POLITECNICO
MILANO 1863

SCUOLA DI INGEGNERIA INDUSTRIALE
E DELL'INFORMAZIONE

# Coverage path planning algorithm for non-homogeneous multi-UAV systems

Tesi di Laurea Magistrale in
Agricultural Engineering - Ingegneria Agricola

Author: **Ludovico Mondinelli**

Student ID: 221396
Advisor: Prof. Marco Lovera
Co-advisors: Alessandro Nazzari, Roberto Rubinacci
Academic Year: 2024-2025

# Abstract

Monitoring and managing large agricultural areas remains a major challenge in the field of precision agriculture. The use of non-homogeneous multi-UAV systems provides a concrete opportunity to optimize area coverage operations, reducing mission time and operational costs. This thesis presents the implementation of a coverage path planning system, an approach aimed at generating paths that allow a drone to completely cover a given area, using the DARP (Divide Areas based on Robot's initial Positions) algorithm, which is particularly well-suited for both monitoring and chemical spreading tasks in agricultural contexts. Additionally, the work includes a critical evaluation of others coverage path planning algorithms found in literature. The DARP algorithm was tested in simulations on fields with varying complexity and obstacle presence, and compared against the Popcorn algorithm (another algorithm for coverage path planning found in literature). Results demonstrate that DARP, especially when using the "better coverage" mode, achieves a good field coverage, lower computational time, fewer turns, and the ability to manage heterogeneous UAV swarms and no-fly zones. These features make it an effective and scalable solution for modern precision agriculture applications.

**Keywords:** Drone, UAV, Precision Agriculture, Coverage Path Planning, DARP, Multi-UAV

# Abstract in lingua italiana

Il monitoraggio e la gestione di vaste superfici agricole rappresentano una delle principali sfide nell'ambito dell'agricoltura di precisione. L'impiego di sistemi multi-UAV non omogenei offre un'opportunità concreta per ottimizzare le operazioni di copertura del territorio, riducendo i tempi e i costi delle attività. Questa tesi presenta l'implementazione di un sistema di coverage path planning (un approccio volto a generare percorsi che permettano a un drone di coprire completamente una determinata area) utilizzando l'algoritmo DARP (Divide Areas based on Robots' initial Positions), particolarmente adatto per attività di monitoraggio e di distribuzione di sostanze chimiche in contesti agricoli. Il lavoro include inoltre una valutazione critica di altri algoritmi di coverage path planning trovati in letteratura. L'algoritmo DARP è stato testato attraverso simulazioni su campi di diversa complessità geometrica e presenza di ostacoli, confrontandolo con l'algoritmo Popcorn (un altro algoritmo per coverage path planning trovato in letteratura). I risultati mostrano che DARP, soprattutto nella modalità "better coverage", garantisce una buona copertura, minori tempi di calcolo, meno cambi di direzione e la possibilità di gestire sciami eterogenei e no-fly zones, rendendolo una soluzione efficace e scalabile per l'agricoltura di precisione.

**Parole chiave:** Drone, UAV, Agricoltura di precisione, Coverage Path Planning, DARP, Multi-UAV

# Contents

# 1 | Introduction

Initially, farmers relied on visual and manual inspections to assess crop health and soil conditions. Although fundamental, these methods were limited by subjectivity and the vast areas that needed monitoring. Aerial photography represented a first step toward remote monitoring. The earliest aerial images were captured in 1858 in France by Nadar (Gaspard-Félix Tournachon) using a hot air balloon.[17] Subsequently, photogrammetry enabled precise measurements of the terrain through the analysis of aerial photographs. These techniques found applications in numerous fields, ranging from cartography to geology, and laid the foundation for the development of modern remote sensing. In the 1980s, the introduction of Global Positioning System (GPS) technology marked a turning point in the agricultural sector, paving the way for the development of precision agriculture. Precision agriculture, also known as precision farming, is a data-driven approach to farm management that leverages advanced technologies to observe, measure, and respond to variability in crops. The core objective of precision agriculture is to enhance the efficiency and sustainability of farming operations by ensuring that the right amount of input (such as water, fertilizer, pesticides, and seeds) is applied at the right time and in the right place. By enabling site-specific management, farmers can reduce waste, increase productivity, and minimize environmental impacts. For example, GPS-guided equipment allows for more accurate planting and harvesting, reducing overlaps and missed areas, while sensor data and predictive analytics help identify crop stress and disease outbreaks before they spread. Additionally, this approach facilitates better record-keeping and decision-making, contributing to more informed and adaptive farm management strategies. The term "precision agriculture" was first used in 1990 during a workshop held in Montana.[11] Starting from the 2000s, drones began to be employed in agriculture. Their adoption subsequently increased, enabling crop condition monitoring, field mapping, and crop management. Drones have become highly versatile tools in precision agriculture, offering advantages such as real-time data collection and targeted application of treatments.

## 1.1. Problem definition

Monitoring vast agricultural lands remains a significant challenge in the field of Precision Agriculture. Among the various approaches discussed in the literature, such as satellite images,[13] employing Unmanned Aerial Vehicles (UAVs) has emerged as a prominent option due to its flexibility and adaptability. However, the application of UAVs in Precision Agriculture is still an evolving area of research. This study explores the most effective coverage strategies for multi-UAV non homogeneous systems in Precision Agriculture, emphasizing the generation of trajectories to ensure complete coverage. By identifying optimal coverage methods and designing smooth trajectories, this research aims to minimize energy consumption and improve the overall efficiency and flexibility of multi-UAV systems in Precision Agriculture applications.

## 1.2. Contributions

This thesis provides an in-depth analysis of the main Coverage Path Planning (CPP) methods available in the literature, offering criteria for their selection based on the specific application context. The selected method is described in detail, with a particular focus on its use in precision agriculture. The main contribution of this work lies in the development of a systematic approach that enables the transformation of the problem of supervising a field using a heterogeneous swarm of UAVs into executable individual trajectories for each drone.

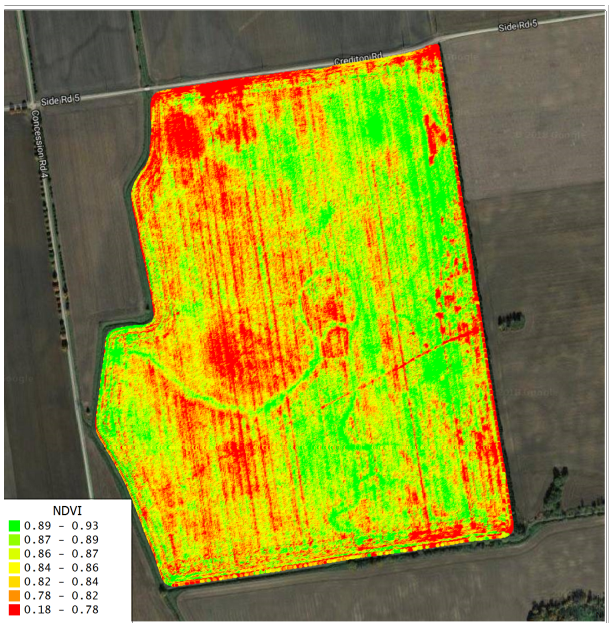



Figure 1.1: Obtain the Normalized difference vegetation index (NDVI) of a field is an example of application where multi-UAV system can have an important role.

## 1.3. Thesis outline

In this section the structure of the thesis is described.

- Chapter 2: Reviews precision agriculture applications and coverage control algorithms from existing literature.
- Chapter 3: Presents the selection criteria and the in-depth analysis of the coverage path planning method selected.
- Chapter 4: Explains the methodology for trajectory generation.
- Chapter 5: Presents and discuss the tests results.
- Chapter 6: Presents the final conclusions of the work.

# 2 | State of the Art

In this chapter an overview of multi-UAV systems used in precision agriculture and their application is presented then a collection of CPP algorithms found in literature is discussed.

## 2.1. Aerial platforms for precision agriculture

In precision agriculture, various categories of UAVs are utilized for a wide range of applications, from crop monitoring to fertilization. Each category has its own advantages and disadvantages, which should be carefully assessed to determine the most appropriate option for each specific case.

### Fixed-Wing UAVs

Fixed-wing UAVs resemble small airplanes and are powered by a single or multiple propellers. These drones are known for their efficiency in covering large areas.[10, 12]

**Advantages:**

- Long flight duration, making them ideal for monitoring large fields.
- High-speed capabilities from 15 to 30 m/s, ideal for covering large distances between target areas efficiently.
- High payload capacity.

**Disadvantages:**

- Require larger takeoff and landing areas compared to Multirotor and Hybrid UAVs.
- Typically more expensive compared to Multirotor UAVs.
- Limited maneuverability, making them less suited for small or irregularly shaped fields.

### Multirotor UAVs

Multirotor UAVs include quadcopters, hexacopters and other variants characterized by their vertical takeoff and landing capabilities.[10, 12]

**Advantages:**

- Highly maneuverable and capable of hovering, allowing for close-up inspections.
- Require minimal space for takeoff and landing.
- Cost-effective and accessible for small to medium-sized farms.

**Disadvantages:**

- Limited flight time due to hight energy consumption.
- Slower maximum speeds between 5 to 15 m/s make them less efficient for covering extensive areas.
- Payload capacity is generally lower than fixed-wing UAVs.

### Hybrid UAVs

Hybrid UAVs combine the features of fixed-wing and multirotor UAVs, offering vertical capability with the ability to cover large areas efficiently.[10, 12]

**Advantages:**

- Versatility in takeoff/landing and efficient coverage.
- Longer flight times compared to multirotor UAVs.
- Medium/high payload capacity.

**Disadvantages:**

- More expensive than single-type UAVs.
- Bigger size compared to multirotor and fixed-wing UAVs
- Generally have intermediate performance between the other 2 categories.

## 2.2. UAVs applications in agriculture

In agriculture multi-UAV systems can be employed across two major categories of applications, each serving distinct purposes. The first category focuses on monitoring tasks,

which include activities such as surveillance, data collection, environmental observation, and infrastructure inspection. These applications are crucial for gathering real-time information and ensuring the efficient management of resources. The second category is dedicated to chemical spreading, which encompasses tasks like fertilizing, disinfection, pest control, and other operations requiring the precise distribution of chemical substances on the field.

## Monitoring task

Multi-UAV systems can be useful in agriculture to collect information in a wide range of situations and environments. Key applications include area surveillance for security purposes, environmental monitoring to assess ecosystem health, crop and animal monitoring and infrastructure inspections such as bridges, roads, pipelines, power lines, and building.

The most significant use in agriculture is the crop monitoring, where drones are employed to evaluate crop health, detect areas of water stress or pest infestations, and optimize resource usage such as water and fertilizers. For these purposes, indices derived from multispectral and hyperspectral image analysis, captured by sensors mounted on drones, are commonly utilized.

Among the most widely used indices is the **Normalized Difference Vegetation Index (NDVI)**, which measures vegetation quantity and quality through the ratio of near-infrared (NIR) to red (RED) spectral bands.[7]

$$NDVI = \frac{NIR - RED}{NIR + RED}$$

Where:

- $NIR$: Reflectance in the near-infrared band.
- $RED$: Reflectance in the red visible band.

Other commonly used indices include:

- **Enhanced Vegetation Index (EVI):** particularly useful for analyzing high-density vegetation areas while minimizing atmospheric effects.[7]

  $$EVI = G \cdot \frac{(NIR - RED)}{(NIR + C_1 \cdot RED - C_2 \cdot BLUE + L)}$$

  Where:

  - $G$: Gain factor (typically 2.5).

  - $NIR$: Reflectance in the near-infrared band.
  - $RED$: Reflectance in the red band.
  - $BLUE$: Reflectance in the blue band.
  - $C_1$: Coefficient for atmospheric correction in the red band (typically 6).
  - $C_2$: Coefficient for atmospheric correction in the blue band (typically 7.5).
  - $L$: Soil adjustment factor (typically 1).

- **Soil-Adjusted Vegetation Index (SAVI):** ideal for areas where soil is partially visible, compensating for soil reflectance influences.[7]

$$SAVI = \frac{(NIR - RED)}{(NIR + RED + L)} \cdot (1 + L)$$

  Where:

  - $L$: Soil adjustment factor, varying with vegetation density (typically 0.5 for moderate vegetation).

- **Normalized Difference Water Index (NDWI):** used to evaluate water content in crops or soil, providing crucial data for precision irrigation.[7]

$$NDWI = \frac{(NIR - SWIR)}{(NIR + SWIR)}$$

  Where:

  - $NIR$: Reflectance in the near-infrared band.
  - $SWIR$: Reflectance in the shortwave infrared band.

- **GNDVI (Green NDVI):** emphasizes the green spectrum for more accurate chlorophyll estimation.

$$GNDVI = \frac{(NIR - GREEN)}{(NIR + GREEN)}$$

  Where:

  - $NIR$: Reflectance in the near-infrared band.
  - $GREEN$: Reflectance in the green band.

Different types of NDVI calculations and multispectral cameras require carefully selected flight altitudes to ensure optimal spatial resolution and data quality. The choice of altitude directly influences the ground sampling distance (GSD), the level of detail captured, and

the overall accuracy of the vegetation indices, making it a critical parameter in remote sensing applications.

Beyond agriculture, these techniques are employed in forestry management to prevent wildfires, monitoring air and water quality, and protecting wildlife in conservation areas by preventing poaching activities.

By deploying multiple drones simultaneously, these monitoring tasks can be carried out more quickly, accurately, and efficiently compared to traditional methods, ensuring comprehensive and detailed coverage of the area under observation.

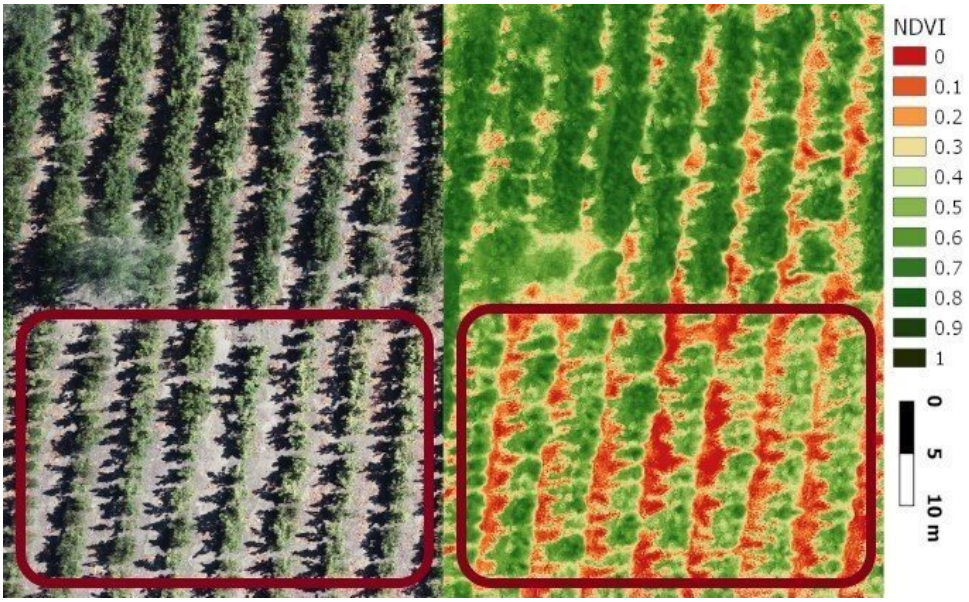



Figure 2.1: This is an example of potential use of NDVI images to locate the vegetated area in a field and their status in fact in this picture can be seen that in the upper part the vegetation is more developed and this can be related for example to the soil condition in that specific part of the field.

## Spreading task

This capability is particularly valuable for tasks that require the precise, efficient, and uniform distribution of substances over large areas.

The use of multi-UAV systems for chemical spreading in agriculture can revolutionize traditional farming practices by introducing precision and efficiency to tasks that are essential for crop health and productivity. These systems are primarily employed for the aerial application of liquid fertilizers, pesticides, herbicides, and other agricultural inputs, ensuring optimal coverage and reducing the environmental footprint.

One of the key advantages of UAV-based chemical spreading is the targeted application it enables. By utilizing GPS navigation and prescription maps, drones can precisely spray chemicals only where needed, avoiding unnecessary application in unaffected areas. This precision minimizes chemical waste, lowers costs, and reduces the risk of contamination to nearby ecosystems. Similar to NDVI indices, altitude remains a key parameter to consider in this context, as it helps reduce waste due to wind and other environmental factors.

Multi-UAV systems are particularly effective in challenging terrains such as steep slopes, irregular fields, or areas with obstacles like trees and water bodies. In these scenarios, traditional ground-based equipment may struggle to provide uniform coverage, whereas UAVs can navigate with ease and deliver consistent results.

Another critical application of UAVs in agriculture is the ability to perform precision spraying during time-sensitive windows. For example, the timely application of pesticides after detecting an infestation can prevent crop loss, while targeted nutrient spraying can enhance yields during key growth stages.

Figure 2.2: This is an example of UAV used for chemical spreading

Multi-UAV systems also provide scalability, allowing farmers to cover large areas quickly by deploying multiple drones simultaneously. This capability is invaluable for managing extensive agricultural operations or responding to emergencies such as sudden pest outbreaks.

In summary, the adoption of multi-UAV systems for chemical spreading in agriculture represents a significant leap forward in modern farming. By combining precision, adaptability, and efficiency, these systems contribute to sustainable agricultural practices, higher

yields, and reduced environmental impact.

## 2.3. Coverage Path Planning algorithms for multi-UAV systems

The literature features a wide range of algorithms applied across diverse scenarios. This section provides a brief classification of Coverage Path Planning algorithms and gives a brief description of the algorithm more suited for agricultural applications.

### 2.3.1. Algorithm Classification

#### Online and Offline

The distinction between online and offline algorithms lies in how they handle information about the environment.

- **Online algorithms:** These algorithms generate paths in real-time, dynamically adapting to new information as the drone explores the environment. They are particularly useful in unknown or changing terrains, where pre-mapped data is unavailable. For instance, an online algorithm can adjust the path to avoid unexpected obstacles or cover newly identified areas.[16]
- **Offline algorithms:** These algorithms generate a complete path before the drone begins its operation, based on pre-existing information about the environment.[1, 4, 14] They are more efficient in well-mapped and static environments, where the absence of unknown obstacles or dynamic events allows for optimal path planning in terms of both time and energy consumption.

#### With Area Decomposition and Without Area Decomposition

The distinction between these algorithms lies in how they process (or do not process) the target area.

- **Algorithms with area decomposition:** These algorithms divide the workspace into smaller sub-areas, one for each drone, and then plan the path for each sub-area. This strategy allows for easier management of complex environments, reducing computational difficulty and facilitating movement control in irregular spaces. Examples of such approaches include those using Voronoi diagrams or Delaunay triangulation.[1, 2, 8, 15]

- **Algorithms without area decomposition:** In this case, the robot plans the path directly on the entire area without dividing it. These algorithms tend to be simpler and faster in regular or simple-shaped environments but may be less efficient in complex spaces since they do not leverage area division to optimize the path.[14, 16]

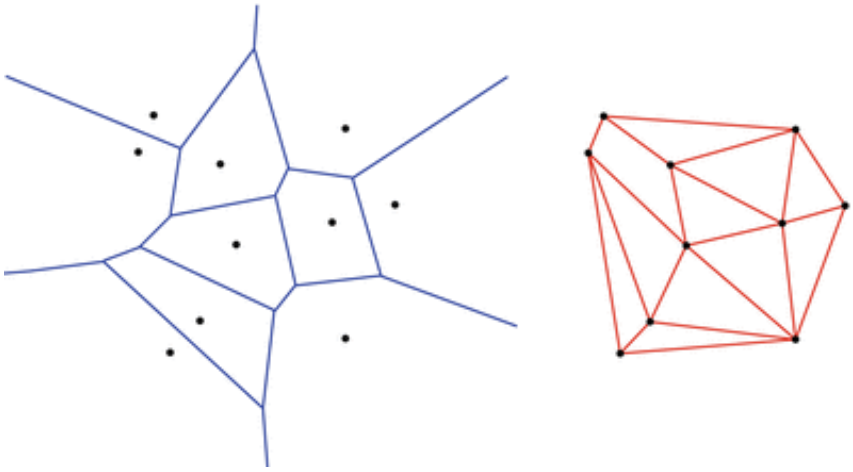


Figure 2.3: On the left there is an example of Voronoi decomposition and on the right an example of Delaunay triangulation.

### 2.3.2. Algorithms overview

This section provides a selection of algorithms developed to address the Coverage Path Planning (CPP) problem. The selected algorithms were chosen to provide a general overview of this class of methods with particular attention to their suitability for multi-drone systems, making them suitable candidates for real-world deployment in precision agriculture contexts.

- **Tsunami** [16]: This algorithm is an online algorithm without area decomposition that is able to dynamically adapt to changes in swarm size, such as failures or battery depletion, ensuring efficient workload distribution and mission performance. It is subdivided in to two phases: The first offline phase let the user define the environment to be scouted using GPS polygons representing the area of interest and no-fly zones, the environment is discretized into a grid of GPS waypoints and in this grid is calculated a wavefront traversal, which prioritizes waypoint coverage based on proximity and reduces the likelihood of collisions. The second phase is the online (Runtime) one where a drone pool [Figure 2.4] manages the available UAVs, with each drone classified as "active," "ready," or "idle", ready drones are assigned the nearest unexplored waypoint while avoiding paths already taken by other drones

through collision detection and avoidance mechanisms, faults, such as drone failures or low batteries, are handled seamlessly by returning affected drones to the pool, allowing the remaining drones to continue the mission.

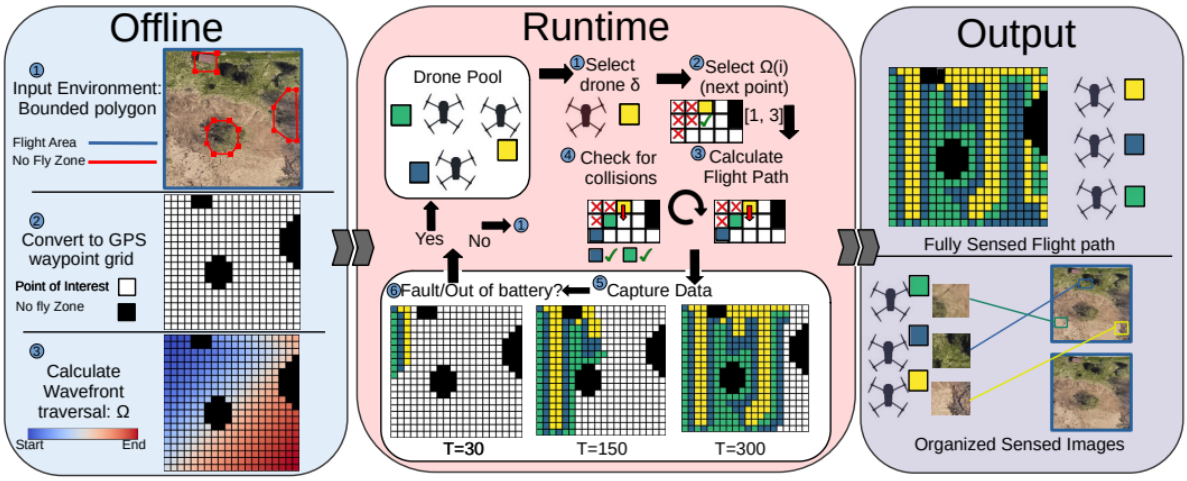



Figure 2.4: This scheme explain how Tsunami works showing clearly the 2 main phases.

- **Spiral** [3]: This is a class of algorithms, generally offline, that create spiral paths to cover an area entirely, ensuring every point is visited without redundancy. It works as follow:

  The target area is initially defined as a grid, polygon, or rectangular region, where each point is considered a location that requires coverage. The starting point is usually chosen at the center of the area or at one of its corners. From there, the algorithm generates a spiral-shaped path that expands outward (or inward), ensuring systematic coverage. For rectangular areas, the path forms concentric rectangles, while for circular areas, it takes a rounded, spiral trajectory. The algorithm calculates the waypoints by moving straight for a fixed distance and then turning at a specific angle depending on the single algorithm. The process repeats, ensuring all waypoints are visited sequentially. This approach is particularly efficient because it minimizes overlap and unnecessary travel. It's simple to implement and scales easily to different area shapes and sizes. Spiral CPP is commonly used for mapping agricultural fields, surveying large open spaces, and inspecting circular or rectangular structures.

Figure 2.5: An example of a simple spiral algorithm used to map a field.

- **Popcorn** [14]: This algorithm is an offline algorithm without area decomposition and addresses the challenge of covering large and complex areas by constructing a grid (lattice) over the survey region, with each node representing a point where data (e.g., images) will be captured. The algorithm then calculates paths for multiple drones, ensuring complete coverage of the area while minimizing redundant travel and optimizing for constraints like limited battery life, altitude limits, and environmental concerns. The core of Popcorn's innovation lies in using a Satisfiability Modulo Theory (SMT) framework, which allows precise encoding of constraints into the planning process. For example, the algorithm ensures that paths start and end at the same location (cyclical paths), enabling safe recall of drones in case of low battery or unexpected issues. The SMT approach iteratively optimizes paths, reducing their length until no further improvement is feasible, resulting in highly efficient route planning. Compared to traditional geometric methods like sweep or spiral patterns, Popcorn is significantly faster and reduces redundant travel. It achieves up to a 17.3% reduction in path length compared to other approaches, making it highly suitable for simultaneous multi-drone operations.[14]

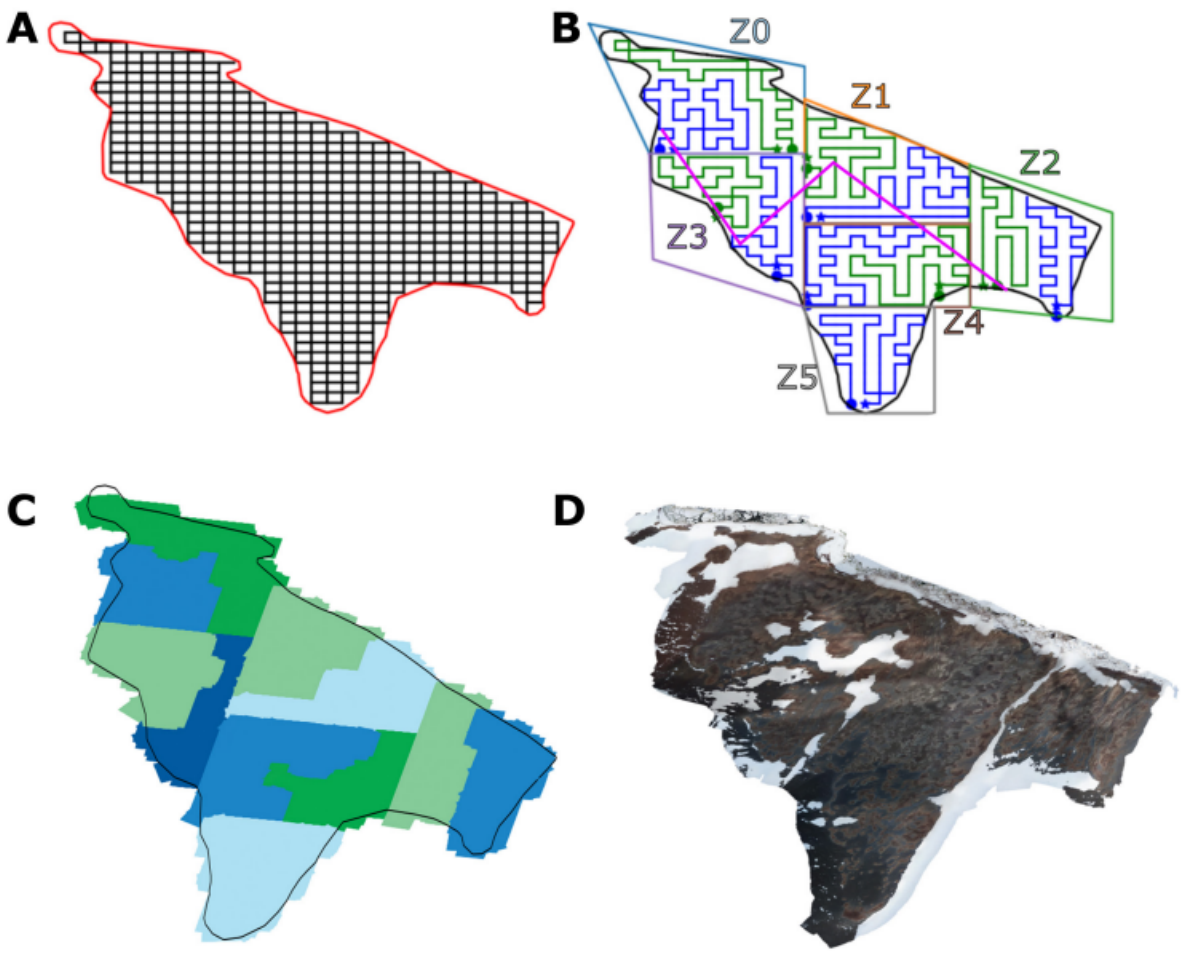



Figure 2.6: This is an example of the Popcorn algorithm used to map a penguin colony in Antarctica.

- **Back and Forward (BF or sweep)** [4]: This is a class of algorithms that are designed to ensure complete coverage of a target area moving in the simplest way possible, these are generally offline algorithms and often make use of area decomposition.

  The algorithm begins by defining the area to be covered. This area is typically represented as a polygon or a rectangular grid with known dimensions and boundaries. When area decomposition is not employed, the input region is generally required to be a convex polygon to ensure proper coverage planning. A starting point is chosen, often at one corner of the area, to initiate the coverage path. The algorithm then divides the area into parallel strips, with the width of each strip determined by the effective range of the sensor or tool being used. The path follows a zigzag pattern, where the UAV moves back and forth along these strips. It traverses one strip

in a single direction and, upon reaching the boundary, shifts to the adjacent strip and reverses direction. This back-and-forth motion continues systematically until the entire area is covered. The algorithm is particularly effective in rectangular or regularly shaped areas, where its simplicity ensures consistent results. However, in irregularly shaped areas, it may require additional steps to handle uncovered edges or adapt to complex boundaries. Furthermore, frequent turns at the strip boundaries can increase energy consumption, particularly for fixed-wing UAVs. Despite these limitations, these algorithms remain a highly practical solution for applications like agricultural field mapping. Its ease of implementation and ability to systematically cover large areas make it a preferred choice in many scenarios.

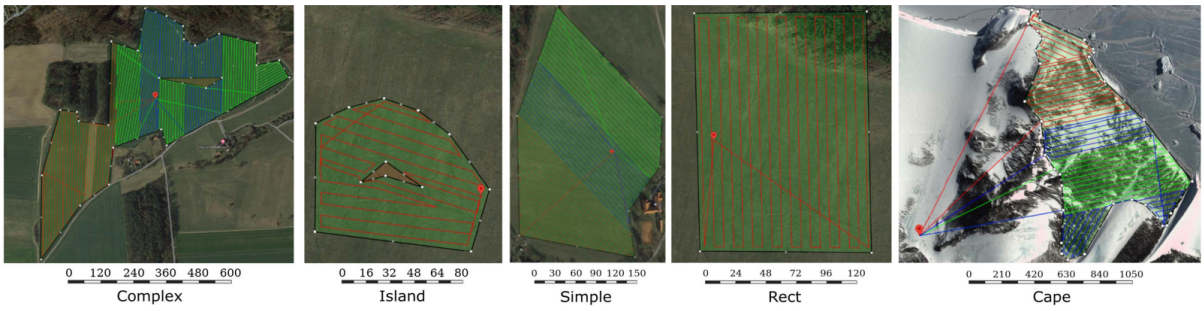



Figure 2.7: Some examples of an algorithm that used the BF technique to map an area.

- **DARP** [1, 2, 8, 15]: This is an offline algorithm with area decomposition and is designed to efficiently coordinate a team of UAVs for coverage missions in remote sensing applications.

  It begins by dividing the region of interest (ROI) into a grid of cells based on the desired scanning resolution. Using simulated annealing, the grid is optimized to maximize coverage and minimize path inefficiencies by adjusting its placement and orientation. Once the grid is optimized, the Divided Area based on Robot's initial Positions (DARP) allocates exclusive sub-regions to each UAV. This ensures that tasks are evenly or proportionally distributed based on the UAVs' capabilities and initial positions, avoiding overlaps and potential collisions. Within each assigned sub-region, paths are generated using a Minimum Spanning Tree (MST) approach. This process minimizes redundant movements and reduces the number of turns, significantly improving energy efficiency and reducing mission duration. The algorithm is highly adaptable, handling complex-shaped ROIs, accommodating obstacles, and managing heterogeneous UAV capabilities. It outputs optimized paths for complete coverage, ensuring efficient use of time and energy, making it particularly valuable for applications such as precision agriculture, search and rescue, and environmental

monitoring.

Figure 2.8: This is an example of an area mapped using DARP.

# 3 | Coverage path planning for agriculture applications

The selection of the CPP (Coverage Path Planning) algorithm for this project was based on several key considerations.

The first aspect evaluated was the domain in which the algorithm would be applied. As discussed in the previous chapter, multi-UAV systems in agriculture primarily perform two tasks: monitoring and spreading. While the specific paths taken by drones during monitoring are not critical for mission success, spreading requires precise path planning to ensure uniform coverage. Minimizing the number of turns is essential to avoid excessive variations in application density due to differences in drone speed between the inner and outer parts of a turn, particularly in cases involving wide fertilizer sprayer arms. Consequently, trajectory patterns such as Popcorn, Tsunami, and Spiral were deemed unsuitable for this application.

Another critical factor was the operating environment. Agricultural fields typically do not contain moving obstacles, making offline algorithms more advantageous than online ones. Offline algorithms not only reduce power consumption but also enable a more balanced workload distribution among UAVs. Given this, online algorithms, including Tsunami, were excluded from consideration.

In addition to dynamic obstacles, static obstacles with complex geometries must also be carefully accounted for during the planning phase. Among these, power lines are particularly critical, as they can reach heights of 30–50 meters with base structures spanning over 10 meters in width. If not preemptively identified and designated as no-fly zones, these structures can pose significant challenges during flight operations. Similarly, trees represent a notable hazard due to their branches, which not only risk damaging the drone but also compromising the success of the mission. This risk is further exacerbated by the fact that branches can move unpredictably in the wind, increasing the likelihood of mid-air collisions. Algorithms such as Popcorn are particularly vulnerable in these scenarios, as they rely solely on onboard obstacle avoidance systems, which may lack the neces-

sary precision to detect and avoid such hazards in real time. This limitation increases the risk of drone loss and may result in substantial financial consequences for the user, especially given the inability of such algorithms to effectively handle predefined no-fly zones. Moreover, proper obstacle management becomes even more critical when considering operational constraints such as altitude limits imposed by NDVI data acquisition and chemical spraying activities. In such cases, simply flying above the obstacles is not always a viable or acceptable solution.

The final considerations involved the type of UAV and the composition of the multi-UAV system. The selected platform is multirotor UAVs, chosen for their maneuverability (as they do not require a minimum turn radius), affordability (making them accessible to small-scale farmers), and ease of use (as they do not require large takeoff and landing areas). Additionally, the project involves a heterogeneous multi-UAV system, meaning that an algorithm capable of assigning different coverage areas to each UAV is preferred.

After evaluating all these factors, DARP (Divide Areas based on Robot's initial Positions) was selected as the most suitable algorithm, as it satisfies all the key requirements:

- Suitable for both monitoring and spreading tasks.
- Offline algorithm to optimize power consumption and task distribution.
- Supports no-fly zones as input.
- Well-suited for multirotor UAVs.
- Allows variable area allocation for different UAVs.

This selection ensures an efficient, reliable, and well-adapted path-planning approach for the intended agricultural applications.

## 3.1. Algorithm description

This section provides a detailed description of the selected algorithm, which can be divided into three main parts:

- Grid placement and optimization. [1]
- Region of interest division (DARP). [8]
- Path generation (MST). [1]

### 3.1.1. Grid placement and optimization

- **Coordinate Transformation**

  The coordinates of the Region of Interest (ROI), obstacles, and initial UAV positions are initially expressed in the WGS84 (World Geodetic System 1984) reference system. To facilitate mission management, these coordinates are transformed into the local NED (North East Down) system, which uses a Cartesian coordinate system with units in meters. This conversion simplifies node placement and path optimization. Once paths are calculated, the coordinates are converted back to WGS84 for practical implementation.

- **Node Placement**

  The user-defined Region of Interest (ROI) is represented on a grid. The grid resolution is based on the user-defined scanning density $d_s$, with the spacing between grid nodes set to $d_n = 2d_s$ to ensure proper separation between planned paths.

  To determine the grid size, a bounding box is created around the ROI polygon [Figure 3.1]. The grid will have dimensions:

$$x = \lfloor \frac{(x_{max} - x_{min})}{d_n} \rfloor \tag{3.1}$$

$$y = \lfloor \frac{(y_{max} - y_{min})}{d_n} \rfloor \tag{3.2}$$

  Each grid cell's center becomes a node, which will later be used to construct a Minimum Spanning Tree (MST), guiding the path layout. Each node placed on the grid is classified into one of these three categories that determines how and if it will be used during path generation:

  – **Obstacles:** Nodes outside the ROI or inside no-fly zones (NFZs).

  – **Free Space:** Nodes used for path planning and MST construction.

  – **Drones:** Nodes representing the initial UAV positions.

  Two approaches are used for node classification: the first named "Strictly in Polygon" keeps paths within the ROI, while the second named "Better Coverage" allows slight extensions beyond the boundaries to maximize coverage.

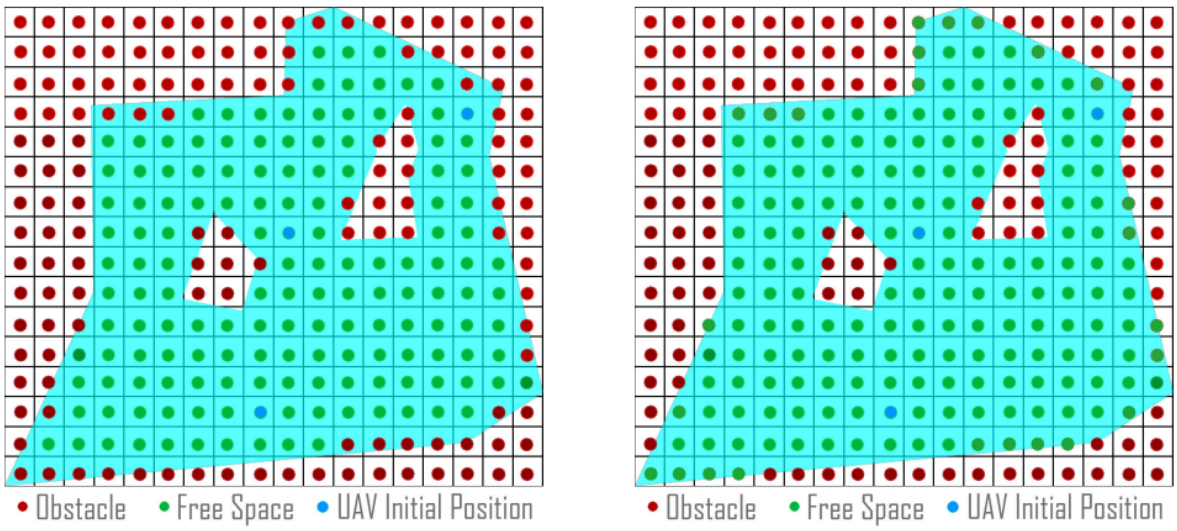



Figure 3.1: The two approaches implemented on the same polygon: on the left the "Strictly in Polygon" approach and on the right side the "Better Coverage" approach

- **Node Placement Optimization**

  One major issue with grid-based methodologies is the loss of accuracy in ROI representation due to discretization. This can reduce effective mission coverage. Increasing the grid resolution may improve coverage; however, it also leads to longer mission durations, higher energy consumption, and increased computational demands, as finer resolution reduces the spacing between paths, thereby increasing overall path length. To address this issue, node placement optimization using the Simulated Annealing algorithm is introduced. This algorithm seeks the configuration that maximizes coverage by adjusting three parameters:

  - **Translation in X:** Moving the ROI polygon along the X-axis with a displacement $s_x \in [0, d_n]$.
  - **Translation in Y:** Moving the ROI along the Y-axis with a displacement $s_y \in [0, d_n]$.
  - **Rotation:** Rotating the polygon relative to the grid with an angle $\theta \in [0, 90°]$.

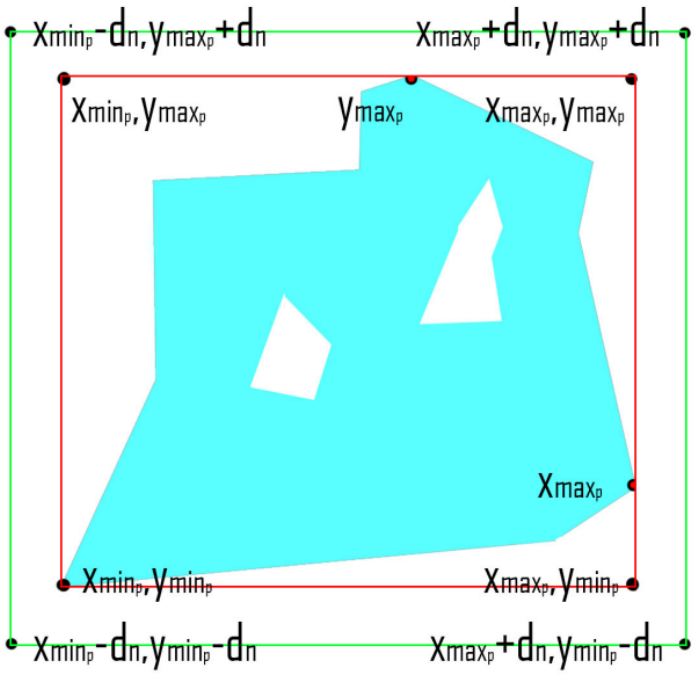
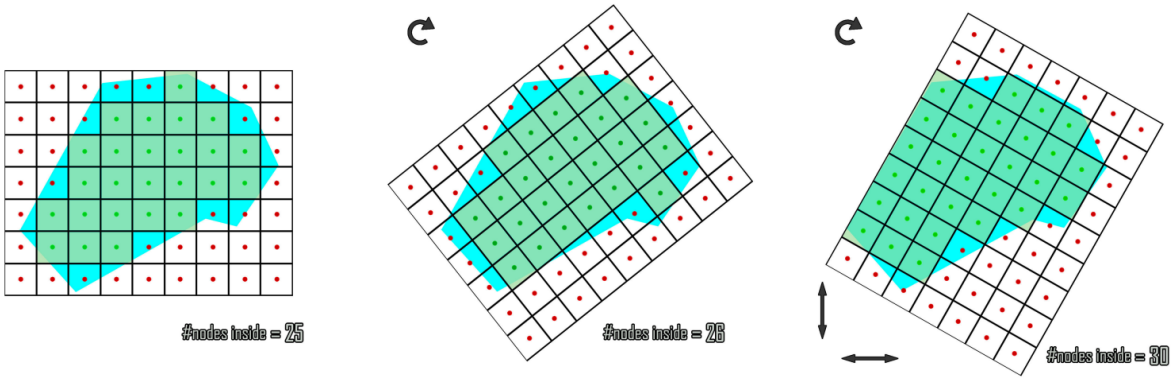



Figure 3.2: The polygon with its displacement limits

The algorithm explores different configurations to find the optimal node layout and enhance mission coverage. To evaluate each configuration's quality, an optimization index is calculated based on three factors: Maximizing the number of nodes inside the ROI, better positioning to cover marginal areas and aligning the polygon with the grid to minimize boundary distances. This optimization ensures that UAVs follow more efficient paths, reducing unnecessary maneuvers and improving mission efficiency.




(a) Initial placement (b) Rotation over grid (c) Shift and rotation over grid

Figure 3.3: The grid optimization process

### 3.1.2. Region of interest division (DARP)

This section describes the part of the algorithm responsible for the decomposition of the area, first for homogeneous swarms and then for heterogeneous swarms, to ensure greater clarity.

- **DARP for homogeneous multi-UAV systems**

  The DARP (Divide Areas based on Robots' Initial Positions) algorithm is a method that partitions a given terrain into exclusive sub-regions for multiple robots. The algorithm ensures that each operational robot is assigned a unique, connected, and fairly distributed area for coverage (defined as a set of nodes $L_i$) respecting the following conditions:

  1. $L_i \cap L_j = \emptyset, \quad \forall i, j \in \{1, \ldots, nr\}, \quad i \neq j$
  2. $L_1 \cup L_2 \cup \cdots \cup L_{nr} = L$
  3. $|L_1| \approx |L_2| \approx \cdots \approx |L_{nr}|$
  4. $L_i$ is connected, $\quad \forall i \in \{1, \ldots, nr\}$
  5. $\chi_i(t_0) \in L_i$

  To achieve this, the algorithm employs a cell-to-robot assignment scheme based on evaluation matrices. Each robot $i$ maintains an evaluation matrix $E_i$, which measures the reachability of each cell in the terrain measuring the distance from the robot's initial position $\chi_i(t_0)$ and the cell in meters . Using this evaluation, an assignment matrix $A$ is created, defining the cell-to-robot allocation:

  $$A_{x,y} = \arg \min_{i \in \{1,\ldots,nr\}} E_{i|x,y}, \quad \forall (x, y) \in L \tag{3.3}$$

  where $L$ represents the set of available cells in the terrain.

  Once the assignment matrix is established, the operational region $L_i$ for each robot is determined:

  $$L_i = \{(x, y) \in L : A(x, y) = i\}, \quad \forall i \in \{1, \ldots, nr\} \tag{3.4}$$

  This assignment process ensures that:

  – Each cell belongs to a single robot, preventing overlaps.

- Every available cell is assigned to at least one robot.
- Robots start from their designated areas.

However, the challenge lies in ensuring that the areas are equally divided, spatially connected, and optimized for minimal path coverage.

Initially, each robot's evaluation matrix $E_i$ is based purely on distance metrics (e.g., Euclidean distance). This results in an initial Voronoi-based partitioning of the space. However, this direct division may not result in an equal workload distribution.

To refine this, a correction factor $m_i$ is introduced, modifying the evaluation matrix:

$$E_i = m_i E_i \tag{3.5}$$

where $m_i$ is dynamically updated using gradient descent to balance the number of cells assigned to each robot. The objective function guiding this balance is:

$$J = \frac{1}{2} \sum_{r=1}^{nr} (k_i - f)^2 \tag{3.6}$$

where $k_i$ is the number of cells assigned to robot $i$, and $f$ is the fair share of cells per robot ($f = \frac{l}{nr}$).

Using standard gradient descent method, each robot iteratively updates its correction factor $m_i$ to reach a balanced cell distribution.

$$m_i = m_i - \eta \frac{\partial J}{\partial m_i}, \quad \eta > 0, \quad \forall i \in \{1, \ldots, nr\} \tag{3.7}$$

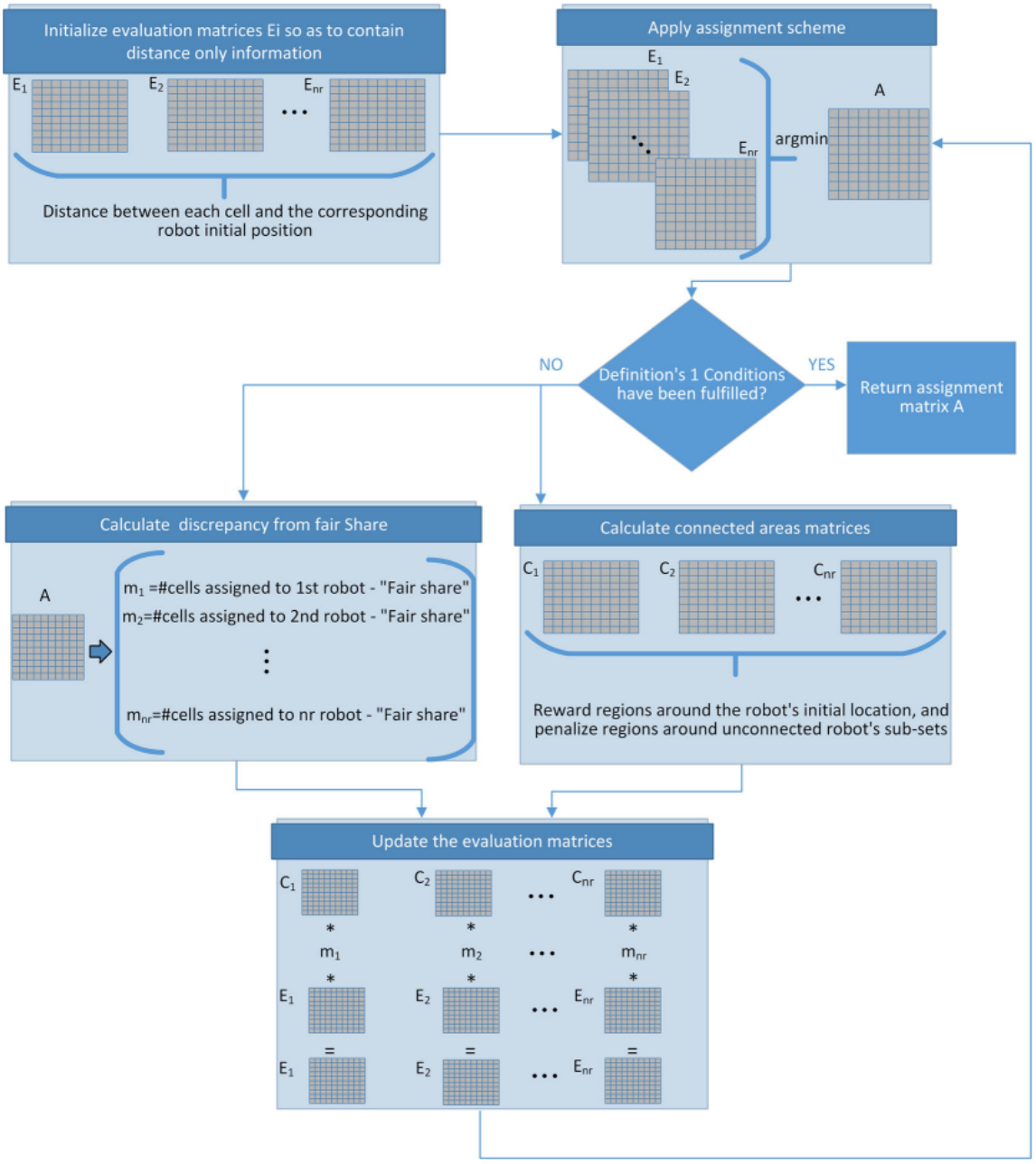



Figure 3.4: DARP algorithm flowchart

This ensures that all robots receive a nearly equal share of the operational terrain.

Although the space is now divided evenly, a critical issue remains: ensuring each robot's assigned area is a single connected region. Without this constraint, robots may be assigned disjoint sub-regions, leading to inefficient navigation.

To enforce spatial connectivity, the algorithm introduces a connectivity matrix $C_i$ for each robot. This matrix modifies the assignment process by penalizing disconnected

cells:

$$C_i|_{x,y} = \min(||[x,y] - r||) - \min(||[x,y] - q||) \tag{3.8}$$

where:

- $r \in R_i$ represents cells in the main connected region for robot $i$.
- $q \in Q_i$ represents disconnected cells assigned to robot $i$.

The connectivity matrix adjusts the evaluation function by promoting cells near the robot's main region while discouraging assignments to isolated areas. The final correction to the evaluation matrix is:

$$E_i = C_i \circ (m_i E_i) \tag{3.9}$$

where $\circ$ represents element-wise multiplication. This ensures that each robot's assigned region forms a spatially contiguous cluster.

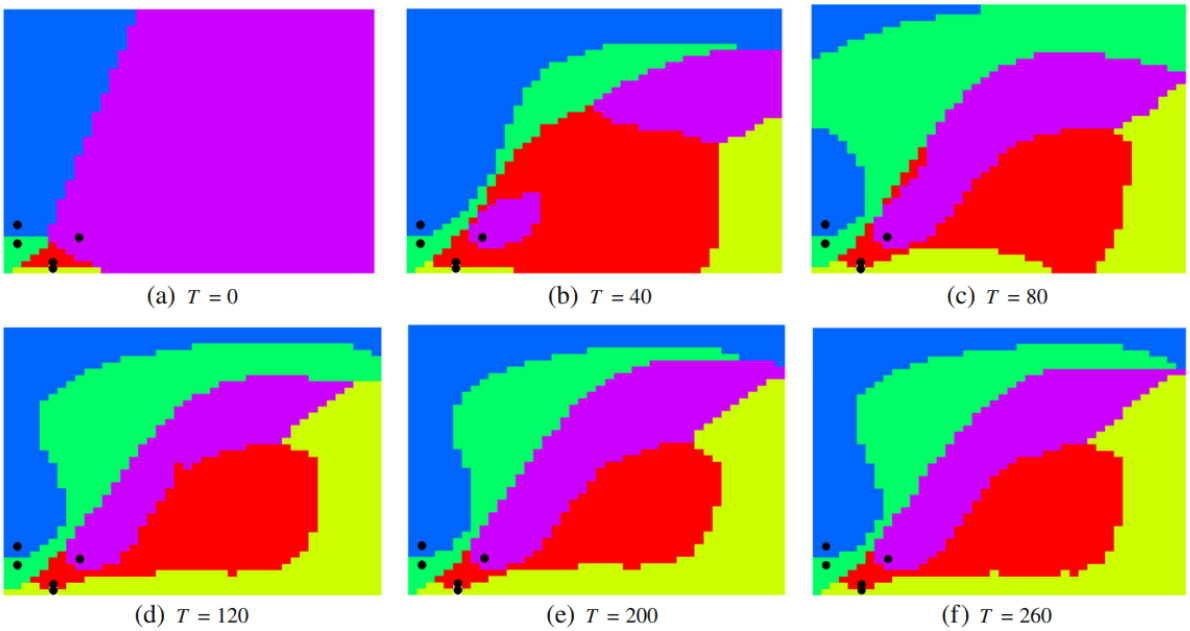



Figure 3.5: DARP Area subdivision over iterations

- **DARP for heterogeneous multi-UAV systems**

  To enable DARP to deal with different proportional area allocation, allowing UAVs with different energy levels or operational capacities to receive customized work-

loads based on user-defined percentages, the objective function previously used $J$ is redifined in this way:

$$J = \frac{1}{2}\sum_{i=1}^{n_r}(k_i - p_i)^2, \quad \sum_{i=1}^{n_r} p_i = 1 \tag{3.10}$$

where:

- $J$ is the cost function to minimize,
- $n_r$ is the number of UAVs,
- $k_i$ is the portion of the area assigned to UAV $i$,
- $p_i$ is the target proportion for UAV $i$,
- The sum of all $p_i$ values must equal 1 (100% of the area is allocated).

This proportional area allocation allows UAVs to share the workload based on their battery life, sensor coverage, and flight capabilities, ensuring an optimized mission execution.

### 3.1.3. Path generation (MST)

Once the DARP algorithm assigns an exclusive area to each UAV, the Single-Robot Spanning Tree Coverage (STC)[6] algorithm is used to generate paths. First, an MST (Minimum Spanning Tree) is created for each sub-region, and then a coverage path is computed based on the tree structure.

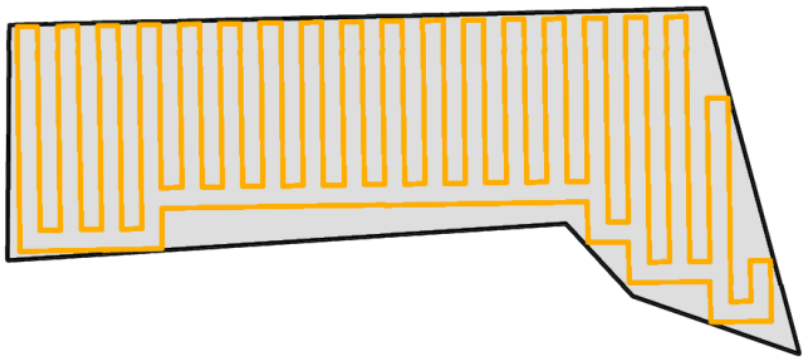
However, unoptimized MST paths tend to have excessive turns, increasing flight time and energy consumption. To reduce turns and optimize the path, the following strategy is implemented:

1. Testing four different MST configurations for each sub-region:
   - Upper MST connection
   - Lower MST connection
   - Right-most MST connection
   - Left-most MST connection
2. Selecting the MST configuration that results in the least number of turns, reducing unnecessary movement.

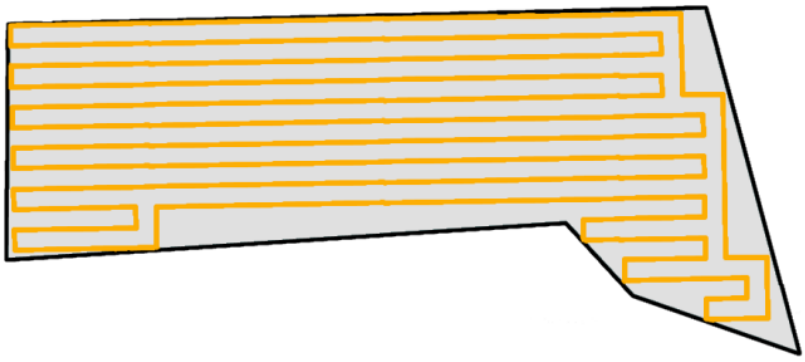
(a) Upper (b) Lower (c) Most right (d) Most left

Figure 3.6: The four differente MST configuration

This approach significantly enhances UAV efficiency, leading to shorter flight paths, lower energy consumption, reduced mission duration and more uniform coverage of the ROI.



(a) ROI coverage with 84 turns



(b) ROI coverage with 42 turns

Figure 3.7: An example to show the effect on the number of turns in using different MST configuration

# 4 | Trajectory smoothing

Once the waypoints have been defined, the final step before the mission can be considered complete and ready for execution is trajectory generation. This process aims to interpolate the waypoints using continuous and differentiable curves in order to obtain feasible paths that comply with the kinematic and dynamic constraints of the system.

In the case of quadrotors, which are differentially flat systems [5, 9], trajectory planning can be significantly simplified. Differential flatness allows all system states and inputs to be expressed as functions of a set of flat outputs and a finite number of their derivatives. This property enables the generation of polynomial trajectories in the space of flat outputs, ensuring continuity up to a desired order of derivatives. As a result, the trajectories are smooth and dynamically feasible.

Therefore, trajectory generation constitutes a crucial step in ensuring the safe, smooth, and efficient execution of the mission within the intended operational environment.

## 4.1. Description

The input required to generate the trajectory are the following:

- $v_{mean}$: mean velocity during the mission.
- $v_{max}$: maximum velocity during the mission.
- $v_{min}$: minimum velocity during the mission.
- $a_{max}$: maximum acceleration during the mission.
- $a_{min}$: minimum acceleration during the mission.
- $lat_0$: latitude of the take-off/landing point.
- $lon_0$: longitude of the take-off/landing point.
- $P$: order of the polynomial used in the interpolation.

- $t$: the period of time in seconds between the previous and the following point in the trajectory.

After providing the necessary input data, the trajectory is generated in such a way that the mean velocity between each pair of consecutive waypoints remains constant. Additionally, the generation process ensures that the resulting trajectory does not violate any predefined velocity or acceleration constraints imposed by the system.

## 4.2. Problems and solutions

Transitioning directly from the waypoints generated by planning algorithms to trajectory generation is not always feasible. This is primarily because such algorithms typically produce a waypoint only when a change in direction occurs. As a result, in algorithms like DARP, the distance between consecutive waypoints can span several hundred meters. This large spacing can introduce significant challenges during the interpolation phase, potentially leading to poorly shaped or highly distorted trajectories, as illustrated in the following figure.

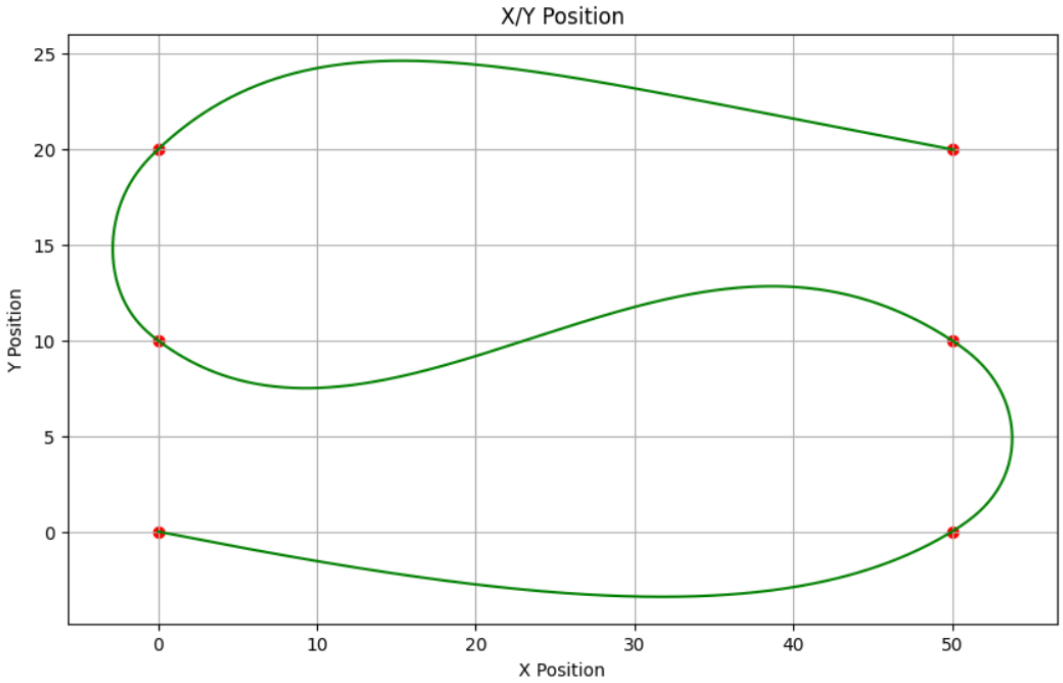
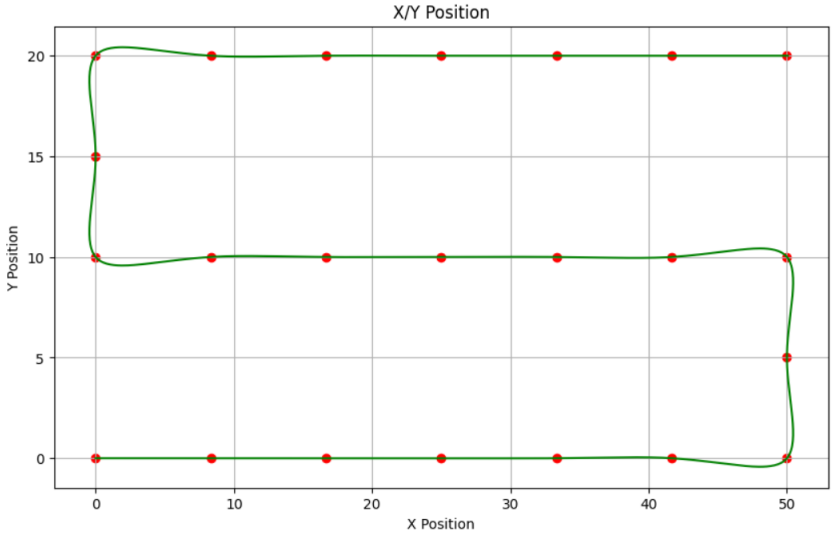


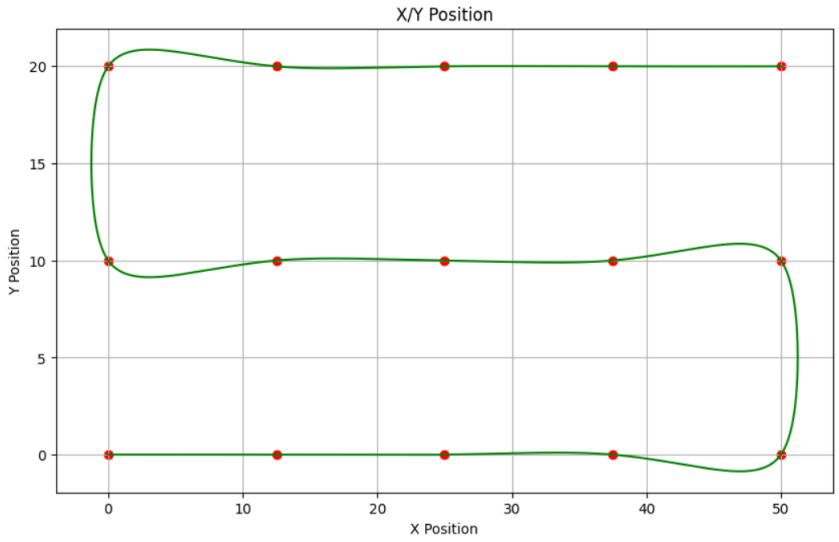

Figure 4.1: An example of trajectory distorted due to the distances between waypoints.

To address this issue, a straightforward solution involves modifying the input waypoints by adding intermediate points at shorter intervals. This densification of the waypoint

set helps to reduce interpolation errors and limits the extent of trajectory deformation, resulting in smoother and more accurate path generation.




Waypoint added if the interval is less than 10m.




Waypoint added if the interval is less than 15m.

Figure 4.2: Improvement for the previous trajectory adding new waypoints at different distances.

Due to the way the trajectory generation algorithm is designed, it is unfortunately not possible to insert waypoints that are too close to each other. Since the algorithm attempts to maintain a constant average velocity across each segment, excessively short segments, especially those following a change in direction, may result in situations where the drone is unable to achieve the required acceleration to meet the average speed con-

straint. This limitation must be carefully considered when refining the waypoint set to ensure both feasibility and adherence to the system's dynamic constraints. One possible approach to address this issue is to allocate time to each waypoint by formulating an additional optimization problem, or alternatively, by iteratively refining both the path and the corresponding timing.

# 5 | Simulation and testing

This chapter presents a series of tests designed to evaluate the performance and limitations of the DARP algorithm. The tests are divided into two main groups. The first group focuses on a comparative analysis between DARP and another previously discussed algorithm, Popcorn, to highlight the respective strengths and weaknesses of each method. This comparison is conducted on two test fields: the first is a perfectly rectangular area measuring 200 meters in length and 100 meters in width; the second features a more complex geometry and covers a larger area.

The second part of the testing is dedicated to evaluating characteristics unique to DARP that are not directly comparable with Popcorn, specifically its capability to handle obstacles and manage heterogeneous swarms. This set of tests also includes two scenarios: the first involves a field with two pylons that the drones must avoid by generating feasible paths, while the second is a simpler geometric area where the drones are required to manage different surface sections, each assigned to specific units within the swarm.

For all tests, the main parameters on which the comparison will be based are as follows: the ability to cover the field as completely as possible, adherence to the drones' flight time limits without being overly conservative, the length of the generated path (excluding the distance from the start/finish points), the flight time of the mission (considering the distance from the start/finish points), the number of turns in the mission, and the computational time required to generate the waypoints.

## 5.1. Parameters

The parameters required to perform the tests are the following.

### DARP

- The number of robots/vehicles.
- The desired scanning density (scanning density corresponds to the desired distance

between two sequential paths in meters).

- A polygon Region of Interest (ROI), formatted in WGS84 coordinate system.
- A set of obstacles (polygons formatted in WGS84 coordinate system) inside the ROI.*
- A boolean variable named pathsStrictlyInPoly, to select mode between (paths strictly in poly/better coverage). The "Strictly in polygon" option means that the path will remain entirely within the previously defined polygon, which carries the risk of leaving certain areas uncovered. In contrast, the "Better coverage" option allows the path to extend slightly outside the Region of Interest (ROI) to ensure better coverage of the target area.
- The initial positions of the vehicles. Must be equal to the number of drones. In case, the initial positions are not provided a random position inside the ROI will be used. Random position can be used if the initial position specified are to close to each other or not optimally placed.*
- The desired percentages for proportional area allocation. Must be equal to the number of drones. In case the percentages are not provided equal will be used instead.*

These data must be specified in a JSON file. An asterisk (*) indicates optional inputs.

## Popcorn

- A polygon Region of Interest (ROI), formatted in WGS84 coordinate system in a CSV file.
- The desired scanning density (scanning density corresponds to the desired distance between two sequential paths in meters).
- The flight altitude in meters.
- The mean velocity of the mission in meters per second.
- The initial position of the vehicles.
- The maximum flight time of the drone in seconds.

As can be observed, this algorithm is not capable of handling obstacles or heterogeneous swarms; however, it ensures that the generated paths are completed within a specified maximum time limit. The number of UAVs is not fixed in advance; rather, it is determined

dynamically based on the time constraint, by generating the number of paths required to achieve complete coverage.

## 5.2. Comparison

As previously mentioned, this first group of tests is divided into two parts. The first part is conducted in an ideal, perfectly rectangular field measuring 200 meters in the north-south direction and 100 meters in width east-west, with a total area of 2.08 hectares free of any obstacles. After running the simulation with Popcorn, the same test is performed using DARP, employing the same number of drones as used by Popcorn to ensure the fairest possible comparison. The second part utilizes a real-world field with a fairly irregular geometry and a total area of 7.29 hectares, aimed at evaluating both the algorithm's ability to cover complex shapes and its effectiveness in optimizing the available flight time.

### 5.2.1. Rectangular field

#### Popcorn

As input, in addition to the CSV file containing the field polygon, the following parameters were used: a scanning density of 10 meters, a flight altitude of 10 meters, an average mission speed of 3 meters per second, and a maximum flight time of 5 minutes, expressed in seconds. The flight time constraint is intentionally set to a low value relative to the field dimensions, in order to produce multiple paths.

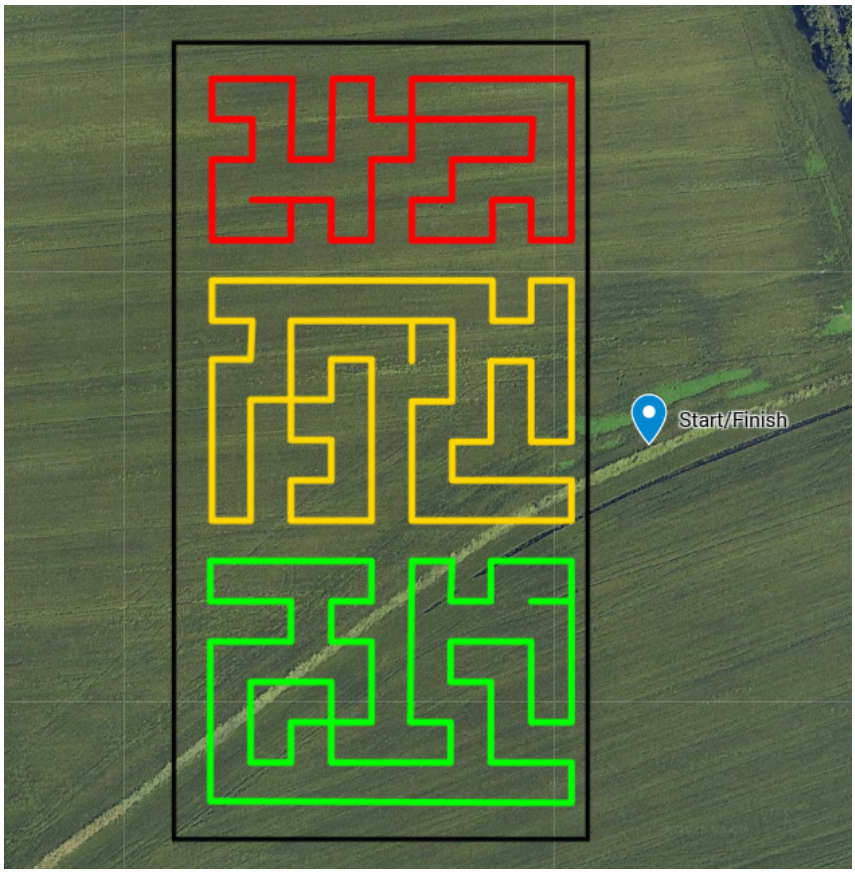



Figure 5.1: The generated paths with the selected takeoff and landing location

After running Popcorn, as shown in the figure above, three distinct paths were generated. Excluding the displacement between the start and end points, the length of the sum of the flight paths and the computational time to obtain these results are:

- Total length = 1960 m
- Computational time = 7 s

After obtaining the waypoints to generate the trajectories, the following parameters were used consistently across all tests: a mean velocity $v_{mean}$ of 3 m/s, a maximum speed $v_{max}$ and a minimum speed $v_{min}$ respectively of 5 m/s and -5 m/s, a maximum acceleration $a_{max}$ and minimum acceleration $a_{min}$ respectively of 4 m/s$^2$ and -4 m/s$^2$, the latitude and longitude of the Start/Finish point $lat_0$ and $lon_0$, the order of the polynomial to interpolate the waypoints $P$ equal to 5 and the temporal distance between two consecutive trajectory points of 0.05s. (The only changing parameter is the Start/Finish point)

The graphs shown below, for this test as well as the others, display only the trajectory of the first path generated. The other trajectories can be found in the appendix.

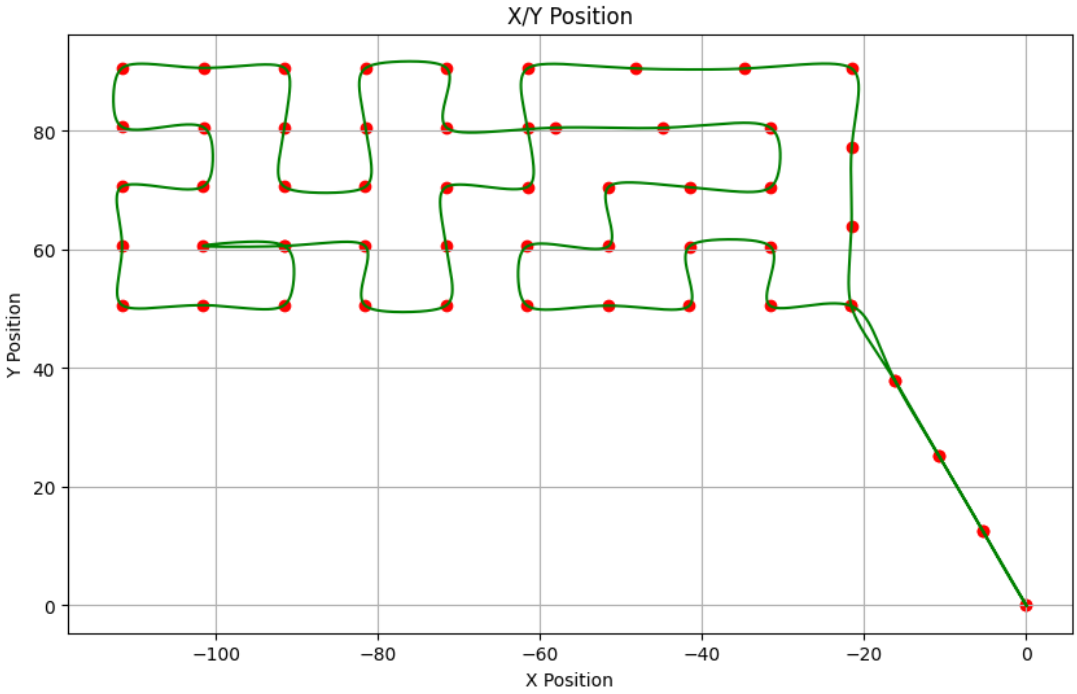



Figure 5.2: The position along x and y axis in meters of the drone

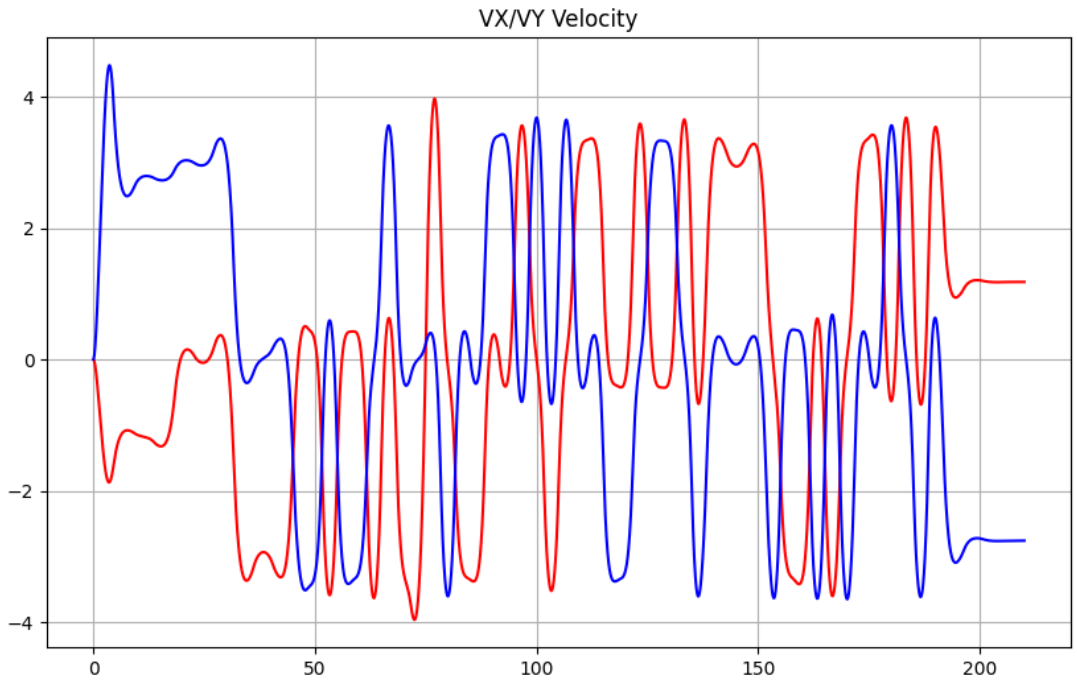



Figure 5.3: The x-axis represents the flight time in seconds, while the y-axis shows the drone's velocity (red for the x-direction, blue for the y-direction).

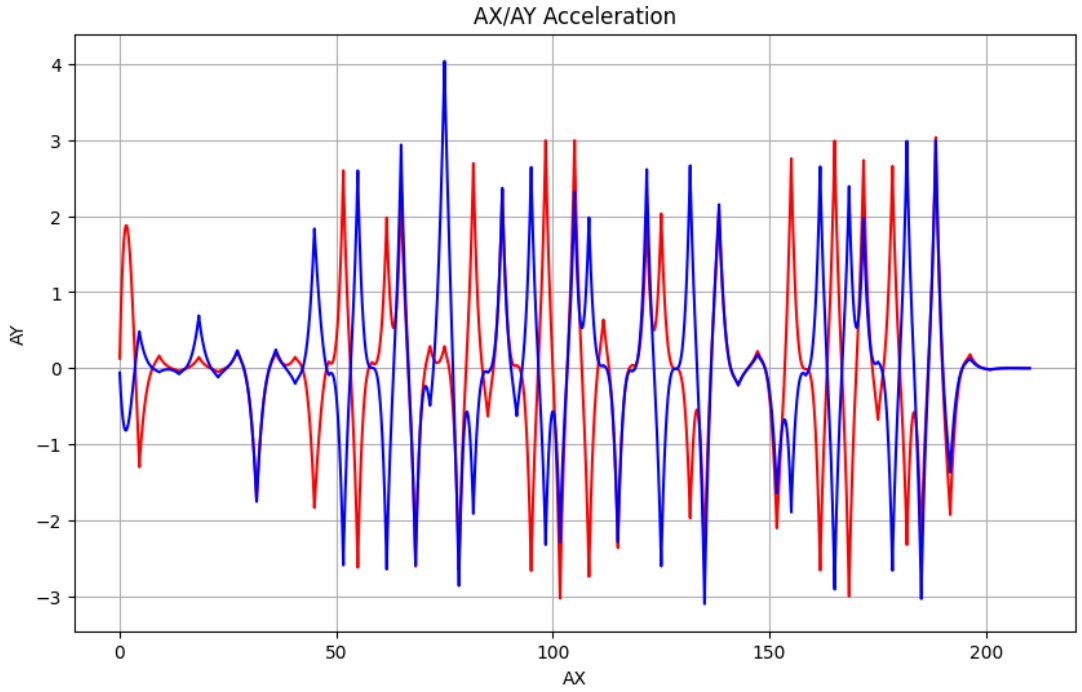



Figure 5.4: The acceleration along x and y axis in $m/s^2$ of the drone during time in seconds (red for the x-direction, blue for the y-direction)

Table 5.1: Popcorn Results

| Drone | Length (m) | Flight time (s) | Turns |
|---|---|---|---|
| Red | 520 | 210 | 35 |
| Green | 720 | 266 | 41 |
| Yellow | 720 | 256 | 40 |

## DARP

In this case, as input, in addition to the the field polygon coordinates, the following parameters were used: a number of drones equal to 3, a scanning density of 10 meters, the boolean variable set to true (StrictlyInPoly approach), the initial position set to random (not provided), and the desired area percentage of 33,3% for each drones.

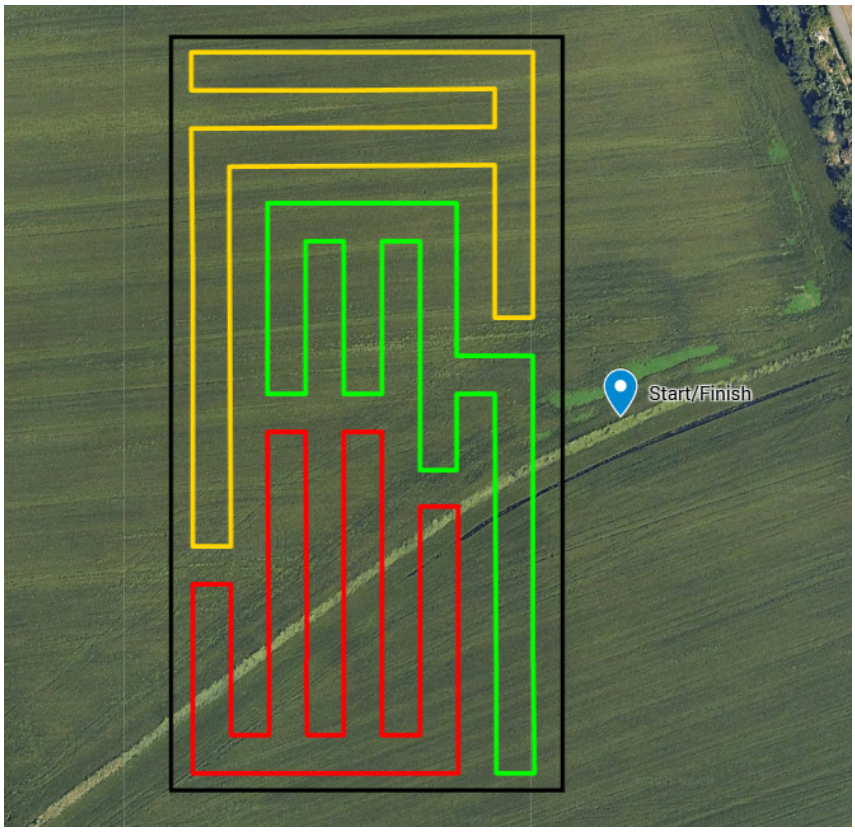



Figure 5.5: The generated paths with the selected takeoff and landing location

After running DARP, as shown in the figure above, three distinct paths were generated. Excluding the displacement between the start and end points, the length of the sum of the flight paths and the computational time to obtain these results are:

- Total length = 2002 m
- Computational time = 3 s

After generating the waypoints, the trajectories were generated as follows:

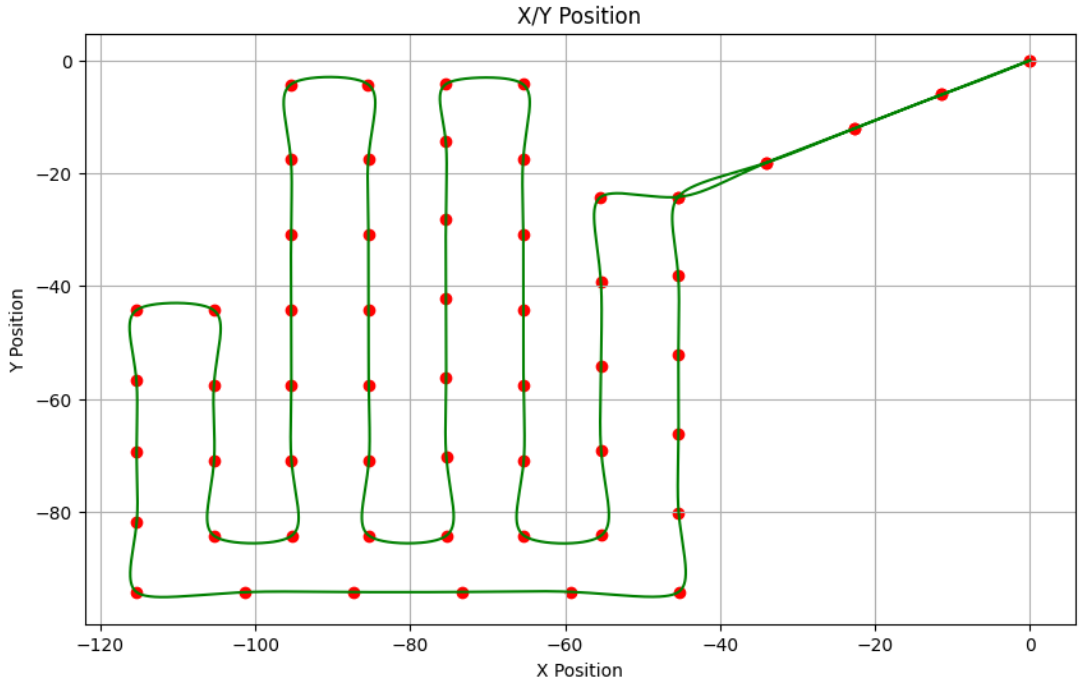



Figure 5.6: The position along x and y axis in meters of the drone

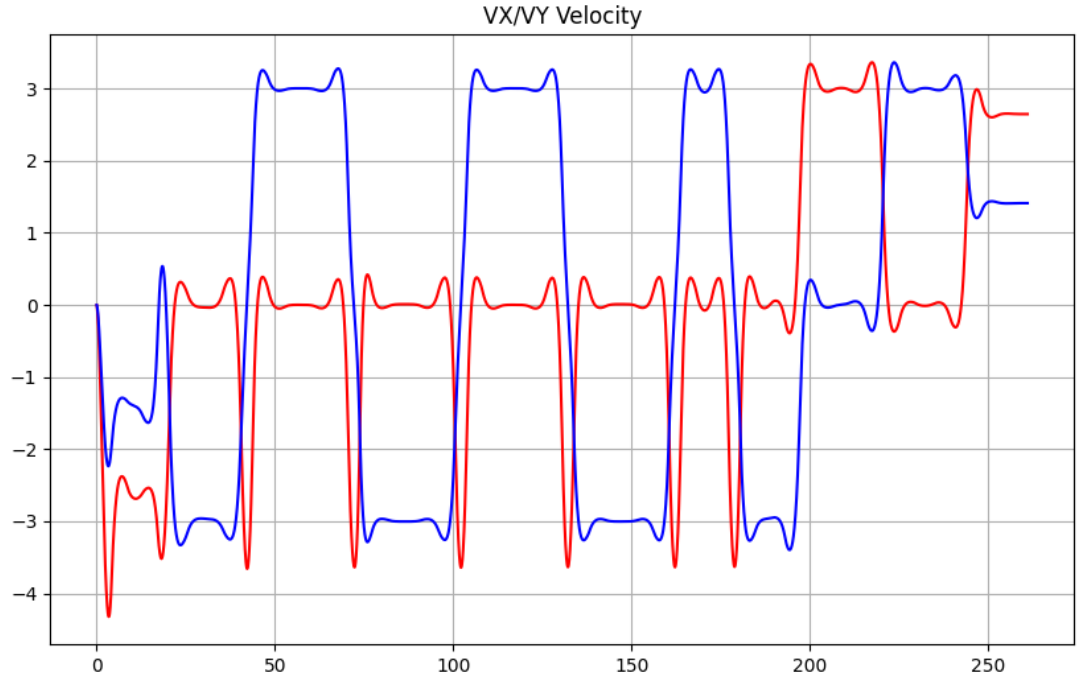



Figure 5.7: The x-axis represents the flight time in seconds, while the y-axis shows the drone's velocity (red for the x-direction, blue for the y-direction).

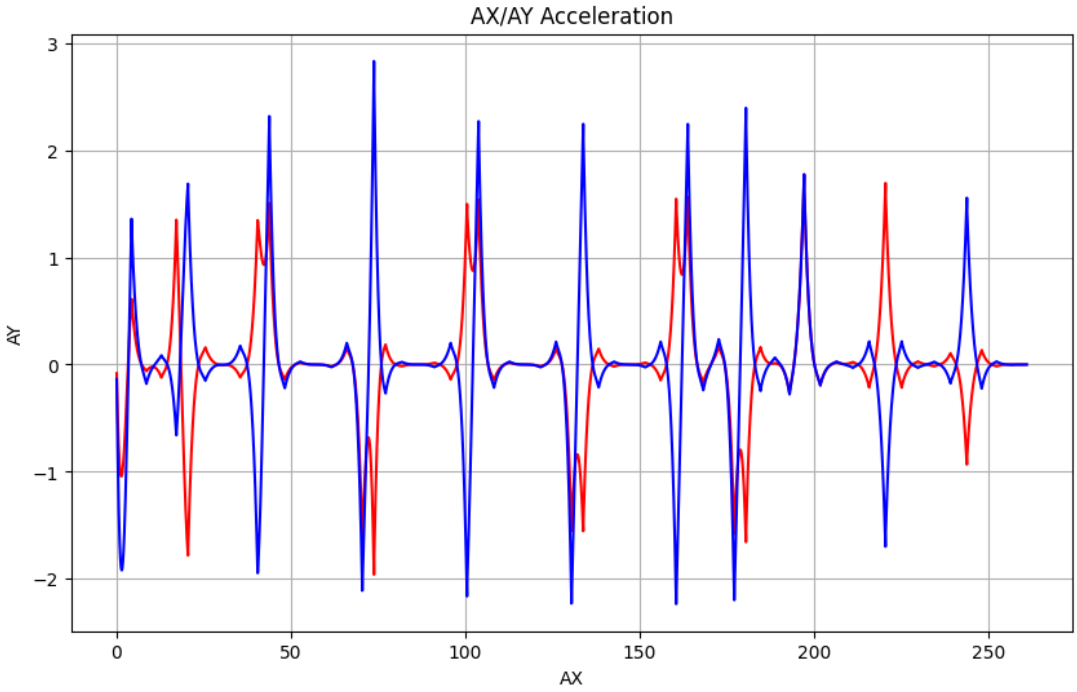



Figure 5.8: The acceleration along x and y axis in m/s$^2$ of the drone during time in seconds (red for the x-direction, blue for the y-direction)

Table 5.2: DARP Results

| Drone | Length (m) | Flight time (s) | Turns |
|---|---|---|---|
| Red | 681 | 262 | 18 |
| Green | 641 | 234 | 19 |
| Yellow | 680 | 252 | 14 |

## Results discussion

In these two tests, both methods demonstrated the ability to fully cover the assigned field while complying with the flight time limits. In terms of computational speed, Popcorn proved to be slightly slower than DARP, although the difference was minimal. Regarding the path length and the flight time, Popcorn and DARP showed comparable values, with Popcorn achieving slightly better performance overall. However, when it comes to the number of turns, Popcorn was significantly less efficient than DARP, exhibiting a considerably higher number of turns, this has a direct impact on the accelerations required and, consequently, on the energy efficiency of the mission.

Table 5.3: Results comparison

| Algorithm | Total Length (m) | Total Flight time (s) | Total Turns | Computational time (s) |
|---|---|---|---|---|
| Popcorn | 1960 | 732 | 116 | 7 |
| DARP | 2002 | 748 | 51 | 3 |

### 5.2.2. Complex field

For this test, a cluster of fields with a total area of 7.29 hectares was selected, located in the Parco Agricolo Sud within the municipality of Milan. Unlike the first comparison, this field is significantly larger and more irregular. The aim is to assess the ability of both systems to cover the area as completely as possible while also optimizing the efficiency of the path in terms of length, adherence to the flight time limit set to 20 minutes per drone, and minimization of the number of turns.

#### Popcorn

As input, in addition to the CSV file containing the field polygon, the following parameters were used: a scanning density of 10 meters, a flight altitude of 10 meters, an average mission speed of 3 meters per second, and a maximum flight time of 20 minutes, expressed in seconds.

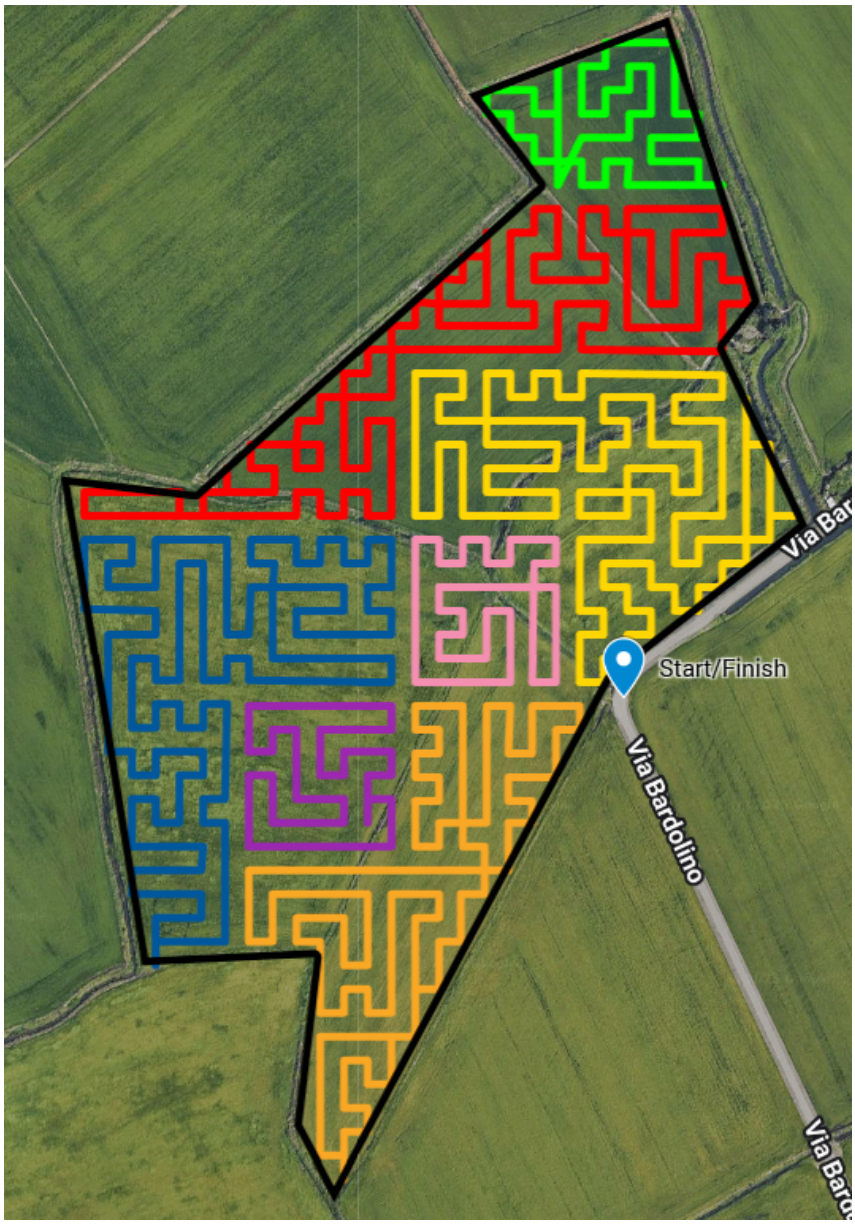



Figure 5.9: The generated paths with the selected takeoff and landing location

After running Popcorn, as shown in the figure above, seven distinct paths were generated. Excluding the displacement between the start and end points, the length of the sum of the flight paths and the computational time to obtain these results are:

- Total length = 7593 m
- Computational time = 239 s

After generating the waypoints, the trajectories were generated as follows:

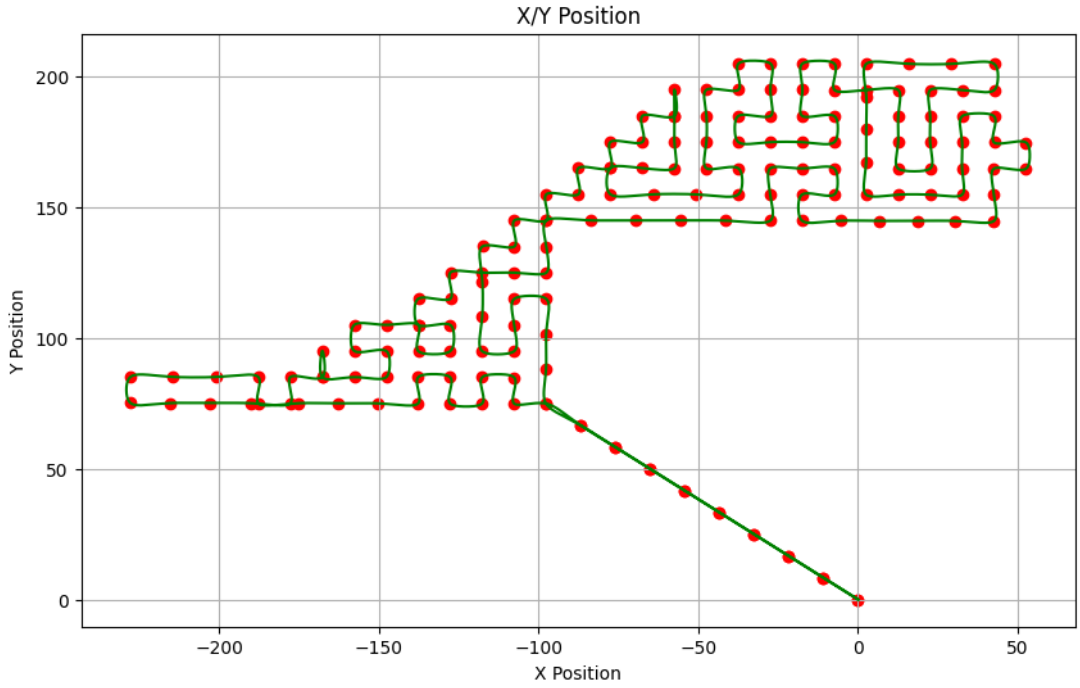



Figure 5.10: The position along x and y axis in meters of the drone

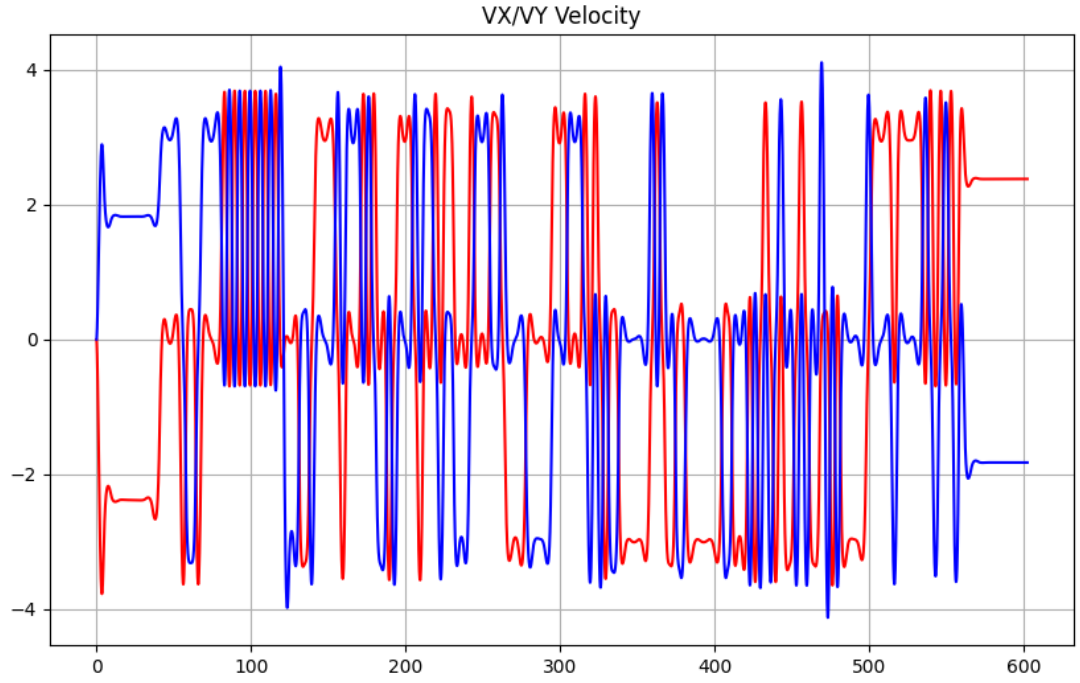



Figure 5.11: The x-axis represents the flight time in seconds, while the y-axis shows the drone's velocity (red for the x-direction, blue for the y-direction).

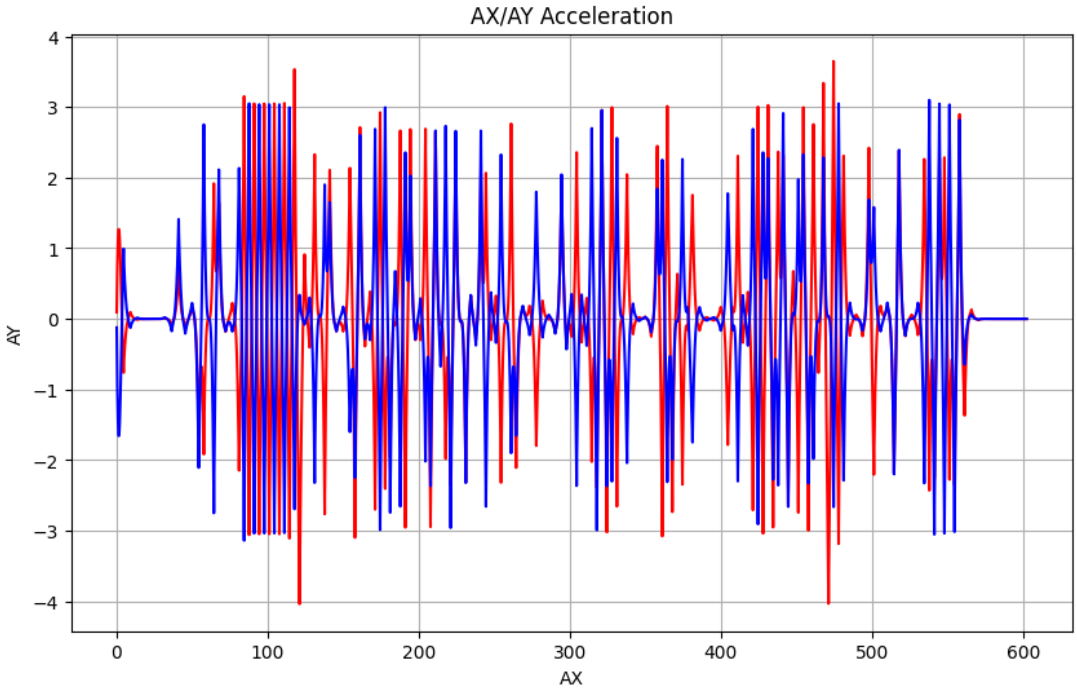



Figure 5.12: The acceleration along x and y axis in m/s$^2$ of the drone during time in seconds (red for the x-direction, blue for the y-direction)

Table 5.4: Popcorn Results

| Drone | Length (m) | Flight time (s) | Turns |
|---|---|---|---|
| Red | 1560 | 605 | 91 |
| Green | 593 | 342 | 40 |
| Yellow | 1540 | 522 | 91 |
| Blue | 1540 | 582 | 88 |
| Orange | 1360 | 468 | 76 |
| Purple | 500 | 234 | 26 |
| Pink | 500 | 186 | 27 |

## DARP

In this case, as input, in addition to the the field polygon coordinates, the following parameters were used: a number of drones equal to 2, a scanning density of 10 meters, the boolean variable set to true (StrictlyInPoly approach), the initial position set to random (not provided), and the desired area percentage of 50% for each drones.


Figure 5.13: The generated paths with the selected takeoff and landing location

After running DARP, as shown in the figure above, two distinct paths were generated. Excluding the displacement between the start and end points, the length of the sum of the flight paths and the computational time to obtain these results are:

- Total length = 6640 m
- Computational time = 12 s

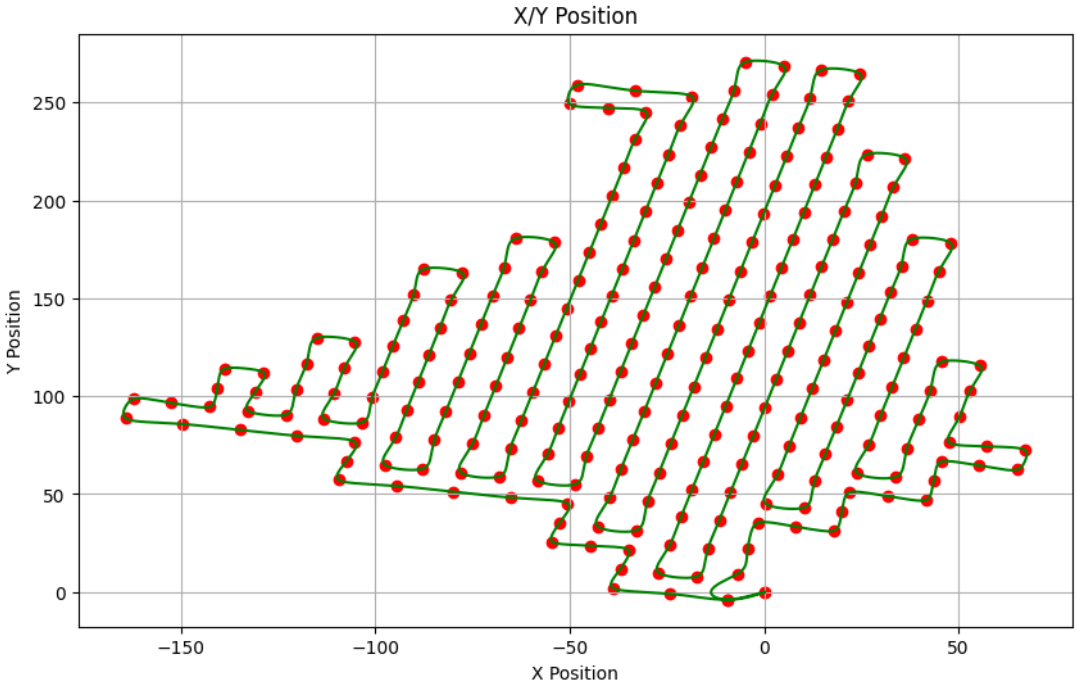



Figure 5.14: The position along x and y axis in meters of the drone

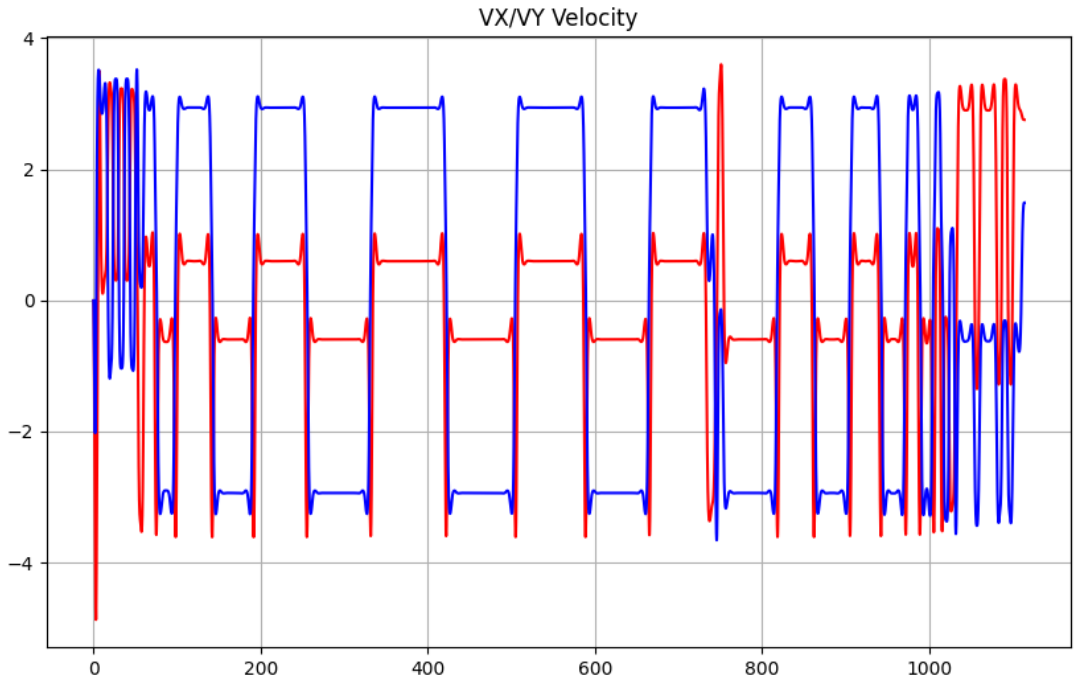



Figure 5.15: The x-axis represents the flight time in seconds, while the y-axis shows the drone's velocity (red for the x-direction, blue for the y-direction).

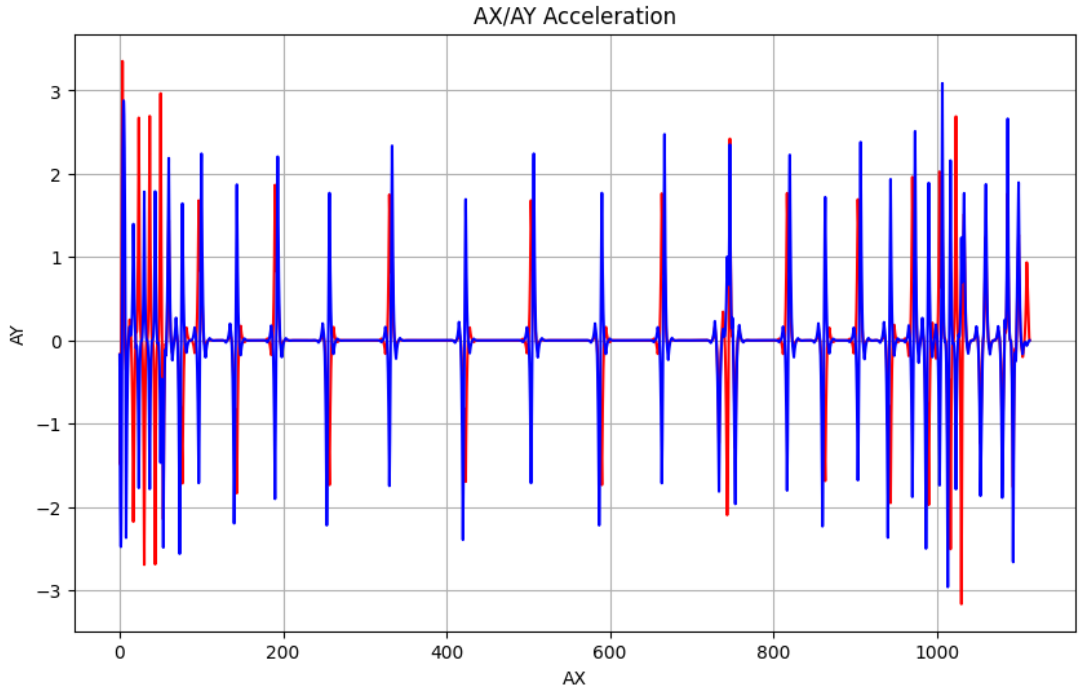



Figure 5.16: The acceleration along x and y axis in m/s$^2$ of the drone during time in seconds (red for the x-direction, blue for the y-direction)

Table 5.5: DARP Results

| Drone | Length (m) | Flight time (s) | Turns |
|---|---|---|---|
| Red | 3320 | 1117 | 60 |
| Green | 3320 | 1132 | 52 |

## Results discussion

In this second series of tests, unlike the first, both methods revealed their respective strengths and weaknesses. In terms of computational time, DARP demonstrated significantly superior performance compared to Popcorn, with the former operating on the order of seconds and the latter on the order of minutes. Both methods were able to cover the field while respecting the flight time limits; however, Popcorn adopted an overly conservative approach, generating a large number of trajectories often with durations below 10 minutes. Regarding field coverage, Popcorn achieved better results than DARP, as DARP failed to fully cover the area, leaving some regions near the boundaries uncovered. However, in terms of trajectory length, flight time, and especially the number of turns, DARP again outperformed Popcorn. In summary, aside from the field coverage metric, DARP proved superior to Popcorn across all other evaluated parameters.

Table 5.6: Results comparison

| Algorithm | Total Length (m) | Total Flight time (s) | Total Turns | Computational time (s) |
|---|---|---|---|---|
| Popcorn | 7593 | 2939 | 439 | 239 |
| DARP | 6640 | 2249 | 112 | 12 |

## 5.3. Obstacles management and different partitions tests

In this second group of tests, features of DARP that cannot be directly compared with Popcorn but are essential to meeting the requirements of this work will be evaluated. Specifically, the first test will assess DARP's ability to handle obstacles, while the second will evaluate its ability to manage different percentages of the field assigned to each drone. In the second test, the "better coverage" mode will also be tested and compared to the "strictly in poly" version, in order to assess its improvement in field coverage; which was identified as DARP's main weakness in the previous tests.

### 5.3.1. Obstacles field

For this test, a cluster of fields was selected, also located in the Parco Agricolo Sud, with a total area of 9.79 hectares and two pylons situated in the middle of the field. In this case, as input, in addition to the the field polygon coordinates, the following parameters were used: a number of drones equal to 3, a scanning density of 10 meters,the coordinates of the polygons containing the pylons, the boolean variable set to true (StrictlyInPoly approach), the initial position set to random (not provided), and the desired area percentage of 33.3% for each drones.

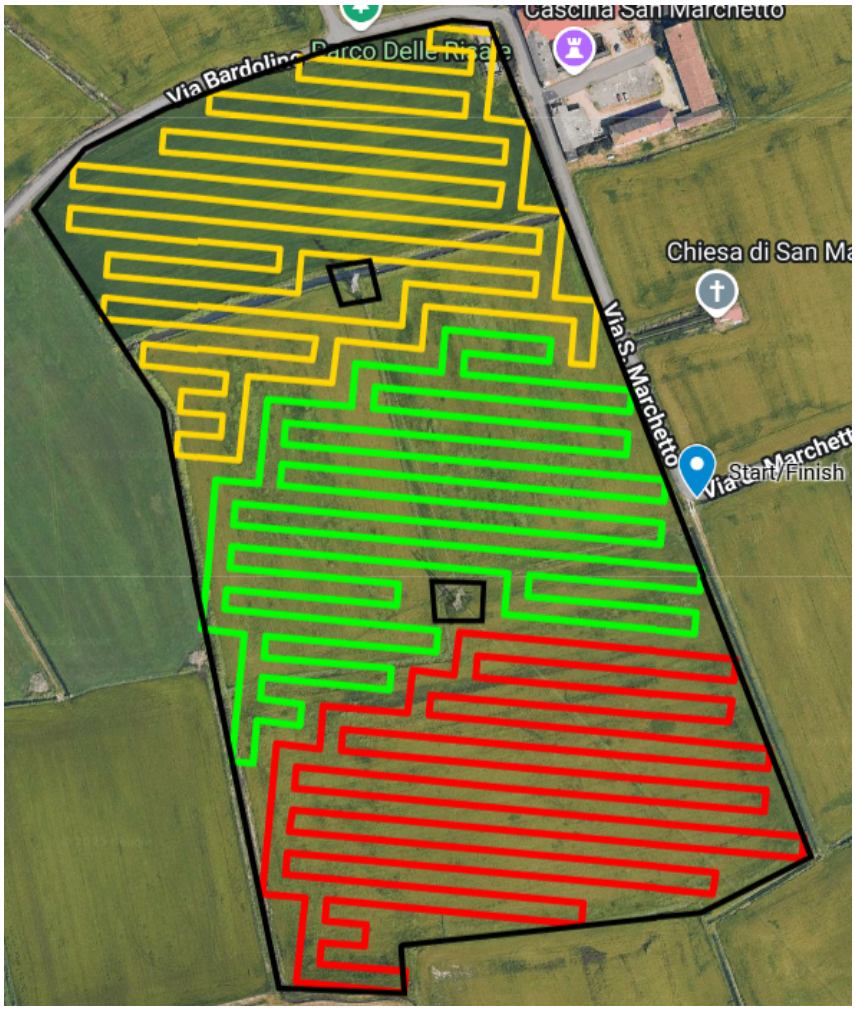



Figure 5.17: The generated paths with the selected takeoff and landing location and obstacles

After running DARP, as shown in the figure above, three distinct paths were generated. Excluding the displacement between the start and end points, the length of the sum of the flight paths and the computational time to obtain these results are:

- Total length = 9040 m
- Computational time = 26 s

In this case, as well as for the following test, it was not considered necessary to include the trajectories directly in the main text, since unlike the first group of tests, they are not directly useful for comparison with another method. Therefore, only the summary table reporting the method's performance will be presented.

Table 5.7: DARP Results

| Drone | Length (m) | Flight time (s) | Turns |
|---|---|---|---|
| Red | 3000 | 1058 | 46 |
| Green | 3000 | 1011 | 58 |
| Yellow | 3040 | 1067 | 64 |

### Results discussion

In this test, DARP demonstrated its ability to effectively handle no-fly zones while still ensuring good, although not perfect, field coverage. At the same time, it was able to comply with the flight time constraints (20 minutes) and consistently delivered performance in line with the other tests in terms of the number of turns, path length, and flight time.

## 5.3.2. Non-homogeneous swarm and better coverage test

For this test, a single field with a slightly irregular geometry was used, again located in the Parco Agricolo Sud, with a total area of 4.89 hectares. The pair of tests conducted on this field will differ by only one input parameter, namely the mode, which in the second case will be set to false activating the "better coverage" option. For the others inputs in addition to the the field polygon coordinates, the following parameters were used: a number of drones equal to 2, a scanning density of 10 meters, the initial position set to random (not provided), and the desired area percentage of 60% for the first drone and 40% for the second.


Figure 5.18: The generated paths with the selected takeoff and landing location with "strictly in poly" approach


Figure 5.19: The generated paths with the selected takeoff and landing location with "better coverage" approach

Table 5.8: Strictly in poly

| Drone | Length (m) | Real portion | Turns |
|---|---|---|---|
| Red | 2760 | 60 | 40 |
| Green | 1840 | 40 | 31 |

Table 5.9: Better coverage

| Drone | Length (m) | Real portion | Turns |
|---|---|---|---|
| Red | 2920 | 59.3 | 44 |
| Green | 2000 | 40.7 | 31 |

## Results discussion

In this final pair of tests, the algorithm demonstrated both its ability to accurately assign different portions of the field to each drone and the potential to further improve field coverage (from 92% to 97%) by switching modes. As for the other parameters, the results were consistent with the previous tests, showing no substantial variations with the change of mode.

# 6 | Conclusion

In conclusion, after testing the various features of DARP and comparing it with another equally valid method, Popcorn, it can be stated that despite its main weakness, leaving some small boundary areas unexplored, which can nevertheless be mitigated through the better coverage mode, DARP outperformed or matched Popcorn in the majority of tests across all evaluated parameters, particularly in terms of the number of turns and computational time as problem size increases.

Moreover, DARP offers unique capabilities that are not available in other methods such as Popcorn, including: guaranteeing that already-explored areas are never revisited, the ability to manage obstacles, and, most importantly for this study, the capacity to assign different portions of the field to each drone, thus enabling the management of heterogeneous swarms.

For these reasons, although not perfect, DARP is an excellent algorithm for both monitoring operations and spreading task. When combined with the previously presented trajectory generation system, it provides an efficient and reliable tool for precision agriculture applications.

# Bibliography

[1] S. Apostolidis, P. Kapoutsis, A. Kapoutsis, and E. Kosmatopoulos. Cooperative multi-uav coverage mission planning platform for remote sensing applications. *Autonomous Robots*, 46:1–28, 02 2022. doi: 10.1007/s10514-021-10028-3.

[2] S. Apostolidis, G. Vougiatzis, A. Kapoutsis, S. Chatzichristofis, and E. Kosmatopoulos. Systematically improving the efficiency of grid-based coverage path planning methodologies in real-world uavs' operations. *Drones*, 7, 06 2023. doi: 10.3390/drones7060399.

[3] T. Cabreira, C. Di Franco, and P. Ferreira Jr. Energy-aware spiral coverage path planning for uav photogrammetric applications. *IEEE Robotics and Automation Letters*, PP:1–1, 07 2018. doi: 10.1109/LRA.2018.2854967.

[4] D. Datsko, F. Nekovar, R. Pěnička, and M. Saska. Energy-aware multi-uav coverage mission planning with optimal speed of flight. *IEEE Robotics and Automation Letters*, PP:1–8, 03 2024. doi: 10.1109/LRA.2024.3358581.

[5] S. Formentin and M. Lovera. Flatness-based control of a quadrotor helicopter via feedforward linearization. In *2011 50th IEEE Conference on Decision and Control and European Control Conference*, pages 6171–6176, 2011. doi: 10.1109/CDC.2011.6160828.

[6] Y. Gabriely and E. Rimon. Spanning-tree based coverage of continuous areas by a mobile robot. *Annals of Mathematics and Artificial Intelligence*, 31:77–98, 02 2001. doi: 10.1023/A:1016610507833.

[7] J. Jena, S. Misra, and K. Tripathi. Normalized difference vegetation index (ndvi) and its role in agriculture. 11 2019.

[8] A. Kapoutsis, S. Chatzichristofis, and E. Kosmatopoulos. Darp: Divide areas algorithm for optimal multi-robot coverage path planning. *Journal of Intelligent and Robotic Systems*, 86, 06 2017. doi: 10.1007/s10846-016-0461-x.

[9] D. Mellinger and V. Kumar. Minimum snap trajectory generation and control for quadrotors. pages 2520 – 2525, 06 2011. doi: 10.1109/ICRA.2011.5980409.

[10] S. A. H. Mohsan, N. Oth Man, L. Yanlong, M. Alsharif, and M. Khan. Unmanned aerial vehicles (uavs): practical aspects, applications, open challenges, security issues, and future trends. *Intelligent Service Robotics*, 16:109–137, 01 2023. doi: 10.1007/s11370-022-00452-4.

[11] M. Oliver. *Geostatistical Applications for Precision Agriculture*. 01 2010. ISBN 978-90-481-9132-1. doi: 10.1007/978-90-481-9133-8.

[12] R. P S and D. Jeyan. Comparative analysis of fixed-wing, rotary-wing and hybrid mini unmanned aircraft systems (uas) from the applications perspective. *INCAS BULLETIN*, 14:137–151, 03 2022. doi: 10.13111/2066-8201.2022.14.1.12.

[13] J. Segarra. *Satellite Imagery in Precision Agriculture*, pages 325–340. 01 2024. ISBN 978-3-031-43547-8. doi: 10.1007/978-3-031-43548-5_10.

[14] K. Shah, G. Ballard, A. Schmidt, and M. Schwager. Multidrone aerial surveys of penguin colonies in antarctica. *Science Robotics*, 5:3000, 10 2020. doi: 10.1126/scirobotics.abc3000.

[15] A. Stefanopoulou, E. K. Raptis, S. Apostolidis, S. Gkelios, A. Kapoutsis, S. Chatzichristofis, S. Vrochidis, and E. Kosmatopoulos. Improving time and energy efficiency in multi-uav coverage operations by optimizing the uavs' initial positions. *International Journal of Intelligent Robotics and Applications*, 8:629–647, 04 2024. doi: 10.1007/s41315-024-00333-2.

[16] M. Szklany, A. Cohen, and J. Boubin. Tsunami: Scalable, fault tolerant coverage path planning for uav swarms. In *2024 International Conference on Unmanned Aircraft Systems (ICUAS)*, 06 2024. doi: 10.1109/ICUAS60882.2024.10556935.

[17] H. Wickstead and M. Barber. A spectacular history of survey by flying machine! *Cambridge Archaeological Journal*, 22, 02 2012. doi: 10.1017/S0959774312000054.

# A | Appendix

## Rectangular field / Popcorn / Drone 2

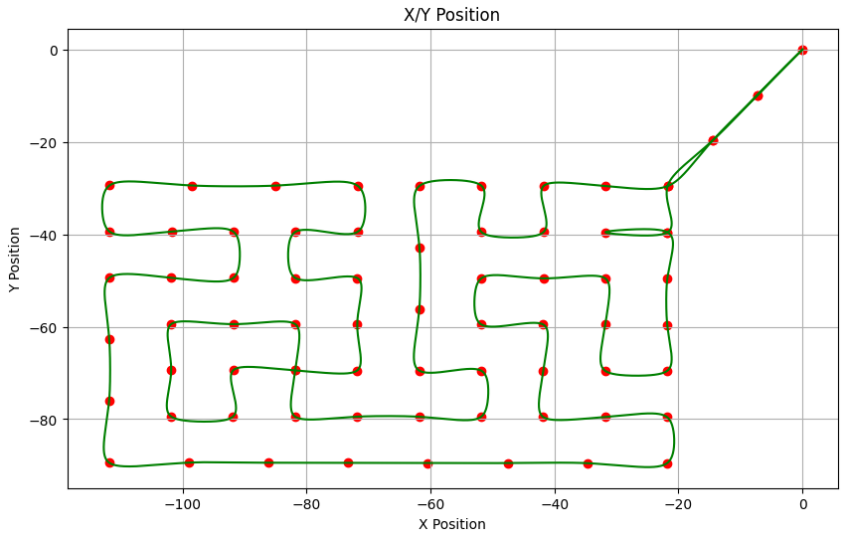



Figure A.1: The position along x and y axis in meters of the drone

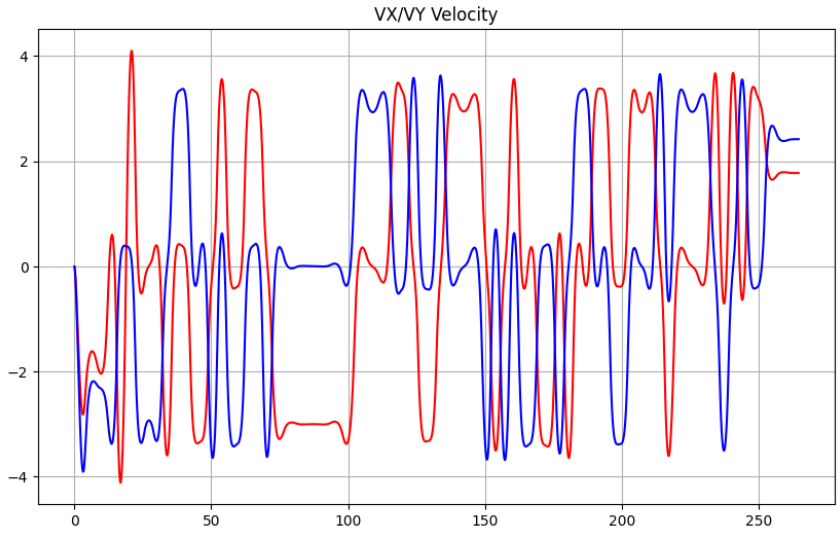



Figure A.2: The x-axis represents the flight time in seconds, while the y-axis shows the drone's velocity (red for the x-direction, blue for the y-direction).

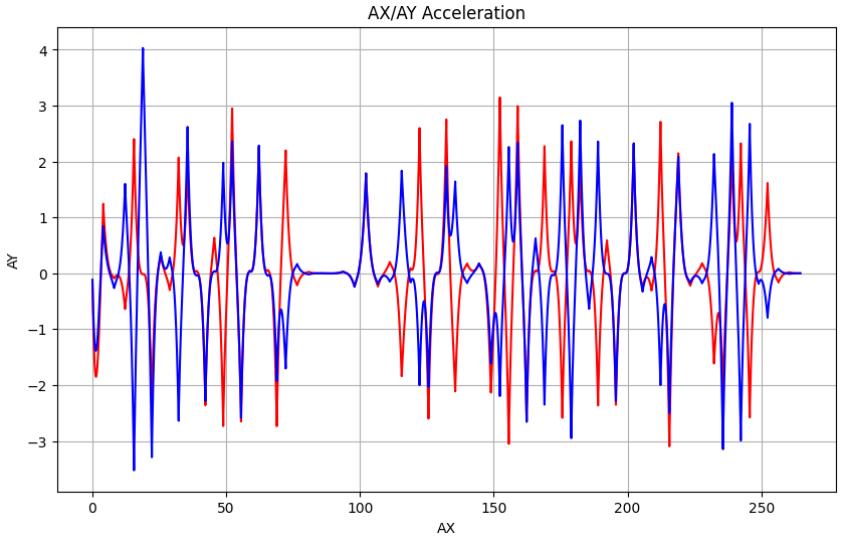



Figure A.3: The acceleration along x and y axis in $m/s^2$ of the drone during time in seconds (red for the x-direction, blue for the y-direction)

## Rectangular field / Popcorn / Drone 3

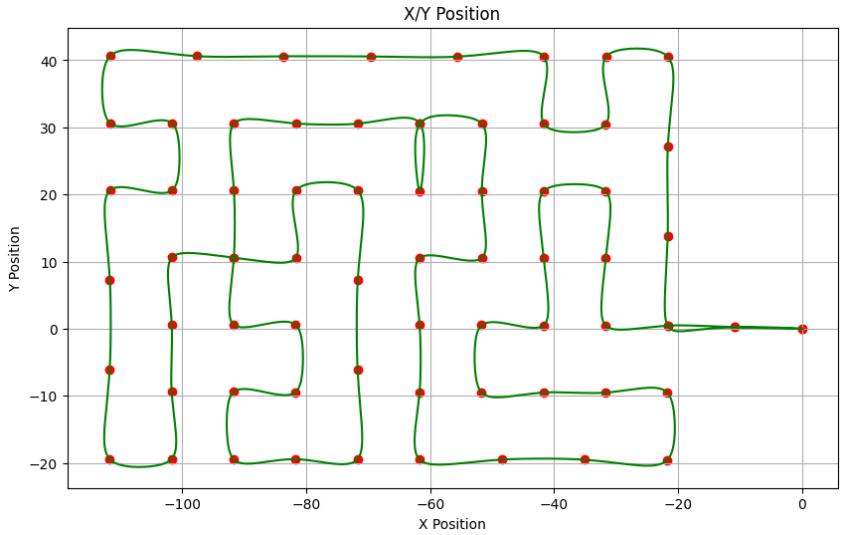



Figure A.4: The position along x and y axis in meters of the drone

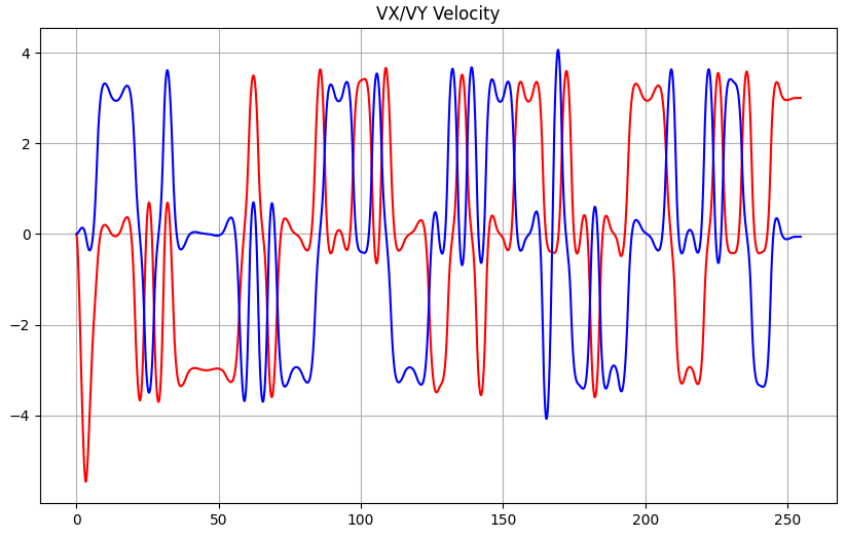



Figure A.5: The x-axis represents the flight time in seconds, while the y-axis shows the drone's velocity (red for the x-direction, blue for the y-direction).

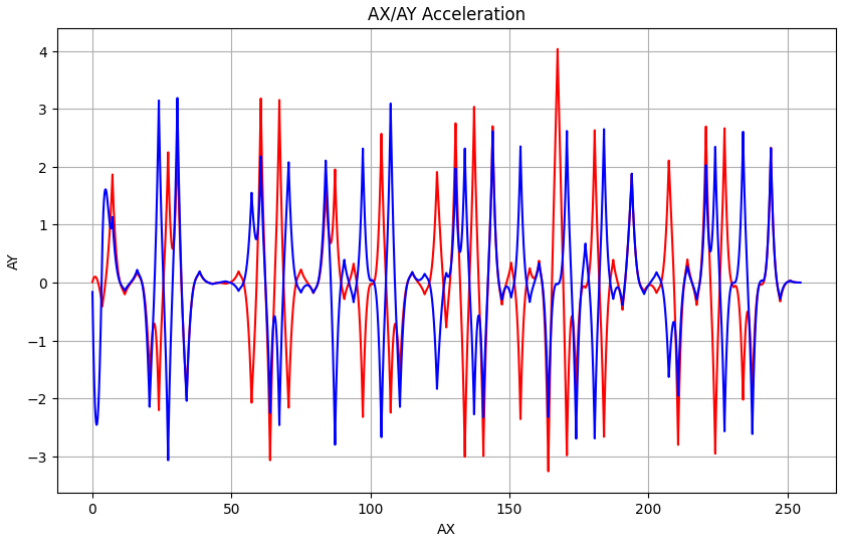



Figure A.6: The acceleration along x and y axis in m/s$^2$ of the drone during time in seconds (red for the x-direction, blue for the y-direction)

## Rectangular field / DARP / Drone 2

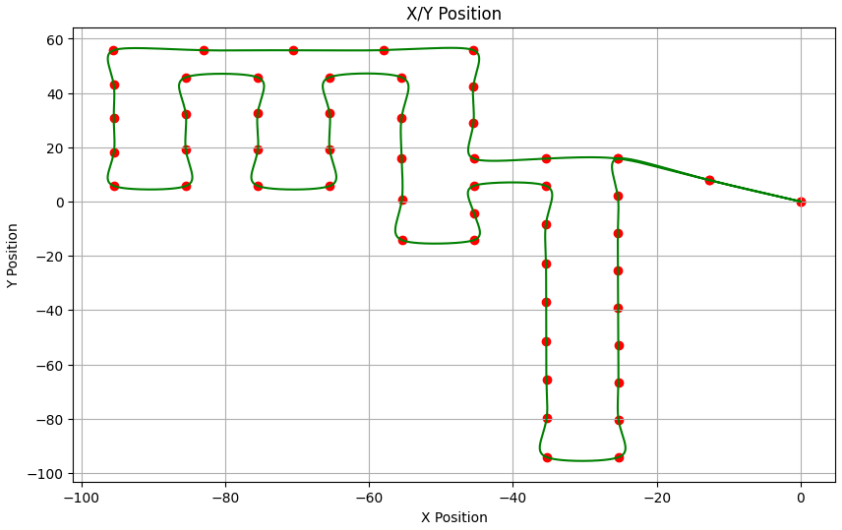



Figure A.7: The position along x and y axis in meters of the drone

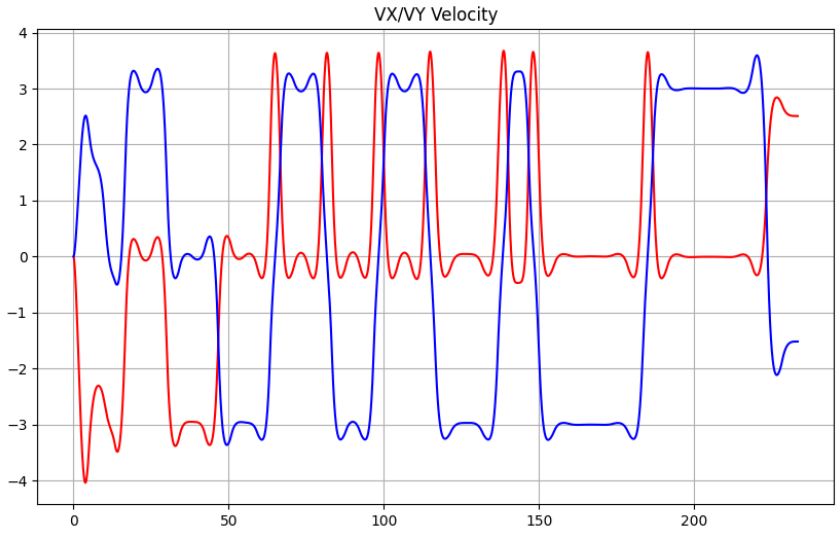



Figure A.8: The x-axis represents the flight time in seconds, while the y-axis shows the drone's velocity (red for the x-direction, blue for the y-direction).

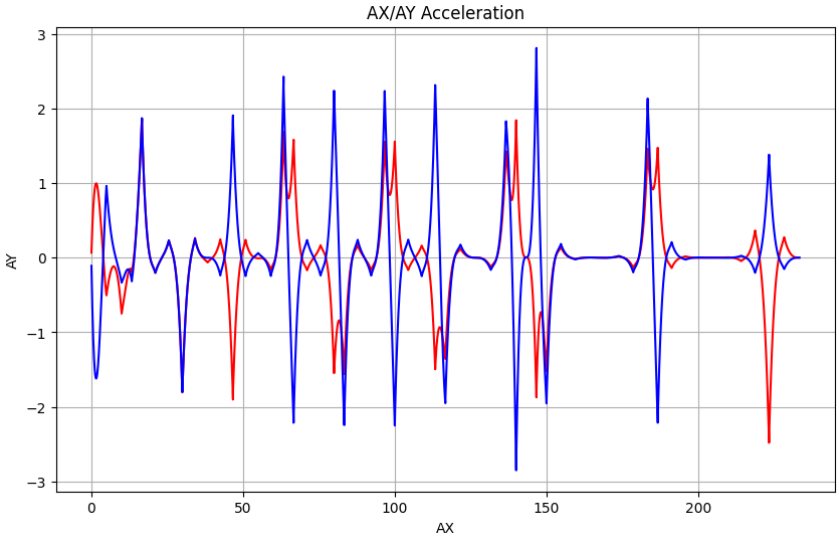



Figure A.9: The acceleration along x and y axis in $m/s^2$ of the drone during time in seconds (red for the x-direction, blue for the y-direction)

# Rectangular field / DARP / Drone 3

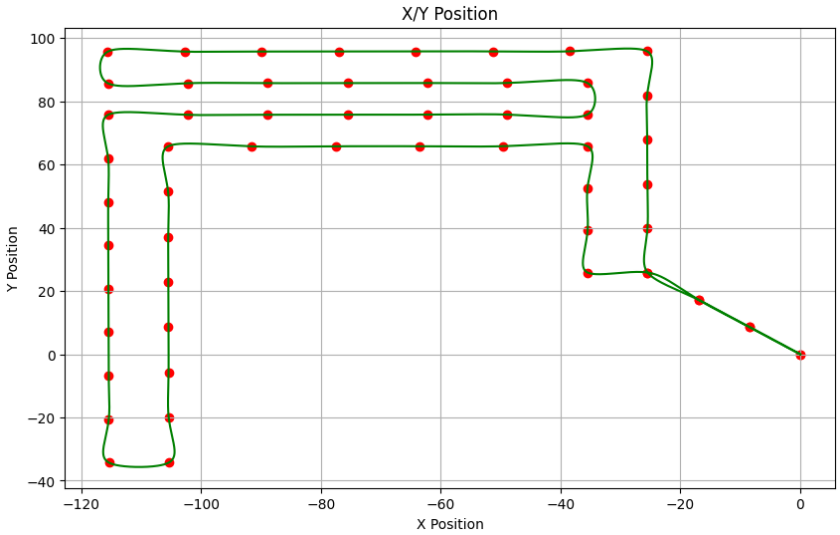



Figure A.10: The position along x and y axis in meters of the drone

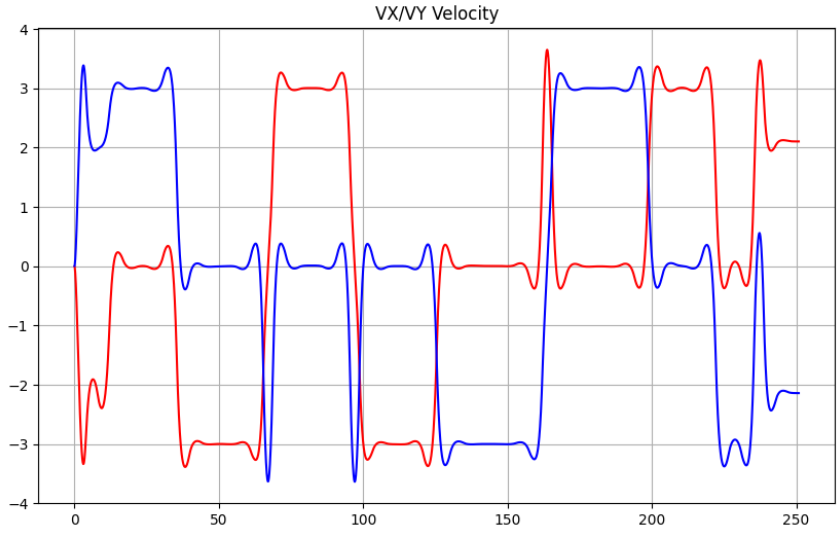



Figure A.11: The x-axis represents the flight time in seconds, while the y-axis shows the drone's velocity (red for the x-direction, blue for the y-direction).

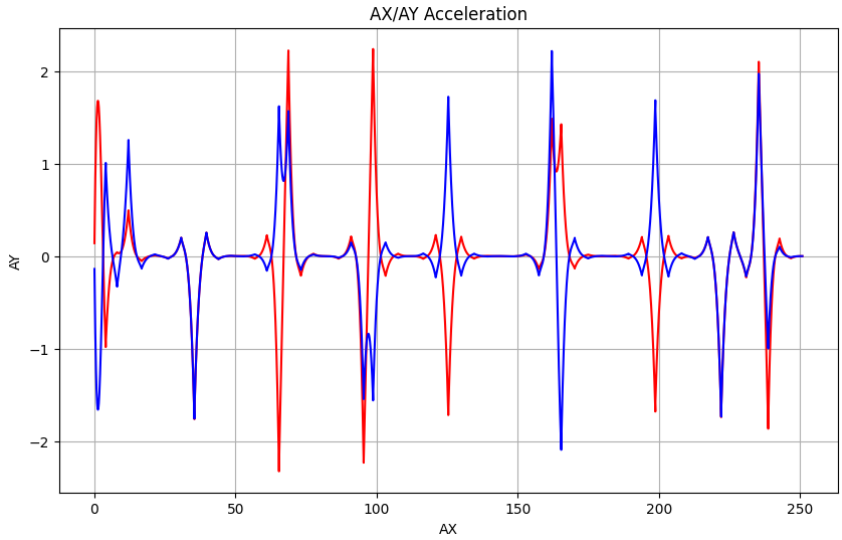



Figure A.12: The acceleration along x and y axis in $m/s^2$ of the drone during time in seconds (red for the x-direction, blue for the y-direction)

## Complex field / Popcorn / Drone 2

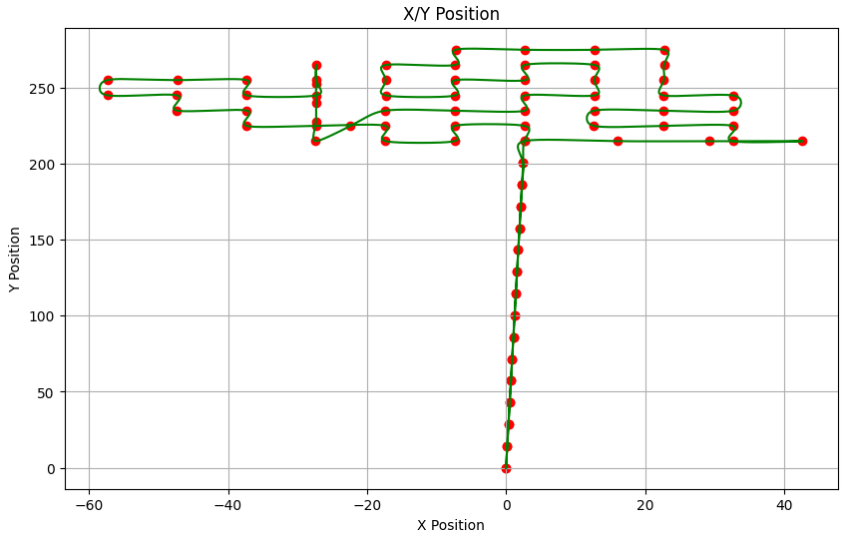



Figure A.13: The position along x and y axis in meters of the drone

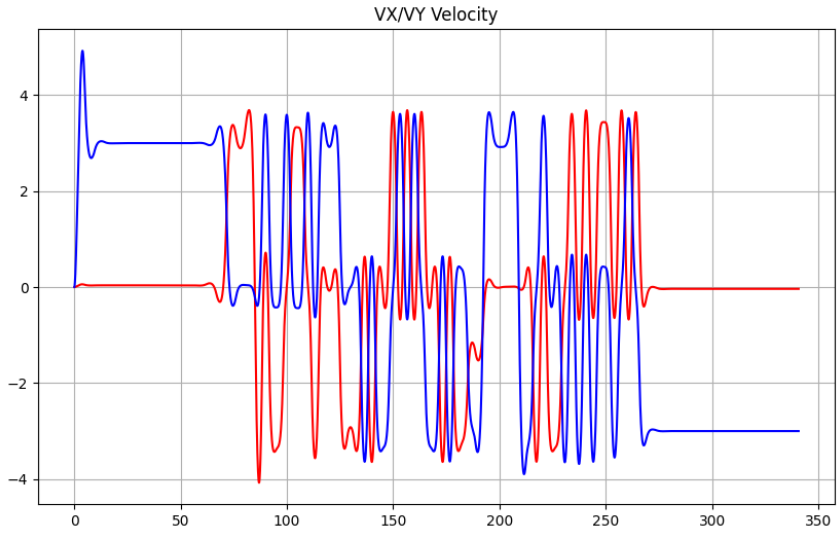



Figure A.14: The x-axis represents the flight time in seconds, while the y-axis shows the drone's velocity (red for the x-direction, blue for the y-direction).

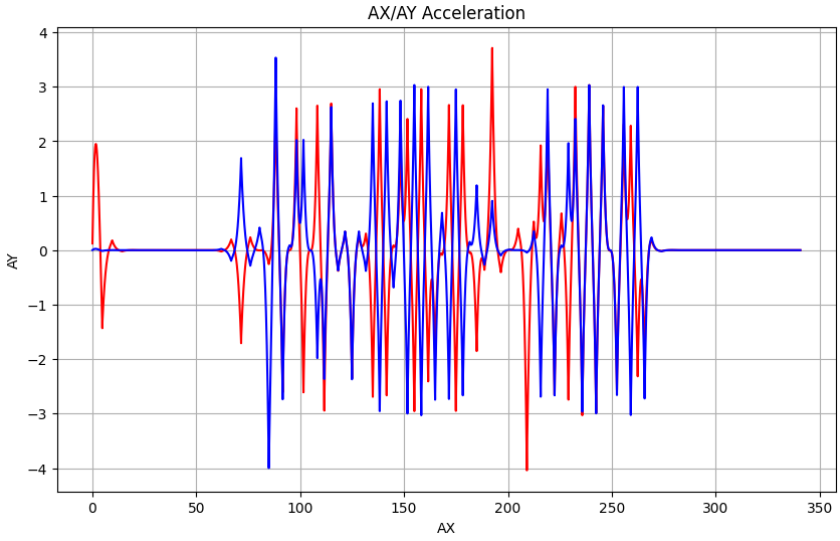



Figure A.15: The acceleration along x and y axis in $m/s^2$ of the drone during time in seconds (red for the x-direction, blue for the y-direction)

## Complex field / Popcorn / Drone 3

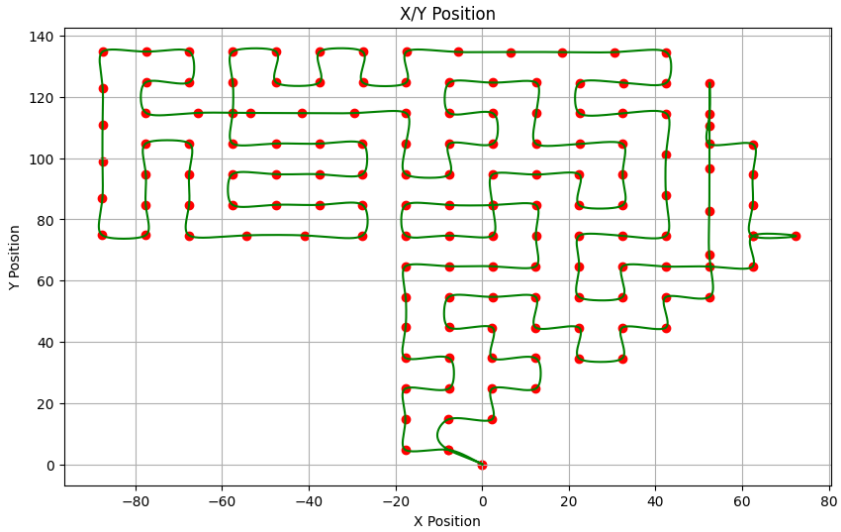



Figure A.16: The position along x and y axis in meters of the drone

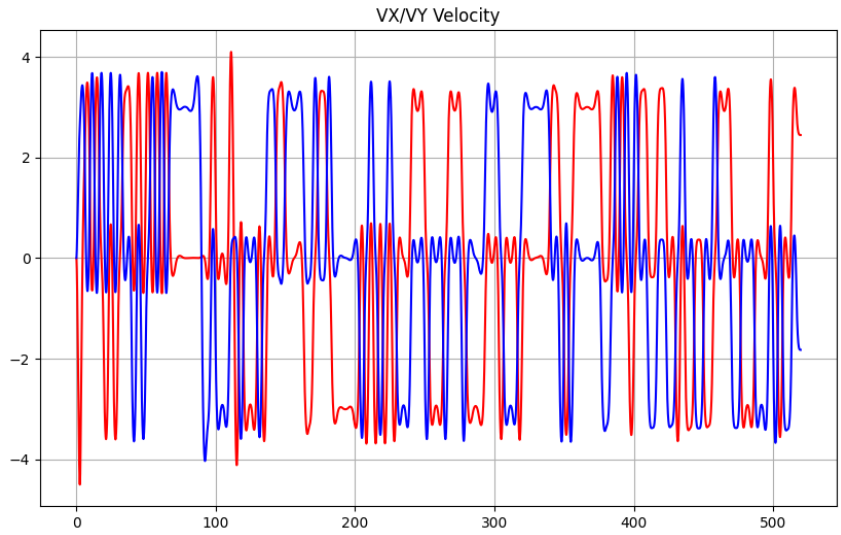



Figure A.17: The x-axis represents the flight time in seconds, while the y-axis shows the drone's velocity (red for the x-direction, blue for the y-direction).

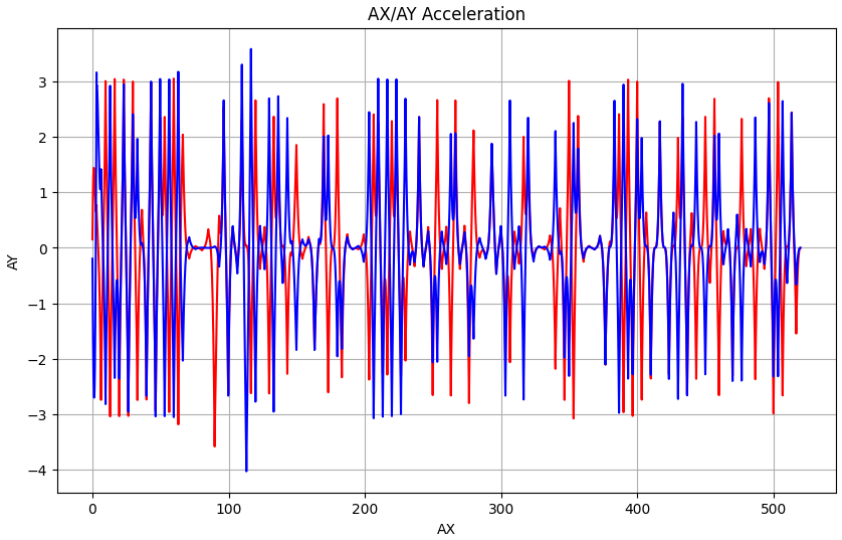



Figure A.18: The acceleration along x and y axis in $m/s^2$ of the drone during time in seconds (red for the x-direction, blue for the y-direction)

## Complex field / Popcorn / Drone 4

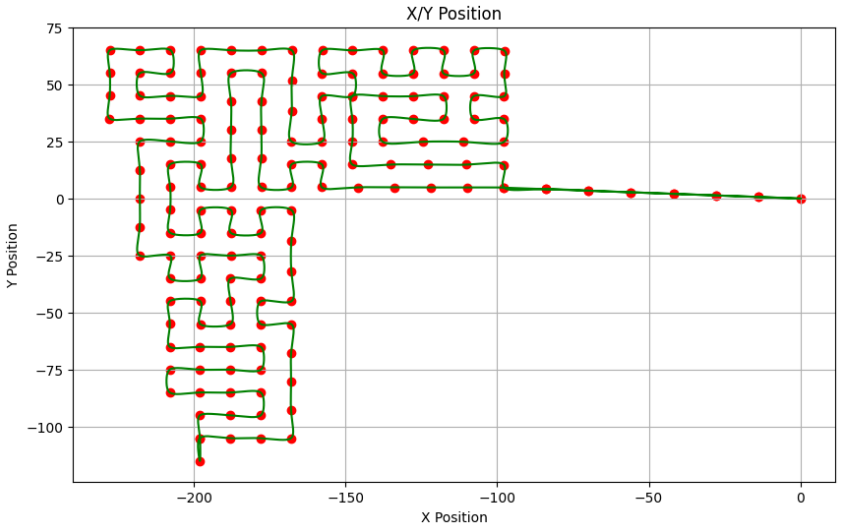



Figure A.19: The position along x and y axis in meters of the drone

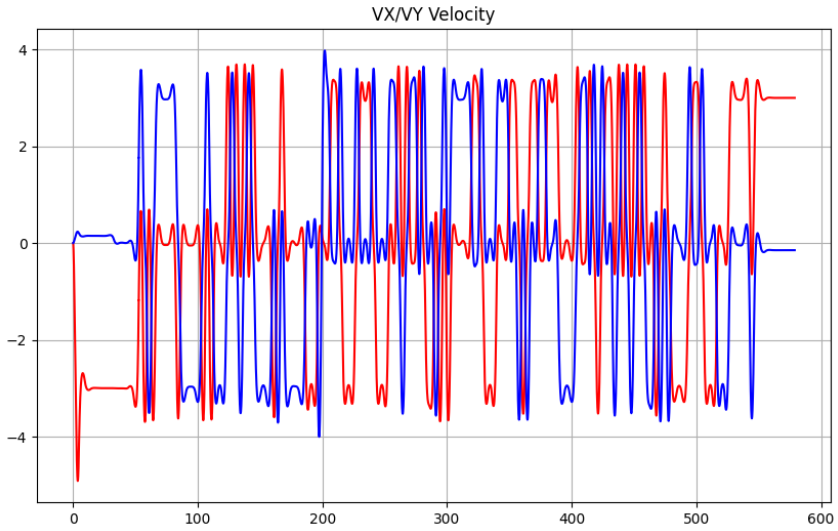



Figure A.20: The x-axis represents the flight time in seconds, while the y-axis shows the drone's velocity (red for the x-direction, blue for the y-direction).

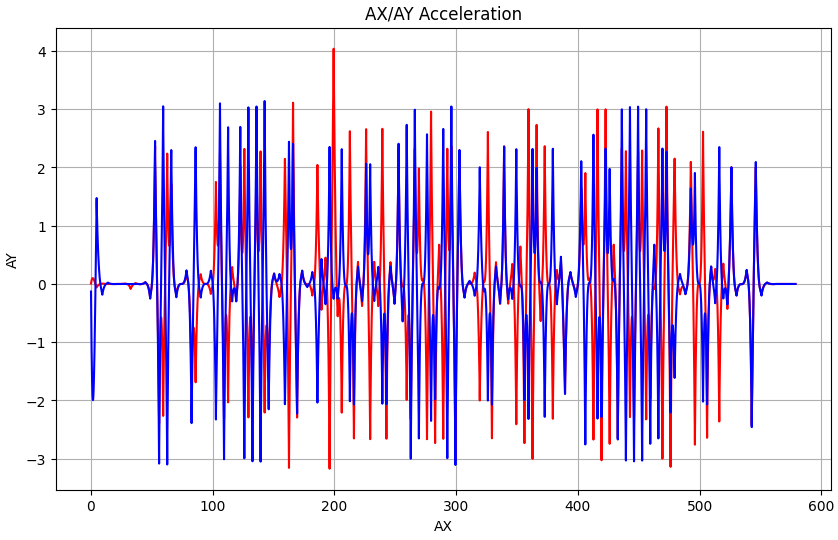



Figure A.21: The acceleration along x and y axis in $m/s^2$ of the drone during time in seconds (red for the x-direction, blue for the y-direction)

## Complex field / Popcorn / Drone 5

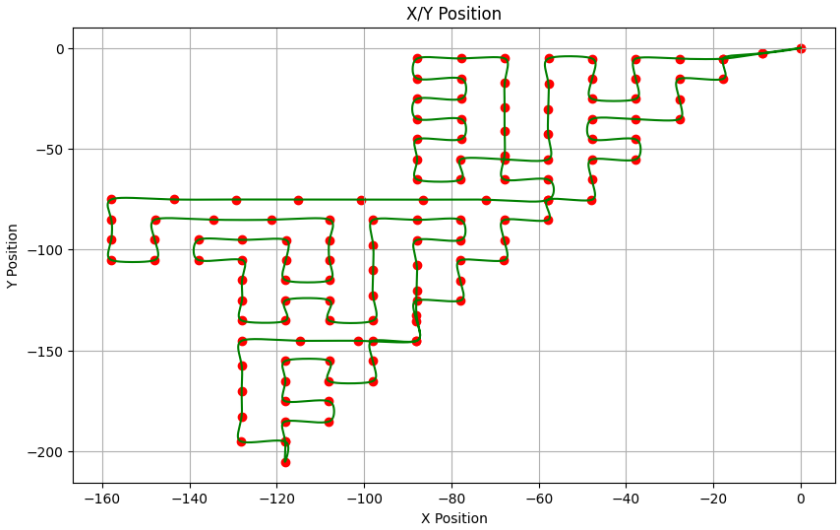



Figure A.22: The position along x and y axis in meters of the drone

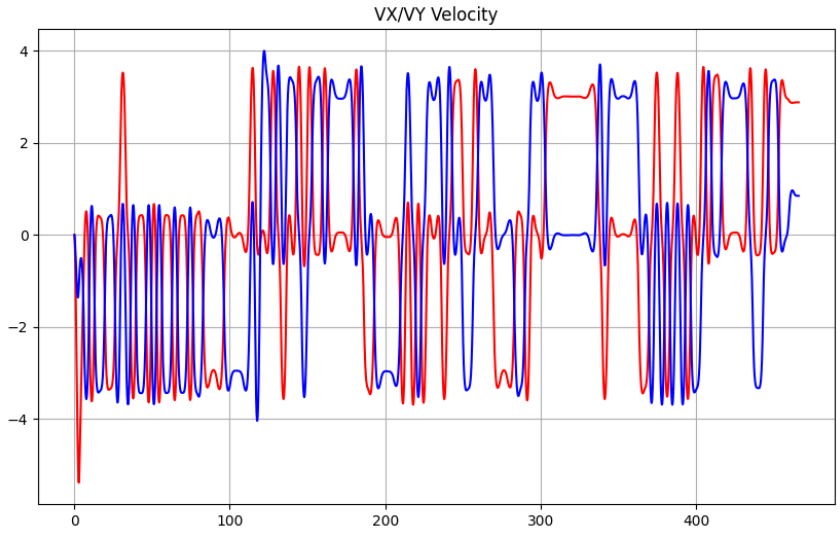



Figure A.23: The x-axis represents the flight time in seconds, while the y-axis shows the drone's velocity (red for the x-direction, blue for the y-direction).

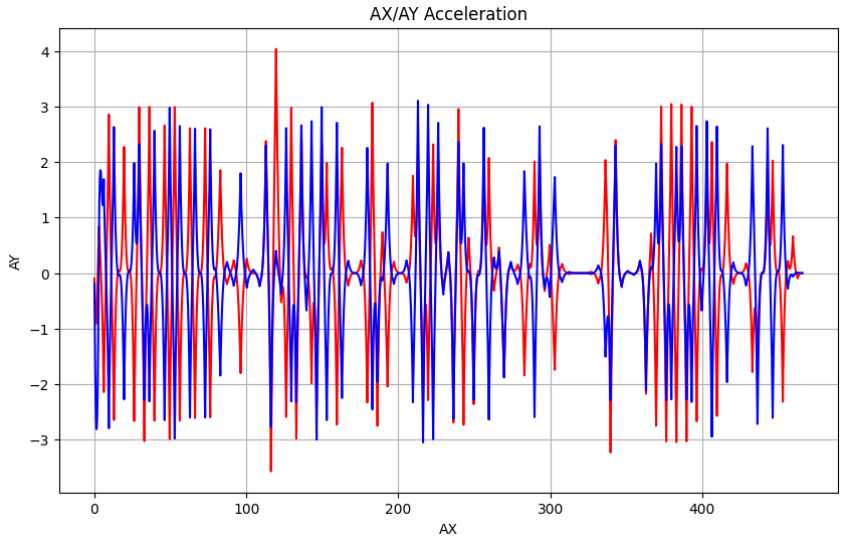



Figure A.24: The acceleration along x and y axis in $m/s^2$ of the drone during time in seconds (red for the x-direction, blue for the y-direction)

## Complex field / Popcorn / Drone 6

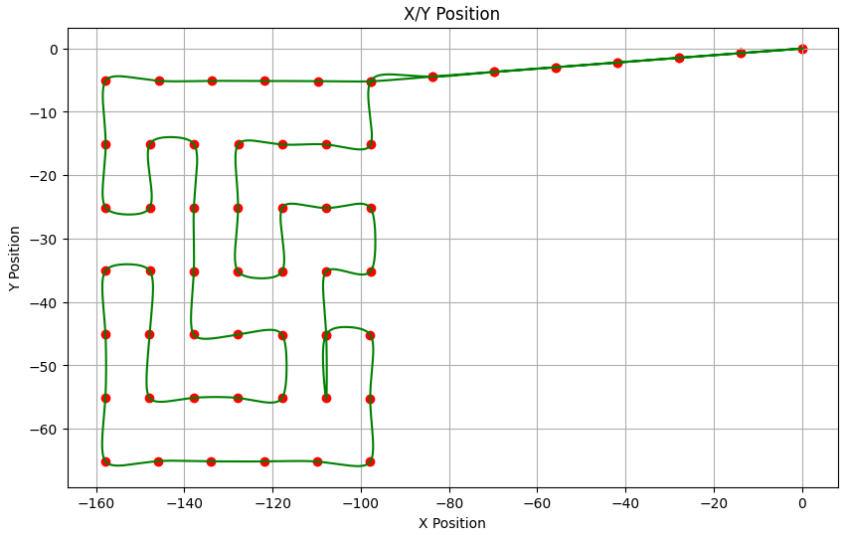



Figure A.25: The position along x and y axis in meters of the drone

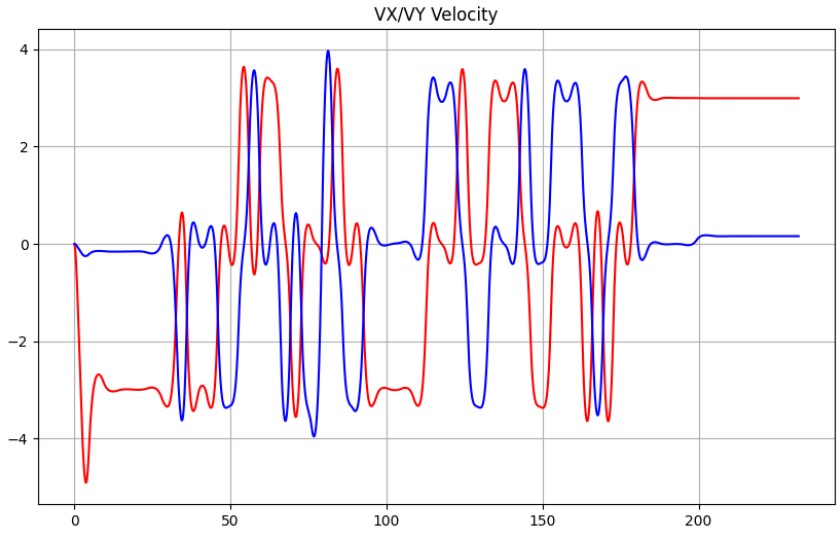



Figure A.26: The x-axis represents the flight time in seconds, while the y-axis shows the drone's velocity (red for the x-direction, blue for the y-direction).

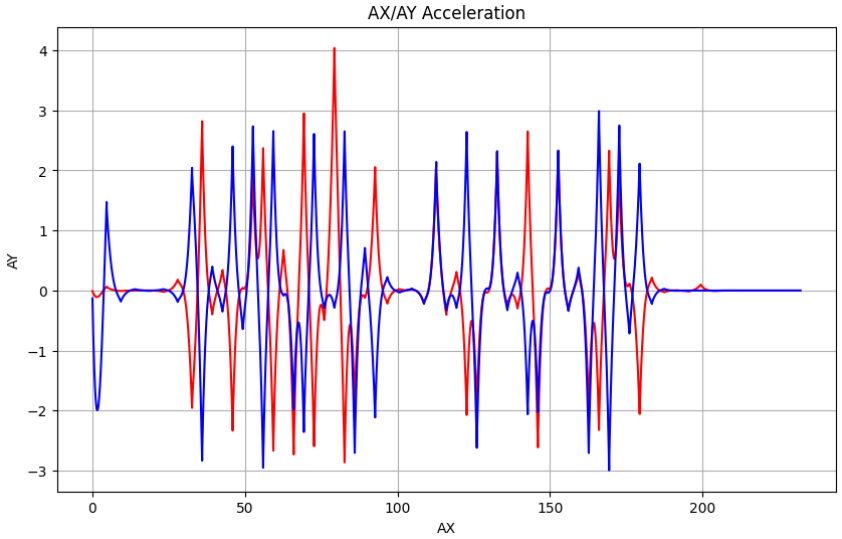



Figure A.27: The acceleration along x and y axis in $m/s^2$ of the drone during time in seconds (red for the x-direction, blue for the y-direction)

## Complex field / Popcorn / Drone 7

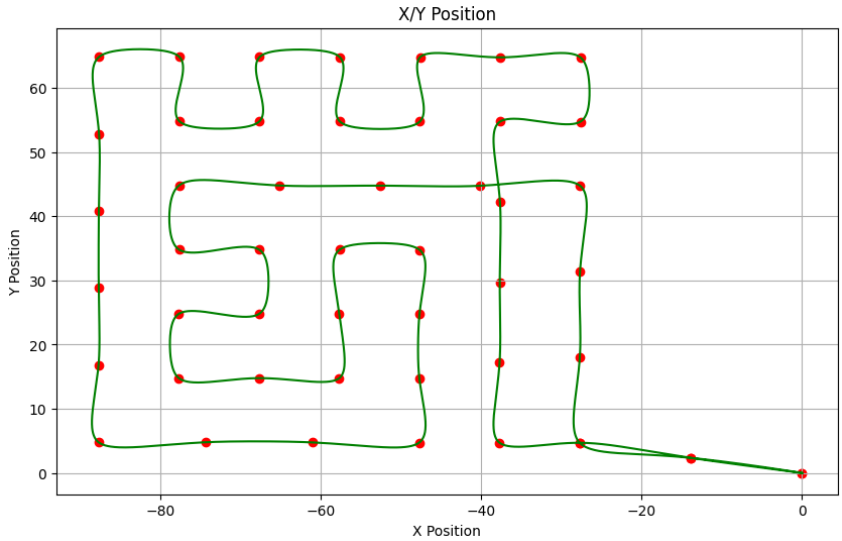



Figure A.28: The position along x and y axis in meters of the drone

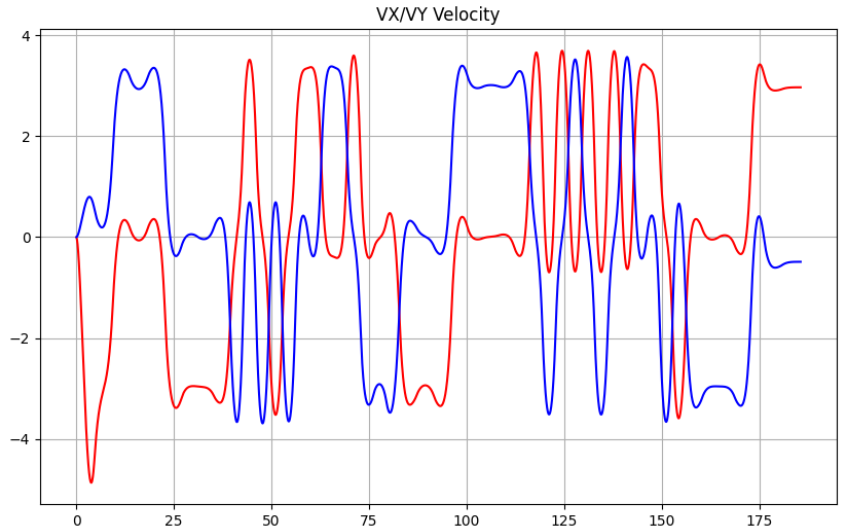



Figure A.29: The x-axis represents the flight time in seconds, while the y-axis shows the drone's velocity (red for the x-direction, blue for the y-direction).

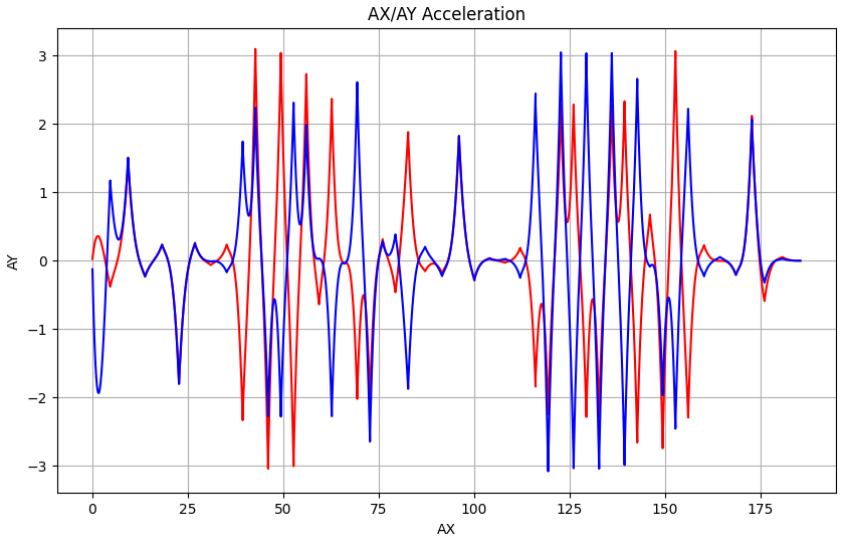



Figure A.30: The acceleration along x and y axis in $m/s^2$ of the drone during time in seconds (red for the x-direction, blue for the y-direction)

## Complex field / DARP / Drone 2

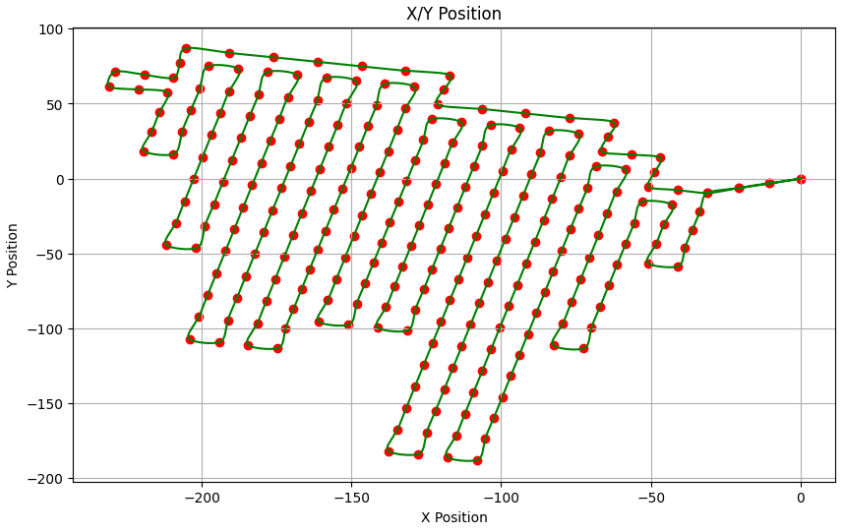



Figure A.31: The position along x and y axis in meters of the drone

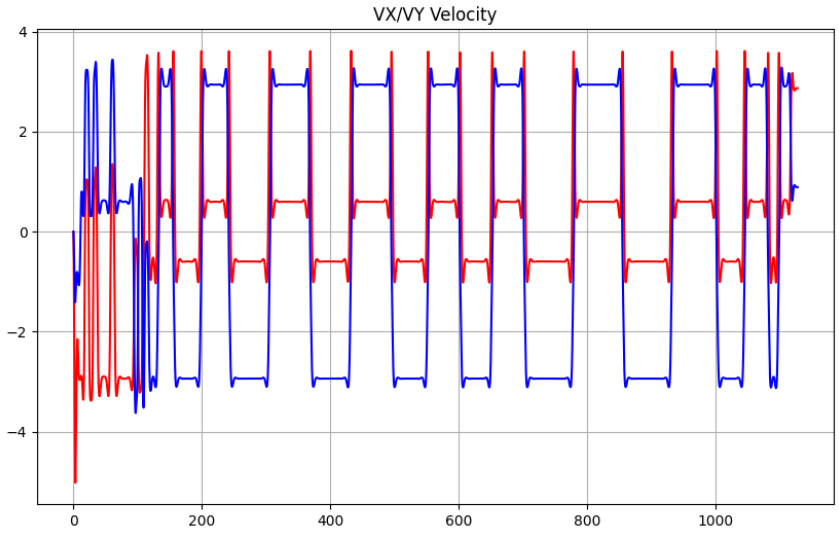



Figure A.32: The x-axis represents the flight time in seconds, while the y-axis shows the drone's velocity (red for the x-direction, blue for the y-direction).

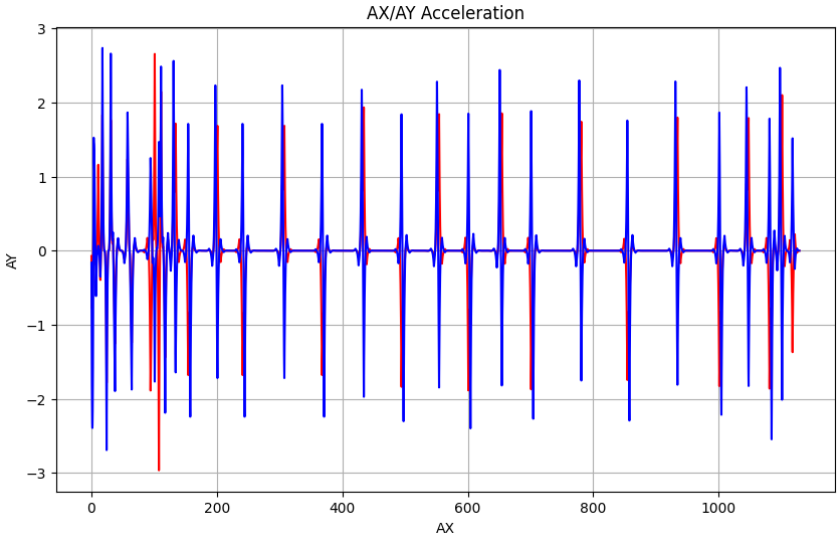



Figure A.33: The acceleration along x and y axis in $m/s^2$ of the drone during time in seconds (red for the x-direction, blue for the y-direction)

# Obstacles field / DARP / Drone 1

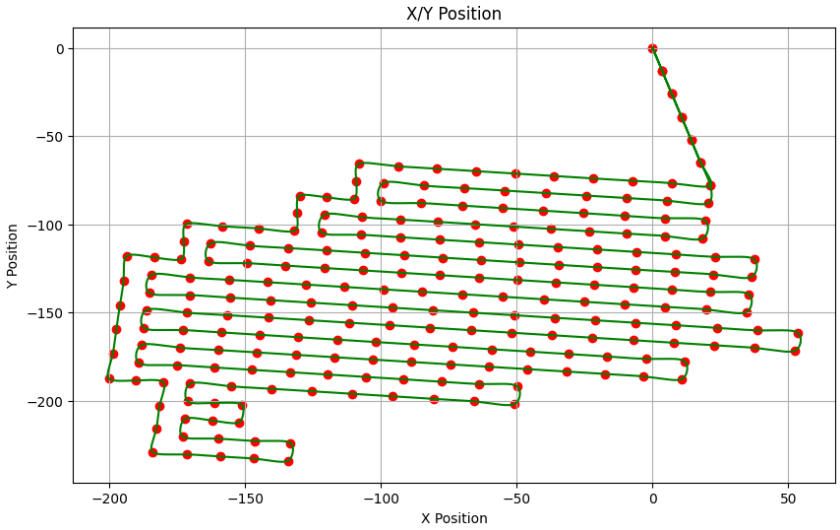



Figure A.34: The position along x and y axis in meters of the drone

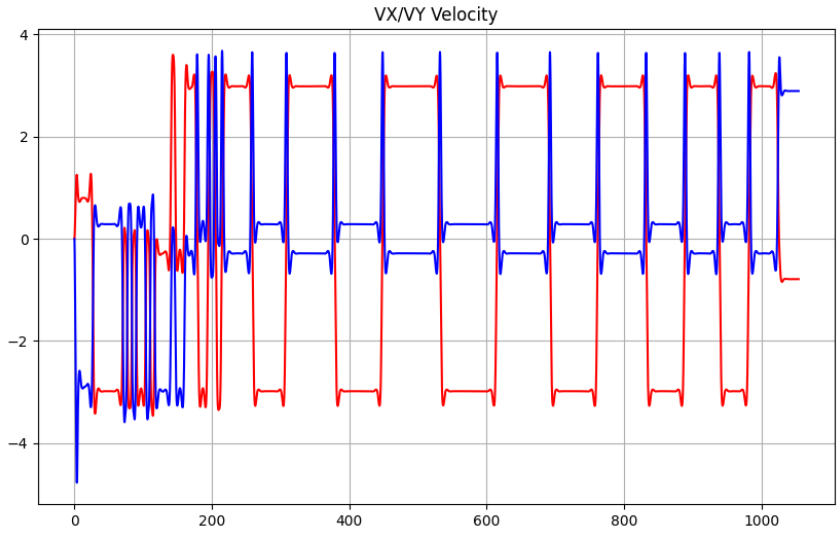



Figure A.35: The x-axis represents the flight time in seconds, while the y-axis shows the drone's velocity (red for the x-direction, blue for the y-direction).

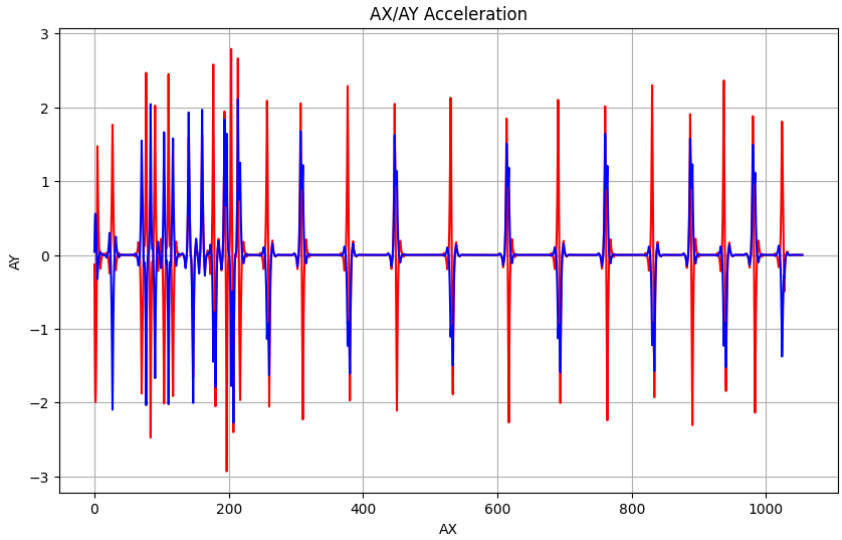



Figure A.36: The acceleration along x and y axis in $m/s^2$ of the drone during time in seconds (red for the x-direction, blue for the y-direction)

## Obstacles field / DARP / Drone 2

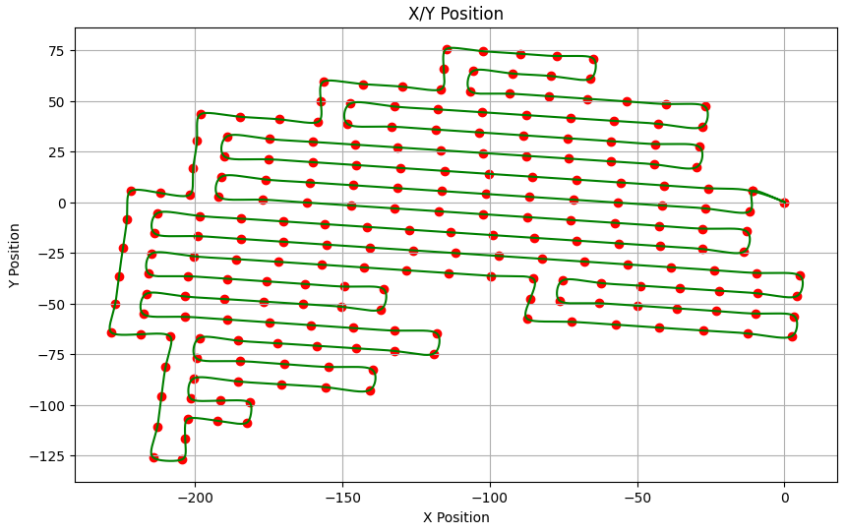



Figure A.37: The position along x and y axis in meters of the drone

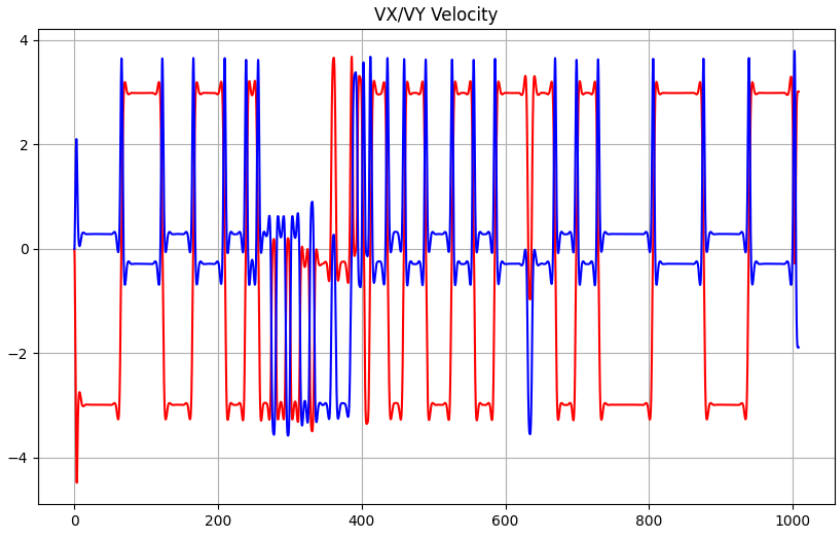



Figure A.38: The x-axis represents the flight time in seconds, while the y-axis shows the drone's velocity (red for the x-direction, blue for the y-direction).

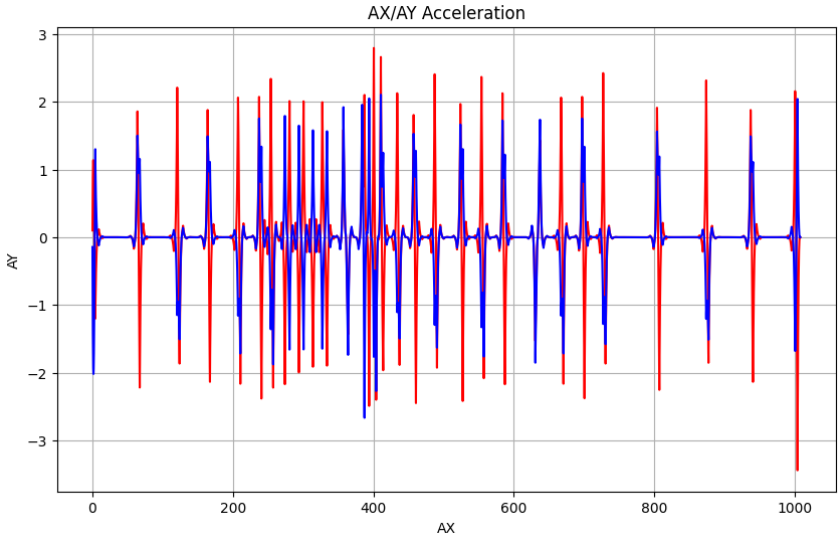



Figure A.39: The acceleration along x and y axis in $m/s^2$ of the drone during time in seconds (red for the x-direction, blue for the y-direction)

## Obstacles field / DARP / Drone 3

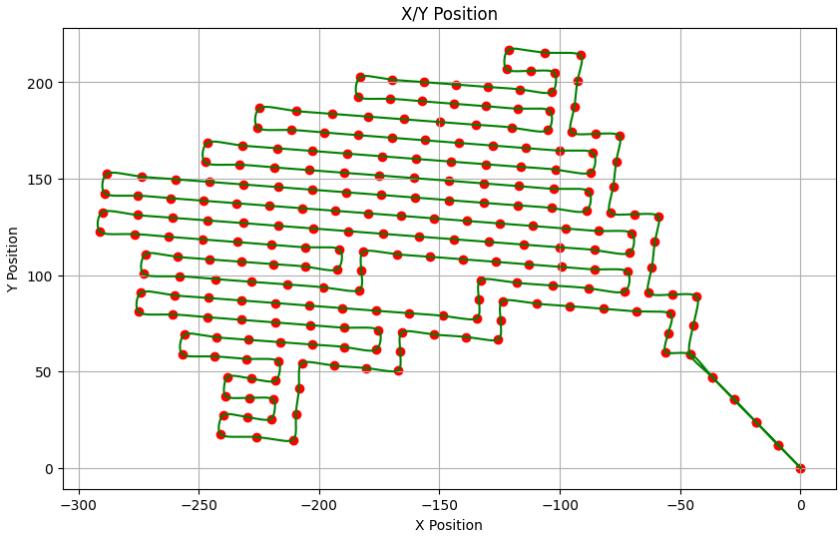



Figure A.40: The position along x and y axis in meters of the drone

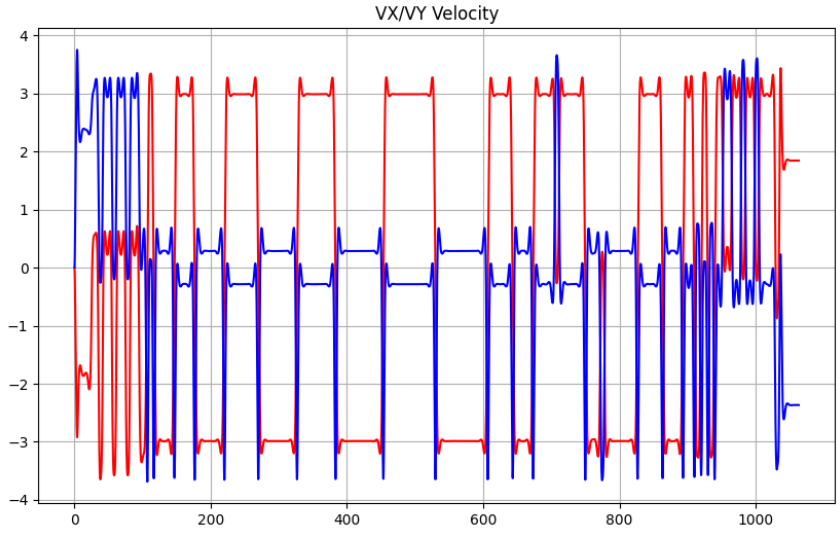



Figure A.41: The x-axis represents the flight time in seconds, while the y-axis shows the drone's velocity (red for the x-direction, blue for the y-direction).

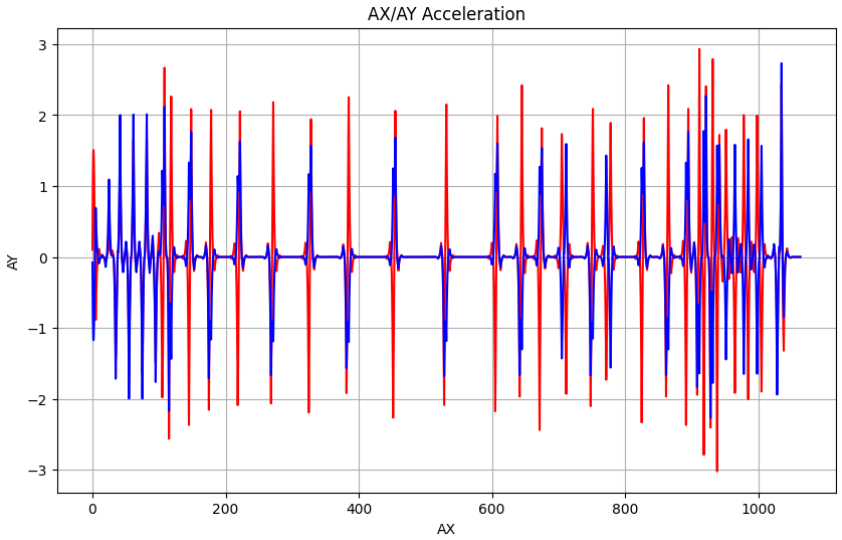



Figure A.42: The acceleration along x and y axis in m/s$^2$ of the drone during time in seconds (red for the x-direction, blue for the y-direction)

## Non-homogeneous swarm and better coverage test / DARP / Drone 1

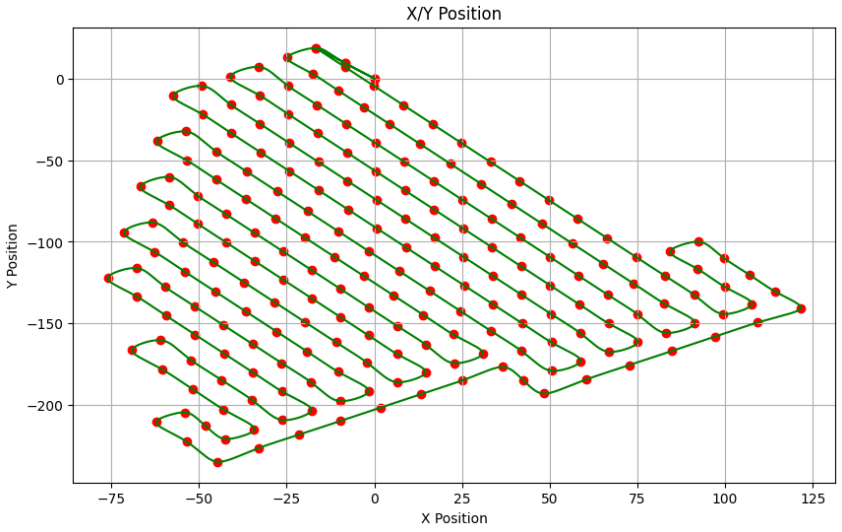



Figure A.43: The position along x and y axis in meters of the drone

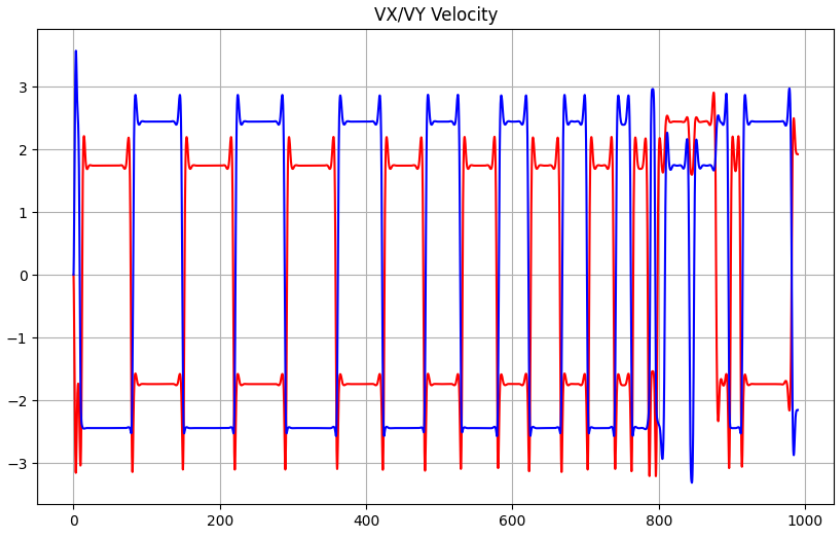



Figure A.44: The x-axis represents the flight time in seconds, while the y-axis shows the drone's velocity (red for the x-direction, blue for the y-direction).

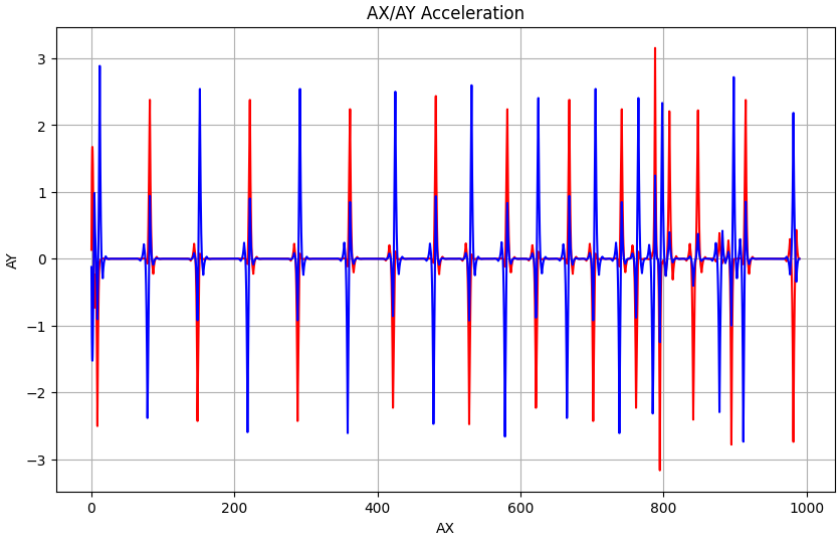



Figure A.45: The acceleration along x and y axis in $m/s^2$ of the drone during time in seconds (red for the x-direction, blue for the y-direction)

# Non-homogeneous swarm and better coverage test / DARP / Drone 2

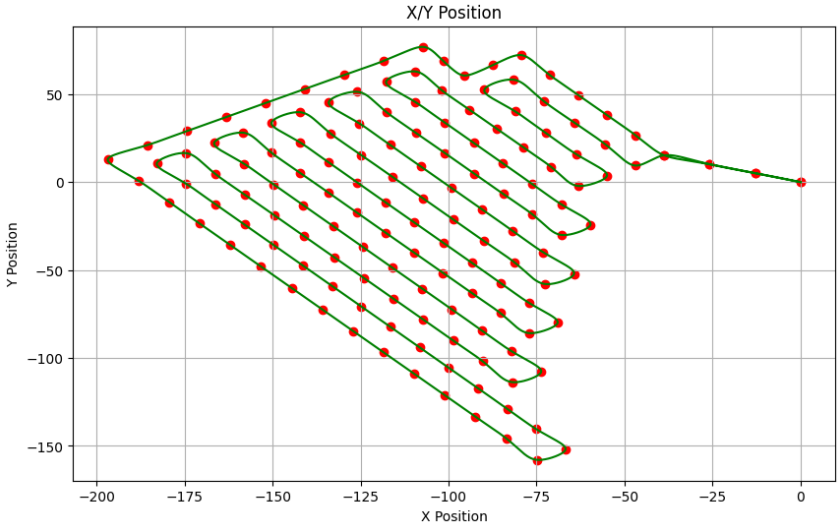



Figure A.46: The position along x and y axis in meters of the drone

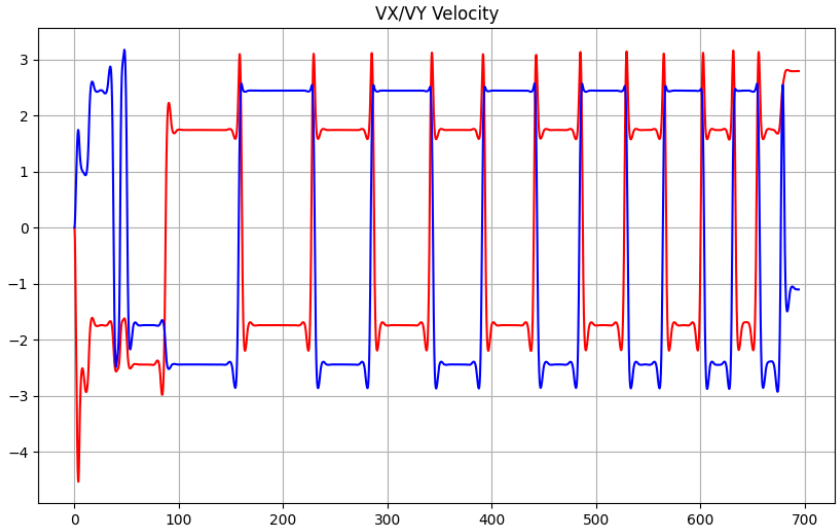



Figure A.47: The x-axis represents the flight time in seconds, while the y-axis shows the drone's velocity (red for the x-direction, blue for the y-direction).

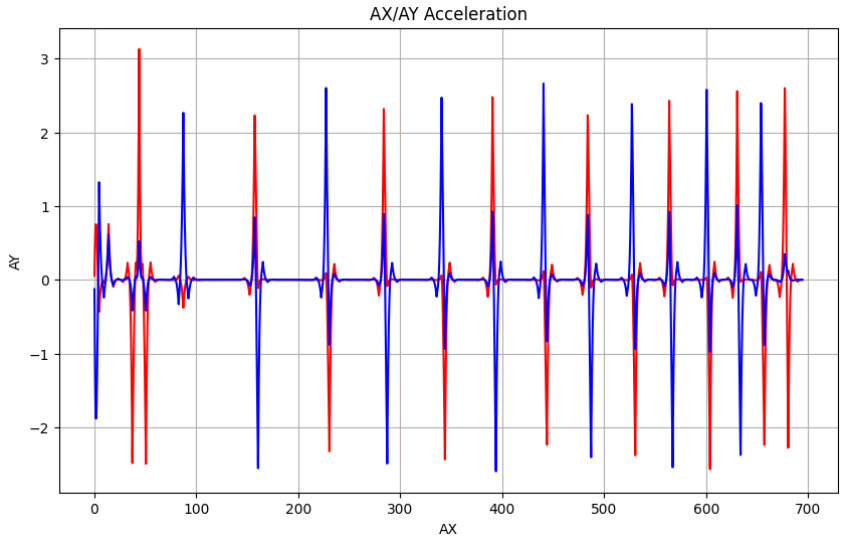



Figure A.48: The acceleration along x and y axis in $m/s^2$ of the drone during time in seconds (red for the x-direction, blue for the y-direction)