INGENIA SE | COURSE 2022-2023

System Design (Final)

Hell-ix Team

May 23, 2022

INDUSTRIALES
ETSII | UPM

Contents

1. Introduction

1.1 Purpose of the Document

The present document aims at presenting an introduction to the design procedure of the system, as well as a brief explanation of the overall structure of the designed model, contained in the Hell-ix group GitHub repository https://github.com/Ingenia-SE/Hell-ix/tree/main/Hell-ix_Project.

1.2 Scope and Overview

In first place the overall system architecture will be commented, together with the identified requirements that justify design decisions. Next, the design details on each of the subsystems will be explained, specifying the constraints and assumptions as well as the design decision rationale that has been followed.

2. System Architecture Overview

2.1 System and System Requirements Description

The system consists of a competition of autonomous drones in the context of INGENIA SE 2022-2023. In order to design the system, MATLAB System Composer tool has been used, taking into account the requirements that were defined in the SyRS document. The design procedure is up-to-down, meaning that the design is more generic in the first steps and as the design process advances, the design of each subsystem is furtherly detailed.

2.2 System Architecture Overview

Figure 2.1 illustrates the comprehensive model architecture, comprising the Droning Team, the Hell-ix team, the environment, and the trials. These four interdependent architectures are meticulously designed to operate in harmony, as they exert mutual influence on each other. The environment primarily encompasses the materials and individuals involved in the competition, significantly impacting both the trials and the architectures of both teams. Conversely, the teams directly influence the competition by defining its parameters and shape the environment by designing obstacles and participating as individuals. Finally, the trials have a profound impact on both teams, as their work is rooted in adherence to the competition rules, and on the environment, as the obstacles must be conceived and constructed in accordance with these regulations.

3. System Design Details

In the present section, the system design details are indicated, such as the identified functional requirements and use cases, the desired system behavior and interactions with its environment and the design constraints and assumptions. All these details lead to the different design decisions that have been made to devise the system. The most relevant design decisions are included in this section, while the most particular ones are indicated together with

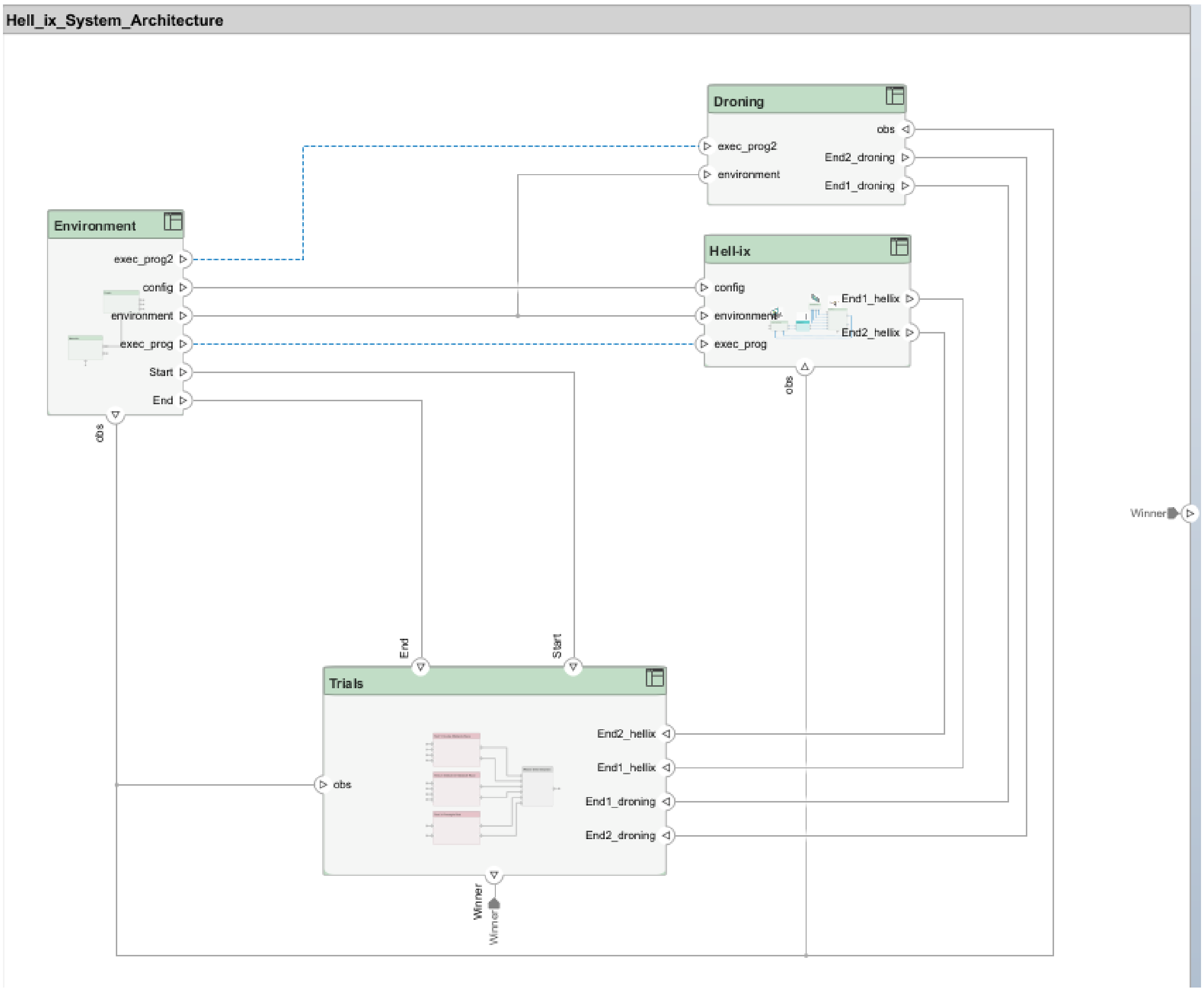

Figure 2.1: General system model.

the description of the subsystem in section 3.5.5, which describes the subsystem structure of the design.

3.1 Functional Requirements

The functional requirements are described in detail in the `System Requirements Specification` document. This document outlines the specifications and design considerations for the autonomous racing drone project. It provides a comprehensive guide for stakeholders and the Hellix Group, detailing the project's goals, requirements, and limitations.

User Requirements

The User Requirements Definition section describes the services provided to the user and specifies non-functional requirements. It includes requirements related to competition deadlines, the competition environment, compliance with drone flight regulations, negotiation of competition rules, specific competition tests such as speed, precision, and obstacle avoidance, and the scoring system.

System requirements

The system requirements outlined in the document aim to ensure the controlled flight and efficient operation of the drone. The drone should be capable of flying in a controlled manner, maintaining stable flight conditions, a specific altitude, and following a predefined flight path with minimal deviation (SY-001). This includes numerical constraints for altitude, stability, and trajectory to ensure precise control and adherence to the intended path.

To achieve autonomous flight, the drone's motors must provide sufficient power to lift its own weight along with the mounted components (SY-001.1). Additionally, the Ground Control Station (GCS) should have the ability to manipulate the drone manually in case of security threats or unexpected events (SY-001.2). The GCS should also be able to control the drone's movement automatically, ensuring reliable and safe operation (SY-001.3).

The GCS should relay real-time information about the drone's status, including location, altitude, battery life, and any potential errors or malfunctions (SY-001.4). The drone should incorporate automatic lift-off and landing procedures for safe and efficient operation (SY-001.5). An onboard control algorithm should be implemented to enable autonomous operation, processing sensory inputs, and controlling the drone's movements (SY-001.6).

Visual indications of the drone's current status should be clear, easily visible, and distinguishable, providing information such as battery level, altitude, speed, and active modes or commands (SY-002). The installed battery should have sufficient capacity to ensure completion of the competition tasks without the need for recharge (SY-003). At any moment during the competition, there should be a minimum of two replacement batteries available for uninterrupted operation (SY-003.1).

The drone should be capable of autonomous navigation, following a predefined path and avoiding obstacles (SY-004). It should detect and avoid obstacles, such as hoops, poles, doors, and split rectangles, using a camera mounted on the drone (SY-004.1, SY-004.1.1). The drone must react in a timely manner to obstacles in its path to avoid collisions (SY-004.1.2).

The drone should also have the ability to follow a target position accurately and reliably (SY-004.2). The GCS, using Bezier curves, should generate trajectories consisting of target positions for the drone to execute (SY-004.3).

Software Requirements

The software requirements for the project encompass various aspects of the system, ensuring its functionality and reliability. Firstly, the source code (SO-001) must be traceable and stored in a repository for easy tracking of changes and access to previous versions. Next, the software should facilitate communication between the drone and the Ground Control Station (GCS) wirelessly (SO-002). This enables the transfer of collected sensor data from the drone to the GCS and the receipt of instructions from the computer.

To address potential connection loss, the control system incorporates a fail-safe mechanism (SO-002.1) that activates when the drone-GCS connection is disrupted. This mechanism ensures the completion of the drone's last assigned task and maintains its current position. Additionally, in the event of an extended connection loss, an automatic landing mechanism (SO-002.2) is triggered. The drone initiates a controlled descent, gradually and vertically, until it safely reaches the ground.

The estimation of the drone's position (SO-002.3) involves collaboration between the drone and the GCS. Sensor data, odometry, and mathematical models are utilized for position estimation. Deviations from the target position are determined by comparing the current drone estimate with the GCS estimate. Furthermore, the drone sends captured images directly to the GCS (SO-002.4), bypassing onboard image processing.

For the central computer's vision capabilities, an image processing pipeline is created (SO-003). This pipeline takes images as inputs and generates obstacle detections as outputs. These detections are then processed into instructions for the drone to execute, ensuring safe navigation and obstacle avoidance.

The GCS requires access to the drone's battery status (SO-004) to provide real-time updates to the user. A battery monitoring system continuously measures the remaining charge level, allowing the GCS to warn the user when the battery is low. This enables appropriate actions to be taken, such as ensuring a safe landing or replacing the battery if necessary.

Lastly, the software's control loop must be fast enough to ensure the reactivity of the drone (SO-005). To achieve this, the processing pipeline for image processing (SO-003) is designed to be as lightweight as possible, incorporating optimizations and efficient technologies. Additionally, a compiled programming language (SO-005.1) is utilized to minimize latency between detection and the resulting instruction.

3.2 Use Cases

The Use Cases that have been identified are described in detail in the `OpsCon` document. Two scenarios are presented: Test Scenario and Emergency Scenario, with 4 emergency subcases.

The Test Scenario focuses on safe indoor autonomous drone flight tests and includes considerations such as a spacious area, adequate lighting, absence of obstacles, air traffic control system, communication system, network security, safety equipment, and regulatory compliance. The goal is to ensure the safety of people, the drone, and the test space while conducting autonomous drone flights indoors.

The Emergency Scenarios section addresses potential situations in which the drone may encounter issues or stop responding. It provides guidelines for actions and implementation in different emergency scenarios, such as the drone stopping when changing direction, loss of connection between the drone and ground station, loss of control with the remote control, and short circuit during operation. The procedures include steps to manually take control of the drone, check for failures in hardware, programming code, or network units, and perform necessary troubleshooting and checks.

3.3 System Behavior and Interactions

The drone should exhibit specific characteristics and behaviors to navigate its surroundings effectively.

First and foremost, the drone should possess stable flight capabilities, maintaining control and balance throughout its operation. It should be capable of staying in the air without excessive vibrations and respond accurately to controller commands. This stability and control ensure smooth maneuverability, allowing the drone to perform tasks and navigate various race scenarios effectively.

To interact with the drone's environment, the Ground Control Station calculates a preliminary trajectory given the gates' positions and other parameters as inputs. The drone needs to employ obstacle detection mechanisms using computer vision techniques. By detecting and recognizing the obstacles, the drone can safely avoid collisions and navigate towards them.

This capability is essential to ensure the drone's safety and protect it from potential damage during flight, as well as completing the competition successfully.

Furthermore, the drone should be capable of achieving competitive speeds and accelerations. It should undergo tests to assess its speed and acceleration on a designated circuit. This ensures that the drone can meet the requirements of the race and efficiently maneuver through the course. Additionally, it should be able to brake safely and efficiently at the end of each test.

Battery autonomy is another critical factor for the drone's behavior. The drone should undergo tests to evaluate its battery life and ensure that it can complete a full race without running out of power. It is vital for the drone to have sufficient endurance to participate effectively and complete the race successfully.

The drone's behavior and interaction also heavily rely on its programming and software capabilities. It should be thoroughly tested to ensure that it can execute all necessary tasks during the race, including obstacle detection, flight control, and navigation. Simulation tests can be conducted to assess the drone's ability to adapt to different race situations, making it adaptable and responsive to changing environments.

Safety is paramount, and the drone should comply with all safety regulations and requirements. It should be evaluated to ensure that it does not pose any risk to spectators or other competitors during the race. The drone's behavior should prioritize the safety of people and property, employing appropriate safety measures and avoiding potentially dangerous situations.

In emergency scenarios, where the drone stops responding to programmed or manual commands, specific guidelines and actions should be followed. These guidelines may include steps to manually take control of the drone, perform system restarts, improve lighting or proximity to the ground station, and check for failures in programming code or hardware components. These procedures ensure quick responses and effective troubleshooting in emergency situations.

3.4 Design Constraints and Assumptions

Due to the project characteristics, the design has been limited and several assumptions have been made.

Several constraints affecting the drone have been identified. To accommodate the total budget of 2000, the drone design, including its components, accessories, and any required add-ons, must be within this financial limit. Furthermore, the selected drone type must have the capability to race autonomously, aligning with the project's requirements. The drone's battery should have sufficient capacity to last at least one minute, the freestyle competition duration, in addition to the duration required to complete the speed races. Additionally, the drone must be able to establish and maintain a stable communication link with the central ground station throughout the competition. The drone must be able to be controlled in an auxiliary way, in case the software implementation does not work appropriately. Accessibility to spare parts, repair tools, and technical support should be taken into account, in case that any drone component is broken during testing. The project should strive to minimize its environmental footprint by considering sustainable practices, recycling materials, and minimizing waste generation, as expressed in the Sustainability Report.

Several constraints affecting the competition have been identified. The drone must be equipped with a camera or an addable camera module capable of capturing real-time images for obstacle recognition and avoidance. The design of the drone must allow it to fit within the dimensions of the obstacles used in the competition, which consist of hollow squares with sides measuring 1.4 meters. The competition will take place in an enclosed room within the university facilities, imposing restrictions on the maximum ceiling height and lateral dimensions of the space. All aspects of the project, including the drone development, obstacle acquisition or creation, software implementation, and event organization, must be completed within the specific timeline from September 2022 to May 2023. As per the given information, the total budget of 2000 euros should cover all expenses related to the project, including the drone, obstacles, add-ons, and any other relevant costs. It is encouraged to utilize recycled materials for obstacle construction if possible, or alternatively, utilize obstacles already available in the possession of the CAR group. The design should allow for potential scalability of the competition, accommodating different obstacles or unexpected changes as the project is developed. The competition should be held in an accessible space for the spectators, or with the appropriate recording facilities so that it can be broadcasted.

3.5 Most Relevant Design Decisions

In this section, the most relevant design decisions are indicated. The design decisions that are more specific for each component or subsystem of the model are indicated in section 3.5.5, which describes the overall structure of the design.

3.5.1 Drone Selection

One of the pivotal decisions in the project revolved around selecting an appropriate drone model. A comprehensive study was conducted to evaluate the various models available in the market, taking into consideration the following parameters:

- Specific technical requirements tailored to our competition, including factors such as size, weight, power, and compatibility with external devices such as a camera and an IA deck.
- Expert advice and insights from the CAR (Centre for Automation and Robotics) CSIC-UPM.
- Financial and budgetary constraints.

Based on these considerations, the choice was narrowed down to two models: the DJI Ryze TELLO and the Crazyflie 2.1 from Bitcraze. The main differences between these two models can be reduced to the following:

- **Size and weight:** The Crazyflie 2.1 is smaller and more lightweight, allowing for agile maneuvers and versatile flying capabilities than the TELLO.
- **Power and Flight Performance:** The TELLO is equipped with a battery that offers a flight time of around 13 minutes, providing decent endurance for its size. The Crazyflie 2.1 allows users to select and customize the battery, which for our project would be of 6 minutes.
- **Compatibility and Expandability:** The TELLO has limited expandability options, with pre-defined interfaces primarily focused on camera usage, whereas the Crazyflie 2.1 is highly expandable, with support for various add-ons and decks, allowing users to customize functionality according to specific requirements.
- **Sensor and Control Features:** The TELLO incorporates basic stabilization and obstacle

detection features, suitable for entry-level drone operations. The Crazyflie 2.1 provides advanced sensor fusion capabilities, such as an IMU, barometer, and optional expansion decks for additional functionalities.

- **Software and Development Ecosystem:** The TELLO has a user-friendly interface and a robust software ecosystem, allowing for easy control and integration with mobile devices. The Crazyflie 2.1 offers an open-source development platform, providing more flexibility for customization and extensive community support.
- **Cost:** The TELLO is relatively affordable, while the Crazyflie 2.1 is priced slightly higher, reflecting its expandability and advanced capabilities. The Crazyflie presents an extra cost in terms of the extra components that need to be purchased (camera, LED module, AI-deck, batteries...).
- **Variability for the competition:** The Crazyflie allows us for a more varied competition, considering the other team selected the TELLO.

After considering these factors, we have concluded that the Crazyflie drone is better suited to meet our specific requirements. Its notable advantages, such as its customizable nature and ability to deliver fast and precise performance, align well with our objectives of achieving optimal performance and securing a competitive edge in the competition. By leveraging the Crazyflie's customization capabilities, we anticipated gaining an advantage that could potentially lead to a successful outcome in the competition that could compensate the higher economic effort.

Another substantial factor was the consideration of the environmental effect of each drone's lifecycle: although the composition of the two models is similar, the Crazyflie has a substantially smaller size, resulting in the saving of plastic and other materials.

3.5.2 Ground Central Station

Another crucial design decision was whether to utilize a ground central station to which the drone is connected, or to do all the necessary processing for following the trajectory and detecting obstacles on an on-board computer.

Utilizing a ground central station for computing in autonomous drones racing scenarios provides a range of advantages.

Firstly, the ground central station offers significantly increased processing power compared to on-board computing systems. This enhanced computational capability enables real-time obstacle detection and analysis with high precision. Complex algorithms can be executed in a fast manner, allowing the drones to identify and respond to obstacles efficiently during the race.

In addition to processing power, the ground central station provides extended memory and storage capacities. This ample storage space enables the central station to store extensive datasets and historical race information which can be useful for AI tasks, and advanced obstacle detection algorithms. The availability of such resources contributes to more sophisticated obstacle identification techniques, resulting in improved accuracy and reliability.

Moreover, the ground central station offers enhanced connectivity through reliable and high-bandwidth network connections. This communication between the central station and the drone facilitates the transmission of crucial data, commands, and updates. The drone can receive real-time instructions and relay data back to the central station, allowing for coordinated obstacle identification and avoidance strategies.

Another notable advantage is the flexibility and scalability provided by the ground central station. It offers the opportunity for hardware and software upgrades, ensuring adaptability to evolving race requirements and technological advancements. Upgrades and replacements can be easily implemented in the central station setup, simplifying maintenance procedures and minimizing operational disruptions.

Furthermore, utilizing a ground central station reduces the weight and power consumption of the on-board computing systems in the drone, which is limited. Offloading computationally intensive tasks to the central station lightens the workload of the drone, enhancing its agility and overall performance during the race. The reduced power requirements also extend the flight duration, enabling longer and more competitive races.

The combination of increased processing power, extended memory and storage, enhanced connectivity, flexibility, simplified maintenance, and reduced weight and power consumption that empowers the drone to navigate races with exceptional efficiency, accuracy, and safety are what led the team to choose the option of using a ground central station.

3.5.3 Competition Tests

The selection of competition tests was conducted in collaboration with the Droning team, aiming to optimize the adaptation of the competition to the capabilities of both drones. Our drone demonstrated superior capabilities in trajectory calculation and tracking, while the Droning team focused their efforts on drone vision and gate recognition. Consequently, to accommodate both drones effectively, we made the decision to simplify the trajectories and ensure an adequate distance between gates.

The competition is therefore structured into three distinct tests, each designed to evaluate specific aspects of the drones' performance.

The first test is a speed test, which assesses the drones' ability to navigate quickly and accurately. In this test, the drones initiate takeoff, follow a predetermined straight line trajectory, pass through four gates positioned along the path, and then successfully land.

The second test focuses on assessing the drones' maneuverability and agility. The drones take off and navigate through a circuit shaped like a semicircle, going through four gates strategically placed along the route. Following completion of the circuit, the drones must execute a safe landing.

The third and final test is the freestyle test, allowing the drones to showcase their capabilities and entertain the audience. This test provides an opportunity for the drones to perform various aerial maneuvers, demonstrating their agility, precision, and creativity, all while captivating and impressing the spectators.

By dividing the competition into these three distinct tests, the evaluation process comprehensively encompasses speed, maneuverability, precision, and entertainment factors, thereby providing a comprehensive assessment of the drones' overall performance.

3.5.4 Performance and Scalability of the Software

The software design was created to facilitate scalability, ensuring the system's capability to accommodate future expansion and advanced functionalities. For the project competition, a set of specific tests was chosen, including obstacle races in a straight line and a semicircle, as well as a freestyle competition. These tests primarily require relatively straightforward trajectories.

However, the implemented software possesses the capability to compute more intricate and complex trajectories based on the positioning of the gates and the desired orientations for the drone to traverse them. By leveraging this software functionality, any trajectory can be calculated and executed, offering the potential for diverse and challenging race scenarios.

The software's flexibility allows race organizers and participants to introduce varying gate configurations, orientations, and course layouts, thereby promoting creativity and innovation in the competition. This adaptability enables the inclusion of more demanding tests in the future, incorporating intricate flight paths and obstacles that push the boundaries of drone racing.

Additionally, the software's trajectory calculation algorithms can incorporate factors such as tightness of the curve, that can provide optimal flight paths to enhance the overall performance of the drone during the competition by taking into consideration inertial parameters. By leveraging advanced mathematical models, the software can optimize the drone's trajectories, ensuring efficient and precise navigation through the designated gates and obstacles.

In this way, the designed software not only caters to the current competition requirements but also provides the foundation for future growth and expansion.

3.5.5 Safety and Reliability

The control system incorporates essential safety measures to ensure the secure operation of the drone. These measures include the implementation of an algorithm specifically designed for emergency landings in the event of low battery levels. This algorithm detects the battery's remaining charge and initiates an automated landing procedure to prevent potential accidents or damage due to sudden loss of power.

Additionally, the control system integrates a fail-safe mechanism to avoid situations where the drone may go out of control. This fail-safe mechanism involves connecting the cfclient antenna to an external device, such as a mobile phone, to enable manual control of the drone in case of emergencies or unexpected flight behavior. This redundancy feature allows the operator to regain manual control and take corrective actions to ensure the safety of the drone and its surroundings.

By incorporating these safety features, the control system enhances the overall reliability and robustness of the autonomous drone, promoting safe and efficient operations during the race.

4. Overall Structure of the Design

In the present section, a more detailed insight on how the system was designed is given. The system design is detailed as it was made: up-to-down.

4.1 High-Level Model Design Structure

To summarize, the high-level design decisions for the project entail:

- Drone Selection: The Crazyflie 2.1 drone was chosen based on its notable advantages, such as its customizable nature and ability to deliver fast and precise performance, aligning well with our objectives of achieving optimal performance and securing a competitive edge in the competition.

- Use of Ground Central Station: The utilization of a ground central station offers increased processing power, extended memory and storage, enhanced connectivity, flexibility, simplified maintenance, and reduced weight and power consumption. These factors empower the drone to navigate races with exceptional efficiency, accuracy, and safety.
- Division of the Competition: The competition is divided into three different races to comprehensively evaluate speed, maneuverability, precision, and entertainment aspects. This approach provides a thorough assessment of the drones' overall performance.
- Software Design: The software has been developed not only to meet the current competition requirements but also to lay the groundwork for future growth and expansion.
- Incorporation of Safety Features: The design incorporates safety features to enhance the overall reliability and robustness of the autonomous drone. This ensures safe and efficient operations during the race, promoting a secure environment for participants and spectators.

4.2 Subsystem Breakdown and Interconnections

Regarding the design decisions made for the various subsystems, numerous factors have also been taken into consideration.

In the context of the environment subsystem, as described in the model, the obstacles must possess suitable dimensions for the drone to navigate through. The Crazyflie 2.1 has specific size specifications of 92x92x29 mm, which are not excessively restrictive but need to be taken into account. Additionally, the obstacles need to be detectable and distinguishable by the vision algorithm. Therefore, it is essential for them to have a predominant color that significantly contrasts with the rest of the racing room. Accordingly, the designated hollow square obstacles, depicted in Figure 4.1, provided by the CAR team, were utilized.

Figure 4.1: Obstacle used in the competition.

The recording cameras should ensure both high image quality and a wide coverage area in order to effectively capture the trajectories of the drones during the competition. Additionally, they should be capable of recording the various testing sessions to facilitate comprehensive documentation for future reference. Consequently, the cameras installed in the practice room by the CAR team were utilized, as they fulfill all the necessary requirements. These cameras offer two distinct angles to record the entire room space, and the option to utilize zoom enhances the capture of the drones' movements. Regarding the drone itself, the decision to use the Crazyflie 2.1 has already been discussed. However, several additional decisions have been made regarding the implemented modules.

In the freestyle competition, the aesthetics and visual impact of the drone play a crucial role. Therefore, the team sought visual features to enhance the overall performance. Consequently, the LED ring deck, illustrated in Figure 4.2, was incorporated into the drone to deliver a more impressive display.

Figure 4.2: Bitcraze LED ring deck implemented in the drone.

Due to the necessity of capturing real-time images from the camera and utilizing them to effectively navigate around obstacles, the Hell-ix team made the decision to implement the AI deck, depicted in Figure 4.3, provided by Bitcraze. This deck enables image capture and analysis, which was deemed more suitable compared to alternative options such as attaching additional cameras to the drone.

Figure 4.3: Bitcraze AI deck implemented in the drone.

Furthermore, the Crazyflie drone offers various methods for calculating its position, which are crucial for determining the appropriate actions to avoid obstacles. These methods range from utilizing sensors installed in the competition room to leveraging the flow deck provided by Bitcraze. The decision was made to implement the flow deck, illustrated in Figure 4.4, as it is a scalable implementation that can be used in a wide range of different scenarios. Furthermore, it provides effective means for the drone to determine its precise altitude, which is achieved by measuring the distance between the drone and the object directly beneath it.

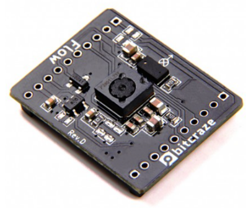

Figure 4.4: Bitcraze flow deck implemented in the drone.

The decision was made to utilize the Cfclient mobile phone app, as it serves as an auxiliary control system for the drone. This allows for emergency situations where the drone needs to be stopped promptly. Consequently, the control manager's mobile phone was equipped with the cfclient app. Although other alternatives such as a video game controller were considered, they were ultimately dismissed in favor of the mobile app.

Furthermore, several crucial decisions have been made regarding the successful execution of the competition trials. It was imperative to design tests that were not excessively long, so as not to drain the drone's battery, while still providing comprehensive evaluation of the drone's capabilities with various implementations. Consequently, three distinct tests were devised to assess the performance of the drones, encompassing different aspects of their implementation.

The first test involves four obstacles arranged in a straight line. This test measures the drone's ability to take off, reach the required height to clear the obstacles, maintain stability while following a straight trajectory, and perform a smooth landing.

The second test entails four obstacles arranged in a semi-circular trajectory. This test focuses on evaluating the drone's capacity to rotate around its axis and maneuver along a curved path. The arrangement of obstacles for these two tests is depicted in Figure 4.5.

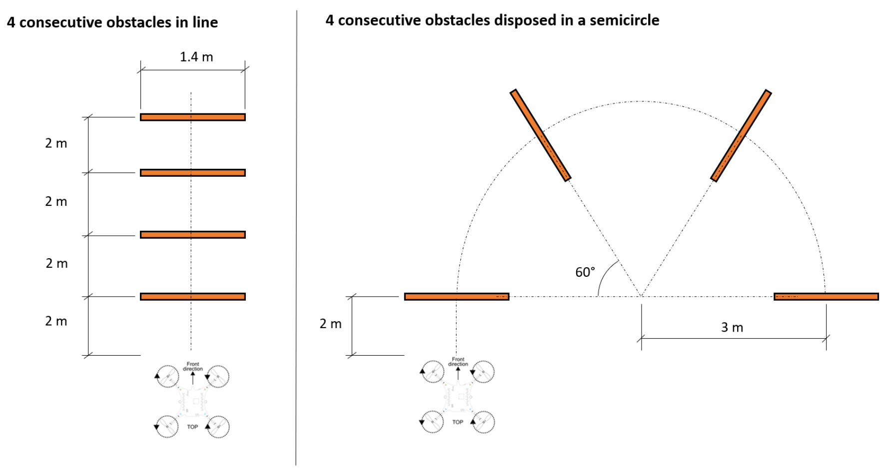

Figure 4.5: Obstacles disposition in the first two tests.

Lastly, the freestyle test was implemented to assess the teams' creativity and the drone's ability to showcase exceptional performance beyond the confines of the standard competition rules. This test encourages participants to demonstrate innovative maneuvers, artistic expression, and unique flight patterns that highlight the full potential of their drones.