UNIVERSIDAD POLITÉCNICA DE MADRID

ESCUELA TÉCNICA SUPERIOR
DE INGENIEROS DE TELECOMUNICACIÓN

MASTER IN TELECOMMUNICATION ENGINEERING

MASTER'S THESIS

DESIGN AND IMPLEMENTATION OF AN AUTONOMOUS SYSTEM TO GUIDE UNMANNED AERIAL VEHICLE INDOORS.

RAQUEL HIGÓN GALÁN

2019

UNIVERSIDAD POLITÉCNICA DE MADRID

ESCUELA TÉCNICA SUPERIOR
DE INGENIEROS DE TELECOMUNICACIÓN

MASTER IN TELECOMMUNICATION ENGINEERING

MASTER'S THESIS

DESIGN AND IMPLEMENTATION OF AN AUTONOMOUS SYSTEM TO GUIDE UNMANNED AERIAL VEHICLE INDOORS.

RAQUEL HIGÓN GALÁN

2019

Tutor: ÁLVARO GUTIÉRREZ MARTÍN

Abstract

The current status of UAVs (unmanned aerial vehicle) or drones is really spread. There are a lot of areas that use them, such as military, surveillance, hobby research. It will be also able to find them as deliverers. At this moment, UAVs technology is booming at the moment and there are a lot of applications where they could be involved to improve human life.

The motivation of this Master's Thesis arises from the enhancement of the daily travelling activities, using small drones to guide inside big buildings or industrial complexes, without damaging the experience of people and to achieve the desired locations taking the short way and saving time.

The main objective of this Master's Thesis consists of the design and development of an autonomous system to control a drone without the human's intervention.

To achieve this objective, the system is comprised by a drone and a camera; they will be connected by means of a computer. The camera will recognize the environment and compute the position of the drone. The data will be used to send information to the drone in real time and to provide the adequate instructions to fly inside a building.

Keywords: UAVs, Drones, OpenCV, indoor drone, autonomous control, motor schema behaviour.

Resumen

El uso actual de VANT, (vehículo arreó no tripulado), del inglés Unmanned Aerial Vehicle UAV o comúnmente dron está bastante extendido. Sus aplicaciones se encuentran en numerosos campos, como militar, vigilancia, entretenimiento o investigación. También es posible encontrarlos actualmente como repartidores. Esto significa, las tecnologías están en su momento álgido, estando involucradas en diversas aplicaciones donde los drones pueden mejorar las condiciones y vida humana.

El propósito de este Trabajo de Fin de Máster surge de la mejora en las actividades diarias de desplazamiento, usando pequeños drones para guiar dentro de grandes edificios o complejos industriales, sin dañar la experiencia de las personas, alcanzando el destino deseado por el camino más corto y ganando tiempo.

El principal objetivo de este proyecto consiste en el diseño e implementación de un sistema autónomo para controlar un mini drone sin la intervención humana.

Para lograr este objetivo, el sistema está compuesto por un dron pequeño y una cámara, los cuales estarán conectados por medio de un ordenador. La cámara reconocerá el entorno y calculará la posición del dron. Estos datos se procesan, generando las instrucciones y las direcciones adecuadas para controlar el dron en tiempo real dentro de un edificio.

Palabras Claves: VANT, UAVs, Drones, OpenCV, drone de interior, control autónomo, motor schemac behavior

Acknowledgment

"Vive como si fueras a morir mañana, aprende como si fueras a vivir para siempre"

Gracias a todos aquellos que han inspirado y dado lugar a conseguir este proyecto. Nunca voy a poder dar las gracias suficientes por no estar sola, y como me han ayudado a retomar el camino, cuando me he sentido perdida. Gracias.

Contents

List of Figures

Index of Tables

Acronyms

UAVs: Unmanned Aerial Vehicles

UASes: Unmanned Aircraft Systems

IoT: Internet of Thing

VTOL: Vertical Takeoff and Landing

LIDAR: Light Detection and Ranging

FPV: First Person Viewing

GPS: Global Positioning System

RTF: Ready-to-Fly

VLOS: Visual Line of Sight

BVLOS: Beyond Visual Line of Sight

NAA: National Aviation Authority

EU: European Union

BOE: Boletín Oficial del Estado

EASA: European Aviation Safety Agency

AESA: Agencia Estatal de Seguridad Aérea (State Agency of Air Security)

PC: Personal Computer

TR: Control de Tráfico Aéreo (Air traffic control zones)

RTH: Return To Home

WIFI: WIreless FIdelity

LED: Light Emitting Diode

PIR: Passive Infrared

MJPEG: Motion Joint Photographic Experts Group

RJ45: Registred Jack 45

DC: Direct Current

Mbps: Megabit Per Second

IR: Infrared Detector

VR: Variable Resistor

BLE: Bluetooth Low Energy

USB: Universal Serial Bus

HSV: Hue Saturation Value

UART: Universal Asynchronous Receiver/Transmitter

Chapter 1

Introduction and Main Goals

1.1 Introduction to Drones

A drone is defined as an unmanned aircraft or formally known as unmanned aerial vehicles (UAVs) or unmanned aircraft systems (UASes). They can be considerated as a flying robot that can be remotely controlled without a human pilot onboard. Also, they can fly autonomously through software-controlled flights with a number of sensors and/or a GPS in their embedded systems [62].

In most cases, UAVs were typically connected with military applications, where they use it for anti-aircraft target, intelligence surveillance and more controversial, as weapons platforms. However, the use of drones outside the military areas has increased remarkably over the last decade. They rapidly spread in commercial, scientific, healthcare, agriculture, journalism, search and rescue, wildlife monitoring, firefighting and other application areas, as photography, drone racing or hobby. Several enterprises have integrated the drones and internet of thing (IoT) technologies. They have created applications such as: drones which are working with the ground sensors network, the drones can help to monitor conditions of land and harvests, in a similar way, the drones can check the state of the power lines and the operational equipment.

1.1.1 Type of drones

The term drone dates to a play on the "Queen Bee" nomenclature [62]. It comes from the Havilland DH82B biplane (see Figure 1.1), which was equipped with a radio and servo control in the back seat.

Figure 1.1: Havilland DH82B biplane.

The drones can be classified depending on several attributes:

- **According to the size**: The size of the drone can vary greatly. They can be classified as shown in Table 1.1 [34].

	Size	Typical
Very small drones (Nano or micro Drones)	From the size of an insect, up to 50 cm.	Usually used to tiny structures or to spy people.
Small Drones or Mini Drones	Will go above 50 cm but will not exceed 2 meters.	Recreation and surveillance. Including the most indoor drones.
Medium Drones	Smaller than aircrafts. Usually need to be carried by two people and can weight up to 200 kg.	Military.
Large Drones	As large as small aircrafts.	Military use them in areas of high risk.

Table 1.1: Size of drones

- **According to the type of aerial platform:** The drones can be separated into two types of groups: the first group is based on rotors, including the single rotor or multi-rotor (such as tricopters, quadcopters, hexacopters and octocopters). Similar to the helicopters, all of them use the helixes to push down the air. The second one is the fixed-wing, which includes the hybrid VTOL (vertical takeoff and landing) these kinds of drones are similar to the conventional planes [35]. Table 1.2 shows the differences between them.

	Pros	Cons	Typical Uses	Price (€)
Multi-Rotor	– Accessibility – Ease of use – VTOL and hover flight – Good camera control – Can operate in a confined area	– Short flight times (20-30 minutes) – Small payload capacity	Aerial Photography and Video Aerial Inspection	4k€ - 41k€
Single-Rotor	– VTOL and hover flight – Long endurance (with gas power) – Heavier payload capability	– More dangerous – Harder to fly, more training needed – Expensive	Aerial LIDAR (laser scanning)	15€ - 200k€
Fixed-Wing	– Long endurance about 16 hours or more – Large area coverage – Fast flight speed	– Launch and recovery needs a lot of space – no VTOL/hover – Harder to fly, more training needed – Expensive	Aerial Mapping, Pipeline and Power line inspection	15€ - 80k€
Fixed-Wing Hybrid	– VTOL and long-endurance flight	– Not perfect at either hovering or forward flight – Still in development	Delivery	In development

Table 1.2: Pros and Cons from [35]

- **According to the Range**

  The range of the flight refers to how far away from the user the drone is able to fly. It is limited by different things, on one hand the supply power, on the other hand the task and shape. These can be classified in [15, 27]:

  - **Very Close range**: The duration of flights is maximum 20 minutes or up to an hour in the case of powerful batteries. They can fly around 5 km. This category includes most of the toys in the market; these toys have about 8-10 minutes of duration.
  - **Close range**: These ones could be controlled up to 50 km away from the pilot. They have higher autonomy than the previous ones, up to 6 hours. Moreover, they can fly for long durations and cover far distances.
  - **Short range**: This group includes drones that can fly up to 150 km from the user and the estimated flight time is about 8 - 12 hours.
  - **Mid range**: These drones can fly up to 650 km and get higher speeds than the anterior groups. They are commonly used in meteorological data collection and the basic surveillance applications.
  - **Endurance range**: These types of drones are used in long distances requirements. There are capable to fly from 36 hours up to days and can reach high altitudes, about 3000 ft over the sea level (around 914.4 m).

- **According to the Abilities or equipment**

  A drone is a machine than can be used to a particular purpose or task. Its abilities to realise a specific job can create different configuration. For instance [15, 27]:

  - **GPS Drones**: These drones are connected to satellites by means of a GPS. They can take location information during the flight and return home after finishing the task.
  - **RTF Drones**: These drones are named Ready-to-Fly, it means they only need to be unboxed and charged before flying. They are easy and quick to fly.
  - **Delivery Drones**: These drones are new and becoming more popular. They are used to transport packages due to the basket attached to their body. The sort of goods that can delivery are limited in size and weight. However, the main companies are investing in bigger and better delivery drones.

- **Racing Drones**: These drones can fly fast and agile and they are used for entertainment. Nowadays, there are huge number and type of races. The FPV Drones (First Person Viewing) are very common. These drones record and transmit the video in real time. Its control is possible by using mobile monitors. These drones can also be used in other applications like film industry, for example when it is needed to control the area or to check the scenery in real time. .
- **Photography Drones**: These drones typically use accessories as cameras attached to their body. They allow to record and taking HD quality videos and photos of the flight from the drone perspective. Besides with the racing drones is in most cases the video is recorded and storaged to be processed when the flight is over.
- **Alternative-Powered Drones**: Even thought the most popular drones are electrical, the limit of their batteries does not allow to fly long distances. This is one of the reasons why the gas power or nitro-fuel drones are developed.
- **Toy Drones**: These drones have the typical shape of quadcopters. They use 4 rotors located at the end of a squared body. In this category we can found the trick drones, which are used to perform battles, flips and other entertaining motions in the air.

1.1.2 Regulation Drones

It is also important to revise the regulation. The EU has designed a new regulation for the drone flight planes. The main objective is the "safety flight" even when the sky is busier. This new regulation has been approved the 12^{th} of June, 2018 [18, 21, 70] and was a revision and an update from the 2008 regulation. Until now, the drones lighter than 150 kg have not been controlled. However, the new regulatory framework covers all type of existing and future drones operations, which is based on the following principles:

A **risk-based and proportionate approach.** Three categories have been included: open, specific and certified in accordance with the level of risk. Each category needs a different regulation. The "open category" is referred to the low risk and does not require any authorisation, but will have strict limitations. The "specific category" is related to the medium risk, the users need an authorization from the National Aviation Authority. The last one "Certified category" is associated to the high risk and the classical aviation rules are applied.

A sharing of responsibilities between the EU and the member states. The member states need to define some "zones" to restrict the access to some airspace or

<table>
<tr><th>Category of operations</th><th>OPEN
Low risk</th><th>Specific
Medium risk</th><th>Certified
High risk</th></tr>
<tr><td>Authorisation needed</td><td>None</td><td>Authorization from NAA based on operational risk assessment or specific scenario</td><td>Authorisation from NAA/EASA</td></tr>
<tr><td>UAS</td><td>Compliant with Commission Delegated Regulation on UAS</td><td>Compliant with requirements included in the authorisation</td><td>Certified UAS</td></tr>
<tr><td>Operations allowed</td><td>Restrict to:
• VLOS
• Altitude < 120 m
• Other limitations defined by:
– Commission Regulation on UAS operation
– National airspace zones</td><td>Restrict to:
• Operations specified in the authorisation
• Limitation defined by National airspace zones</td><td>Controlled airspace
U-Space</td></tr>
<tr><td rowspan="2">Regulations</td><td colspan="2">Commission Regulation on UAS operation in open and specific</td><td rowspan="2">Revision of the existing aviation regulation</td></tr>
<tr><td>Commission Delegated regulation on UAS</td><td>No regulatory requirement (UAS requirement included in the authorisation)</td></tr>
</table>

Table 1.3: Summary of the authorizations [18]

on the opposite relax the conditions due to a necessary flexibility. Our system will be included in the open categories because it is controlled in an inside building and with the idea of control research. Even thought, a pilot licence is not required, there are several strict rules: the drones must fly away from any airport, in visual line of sight bellow 120 m altitude and respect the specific rules defined by each zone. Also, the distance from the people varies from the weight: under 900 g, they can fly over uninvolved people (not over crowds), under 4 kg they can fly at a safe distance from people and under 25 kg they should fly keeping a secure distance from urban areas or limited areas. However those regulations and conditions are subject to change during the adoption process, Table 1.3 shows a summary of the authorizations [18].

1.1.2.1 Drone regulation in Spain

Spain has approved several new important regulations about the use of drones with the idea to protect the user, as well as the physical well-being, residential areas and others environments. The changes have been added on 30^{th} of December, 2017, and the new law is published on the BOE [3, 74]. This law favours the user, since it is not necessary to have a licence to fly as recreation or hobby. The professional cases require specific licences that the AESA must grant. The conditions to fly a drone without licences are [32, 20, 63]:

- The distance to fly with regard to the airport has to be more than 8 km, including the aerodrome. These areas are controlled and limited to create a safe perimeter.
- There are areas where the flight is forbidden, they are called Controlled Airspace.
- The flight must be less than 120 meter of high over the ground level or the level of the highest obstacle around 150 m.
- The flight conditions should be on peaceful weather. It is recommended to fly during the light hours to avoid accidents. However, night flights are enabled but the drone must weigh under 2 kg and must have lights on to control its location. Also, the flight could not be over 50 m.
- All flights have to be at the visual line of sight.
- The flight is allowed at the urban areas and above the crowd, but only with drones under 250 g and they should fly at 20 m over the ground level and far away from the buildings.
- The drones are considerated as aircrafts. Due to that fact, they cannot enter in the airspace. These zones are controlled by the AESA, which has created several places of open source to realise a safe flight.

It is important to know where are that zones and it can be checked them in different websites as IcarusRPA [1] or enaire[2]. An example from these zones is shown in Figure 1.2 from the IcarusRPA pages. The app shows the Spanish map with different zones. The categories of those zones mean different flight conditions.

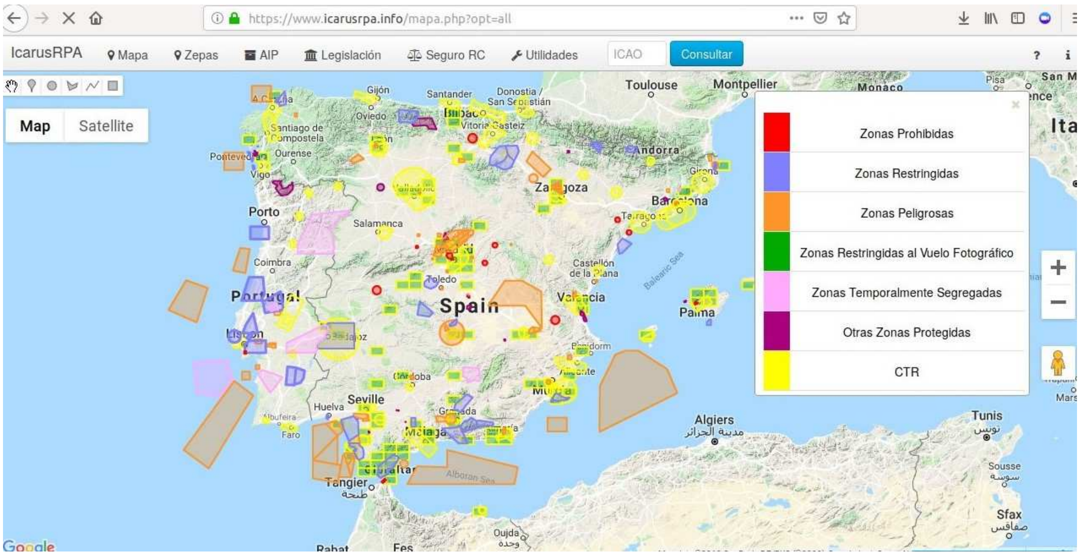


Figure 1.2: IcarusRPA pages: flight zones

- **Forbidden zones:** Every place close to airports, naval bases, air bases, military control areas, governmental places, etc, are prohibited.
- **Restricted zones:** Zones such as national parks or historic places where their wildlife or structures can suffer damage. However, it could be possible to fly if access permission is applied.
- **Danger zones:** These areas have structures or power supply lines, industrial activity or even nuclear areas.
- **Sensitive zones:** Those areas in which a high risk of damage to the fauna exists. It is right that in every zone birds can be found; however, the sensitive zones are referred to the habitat or migratory routes for huge number of birds. That is the reason why many drone flights are restricted.
- **Air traffic control zones (CTR):** The major areas around airport and heliports.

[1] https://www.icarusrpa.info/mapa.php?opt=all
[2] https://drones.enaire.es/

- **Restricted to photographic flight zones:** Those places where the use of cameras is limited due to privacy or security reasons.
- **Protected zones:** are areas related to the national parks, natural reserves, archeological sites, etc

The new law demands to each user, professional and amateur pilots, that the drone and the remote control are clearly identified by using a fireproof identification plate, with the owner's name and main contact dates. The identification information of the drone has to be the manufacturing name, type of drone, model of drone, serial number, owner's name and telephone.

If we take a look to the conditions to fly at a professional way, they are controlled by the AESA and required an authorization from that governmental agency.

The urban or rural zones are allowed, but the drone weight has to be less than 10 kg, otherwise the flight is cancelled. Also, the drones have to fly at Visual Line Of Sight (VLOS) to avoid accidents. A safe margin has to be as follow: no person should be 50 m around the pilot, and other horizontal 50 m far from the buildings too. The drones must have a fall absorber to minimize the damage in an accident.

The night flights are also possible, but they have to be approved by the AESA after presenting a flight plane and safety study. The drone has to be identified with lights too.

The flights beyond the visual line of sight have to obey similar rules to night flight. The same studies must be presented and approved by the AESA. Moreover the drones have to count with a tracking system and never fly over the limited frequency control.

To summarize, the drone flight are allowed at urban areas, flights over crowd, night flights, flights under authorization at airspace controlled, visual line of sight flight and beyond visual line of sight. However, they have to comply some restrictions and the drone should have a fireproof identification plate. This law is what some people call "Dronecode".

- **D**on't fly near airport or airfields.
- **R**emember to stay below 120 m (400 ft) and at least 50 m (150 ft) away from buildings and people.
- **O**bserve your drone at all time.
- **N**ever fly near aircraft.
- **E**njoy responsibly.

1.1.3 Commercial programmable drones

Most drones that can be found at the commercial market are for entertainment or to professional tasks. Even though, the last group is more typical as a service enterprise, where the customer can rent the drones during some hours or days with or without the pilot. Also, most drones are designed to be flying with a pilot who controls their movements.

The drones use at investigation or programmable are really scarce and limited at the different things or sensors that they possess. In particular, these are examples of drones that are possible to find:

- **Matrice 100 (see Figure 1.3a) [22, 23] :** This drone is from DJI company. It can be programmable using the SDJ DJI software. This system automatically manages the most complex tasks using their elements. The drone includes the flight controller, propulsion system, GPS, DJI Lightbridge, a dedicated remote controller, and a rechargeable battery. It has an autonomy of around 40 minutes and can support up to 1 kg payload capacity. The flight time varies depending on the payload and flight conditions. It counts with an expansible connector where new sensors can be added. It is usually found for research or for business, but it is sold for fun too. Its price is about 3599 €. DJI has other autonomous drones such as **Matrice 600** or **Mavic Air**.
- **Shooting Start (see Figure 1.3b):** Intel entered to the drone market with the Falcon8+. Although it has another commercial drone, the Sirius pro drone, it is focused at agriculture activates. Moreover, the Shooting Start drone was used to impress to the world by Intel with their shows of 1218 Shooting Start flying together at the PyeongChang Olympic Games. This drone is used to prove the high quality of their drones at real time and aerial systems. However, the Ehang company has the same purpose in its GhostDrone 2, they are focused at light shows too. The Shooting Start is controlled by the GPS and a centralized system. Intel has not posted their prices of any of their drones, and it is necessary to contact a commercial person. However the estimation is over 10000 € each one. Other important feature from that company is the special software that complements them, such as Intel Insights Platform [10, 12, 73, 33, 14, 42].
- **Tarot 650 Ready To Fly (see Figure 1.3c):** it is an alternative to the Matrice 100. It is manufactured by the United States and Mexico. It is designed to carry a wide different sensors, cameras and payloads under 1.5 kg. It is mainly used for research missions, general photography and surveillance. The cost is around $1,499.99 USD (about 1334 €). This company has another similar drone **Tarot X6**, as an alternative to DJI Matrice 600 [75].

(a) Matrice 100

(b) Shooting Start

(c) Tarot 650 Ready To Fly

(d) Kargu

(e) Alpagu

(f) Togan

Figure 1.3: Commercial models of drones

- **Kargu, Alpagu and Togan (see Figure 1.3d to 1.3f):** Both of them are from **STM** organization. **Kargu** is an autonomous rotary wing attack drone; it can fly in autonomous or manual mode. The system has embedded real-time image processing and deep learning algorithms. While **Alpagu** is a fixed-wing autonomous tactical attack UAV. It can be operated autonomously or via a remote control, has embedded processing and learning algorithms too. However,a launcher is needed to put it to fly. Meanwhile **Togan** is similar to Kargu. It is an autonomous rotary wing reconnaissance UAV system, although its exact price is difficult to find [36, 4, 72, 68].

- **Crazyflie 2.1 (see Figure 1.4a):** It is the last version of the Crazyflie. It is an open source flying development platform which documentation of software and hardware is available online. It weighs only 27 g and can be controlled by a long range radio or Bluetooth LE. It was designed for developers and researchers. The board has a JTAG/SWD connector to update the firmware and test the systems. Other expansion boards can be attached to add different sensors. The flight time is about 7 minutes and a payload weight under 15 g is recommended. Their price is about $243.75, around 217 € [13].

- **CoDrone (see Figure 1.4b):** Robolink has created this drone, the company has different kits; Codrone Pro [57] and CoDrone Lite [56]; the first one includes a programmable remote control base on the AtMega microcontroller, the second one has not the programmable remote control. The kit needs to be built. The programming can be made by different options, using Arduino, Python or Snap;

the last one is an HTML program where different blocks and labels are joint to create a simple and easy program. The drone can be transformed to create a car by different accessories or kits. Also, it is possible to add a different frame with an embedded camera. The flight time is up to 10 minutes. The control and connections are made by Bluetooth. The small design and the stabilizer made it perfect for indoor use. It was designed to teach how to program. Its price is about $179.99, around 158 €.

(a) Crazyflie 2.1

(b) CoDrone

(c) Parrot Mambo Code

(d) Airblock

Figure 1.4: Commercial models of drones

- **Parrot Mambo Code (see Figure 1.4c):** This drone is created by the company of Parrot [54]. This kit includes a Parrot Mambo and a 6 month subscription to Tynker. The Tynker application is based on simple labels and visual blocks that the users have to joint to create their own program, as the Snap of Codrone. The flight can be controlled by mobile or tablets, where the link connection is based on Bluetooth BLE. It can fly up to 8 minutes. In regard to the information, the kit is mainly developed to teach how to program to the kids. Its price is about $149.99, around 134 €.

- **Airblock (see Figure 1.4d):** Airblock has a magnetic design, where the different modules can add to a central core by a simple touch. The main idea is to connect up to six power modules and create a drone that can fly. However, it can create other structures and different robots which can move at sky, water or land areas. The time flight is about 6 or 8 minutes at the sky and up to 40 minutes by water or land. As in some others, the link connection is made by Bluetooth. Although, it is a modular and programmable flying robot, it does not have any kit or connector to add other sensors as cameras. It is a transferable robot specially for children. Their price is about 109 € [40, 17].

- **STEVAL-FCU001V1:** Recently ST has designed their its flight controller evaluation board for mini drones. They count with a simple kit that allows mounting the complete drone. It is designed to support quadcopter structures which is frame can be printed by the 3D stl file given on the page [64, 66, 65]. The FCU can be controlled by a standard external remote controller or by a device though the Bluetooth. The time flight depends on the battery selected, but it is around 8 minutes. The price of the board is $33 about 27.73 €.

The drone selected for this Thesis is the CoDrone pro, and the important parts and elements will be explain in detail in the Chapter 2.

1.1.4 Problems indoor drone

This section collects some possible problems that the indoor flights present. Some of the problems are found in an outside area too [39, 9, 31, 25].

Most drone users would prefer to fly outside, because of the landscape. All kind of buildings have been record inside and outside with professional drones for commercial purpose.

However, flying an inside drone is very different to an outside one. There are some very important things to be conscious about.

As well as in the outside, before starting any flight, it is necessary to check that the batteries and remote control are fully charged and corrected adjusted and when the drone takes off, it is necessary to analyse if the drone is flying as normally. When the drone is flying indoor, the current air makes it to lose the balance easier due to the reduced space. In other words, the drone will take longer to stop and balance than outside where there is more area to land. Also, it is important to verify the motors and the correct propellers direction.

At open areas, it is easier to see whether someone cames into the place that the drone is flying and there are plenty of space to move away anywhere else. Indoor environments are very different, and even though the people are warning or there are signals advising the operating area, somebody could suddenly walk into the area and the reaction time and space is reduced.

It is important to remark that to touch the propellers of a drone could injure the person. The bigger propellers are more dangerous than the small ones. However, the small ones can produce haematoma or bruise.

If a signal control is lost, it is necessary to determine what could happen. When the flight is outside and the drone lost the connection with the remote control; it can send the order to return to home (RTH), that means, the drone will fly until an established localization, which is normally configured before to take off, and mainly makes the drone to land. But, maybe that possibility is not suitable for indoor flights. There may be a lot of magnet or electronic interference from the wifi, as well as a lot of mobile phones, radio and machines, even the material of the building may cause to have signal problems. Most importantly, the GPS mode will not work indoors.

When the drone takes off, the downdraft moves things that may not suppose to be there, such as dust, sand, fluffs or debris, which can hook the rotors and hit the drone. Moreover, if the drone gets too close to the ceiling the rotors can crash it. Therefore, it is necessary to watch the roof. In a building, it is not so simple to realise how close the ceiling is and most drone cameras are not designed to check it. The angle and perfectives are focused on the ground so on unless the camera is changed or other sensor are used to the help in the location of the drone, an extra attention is required, with the furniture, walls and doorways too.

Mini drones are better to fly indoor, but the low duration of the batteries around 8-10 minutes make the flight limited and sort. Creating the necessity of having more than one battery.

(a) Crash proof frame (b) 360º Structure

Figure 1.5: Structures and protections

The small body and structures are broken easily if the drone crashes to the walls or objects. Most of the drones have accessories to protect the drone. These structures protect the drones from hitting the wall and touching the object safety to avoid troubles. Other option is to use a tripod that holds the body of the drone and limits its movement to a determinate area [30]. Those structures allow to control the flight and move 360º. An example of the structures and frames can be shown in Figure 1.5.

The calibration process is the same for most mini drones [24]. After pairing the drone and the controller, the drone is placed on a flat surface and made it fly (taking out and land again). This process depends on the manufacture and the drone too. This motion is required to recover and restore the measurements from the different elements and sensor, such as accelerometer, gyroscope, thermometer, barometer, etc.

Other important recommendation is to have an accident insurance which covers the repairs or the flaws, what can be cased during the flights. Due to the fact that the drone could crash and break the objects or if any person is damaged during the flying. It is needed to emphasize that this insure are a recommendation, the regulation does not oblige to have one.

1.2 Objectives and methodology

According to Section 1.1, the main goal of this Thesis is to develop an autonomous system which controls an inside drone without the human's interaction.

The motivation of this Master's Thesis arises from the enhancement for the daily travelling activities, using small drones to guide inside big buildings or industrial complexes, without damaging the experience of the people and to achieve the desired locations taking the sort way and saving time.

To achieve this objective, the system is comprised by a drone and a camera; they will be connected by means of a computer. The camera will recognise the environment and compute the position of the drone. The data will be used to send information to the drone in real time and provide the adequate instructions to fly inside of a building.

The project has been split into four intermediary milestones to cover the different objectives and getting the main goals of this Master's Thesis:

1. A deep study about different commercial programmable drones, focused in the economical aspects due to the tight budget.
2. The connection and detection of the drone by the camera.
3. The communication from the main PC to the drone.
4. The autonomous flight control plan by the main PC and the different environments.

1.3 Document organization

This document is organized in six chapters, which involve the different parts to achieve the objectives stated in Section 1.2.

- Chapter 1 has developed the introduction, the motivation and main goals of the autonomous systems and UAVs.
- Chapter 2 selects the tools and material to create the secure system including the required programs.
- Chapter 3 details the software and the algorithms implementation.
- Chapter 4 introduces the motor schema behaviour and the drone motion has been described.
- In Chapter 5 some experimental test and an validation study from the hole system has been described.
- Chapter 6 exposes conclusions and future lines of this Master's Thesis.
- The economical and environmental studies of the drones in the human areas are shown in the Appendices.

Chapter 2

Elements and programs

The material and equipment, as the software or program required are going to be detailed and exposed in this Chapter.

2.1 The components of the system

The system is composed by a PC, a camera and an indoor drone, from the kit of Codrone Pro. In this Section the main elements are going to be explained.

The system is controller and executed from a computer i7 core, 4GB of RAM and 100Gb of HD (see Figure 2.1), who has an Ubuntu 18.04.02 LTS installed. This computer executes, processes, estimates and controls the main communications. In other words, it is the core of the whole system. The reason why a computer is the centre is because the huge calculation and power supply consumption.

Figure 2.1: Characteristic of the operative system

The camera is responsible for detecting the environment, as well as the drone. The camera selected is a stream camera AXIS M1033-W shown in Figure 2.2 [77, 78]. This camera is an indoor wireless IP camera connected by the WIFI. It has a microphone and a speaker. A white light LED located in the front can illuminate the scene, getting better images. The PIR sensors can detect heat sources and send events based on the motions at darkness environments. This compact camera is sold mainly for surveillance and remote monitoring in houses, offices, shops, restaurants and so on. It means places where a 24h monitoring is necessary. This camera is updated by the Axis M1034-W.

The whole system is configured by accessing to the page of Live view. Those settings allow configuring the image at different formats, from the maximum of 800x600 to 320x240, and up to a megapixel of resolution. The video can be compressed by using MJPEG and H.264 formats and the network is achieved by a 10/100 Mbps RJ45 plugging. A power connection is mandatory too. The power connection is a DC jack type. When the input voltage is connected, the camera notifies it by a green LED. In this Thesis, the camera has been configured and fixed using the 800x600 resolution. The stream video profile used is the motion JPEG and an unlimited maximum frame rate is selected. The rest of sensors have been disconnected [19].

Figure 2.2: M1033-W Camera

The last important element is the Drone [67]. As it was introduced in Chapter 1, the CoDrone pro kit has been selected. The kit includes a CoDrone, the controller build set (joysticks, wires, battery pack, various frame, etc), a Smart inventor board, bluetooth module, the USB cable, a battery charger and the online tutorials.

The Drone is similar to Petrone fighter, which is also from By ROBOT. The drone has an autonomy up to 8 – 10 minutes, what can get from a LiPo 3.7V battery of

300mAh. However, the battery charge time is 40 minutes. The size of the drone is 133x133x25mm and just 37 g of weight. The drone is connected by Bluetooth 4.0, getting a flight range around 160ft (nearly 50m). The drone includes several sensors, for instance: an IR sensor, optical flow sensor for hovering, barometer sensor that assists with the altitude control a 3-axis gyroscope and a 3-axis accelerometer for altitude control too. Moreover, the drone includes LEDs to be located during a night flight Figure 2.3 shows these features.

Figure 2.3: Characteristic of the Codrone

The smart inventor board is an Arduino compatible circuit board; it has an ATmega microcontroller embedded, where the different sensors are connected, as well as the peripheral connectors. There are 7 IR sensors along the bottom layer of the board, and 8 LEDs located at the top layer, which can be used to signalization. The IR bottom sensors can calibrate their sensibility by a variable resistor located at the bottom. Moreover, the board has a piezoelectric buzzer that uses the vibration to produce different tones. There are 3 analogue IR sensors, used to navigate and select which program is required to run. It is important to avoid light or sunlight ambient; on the other hand the sensors will be also affected. The board counts with a DIP switch of 3 different modes, these modes permit to select the program to run or be updated. These switches are used to control de IR receiver sensor, by means of 4 distinct channels. Respect to the connectors, the board has 4 DC motor control pins, 5 digital pins that can be used to expand the board by adding any Arduino compatible components as LEDs, servos, etc. Moreover, there are 5 analog pins too, which have the same purpose than the digital pins. There is an UART and a Bluetooth connector to connect other elements. In addition the board can be powered by an array of batteries. It is important not to connect 2 different powers at the same time. These elements are located at the pinmap image from Figure 2.4 [61].

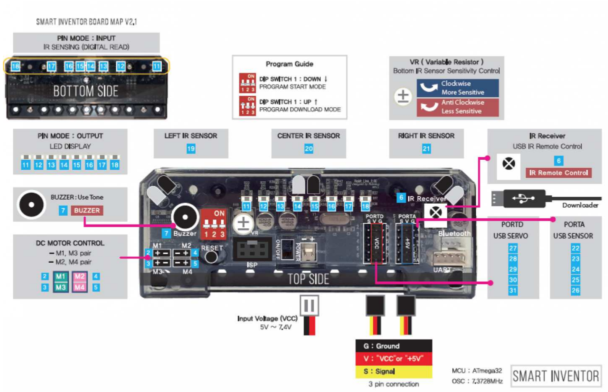


Figure 2.4: Smart inventor board

The CoDrone BLE (see Figure 2.5) is the Bluetooth Low Energy board. This board has a USB connector and a UART connector to connect to the Smart Inventor Board. The data structure used and the different commands are documented on its manual [60]. However, that information is poor and shows that there are a lot of functions that are currently not implemented.

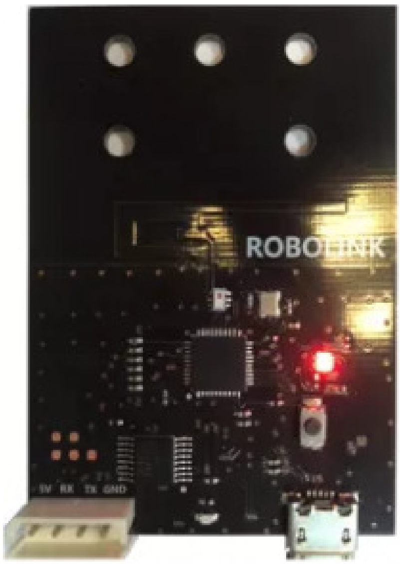


Figure 2.5: CoDrone BLE board

The documentation about the smart inventor board shows that it can use different serial ports. However, it is not right, the second serial port is not initialised at the libraries, and the Robolink support could not solve it too. That is the reason why the Smart Inventor Board has been changed to an **Arduino Mega board**, to interconnect the different systems [59].

However, the smart inventor board is used to verify and calibrate the drone flight, and to check the link connection. In addition, it is used when a problem is detected and needed to be diagnosed.

The system is connected by a CP2102 converter from USB to UART. The module permits to communicate between the microcontroller and the PC by the USB protocol. The module is suitable with any microcontroller as PIC, Atmel, Arduino and ESP8266. It is similar to FDI232 and PL2303HX controllers, but with the advantage of better prices and higher number of support drivers. The use of a USB converter has facilitated the integration of the PC programs, as Matlab, Labview, Processing, etc. At the program level, the converter is transparent because the microcontroller only uses the classical serial protocol and avoids the complexity of the USB protocol.

The technical specifications are collected in Table 2.1 [37, 41]:

Description
2.0 USB specifications and a speed of 12 Mbps
USB connector of type A
Out pins at TTL level: +3,3V, RST, TXD, RXD, GND and +5V
USB embedded transceiver and the external residence are not required
A crystal oscillator is integrated
Internal regulator of 3,3V
The reception buffer up to 576 Bytes
The transmission buffer up to 640 Bytes
The operating temperature from -40°C to 85°C
Operation system : Windows 10, 8, Vista, 7, XP, 2000, 98SE and Linux 2.40 and higher
Measurements: 21mm x 16mm

Table 2.1: Technical specifications of CP2102 converter

Figure 2.6 explains an example of the system diagram and performances. The characteristics were taken from the datasheet [38] and represent the main use of the driver as is manufactured in a PCB module, the modulo is represented in Figure 2.7.

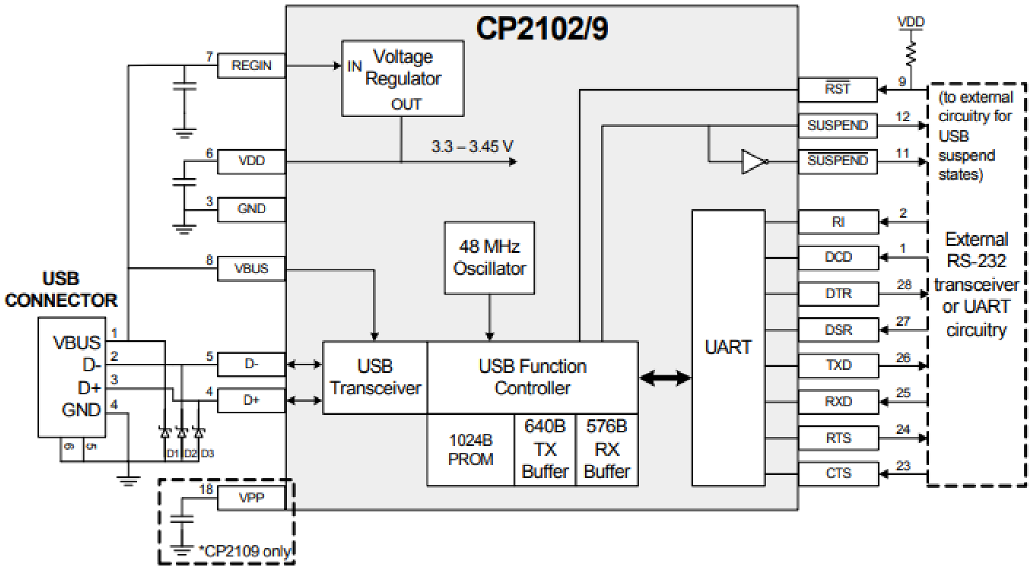


Figure 2.6: Diagram of the CP2102 converter

Figure 2.7: CP2102 converter board

As well, the secure environment created is shown in Figures 2.8.

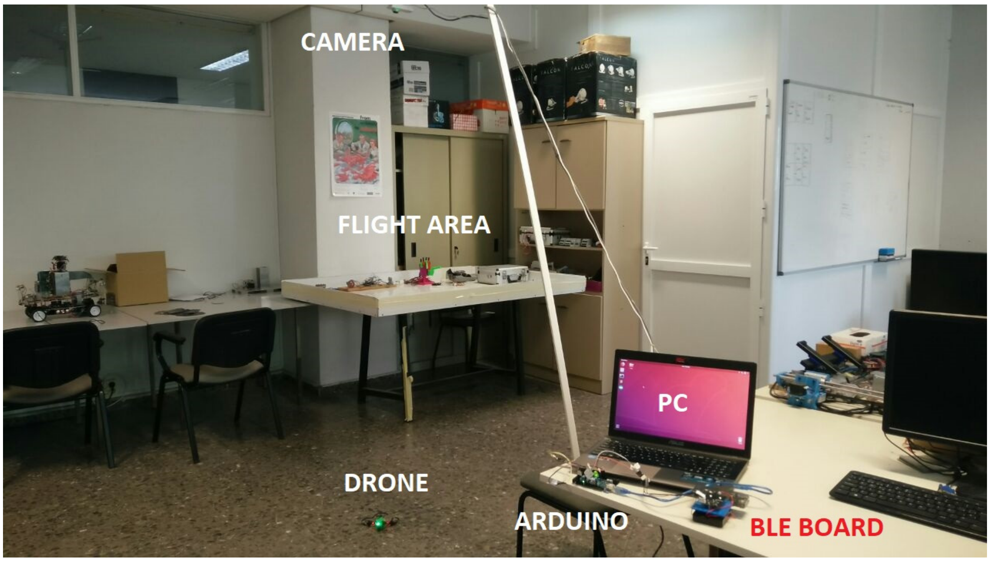


Figure 2.8: Environmental system

2.2 Software required

This Section introduces the main programs to execute, program and control the system. These programs or libraries could be separated into two groups, OpenCV and Arduino.

OpenCV is an open source Computer Vision Library, officially launched in 1999 [45, 44]. It was developed to create and produce a general infrastructure for computer visual applications and to stimulate the use of electronic systems in commercial products. There are over 25.000 optimized algorithms in the library, which include the classical and machine learning algorithms, which can be used to recognise or identify facial expression, detect objects, classify human behaviour in videos, record and track camera movements, as well as tracking objects, generate 3D models from objects, create 3D point from stereo cameras, join images together to construct a high resolution image of an entire scene, find and locate similar images from a database, modify and remove photos, as red eyes, follow motions from the eyes, identify scenery, etc.

OpenCV has a huge community around 47 thousand users and an estimation of 18 million of downloads. The library could be found at different sectors like in companies, research groups, and governmental groups. For example, important and well established companies could be found using the libraries, within these organizations highlight Google, Intel, Microsoft, IBM, Sony, Yahoo, Honda, Toyota, so on. However there are lot startups that have introduced it too, like VideoSurf, Zeitera,or Applied Minds.

OpenCV counts with several interfaces: C++, Phyton, Java and MATLAB and supports different operative system: Linux, Windows, Mac, and Android. Nevertheless OpenCV has been written nativity in C++. OpenCV has different stable versions [29]. The version installed and selected to develop the whole system is 3.2.0.

The second main program is the **Arduino IDE** (integrated development environment). It is an open-source to write code and upload it to the board, the environment is written in Java and supports Linux, Windows, and Mac operating systems. Arduino support the languages C and C++. This interface allows to install different libraries from contributions or other companies. The library could be installed from the Library manager of Arduino [5], and it can be modified and adapted to user requirement. The CoDrone library has the principal function documented on their page [58]. However, there are other functions too, which are required to analyse and to understand the code, if any changed is needed. The Arduino IDE version used is the 1.8.7, it is used to program the ATmega board, using the CoDrone library. This library permits to control the communication with the Drone. It has implemented the Bluetooth communication to send the principal values.

In addition a driver is needed to connect the different systems, the **CP210x USB to UART Bridge VCP Driver**. This driver allows the USB communication to UART and vice versa. It is made by creating a virtual port and using the CP2102 module.

Chapter 3

Software applications

This Chapter focuses on the description of the different developed program parts, as well as the code running on the computer or on the Arduino. The main program can be divided into tree principal functions:

- Firstly, the camera; referring to image processing.
- Secondly, the serial communication between the computer and the Arduino board.
- Finally, the autonomous control.

This Chapter details the camera and the communication. The autonomous control is explained in Chapter 4.

3.1 The camera image processing

The main objective can be summarized into: take the image from the camera, to filter the image to detect the drone, to track it during the motions and to display the executed results. The whole process is developed using the OpenCV library in the C++ language.

As it was explained in the Chapter 2, the camera is a stream type. The image is opened using the VideoCapture class. Those types of camera are supported by the library at version 3 of OpenCV or higher. That is another reason why a computer is selected instead an embedded board. The VideoCapture class opens the camera by using the complete path, it means, including the protocol, user and password. The computer used to run the program has to be at the same network that the camera. Because the camera has a fixed private IP, the system will unable to connect and establish the communication if the PC is outside the network. The instruction to open the camera is (see Code 3.1):

```
VideoCapture cap("http://USER:PASSWORD@XXX.XXX.XXX.XXX//mjpg/video.mjpg
   ");
```

Listing 3.1: Open the camera code

Once the camera is assumed opened, it is necessary to check if the access was successful (see Code 3.2).

```
if(!cap.isOpened()) {
  cout<<"Camera not found"<<endl;
  return -1;
}
cout<<"Camera 1 open"<<endl;
```

Listing 3.2: Access camera checked code

The OpenCV library works with frames, that it is to say that an image from the camera is read and transformed to numerical values for each point of the image. It means that inside the computer all the images are reduced to numerical matrices and other information describing the matrix (see Figure 3.1).

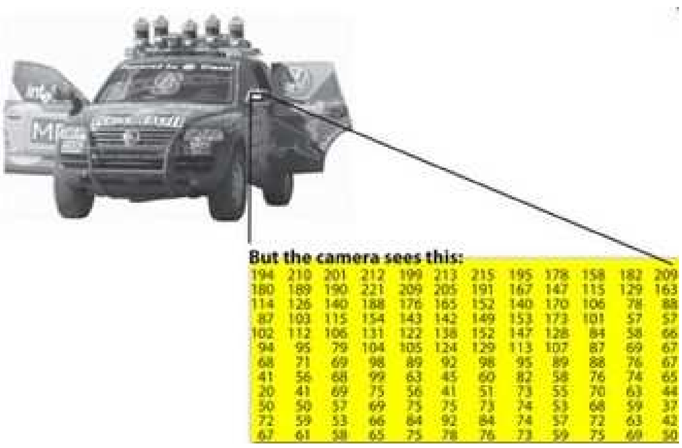


Figure 3.1: Example of how the OpenCV treats an image [51]

These values are stored at the library as **Mat** [51]. Mat is a class with two parts, the matrix header and a pointer to the matrix of the pixel information. The first part, the header, is a constant, whereas the size of the matrix can vary from one image to other, and the differences may be of orders of magnitudes. With this in mind, when a copy from a frame is made, the stored information only keeps the header and the pointer, saving computational resources. Hence, the matrix with the complete original numbers is stored once at the beginning and a pointer to it is generated.

In particular, the program will create a matrix call “frame” with the original values of the camera (see Code 3.3). So on, during the program, other matrices are created to compute and store the information without modifying the original one. In addition, while the camera is open, a new frame is taken to pass through the same filters. There are distinct methods to record a frame from the camera. The most used is the function “read” or the symbol “≫”.

```
Mat frame; //current frame

cap.read(frame); //option 1 to record frame by frame
cap >> frame;    //option 2 to record frame by frame
```

Listing 3.3: Read a frame code

When the image matrix is saved on the memory, the software treats it and detects the drone by its colour [47]. First of all, the original frame is converted from RGB to HSV colorspace. The HSV is a model to present the colorspace as RGB model. The HSV means Hue, Saturation and Value. The Hue value generates the colour type, the Saturation covers from the unsaturated; referring to the shades of gray; to completely saturate without the white variable, and the last one Value, describes the intensity or brightness of the colour. These can see clearly at Figure 3.2. However the RGB colorspace uses three channel to modulate the colour. It results on a more difficult segmentation to detect an object in the image. The RGB cube is plotted at Figure 3.3.

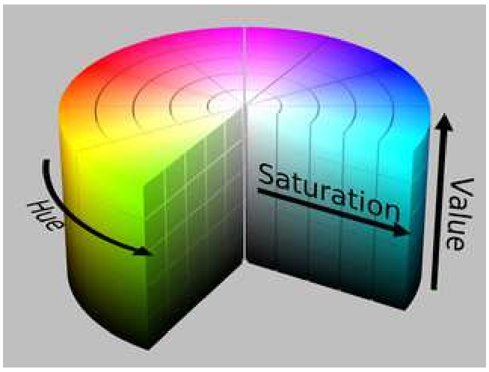


Figure 3.2: HSV cylinder from [47]

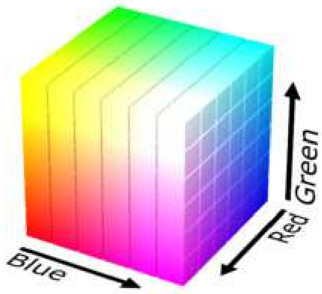


Figure 3.3: RGB cube from [47]

The RGB conversion at 8 bit is made using the following equations [46]:

$$V \leftarrow \max(R, G, B) \tag{3.1}$$

$$S \leftarrow \begin{cases} \dfrac{\max(R,G,B) - \min(R,G,B)}{\max(R,G,B)} & if \ \max(R,G,B) \neq 0 \\ 0 & otherwise \ R = G = B = 0 \end{cases} \tag{3.2}$$

$$H \leftarrow \begin{cases} \dfrac{60(G-B)}{\max(R,G,B) - \min(R,G,B)} & \text{if } \max(R,G,B) = R \\ 120 + \dfrac{60(B-R)}{\max(R,G,B) - \min(R,G,B)} & \text{if } \max(R,G,B) = G \\ 240 + \dfrac{60(R-G)}{\max(R,G,B) - \min(R,G,B)} & \text{if } \max(R,G,B) = B \\ 0 & \text{if V = R = G = B = MAX = MIN} \end{cases} \tag{3.3}$$

If $H < 0$ then $H \leftarrow H + 360$. Moreover, $V \in [0,1]$, $S \in [a,b]$, $H \in [0,360]$. The RGB values must be at the range $[0,1]$.

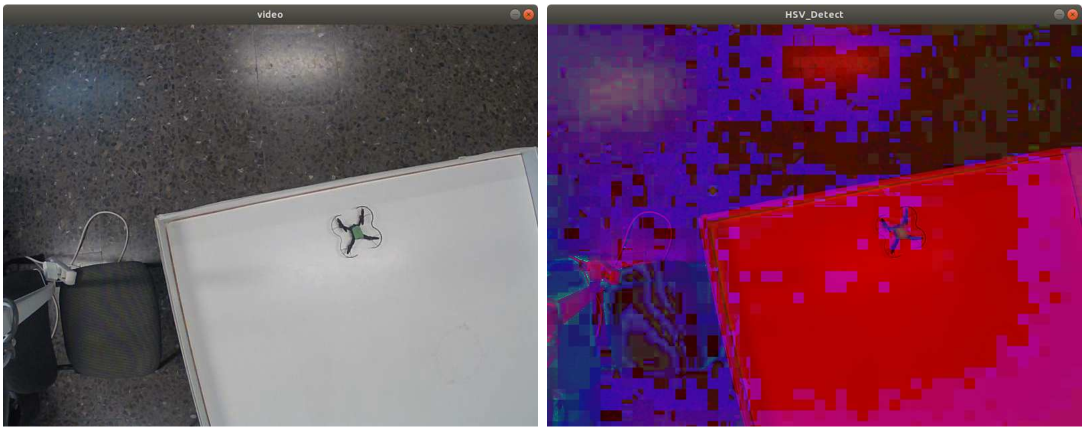
After identifying how the process is made and the reasons to transform the image to HSV. The colour detection is optimized in OpenCV in an easy way. It needs the Mat of the original image to transform it and the same or other matrix where the information will be stored (see Code 3.4).

```
cvtColor(frame, frame_HSV, COLOR_BGR2HSV);
```

Listing 3.4: Conversion from RGB to HSV code

The best way to understand the process is to compare the original image with the HSV conversion (see Figure 3.4).

The next step is to detect the desired colour by creating a colour range to select [50]. The frame must be processed by several filters to reduce the noise, the isolate elements or finding holes and crones. The filters to apply are dilatation and erosion.



(a) Original image (b) Result of the HSV convertion

Figure 3.4: Comparation of the original and the HSV image

- The **Dilatation** filter consists of convoluting the image with some kernel (also known as null space). The Kernel is scanned over the image and replaces the image pixel with the maximum values, in other words, make the image bigger. The best method to understand the process is to compare the image of Figure 3.5 from the tutorial of morphological operations [49]. The mathematical equation utilized is:

$$\mathrm{dst}(x, y) = \max_{(x', y'):\ \mathrm{element}(x', y') \neq 0} \mathrm{src}(x + x', y + y') \tag{3.4}$$

Figure 3.5: Dilatation processing

Figure 3.6: Erosion processing

- The **Erosion** filter is the complementary from the dilatation function. The area of the kernel is computed with a local minimum and, as before, replaces the image pixel with that minimum value. The Erosion process is plotted at Figure 3.6 from the tutorial of morphological operations too [49]. We can deduce that the erosion filter makes the objects smaller. The mathematical equation use is:

$$\mathrm{dst}(x,y) = \min_{(x',y'):\ \mathrm{element}(x',y')\neq 0} \mathrm{src}(x+x', y+y') \tag{3.5}$$

These filters can be applied several times to the same image, and in case of several images channel, each one is executed independently.

In this Master' Thesis, the drone has two different tag: the green tag is used to identify the location in the camera and blue tag is used to check the orientation (see Figure 3.7).

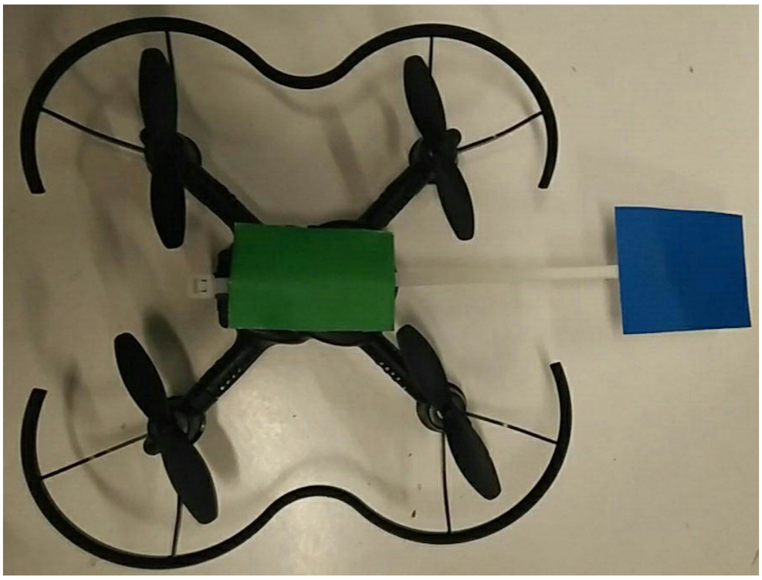


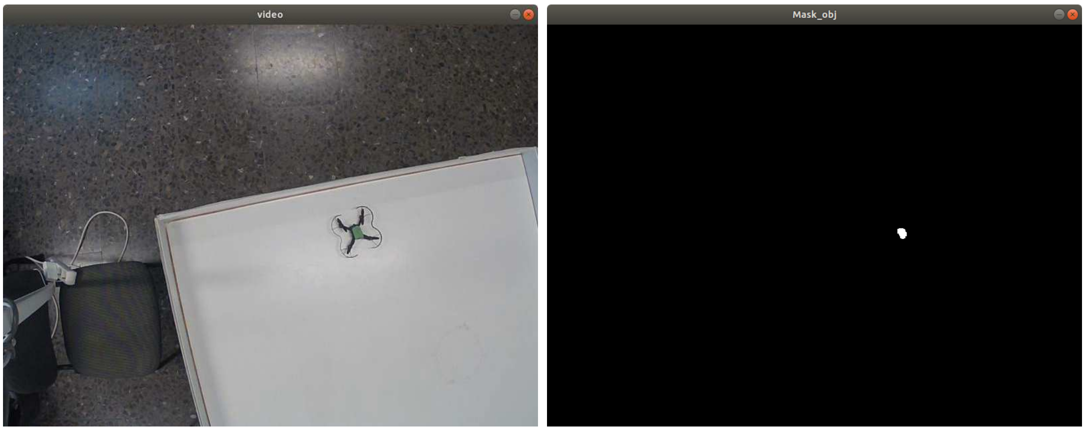
Figure 3.7: Tag of the drone

The main program detects the **Green** colour range because the nature light and other factors, the image is filter four times. First an erode method, then two separate dilatation filters and finally with other erode filter. The shape of the kernel selected is an ellipse element. The reason is the shape of green label.

Then, the same process is repeated to detect the **Blue** colour to detect the orientation of the drone (see Code 3.5).

```
inRange(frame_HSV, Scalar(LOW_H,LOW_S,LOW_V),
Scalar(HIGH_H,HIGH_S,HIGH_V), mask);
erode(mask, mask, getStructuringElement(MORPH_ELLIPSE, Size(5, 5)));
dilate(mask, mask, getStructuringElement(MORPH_ELLIPSE, Size(5, 5)));
dilate(mask, mask, getStructuringElement(MORPH_ELLIPSE, Size(5, 5)));
erode(mask, mask, getStructuringElement(MORPH_ELLIPSE, Size(5, 5)));
```

Listing 3.5: Erosion and dilatation code



(a) Original image (b) Result of the drone detection

Figure 3.8: Original image and drone detection.

Until now, the colour of the drone is detected and the image achieved has the drone location pixel, as shown in Figure 3.8.

The last main task is to track the drone. There are several options to track an element. On the one hand, it is made by finding the contours; on the other hand, by moments. The best results are achieved when both options are used.

To find the contours in a binary image, there is a function of OpenCV that can obtain a curve or line that joints all the continuous points what have the same intensity or colour. This function is useful to detect, recognise and analyse objects. The best results are obtained when the images are monochromatic colour[53].

The function needs the following parameters (see Code 3.6): the monochromatic image, a vector point to store the contours, the hierarchical topology of the image (the information of how many elements or contours has an image and are grouped by level), the mode and the method. This last options selected are, CV_RETR_TREE and CV_CHAIN_APPROX_SIMPLE. The CV_RETR_TREE recovers all the contours and rebuild the completely hierarchy of the related shapes.

The CV_CHAIN_APPROX_SIMPLE method compresses the image in a horizontal, vertical and diagonal way and gets the end points. For instance, a rectangle is compressed by its four vertexes. Although a mode and a method are selected, there are other ones which can be implemented to get better results in other applications [52].

```
findContours( cloneMask , contours , hierarchy , CV_RETR_TREE ,
CV_CHAIN_APPROX_SIMPLE , Point(0, 0) );
```

Listing 3.6: Find contours code

Once the contours are detected the next step is to draw them [48]. The OpenCV provides an optimized function, which can be used as the following Code 3.7:

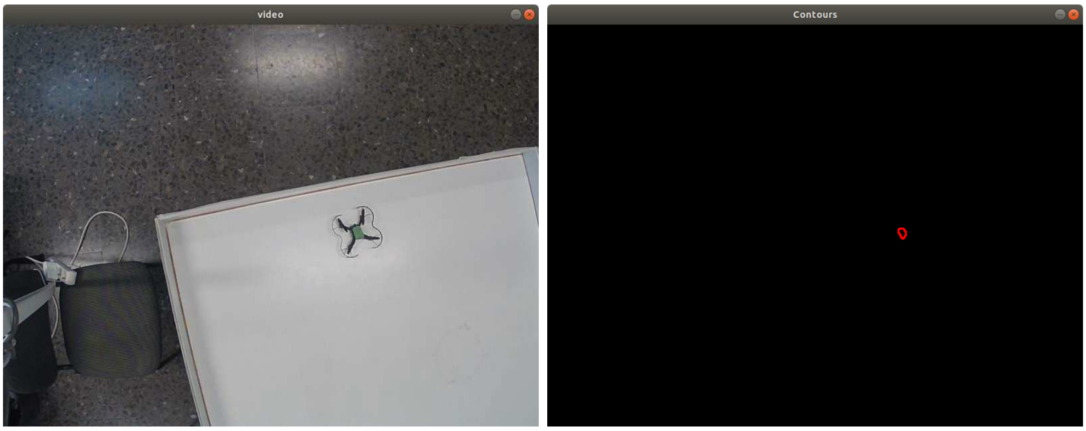
```
Mat drawing = Mat::zeros( cloneMask.size(), CV_8UC3 );

for( int i = 0; i< contours.size(); i++ ) {
  Scalar color = Scalar( (255, 0, 0), (0, 255, 0), (0, 0, 255));
  drawContours( drawing, contours, i, color, 2, 8, hierarchy,
  0, Point() );
}
```

Listing 3.7: Drawing contours code

As the findContours(), the drawContours() use similar parameters: the image, where the information will be stored, all the points from the contour detection, the integration index because the draw method is normally used inside a loop, the colour of the contours, the thickness of the lines and the line type. The optional data to draw some of the hierarchy contours and the rest of the parameters were not selected.

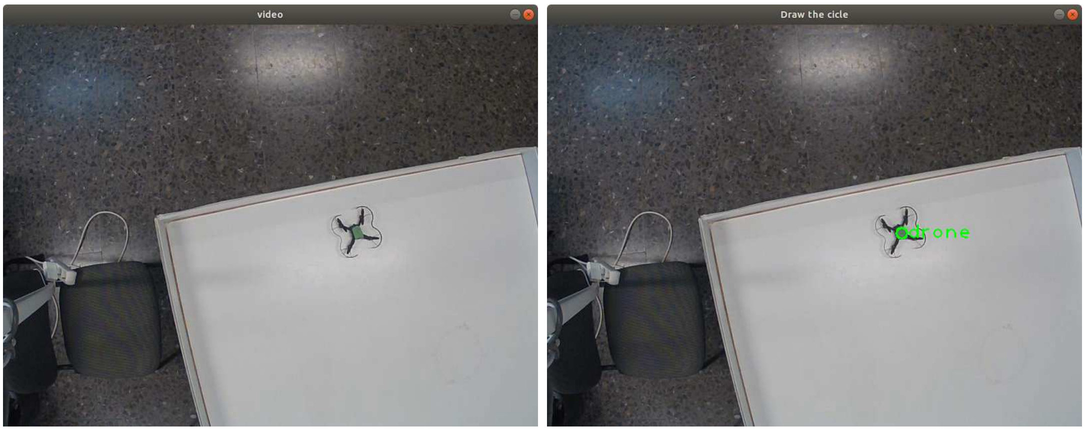
The result of applying the found contours and drawing them can be seen at Figure 3.9.



(a) Original imagen (b) Result of the contour detection

Figure 3.9: Original image and contour detection

According to the shape of the object and the real detection, the contours are not detected geometrically. This is why an enclosing form is utilized, if the drone form is suggested to a circle, the form can approximate to the minimum enclosing circle which is defined, it can be seen in Figure 3.10. This function needs to compute and find the area of each contour, and safe the location, the radius and the index.



(a) Original imagen (b) Result of the draw a circle

Figure 3.10: Original image and drawing a circle

As a result the drone is detected (see Code 3.8). The last step is to tracking the motions and obtain a more precise location from the previous placements. The tracking can be programmed by the moment's information of the main frame.

```
for(int i= 0; i < contours.size(); i++){
  //Find the area of contour
  area = contourArea( contours[i] );

  if( area > max_area )
  {
    max_area = area;
    largest_contour_index = i;
    //Store the index of largest contour
    largest_contour = contours[i] ;
    // 6. Fetch th center
    minEnclosingCircle(largest_contour, center, radius);
    //Find the bounding circle for biggest contour
  }
}
```

Listing 3.8: Drone detection code

The moments of an image are a particular average of the intensities of the pixels, or a function of their instants. Those functions are usually chosen to have some special properties or interpretation, it means that they can calculate the area, perimeter and centroid of the object (the centre of mass of the object). In addition, it can remark that the contours area or the minimum enclosing circle is based on those mathematical equations.

```
oMoments  = moments(mask);
dM01      = oMoments.m01;
dM10      = oMoments.m10;
dArea     = oMoments.m00;
```

Listing 3.9: moments code

As can be seen at Code 3.9, the moments function calculates all the moments, this function goes up to the third order (m00, m10, m01, m20, m11, m02, m30, m21, m12, m03). The results are saved in structures that define: the spatial moments, the central moments, as well as the central normalized moments.

These moments can be computed mathematically as:

1. The spatial moments m_{ji} are generated in an image as:

$$m_{ji} = \sum_{ji} (\text{array}(x, y) \cdot x^j \cdot y^i) \tag{3.6}$$

2. The central moments mu_{ji} are calculated as:

$$m_{ji} = \sum_{ji} (\text{array}(x, y) \cdot (x - \overline{x})^j \cdot (y - \overline{y})^i) \tag{3.7}$$

where $(\overline{x}, \overline{y})$ is the mass centre $\begin{cases} \overline{x} = \frac{m_{10}}{m_{00}} \\ \overline{y} = \frac{m_{01}}{m_{00}} \end{cases}$

3. There are cases where the information needs to be normalized. The normalization is needed to have certain independence from the properties of the image, such as the brightness and the contrast and allow for comparison using the quality index. The normalized central moment, nu_{ji}, is computed as:

$$nu_{ji} = \frac{mu_{ji}}{m_{00}^{1+(i+j)/2}} \tag{3.8}$$

The equations of the moments of an image or the moments of a contour are defined in a similar way. However, OpenCV library computes it by using the Green's formula [43]. That is the reason why the moments of a frame are slightly different for the same rasterized frame, due to the limited resolution.

The Green's theorem relates the calculus of a curvilinear integral along a simple closed curve C and a double integral over the plane region D, where D is oriented in counterclockwise along the boundary C as shown at Equation 3.9.

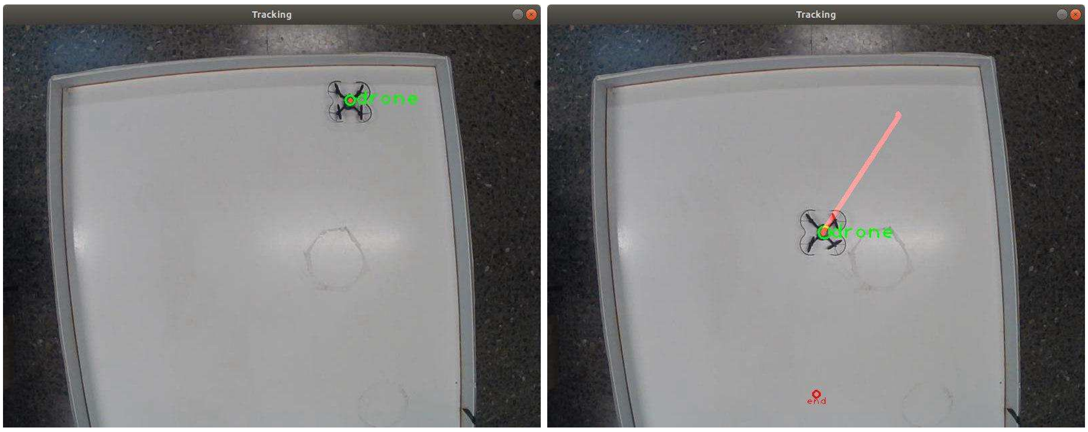
$$\oint_C (Pdx + Qdy) = \oiint_D \left(\frac{\partial Q}{\partial x} - \frac{\partial P}{\partial y} \right) dxdy \tag{3.9}$$

According to the tracking of the drone, the positions x and y of the object are computed as the centre of mass as shown in Equation 3.9. It is considered that there are not other objects in the image. That is to say that if the area is below 10000 of surface, it could not find the object, the area never is zero because of the noise samples (see Code 3.10). Once the x and y coordinates are obtained, they are compared to the previous position. These values are used to calculate the control base on the motor schema behaviour, which is explained in Chapter 4. The following part of the code represents how it is programmed and Figure 3.11 identifies the tracking of the image.

```
// if the area <= 10000, I consider that the there are no object in
// the image and it's because of the noise, the area is not zero

if (dArea > 10000) {
  posX = dM10 / dArea;
  posY = dM01 / dArea;
}
```

Listing 3.10: Tracking code



(a) Original of the drone tracking (b) Result of the drone tracking

Figure 3.11: Tracking of the drone

3.2 Serial port communications

The main communications are designed to connect the PC to the board which will transmit the data of the roll, the pitch, the yaw and the throttle values to the drone. The communication is made by the UART protocol. This connection is used at 2 points of the ATmega board: first one from the PC to the Arduino using the Serial1 and the other from the Arduino board to the Bluetooth module, therefore the schema from the connection is represented at Figure 3.12.

Before going into the software description, a sort introduction of the UART protocol will be done [11, 26].

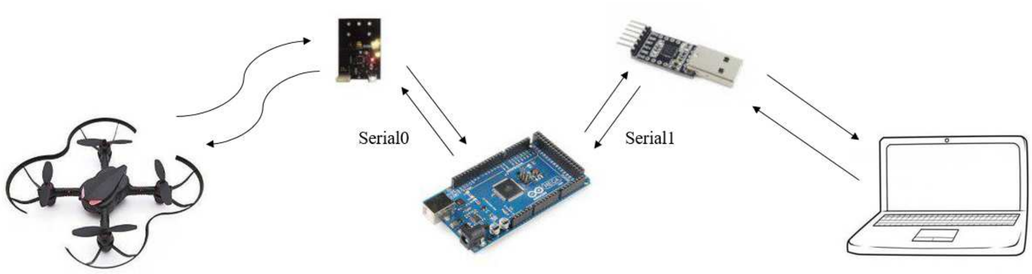
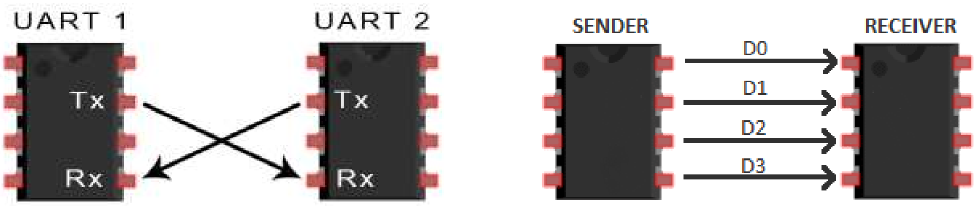


Figure 3.12: Main communications

The UART has the full name of Universal Asynchronous Receiver/Transmitter. Its main function is the serial data communication. There are two different ways to make a communication between two devices: serial communication or parallel data communication.



(a) Serial data communication (b) Parallel data communication

Figure 3.13: Type of data communication

In serial data communications, the data can be transferred through a single line bit by bit. This case only requires two cables or path, what reduces the price compared with the parallel communication. Thus, this communication is very spread and useful in most of devices and systems nowadays.

In parallel data communications, the data are transferred through multiple lines at once; it is to say, by using N lines one by each bit, what increases the price.

However the links are realised very fast. The best examples of this communications are the old printers, RAM or PCI.

Both communications can be compared at Figure 3.13. As it can be seen, the UART block consists of two components the transmitter and the receiver. The transmitter section includes three blocks: the hold register where the transmitter comprises the data-bytes to be transmitter, the shift register where the bits are moved to the left or right until the data is transmitted, and logic control which is used to establish when to write or read. Similarly, the receiver part has the identical blocks, the receive hold register, the shift and the logical control. Both blocks are generally generated at the same speed, using in the middle a baud rate generator. The baud rate has a range from 110 bps to 2304000 bps. Typically, the baud rates of the microcontrollers are 9600 or 115200 bps. This explanation can be understood by the block diagram as shown in Figure 3.14.

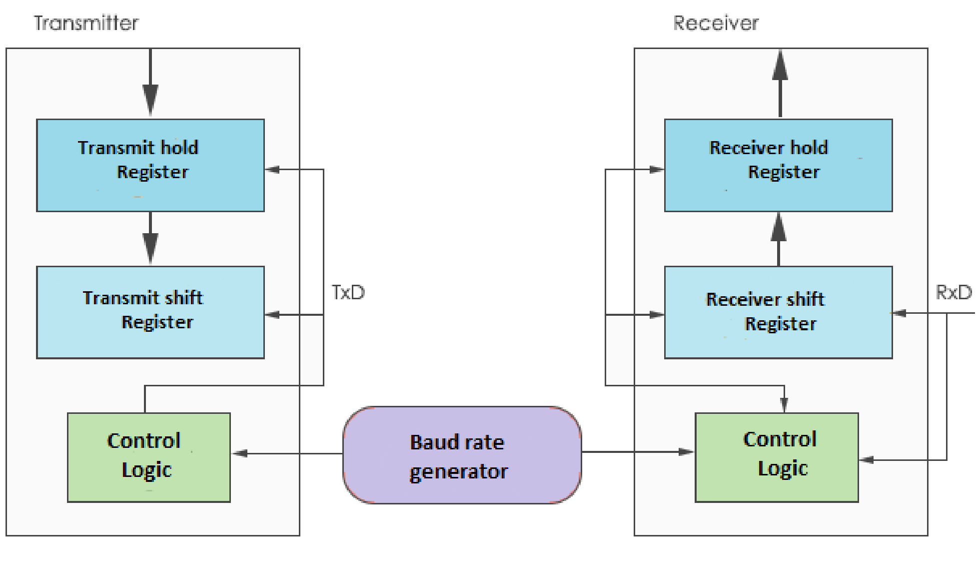


Figure 3.14: Block diagram

The information flows from the Tx pin of the transmitting UART to the the Rx pin of the receiving UART and it is done asynchronously, there is not clock signal for synchronizing the devices. Instead of the clock a start and a stop bit are included at the beginning or the end respectively. For instance, a UART can transmit the data from a data bus to another device like a CPU, memory or microcontroller. The data from the bus is transmitted in parallel form. The UART transforms them in serial and adds the start, the stop and sometimes a parity bit before the stop bit. Then the receiving UART reads the data packet bit by bit at the Rx pin. After received all bits, the UART converts them into parallel form and removes the bits which are not necessary now, it is say the start, the stop and the parity bits, and the packets are transmitted by other bus following the path.

Therefore, the UART packets are composed by 1 start bit, 5 to 9 data bits, 1 optional parity bit, and 1 or 2 stop bits (see Figure 3.15).

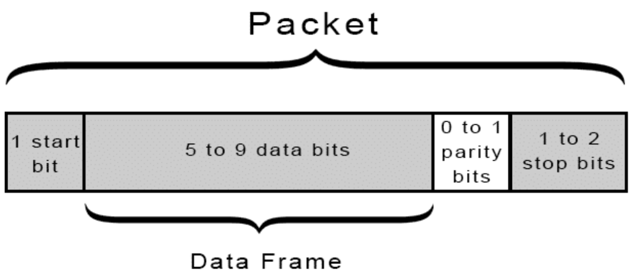


Figure 3.15: Packets diagram from [11]

The **Start bit** is used to notify to the receiver that the transmission will start, the UART lines normally are held at a high level by the pull up resistance when it is not transmitting data. Then, when transmission begins the line is put from high to low during a clock cycle, after it, the UART begins to read the data bit at the baud rate frequency.

The **data frame** contains the information to be transmitted, it can be 5 or 8 bits when the parity bit is used, on the other hand, the data frame can be 9 bits of length.

The **parity bit** provides the information if a change has been occurred during the transmission. It describes the even or odd numbers, where the parity bit is 0 or 1 respectively.

The **stop bit** is used to identify the end of the transmission. The line change from low to high voltage for at least two bit durations.

In addition, some advantages an disadvantages of UARTs can remark:

As **pros**: it only needs two wires for data. It is not necessary a clock signal. It includes a parity bit allowing to check the errors, the structure of the data packet can be changed if both sides are configured for it.

As **cons**: the size of the data frame is limited to a maximum of 9 bits, the baud rate must be within a 10% each other in both sides and it does no support several slave or master systems. Moreover, a direct cable is necessary.

Regarding the system program, 2 different types of structure frames have been created.

The **main frame** to transmit is composed by 8 signed char characters which are stored in sequence by the program. The frame has the information of the roll, the pitch, the yaw, the throttle (these four previous values are the directions to control and move the drone), the control characters (one at the beginning, other at the end, the crc data before the stop one and the type of frame, after the start one). The start and stop characters are selected from values that are not typical used at the ASCII table as the reference from [16, 8]. The START character is the **0x05** the ENQ (knows as enquiry), meanwhile the END char is the **0x07** the BEL (also call bell). The type character is fixed a zero, in ASCII has the **0x30** value because is the principal frame needed to control the Drone. This frame is shown in Figure 3.16:

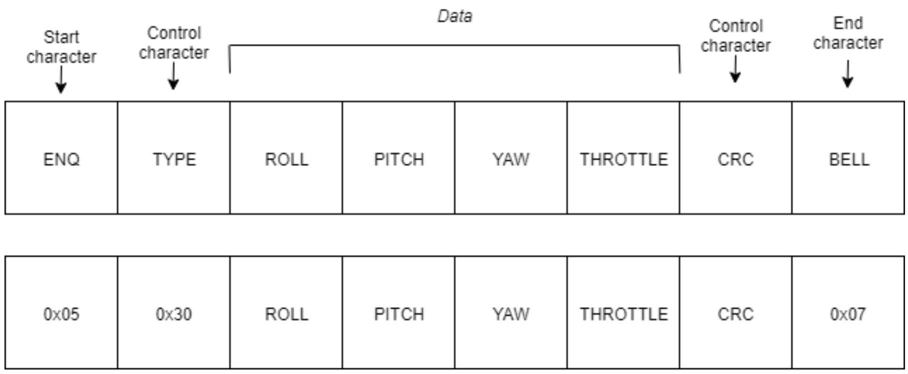


Figure 3.16: Main frame of the characters for the serial communication

The second frame is the **debug frame**. It is composed of 3 characters more than the main frame. The main idea of this frame is to transmit the data from the Drone to the PC and check if the data is correct or the communication fails. Therefore, the total length of the frame is 11 symbols. These three new values correspond to the battery, the altitude of the drone flying and the possible errors which have been established. The errors can be identified due to a negative number. The next list collects the possible errors:

If the error occurs at the **Tx side**, the number will be only a possible option:

- TX_ERROR -1

If the error occurs at the **Rx side**, the number could be:

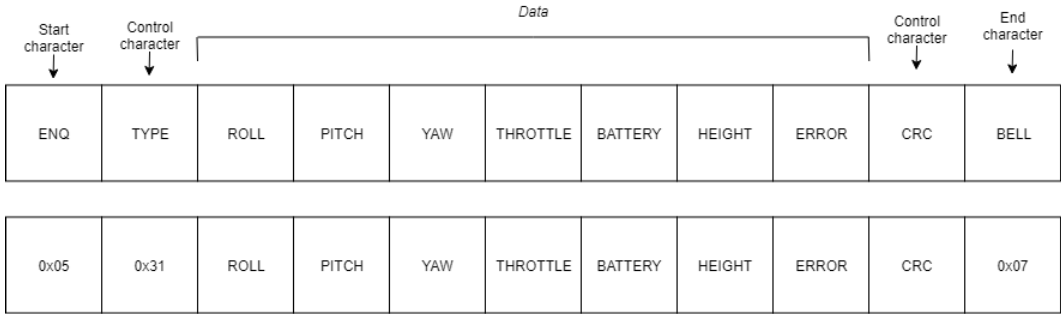


Figure 3.17: Debug frame of the characters for the serial communication

- RX_ERROR -2
- NO_DATA -4
- WRONG_ETX -5
- WRONG_NUMFRAME -7
- WRONG_CRC -6
- WRONG_TYPE -3

This frame has the type **0x31** and its schematic is shown in Figure 3.17.

The **CRC** is a control character that is a simple function to control the transition information. The CRC is calculated by adding the data and the type frame. In other words, the ENQ, the CRC and BELL characters are removed. As a result, if both values, the received and the computed match, the frame is accepted, otherwise the frame is discarded.

Although no other frames have been created, the code is prepared to incorporate new packets, adjusting the structure frame to the desired requirements.

Once the distinct frame have been detailed, the upcoming step is to define how the communication is made. It can identify four different states: first where the init port is prepared, then the transmission and the reception and the last stage the send. This state machine is shown in Figure 3.18.

The **init state** manages to prepare the different options to configure the link. It gets the port options and modifies it to set the baud rate, enable the local communication, allow the receiving and reading character, and configure the number of bits. It selects a CS8 communication, which has 8 bits, no parity and one stop bit. Once the configuration is established, the queues from the buffers are cleaned. After it, the port can be considered open.

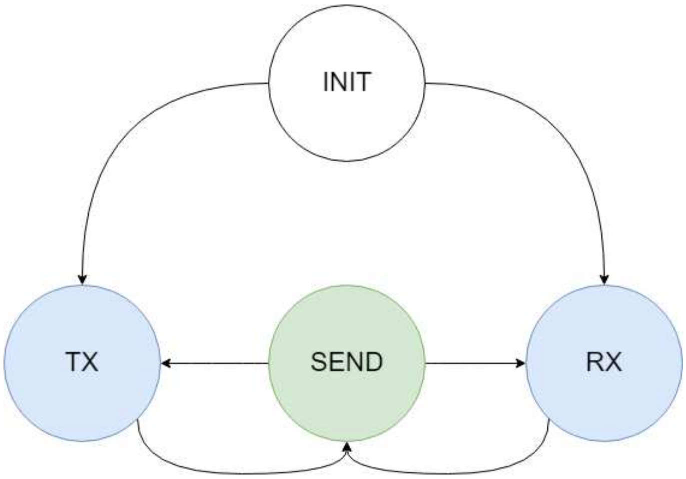


Figure 3.18: State diagram

The **TX state** creates the packet, takes the distinct measures from the control and prepares the frame depending on the type. Once the packet is prepared, the frame is printed at a log file to store and compare the information with the computed program and the motions. Whether the transmission is corrected the function returns 1, and that means that there are data to be transmitted.

The **SEND state** gets the data from the packet generated and sends it for the correct line. It waits to determine if the communication fails and returns the error.

The **RX state** receives the data from the line and decodes it. It waits until the ENQ character is received. If the frame has not got the start char, the frame is discarded and it sends the error NO_DATA. But when the header is correct, the next character is read and the type frame is identified. After it, the rest of the body at its corresponding parameters are stored. When all the data are decoded, the program notes if the end header is correct, and validated the complete frame, if not, the frame is deleted. Then the CRC is verified and compared to the received value. This behaviour is summarized in Figure 3.19.

The system transmits the frames every two seconds. The main reason is that the minimum clock cycle that the CoDrone can move toward the direction is a second, regarding to a stable flight and control the inertia from the rotors. But because of the motion of the drone and the uncontrolled throttle of the drone from the camera, the height is fixed to 350 mm from the surface. In other words, the camera can control 3 degrees of freedom in 2D (x, y and orientation), the height (z) tries to be a fixed value. The link is created from the PC to the **Serial1** of Arduino board. The Arduino board takes the roll, pitch, yaw and throttle values and converted to the Bluetooth structure using the CoDrone library. The Arduino board is connected by the **Serial0** to the Bluetooth module. The BLE protocol is implemented and documented on its manual [55]. There are several commands and structures implemented in this library. Moreover, there are other ones that are under development.

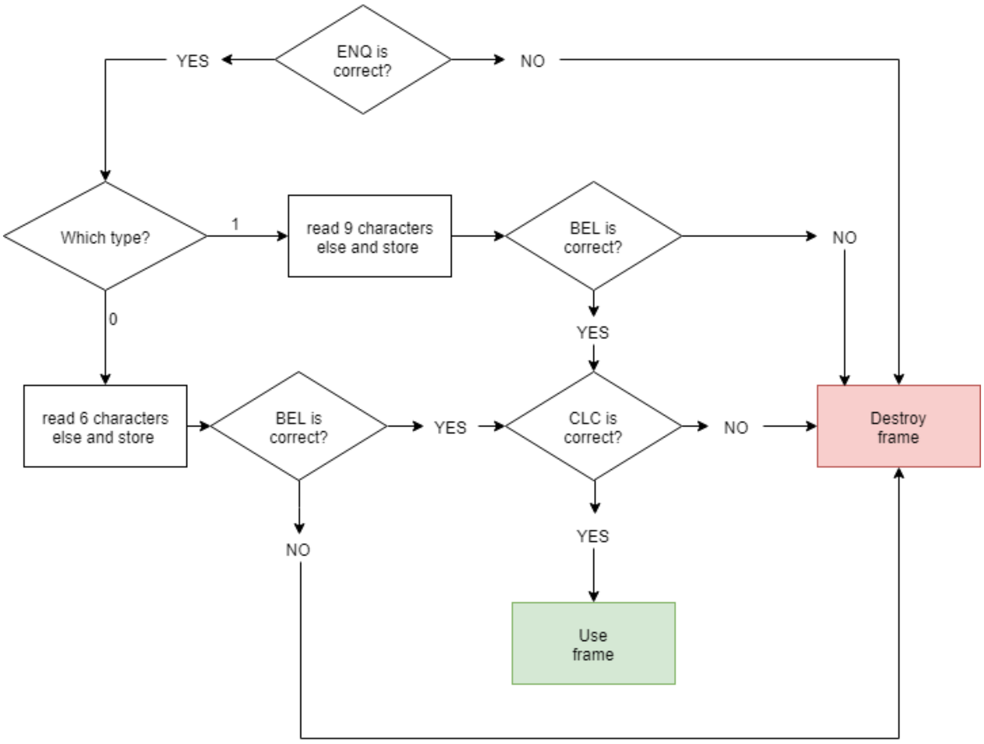


Figure 3.19: Rx diagram

The control structure has been selected from its library. It has four parameters to control the CoDrone motion. These parameters have the same type S8, which is referred to int8_t, their name are roll, pitch, yaw, throttle. Each value has the range from -100 to 100 and they represented the different directions of movements (see Figure 3.20).

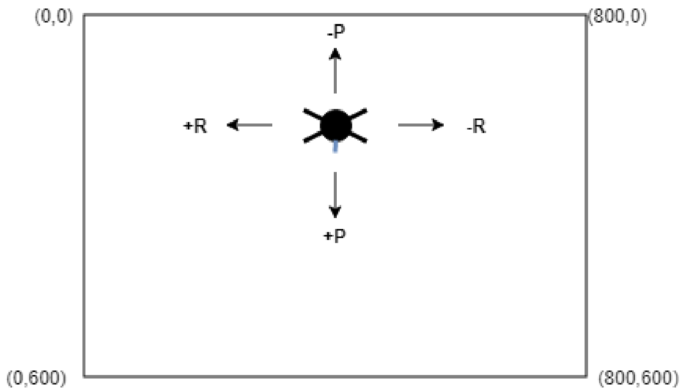


Figure 3.20: Reference plane

The **roll parameter** is used to move from left to right (x axis). It can compare to tilting the head towards one of the shoulders. The negative value means the drone will go to the left, and the positive will go to the right. For instance, if the drone needs to go at maximum speed to the left, the roll value is -100, and if it should go to the right, the roll value will be 100.

The **pitch parameter** controls the front and rear directions (y axis). It can compare with the head, is like when the head tilts to see up or down. In a similar way that the roll, negative values will be used to move to the rear, meanwhile the positive ones are used to the front. For example, if the drone wants to advance forwards in a straight line the value must be > 0.

The **yaw parameter** manages the heading and turns. These turns can be made at counterclockwise or clockwise and it can compare to shake the head. As aforementioned, negative numbers rotate to the left and positive numbers rotate to the clockwise.

The last parameter, the **throttle** guides the drone up and down. The up action will be made by positive values, and negative will decrease the height of the drone.

These parameters can be plotted to a map to clarify the directions (see Figure 3.21).

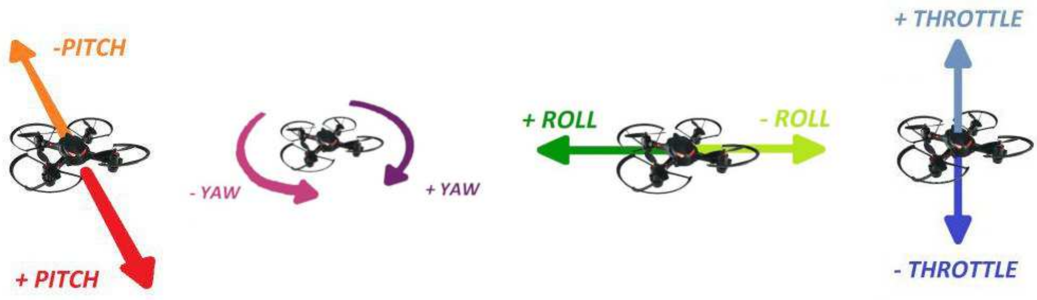


Figure 3.21: Motions of the drone

To summarize, Table 3.1 shows the connections between the different electrical boards.

Serial	CP2102 module	ARDUINO ATMEGA board	BLE module
Serial1	TX	RX1	-
	RX	TX1	-
Serial0	-	TX0	TX0
	-	RX0	RX0

Table 3.1: Serial connections

Chapter 4

The controller

A briefly introduction about the motor schematic behaviour and the motion controller of the drone will be exposed in this Section. Before going into the matter, a short overview of how the drones fly is presented.

4.1 The flight control of a quadcopter

As was introduced at the end of the Section 3.2, there are four main quadcopter controls:

- Roll
- Pitch
- Yaw
- Throttle

These directions are related to the remote control as shown in Figure 4.1. The remote control is configured to have the acceleration at the left and the direction and forward to the right [6, 28, 2].

The quadcopter name comes from quadcopter Helicopter, it is said that is very similar to an helicopter. The quadcopter uses four helix and four motors. These elements create flow currents and thrust to elevate the drone. There are two motors which rotate in clockwise and other two counterclockwise. The location of this rotors are confronted the motors which turn to the same side, separated by the other rotors (see Figure 4.2).

The **roll** and **pitch axes** are controlled by making the motors on one side spinning faster than the other side motor. It is to say that one side of the rotor will have more power than the other side, causing inclinations. For instance, to make the drone fly

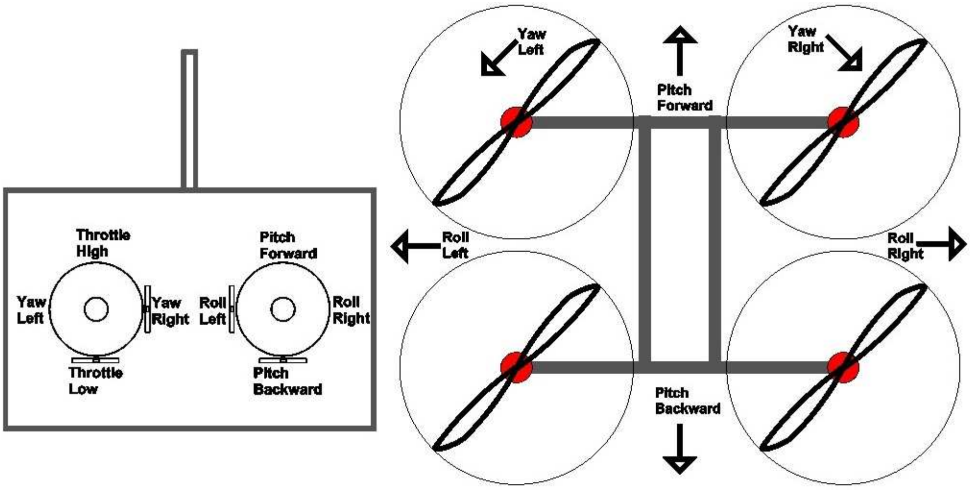


Figure 4.1: Control remote and directions from [6]

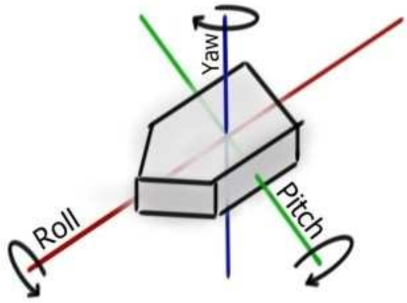


Figure 4.2: Orientation of the quadcopter from [2]

to the right, it is mandatory that the two motors on the left side spins faster than the right side. To go to the front, the flight controller needs to spin faster the back rotors than the two motors on the front (see Figure 4.3).

The yaw control is a little bit more complex than the previous directions. As aforementioned, the configuration of the motors are in the opposite direction than its neighbours (see Figure 4.4). When the motors rotate in alternative directions, this configuration can neutralize or cancel each motor direction making them to turn.

Figure 4.5 demonstrates how each motor has been configured in the opposite direction. When a helix spins, for instance at clockwise, the conservation of angular moments tends to spin counterclockwise, in other words the body or frame of the drone will rotate at the opposite direction. The reason is the Newton's third law of motion, **"for every action, there is an equal and opposite reaction"**.

That action may be a little confusing.

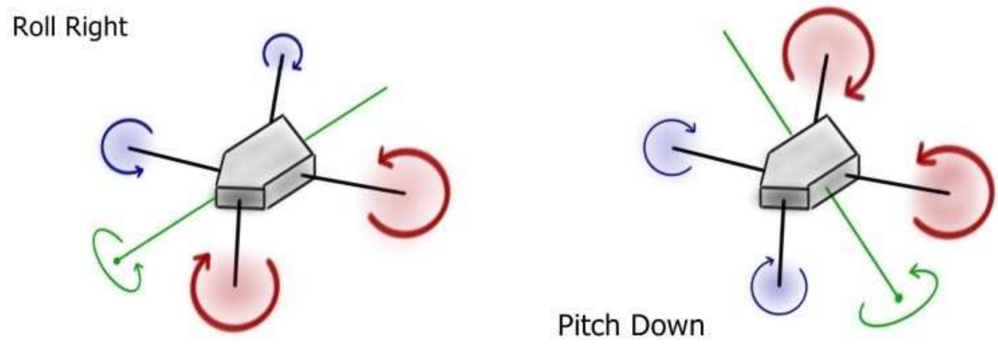


Figure 4.3: Example of Roll and pitch inclinations from [2]

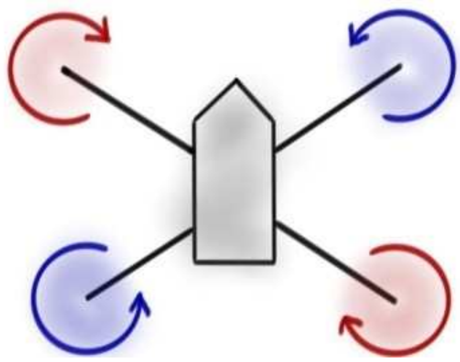


Figure 4.4: Example of yaw inclinations from [2]

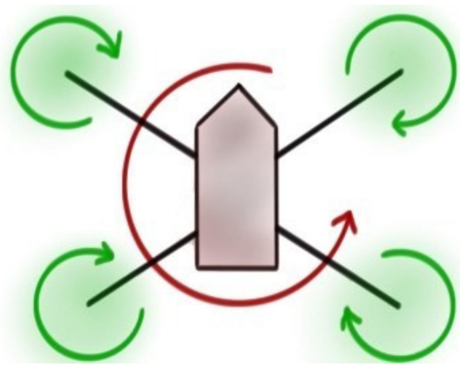


Figure 4.5: Example body spin from [2]

Therefore, using pairs of rotors rotation in opposite direction, the effect about rotate at the yaw axis, will be cancelled. To rotate about the yaw axis, the controller should be slower in opposite pair of motor to the yaw direction. It is to say, the angular moment of the two pair of helix is not balanced and the drone rotates. As a result, it can spin in both directions by slowdown the different pair of motors. For instance, if the drone has to turn at the counterclockwise axis, the clockwise motor should be faster than the others (see Figure 4.6). And if the clockwise yaw is desired, the counterclockwise should be lower than the clockwise.

The **throttle** controls the altitude or hovering from the aircraft. The aircraft will hover; if it stays at a constant altitude without spinning in any location. In this case the rotors should balance the forces. The flight controller needs to counteract the force of gravity with the elevation obtained from the motors. Mathematically, the force of the gravitation on the motors is equal to the mass of the rotors times gravitational acceleration (which is a constant while the drone is on the Earth). The raise caused by the rotors is equals to the sum of the elevation generated by each of

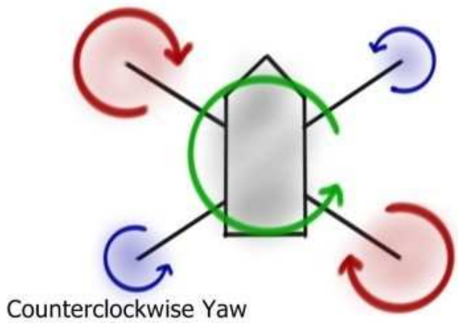


Figure 4.6: Counterclockwise spin from [2]

its motors. Consequently, whether the force of gravity is equals to the force of the lift generate by the rotors, the drone will maintain a constant altitude (see Equation 4.1).

$$\vec{g} \cdot m_{rotors} = \sum_{r \in rotors} \vec{F}_r \tag{4.1}$$

Thus, the ascent and descent is achieved by disturbing that balance. When the elevation produces by the rotors are greater than the gravitational force, the aircraft will gain height. In the other case, when elevation is lower than the force of gravity, the drone will decrease.

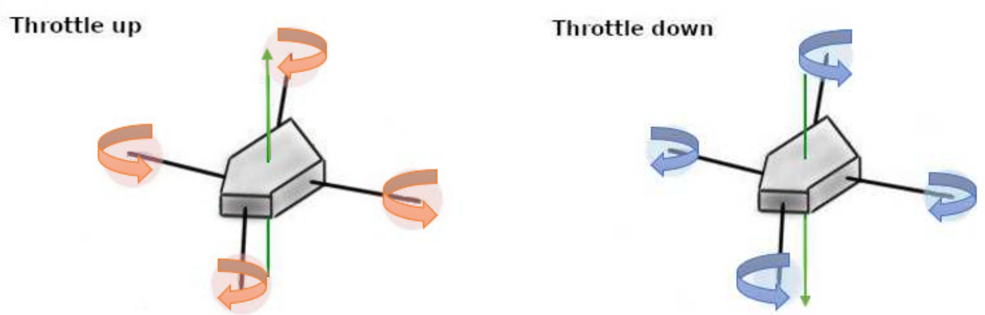


Figure 4.7: Example of gravitational forces

One of the problems is when the drone moves towards any direction, the drone needs to incline, this action creates a loss of height that needs to be compensated by the controller.

4.2 Behaviour-Based Architectures

Behaviour-Based architectures are a methodology for developing and designing autonomous control systems. The concept was introduced in the mid 1980s [7, 76]. Behaviour-Based architectures were defined as (Ronald C. Arkin [7], pag 125): "Robotic architecture is the discipline devoted to the design of highly specific and individual robots from a collection of common software building blocks". Robotic architectures are relative similar each one in the robotic control context. However, it usually refers to the software, rather than the hardware layout of the system.

There are two main reactive robotic architectures: the subsumption and motor schemas. This Master's Thesis has focused on the last one, the Motor schemas behaviour.

4.2.1 Motor schemas

The target of this architecture is the coordination of behaviours inspired by the biological sciences. The motor schemas details the control based on several activities. This model is a contribution from different sensors, it connects the action and perception using a programming language. This model is employed for explaining the brain's functions as Artificial Intelligence applications.

The motor schemas model differs from other behaviour systems in several ways (Arkin [7], page 142):

- The behaviour responses is represented as vectors.
- The coordination is made by the vector addition.
- This method is configured to run in a real-time base on the robot interactions and the environment. It means that there are not predefined any coordination.
- The response of the robot varies by the contribution of the behaviour. It means that not mediation is needed.
- The unpredictability can be reflected by the addition of a noise signal as the rest of vector contributions.

The motor schemas is fundamental used at navigation tasks and it is similar to the animal behaviour (Arkin [7], Chapter 2). The environment information is recorded by the sensors, that information is applied at a specific behaviour to react. Each motor schema has an output action vector that specifies the direction that the robot should move in reaction to the previous stimuli (see Figure 4.8).

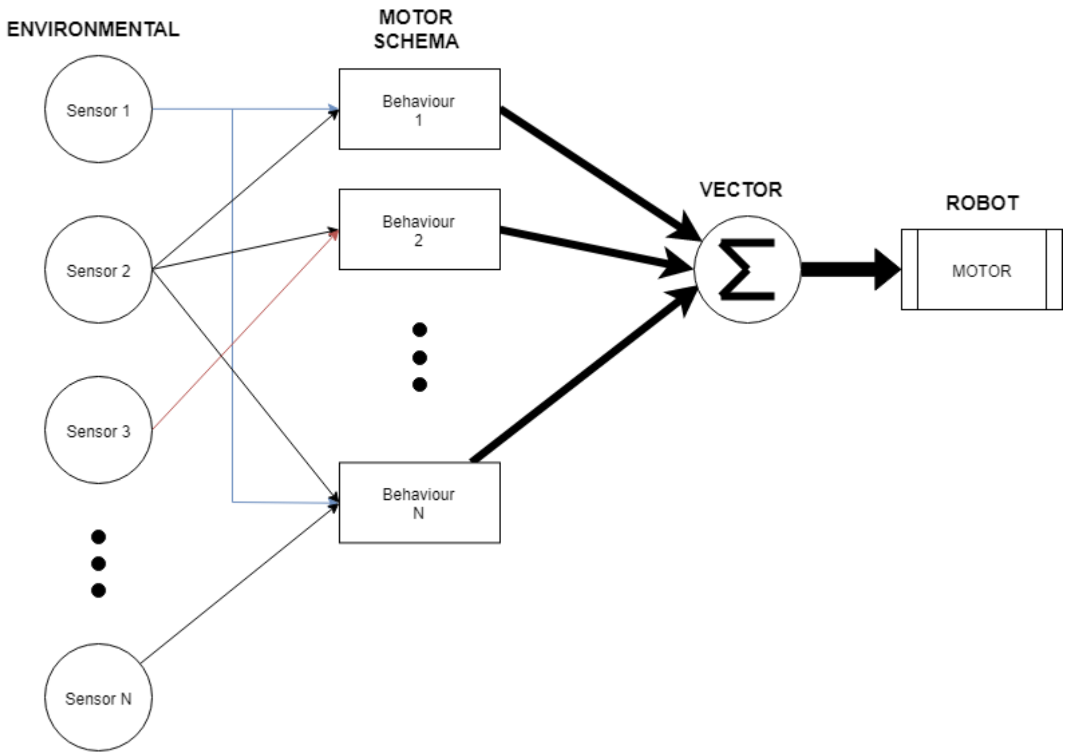
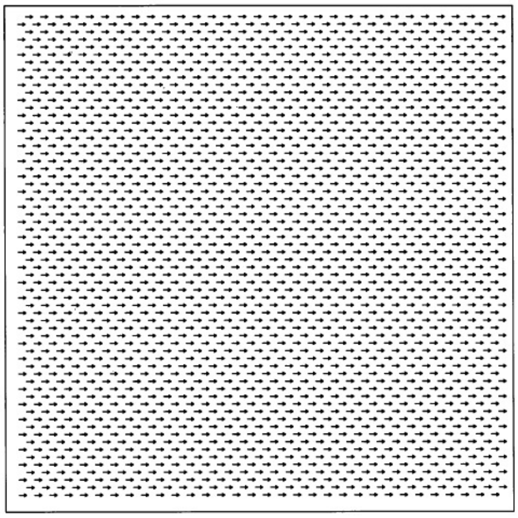
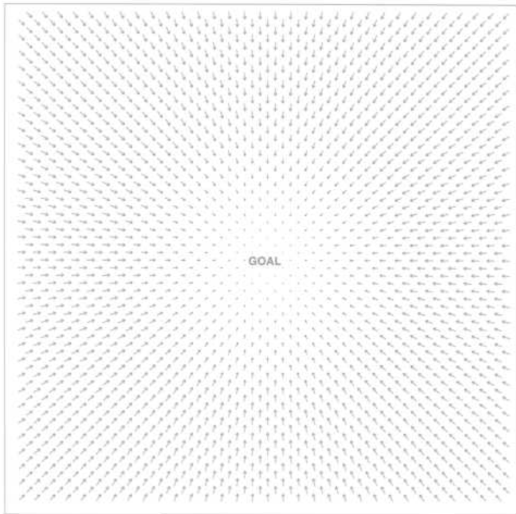


Figure 4.8: Motor schematic diagram

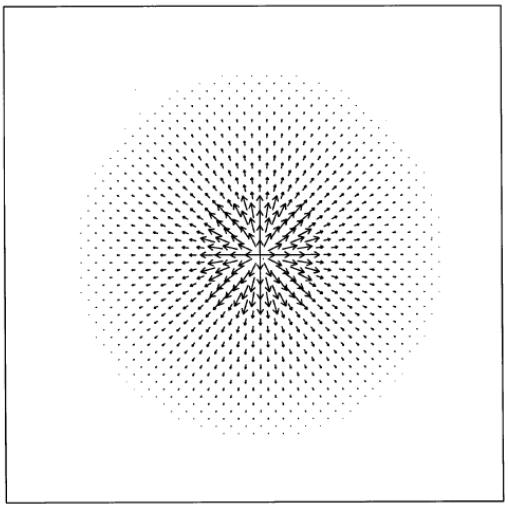
The best way to understand the contribution of the different vectors is by a simple example of two dimensions, Figures 4.9a to 4.9f are from the Arkin [7], and show how a real implementation can be done. The blocks selected are: move ahead, move to the goal, avoid statics obstacles, stay on path and noise.



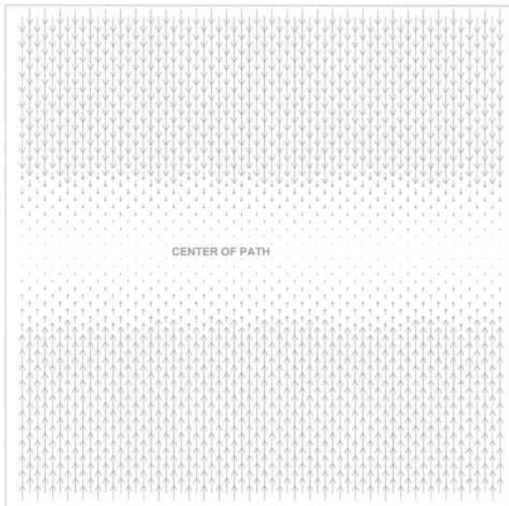
(a) Move ahead vector



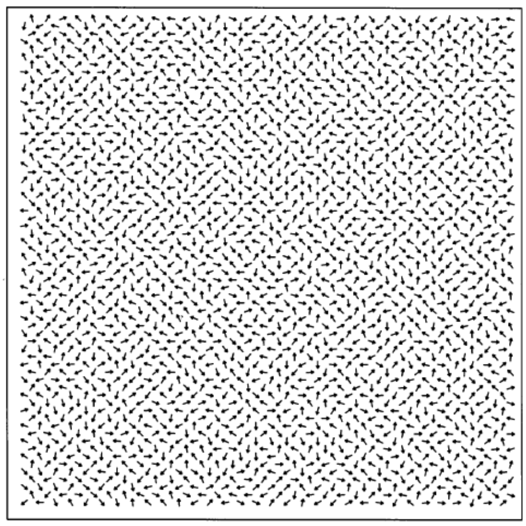
(b) Move to goal vector



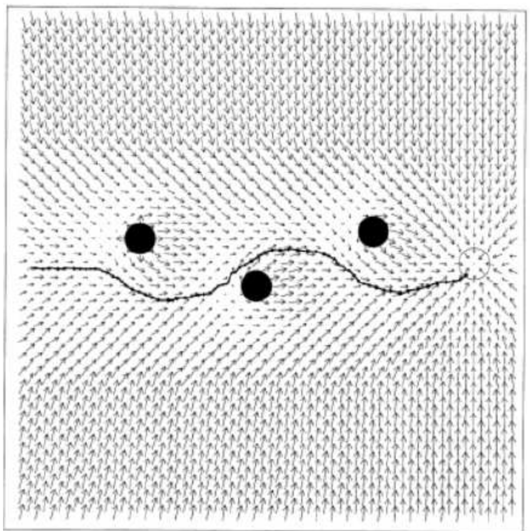
(c) Avoid static obstacles vector



(d) Stay on path vector



(e) Noise vector



(f) Result direction

Figure 4.9: The contribution of the different vectors

As it can see the separated vectors are added to generate the final direction around the object.

The design process of a schematic model system usually has the following states:

1. Characterize and analyze the task required in terms of the motor behaviour.
2. Disintegrate and decompose the task into the simplest level, by applying biological studies.
3. Generate the functions and equations to model the robots response to the perceived environment.
4. Manage simple simulations and evaluate the behaviour in the environment, by remaining the approximation of the real word.
5. Decide the requirements needed to cover the inputs for each motor schema.
6. Develop the particular perception algorithms that obtain the information for each behaviour.
7. Incorporate the control system to the robot.
8. Test and validate the system efficiency.
9. Reiterate and extend the behaviour.

With this in mind, the flight control is modelled and implemented in the system. The main task that the drone has to carry out is to arrive to the target. If there is any obstacle, it must avoid it to follow the route. Due to fact that the control of the drone involves the craft inside the camera, it must force the airplane to be in the limit camera as well as to stay at the centre of the path. As a result, four behaviours are identified: move to the goal, avoid static obstacles, move to the centre of the camera and stay on the path (see Figure 4.10).

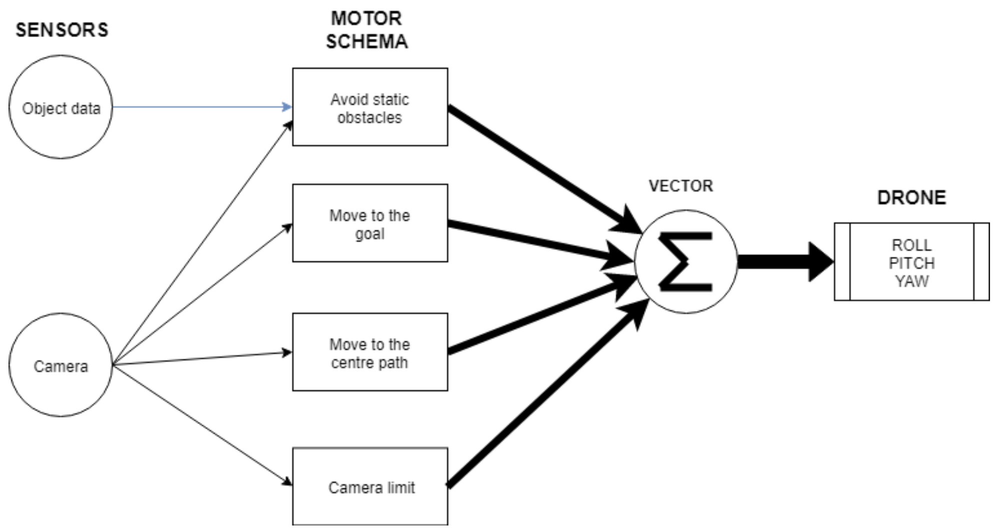


Figure 4.10: Motor schema of the system

The four behaviours can be discomposed and formulated in four vector contribution:

1. **Avoid static obstacles.** The drone position will be denoted as the coordinates (posX, posY) and object location is named as (posObjX, posObjY).

   This behaviour is formulated using three parameters, the distance from the object to the centre of obstacle, (d), the radius of circumscribed circumference of the object, (R), due to the width and lengthy. Those parameters are known and the sphere of influence to avoid the obstacle, (S); (see Figure 4.11).

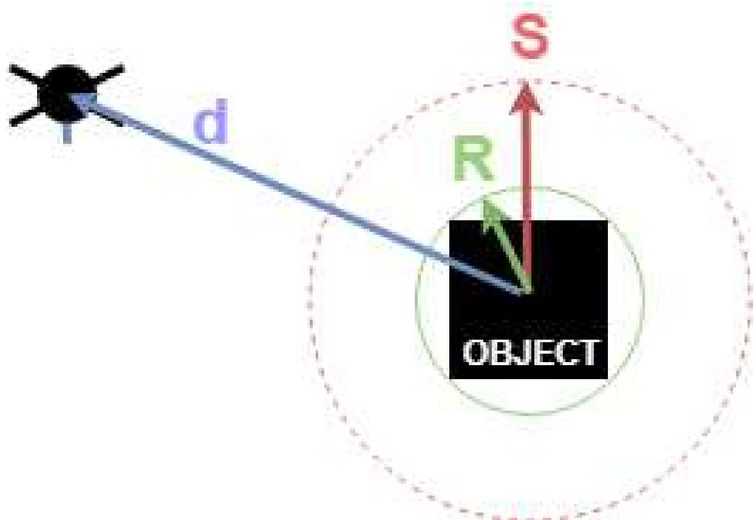


Figure 4.11: Avoid static object notation.

Mathematically:

$$S = \sqrt{\left(\frac{W}{2} + M\right)^2 + \left(\frac{H}{2} + M\right)^2} \tag{4.2}$$

$$R = \sqrt{\left(\frac{W}{2}\right)^2 + \left(\frac{H}{2}\right)^2} \tag{4.3}$$

$$d = \sqrt{(posObjX - posX)^2 + (posObjY - posY)^2} \tag{4.4}$$

where W is the width of the object, H is the length of the object and M is the margin of influence which is fixed by the user. With these values, the program generates the contribution vector to avoid the obstacle. Moreover, it can be calculated as three possible locations.

When the drone is outside the influence sphere, the drone will be out of the attracting field of the object and the drone can follow a straight path.

Other case is when the drone is next to an object, inside the influence sphere, a proportional force respect the distance to the object should oblige the drone to fly at the opposite direction. In other words, when the drone is very close to the object the flight speed will be higher than when the craft just entered to the influence area, and if the object is detected by the left, the drone will fly to the right and vice versa.

The last case is when the drone crash to the object, as a result, the drone has to fly an infinite speed from the object, as the real world has limits the flight controller will be adjusted to the maximum possible controller speed. In the previous case the direction must be the opposite too (see the equation 4.5).

$$(posX, posY) = \begin{cases} 0 & if \quad d > S \\ \dfrac{S-d}{S-R} \cdot G & if \quad R < d \leq S \\ \text{VMAX} & if \quad d \leq R \end{cases} \tag{4.5}$$

where G is the gain to avoid the obstacle and VMAX is the maximum speed that the drone can support to fly away from the obstacle. Both parameters are adjusted by the user.

Finally, the direction vector is calculated. When the obstacle is detected by the left, the flight controller should fly to the right and when it is recognized at the front, the drone will back from the path. Similar considerations must be done to the rest of locations (see Equations 4.6 and 4.7)

$$\begin{cases} +(posObjX - posX) & < & 0 \\ -(posObjX - posX) & \geq & 0 \end{cases} \tag{4.6}$$

$$\begin{cases} +(posObjY - posY) & < & 0 \\ -(posObjY - posY) & \geq & 0 \end{cases} \tag{4.7}$$

The previous equations are applied to each object and the avoid behaviour is computed by all the contribution added at each coordinate x and y from every obstacle detected. The values are supposed to be stored at the vector: (xAvoidValue, yAvoidValue)

2. **Move to the goal:** The behaviour is modulated by a fix action value and the direction toward the goal. So, when the end location is at the right front, the drone will fly at a constant speed to the goal at the right direction following the straightway. The same applies to the rest of combinations (see Equation 4.8 and 4.9).

$$\begin{cases} \text{+VMAG} & if & (posEndX - posX) < 0 \\ 0 & if & (posEndX - posX) = 0 \\ \text{-VMAG} & if & (posEndX - posX) > 0 \end{cases} \tag{4.8}$$

$$\begin{cases} \text{-VMAG} & if & (posEndY - posY) < 0 \\ 0 & if & (posEndY - posY) = 0 \\ \text{+VMAG} & if & (posEndY - posY) > 0 \end{cases} \tag{4.9}$$

where VMAG is the fix step towards the goal. As before the values are supposed to be stored at the vector (xGoal, yGoal).

3. **Move to the centre path:** This behaviour is used to centre the drone at the path. The centre path is considered the vertical centre of the image. It is determined by the distance of the drone to the image centre, where the centre position is named as (posCenterX,posCenterY)=(posCenterX,0). It can be noticed that this behaviour has only a contribution for the x coordinate. It means, that the behaviour will not focus on the goal direction. Instead, it will focus on being at the road path. The Equation 4.10 estimates the distance mathematically:

$$d = \sqrt{(posX - posCenterX)^2 + (posY - posCenterY)^2} \tag{4.10}$$

Once the distance is obtained, the contribution weights to model the actions, are programmed. The method has two different possibilities: that the drone is far away from the path or it is on the road. When the drone flies outside the road, a static speed is selected to force the drone to go to the route. When the centre of the road is close, the speed decreases in a proportional way to the distance. The parameter can be express as Equation 4.11.

$$\begin{cases} \text{PPATH} & if \quad d > \left(\frac{\text{PPATH}}{2}\right) \\ \frac{d}{\frac{WPATH}{2}} \cdot \text{GAIN} & if \quad d \leq \left(\frac{\text{PPATH}}{2}\right) \end{cases} \tag{4.11}$$

where PPATH is the weight of motion, WPATH is the width of the whole path and the GAIN is the gain to reduce the contribution when the drone approaches to the target. These parameters are fixed by the user.

As the rest of behaviours the last state is to calculate the direction vector as:

$$\begin{cases} + & if \quad (posCenterX - posX) < 0 \\ - & if \quad (posCenterX - posX) \geq 0 \end{cases} \tag{4.12}$$

Assuming the value is saved at (xCenter).

4. **Camera limit:** the last weight contribution is that the drone should stay on the limit of the camera, otherwise the control is very complex. As it was explain at Section 3.1, the maximum size of the camera is configured previously to open the program and the variables are defined as constant values. Equation 4.13 identifies 3 different regions of action in each coordinate. The first region referring to the x is: when the drone leaves the camera by the left side, the second is when the drone leaves the camera for the right and the last region is when the drone maintain inside the camera. These regions are similar to the y coordinate, but referring to the up or the down side. Therefore, when the drone tries to leaves the camera the flight controller forces the fly to the opposite side. If the drone left the camera by the lefts side, the controller will force to flight at the right direction. The speed will be assumed as a constant step. Thus, the formulate expression is estimated as the following:

$$\begin{cases} -\text{VCENTER} & if \quad (posX) \leq \text{MARGIN} \\ +\text{VCENTER} & if \quad (posX) \geq (\text{image}_{centerX} \cdot 2 - \text{MARGIN}) \\ 0 & \text{rest} \end{cases} \tag{4.13}$$

$$\begin{cases} -\text{VCENTER} & if \quad (posY) \leq \text{MARGIN} \\ +\text{VCENTER} & if \quad (posY) \geq (\text{imagen}_{centerY} \cdot 2 - \text{MARGIN}) \\ 0 & \text{rest} \end{cases} \tag{4.14}$$

where VCENTER is a fixed speed to arrive to the camera and MARGIN is the square created to force the drone to be inside. The vector direction will be stored at the (xCenterCamera, yCenterCamera).

The four behaviour vectors are added, each weigh of x at its x coordinate and the same is applied to y. As a result, the motor vector is generated and stored at the (xValue, yValue) (see Equation 4.15 and 4.16).

$$xValue = xAvoidValue + xGoal + xCenter + xCenterCamera \tag{4.15}$$

$$yValue = yAvoidValue + yGoal + yCenterCamera \tag{4.16}$$

As aforementioned, the drone direction supports the -100 to 100 range, whether the precious values are out of range, they will be truncated at the maximum or minimum before to pass to the drone.

The roll parameter corresponds to the xValue and the pitch refers to yValue. This design decision is chosen by the flight area and the camera collocation (see Figure 3.20). As was detailed in the previous Chapters, the throttle angle can not be controlled by the camera, so the drone is forced to fly at an altitude as constant as possible.

Chapter 5

Validation and experiments

The implementation system that was explained at Chapters 3 and 4 will be validated and adjusted by the next experiment procedures. The process is divided into three groups. The first category of validation aims the test and experiments about the colour range from the camera. The second category of validation purpose carries out the communications between the computer and the Arduino board. The final category of validation has the mission of controlling the drone in an autonomous way.

5.1 Camera colour validations

This validation process tries to adjust the camera to the proper colour in the flight area. In order to test the performances of the colour, several samples of RGB range have been chosen which were stamped by a printer in white papers (see Section C.1). It can be noticed that because of the quality of the printer the colour, can vary and lose performance with respect to the colour of a display or screen. It is important to calibrate it by using the same camera that will be used in the rest of the Master' Thesis. On the other hand, the validation might not match with the requirements and the detected object could fail or be hidden.

Once the templates are printed, in sequence by the RGB category (see Figure 5.1), they are collocated over a white surface. However, a bit of background floor is left around the surface to assure that the range selected would not conflict with the flight area, so on interference could happen.

After having the scenery ready, distinct colour values are implemented at the HSV level for the different RGB parameters. The values are configured by the toggle control (see Figure 5.2). These toggles are programmed by an independent application which can be used to validate different cameras.

The validation aims that the colour range is introduced into the main program. Several range of red, blue or green colour have been tested in HVS scale.

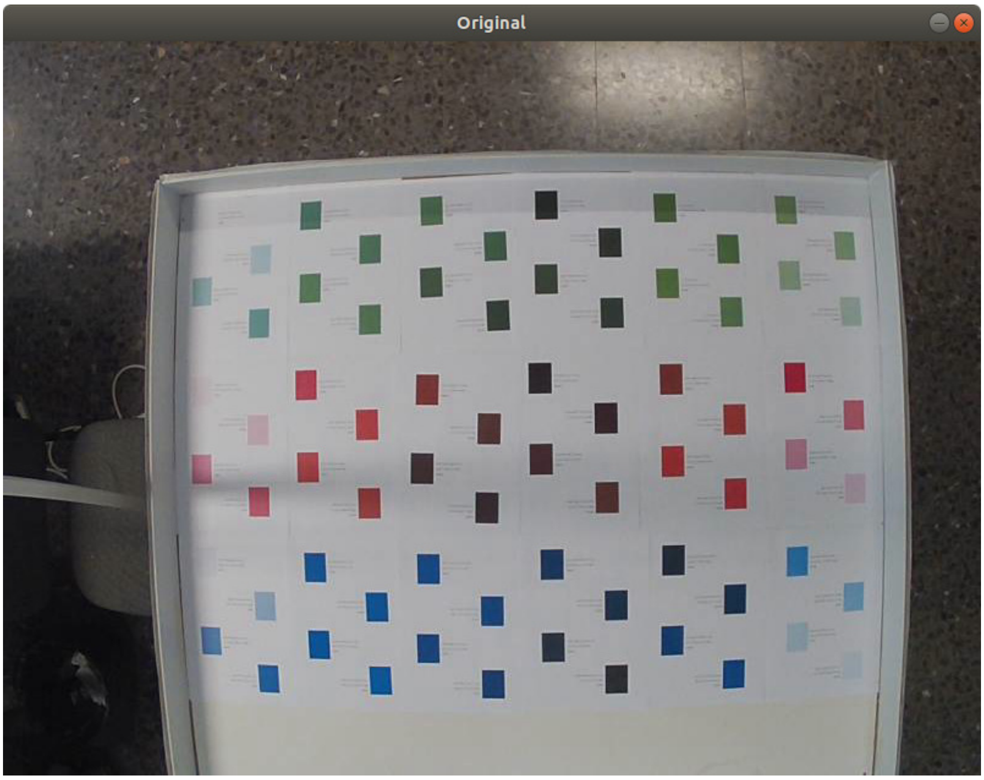
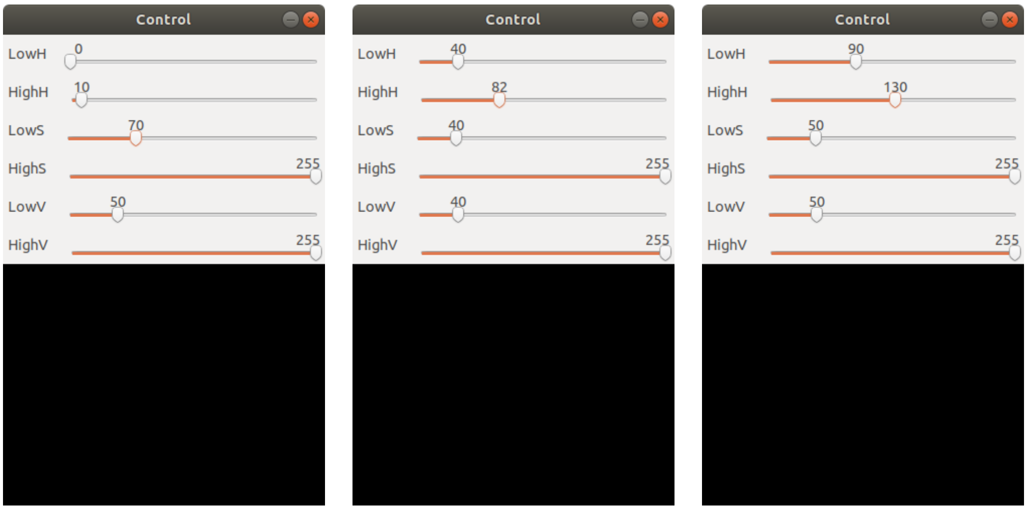


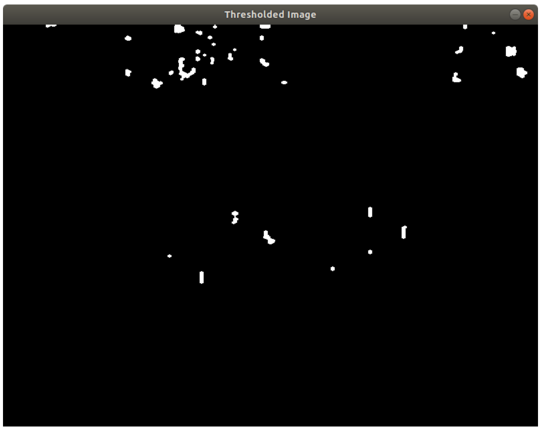
Figure 5.1: Scenery of RGB templates



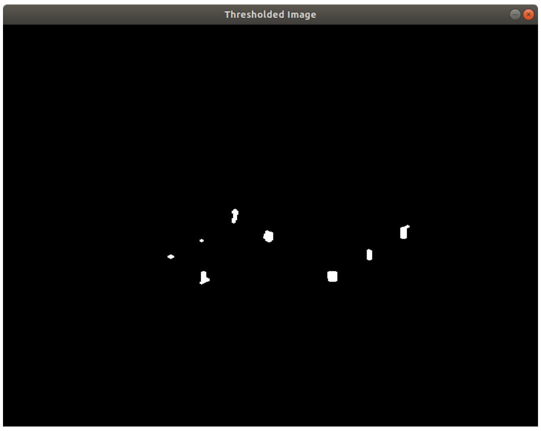
(a) Red range example (b) Green range example (c) Blue range example

Figure 5.2: Toggle control

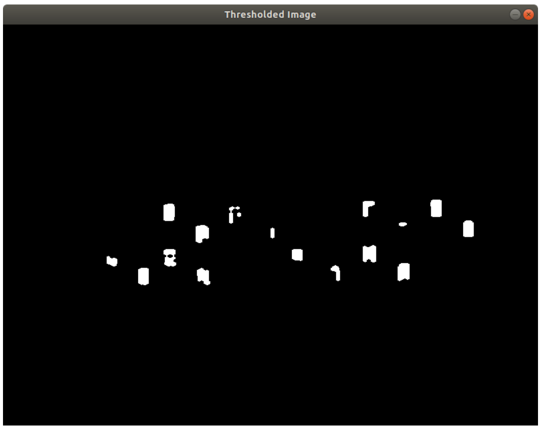
Regarding to the Red colour, several ranges of H value and the S value have been selected. The H value has been varied from scale next to zero and the other to the maximum value. On the other side, the saturation value has been vary in order to reduce the noise. The main values obtained to compare, are the ranges from Tables 5.1. The low hue values used can present high noise by the floor of the laboratory. To reduce these perturbations, it is needed to increase the low value of saturation (see Figure 5.3a and 5.3b). In addition, the range of red detected is a bit limited and the contours are not well defined. With respect to the high hue value of red, the colour is better sensed; however the contours present distortions (see Figure 5.3c and 5.3d).



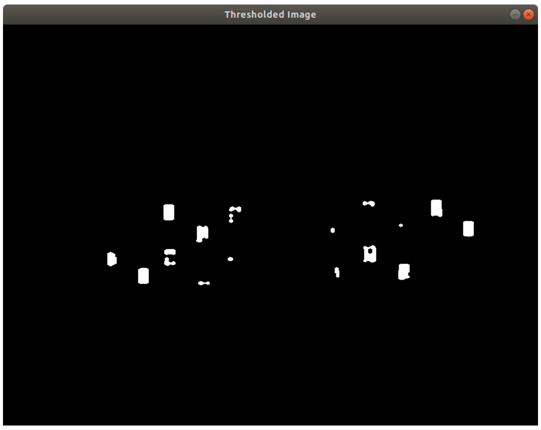
(a) H=0-10 S=50-255



(b) H=0-10 S=70-255



(c) H=170-180 S=50-255



(d) H=170-180 S=70-255

Figure 5.3: Validation of the red colour detection

Regarding to the green colour, two different ranges have been tested (see Tables 5.2). The first range detects higher number of colour with respect to the second one.

	Low	High
H	0	10
S	50	255
V	50	255

	Low	High
H	0	10
S	50	255
V	50	255

	Low	High
H	0	10
S	50	255
V	50	255

	Low	High
H	0	10
S	50	255
V	50	255

Table 5.1: Red HSV ranges

Moreover it can notice that green colour is better detected to the human eyesight than the red. This is due to the Low value of H selected. However, both ranges of green can be selected depending on the application (see Figures 5.4).

(a) H=42-82 (b) H=49-80

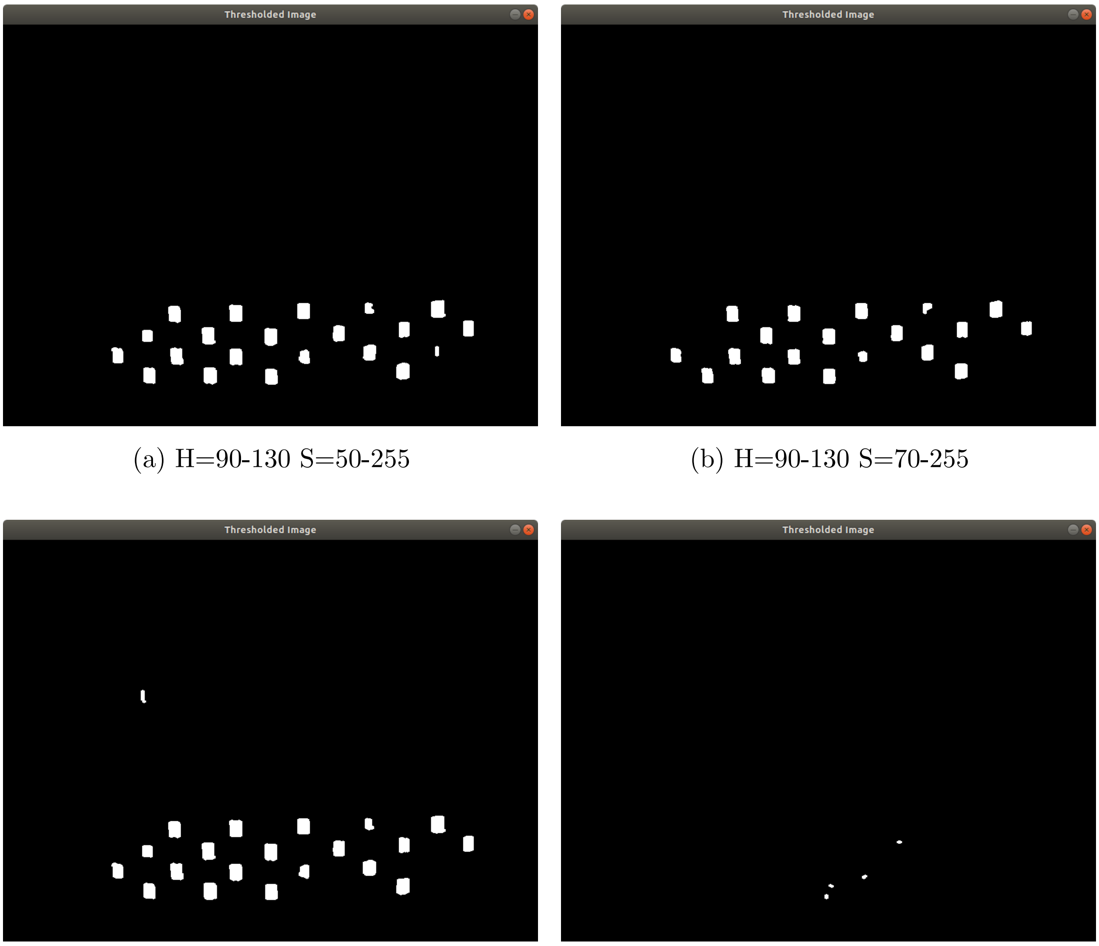
Figure 5.4: Validation of the green colour detection

	Low	High
H	0	10
S	50	255
V	50	255

	Low	High
H	0	10
S	50	255
V	50	255

Table 5.2: Green HSV ranges

According to the blue range, several values of Hue have been tested, the only difference appreciated are the number of colour samples identify in the papers. It means when the Hue values are increased, the blue colour is lower detected as the previous test (see Figures 5.5) as the green detection. Similar to the previous test, it was detected that a green colour; the #7dffc5, depending on the camera position and the light conditions could blinking (compare Figure 5.5a and 5.5c). To avoid that problem, the saturation parameter should increase a little bit.



(a) H=90-130 S=50-255

(b) H=90-130 S=70-255

(c) H=90-130 S=50-255

(d) H=110-255 S=50-255

Figure 5.5: Validation of the blue colour detection

	Low	High
H	90	130
S	50	255
V	50	255

	Low	High
H	90	130
S	70	255
V	50	255

Table 5.3: Blue HSV ranges

Regarding to the validation colour, the green and the blue colour might be selected. The green colour was selected to be the main colour detected. And the blue colour provides the orientation of the drone. This decision was taken for the computational tracking of the final application (Figure 3.7 has the configuration of the drone).

5.2 Serial communication validations

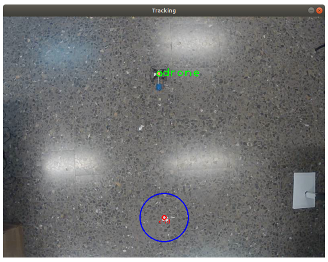
One of the main process is the communication between the PC and the Arduino board. During the validation, the Smart Inventor Board was discarded because the board only has a serial port, and that Serial port is used to transmit the control parameters from the Smart inventor to the BLE. In other words, it was detected that the BLE modules write to the Serial port frames of debugs which were used by the CoDrone library in their own procedures.

The validation aims to receive the different frames which were explained at Chapter 3. These experiments force to send and decode a fix frame during a controlled time. Repeating the test in each error programmed. As it can be considered, the serial port is a direct cable, which is supposed to receive all frames without interferences. The results obtained from the different cases were successful and each frame was decoded on time. In other words, 100% of the frames were decoded and saved at the log file without errors.

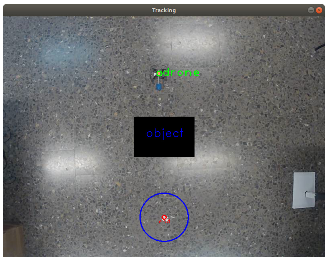
5.3 Autonomous control validation

The last validation is the flight controller. 3 different sceneries have been simulated. These sceneries are represented in Figure 5.6. It has been tried to simulate and repeat the same conditions during each experiment. The conditions are: the drone will take off from the same point and use the same light ambient, as well as, air flows. The started point is marked by a white sticky in the floor and is used to repeat the reference start coordinates. It is necessary to understand that the drone is positioned by the user and it is really difficult to create the same conditions of orientation. This validation tries to analyse if the system can make the drone achieve an area goal to land.

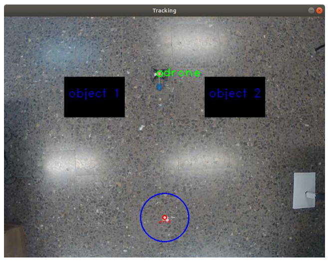
The first experiment is a stage where there is not any object (see Figure 5.6a). Figure 5.7 shows a qualitative analysis. These data were taken by an output video generated during the flight. The plot represents the duration time of each route. Different flags have been added, which are referred to problems during the flight. The red flag is the output from the camera and the green flag is the take-off problems.



(a) Scenery without objects



(b) Scenery with 1 object



(c) Scenery with 2 objects

Figure 5.6: Sceneries of the test

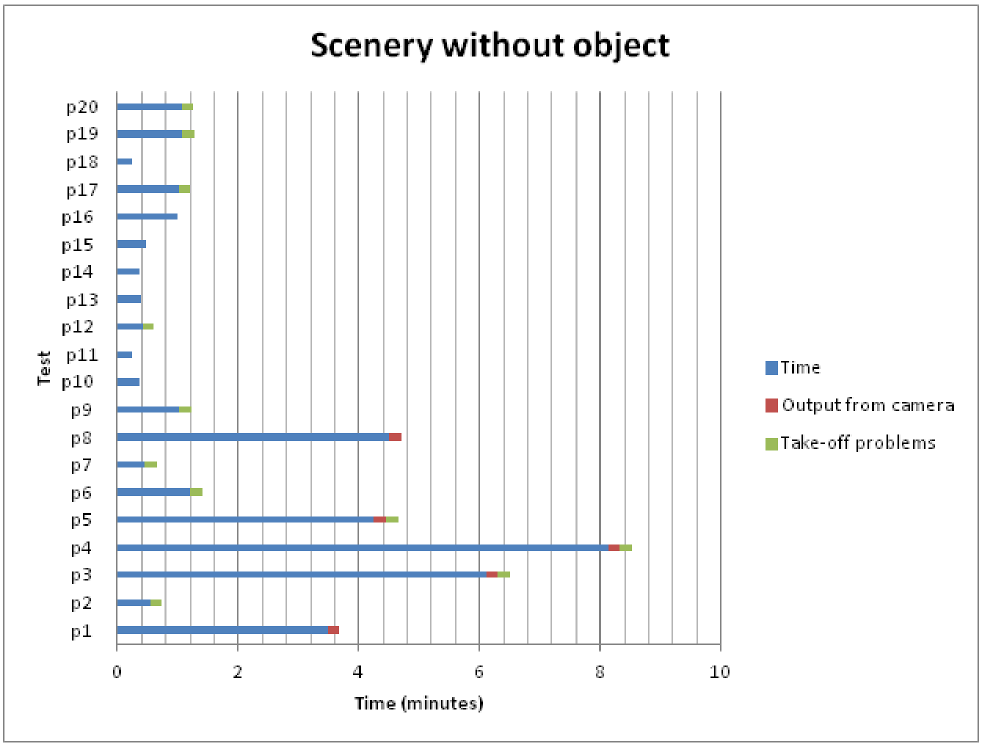
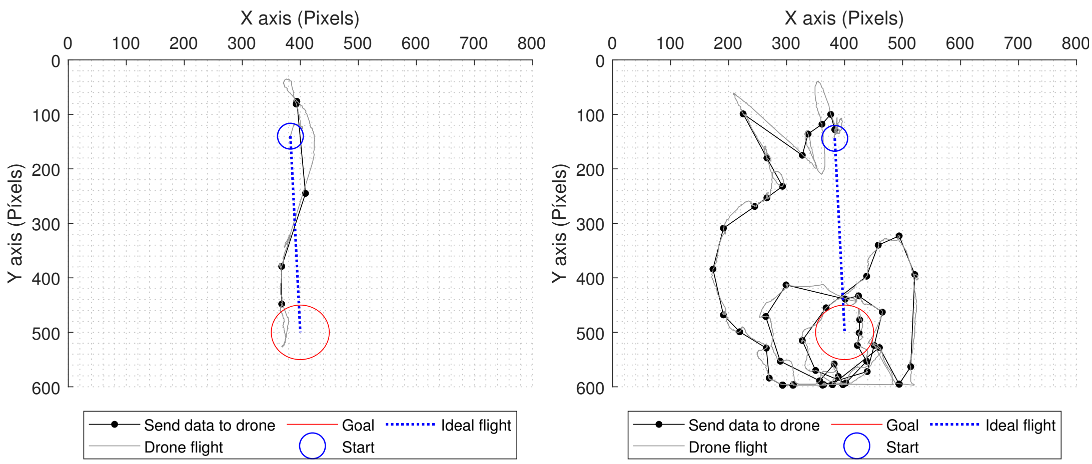


Figure 5.7: Qualitative analysis of the flight time in the scenery 1

As it was expected the longer flights are the ones that have almost one output from the camera, generating a flight of more than 2 minutes. The main reason is because of the uncontrolled location. It is really difficult to make the drone comes back to the camera. It usually returns when the drone loses height and is near the edges. The rest of times, the drone crashes or flies under the furniture as well as towards the walls, these crashed flights were discarded. The second factor that makes the flight longer are the problems regarding to achieve enough altitude. If the drone does not get a corrected altitude, the motions generate air flows that disrupt the flight. Regarding to a normal flight, the duration time is in the expected limit expected, lower than a minute.

For a quantitative analysis, the data from the flight were stored on a log file. A file stores the whole data control generated from each frame of the camera, and a second file stores the send data to the drone and the last position where the drone was located. Figure 5.8 shows some of the flight test. The rest of the analysed flights are plotted in Appendix C.2. The real flight is plotted in Figure 5.8, these values are referred to the drones flight graph. Over these data, the information coordinates have been drawn, and the ideal path from the start to the end area. The red circle is the land area and the blue one is the starting point. In addition, it has been detected that the output log files have lost some samples, and the normal lost samples are in the send file, the most lost samples are at the final conditions of landing. The reason of these lost samples, it is because the control program has priority over to write the file, and sometimes the program has not got enough time to fill it.



(a) Direct flight (b) Turns flight

Figure 5.8: Examples of drone flights of the scenery 1

When the drone lands in the end area, it sometimes bounds with the floor. These jumps and rebounds are in the drone flight graph, and it is marked as the outside samples of the red circle. The normal flight is when the drone goes directly to the target, it achieves the orientation quickly. In the rest of the cases the drone starts to turn around the target until it gets the correct angle. These turns are made outside the camera and the reason why the drone appears in the opposite side of the camera. Two different types of flight can be identified, the one that the drone goes direct to the goal and the flight where the drone turns around the target.

As a conclusion, the oscillations around the end involve the increase of the elongation. There is not a clear relation between the speed of the drone and the oscillations from the ideal path. Consequently, it can be related with external factors of the drone, such as the air flow, light conditions, or the unbalance motor. It is important to highlight that one of the motor has problems because of the several crashing during the tests and the developed process. During some motions the motor is compensated by software. To clarify it, the oscillations are calculated as the number of times where the drone crosses the ideal path, and the elongation is obtained from the sum of the distance of all points of the real route of the drone, it means by adding the vector module and this will be used in the rest of experiments.

Regarding to the second scenery (see Figure 5.6b), an object in middle of the route has been simulated by software. The drone needs to avoid to achieve the goal. As in the previous experiment, the qualitative analysis is shown in Figure 5.9. The avoid motion flight time is higher than without objects. The time rounds about several minutes. And the outputs from the camera are more frequently than the direct path. The main reason is the limited flight area.

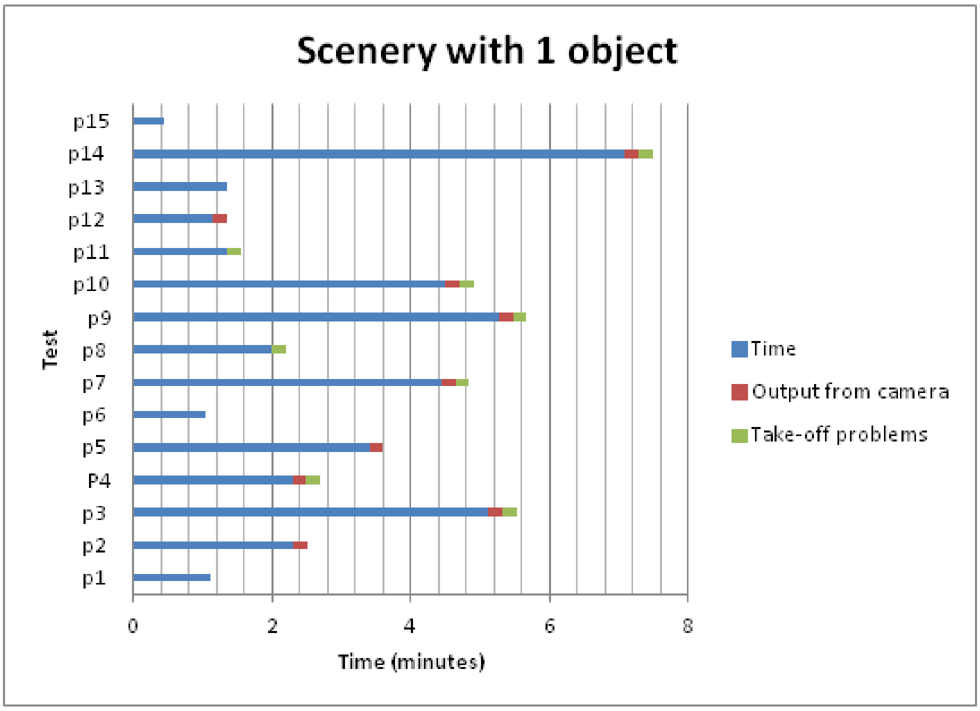
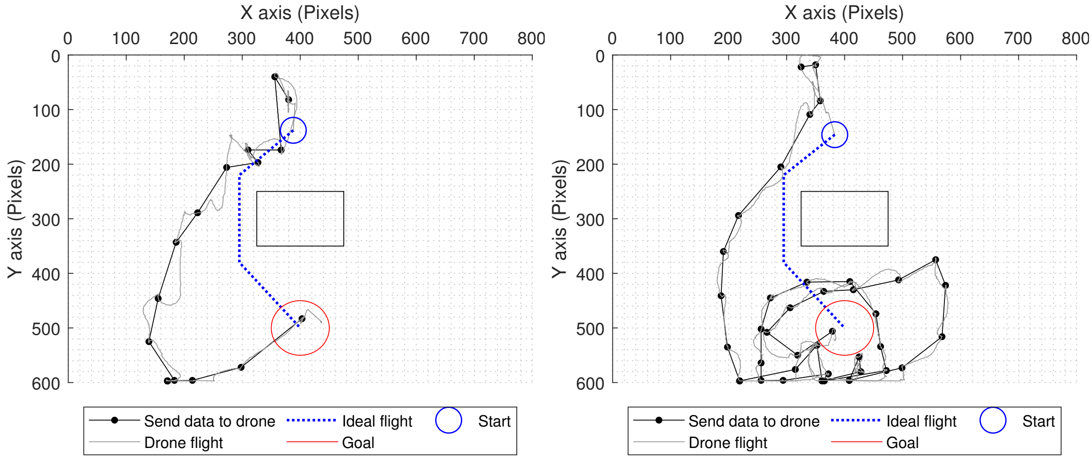


Figure 5.9: Qualitative analysis of the flight time in the scenery 2

Concerning the quantitative analysis, the some flights are shown in Figure 5.10.

The rest ones are in Appendix C.2.



(a) Direct flight (b) Turns flight

Figure 5.10: Examples of drone flights of the scenery 2

In conclusion, the drone tends to go to a side to avoid the obstacle; this is owing to the software compensation of the motor. The obstacle is simulated and it changes the conditions of the flight. It means that the drone does not crash against anything, and the drone is not held by anything. As a result the drone crosses the object near the corners. In addition, the speed to avoid the obstacle is adjusted to maintain the drone inside the camera, and these speeds are under the 50% of the motor power.

The last scenery is a combination from the first experiment and the second one (see Figure 5.6c). Two objects have been simulated at the side of the main centre path. If the drone deviates from the path, it will need to avoid the obstacles. The influence sphere of the object generates a sort path where the objects are not affected. Due to the motion problems of the drone, the drone enters in the area of the objects. According to the quantitative analysis (see Figure 5.11), the flight time goes from some seconds to several minutes depending on how the object was avoided. As the scenery of an object, there are frequently output from the camera.

The quantitative analysis the some flights are shown in Figure 5.12, the rest ones are in Appendix C.2.

These experiments obtain the result of both situations aforementioned. When the drone flies oriented, the object impulses it to the goal. However, when the drone goes disoriented, the object tends to make it oscillate around the ended area, up to get the correct approximation. These turns make the drone to go out of the camera and increase the total elongation.

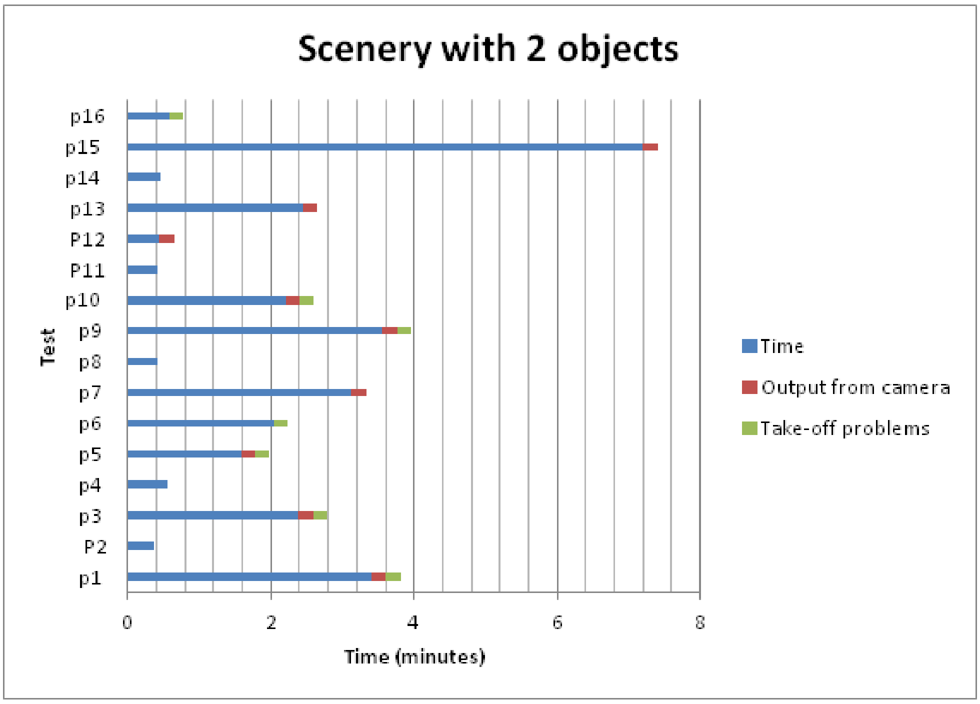
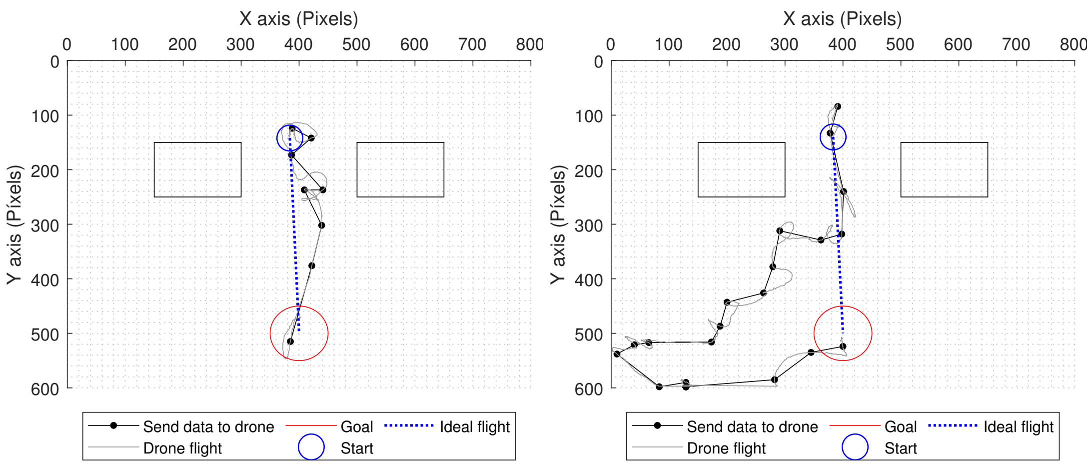


Figure 5.11: Qualitative analysis of the flight time in the scenery 3



(a) Direct flight (b) Turns flight

Figure 5.12: Examples of drone flights of the scenery 3

To sum up, the system presents some calibration problems detected during the test. However, the drone manages the flight until the area of landing, arriving to the end point. It is important to remark that the motor problems generate a worse control of the movements. The yaw control should be made at the location area without another motion, but the real flight presents some deviations because the motor. The limitation from the batteries made a complex experiment, because after the 8-10 minutes, the batteries need 40 minutes to be charged on. In addition, the condition of the light affects to the camera detection, when the drone flies near the floor. As a result, the camera confuses some points under the shadow of the white table at the corner, the chairs or the stones of the floor. It can be concluded that the system is ready to test different values of control parameter to improve its functionality.

Chapter 6

Conclusions and future lines

6.1 Conclusions

The main objective of this Master's thesis, the creation of an autonomous system have been achieved. The principal elements are the PC, the Arduino board, the Bluetooth board and the drone, are integrated and configured to work on real time, to performance their mission: flying without human interaction. The user only needs to initiate and locate the drone at the proper position, as well as to provide the serial port when the program is started. Although, the main performance has been completed, there are some features that do not follow the user requirements. Most of the peculiarities are due to real world environment, the rest are owing to the limitations of the elements and space.

We can emphasize that the camera detects the drone by an processing image, this part is adjusted to the stream camera utilized. It means that the program is calibrated using several parameters referring a particular camera, as the type and the megapixels. Then, if another camera is selected or the camera needs to be changed, the configuration process should be applied again and the colour range has to be modified. The password and user, as well as the size or centre of the camera have to be readapted and synchronized to the rest of the system.

One of the principal problems detected by the camera, it is the short and limited space dedicates to fly. When the drone goes out the camera, it is difficult to turn it back and control it. Therefore, an important requirement to fly is to have the drone in an open area inside the camera. However, the detection and the tracking satisfies the expectations in relation to the performance from the camera. Also, the code is prepared to be parallelized and incorporated new cameras, giving a better approach with more landscape of using, as others axis of mobility.

According to the communications, the serial port from the PC to the Arduino runs on an effective way, in spite of the problems encountered. The roll, yaw, pitch and throttle values arrived and are sent to the CoDrone as the specifications. However, the Bluetooth loses the link by making the drone flying without control. When the link is established, the control flows the cycle after some seconds. The reason of that is because the PCB antenna is suitable to interferences and it is a simple monopole.

Regarding to the control, a behaviour-based architecture has been programmed. It tries to adapt to the environment, allowing to fly at different sceneries and places by moving the camera.

In conclusion, the main objective are covered, but the project is at the point to design a new communication board antenna and/or a drone which can adjust to the researching and cognitive areas.

6.2 Future line works

This Master's thesis has left some new future development projects. Some of them could be:

- The improvement of the environmental area. This could be achieved by using a new camera which controls the throttle data what add a new level of the vector. It means that the drone could avoid obstacles by flying over them, so if the obstacles are closed or a wall is in the drone direction, the system will find an easy solution to move. However, the control of different cameras creates a high computer cost.

- Related to the cameras, the software could be modified to detect different types of obstacles as the drone, creating versatility and other ways to avoid the objects. Moreover, several cameras could extend the flight area inside the room.

- To design a new control board, adjusting to the user requirements and functionality. That means to create a simple board with an USB connector to plug direct to the computer and avoid cables. In addition, the board just could need a wireless module, like Bluetooth, and RGB or different LEDs to identify the transmission, and the state of the system. This is important to realise if the drone and the computer are linked and paired, then the program could be started. Also, it must be taken in account to add a microprocessor or a pin connector to incorporate new sensors or boards. Other solution is to use a commercial Bluetooth module and develop the drone control by creating the protocol and the transmission frame.

- Regarding to design a new small drone. These drones could add infrared sensors or/and ultrasound sensors, what permit to move the cameras and incorporate the whole autonomous control to the drone. Those drones need other sensors and protector frames, but always remembering the consumption of the batteries and the weight. Apart from the main components and the current regulations, that is to say the LEDs and microcontroller.

- One of the main problems of flight an indoor drone is the short autonomy time. The actual time is about 10 minutes; it could be extended by developing a new drone that uses the magnetic field or wireless charger to charge their batteries. In other words, to incorporate an electromagnetic field in the flight area capable to supply the drone during its motion.

- The main program is focused on the control of the drone; however it could be modified to interact with other programs or systems. That is to say, the drone could monitor an area and provide the information to a core which could modify the system to get high efficiency. For instance in a solar cell field, the drone could check and identify where the shadows are. This information is sent to the core where the panels could be oriented to theirs optima and higher production surface.

- Other complete different line is to incorporate to the main task other several functionalities and not only flying. This functionality could be to find particular or specific kind of objects, to monitor an area to look for hot points or intruders, also to follow a person or mobile object to check their healthy state; it means to generate alerts if they suffer any accident.

- On the other side, the colour camera detection program can be used and utilized in other control areas, as in a table where the robots need to have a green label to be located and a blue label to identify theirs orientations.

In conclusion, any activity or area what needs or requires a drone.

Appendix A

Economical budget

This Project has been developed from the beginning of September to June 2019, using the resources from the Robolabo laboratory. The budget is calculated by the labour cost and the material cost.

The salary of an engineer has been extracted by the salary received by a junior engineer working 5 hour per day, a total of 25 hours per week. This salary is based on the RBZ company.

	hours	€/hour	TOTAL
Labour cost (direct cost)	500	20 €	10000 €

Material cost (direct cost)					
	Units	Purchase cost	Time used (month)	Depreciation (year)	TOTAL
Personal computer	1	750,00 €	8	5	100,00 €
Camera AXIS M1033-W	1	229,00 €	8	5	30,53 €
CoDrone Kit	1	158,00 €	8	0	158,00 €
Battery kit	1	30,39 €	4	0	30,39 €
Inserction Led	1	0,10 €	4	0	0,10 €
1 kΩ resitance	1	0,03 €	4	0	0,03 €
Arduino Atmega	1	14,00 €	4	0	14,00 €
CP2102	1	8,00 €	5	0	8,00 €
TOTAL Material cost					**341,05 €**

Overhead cost (indirect cost)	15%	over CD	**1.551,16 €**
Industrial benefit	6%	over CD+CI	**713,53 €**

Consumables	
120 Cables	**6,80 €**
RGB Ink Printer	**180,00 €**
A4 paper packet of 2500 sheets	**30,00 €**

SUBTOTAL bUDGET		12.822,54 €
Applicable VAT	21%	2.692,73 €

The material cost is estimated by the product bought and price from official and secure delivery pages. It means, some products can be found cheaper but the quality could vary from the official it, due to the certification test.

TOTAL Budget	**15.515,28 €**

Appendix B

Ethical, economic, social and environmental aspects

B.1 Introduction

The main aim of this Master's Thesis is to develop and implement a secure environment system for the controlled flight of a drone. This system could be utilized to program different control algorithms to be tested in the real word. The objective, regarding to the social aspect, is the creation of an area where the drones can be used to help people in different task, as they can be used to guide a person inside of a confuse build like a hospital or a university. The system should incorporate other complex tasks such as voice recognition, machine learning algorithms and/or cognitive networks to control several numbers of drones. Other possible tasks are the use to search a particular object across a room or building, like the keys or the mobile phone.

There is a positive impact of the creation a secure environment applying the regulation (see Chapter 1). This scenery could be the first step to explore the reaction of human habits with aircraft robots. As the anxieties of some human presents to the surveillance roles, where the drones have been involved. It means, to teach the huge possibility that the drones can introduce. In addition, the system was focused to control an aircraft. However, it can be modified in at easy way to introduce the 2D robots as the e-puck or autonomous cars. The main target group of this project is the scientific. Although if the guide system is developed, the group could be included in open organizations, as hospitals, universities, fair organizations, each location where an inside area is prepared. Table B.1 has identified the production process impacts.

	Ethical aspects	Economics aspects	Legal aspects	Social aspects	Environment aspects
Production and test Distribution	Use the open source (OpenCV)	Low cost design and implementation	The identification cards and the flight areas	Design for inside drone, but adapt to other robots	The Li-ion batteries
Maintenance		Low cost maintenance			
Reuse and recycling		Easy adaptable			

Table B.1: Impacts in relation to the area of influence and the life cycle phase of the product

B.2 Description of significant impacts associated with this Project

In this Section the main highlight impacts are approached which were introduce in Table B.1.

Concerning the ethical aspect, all programs involved are open source. The OpenCV library is programmed in C++ which has a huge community of users and contributions. Also, it counts with a database with the function and methods spread by the version installed for the user. It allows the redistribution and use of the software without legal conditions. Similarly occurs to the Arduino IDE and libraries. The ethical aspect of the drone must be analysed when the drone is controlled by a person. We can evaluate the actions of a person bases on the use of the drone. Consequently, the autonomous control have to be appropriated to the final application, where the drone is not involved on killer activities.

With regard to the economic aspects, one of the objectives is the implementation of a system with low-cost design and reusing the principal components, as the camera or the PC. This explains the reuse of the previous materials from the laboratory, reducing the purchases to the kit of the drone. Respect to the maintenance, it is also low-cost and easy to maintain, where the elements could be switched to other by just calibrating the main program again. It means, that other cameras could be employed and obtained better or worst results, as well as computers or drones. As a result a great flexibility and versatility of the system have been achieved.

In terms of the legal aspects, the main regulation aspect has been collected at Chapter 1. Recalling what has been already said, it is mandatory to have an identification card of the drone which includes the manufacturing name, type of drone, model of drone, serial number, owner's name and telephone and be fireproof.

Regarding the social aspect, the design has been focused to be accessible to everybody, by using open source software and low-cost elements. Also, a multiversity platform to test and control different algorithms in the real word have been created. The project has centred its attention to the different investigation scientist, with the idea of transferring to large organizations like universities or hospitals.

According to the environmental aspect, the use of drones has a direct impact on the environmental. The power supply is getting from the Lithium batteries which have several effects to the nature; other effect is the wireless communications. Although the flight is inside of a building, the effect of the drones can affect to the migratory birds (see Section B.3).

B.3 Detailed analysis of a significant impact

In this Section, the environment impact of the Li-ion battery as power supply is analysed in detail along with the Bluetooth problems.

A recent life cycle analysis of the lithium-ion batteries has remarked the importance of recycling it to reduce the health and environment impact. This research shows the highest environment impacts are the resource depletion, global warming, the ecological toxicity and the human impacts [69, 1].

In addition, the lithium mining industry is a growing sector due to increasing the demand of lithium-ion batteries to Smartphones, electric cars, etc. However it is having repercussions on the environmental. One of the first problems is the water required to the production. It needs 500,000 gallons of water (about 1892705,89 litters) to produce a single tonne of lithium. In a climate so dry as the Atacama Desert and the Salar Uyuni, it impacts to the neighbouring farmer, who uses the water to cultivate their crops and livestock. Other catastrophe occurred in the Liqi River in Tibet, where the toxic chemical compounded substances used to extract the Lithium was filtered to the rivers. These substances could be filtered to the streams, wells and water supplies too. As a result, it can affect the wildlife species, for instance fish or cattle were poisoned [71].

As a reflection, the manufacturing techniques and power supply capacities presents several limitations, where the time flight is really short. However this is the only way that presents better capacity and performances than other supply techniques. In other words, to hold the freedom that an umbilical drone can give.

Regarding to the Bluetooth technology, the wireless personal area network and the wireless local area networks share the same 2.4GHz frequency band, what produces the both protocols works together. So as much networks and devices working in the same band, much interference between them are obtained. These interferences can be affected at the control system, creating unstable flights or other systems that could even kidnap the drone by entering in middle of the communication and adapt the same protocol. The drone kidnapping can be utilized to crash the drone against to the people or with the idea of creating an accident.

B.4 Conclusions

Summarizing, the study of the impacts offers a final prototype system with positive effects to the society. The low-cost decisions have presented several problems of implementations but they highlight the important requirements to develop an embedded board or drone, as well as the sensors needed to be friendly with the environmental aspects. In addition, the legal regulations have been taking into account, as the accessibility to everybody and the coherent ethic principles.

Appendix C

Appendix C

C.1 Colour test and template of the sheets

In this Appendix is recapped the colour text during the validation experiments. Table C.1 has the red colour values, Table C.2 has the green ones and Table C.3 has the blue ones. After the tables an example about the configuration template is shown.

HEX	RGB	HSV
#ffcecc	R = 255, G = 206, B = 204	H =2.4, S=20, V=100
#ffa9a5	R = 255, G = 169, B = 165	H =2.7, S=35.3, V=100
#ff847d	R = 255, G = 132, B = 125	H =3.2, S=51, V=100
#ff4c43	R = 255, G = 76, B = 67	H =2.9, S=73.7, V=100
#ff261b	R = 255, G = 38, B = 27	H =2.9, S=89.4, V=100
#f30c00	R = 243, G = 12, B = 0	H =3, S=100, V=95.3
#cc0a00	R = 204, G = 10, B = 0	H =2.9, S=100, V=80
#a50800	R = 165, G = 8, B = 0	H =2.9, S=100, V=64.7
#910700	R = 145, G = 7, B = 0	H =2.9, S=100, V=56.9
#6a0500	R = 106, G = 5, B = 0	H =2.8, S=100, V=41.6
#430300	R = 67, G = 3, B = 0	H =2.7, S=100, V=26.3
#1b0100	R = 27, G = 1, B = 0	H =2.2, S=100, V=5.3
#2f0200	R = 42, G = 2, B = 0	H =2.6, S=100, V=9.2
#560400	R = 86, G = 4, B = 0	H =2.8, S=100, V=33.7
#7d0600	R = 1250, G = 6, B = 0	H =2.9, S=100, V=49
#a50800	R = 165, G = 8, B = 0	H =2.9, S=100, V=64.7
#cc0a00	R = 204, G = 10, B = 0	H =2.9, S=100, V=80
#f30c00	R = 243, G = 12, B = 0	H =3, S=100, V=95.3
#ff261b	R = 255, G = 38, B = 27	H =2.9, S=89.4, V=100
#ff4c42	R = 255, G = 76, B = 66	H =3.2, S=74.1, V=100
#ff716a	R = 255, G = 113, B = 106	H =2.8, S=58.4, V=100
#ff9691	R = 255, G = 150, B = 145	H =2.7, S=43.1, V=100
#ffbcb8	R = 255, G = 188, B = 184	H =3.4, S=27.8, V=100
#ffe1df	R = 255, G = 255, B = 223	H =3.7, S=12.5, V=100

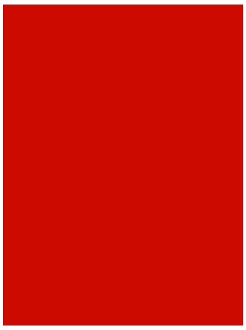
Table C.1: Red colour

HEX	RGB	HSV
#cef0bc	R = 206, G = 240, B = 188	H =99, S=22, V=94
#b6e89c	R = 182, G = 232, B = 156	H =99.5, S=62.3, V=76.1
#abe58c	R = 171, G = 229, B = 140	H =99.1, S=38.9, V=89,8
#93dd6c	R = 147, G = 221, B = 108	H =99.3, S=51.1,7, V=86.7
#7cd64c	R = 124, G = 214, B = 76	H = 99.1, S=64.5, V=83,9
#65cc2f	R = 101, G = 204, B = 47	H =99.4, S=77, V=80
#56ac27	R = 86, G = 172, B = 39	H =99.8, S=77.3, V=67.5
#65cc2f	R = 70, G = 140, B = 32	H =99.4, S=77, V=80
#7cd64c	R = 54, G = 108, B = 25	H =99.1, S=64.5, V=83.9
#264c11	R = 38, G = 76, B = 17	H =98.6, S=77.6, V=29.8
#162c0a	R = 22, G = 44, B = 10	H =98.8, S=77.3, V=17.3
#0e1c07	R =14, G = 28, B = 7	H =100, S=75, V=11
#004325	R = 0, G = 67, B = 37	H =153.1, S=100, V=26.3
#006a3a	R = 0, G = 106, B = 58	H =152.8, S=100, V=20.8
#009150	R = 0, G = 145, B = 80	H =153.1, S=100, V=56.9
#00b866	R = 0, G = 184, B = 102	H =153.3, S=100, V=72.2
#00df7b	R = 0, G = 233, B = 123	H =153.1, S=100, V=87.5
#08ff90	R = 8, G = 255, B = 144	H =153, S=96.9, V=100
#2fffa2	R = 47, G = 255, B = 162	H =153.2, S=81.6, V=100
#56ffb3	R = 86, G = 255, B =179	H =153, S=66.3, V=100
#7dffc5	R = 125, G = 255, B = 197	H =153.2, S=100, V=74.5
#a5ffd6	R = 8, G = 255, B = 144	H =152.7, S=35.3, V=100
#ccffe8	R = 204, G = 255, B = 232	H =152.9, S=20, V=100
#f3fffa	R = 243, G = 255, B = 250	H =155, S=4.7, V=100

Table C.2: Green colour

HEX	RGB	HSV
#dfeeff	R = 223, G = 238, B = 255	H =211.9, S=12.5, V=100
#cce3ff	R = 204, G = 227, B = 255	H =212.9, S=20, V=100
#a5cdff	R = 165, G = 205, B = 255	H =213.3, S=35.3, V=100
#6aadff	R = 106, G = 1773, B = 255	H =213, S=58.4, V=100
#4298ff	R = 66, G = 152, B = 255	H =212.7, S=74.1, V=100
#1b82ff	R = 27, G = 130, B = 255	H =212.9, S=89.4, V=100
#006ef3	R = 0, G = 110, B = 243	H =212.8, S=100, V=95.3
#005ccc	R = 0, G = 92, B = 204	H =212.9, S=100, V=80
#004aa5	R = 0, G = 74, B = 165	H =213.1, S=100, V=64.7
#00397d	R = 0, G = 57, B = 125	H =212.6, S=100, V=49
#002756	R = 0, G = 39, B = 86	H =212.8, S=100, V=33.7
#00152f	R =0, G = 21, B = 77	H =213.2, S=100, V=18.4
#02002f	R = 2, G = 0, B = 47	H =242.6, S=100, V=18.4
#040056	R = 4, G = 0, B = 86	H =242.8, S=100, V=33.7
#06007d	R = 6, G = 0, B = 125	H =242.9, S=100, V=49
#0800a5	R = 0, G = 184, B = 102	H =153,3, S=100, V=72.2
#0a00cc	R = 10, G = 0, B = 204	H =242.9, S=100, V=80
#0c00f3	R = 12, G = 0, B = 243	H =243, S=100, V=95.3
#261bff	R = 38, G = 27, B = 255	H =242.9, S=89.4, V=100
#4c42ff	R =76, G = 66, B = 255	H =243.2, S=74.1, V=100
#716aff	R = 113, G = 106, B = 255	H =242.8, S=58.4, V=100
#9691ff	R = 150, G = 145, B = 255	H =242.7, S=43.1, V=100
#bcb8ff	R = 188, G = 184, B = 255	H =243.4, S=27.8, V=100
#e1dfff	R = 225, G = 223, B = 255	H =243.8, S=12.5, V=100

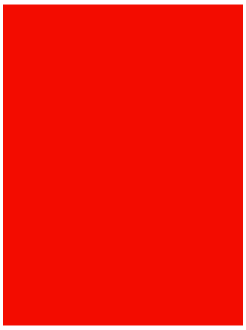
Table C.3: Blue colour



#cc0a00

RGB => R = 204, G = 10, B = 0

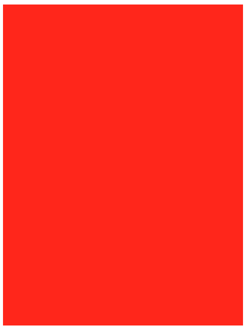
HSV =>H =2.9, S=100, V=80



#f30c00

RGB => R = 243, G = 12, B = 0

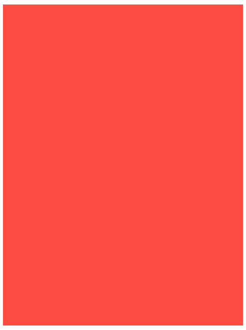
HSV =>H =3, S=100, V=95.3



#ff261b

RGB => R = 255, G = 38, B = 27

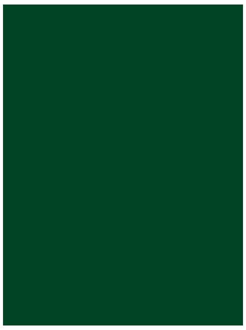
HSV =>H =2.9, S=89.4, V=100



#ff4c42

RGB => R = 255, G = 76, B = 66

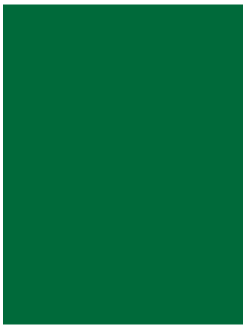
HSV =>H =3.2, S=74.1, V=100



#004325

RGB => R = 0, G = 67, B = 37

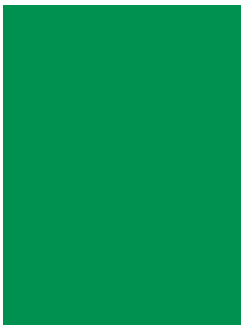
HSV =>H =153.1, S=100, V=26.3



#006a3a

RGB => R = 0, G = 106, B = 58

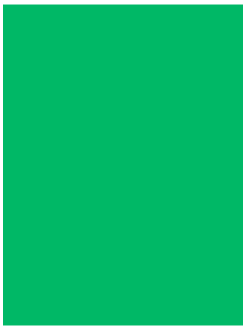
HSV =>H =152.8, S=100, V=20.8



#009150

RGB => R = 0, G = 145, B = 80

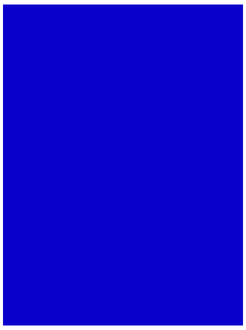
HSV =>H =153.1, S=100, V=56.9



#00b866

RGB => R = 0, G = 184, B = 102

HSV =>H =153.3, S=100, V=72.2



#0a00cc

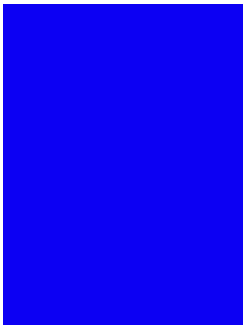
RGB => R = 10, G = 0, B = 204

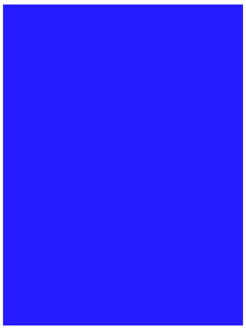
HSV =>H =242.9, S=100, V=80

#0c00f3

RGB => R = 12, G = 0, B = 243

HSV =>H =243, S=100, V=95.3





#261bff

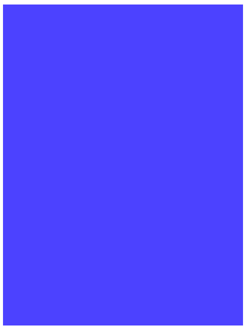
RGB => R = 38, G = 27, B = 255

HSV =>H =242.9, S=89.4, V=100

#4c42ff

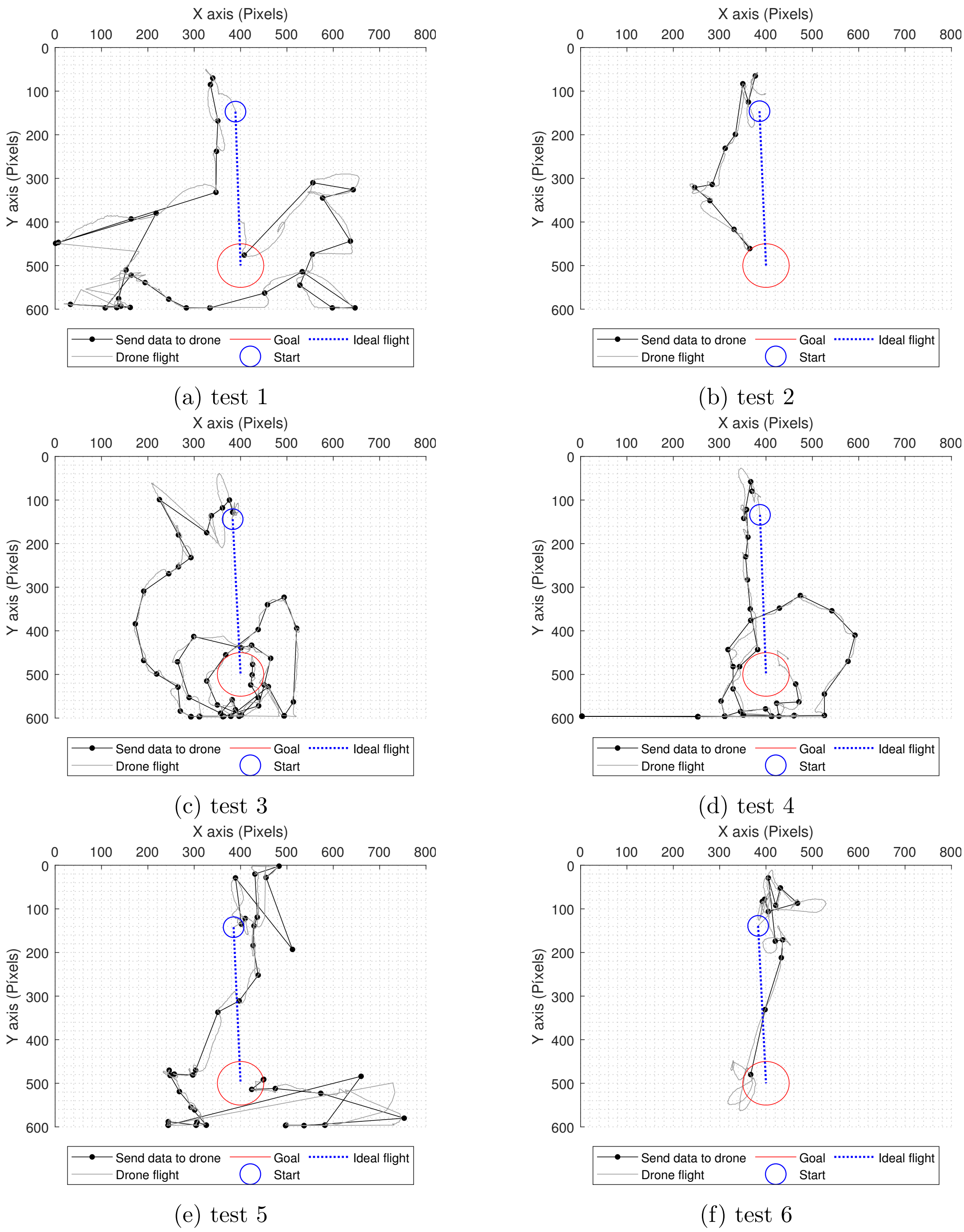
RGB => R =76, G = 66, B = 255

HSV =>H =243.2, S=74.1, V=100



C.2 Flight experiments

C.2.1 Results from the scenery without objects

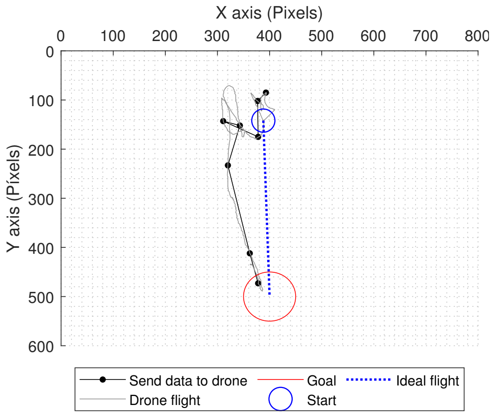


(a) test 1 (b) test 2

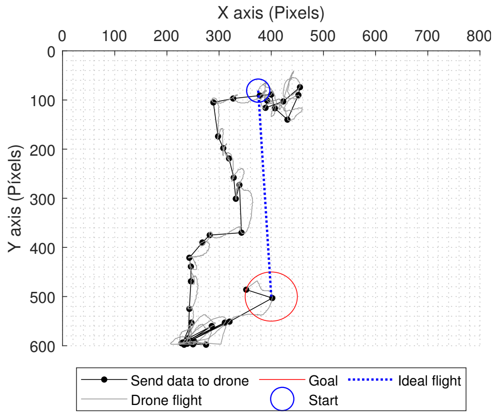
(c) test 3 (d) test 4

(e) test 5 (f) test 6

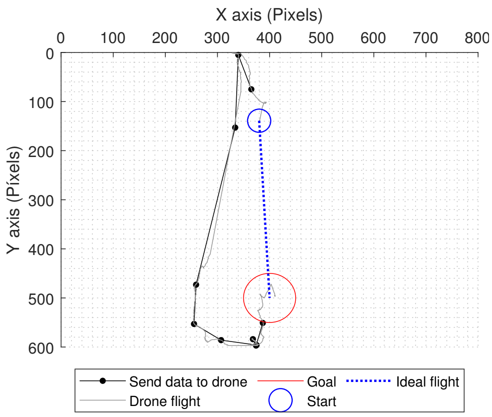
Figure C.1: Results from the scenery without objects



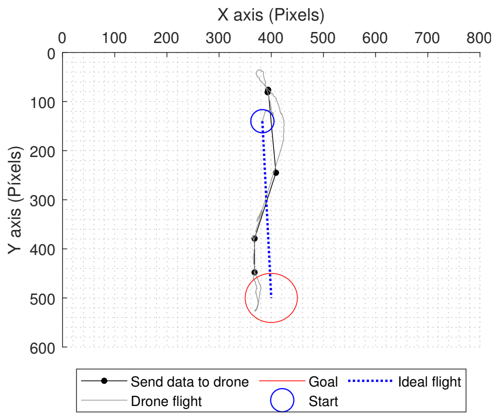
(a) test 7



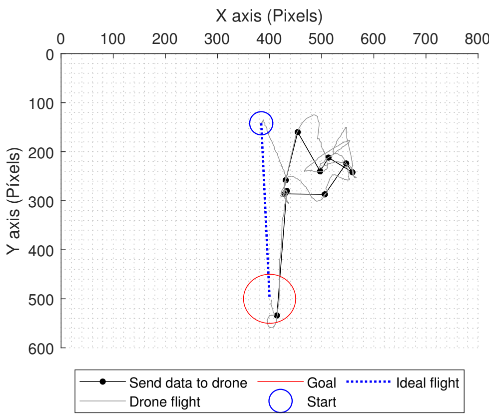
(b) test 8



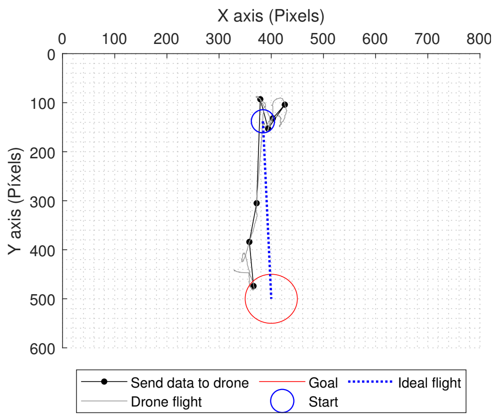
(c) test 9



(d) test 10

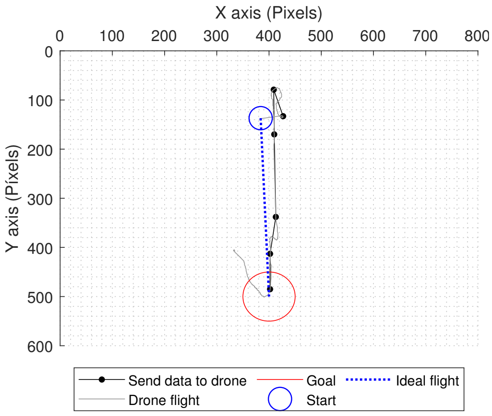


(e) test 11

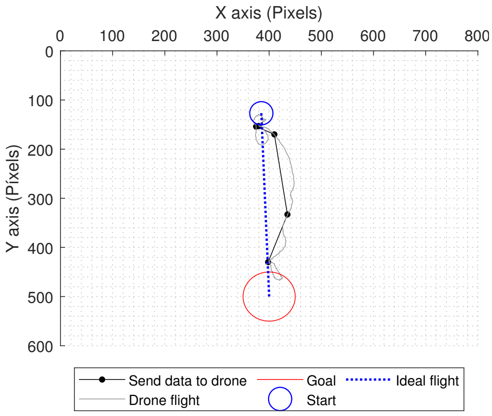


(f) test 12

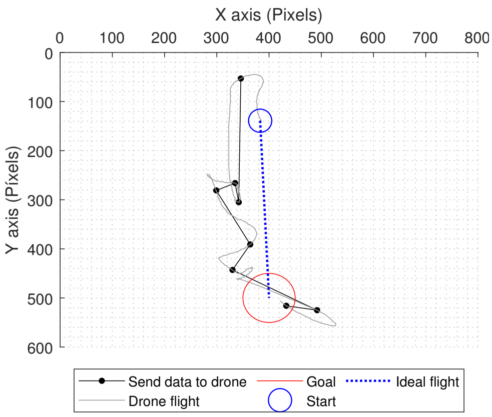
Figure C.2: Results from the scenery without objects



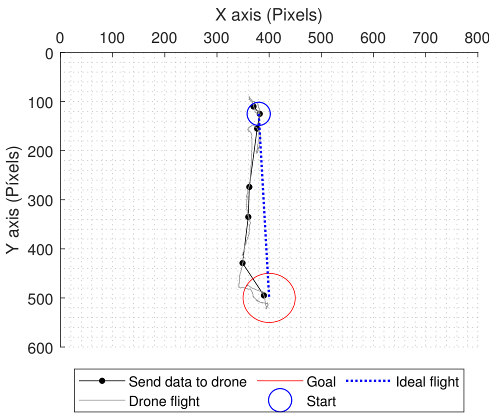
(a) test 13



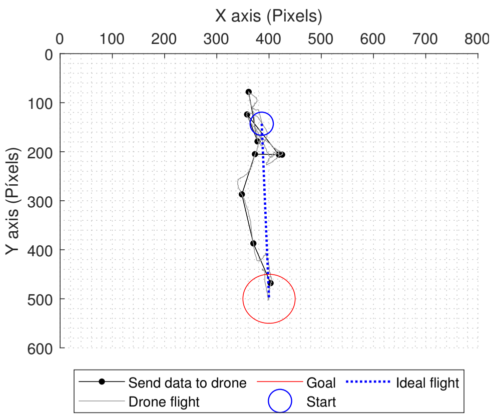
(b) test 14



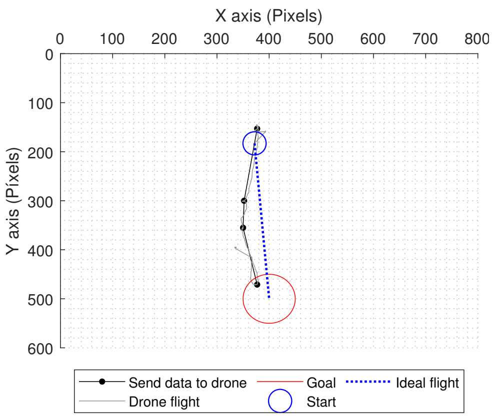
(c) test 15



(d) test 16

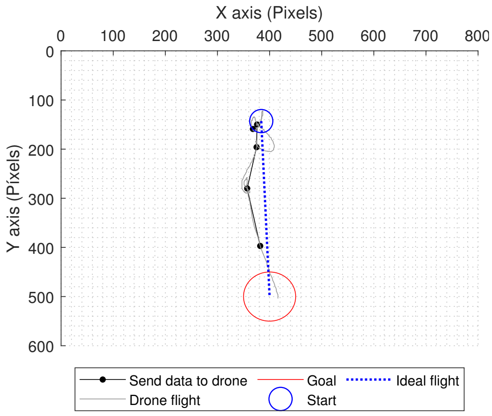


(e) test 17

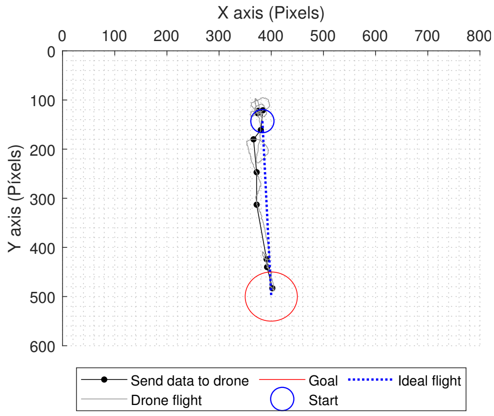


(f) test 18

Figure C.3: Results from the scenery without objects



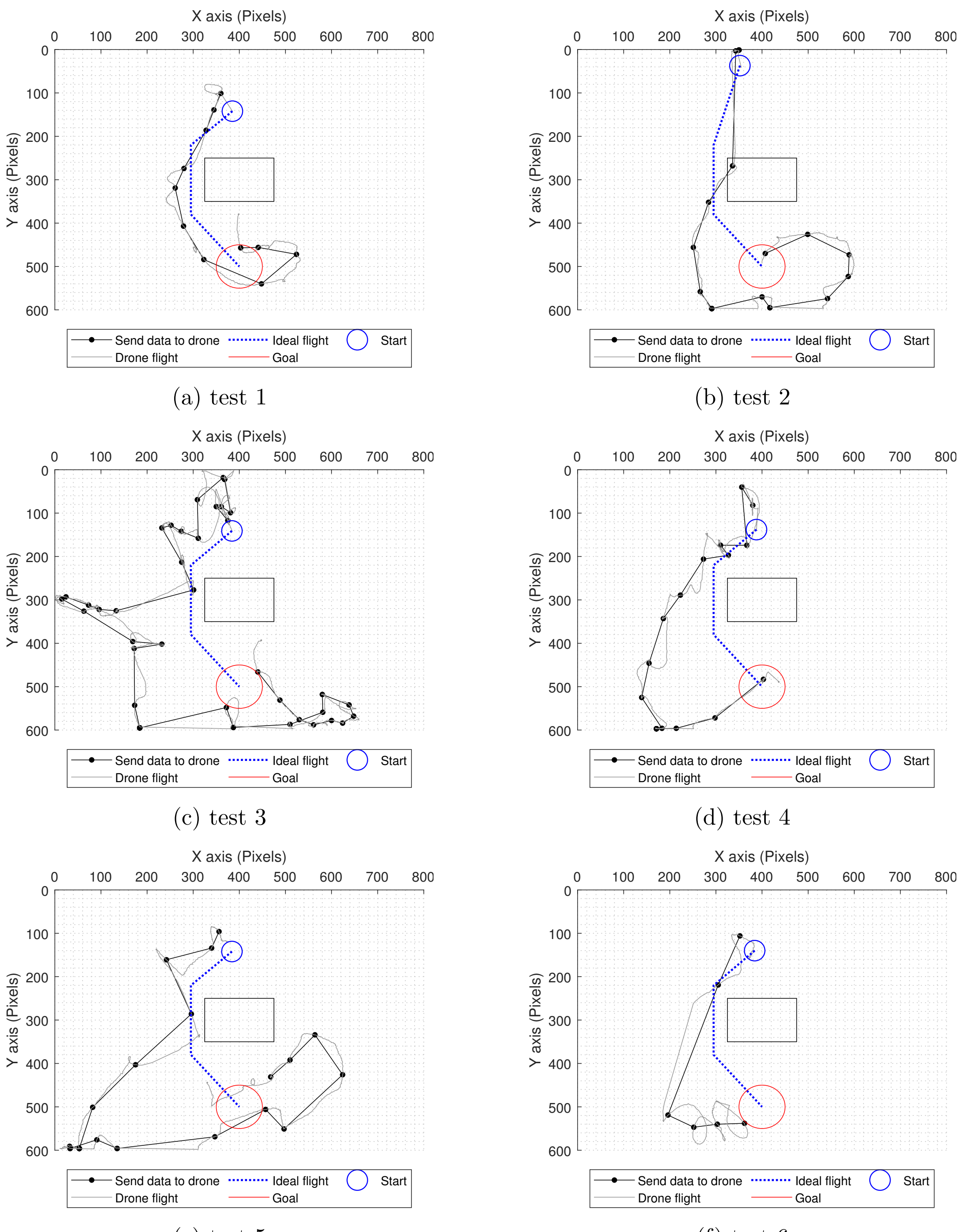
(a) test 19



(b) test 20

Figure C.4: Results from the scenery without objects

C.2.2 Results from the scenery with 1 object

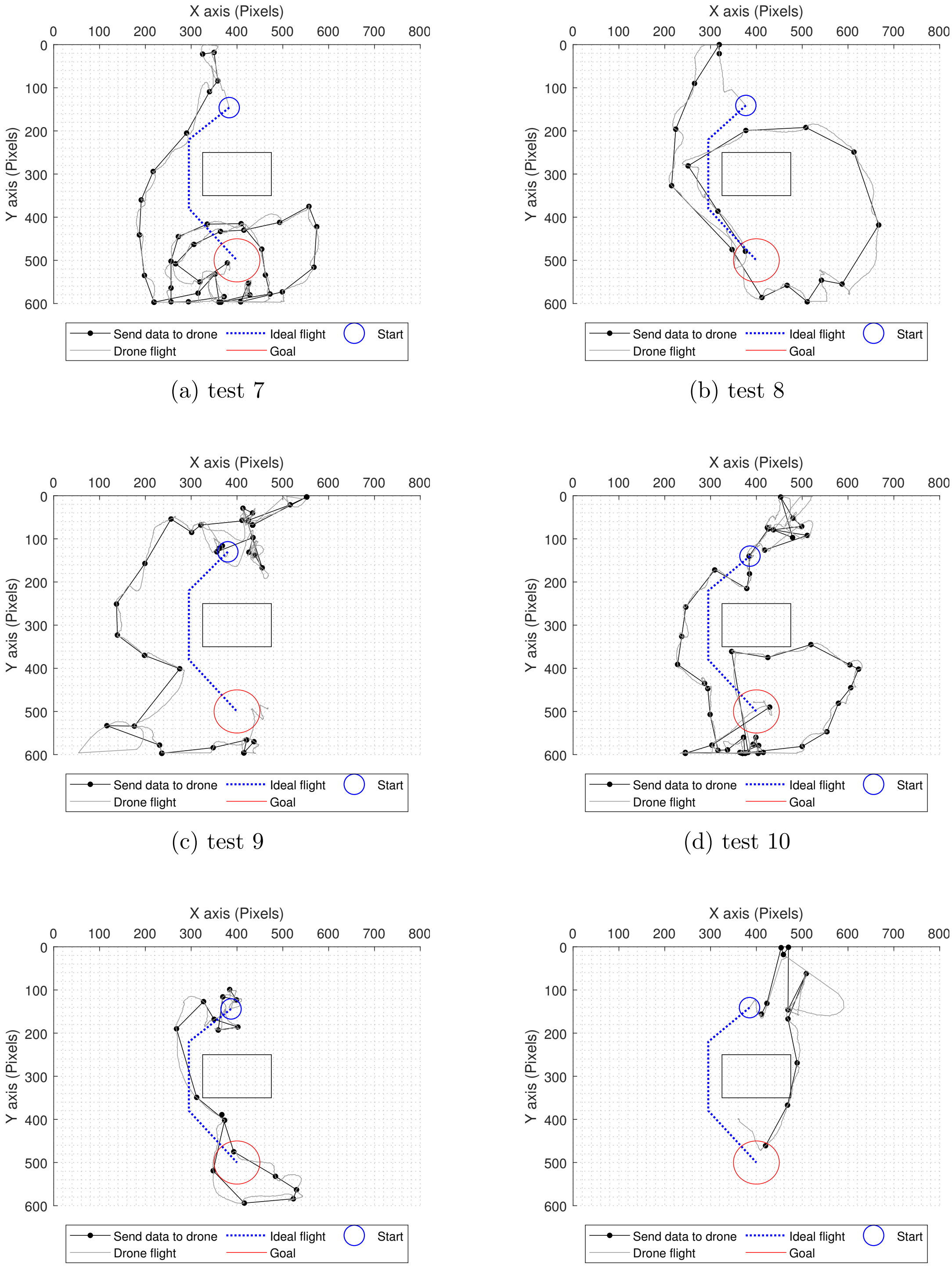


(a) test 1 (b) test 2

(c) test 3 (d) test 4

(e) test 5 (f) test 6

Figure C.5: Results from the scenery with 1 object



(a) test 7

(b) test 8

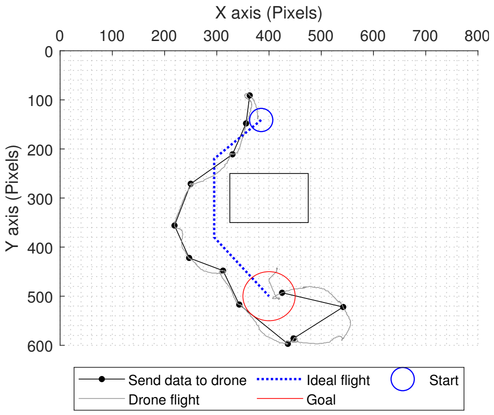
(c) test 9

(d) test 10

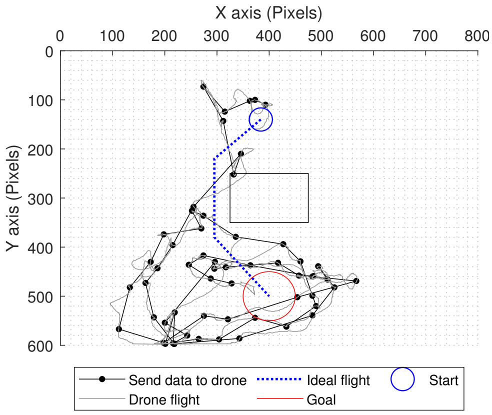
(e) test 11

(f) test 12

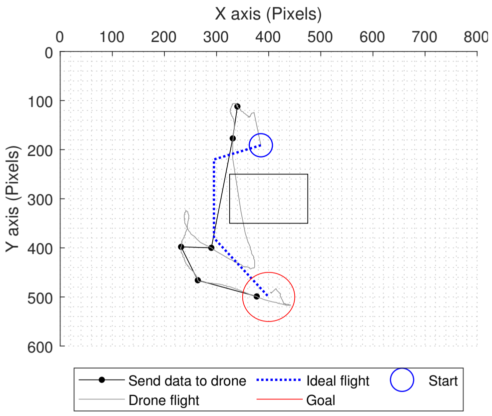
Figure C.6: Results from the scenery with 1 object



(a) test 13



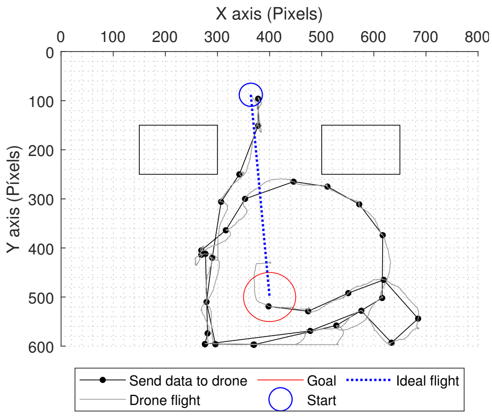
(b) test 14



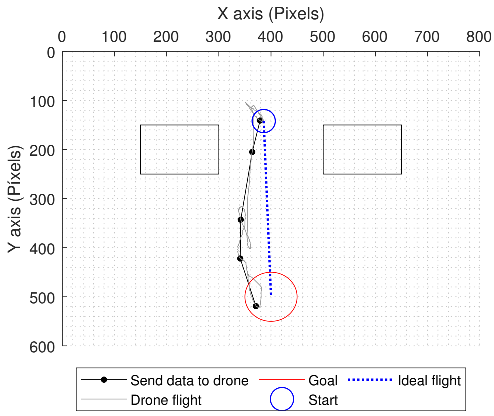
(c) test 15

Figure C.7: Results from the scenery with 1 object

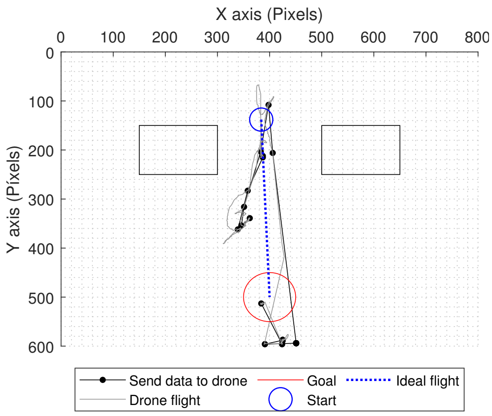
C.2.3 Results from the scenery with 2 objects



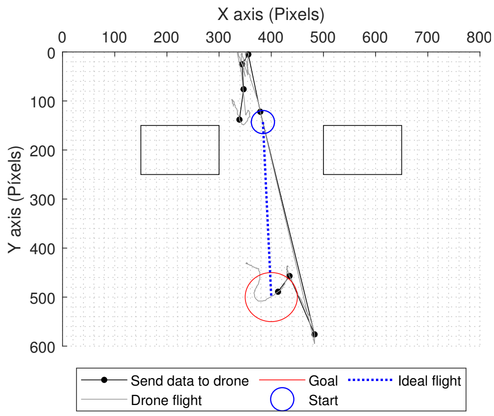
(a) test 1



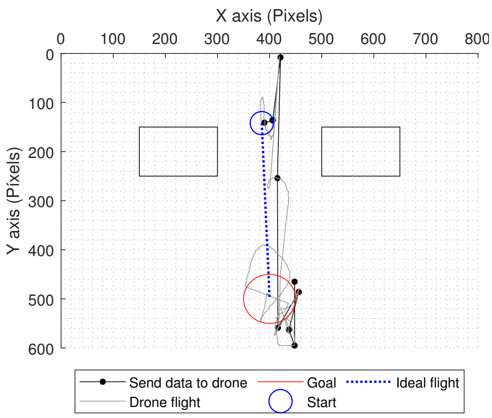
(b) test 2



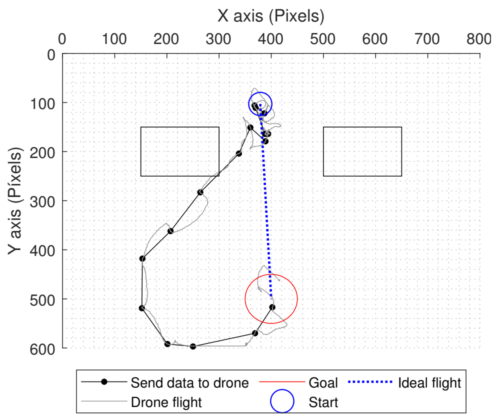
(c) test 3



(d) test 4

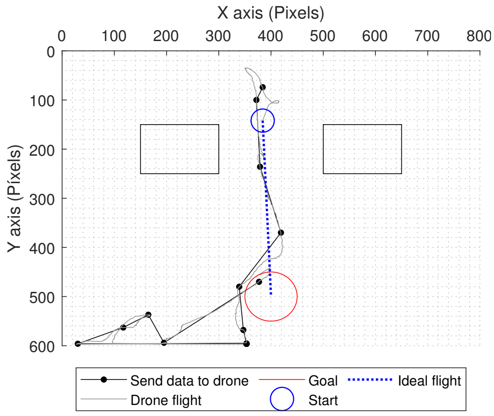


(e) test 5

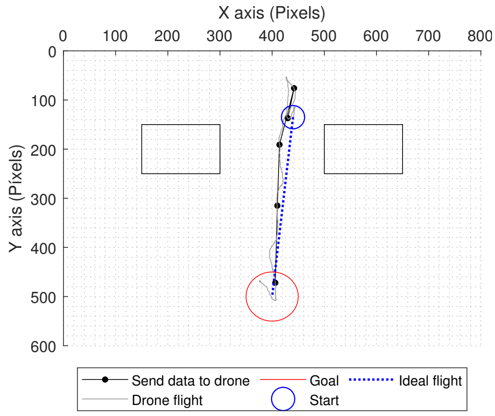


(f) test 6

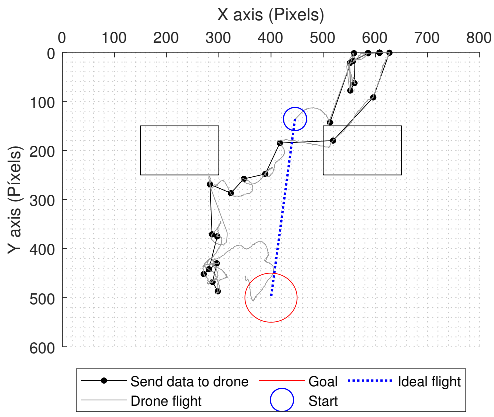
Figure C.8: Results from the scenery with 2 objects



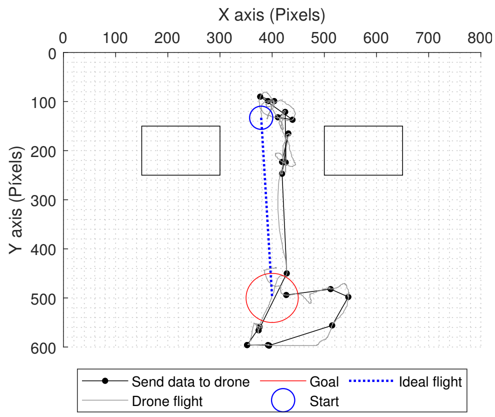
(a) test 7



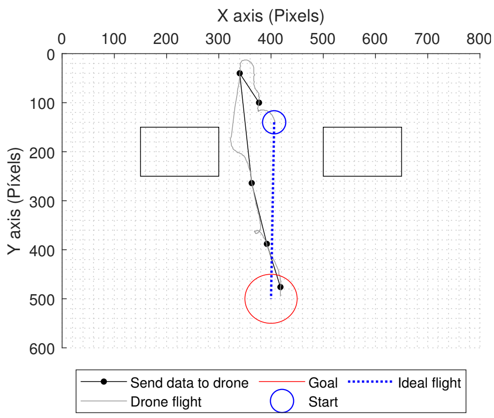
(b) test 8



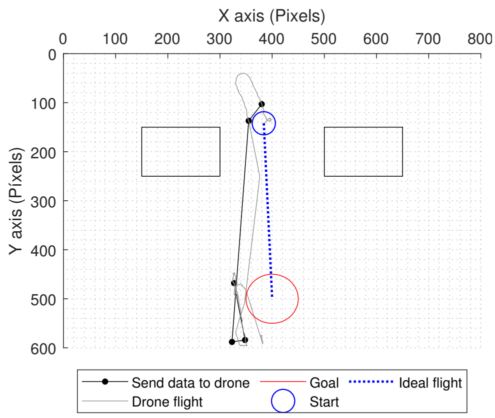
(c) test 9



(d) test 10

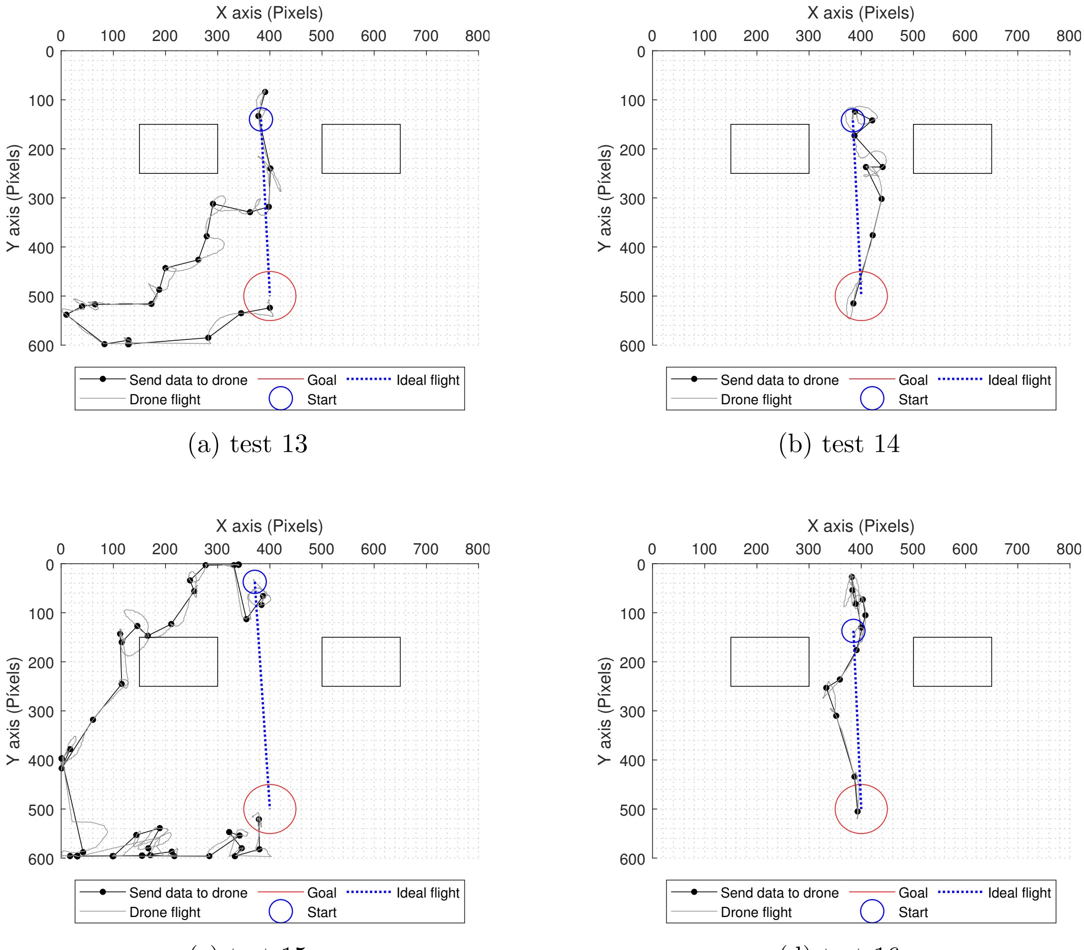


(e) test 11



(f) test 12

Figure C.9: Results from the scenery with 2 objects



(a) test 13

(b) test 14

(c) test 15

(d) test 16

Figure C.10: Results from the scenery with 2 objects

Bibliography

[1] ABRI. Study looks at environmental impacts of lithium ion batteries. https://batteryrecycling.org.au/environmental-impact-of-lithium-ion-batteries/, 2015. 28/05/2019.

[2] T. B. T. Aerial. The physycs of the quadcopter flight. https://blacktieaerial.com/the-physics-of-quadcopter-flight/, 2014. 21/05/2019.

[3] AESA. Preguntas frecuentes sobre el real decreto 1036/2017. https://www.seguridadaerea.gob.es/lang_castellano/cias_empresas/trabajos/rpas/faq/default.aspx, 2019. 17/04/2019.

[4] S. ALPAGU. Alpagu. https://www.stm.com.tr/en/products/alpagu. 20/04/2019.

[5] Arduino. Download the arduino ides. https://www.arduino.cc/en/Main/Software. 03/05/2019.

[6] Q. are fun. Quadcopter — drone flyers. http://quadcoptersarefun.com/HowToFlyAQuadcopter.html. 21/05/2019.

[7] R. C. Arkin. *Chapter 4 Behavior-based robotics.* http://citeseerx.ist.psu.edu/viewdoc/download?rep=rep1&type=pdf&doi=10.1.1.208.4463, 1998.

[8] Asci-Code. Ascii code - the extended ascii tables. https://www.ascii-code.com/. 19/05/2019.

[9] D. Atkinson. Tips on flying drones indoors. https://www.heliguy.com/blog/2018/04/06/tips-on-flying-drones-indoors/, April 2018. 21/04/2019.

[10] A. Baldi. Intel shooting star tm, así es el segundo drone de intel - islabit. https://www.islabit.com/69581/intel-shooting-star-tm.html, November 2016. 19/04/2019.

[11] C. Basics. Basics of uart communication. https://www.elprocus.com/basics-of-uart-communication-block-diagram-applications/, 2016. 18/05/2019.

[12] P. G. Bejerano. Intel vuela más de 1.200 cuadricópteros simultáneamente en un nuevo récord. https://www.todrone.com/intel-vuela-mas-1-200-cuadricopteros-simultaneamente-nuevo-record/, February 2018. 19/04/2019.

[13] Bitcraze. Crazyflie 2.x. https://store.bitcraze.io/collections/kits/products/crazyflie-2-1. 19/04/2019.

[14] P. Boyd. Why drone light shows are going to be everywhere. https://medium.com/\spacefactor\@m{}peterb0yd/why-drone-light-shows-are-going-to-be-everywhere-c1813d408a67, June 2018. 19/04/2019.

[15] L. Brown. Types of drones: Explore different types of drones. https://filmora.wondershare.com/drones/types-of-drones.html, November 2017. 17/04/2019.

[16] S. Buddies. Table of 8-bit ascii character codes. https://www.sciencebuddies.org/science-fair-projects/references/table-of-8-bit-ascii-character-codes. 19/05/2019.

[17] C. Claude. Airblock, the drone that takes itself for a transformer. http://www.makery.info/en/2016/12/19/airblock-le-drone-qui-se-prend-pour-un-transformer/, December 2016. 20/04/2019.

[18] E. Commission. Eu regulations updates. http://dronerules.eu/en/recreational/eu_regulations_updates, 2018. 17/04/2019.

[19] A. Comunication. Cámara de red axis m1033-w. https://www.axis.com/es-es/products/axis-m1033-w. 29/04/2019.

[20] T. de Drones. Nueva ley de drones en españa 2019. https://www.tierradedrones.com/nueva-ley-de-drones-en-espana, 2019. 17/04/2019.

[21] C. de la Unión Europea. Drones: reforma de la seguridad aérea de la ue. https://www.consilium.europa.eu/es/policies/drones/, November 2018. 17/04/2019.

[22] DJI. Matrice 100. https://store.dji.com/es/product/matrice-100?site=brandsite&from=buy_now_bar. 19/04/2019.

[23] DJI. Matrice 100. https://developer.dji.com/onboard-sdk/. 19/04/2019.

[24] DRONEPROSKILLS. Calibration: The complete guide. https://droneproskills.com/

drone-calibration-guide-dji-mini-drone-calibration-trimming/, November 2018. 21/04/2019.

[25] Droneu. What you need to know about flying drones indoors. https://www.thedroneu.com/blog/7-tips-flying-drones-indoors/, September 2017. 21/04/2019.

[26] EL-PRO-CUS. Basics of uart communication, block diagram, applications. https://www.elprocus.com/basics-of-uart-communication-block-diagram-applications/. 18/05/2019.

[27] J. Flynt. 21 types of drones. https://3dinsider.com/types-of-drones/, July 2017. 17/04/2019.

[28] F. for you.), how to fly a quadcopter. https://www.quadcopterflyers.com/2015/02/quadcopters-yaw-roll-and-pitch-defined.html, 2015. 21/05/2019.

[29] Github. Opencv change logs. https://github.com/opencv/opencv/wiki/ChangeLog#version410, April 2019. 02/05/2019.

[30] . D. Hover. Flight dynamics and control of vertical lift-off vehicles. https://www.quanser.com/products/3-dof-hover/. 22/04/2019.

[31] iFixit. Drone troubleshooting. https://www.ifixit.com/Wiki/Drone_Troubleshooting. 21/04/2019.

[32] A. Insights. Normativa sobre drones en españa [2019]. http://www.aerial-insights.co/blog/normativa-drones-espana/, January 2019. 17/04/2019.

[33] Intel. Intel drones light up the sky. https://www.intel.com/content/www/us/en/technology-innovation/aerial-technology-light-show.html. 19/04/2019.

[34] A. J. L. Javier Burgués, Victor Hernández and S. Marco. Smelling nano aerial vehicle for gas source localization and mapping. https://www.mdpi.com/1424-8220/19/3/47817/04/2019, January 2019. 17/04/2019.

[35] A. J. L. Javier Burgués, Victor Hernández and S. Marco. Types of drones: Multi-rotor vs fixed-wing vs single rotor vs hybrid vtol. https://www.auav.com.au/articles/drone-types/, June 2016. 17/04/2019.

[36] S. KARGU. Kargu. https://www.stm.com.tr/en/products/kargu. 20/04/2019.

[37] S. Laboratory. Cp210x usb to uart bridge vcp drivers. https://www.silabs.com/products/development-tools/software/usb-to-uart-bridge-vcp-drivers. 03/05/2019.

[38] S. Laboratory. Single-chip usb-to-uart bridge. https://www.silabs.com/documents/public/data-sheets/CP2102-9.pdf, June 2017. 18/05/2019.

[39] Leicester. Flying your drone inside a building. https://www.leicesterdrones.com/blog/flying-your-drone-inside-a-building. 21/04/2019.

[40] makeblock. Airblock. https://www.makeblock.com/steam-kits/airblock-2#Specifications. 20/04/2019.

[41] N. mechatronics. Módulo cp2102 conversor usb a ttl. https://naylampmechatronics.com/conversores-ttl/79-modulo-cp2102-conversor-usb-a-ttl.html, June 2017. 18/05/2019.

[42] P. Moya. Skydio r1, la cámara voladora 100% autónoma: así es el dron más avanzado con ia. https://omicrono.elespanol.com/2018/02/skydio-r1-camara-voladora-dron/, February 2018. 19/04/2019.

[43] E. of Mathematics. Green formulas. https://www.encyclopediaofmath.org/index.php/Green_formulas, September 2012. 18/05/2019.

[44] OpenCV. https://opencv.org/. 02/05/2019.

[45] OpenCV. About. https://opencv.org/about/. 02/05/2019.

[46] OpenCV. Color conversions. https://docs.opencv.org/3.4.3/de/d25/imgproc_color_conversions.html, Augus 2018. 11/05/2019.

[47] OpenCV. Thresholding operations using inrange. https://docs.opencv.org/3.4.3/da/d97/tutorial_threshold_inRange.html, Augus 2018. 11/05/2019.

[48] OpenCV. Contour features. https://docs.opencv.org/3.4/dd/d49/tutorial_py_contour_features.html, May 2019. 12/05/2019.

[49] OpenCV. Eroding and dilating. https://docs.opencv.org/2.4/doc/tutorials/imgproc/erosion_dilatation/erosion_dilatation.html#eroding-and-dilating, May 2019. 11/05/2019.

[50] OpenCV. Image filtering. https://docs.opencv.org/2.4/modules/imgproc/doc/filtering.html?highlight=erode#voiddilate(InputArraysrc,OutputArraydst,InputArraykernel,Pointanchor,intiterations,intborderType,constScalar&borderValue), May 2019. 11/05/2019.

[51] OpenCV. Mat - the basic image container. https://docs.opencv.org/2.4/doc/tutorials/core/mat_the_basic_image_container/mat_the_basic_image_container.htmll, May 2019. 11/05/2019.

[52] OpenCV. Structural analysis and shape descriptors. https://docs.opencv.org/2.4/modules/imgproc/doc/structural_analysis_and_shape_descriptors.html?highlight=findcontours#findcontours, May 2019. 12/05/2019.

[53] OpenCV. Contours : Getting started. https://docs.opencv.org/3.3.1/d4/d73/tutorial_py_contours_begin.html, October 2017. 12/05/2019.

[54] Parrot. Parrot mambo code. https://www.parrot.com/us/drones/parrot-mambo-code#parrot-mambo-code-details. 20/04/2019.

[55] Robolink. Codrone link. https://www.robolink.com/wp-content/uploads/2016/07/CoDrone-LINK_05_28-1.pdf. 20/05/2019.

[56] Robolink. Codrone lite. https://shop.robolink.com/collections/frontpage/products/codrone-lite. 20/04/2019.

[57] Robolink. Codrone pro. https://shop.robolink.com/collections/frontpage/products/codrone. 20/04/2019.

[58] robolink. Functions guide. https://www.robolink.com/codrone-functions-guide/. 03/05/2019.

[59] Robolink. Step 2: Connecting the codrone to snap! https://www.robolink.com/rokitbrick-connecting/, Agust 2016. 01/05/2019.

[60] Robolink. Codrone documentation. https://www.robolink.com/codrone-documentation/, April 2016. 01/05/2019.

[61] Robolink. Learn more about the smart inventor board. https://www.robolink.com/learn-smart-inventor-board/, January 2018. 01/05/2019.

[62] M. Rouse. Drone (unmanned aerial vehicle, uav). https://internetofthingsagenda.techtarget.com/definition/drone, October 2018. 16/04/2019.

[63] A. M. Sandoval. 2019, el año de los drones. http://infodron.es/id/2019/01/14/opinion-drones.php, January 2019. 17/04/2019.

[64] ST. Build the drone seen flying at maker faire rome. https://www.st.com/content/st_com/en/campaigns/build-the-drone-seen-flying-at-maker-faire-rome.html. 20/04/2019.

[65] ST. Steval-fcu001v1 - flight controller unit evaluation board for toy drones – stmicroelectronics. https://www.st.com/en/evaluation-tools/steval-fcu001v1.html#. 20/04/2019.

[66] ST. Build and fly your own drone with st's first controller unit. https://www.st.com/content/st_com/en/campaigns/build-the-drone-seen-flying-at-maker-faire-rome.html, November 2017. 20/04/2019.

[67] Stemfinity. Codrone pro. https://www.stemfinity.com/CoDrone. 01/05/2019.

[68] STM. Autonomous drone systems. https://www.stm.com.tr/en/business-areas/technology/competencies/autonomous-drone-systems. 20/04/2019.

[69] E. study. Application of lifecycle assessment to nanoscale technology: Lithium-ion batteries for electric vehicles. https://www.epa.gov/sites/production/files/2014-01/documents/lithium_batteries_lca.pdf, April 2013. 28/05/2019.

[70] Sur. La ue lanza la primera normativa europea para el vuelo de drones. https://www.diariosur.es/tecnologia/gadgets/drones-normativa-ue-20180626133831-ntrc.html, June 2018. 17/04/2019.

[71] E. Technology. How do lithium batteries affect the environment? https://blacktieaerial.com/the-physics-of-quadcopter-flight/, September 2018. 28/05/2019.

[72] S. TOGAN. Togan. https://www.stm.com.tr/en/products/togan. 20/04/2019.

[73] TopCon. Intel falcon 8+ drone – topcon edition high performance, down-to-earth simplicity. https://www.topconpositioning.com/aerial-mapping-mass-data-collection/aerial-mapping/intel-falcon-8-plus-drone-topcon-edition. 19/04/2019.

[74] uavcoach. Spain drone regulations. https://www.diariosur.es/tecnologia/gadgets/drones-normativa-ue-20180626133831-ntrc.html, 2019. 17/04/2019.

[75] UAVSYSTEMS. Tarot 650 ready to fly. https://www.uavsystemsinternational.com/best-alternative-to-dji-drones/. 19/04/2019.

[76] M. Wahde. *Chapter 3 Behavior-based robotics*. http://www.am.chalmers.se/~wolff/AA/Chapter3.pdf, 2007.

[77] N. Webcams. Axis m1033-w indoor wireless ip camera with integrated audio, pir, white-light led, wps support. https://www.networkwebcams.co.uk/product_info.php?products_id=1745. 29/04/2019.

[78] N. Webcams. Axis m1033-w indoor wireless ip camera with pir, led illumination, 2-way audio and edge recording. https://www.networkwebcams.com/axis-m1033-w-0521-002. 29/04/2019.