the fundamentals of wildfires, the shortcomings of current prevention and suppression resources, and the objectives set by the national agency to improve them. It is time now to present the proposed solutions and tools offered by drone swarms enhanced with Rule-Based Algorithms, and how they can be implemented to address the challenges ahead.

This project considers the potential use of this technology for the following major operational tasks in wildfire suppression: **Prevention and Surveillance, Search and Detection and less extensively, Monitoring**. The extinguishing task is excluded: as of today, no drone-based technology can perform this task. While there are ongoing projects such as [6], none have achieved conclusive results, only experimental results, as described in section 3.4. For this reason, this aspect remains an open field for future research and development, as stated in the final conclusions.

Drone swarms adopt the movement mechanics of what could resemble a flock of birds in migration. They consist of a homogeneous group of drones—where different drone models could be combined. This analysis focuses on individual units working together to perform the same task. The emphasis is placed on cooperation between agents to complete any given mission, rather than adapting specialised drones to specific tasks.

5.2. Simulation Resources

This section outlines the key resources and tools employed in the development of the control logic and the simulation environment.

5.2.1 Unreal Engine: AirSim

Unreal Engine is a game engine developed by Epic Games. Apart from the video game industry, it also has powerful applications in other fields such as architecture, simulation, and scientific research due to its powerful rendering engine, advanced physics, and compatibility with other technologies. In this project I have used Unreal Engine 4.27 because it is fully compatible with the AirSim simulator. This allowed to create a visual and immersive environment to simulate the behaviour of autonomous drones in wildfire scenarios.

AirSim is an open-source, cross-platform simulator developed by Microsoft Research. It allows to design realistic simulations of autonomous of autonomous vehicles like drones and cars. It supports the development and testing of control, navigation, and perception algorithms, and provides an API* in languages such as Python and C++.

* *Application Programming Interface* is a connection protocol between different programs.

The integration of AirSim within Unreal Engine facilitates the creation of high-fidelity 3D environments, with adjustable parameters including wind, gravity, and collisions. Custom models can be imported to represent complex natural terrains, forested areas, and objects such as fire markers or obstacles. This setup enables for the accurate simulation of drone flight and takes into account environmental factors. It has allowed me to develop different behavioural models.

In this work, the behaviour of the agents has been developed using AirSim's Python API. This interface allows for full control of the drones, access to sensor data, trajectory planning, and reactive behaviours like fire detection or obstacle identification. The simulation is configured to operate with multiple drones in parallel, each controlled by independent scripts and capable of sharing information and coordinating their actions.

One of the key advantages of this platform is the ability to simulate complex scenarios and to replay simulations under identical or different conditions. This significantly enhances the evaluation of agent performance. Furthermore, the flexibility to quickly modify scenes and adjust flight or detection parameters allows to easily run multiple iterations in the process.

In summary, the combination of Unreal Engine 4.27 and AirSim offers a solid, flexible, and reliable platform for the development and validation of autonomous drone behaviours in realistic environments.

5.2.2 Python

Python was chosen as the primary development tool for this project not only due to its flexibility and broad applicability in robotics and artificial intelligence, but also because it is a language in which I have received formal training throughout my academic programme. Over the course of my studies, I have acquired solid experience working with Python in various contexts. This background has facilitated the implementation of autonomous drone behaviours within the AirSim simulation framework. In particular, Python's compatibility with the AirSim API enabled real-time control of drone agents, efficient access to sensor data, and structured programming of behaviour-based logic. As a result, Python provided a familiar, powerful, and well-supported environment for addressing the technical challenges of this work.

5.2.3 PyCharm

PyCharm is a professional Integrated Development Environment IDE used to write manage and execute the Python code. Throughout the elaboration of this paper, PyCharm has been the main platform in which I have learned to develop and deploy Python-based projects. It has a user-friendly interface and its structure allows to easily organize simulations that include several scripts and components.

In this project, PyCharm has served as the central environment for launching and controlling the behaviour of drone agents within the AirSim simulation. From this interface, I run the main scripts that connect to the simulated environment and command the drones in real time. By simply pressing a run button or executing selected scripts, I can start the simulation, observe the system's behaviour, and make adjustments as needed. The layout of the console and built-in tools makes it easy to follow the simulation's output, detect issues, and improve the code progressively.

5.3. Drones

As shown in Figure 5.1, various drone models are currently employed by the Fire Department of the Generalitat de Catalunya. Their operation falls under the GROS CER (Grup Operatiu de Suport dels Bombers de la Generalitat de Catalunya).

57.91	GROS CER	AMB CAMERA TÈRMICA	MATRICE 300		LIVE SEEKER
57.95	GROS CER	AMB CAMERA TÈRMICA	MAVIC 2 ENTERPRISE		
57.96	GROS CER	EQUIP D'IMATGES	MAVIC 2 ZOOM		
57.98	GROS CER	AMB CÀMERA TÈRMICA	MAVIC 3T		

Figure 5.1 *TRI MAER. Source [14]*

After having the opportunity to speak with a member of this unit, he explained that each drone model provides distinct functionalities and capabilities according to different operational situations. However, the MATRICE 300 RTK stands out as the most reliable and multiple features. This drone model can be seen on the far left of [5.1].

In AirSim for Unreal Engine 4.27, the default drone used is a virtual multirotor known as **`SimpleFlight`**. This agent is based on the Parrot AR Drone [28], a commercially available quadcopter.

Figure 5.2 *AirSim Multirotor. Source: Author*

5.3.1 Sensors

Drones and UAV (Unmanned Aerial Vehicles) in general, require sensor to perceive the surroundings and make decisions. Therefore, a set of sensors must be integrated into the system. In this project, real-world sensors will be replicated within the simulation environment to emulate realistic drone perception. These sensors will be installed on all aerial units. Among them:

- **GPS Sensor Module:** This component allows each UAV to continuously determine its own position in space. In the simulation, this functionality is achieved by retrieving the drone's current x, y, and z coordinates within the virtual environment.
- **Gyroscope and Velocity Sensor:** These are typically integrated into the drones's flight controller and are responsible for calculating its orientation, angular rotation, and velocity in real time. In the AirSim engine, these sensors are emulated by retrieving the drone's orientation (quaternion), angular velocity, and linear velocity directly through the AirSim API using the **`getMultirotorState()`** function.
- **High-resolution RGB camera:** In the simulation, various standard RGB camera are integrated by default into the multirotor drone model. My opinion is that the camera that best fits with the drone's purpose is the camera **`back_center`**, which is the one that I have used. This virtual camera is pointing 90 ° down and provides real-time visual data through API calls such as **`simGetImage()`** or **`simGetImages()`** from a satellite perspective. This type of camera is essential for 1vegetation providing an instant view of the terrain during fire detection by giving real time HD images.
- **Thermal camera:** A downward-facing camera intended to detect wildfires from a cenital perspective. This module will be integrated in the simulation by using the **`simGetDetections()`** [8] function and again, **`back_center`** camera. This allows the identification of objects in a ad-hoc environment that represents active fires, simulating the role of a thermal camera. When the drone detects one of these elements in its view range, it determines that a burning area has been located. This information is then used to trigger autonomous responses.

5.3.2 Architecture of Agent Behaviours

The simulation and the proposed solutions are based on a set of behaviours that the agents must perform within the defined environment. In order to develop and implement these behaviours, it is first necessary to define the tasks assigned to the agents and their purpose, so that the collective execution of these tasks can ultimately be transformed into an effective solution.

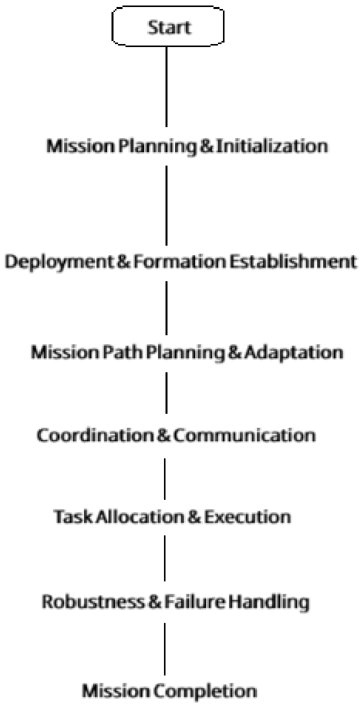

Figure 5.3 *Drone Swarm operations diagram. Source: Author*

These operations follow a structured sequence that begins with **mission planning and initialization**, where operators define the mission's primary and secondary objectives. Each drone is pre-assigned a specific role (e.g., leader or follower) and an initial flight path. In the **deployment and formation phase**, drones take off and organize themselves.

Once airborne, the swarm engages in **in-mission path planning and adaptation**, relying on either predefined waypoints or dynamic algorithms that can react to obstacles and changing conditions. During the mission, **coordination and communication** are continuous; drones exchange information to make group decisions.

Task allocation and execution are handled through distributed approaches, with drones dynamically dividing responsibilities adapting roles as needed while in mission. The system must be designed for **robustness**, allowing automatic reconfiguration when issues arise, and mantaining stable communication

Finally, upon **mission completion** the collected data such as video, sensor logs, or maps in the case of this project are compiled and processed for further analysis [29].

For this project, I have chosen to divide these behaviours into two categories, based on the purpose of the tasks. These are described below:

- **Behaviours designed for prevention missions:**
  The prevention stage is the first and most critical stage, as it aims to ensure that wildfires do not happen in the first place, and if they do, to deploy early extinction duties. The tasks intended to be performed by the agents are the following:
  - **Surveillance** of an area of wildfire records or considered to be high-risk.
  - **Search** for elements of risk, such as early-stage undetected fires or objects that may contribute to fire ignition, including litter or accumulated dry vegetation.
- **Behaviours designed for reaction missions:**
  The reaction mission happens when prevention has not succeeded and hence there is an active wildfire. In this context, the tasks in this phase are performed under risk and time constraints, which makes them more critical and complex. These tasks include:
  - Primary, secondary fires **detection**.
  - Fire **monitoring**.

This behavioural architecture aims for a structured approach to agent decision-making. By distinguishing between prevention - reaction missions, the system ensures that agents can adapt their task and actions according to the situation.

5.4. Drone agent task-drive actions

This section presents the behaviours implemented in the system, detailing their intended functionality and how they have been developed.

5.4.1 Fire Detection

To enable fire detection, this behaviour has been developed as the foundational component upon which subsequent behaviours will be built. The mission and objective of this agent behaviour is to distinguish and identify, through the use of a thermal camera, both primary and secondary fire sources on the terrain, as well as more advanced fire outbreaks. Once the behaviour is activated, the UAV will navigate to a designated point or conduct surveillance over surrounding areas where the fire has been located or is suspected to be present. Owing to the flight altitude of the aircraft and the field of view (FOV) of the thermal camera mounted on the drone, in real-world applications the system will detect a fire whenever the temperature of a given area exceeds a predefined threshold.

The main output of executing this behaviour is the most basic and primary one that can be expected, yet it remains the most important and essential for proceeding: to confirm and report, through first-hand visual resources from the field, the presence or absence of fire at specific coordinates.

To carry out the detection process, the drone uses its onboard camera in combination with a visual detection algorithm integrated into the system. This algorithm continuously analyses the images captured by the thermal camera during flight. The key idea is to identify specific visual patterns in the terrain that correspond to the presence of fire, such as flames, smoke, or burnt vegetation. In the simulation these elements are replicated by so called “Actors”, an Actor (figure 5.4) in Unreal Engine is any object that can be placed within a level and can possess position, rotation, and behaviour.

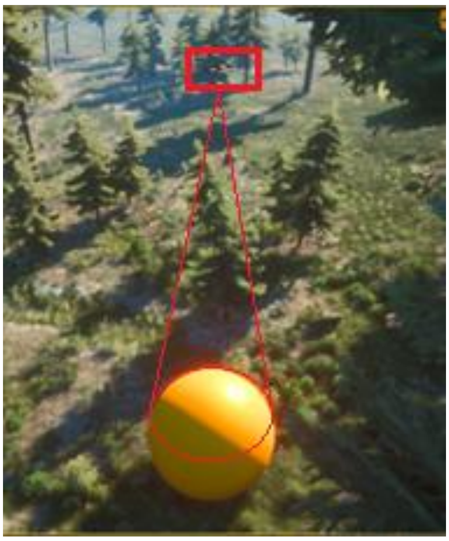

Figure 5.4 *“Sphere” actor representing a fire focus, along with a sketch illustrating the detection radius of the camera. Source: Author*

If a potential fire is detected within the scanning radius (figure 5.4), the system highlights the area of interest on the image and marks it for further inspection (figure 5.5).

Figure 5.5 *Potential fire detected by thermal camera. Source: Author*

This behaviour is individual to each agent involved; each aircraft is equipped with detection capabilities to ensure that, in subsequent behaviours, responsibility can be distributed evenly among the agents and mutual assistance is enabled. This means that each drone is capable of performing detections independently and can identify fire sources or secondary hotspots that may not be visible to, or may have been missed by, other agents.

5.4.2 Fire Monitoring

When a fire is detected and the firefighting response is activated, it is initially classified as being in the "Active" phase. This means that the fire is spreading freely, although firefighting units are already working to stabilise it. The aim of this behaviour is to support firefighters during this critical initial stage when they first confront the fire. The eventual outcome and impact of the wildfire largely depend on the firefighters' ability to effectively address it during this "Active" phase.

The objective of the agents in this behaviour is to provide operational support and assist in decision-making by monitoring the various fire sources that can be detected, as well as the surrounding area.
This fire monitoring behaviour enables a pair of autonomous drones to collaborate in the aerial detection and monitoring of a fire source. This type of operation is particularly useful during the mentioned "Active" phase of a wildfire, as it allows the collection of visual information from the affected environment at different altitudes and with varying coverage levels.
Figure 5.6 shows the aerial POV of a fire taken from a helicopter using the flight operator’s phone, this is a common way to take images of a wildfire, however, it is relatively rudimentary and lacks methodological rigour, which can lead to the omission of key data and reduce the overall reliability of the visual information gathered.

Figure 5.6 *"Bigues i Riells i del Fai" wildfire 11/06/2025, image taken from an Airbus H125 with callsign 50.41. Source [15]*

The procedure assumes previously known approximated coordinates of a fire in which an object simulating fire has been placed. Both drones take off from an initial position and ascend to different predefined altitudes, these vary depending on the parameters that the firefighters responsible for the drones consider optimal, it is necessary to take into account the terrain's geography, meteorological factors and other relevant conditions: Drone 1 (leader) flies at a lower altitude to carry out close-range detection tasks, while the other (Drone 2) maintains a higher altitude to obtain a panoramic view of the scene. This vertical separation is intended to maximise the complementarity of the visual information collected by each vehicle.

Figure 5.7 *A column of smoke has been reported to the emergency services by a local farmer. Firefighters approach the affected area and deploy the drone fleet from the anchoring point*. This figure is a simulated smoke column in Unreal Engine 4.27. Source: Author*

*Anchoring point:** The anchoring point refers to any infrastructure, natural or artificial, free of combustible material, from which a safe attack on the fire can be initiated. It also serves as a stabilisation point for securing perimeters or controlled burns.

Figure 5.8 *Drone 1 and Drone 2 airborne and moving to coordinates. Source: Author*

Once airborne, both drones autonomously navigate to the estimated location of the fire source. Drone 1 (assigned to detection tasks) activates a thermal camera vision using the sensor and function explained in 4.3.1 that enables it to identify the target object within a given radius, Drone 2 hovers close by waiting for the detection to be confirmed.

Upon detection, the Drone1 draws a bounding box over the object in the image, thereby confirming the visual location of the fire, figures 5.4 and 5.5.

From that point on, both drones initiate a coordinated surveillance behaviour that involves orbiting around the detected point. During this phase, a circular pattern is generated around the fire, allowing images to be captured from multiple angles and altitudes.

To implement this behaviour, the following key parameters define the drone's movements, firefighters can modify these when needed:

- Assigned flight altitude for each drone: **`VEHICLES={"Drone1": x, "Drone2": x}`**
- Radius of the surveillance orbit: **`ORBIT_RADIUS = x`**
- Angular velocity around the target: **`ORBIT_SPEED = x`**
- Time interval between image captures: **`SNAPSHOT_INTERVAL = x`**
- Number of complete orbits during the monitoring phase: **`ORBIT_TURNS = x`**

These two functionalities have been implemented using the **`orbit.py`** behaviour for orbiting. The orbit script in AirSim controls a drone to fly in a circular path around a defined centre point. It sets parameters like orbit radius, altitude, speed, and the number of desired orbits. Once the drone takes off and reaches the correct altitude, it uses velocity commands to continuously move around the centre, calculating look-ahead points to stay on course. It tracks how many full orbits have been completed and takes images at certain intervals. The **`take_snapshot`** function pauses the drone briefly to capture an image before resuming the flight, also using **`simGetImages`** API. When the mission is complete, the drones land and disarm.

Figure 5.9 *Fire is active and starting to spread. Source: Author*

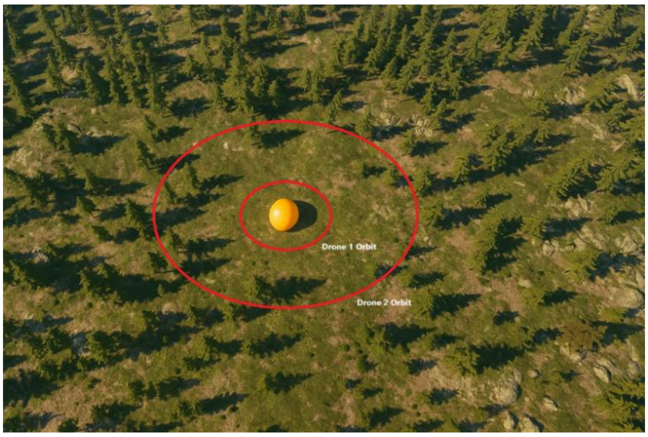

Figure 5.10 *Drone's 1 and 2 orbital monitoring patterns. Source: Author*

These images are stored in separate folders for each drone and are sequentially numbered, allowing for the visual reconstruction of the fire's evolution and its immediate surroundings from a stable, top-down perspective with circular coverage.

Response teams are also able to monitor the drone's live video feed to evaluate the fire's progression in real time (Figure 5.11).

Figure 5.11 *Real time imagery from drone camera being evaluated by firefighters. Source: [17]*

As said, image capture is carried out alongside the orbital movement, and therefore its algorithmic development is directly linked to it. Using the **`take_snapshot`** function, the drone captures RGB-type images** (**`airsim.ImageType.Scene`**) with the camera mounted on the underside of the chassis. Several adjustments have been made to ensure high-quality image capture of the detected fire source:

- The frequency of image capture is defined by the **`SNAPSHOT_INTERVAL`** parameter, which in this specific case is set to 5. This means an image is taken every five steps of the orbital displacement. This interval aims to optimise the balance between spatial coverage of the fire and the volume of images generated, avoiding both unnecessary redundancy and visual gaps in the documentation of the perimeter.

- A stabilisation function (`hold_position`) has been implemented and is executed just before each capture. This function briefly stops the drone in its current position, applies minimal translational velocity, and fixes its yaw orientation towards the target. This allows the system to stabilise for a short period (`STABILIZATION_TIME = 1.0 seconds`) before proceeding with the snapshot.

****RGB Imagery:** RGB-type images are digital images that use the red, green, and blue colour channels to represent colour. Each pixel in the image is defined by a combination of these three primary colours, allowing the representation of a wide range of colours [30].
This behaviour generates a storable output of 15 images for each orbit executed by each drone, each image is labelled as 005, 010, etc.

***Figure 5.12** Image 055 from Drone 1, flight altitude 60 m. Source: Author*

***Figure 5.13** Image 090 from Drone 2, flight altitude 150 m. Source: Author*

These images can be post-processed or directly transmitted to the command centre to assist decision-makers in tasks such as verifying the exact **location and spread** of the fire front, **detecting changes** in the smoke column or **burnt area geometry** or identifying **anchor points** for ground units or **potential escape routes**.

Drone Trajectory Interpretation and Behaviour Validation

To further validate the fire monitoring behaviour and provide tangible evidence of its successful execution, a data logging mechanism was implemented during the simulation. Each drone continuously recorded its spatial coordinates throughout the mission, generating two .csv files (Drone1.csv and Drone2.csv) containing the chronological sequence of positions (X, Y, Z). These datasets enable a detailed post-simulation analysis of the flight paths and behaviour performance.

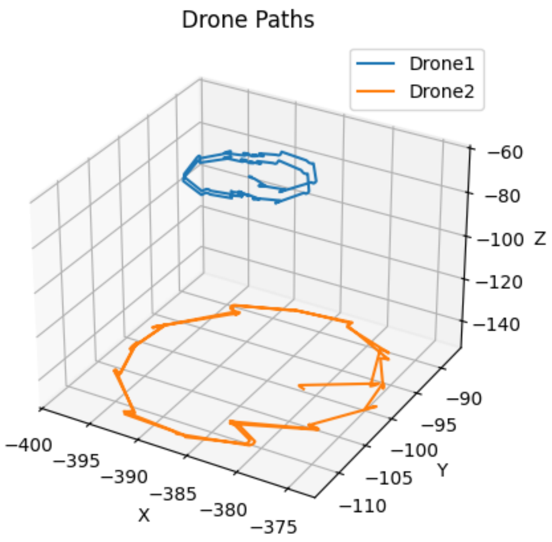

Figure5.14 *3D flight paths of Drone 1 and Drone 2 during the fire monitoring behaviour. Source: Extracted from the logged positional data.*

Using this positional data, a 3D plot was generated (Figure 5.14), which visually reconstructs the orbital paths of both drones around the detected fire source. The figure illustrates two clearly defined circular trajectories at distinct altitudes:

- Drone 1 (in blue) maintained a lower altitude to focus on close-proximity detection and detail gathering.
- Drone 2 (in orange) operated at a higher level to obtain a broader overview of the fire's perimeter.

Flight Paths: Please note that the spatial coordinate system used by AirSim follows the NED (North-East-Down) convention, where **altitude increases are represented by more negative values along the Z-axis**. Consequently, higher altitudes correspond to lower (more negative) Z values. This may appear counterintuitive when visualised, as in the 3D plot shown above: although Drone 1 is depicted at the top of the graph, it was in fact flying at a lower altitude, closer to the fire source. Conversely, Drone 2, which appears lower in the plot, was operating at a higher altitude to provide a wider field of view for monitoring purposes.

This inversion is a result of the graphical representation convention (where "up" in the plot corresponds to a numerically higher Z value) conflicting with AirSim's NED system. It is important to interpret these visuals accordingly to accurately understand the spatial behaviour and coordination between the agents.

This method of trajectory plotting has also served as a practical debugging and refinement tool. By visualising the paths taken by the drones, it becomes possible to identify unintended behaviours, deviations from the planned trajectory, or instabilities in flight execution, that is why this plot is not merely a visual illustration but also allows to confirm that the Fire Monitoring behaviour was executed as designed. It validates the flight control logic, drone coordination, and correct adherence to behavioural parameters such as altitude, separation and orbit radius.

These observations collectively provide robust evidence that the proposed fire monitoring behaviour functions autonomously and reliably, supporting its potential applicability in real-world wildfire response scenarios.

5.4.3 Boustrophedon search and surveillance flight pattern

This behaviour is designed to guide 2 agents across a predefined area of interest, Drone 1 (Explorer) flies in an area coverage motion pattern, while Drone 2 (Monitor) is responsible for monitoring detected fire sources (Fire Monitoring 5.4.2). Its purpose may fall into one of the following operational objectives:

- Mapping and vegetation scanning to assess wildfire risk by identifying flammable vegetation patterns and hazardous elements within the environment, such as litter or human-related debris, that could potentially ignite or exacerbate a fire.
- Locating fires by scanning the designated area, gathering visual and positional data, and establishing communication with firefighting teams to provide precise coordinates and facilitate rapid response actions.

Figure 5.15 *Target Area. Source: [16]*

Figure 5.16 *Flight Pattern over target area. Source: [16]*

This scanning strategy follows a lawn mower pattern, in which Drone 1 alternates its flight direction after each strip, covering the entire rectangular area efficiently. The drone ascends to a fixed altitude (value X) to avoid obstacles, moves to the far-right edge of the first scanning row, and begins a back-and-forth trajectory. The values for altitude, area width, length, row spacing, and flight speed are set as adjustable parameters and will be defined based on the specific mission and environment. This flexibility allows adaptation to open farmland, forest edges, or complex wildland-urban interfaces.

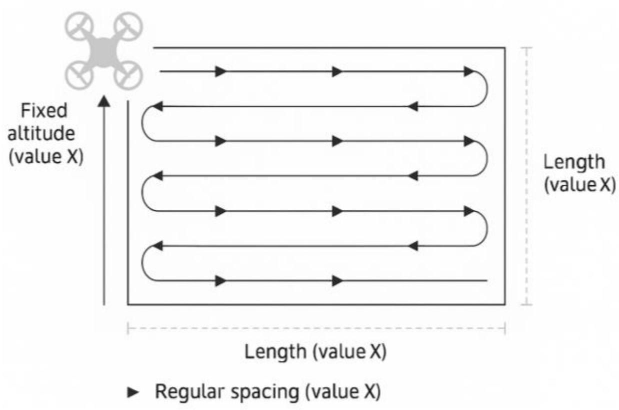

Figure 5.17 *Drone 1 Coverage Path Planning (CPP). Source: Author*

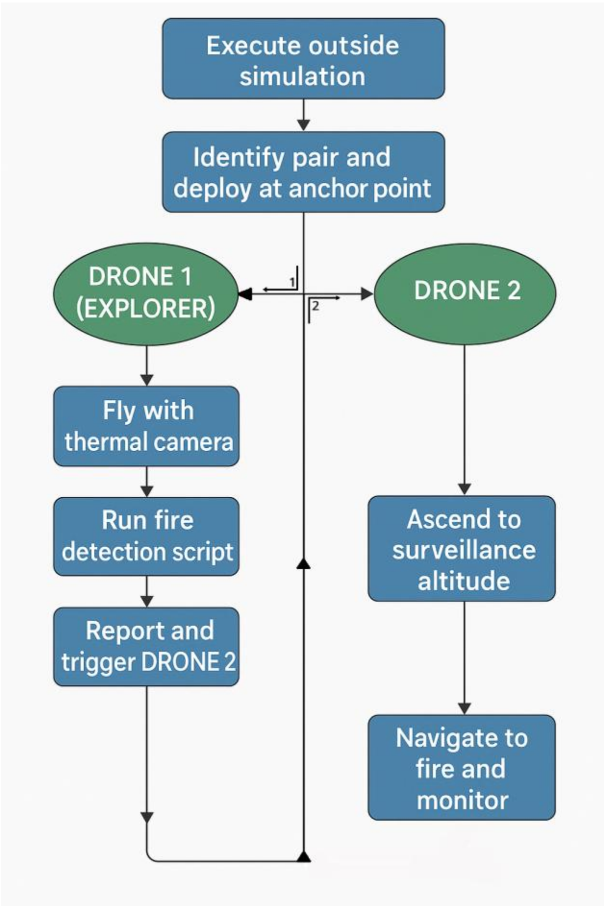

Figure 5.18 *Agent actions logic for the mission. Source: Author*

To execute this behaviour in a real-world scenario outside the simulation, it is essential that the drone pair involved is clearly identified and deployed at the same anchor point. The recommended setup involves placing both drones on a designated launch platform located a few metres away from any personnel or vehicle movement. Drone 1 will initiate its flight upon receiving the command, while Drone 2 must remain on the platform. All personnel should stay at a safe distance, as Drone 2 will take off autonomously upon receiving the signal from Drone 1.

For the first agent (Explorer), the script dynamically tracks the drone's position in real time (**`getMultirotorState`**) and defines movement goals (**`moveToPositionAsync`**) accordingly. During the entire flight operation, the onboard thermal camera runs a continuous detection loop implemented through the **`detect_fire()`** function.

This function mirrors the detection logic previously detailed in Section 5.4.1, enabling real-time thermal fire detection.

Upon fire detection, Drone 1 immediately logs the event through the execution of the **`report_detection(detection)`** function. This function not only stores the geolocation of the fire but also invokes the **`start_drone2(pos, name)`** procedure, which communicates the incident to the monitoring agent. This function call acts as a trigger for the activation of Drone 2, initiating its autonomous inspection behaviour. Drone 2 then ascends to a predefined surveillance altitude (higher than the scanning altitude of Drone 1) to avoid potential collisions and ensure an optimal overhead perspective for perimeter monitoring.

Once airborne, Drone 2 navigates to the fire coordinates and executes the **fire monitoring behaviour** explained above.

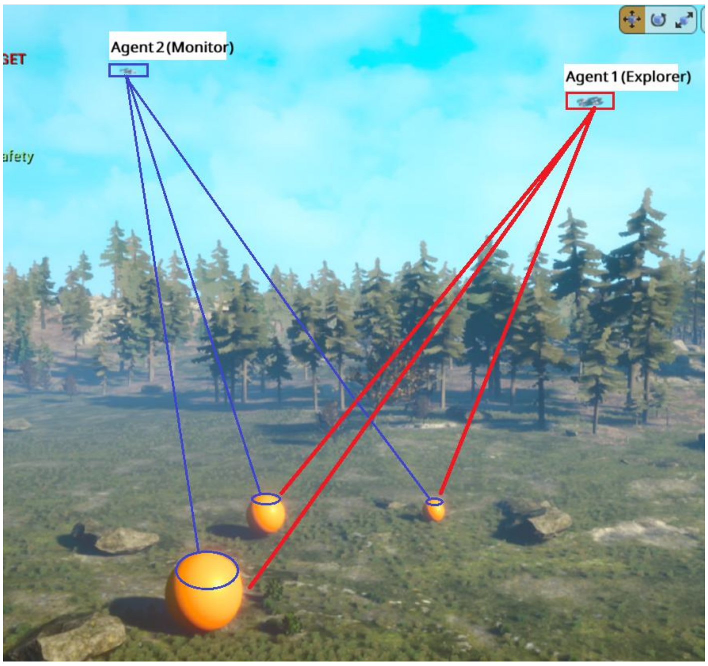

Figure 5.19 *Joint task. Source: Author*

In the figure above can be seen how both agents operate collaboratively. Agent 1 (Explorer), depicted in red, establishes visual links with each of the detected fire sources, these are recorded in a log which is shared with Agent 2 and firefighters.

```
2025-06-11 16:31:10,Drone1,FireFocus1,47.641316958094606,-122.1409733215895,125.15657806396484
2025-06-11 16:31:45,Drone1,FireFocus3_9,47.6412810217001,-122.14139996622903,126.25657653808594
2025-06-11 16:32:10,Drone1,FireFocus2_8,47.641610704909695,-122.14124798070128,123.95658111572266
```

Figure 5.20 *Detections log generated by Drone 1. Includes* ***date, time,*** *and* ***geographical coordinates*** *of the fire source detected, in that order. Source: Author*

Simultaneously, Agent 2 (Monitor), illustrated in blue, takes off and maintains communication with the same set of reported targets, preparing to execute monitoring patterns framing each confirmed fire location.

Figure 5.21 *Image_005 from Drone 2. Source: Author*

The output is the same as section 5.4.2, visual data from different angles from the fire sources as well as the surrounding areas.

Drone Trajectory Interpretation and Behaviour Validation

Again, to complement the functional description of this behaviour, a 3D trajectory plot was generated using the logged position data from both agents (Figure 5.20).

- Drone 1 (in blue) performs the explained lawnmower-style coverage pattern over the defined area of interest. The regularity and structure of this pattern confirm the correct application of the area scanning logic.
- Drone 2 (in orange) is activated once Drone 1 detects a fire focus. The trajectory shows a direct displacement from its take-off location to the area of interest, followed by the initiation of the orbital monitoring routine, in accordance with the fire monitoring behaviour described in section 5.4.2.

Flight Paths: Please remember that as in previous figures, the AirSim coordinate system follows the NED (North-East-Down) convention. This means that altitude is represented by negative Z-values, with more negative values corresponding to higher altitudes. Therefore, although Drone 1 appears higher in the plot, it is actually flying at a lower altitude than Drone 2, which hovers above the detected fire source to maintain a wide observation angle.

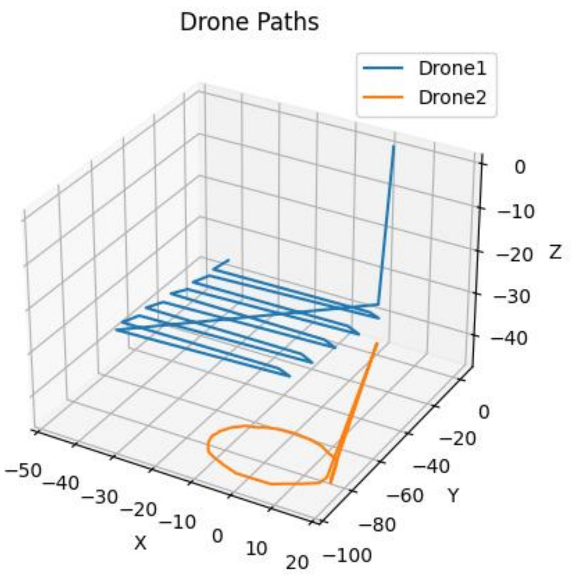

Figure 5.22: *3D trajectory plot of Drone 1 (search) and Drone 2 (monitoring) during the boustrophedon and fire response behaviour. Source: Author*

These observations collectively illustrate the correct execution of the proposed search and monitoring sequence, highlighting the system's capacity to autonomously coordinate multi-agent operations in response to dynamic fire detection.

6. Conclusions and further lines for development

6.1. Conclusions of the study

This study has explored the strategic integration of drone swarms and RBA in wildfire suppression operations. The results highlight the potential of this technology to improve both efficacy and efficiency in critical response scenarios, particularly during the early stages of fire detection and monitoring.

- One of the most significant insights derived from this research is the importance of data governance—ensuring access to reliable, real-time, and structured environmental data is essential for any AI-driven system to function optimally. The effectiveness of UAV-based operations is directly linked to the availability of updated and precise geospatial, meteorological, and vegetation-related data.
- Another key takeaway is the critical need for ongoing investment in infrastructure. This includes not only the physical deployment of UAVs and sensors but also the backend systems that support coordination, communication, and decision-making in emergency contexts.
- Equally important is the technical training of firefighting professionals, who must not only operate new tools effectively but also understand the decision logic behind autonomous agents. Bridging the gap between traditional operational procedures and emerging technologies will be a central challenge in the coming years.

6.2. Further lines for development

Looking ahead, there are several areas that deserve further exploration and development:

- RL and DSA have a considerable potential for development. So it is crucial to have support from a institutional, financial, personnel and technological terms to be able to take advantage of the many possibilities that these technologies may offer in the present and future.
- There is also room for advancing the autonomous decision-making capabilities of drones through reinforcement learning and adaptive algorithms. While this project has focused on behaviour-based logic, the introduction of learning mechanisms could allow agents to evolve and respond to unpredictable or rapidly changing environments more intelligently.
- Incorporating multi-agent reinforcement learning can enable drone swarms to operate in a more autonomous and cooperative manner, improving coordination and decentralised task execution in complex fire scenarios.

- Drones have visual access to sensitive data, such as buildings, people and private property. Therefore work must be done to ensure that this data complies with aviation and privacy legislation and prevents data hacking or virus that may compromise security standards.
- There is also room for improvement in the integration with weather and terrain monitoring satellites initiatives from the the European Space Agency or early warning services promoted by the World Meteorological Organization [32, 33].
- Finally, efforts should be made to standardise protocols and promote interoperability across firefighting agencies, allowing UAV systems to be shared, scaled, and deployed across borders in response to large-scale fire emergencies.

References

[1] *Roise, J. P. (n.d.). Joseph P. Roise. North Carolina State University.* https://cnr.ncsu.edu/directory/joseph-p-roise/

[2] SBIC Noordwijk. (n.d.). FireDynamo: *Tool for predicting wildfires – Dutch Prize of Copernicus Masters 2022. SBIC Noordwijk.* https://www.sbicnoordwijk.nl/firedynamo-tool-for-predicting-wildfires-dutch-prize-of-copernicus-masters-2022/

[3] Ministerio de Agricultura, Alimentación y Medio Ambiente. (2016*). Los incendios forestales en España. Decenio 2006-2015. Gobierno de España.* Recuperado de https://www.miteco.gob.es/content/dam/miteco/es/biodiversidad/temas/incendios-forestales/incendios-decenio-2006-2015_tcm30-521617.pdf

[4] Aircraft Cost Calculator. (n.d.). *Airbus H125-Eurocopter AS350B3e operating costs.* Retrieved March 13, 2025, from https://www.aircraftcostcalculator.com/AircraftOperatingCosts/437/Airbus+H125-Eurocopter+AS350B3e

[5] Ministry for the Ecological Transition and the Demographic Challenge. (2022). *Strategic guidelines for wildland fire management in Spain (p. 10,11,12,13). Government of Spain.* https://www.miteco.gob.es/content/dam/miteco/es/biodiversidad/temas/incendios-forestales/strategic_guidelines_wildland_fire_management_2022_tcm30-522093.pdf

[6] Boon, F. A., & Tolpekin, V. A. (2021). *Detecting and tracking firefighters in UAV video using deep learning and an object tracker.* Drones, 5(1), 17. https://doi.org/10.3390/drones5010017

[7] Generalitat de Catalunya, Departament d'Acció Climàtica, Alimentació i Agenda Rural. (2025). *Pla anual de prevenció d'incendis forestals* 2025. https://agricultura.gencat.cat/web/.content/06-medi-natural/incendis-forestals/home/pla-anual-incendis-2025.pdf

[8] Microsoft. (s.f.). AirSim: detection.py [Código fuente]. GitHub. https://github.com/microsoft/AirSim/blob/main/PythonClient/detection/detection.py

[9] Muñoz, J., López, B., Quevedo, F., Monje, C. A., Garrido, S., & Moreno, L. E. (2021). Multi UAV *Coverage Path Planning in Urban Environments.* Sensors, 21(21), 7365. https://doi.org/10.3390/s21217365

[10] Anders, N., Valente, J., Masselink, R., & Keesstra, S. (2019). *Comparing filtering techniques for removing vegetation from UAV-based photogrammetric point clouds*. Drones, 3(3), 61. https://doi.org/10.3390/drones3030061

[11] Ender, A. (2023). *Decentralised drone swarm for wildfire detection and monitoring [Bachelor's thesis*, Eindhoven University of Technology]. https://pure.tue.nl/ws/portalfiles/portal/283792115/Final_Thesis_Report.pdf

[12] Madridano Carrasco, Á. (2021). *Modelado computacional de propagación de incendios forestales mediante inteligencia artificial y teoría de grafos* [Tesis de grado, Universidad Politécnica de Madrid]. https://doi.org/10.20868/UPM.thesis.67228

[13] Seraj, E., Silva, A., & Gombolay, M. (2019). *Safe coordination of human-robot firefighting teams. Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence.* https://arxiv.org/abs/1903.06847

[14] Valente, J., Sari, B., Kooistra, L., Kramer, H., & Mücher, S. (2020). *Automated crop plant counting from very high-resolution aerial imagery. Precision Agriculture,* 21(6), 1366–1384. https://doi.org/10.1007/s11119-020-09725-3

[15] Microsoft. (2017). *AirSim: Open-source simulator for autonomous vehicles built on Unreal Engine.* GitHub. https://github.com/microsoft/AirSim

[16] Ministry of Agriculture, Fisheries and Food. (n.d.). Government of Spain. https://www.mapa.gob.es/es/

[17] Generalitat de Catalunya. (n.d.). *Fire and Rescue Service. Department of the Interior.* https://interior.gencat.cat/ca/arees_dactuacio/bombers

[18] BULTRANS. (2019). *Proceedings of the National Conference on Transport Infrastructure* (p. 30). Bulgarian Society for Transport Research. https://www.bultrans.org/files/proceedings/bultrans-2019.pdf

[19] International Union of Radio Science. (2014). *Autonomous drones for civilian missions: A systems approach.* URSI General Assembly and Scientific Symposium. https://www.ursi.org/proceedings/procGA14/papers/ursi_paper2126.pdf

[20] Herrero-Huerta, M., Rodríguez-Cielos, R., González-Aguilera, D., Picos, J., & Parés-Ramos, I. K. (2024). *UAV and AI fusion for early wildfire detection and monitoring*. npj Climate and Atmospheric Science, 7(1), 127. https://doi.org/10.1016/j.njas.2024.100189

[21] Afghah, F., Razi, A., Chakareski, J., & Ashdown, J. (2019). *Wildfire monitoring in remote areas using autonomous unmanned aerial vehicles.* In Proceedings of the IEEE Global Conference on Signal and Information Processing (GlobalSIP) (pp. 1–5). https://par.nsf.gov/servlets/purl/10133280

[22] Rashid, A. R., Hussain, M., & Khan, M. U. (2023*). A review on forest fire detection using unmanned aerial vehicles (UAVs).* International Journal of Engineering and Science, 12(10), 1–6. http://theaspd.com/index.php/ijes/article/view/913

[23] Merino, L., Caballero, F., Martínez-de-Dios, J. R., Maza, I., & Ollero, A. (2019). *An unmanned aircraft system for automatic forest fire monitoring and measurement*. Drones, 3(1), 17. https://doi.org/10.3390/drones3010017

[24] Rashid, M. M., Hossain, M. S., Hasan, M. K., & Ahmed, F. (2023). *A comprehensive review on forest fire detection techniques.* Journal of Network and Computer Applications, 219, 103697. https://doi.org/10.1016/j.jnca.2023.103697

[25] Turgut, D., & Bölöni, L. (2007). HEED: *A hybrid, energy-efficient, distributed clustering approach for ad hoc sensor networks.* CiteSeerX. https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=d766ddfc20e2e4509d72b728efe4a55bed1f8f10

[26] Santos, J. M., Ferreira, A. F., & Oliveira, A. (2022). *A review of UAV swarms for fire detection, monitoring, and fighting*. Drones, 6 (10), 269. https://doi.org/10.3390/drones6100269

[27] Epic Games. (2021, August 19). Unreal Engine 4.27 released. *Unreal Engine Blog*. https://www.unrealengine.com/en-US/blog/unreal-engine-4-27-released

[28] Wikipedia contributors. (n.d.). *Parrot AR.Drone.* Wikipedia. Retrieved June 20, 2025, from https://es.wikipedia.org/wiki/Parrot_AR.Drone

[29] Ferrández-Pastor, F. J., Garcia-Chamizo, J. M., & Nieto-Hidalgo, M. (2024). *Artificial intelligence and UAVs for wildfire prevention and management: A review.* Drones, 8(8), 350. https://doi.org/10.3390/drones8080350

[30] ScienceDirect. (n.d.). RGB image. *ScienceDirect Topics*. Retrieved June 23, 2025, from https://www.sciencedirect.com/topics/computer-science/rgb-image

[31] Roldán-Gómez, J. J., González-Gironda, E., & Barrientos, A. (2021). A Survey on Robotic Technologies for Forest Firefighting: Applying Drone Swarms to Improve Firefighters' Efficiency and Safety. Applied Sciences, 11(1), 363.

[32] *Europe's most advanced weather satellite is now fully operational*. (2024, 4 december). World Meteorological Organization.

[33] *Early warning system*. (2023, 14 november). World Meteorological Organization. https://wmo.int/topics/early-warning-system

[34] *Both risks and benefits emphasized in global UFF Recommendations for Safe and Ethical Use of AI in the fire services. (s. f.).* Both risks And benefits emphasized In global UFF recommendations for safe and Ethical use of AI in the fire services | CTIF - International Association of Fire Services for safer citizens through skilled firefighters. *https://ctif.org/news/recommendations-safe-and-ethical-use-ai-fire-service-points-risks-and-advantages*

[35] Firoz, Yadav, S., & Yadav, S. (2024*). A comprehensive review of reinforcement learning algorithms and their applications*.*www.allresearchjournal.com*. https://www.allresearchjournal.com/archives/?year=2018&vol=4&issue=9&part=C&ArticleId=11454

[36] Shah, V., Makati, S., Dugad, D., Gupta, V., Gandhi, T., & Kiran, M. B. (2024). *Implementing Drones: An In-Depth Exploration of Challenges and Innovative Solutions for Drone Applications*. En *Lecture notes in mechanical engineering* (pp. 241-255). https://doi.org/10.1007/978-981-97-3173-2_17

References of figures

[1] Institut Cartogràfic i Geològic de Catalunya. (s.f.*). Foc al bosc [Interactive map].* Extracted from https://visors.icgc.cat/focalbosc/.

[2] Roldán-Gómez, J. J., González-Gironda, E., & Barrientos, A. (2021). *A Survey on Robotic Technologies for Forest Firefighting: Applying Drone Swarms to Improve Firefighters' Efficiency and Safety.* Applied Sciences, 11(1), 363.

[3] Esri. (s.f.). *Wildfire Risk Mapping. ArcGIS.* https://www.arcgis.com/apps/mapviewer/index.html?layers=a77b40183b2b4f60b9419c895160286d

[4] TOT Rubí. (2024, March 8th). "*L'Associació de Propietaris Agroforestals treballa a Can Roig per evitar incendis*. TOT Rubí." https://www.totrubi.cat/actualitat/societat/associacio-propietaris-agroforestals-treballa-can-roig-evitar-incendis_2163106102.html

[5] University of South Carolina. (n.d.). *The fire triangle*. University of South Carolina. https://www.sc.edu/ehs/training/Fire/01_triangle.htm

[6] Agents Rurals Catalunya. *[Tweet]*. X. https://x.com/agentsruralscat/status/1898648977513337097

[7] FireDynamo (s.f.). *Vegetation map based on satellite imagery*, *Estimations of fire progression using agent-based simulations*. sbicnoordwijk.nl/firedynamo-tool-for-predicting-wildfires-dutch-prize-of-copernicus-masters-2022/

[8] Santos, J. M., Ferreira, A. F., & Oliveira, A. (2022). *A review of UAV swarms for fire detection, monitoring, and fighting*. Drones, 6 (10), 269. https://doi.org/10.3390/drones6100269

[9] Valente, J., Sari, B., Kooistra, L., Kramer, H., & Mücher, S. (2020). *Automated crop plant counting from very high-resolution aerial imagery. Precision Agriculture,* 21(6), 1366–1384. https://doi.org/10.1007/s11119-020-09725-3

[10] Cadena SER. (2025, January 4). Los drones de emergencias realizan más de 40 horas de vuelo y participan en 20 operaciones en la zona afectada por la DANA. https://cadenaser.com/comunitat-valenciana/2025/01/04/los-drones-de-emergencias-realizan-mas-de-40-horas-de-vuelo-y-participan-en-20-operaciones-en-la-zona-afectada-por-la-dana-radio-valencia/

[11] Herrero-Huerta, M., Rodríguez-Cielos, R., González-Aguilera, D., Picos, J., & Parés-Ramos, I. K. (2024). *UAV and AI fusion for early wildfire detection and monitoring*. npj Climate and Atmospheric Science, 7(1), 127. https://doi.org/10.1016/j.njas.2024.100189

[12] Merino, L., Caballero, F., Martínez-de-Dios, J. R., Maza, I., & Ollero, A. (2019). *An unmanned aircraft system for automatic forest fire monitoring and measurement*. Drones, 3(1), 17. https://doi.org/10.3390/drones3010017

[13] Ministerio de Agricultura, Alimentación y Medio Ambiente. (2016). *Los incendios forestales en España. Decenio 2006-2015*. Gobierno de España. Recuperado de https://www.miteco.gob.es/content/dam/miteco/es/biodiversidad/temas/incendios-forestales/incendios-decenio-2006-2015_tcm30-521617.pdf

[14] TRI MAER 12/07/2024: Telegram Mitjans Aeris. (Reserved access for workers involved)

[15] Bombers (@bomberscat). (2024, April 20). #bomberscat activem els mitjans aeris per atacar un incendi de vegetació a la zona de... [Tweet]. X. https://x.com/bomberscat/status/1932784133148656055

[16] Google. (n.d.). *Google Earth [Computer software*]. https://earth.google.com/

[17] Generalitat de Catalunya. (2024, March 11). La consellera Parlon presenta l'ampliació de la Unitat de Drons dels Bombers de la Generalitat [Press release]. https://govern.cat/salapremsa/notes-premsa/706722/consellera-parlon-presenta-lampliacio-unitat-drons-dels-bombers-generalitat

Annexes

Python scripts for agent movement logic:

Fire Detection

```
import setup_path
import airsim
import cv2
import numpy as np
import pprint
import time

# Configuration parameters
OBJECT_NAME = "Fire"
CRUISE_ALT = -30
HOVER_ALT = -20
VEHICLE_NAME = "Drone1"
HOVER_DURATION = 15

# Connect to AirSim
client = airsim.MultirotorClient()
client.confirmConnection()
client.enableApiControl(True, VEHICLE_NAME)
client.armDisarm(True, VEHICLE_NAME)

# Takeoff
client.takeoffAsync(vehicle_name=VEHICLE_NAME).join()

# Ascend to the desired cruise altitude before any horizontal movement
client.moveToZAsync(CRUISE_ALT, 10, vehicle_name=VEHICLE_NAME).join()

# Record starting position
start_pos =
client.getMultirotorState(vehicle_name=VEHICLE_NAME).kinematics_estimated
.position

# Get object pose for positioning
pose = client.simGetObjectPose(OBJECT_NAME)
target = pose.position

# Move horizontally to x,y while keeping cruise altitude
client.moveToPositionAsync(target.x_val, target.y_val, CRUISE_ALT, 10,
vehicle_name=VEHICLE_NAME).join()

# Setup detection
CAMERA_NAME = "3"
IMAGE_TYPE = airsim.ImageType.Scene
client.simSetDetectionFilterRadius(CAMERA_NAME, IMAGE_TYPE, 200 * 100)
client.simAddDetectionFilterMeshName(CAMERA_NAME, IMAGE_TYPE,
OBJECT_NAME)

print("Searching for object...")

found = False
detection_info = None

while not found:
```

```
    raw = client.simGetImage(CAMERA_NAME, IMAGE_TYPE)
    if not raw:
        continue
    img = cv2.imdecode(airsim.string_to_uint8_array(raw),
cv2.IMREAD_UNCHANGED)
    detections = client.simGetDetections(CAMERA_NAME, IMAGE_TYPE)
    if detections:
        for det in detections:
            if det.name == OBJECT_NAME:
                detection_info = det
                pprint.pprint(det)
                cv2.rectangle(img, (int(det.box2D.min.x_val),
int(det.box2D.min.y_val)),
                                    (int(det.box2D.max.x_val),
int(det.box2D.max.y_val)), (255, 0, 0), 2)
                cv2.putText(img, det.name, (int(det.box2D.min.x_val),
int(det.box2D.min.y_val - 10)),
                            cv2.FONT_HERSHEY_SIMPLEX, 0.5, (36, 255, 12))
                found = True
                break
    cv2.imshow("AirSim", img)
    if cv2.waitKey(1) & 0xFF == ord('q'):
        break

if found:
    print("Object detected, descending to hover altitude")
    client.moveToPositionAsync(target.x_val, target.y_val, HOVER_ALT, 7,
vehicle_name=VEHICLE_NAME).join()
    time.sleep(1)  # allow stabilization

    # Capture image while hovering
    hover_raw = client.simGetImage(CAMERA_NAME, IMAGE_TYPE)
    if hover_raw:
        png = airsim.string_to_uint8_array(hover_raw)
        img = cv2.imdecode(png, cv2.IMREAD_UNCHANGED)
        filename = f"hover_capture_{OBJECT_NAME}.png"
        cv2.imwrite(filename, img)
        print(f"Image saved as {filename}")

    print(f"Hovering for {HOVER_DURATION} seconds...")
    time.sleep(HOVER_DURATION)

# Cleanup detection filter
client.simClearDetectionMeshNames(CAMERA_NAME, IMAGE_TYPE)
cv2.destroyAllWindows()

# Ascend (or descend) back to cruise altitude before returning
client.moveToZAsync(CRUISE_ALT, 10, vehicle_name=VEHICLE_NAME).join()

print("Returning to start position")
client.moveToPositionAsync(start_pos.x_val, start_pos.y_val, CRUISE_ALT,
10, vehicle_name=VEHICLE_NAME).join()

# Controlled landing
client.landAsync(vehicle_name=VEHICLE_NAME).join()
client.armDisarm(False, VEHICLE_NAME)
client.enableApiControl(False, VEHICLE_NAME)
print("Mission complete")
```

Multiagent Fire Monitoring

```
import setup_path
import airsim
import cv2
import os
import math
import time
import csv

# Configuration parameters
OBJECT_NAME = "FireSphere"
VEHICLES = {
    "Drone1": -60,
    "Drone2": -150
}
CAMERA_NAME_DETECT = "3"
CAMERA_NAME_PHOTO = "3"
IMAGE_TYPE = airsim.ImageType.Scene
ORBIT_RADIUS = 12
ORBIT_SPEED = 5
ORBIT_TURNS = 2
SNAPSHOT_DIR = "fire_images_monitoring_5"
SNAPSHOT_INTERVAL = 5
STABILIZATION_TIME = 1.0
PATH_DIR = "flight_paths"

positions = {v: [] for v in VEHICLES}


def hold_position(client, vehicle, yaw_deg):
    state = client.getMultirotorState(vehicle_name=vehicle)
    pos = state.kinematics_estimated.position
    client.moveToPositionAsync(
        pos.x_val,
        pos.y_val,
        pos.z_val,
        0.5,
        drivetrain=airsim.DrivetrainType.MaxDegreeOfFreedom,
        yaw_mode=airsim.YawMode(False, yaw_deg),
        vehicle_name=vehicle,
    ).join()
    time.sleep(STABILIZATION_TIME)


def take_snapshot(client, vehicle, yaw_deg, index):
    hold_position(client, vehicle, yaw_deg)
    resp = client.simGetImages(
        [airsim.ImageRequest(CAMERA_NAME_PHOTO, IMAGE_TYPE, False,
True)],
        vehicle_name=vehicle,
    )[0]
    filename = os.path.join(SNAPSHOT_DIR, vehicle,
f"image_{index:03d}.png")
    airsim.write_file(filename, resp.image_data_uint8)


def record_positions(client):
    for vehicle in VEHICLES:
        state = client.getMultirotorState(vehicle_name=vehicle)
```

```
        pos = state.kinematics_estimated.position
        positions[vehicle].append((pos.x_val, pos.y_val, pos.z_val))


def save_paths():
    os.makedirs(PATH_DIR, exist_ok=True)
    for vehicle, path in positions.items():
        csv_path = os.path.join(PATH_DIR, f"{vehicle}.csv")
        with open(csv_path, "w", newline="") as f:
            writer = csv.writer(f)
            writer.writerow(["x", "y", "z"])
            writer.writerows(path)

    try:
        import matplotlib.pyplot as plt
        from mpl_toolkits.mplot3d import Axes3D  # noqa: F401
        fig = plt.figure()
        ax = fig.add_subplot(111, projection="3d")
        for vehicle, path in positions.items():
            if not path:
                continue
            xs, ys, zs = zip(*path)
            ax.plot(xs, ys, zs, label=vehicle)
        ax.set_xlabel("X")
        ax.set_ylabel("Y")
        ax.set_zlabel("Z")
        ax.legend()
        plt.title("Drone Paths")
        plt.savefig(os.path.join(PATH_DIR, "paths.png"))
        plt.close()
        print(f"Saved path plot to {os.path.join(PATH_DIR,
'paths.png')}")
    except Exception as e:
        print("Could not generate plot: ", e)
        print(f"Paths saved as CSV in {PATH_DIR} directory")


def main():
    client = airsim.MultirotorClient()
    client.confirmConnection()

    home = {}
    takeoff_jobs = {}

    for vehicle in VEHICLES:
        client.enableApiControl(True, vehicle_name=vehicle)
        client.armDisarm(True, vehicle_name=vehicle)
        takeoff_jobs[vehicle] = client.takeoffAsync(vehicle_name=vehicle)

    for vehicle, job in takeoff_jobs.items():
        job.join()
        state = client.getMultirotorState(vehicle_name=vehicle)
        home[vehicle] = state.kinematics_estimated.position

    move_jobs = {}
    for vehicle, altitude in VEHICLES.items():
        print(f"{vehicle}: Ascending to Z={altitude:.2f}")
        move_jobs[vehicle] = client.moveToPositionAsync(
            home[vehicle].x_val,
            home[vehicle].y_val,
```

```
            altitude,
            10,
            vehicle_name=vehicle,
        )

    for job in move_jobs.values():
        job.join()

    pose = client.simGetObjectPose(OBJECT_NAME)
    target = pose.position
    move_jobs = {}
    for vehicle, altitude in VEHICLES.items():
        print(
            f"{vehicle}: Moving to X={target.x_val:.2f},
Y={target.y_val:.2f}, Z={altitude:.2f}"
        )
        move_jobs[vehicle] = client.moveToPositionAsync(
            target.x_val,
            target.y_val,
            altitude,
            7,
            vehicle_name=vehicle,
        )

    for job in move_jobs.values():
        job.join()
    time.sleep(2)

    client.simSetDetectionFilterRadius(
        CAMERA_NAME_DETECT, IMAGE_TYPE, 200 * 100, vehicle_name="Drone1"
    )
    client.simAddDetectionFilterMeshName(
        CAMERA_NAME_DETECT, IMAGE_TYPE, OBJECT_NAME,
vehicle_name="Drone1"
    )
    print("Searching")

    detected = False
    while not detected:
        raw = client.simGetImage(CAMERA_NAME_DETECT, IMAGE_TYPE,
vehicle_name="Drone1")
        if not raw:
            continue
        png = cv2.imdecode(airsim.string_to_uint8_array(raw),
cv2.IMREAD_UNCHANGED)
        detections = client.simGetDetections(CAMERA_NAME_DETECT,
IMAGE_TYPE, vehicle_name="Drone1")
        if detections:
            for d in detections:
                if d.name == OBJECT_NAME:
                    cv2.rectangle(png, (int(d.box2D.min.x_val),
int(d.box2D.min.y_val)),
                                  (int(d.box2D.max.x_val),
int(d.box2D.max.y_val)), (255, 0, 0), 2)
                    cv2.putText(png, d.name, (int(d.box2D.min.x_val),
int(d.box2D.min.y_val - 10)),
                                cv2.FONT_HERSHEY_SIMPLEX, 0.5, (36, 255,
12))
                    detected = True
                    break
```

```
        cv2.imshow("AirSim", png)
        if cv2.waitKey(1) & 0xFF == ord('q'):
            break

    client.simClearDetectionMeshNames(
        CAMERA_NAME_DETECT, IMAGE_TYPE, vehicle_name="Drone1"
    )

    if detected:
        print(
            f"Detected {OBJECT_NAME} at X={target.x_val:.2f},
Y={target.y_val:.2f}"
        )
        for vehicle in VEHICLES:
            path = os.path.join(SNAPSHOT_DIR, vehicle)
            if not os.path.isdir(path):
                os.makedirs(path)

        dt = 0.2
        steps_per_orbit = int((2 * math.pi * ORBIT_RADIUS) / (ORBIT_SPEED
* dt))
        total_steps = int(steps_per_orbit * ORBIT_TURNS)
        print("Orbiting around detected object")

        for i in range(total_steps):
            angle = (i * 2 * math.pi) / steps_per_orbit
            x = target.x_val + ORBIT_RADIUS * math.cos(angle)
            y = target.y_val + ORBIT_RADIUS * math.sin(angle)
            yaw_deg = math.degrees(angle + math.pi)

            for vehicle, altitude in VEHICLES.items():
                client.moveToPositionAsync(x, y, altitude, ORBIT_SPEED,

drivetrain=airsim.DrivetrainType.MaxDegreeOfFreedom,

yaw_mode=airsim.YawMode(is_rate=False, yaw_or_rate=yaw_deg),
                                           vehicle_name=vehicle)
            if i % SNAPSHOT_INTERVAL == 0:
                for vehicle in VEHICLES:
                    take_snapshot(client, vehicle, yaw_deg, i)
            record_positions(client)
            time.sleep(dt)

    print("Returning to base")
    move_jobs = {}
    for vehicle, altitude in VEHICLES.items():
        move_jobs[vehicle] = client.moveToPositionAsync(
            home[vehicle].x_val,
            home[vehicle].y_val,
            altitude,
            7,
            vehicle_name=vehicle,
        )

    for job in move_jobs.values():
        job.join()

    land_jobs = {}
    for vehicle in VEHICLES:
        print(f"{vehicle}: Landing")
```

```
        land_jobs[vehicle] = client.moveToPositionAsync(
            home[vehicle].x_val,
            home[vehicle].y_val,
            home[vehicle].z_val,
            7,
            vehicle_name=vehicle,
        )

    for job in land_jobs.values():
        job.join()

    for vehicle in VEHICLES:
        client.landAsync(vehicle_name=vehicle).join()
        client.armDisarm(False, vehicle_name=vehicle)
        client.enableApiControl(False, vehicle_name=vehicle)

    cv2.destroyAllWindows()
    save_paths()
    print("Mission executed")


if __name__ == "__main__":
    main()
```

Multiagent boustrophedon search and surveillance flight pattern and monitoring

```
import os
import math
import threading
import time
import csv

import airsim
import setup_path

DRONE1 = "Drone1"
DRONE2 = "Drone2"

CAMERA_NAME = "3"
IMAGE_TYPE = airsim.ImageType.Scene
DRONE2_CAMERA = CAMERA_NAME

SEARCH_ALTITUDE = 35
ORBIT_ALTITUDE_OFFSET = 10
ORBIT_RADIUS = 15
ORBIT_SPEED = 5
ORBIT_TURNS = 1
ORBIT_SNAPSHOTS = 8
IMAGE_DIR = "fire_images_16_06_try2"
LOG_FILE = os.path.join(os.path.dirname(__file__),
"detections_log_16_06_try2.csv")
PATH_DIR = "flight_paths_lawn_mower_try2"

client = airsim.MultirotorClient()
client.confirmConnection()

for name in (DRONE1, DRONE2):
    client.enableApiControl(True, name)
```

```
client.armDisarm(True, DRONE1)

HOME_GEO = client.getHomeGeoPoint()

client.simSetDetectionFilterRadius(CAMERA_NAME, IMAGE_TYPE, 200 * 100,
vehicle_name=DRONE1)

for mesh in ["FireSphere", "FireSphere2", "FireSphere3"]:
    client.simAddDetectionFilterMeshName(CAMERA_NAME, IMAGE_TYPE, mesh,
vehicle_name=DRONE1)

_detected = set()
_pending = []
_drone2_busy = False
_drone2_thread = None
_drone2_home =
client.getMultirotorState(vehicle_name=DRONE2).kinematics_estimated.posit
ion

positions = {DRONE1: [], DRONE2: []}

if not os.path.exists(IMAGE_DIR):
    os.makedirs(IMAGE_DIR)
if not os.path.exists(LOG_FILE):
    try:
        with open(LOG_FILE, "w", newline="") as f:
            csv.writer(f).writerow([
                "timestamp",
                "drone",
                "object",
                "latitude",
                "longitude",
                "altitude",
            ])
    except PermissionError as exc:
        print(f"Could not create log file {LOG_FILE}: {exc}")


def record_position(drone, client_obj=None):
    if client_obj is None:
        client_obj = client
    state = client_obj.getMultirotorState(vehicle_name=drone)
    pos = state.kinematics_estimated.position
    positions[drone].append((pos.x_val, pos.y_val, pos.z_val))


def save_paths():
    os.makedirs(PATH_DIR, exist_ok=True)
    for drone, path in positions.items():
        csv_path = os.path.join(PATH_DIR, f"{drone}.csv")
        with open(csv_path, "w", newline="") as f:
            writer = csv.writer(f)
            writer.writerow(["x", "y", "z"])
            writer.writerows(path)

    try:
        import matplotlib.pyplot as plt
        from mpl_toolkits.mplot3d import Axes3D  # noqa: F401
        fig = plt.figure()
        ax = fig.add_subplot(111, projection="3d")
```

```
        for drone, path in positions.items():
            if not path:
                continue
            xs, ys, zs = zip(*path)
            ax.plot(xs, ys, zs, label=drone)
        ax.set_xlabel("X")
        ax.set_ylabel("Y")
        ax.set_zlabel("Z")
        ax.legend()
        plt.title("Drone Paths")
        plt.savefig(os.path.join(PATH_DIR, "paths.png"))
        plt.close()
        print(f"Saved path plot to {os.path.join(PATH_DIR,
'paths.png')}")
    except Exception as e:
        print("Could not generate plot: ", e)
        print(f"Paths saved as CSV in {PATH_DIR} directory")


def geodetic_to_ned(point, home):
    earth_radius = 6378137.0
    d_lat = math.radians(point.latitude - home.latitude)
    d_lon = math.radians(point.longitude - home.longitude)
    north = d_lat * earth_radius
    east = d_lon * earth_radius * math.cos(math.radians(home.latitude))
    down = home.altitude - point.altitude
    return airsim.Vector3r(north, east, -down)


def report_detection(detection):
    gp = detection.geo_point
    key = (
        detection.name,
        round(gp.latitude, 7),
        round(gp.longitude, 7),
        round(gp.altitude, 2),
    )
    if key in _detected:
        return False
    _detected.add(key)
    try:
        with open(LOG_FILE, "a", newline="") as f:
            csv.writer(f).writerow([
                time.strftime("%Y-%m-%d %H:%M:%S"),
                DRONE1,
                detection.name,
                gp.latitude,
                gp.longitude,
                gp.altitude,
            ])
    except PermissionError as exc:
        print(f"Could not write detection log {LOG_FILE}: {exc}")
    print(
        f"Detected {detection.name}: lat {gp.latitude:.7f}, lon
{gp.longitude:.7f}, alt {gp.altitude:.2f}"
    )
    return True


def detect_fire():
```

```
    raw = client.simGetImage(CAMERA_NAME, IMAGE_TYPE,
vehicle_name=DRONE1)
    if raw:
        detections = client.simGetDetections(CAMERA_NAME, IMAGE_TYPE,
vehicle_name=DRONE1)
        if detections:
            for d in detections:
                if d.name.startswith("FireSphere") and
report_detection(d):
                    pos = geodetic_to_ned(d.geo_point, HOME_GEO)
                    start_drone2(pos, d.name)


def drone2_mission(fire_pos, fire_name):
    global _drone2_busy
    client2 = airsim.MultirotorClient()
    client2.confirmConnection()

    altitude = SEARCH_ALTITUDE + ORBIT_ALTITUDE_OFFSET
    steps = int((2 * math.pi * ORBIT_RADIUS) / (ORBIT_SPEED * 0.2))
    total_steps = steps * ORBIT_TURNS

    ts = time.strftime("%Y%m%d_%H%M%S")
    det_dir = os.path.join(IMAGE_DIR, f"{fire_name}_{ts}")
    os.makedirs(det_dir, exist_ok=True)
    print(f"Drone2 orbiting {fire_name}")

    client2.enableApiControl(True, DRONE2)
    client2.armDisarm(True, DRONE2)
    client2.takeoffAsync(vehicle_name=DRONE2).join()
    client2.moveToZAsync(-altitude, 3, vehicle_name=DRONE2).join()
    record_position(DRONE2, client_obj=client2)
    start = airsim.Vector3r(fire_pos.x_val + ORBIT_RADIUS,
fire_pos.y_val, -altitude)
    client2.moveToPositionAsync(start.x_val, start.y_val, start.z_val,
ORBIT_SPEED, vehicle_name=DRONE2).join()
    record_position(DRONE2, client_obj=client2)

    for step in range(total_steps):
        angle = (step / steps) * 2 * math.pi
        x = fire_pos.x_val + ORBIT_RADIUS * math.cos(angle)
        y = fire_pos.y_val + ORBIT_RADIUS * math.sin(angle)
        yaw = math.degrees(angle + math.pi)
        client2.moveToPositionAsync(
            x, y, -altitude, ORBIT_SPEED,
            drivetrain=airsim.DrivetrainType.MaxDegreeOfFreedom,
            yaw_mode=airsim.YawMode(False, yaw),
            vehicle_name=DRONE2,
        ).join()
        record_position(DRONE2, client_obj=client2)
        if step % max(1, steps // ORBIT_SNAPSHOTS) == 0:
            img = client2.simGetImage(DRONE2_CAMERA, IMAGE_TYPE,
vehicle_name=DRONE2)
            if img:
                airsim.write_file(os.path.join(det_dir,
f"image_{step:03d}.png"), img)

    client2.moveToPositionAsync(_drone2_home.x_val, _drone2_home.y_val, -
altitude, ORBIT_SPEED, vehicle_name=DRONE2).join()
    record_position(DRONE2, client_obj=client2)
```

```
    client2.landAsync(vehicle_name=DRONE2).join()
    record_position(DRONE2, client_obj=client2)
    client2.armDisarm(False, DRONE2)
    client2.enableApiControl(False, DRONE2)
    _drone2_busy = False
    if _pending:
        next_pos, next_name = _pending.pop(0)
        start_drone2(next_pos, next_name)


def start_drone2(pos, name):
    global _drone2_busy, _drone2_thread
    if _drone2_busy:
        _pending.append((pos, name))
        return
    _drone2_busy = True
    _drone2_thread = threading.Thread(target=drone2_mission, args=(pos,
name))
    _drone2_thread.start()


def fly_to(drone, target):
    client.moveToPositionAsync(target[0], target[1], -target[2], 7,
vehicle_name=drone).join()
    record_position(drone)
    if drone == DRONE1:
        detect_fire()


def get_position(drone):
    p =
client.getMultirotorState(vehicle_name=drone).kinematics_estimated.positi
on
    return [p.x_val, p.y_val, -p.z_val]


start_pos = get_position(DRONE1)
start_pos[2] = SEARCH_ALTITUDE

panel_width = 40
panel_length = 60
row_distance = panel_length / 10

current = get_position(DRONE1)
target = [current[0] - panel_width, current[1], SEARCH_ALTITUDE]
fly_to(DRONE1, target)

for row in range(1, 6):
    current = get_position(DRONE1)
    target = [current[0], current[1] - row_distance, SEARCH_ALTITUDE]
    fly_to(DRONE1, target)
    target = [current[0] + panel_width, target[1], SEARCH_ALTITUDE]
    fly_to(DRONE1, target)
    target = [target[0], target[1] - row_distance, SEARCH_ALTITUDE]
    fly_to(DRONE1, target)
    target = [target[0] - panel_width, target[1], SEARCH_ALTITUDE]
    fly_to(DRONE1, target)

fly_to(DRONE1, start_pos)
```

```
while _drone2_busy:
    time.sleep(1)

if _drone2_thread:
    _drone2_thread.join()

client.moveToZAsync(0, 5, vehicle_name=DRONE1).join()
record_position(DRONE1)
client.landAsync(vehicle_name=DRONE1).join()
record_position(DRONE1)
client.armDisarm(False, DRONE1)
for name in (DRONE1, DRONE2):
    client.enableApiControl(False, name)

save_paths()
```