Autopilot and Ground Control Station for UAV

Duarte Lopes Figueiredo

Thesis to obtain the Master of Science Degree in

Aerospace Engineering

Supervisor: Professor André Calado Marta

Examination Committee

Chairperson:	Professor João Manuel Lage de Miranda Lemos
Supervisor:	Professor André Calado Marta
Member of the Committee:	Professor Agostinho Rui Alves da Fonseca

November 2014

Dedicated to my nieces Joana e Sofia.

Acknowledgments

I would like to express my gratitude to Professor André Marta for his guidance and availability throughout this work. I also want to thank my family for the support given during my Masters degree. Finally, I want to thank my colleagues and friends for their friendship which was very important during these years.

Resumo

O sector das Aeronaves Não Tripuladas tem vindo a tornar-se um dos sectores da indústria aeroespacial em maior crescimento. Cada vez mais aplicações civis beneficiam da utilização destas aeronaves. O trabalho desenvolvido neste tese refere-se ao teste e implementação de um autopiloto e de uma estação de controlo para uma Aeronave Não Tripulada que se encontra em fase de desenvolvimento num projeto financiado pelo LAETA (Laboratório Associado de Energia, Transportes e Aeronáutica). Pretende-se também implementar um sistema de *Remote Person View* em conjunto com o autopiloto.

O autopiloto escolhido para ser implementado na aeronave é o *Ardupilot* APM 2.6. Trata-se de um autopiloto *open source* o que permite reduzir os custos do projecto, mas proporcionando performances semelhantes a autopilotos comerciais. Este autopiloto pode ser configurado para ser utilizado em diferentes tipos de veículos. Para a estação de controlo terrestre é utilizado o software *Mission Planner*.

De forma a testar o autopiloto em segurança, este foi primeiro instalado e testado num rover. Neste teste verificou-se o correto funcionamento tanto do RPV assim como dos vários modos que o autopiloto proporciona. De seguida, o sistema foi montado na aeronave. Realizaram-se vários testes em voo e verificou-se que o sistema funcionou corretamente mesmo usando os parâmetros padrão do autopiloto.

Este trabalho permitiu implementar um autopiloto de baixo custo mas que permite performances de boa qualidade. Foi possível verificar a sua flexibilidade ao ser possível implementá-lo em várias plataformas. O autopiloto está preparado para ser instalado no UAV que se encontra a ser desenvolvido.

Palavras-chave: Voo Autónomo, Veículos Remotamente Pilotados, Cockpit Artificial, Estação de Controlo de Voo, Telemetria , Rover

Abstract

The Unmanned Air Vehicle sector has become the most dynamic growth sector in the aerospace industry. There are more and more civillian applications that greatly benefit from the use of these aircrafts. The objective of this work is to test and implement an autopilot and a ground control station for an UAV that is being sponsored by LAETA (*Laboratório Associado de Energia, Transportes e Aeronáutica*). It is also expected to implement a Remote Person View system along with the autopilot.

The autopilot that was chosen to be implemented is the Ardupilot APM 2.6, which is an open source, low cost autopilot. This will help reduce the total cost of the project, while also offering similar performances to commercial autopilots. This autopilot can be used in diferent types of vehicles. For the ground control station it is used the Mission Planner.

In order to safely test the autopilot without the risk of damaging any component it was first installed and tested on a rover. In this test, the autopilot and the RPV system worked correctly. Next, the system was installed on the UAV. Several flight tests were conducted and it was verified that the autopilot was working properly even with the default parameter settings.

This work allowed the implementation of a low cost autopilot that has good quality performances. This autopilot proved its flexibility when being implemented in different platforms. This autopilot is ready to be used in the UAV that is being developed.

Keywords: Autonomous Flight, Remotely Piloted Vehicle, Artificial Cockpit, Flight Control, Telemetry, Rover

Contents

List of Tables

List of Figures

Glossary

ACS	Airborne Control Station
AFCS	Automatic Flight Control System
AHRS	Attitude and Heading Reference System
FAA	Federal Aviation Administration
FEUP	Faculdade de Engenharia da Universidade do Porto
GCS	Ground Control Station
GLONASS	GLObal NAvigation Satellite System
GNSS	Global Navigation Satellite System
GPS	Global Positioning System
HALE	High Altitude Long Endurance
HUD	Head Up Display
INS	Inertial Navigational Systems
ISR	Intelligence, surveillance and reconnaissence
IST	Instituto Superior Técnico
LAETA	Laboratório Associado de Energia, Transportes e Aeronáutica
LEEUAV	Long Endurance Electric Unmanned Air Vehicle
MALE	Medium Altitude Long Endurance
MAV	Micro Air Vehicles
MUAV	Micro UAV
NAV	Nano Air Vehicles
OSD	On Screen Display
RC	Radio Controlled
RPH	Remote Piloted Helicopter
RPV	Remote-person-view
RX	Receiver
SCS	Ship Control Station
TUAV	Tactical UAV

TX	Transmitter
UAS	Unmanned Aircraft System
UAV	Unmanned Air Vehicle
UBI	Universidade da Beira Interior
UCAV	Unmanned Combat Aerial Vehicle
VTOL	Vertical Take-Off and Landing
VTUAV	Vertial Take-Off UAV

Chapter 1

Introduction

1.1 Motivation

The Unmanned Aircraft Vehicle (UAV) sector has become the most dynamic growth sector in the aerospace industry. According to a report entitled ”Unmanned Aerial Vehicle (UAV) Market (2013-2018)” by an independent information provider, the total global UAV market is expected to reach over $8,000 million by 2018 [1].

Due to its potential in replacing human pilots for performing high risk missions and by being a cheaper option, they have been widely used since the World War II [2] with their development being driven by military applications. However, there are several civil applications that can take advantage of UAV capabilities, such as monitoring crops, wildlife, forests fires and traffic, as well as remote area delivery of medicine, aerial news and photography, TV and movie production, among others.

Although not particular prohibitive for the military sector, cost, operational complexity and size are crucial elements for civilian usage. For these reasons, the use of civilian UAVs has not been yet developed to its full extent but with new technological advancements and with the perception of their potential applications, there will be a significant growth in this market.

In May of 2013, there were about 4,000 UAVs operating worldwide [3], with the majority being small ISR (intelligence, surveillance and reconnaissence) platforms and only a small part being of civilian application, mostly in agriculture. Although this small use of civilian UAVs, FAA (*Federal Aviation Administration*) forecasts that, within five years, there will be 7,500 commercial UAVs flying in the U.S. airspace alone [4].

With the growing number of civilian UAVs in the foreseeable future, there will be more and more applications that will take advantage of the numerous UAVs capabilities. There will also be cheaper technologies, that will allow low budget UAVs to have a similar performance to more expensive vehicles. This work aims to implement and analyse the performance that can be achieved with an open source autopilot that is cheaper than off-the-shelf autopilots used in modern UAVs.

1.2 Background

This work is a part of a project of a long endurance electric unmanned aircraft vehicle that is being developed in a collaborative project by IST (*Instituto Superior Técnico*), FEUP (*Faculdade de Engenharia da Universidade do Porto*) and UBI (*Universidade da Beira Interior*) under the sponsorship of LAETA (*Laboratório Associado de Energia, Transportes e Aeronáutica*). The main objective of this thesis is to implement and test an autopilot and a Ground Control Station (GCS) for the UAV. This will allow the aircraft to be flown autonomously and to be remotely controlled, while also monitoring several flight parameters.

The main goal of the project is to develop a low cost, small footprint electric UAV shown in Fig. 1.1, that is capable of being deployed from short airfields, easy to build and maintain, and highly flexible to perform different civilian surveillance missions.

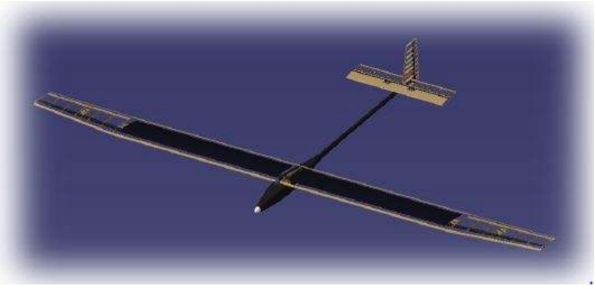


Figure 1.1: Lond endurance electric UAV [5].

The main UAV specifications include:

- Long Endurance: accomplished by using green power technologies such as an electric propulsion system with solar power. This includes the use of highly efficiency solar cells, high capacity/density batteries, efficient compact motors and appropriate long endurance aerodynamic design;

- Autonomous Flight: accomplished by equipping the UAV with autopilot navigation systems such as inertial guiding systems and GPS;

- Obstacle Avoidance: accomplished by implementing an obstacle avoidance technique that includes detection, estimation, and avoidance planning of the obstacle;

- High-strengh, Low-weight Structure: accomplished by using composite materials, with fuselage/wing critical areas designed for good impact resistance on landing, using easy to manufacture techniques;

- Multiple Mission: accomplished by designing a sufficiently large payload range capability and developing upgradable modular avionics, to enable an easy software upload and/or hardware swap to meet the selected mission requirements.

To achieve the requirements and specifications of the UAV, there are several different tasks that need to be adressed in this project. These are the following:

1. Conceptual Design - This is the most important task of the project where several different configurations will be evaluated to meet or exceed the mission requirements in terms of endurance, size and cost. By the end of this task, one or two possible configurations will be selected and will form the basis for a further refined design in the following tasks.

2. Propulsion System - In this task several different electric propulsion system configurations will be evaluated in terms of performance, overall weight and cost. The selection of solar panels will be followed by the selection of possible brushless motors. A secondary energy source will be considered in the form of high-density rechargeable batteries. It will also be evaluated a hybrid alternative in the conceptual design besides the all-electric system. By the end of this task, the propulsion system configuration will be selected and the auxiliary available power for electronics estimated.

3. Aerodynamic Design - Using high-fidelity CFD analysis, the aerodynamic design will define the wing geometry keeping in mind the solar panel dimensions. The fuselage and tail will also be modeled.

4. Noise Prediction - It is of the utmost importance to know the noise signature of the UAV in the vicinity of urban areas. The mechanisms associated with wing and airframe noise generation will be adressed in this task.

5. Structural Design and Aeroelasticity Analysis - The airframe will be designed in this task, where different wing structural solutions will be evaluated using high-fidelity CSM. It will be considered several different materials in order to meet the goal of achieving the lightest and strong enough structure, while also keeping the manufacturing cost reasonable. An aeroelasticity analysis will be performed to guarantee that the UAV will fly outside the flutter region.

6. Design for Manufacturing - In parallel with task 5, a manufacturing feasability and integration study will be undertaken in order to ensure the manufacturability of the proposed designs.

7. Stability and Control - During this task, the control surfaces will be designed to provide enough stability and control authority to the aircraft. It will be used empirical data and results will be tunned in wind tunnel testing. The data gathered from wind tunnel testing will be used to develop the UAV controller.

8. Multidisciplinary Design Optimization - At this point, all the necessary analysis tools for the propulsion, aerodynamics, structures and controls are in place. As such, it is possible to couple all these into a multidisciplinary optimization framework to refine the aircraft design.

9. Communication and Electronics - In this task the communications and electronic systems will be designed. There are several goals for this task: design and implement the autopilot hardware and software, make the aircraft systems capable of flight logging and possibly telemetry to a ground station, and install all the sensors and actuators in the airframe. It will be performed and aerothermodynamic analysis for the management of thermal loads from the internal avionics to guarantee

efficient cooling in the expected tight bay. It will also be installed telemetry equipment to monitor the aeroelasticity behavior.

10. Manufacturing - In this task the construction of the UAV will be accomplished. The goal is to build the airframe according to the detailed structural design, using advanced model building techniques. A total of two models will be built, a first generation prototype and a second generation prototype.

11. Flight Testing - The full-scale prototypes testing will include systems checks on ground, wind tunnel tests to access aerodynamic performance, static thrust under varying solar conditions and, finally, flight tests. The first aircraft will be operated under radio controlled mode, which allows for throughout checks of the solar powered propulsion system. The second aircraft will be used to test the overall design refinement and also the autopilot hardware and software.

In Figure 1.2, it is shown a representation of the several different tasks that are required for the completion of the project of the UAV. This thesis work is part of the Communcations and Electronics area and the main objective will be to implement and test the autopilot hardware and software. In this process, it will also be implemented the Ground Control Station of the UAV which is an important requirement for the operation of the vehicle. The correct implementation of the autopilot is a crucial part of the project in order to fulfill the autonomous flight specification.

1.3 Unmanned Air Vehicles

Unmanned Air Vehicles, or UAVs, are air vehicles capable of operating without any internal pilot, being remotely controled by a Ground Control Station (GCS). The system that comprises all the required elements, personal and network to control and command the UAV is known as a Unmanned Aircraft System (UAS) [6]. They were first developed to support military operations, with focus on reconnaissence and attacking ground targets, but, nowadays, the importance of using UAVs for civilian applications is growing.

1.3.1 UAV Categories

There are several different types of UAVs with different characteristics such as maximum ceiling, endurance, size or mission purpose. Based on these characteristics, Unmanned Aircraft Systems can be categorized as follows, [7]:

- *High Altitude Long Endurance (HALE)*. With a ceiling of over 15000 m of altitude and over 24 hours of endurance, they are used to carry out extremely long-range surveillance and reconnaissence missions, and are increasingly being armed. Usually operated from fixed bases.

- *Medium Altitude Long Endurance (MALE)*. They have a ceiling that varies from 5000 m to 15000 m and have 24 hours of endurance. Usually operated in a fixed base in similar missions of HALE UAVs but with a shorter range.

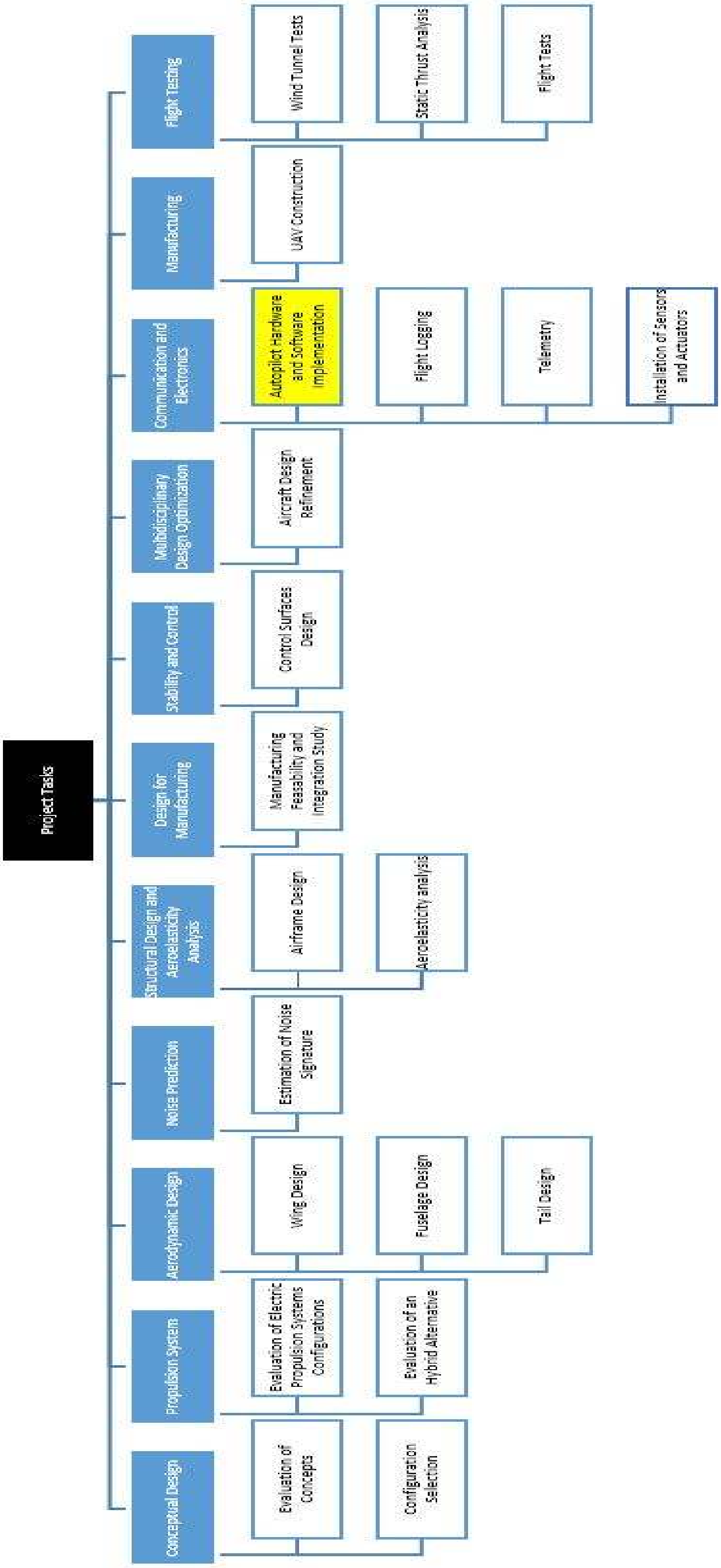

Project Tasks
Conceptual Design
Evaluation of Concepts
Configuration Selection
Propulsion System
Evaluation of Electric Propulsion Systems Configurations
Evaluation of an Hybrid Alternative
Aerodynamic Design
Wing Design
Fuselage Design
Tail Design
Noise Prediction
Estimation of Noise Signature
Structural Design and Aeroelasticity Analysis
Airframe Design
Aeroelasticity analysis
Design for Manufacturing
Manufacturing Feasability and Integration Study
Stability and Control
Control Surfaces Design
Multidisciplinary Design Optimization
Aircraft Design Refinement
Communication and Electronics
Autopilot Hardware and Software Implementation
Flight Logging
Telemetry
Installation of Sensors and Actuators
Manufacturing
UAV Construction
Flight Testing
Wind Tunnel Tests
Static Thrust Analysis
Flight Tests

Figure 1.2: LEEUAV project tasks.

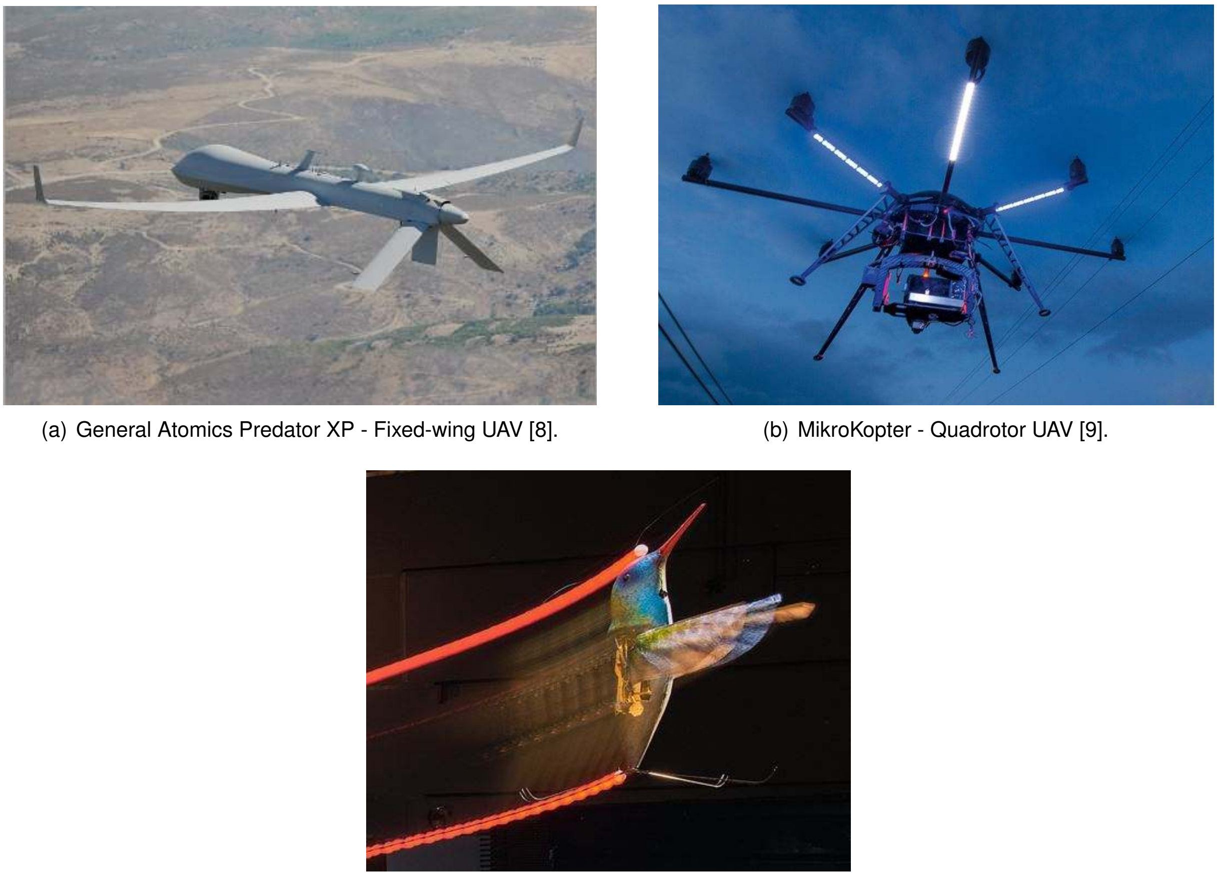
- *Medium Range or Tactical UAV (TUAV)*. They have a range between 100 km and 300 km and are smaller and operated within simpler systems than MALE and HALE UAVs.
- *Close-range UAV*. They usually operate at ranges of up to about 100 km and have both military and civil applications, such as reconnaissence, target designation, airfield security, power-line inspection, crop spraying, traffic monitoring, among others.
- *Mini UAV(MUAV)*. Are usually UAVs lighter than 20 Kg but heavier than a Micro UAV, capable of being hand-launched and operating at ranges of up to 30 km. They are used by mobile battle groups and for diverse civilian purposes.
- *Micro UAV (MAV)*. They were originally defined as being an UAV with a wing span no greater than 150 mm and are principally required to fly in an urban environment. It is required to fly slowly and to have the hability to hover and even to be able to stop and sit on a wall or post. To achieve this some MAVs have less conventional configurations such as flapping wing aircraft. They are usually hand-launched.
- *Nano Air Vehicles (NAV)*. These are very small UAVs and are proposed to be used in swarms for radar confusion and conceivability. If with technological advancements it is possible to make cameras, propulsion and control systems small enough for these UAVs, they could be used for ultra-short range surveillance.
- *Remote Piloted Helicopter (RPH) and Vertical Take-Off UAV (VTUAV)*. They are both UAVs capable of vertical take-off and landing, and also capable of hovering during a mission.
- *Unmanned Combate Air Vehicle (UCAV)*. These are UAVs which are capable of lauching weapons and even air-to-air combat.

In Figure 1.3 several different UAV topologies are represented. While fixed-wing aircrafts are the more used, there are some applications that benefit from the use of less conventional configurations. Rottary-wing UAVs, such as quadcopters, have vertical take-off and landing capabilities, the hability to hover and have great flight stability which makes them a good option for providing a stable camera platform. The flapping-wing configuration is used in MAVs allowing them for blending in with the environment.

1.3.2 UAV Roles

An aircraft, whether manned or unmanned, is designed to perform a particular task. It is up to the designer to decide if that task is better suited with a manned aircraft or not. But there are several roles [7] in which having an UAV performing will always be an advantage.

Dull Role - Military and civilian applications such as extended surveillance can be a dulling experience for the crew, with many hours spent without relief. This can lead to a loss of concentration and, consequently, a loss of effectiveness of the mission. An UAV can be a better and cheaper option for



(a) General Atomics Predator XP - Fixed-wing UAV [8].

(b) MikroKopter - Quadrotor UAV [9].

(c) Nano Hummingbird - Flapping-wing UAV [9].

Figure 1.3: Different UAV topologies.

performing these roles, while also having a better efficiency. The use of UAVs also allows for the ground operators to work in a shift pattern reducing their working hours.

Dirty Role - Also applicable to military and civilian applications, roles such as monitoring the environment for nuclear and chemical contaminations puts the aircrew unnecessarilly at risk, while detoxification an the aircraft in case of an UAV is easier. Another important role that is becoming more and more successfully conducted by UAVs is crop spraying.

Dangerous Roles - In military roles such as reconnaissence in an heavily defended area an UAV will always be better option than a manned aircraft. Due to their smaller size and greater stealh ability, an UAV will be more difficult to detect and to be strike down by the enemy defense system. Another advantage is that the UAV operator is under no personal threat and will, therefore, be able to concentrate and perform the task with a more probability of success. There are civillian tasks such as power-line inspection and forest fire control that an UAV can carry out without the risk of personnel endangerment. Another dangerous role that can be carried out by an UAV is operating under extreme weather conditions.

Covert Roles - In both military and civilian policing operations, there are roles where it is important not to alert the ”enemy” to the fact that they have been detected. UAVs can have lower detectable signatures due to their smaller size which make this role more readily achievable.

Research Roles - UAVs are being used in the aeronautical field for research and development. The use of UAVs as small replicas of manned aircraft, for test purposes, enables airbone testing to be carried out under real conditions but more cheaply and with less hazard.

Environmentally Critical Roles - More related to civilian roles, UAVs will usually cause less environmental pollution than a manned aircraft. UAVs are usually smaller, with less mass and with a lower power comsumption thus reducing the levels of pollution and noise created. Roles where the local inhabitants may object to the noise produced by low flying aircraft or roles that may cause disturbance to farm animals are better suited with an UAV.

1.3.3 UAS Structure

An UAS (Unmanned Aircraft System) comprises several sub-systems where the aircraft in only one. Figure 1.4 aims to give an overall view of the several sub-systems required to operate an UAV. There are some which are not required based on the type of the UAV.

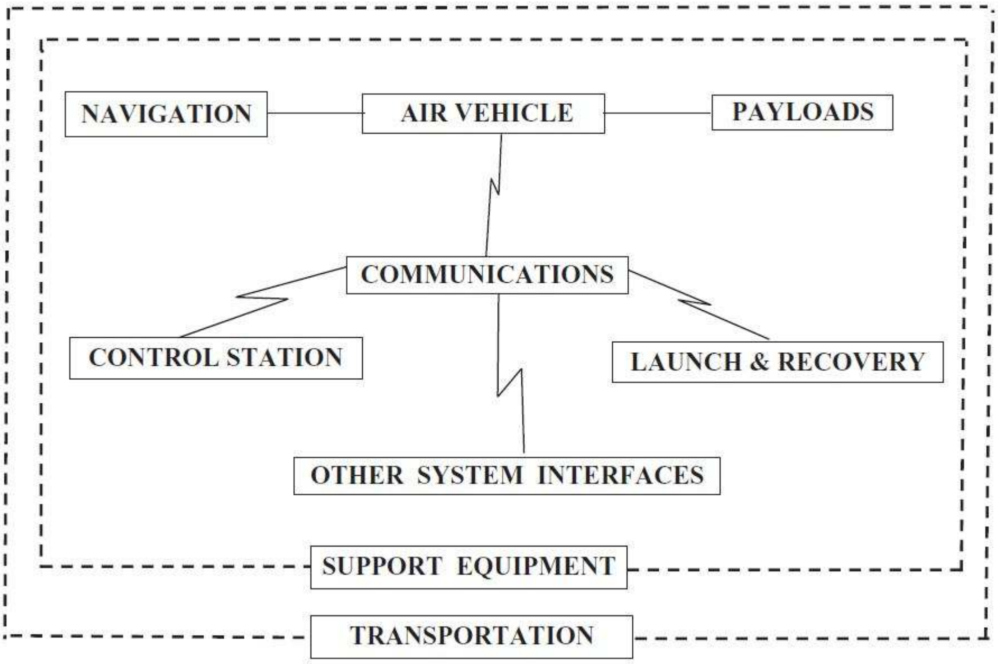


Figure 1.4: UAS systems structure [7].

Control Station

The Control Station can be based on the ground (GCS), can be carried aboard a ship (SCS) or can be carried airborne in an aircraft (ACS). It is the control-centre of operation and allows the interaction between the operator and the UAV. The operator is in contact with the UAV through the communications sub-system in order to send control commands to the UAV, monitor aircraft flight parameters and receive information or images captured by the UAV. The Control Station can also be used to control the launch and recovery of the aircraft. It will usually house systems to communicate with other external systems such as weather data aquiring.

Air Vehicle

The type and performance of the aicraft is determined by the needs of the operational mission. The main task of the aircraft is to carry the mission payload, but it also needs to carry all the required sub-systems for its operation, such as communications link, stabilization and control equipment, powerplant and fuel, airframe structure and all the required mechanisms needed for launching, carrying out the mission and being recovered.

There are other mission requirements that are needed to be taken into account when designing the aircraft. These include the operational range and endurance, which will influence the ammount of fuel that the aircraft needs to carry, and the airspeed which will determine the aircraft configuration that is best suited for the mission.

Payload

There are different types of payload that need to be carried onboard according to the UAV mission. These can vary from a simple sub-system consisting of a unstabilised video camera with a fixed lense that weights around 200 g to a high-power radar that can have a mass as high as 1000 kg. More sophisticaded UAV carry a combination of different sensors within a payload module, or within several modules. The data from these sensors is processed and integrated providing enhanced information, or information that could not be obtained from a single type of sensor.

Navigation

It is important for the operators to know, at any time, where the aircraft is. And it it also required that the aircraft knows its position in order to fly autonomously. This requires that the aircraft carries a navigational equipment onboard. The first equipment used was an Inertial Navigational System (INS), but this system required frequent position updates from the control station. Nowadays, the use of GPS (Global Positioning System) eased this problem. Available GPS systems are extremely light, compact and cheap, and they give continuous position updated, which allow the use of a very symple INS in the navigational system.

There are other sytems used for non-autonomous navigation, when the communications between the control station and the aircraft is virtually continuous, or, there is a risk that the GPS system might be blocked. These include the following:

1. *Radar Tracking* - The aircraft is fitted with a transponder that responds to a control station radar. The response of the transponder is seen on the radar display, giving information of the bearing

and range.

2. *Radio Tracking* - The radio signal carrying data from the aircraft to the control station is tracked in bearing, and the range is determined from the time that the signal takes to travel from the aicraft to the station.
3. *Direct Reckoning* - The position of the aircraft can be calculated with the computer-integration of velocity vectors and time elapsed. If the aircraft carries an onboard camera surveying the ground, the position can be confirmed by relating visual geographical features with known positions on a map.

Communications

The principal requirement of the communication system is to provide a two way link between the aircraft and the control station, usually at radio frequency. The tasks of this system are the following:

1. *Uplink* - from the Control Station to the Aircraft
   (a) Transmit flight path tasking which will be stored in the aircraft's AFCS (Automatic Flight Control System).
   (b) Transmit real-time flight control inputs to the AFCS.
   (c) Transmit control commands to the aircraft mounted payload.
   (d) Transmit positional updates to the INS when required.
2. *Downlink* - from the Aircraft to the Control Station
   (a) Transmit aircraft current position when required.
   (b) Transmit payload imagery and/or data.
   (c) Transmit aircraft data such as engine temperature or fuel state.

The complexity, weight and cost of the communications system will be determined by the range of operation of the air vehicle from the transmiting station, the sophistication required by the transmission of payload and aircraft data and by the need for security.

Launch and Recovery

For air vehicles that do not have vertical take-off capability, nor have access to a suitable runway, there will be the need for a launch equipment. Usually, it will take the form of a ramp where the aircraft is accelerated on a trolley, propelled by a launching system until it reaches an airspeed at which airborne fligh is possible. An example of a launching system is shown in Figure 1.5.

There are several forms of recovering an UAV when a normal or vertical landing is not possible. Usually a parachute, installed in the aircraft, is deployed at a suitable altitude above the landing zone. It will also be required a mean of absorbing the impact energy that, usually, comprises an airbags system or replaceable frangible material. An alternative form of recovery is the use of a large net into which the aircraft is flown and captured.

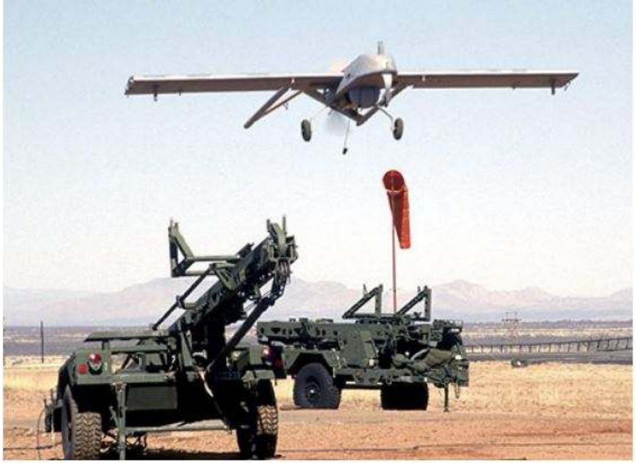


Figure 1.5: RQ-7 Shadow 200 launched from a hydraulic rail launcher [10].

Other System Interfaces

An UAV system exists in order to carry out a given task. Usually, it is not a ”stand alone” task, which means that it may require tasking from a source external to the system and report back to that or other external source. For example, in civilian operations such as fire patrol, the operators in the control station may be tasked from the Fire Brigade headquarters to move the UAV to a given location. Therefore, it is necessary to provide the control station with the necessary equipment to communicate with the external sources and to receive and send the necessary data.

Support Equipment

Support Equipment is an important element of an UAS system that is often underestimated. It ranges from the support and operating manuals, through tools and spares to special test equipment and power supplies.

Transportation

It is often necessary for an UAS system to be mobile. Therefore, it is required a transport means for all the subsystems shown in Figure 1.4. This may vary from a simple transport vehicle that carries a small vertical take-off and landing (VTOL) aircraft that needs no launch, recovery or retrieval equipment and can be operated by two crew members, to a more complex UAS system that needs a large launch ramp, which may have to be dismantled between flights, needs a large crew to operate and therefore need several transport vehicles for all the subsystems. Even UAV operated from fixed bases may have specific transport requirements.

1.4 Objectives

The main goal of this thesis is to implement and test an autonomous flight control system for the UAV described in Section 1.2. This system will be composed by an autopilot, a GCS and all the required equipment for data transmission between the autopilot and the GCS. To meet this goal, there are several tasks that need to be accomplished. The tasks in this project include:

- Market survey on available commercial autopilot solutions;
- Comparison of different technical solutions;
- Detailed design of the autonomous flight control system;
- Assembly of the fligh control system using off-the-shelf components;
- Test, characterization and tunning of the flight control system in a controlled environment;
- Field test using an available flying testbed using remote-person-view (RPV);
- Flight demonstration using waypoint satellite navigation.

1.5 Structure of the Document

This thesis consists of six chapters which are organized as follows:

In Chapter 2 it is first made a brief description about autopilots. Then, the several autopilot solutions that were researched are presented. In the final section, it is made a comparision between the different solutions.

In Chapter 3 it is presented and described the autopilot solution that is used in the following chapters. It is also described in detail the several components that comprise the autopilot, the ground control station and the RPV system.

The Chapter 4 describes how the autopilot is configured to be used in a rover. This chapter presents a detailed description of how to configure and, then, tune the autopilot to increase the rover performance. It is also presented the results achieved by implementing the autopilot on a rover. The RPV system is also analysed in this results.

The Chapter 5 describes how the autopilot must be configured to be used in the UAV. It is made a complete description of how the autopilot must be tunned to achieve the best performance. It is also presented the results achieved during the UAV flight tests.

Finally, the conclusions and recommendations are presented in Chapter 6.

Chapter 2

Autopilots in Unmanned Air Vehicles

2.1 Autopilots

An autopilot is a system that is used to automatically guide a vehicle without the assistance of an human operator. They have been widely used in air vehicles, boats, spacecrafts, missiles and even on autonomous land vehicles. There has been a significant evolution of the autopilots over the time, and they have evolved from simple autopilots, that only held altitude, to autopilots that are capable of complex operations such as landing an aircraft.

The first autopilot was developed in 1912 by Lawrence Sperry [11]. It allowed the aircraft to fly straight and level on a compass course without the need for the pilot to operate. This autopilot became known as ”George”. The picture in Figure 2.1 was taken in a competition in Paris in 1914, where Sperry raised hands demonstrates the autopilot working.

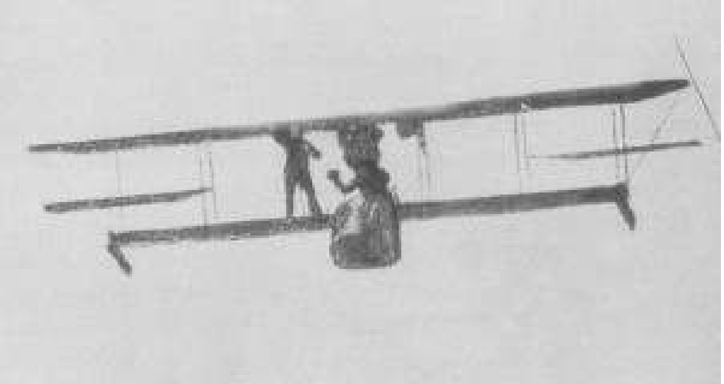


Figure 2.1: ”George” the autopilot [11].

Since Sperry's autopilot there has been a significant evolution in the autopilot's capabilities. Modern autopilots use computer software to control the vehicle and GPS for position determination. This evolution allows the autopilots to perform complex tasks such as waypoint following, and even automatic take-off and landing of some air vehicles. More recently, autopilots are being used in autonomous vehicles such as Google's driverless car shown in Figure 2.2. These vehicles have very expensive and complex control systems that require several sensors due to a variety of reasons. The most important is obstacle avoidance given that in an urban environment there are many obstacles such as other cars, pedestrians, traffic restritions, among others. They also have to be capable of consistently following a road and respect traffic signals. Airplane control systems are less complex than these systems as they do not have so many obstacles and restrictions.

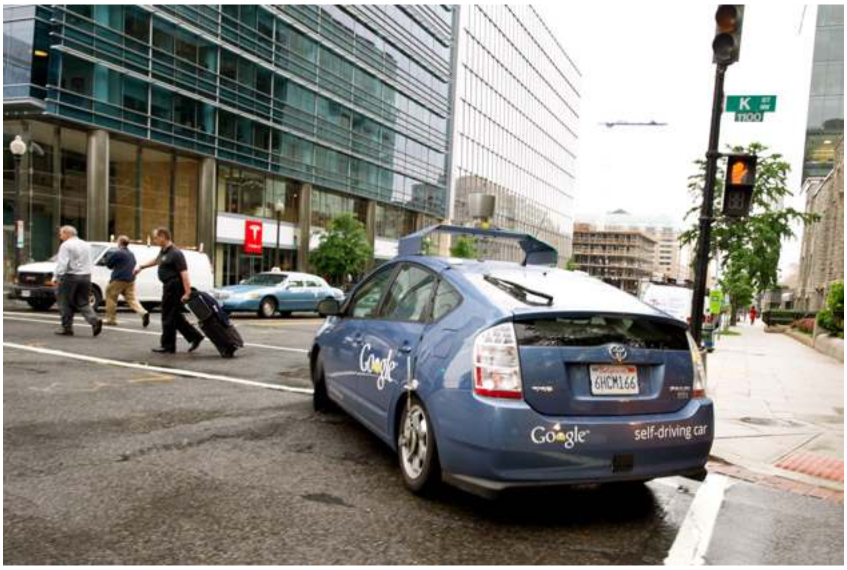


Figure 2.2: Google's driverless car [12].

An UAV autopilot has to be capable of consistently guiding the UAV through waypoints or following reference paths. It is an integrant part of any UAV flight control system. The flight control system communicates with the GCS using telemetry, receives GPS data for position update and sends out control inputs for the servo motors on the UAV.

An UAV autopilot sytem is a closed-loop control system that has two parts: the controller and the state observer. Usually, the state observer is an inertial guidance system that includes gyro, acceleration and magnetic sensors. There are other attitude determination devices such as infrared and vision based ones. The sensor measurements and GPS data is passed to a filter that generates estimates of the current position of the vehicle. These estimates are then passed to the controller that, based on the control strategy employed, will send control inputs to the actuators. A schematic of an autopilot system is shown in Figure 2.3.

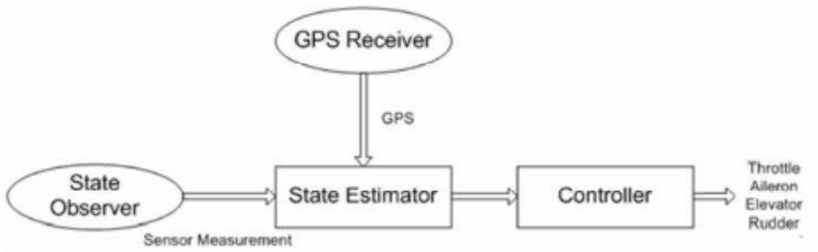


Figure 2.3: Structure of an UAV autopilot [13].

In Section 2.2, it will be presented a market study on several autopilots that could be implemented in the UAV presented in section 1.2. There are many available commercial and open source solutions. For this project it is important that the selected autopilot respects the following requirements:

- Small dimensions and weight;
- Low price;
- Waypoint following capabilities;
- Auto take-off and landing;
- Configurable.

The comparison and selection of the of the autopilot that will be implemented on the UAV will be presented in Section 2.3.

2.2 Market Study

The autopilots presented in this section were selected based on the autopilot requirements of the UAV. The commercial solutions are presented from Sections 2.2.1 to 2.2.4 and the autopilots in Sections 2.2.5 and 2.2.6 are open source solutions. The main characteristics of each autopilot can be found in their description and in Section 2.3 it is presented their technical data.

2.2.1 UAV Navigation VECTOR

VECTOR is a top of the range autonomous Flight Control unit designed for high-end target drones and UAVs, developed by *UAV Navigation*, shown in Figure 2.4. The specifications are shown in Section 2.3. It has a high processing power with dual 200 MIPS CPUs with 8 MB flash memory, combined with a Attitude and Heading Reference System (AHRS) and with an Inertial Navigational Systems (INS). It is capable of fully automatic take-off, flight plan execution and landing. This autopilot can control several different types of UAVs such as fixed wing tactical UAVs, high end subsonic drones and helicopter or multirotor platforms. At the core of this autopilot is a POLAR AHRS/INS unit that combines all the necessary vehicle dynamics sensors and algorithms for state estimation. More detailed information about VECTOR autopilot can be found in [14].

2.2.2 Micropilot MP2x28 Series

Micropilot offers a series of autopilots for small rotary-wing and fixed-wing UAVs with prices ranging from \$1,500 to \$8,000. For comparison, it is chosen the middle range autopilot MP2028^{g}, shown in Figure 2.5. The specifications are shown in Section 2.3. The main advantage of this autopilot is its small size and weight. It has GPS, 3-axis gyros/accelerometers, pressure altimeter and pressure airspeed sensors all integrated in a single circuit board. The MP2028^{g} autopilot supports altitude hold, airspeed hold and waypoint navigation. It also supports completely independent operations such as autonomous take-off,

Figure 2.4: UAV Navigation VECTOR [14].

bungee launch, hand launch and landing. More detailed information about MP2028^g autopilot can be found in [15].

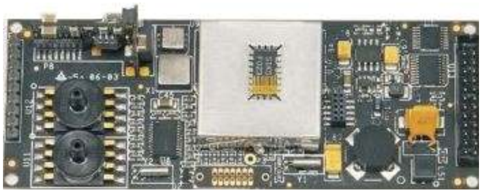


Figure 2.5: Micropilot MP2028^g [15].

2.2.3 Procerus Kestrel v2.4

Procerus Kestrel v2.4 is an autopilot specially designed for small UAVs and MAVs, weighting only 16.7 grams, shown in Figure 2.6. The specifications are shown in Section 2.3. This autopilot has a complete inertial sensor set that includes: 3-axis rate gyros and accelerometers, absolute and diferential pressure sensors for altitude and airspeed measurement and 3 temperature sensors that combined with a temperature compensation algorithm allows it to estimate the UAV attitude and the wind speed accurately. Kestrel has the built-in ability to autonomous take-off and landing, waypoint navigation, speed and altitude hold and it also supports multiple UAV operations. More detailed information about Kestrel v2.4 autopilot can be found in [16].

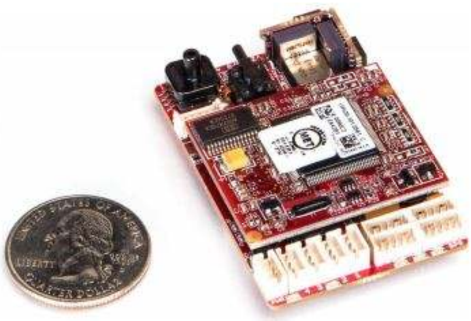


Figure 2.6: Procerus Kestrel v2.4 [16].

2.2.4 Cloud Cap Piccolo SL

Cloud Cap Piccolo SL is an autopilot for small fixed wing UAVs and VTOL applications, shown in Figure 2.7. The specifications are shown in Section 2.3. It provides a complete integrated avionics solution that includes the flight control processor, inertial sensors (3-axis accelerometers and gyros), static and dynamic pressure sensors, GPS receiver and a datalink radio. Piccolo SL supports peripherals that enhance its capabilities such as laser altimeters for a more accurate altitude information, transponders, magnetometers, RTK (Real Time Kinematic) GPS, Iridium StaComm, payload passthrough, among others. It is also possible to extend the autoland performance by using DGPS (Differential GPS). With DGPS this autopilot supports autonomous taxi, rolling take-off and stationary and moving net recovery. More detailed information about Piccolo SL autopilot can be found in [17].

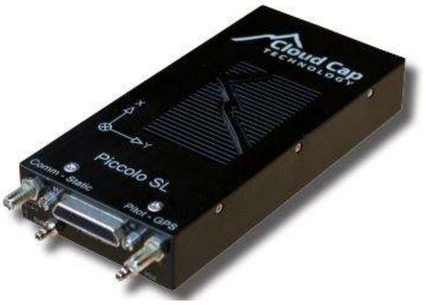


Figure 2.7: Cloud Cap Piccolo SL [17].

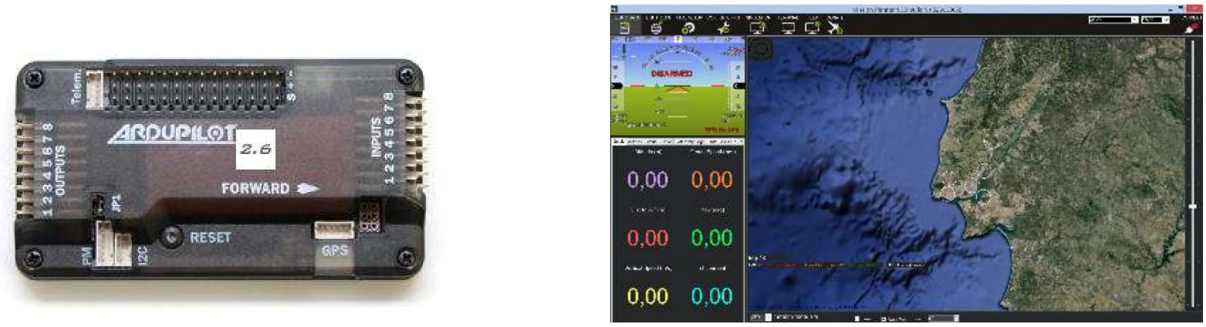
2.2.5 Ardupilot APM 2.6

The APM 2.6 is an open source autopilot system based on an Arduino platform, shown in Figure 2.8(a). This autopilot can be used in fixed-wing and rottary-wing vehicles such as helicopters and multi-rotors. The specifications are shown in Section 2.3. The Ardupilot board consists of the main processor and the Inertial Measurement Unit. It has 4 serial ports, ports for GPS, wireless telemetry, power module and external compass connection, and an USB port. This autopilot is capable of autonomous take-off and landing, waypoint navigation, two way telemetry and has a built-in hardware failsafe that allows the aircraft to return to the launch base when the radio signal is lost.

The APM 2.6 can be used with an open source ground station application such as the Mission Planner, shown in Figure 2.8(b). This software allows the user to calibrate and configure the autopilot, plan and save missions and view live flight data. More detailed information about APM 2.6 autopilot can be found in [18].

2.2.6 Openpilot

Openpilot is an open source UAV autopilot system capable of supporting fixed-wing vehicles, as well as multi-rotor and helicopters. The specifications are shown in Section 2.3. There are two hardware platforms available: the *CopterControl* and the *Revolution*, shown in 2.9(a). While the *CopterControl* board makes use of 3 axis gyros and accelerometers for stabilization, the *Revolution* has a full INS



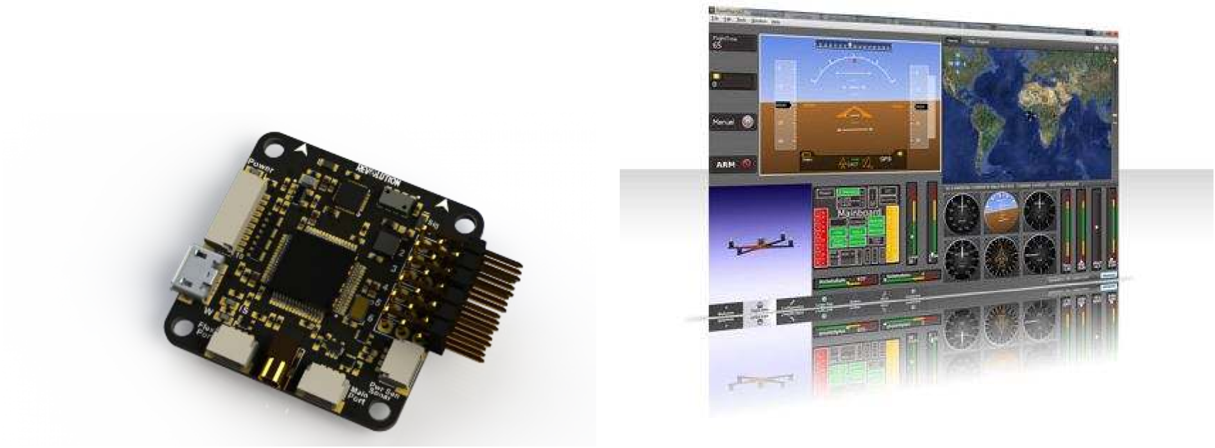
(a) Ardupilot APM 2.6 [19]

(b) Mission Planner

Figure 2.8: Ardupilot hardware and software.

(Inertial Navigation System) unit onboard. The Openpilot *Revolution* board has a built-in modem that provides a direct telemetry link between the controller and the GCS, a barometric pressure sensor, a magnetometer and it can be connected with a GPS module. This autopilot, when connected with a GPS module, is capable of waypoint navigation, position hold and automatic return to launch base.

The GCS is an open source software that allows the user to configure the hardware and to interact with the aircraft during flight. This software also provides waypoint navigation settings and telemetry functions. More detailed information about the Openpilot autopilot can be found in [20].



(a) Openpilot Revolution board

(b) Openpilot GCS

Figure 2.9: Openpilot hardware and software [20].

2.3 Comparison of Solutions

Nowadays, there are several different autopilot solutions on the market. There are sophisticated commercial autopilots but more expensive than open source solutions. Although open source autopilots can not achieve the performance of some commercial autopilots, they provide great flexibility both in the hardware and software, as the users can modify the autopilot based on their own requirements. These autopilots are also a good option for small budget projects.

2.3.1 Physical Specifications

The physical specifications of the autopilots are important as UAVs demand as fewer space, payload and power comsumption as possible. The size, weight, power consumption issues and price are presented in Table 2.1. The price is presented in US Dollars and is the price of the autopilot only. The operating temperature, required voltage input, CPU and memory are presented in Table 2.2.

	Size(cm)	Weight(g)	Power Consumption	Price
VECTOR	6.88 x 4.50 x 7.45	300	2.5W	-
MP2028^g	10.0 x 4.0 x 1.5	28	140mA@6.5V	$3500
Kestrel v2.4	5.08 x 3.5 x 1.2	16.8	500mA@(3.3 or 5V)	$5000
Piccolo SL	13.1 x 5.7 x 1.9	110	4W	-
APM 2.6	6.7 x 4.1 x 1.5	33	200mA@5V	$240
Revolution	3.6 x 3.6 x 1.2	14	-	$120

Table 2.1: Comparison of physical specifications of autopilots.

	Operating Temperature	DC in	CPU	Memory
VECTOR	-40°C to 85°C	7V to 36V	dual 200MIPS	8MB flash
MP2028^g	-	4.2V to 26V	3MIPS	-
Kestrel v2.4	-40°C to 85°C	5V to 30V	-	-
Piccolo SL	-40°C to 80°C	5V to 6V	16MHz	4Mb
APM 2.6	-	5V to 6V	16MHz	4Mb
Revolution	-	4.8V to 8.4V	210MIPS	-

Table 2.2: Comparison of physical specifications of autopilots.

The Openpilot *Revolution* and the Kestrel autopilots are the smallest and lightest solutions while, on the other hand, the VECTOR is the biggest and heaviest autopilot. This autopilot has also the better CPU and more memory than all the others autopilots. It can also be seen that the commercial solutions are much more expensive than open source solutions, as expected.

2.3.2 Sensor Ranges

The sensors information is shown in Table 2.3. It could not be found sensor information for the open source autopilots. VECTOR provides the autopilot that is capable of working at the highest airspeed and it also has a wide range of operating altitudes. Micropilot MP2028^g is the autopilot that has the highest limit of operating altitude but it has a small maximum acceleration limit of only 2g. It also has

	Altitude	Max Acceleration	Max Airspeed	Max Angular Rate
VECTOR	-2000ft to 30000ft	8g	450kt	300°/s
MP2028^g	39000ft	2g	270kt	150°/s
Kestrel v2.4	-2600ft to 22600ft	10g	252kt	300°/s
Piccolo SL	-	6g	192kt	300°/s
APM 2.6	-	-	-	-
Revolution	-	-	-	-

Table 2.3: Comparison of sensors ranges.

a lower maximum angular rate than the other commercial solutions. Kestrel has a smaller maximum altitude operation than VECTOR and lower maximum airspeed, but it is the autopilot with maximum acceleration limits. Piccolo has a reasonable maximum acceleration limits but it is the autopilot with the lowest maximum airspeed limit.

2.3.3 Autopilot Functions

The autopilots main functions are presented in Table 2.4. By analysing this table it can be seen that almost all the autopilots share the same functions. There are some exceptions such as the Openpilot *Revolution* that doesn't provide Auto Take-Off and Landing which is an important function for an UAV. Micropilot MP2028^{g} doesn't provide Multi-UAV Support which can be a drawback in some applications.

	Waypoints Navigation	Auto Take-Off and Landing	Altitude Hold	Airspeed Hold	Multi-UAV Support	Return Home
VECTOR	✓	✓	✓	✓	✓	✓
MP2028^{g}	✓(1000 points)	✓	✓	✓	x	-
Kestrel v2.4	✓	✓	✓	✓	✓	-
Piccolo SL	✓(1000 points)	✓	✓	✓	✓	-
APM 2.6	✓	✓	✓	✓	✓	✓
Revolution	✓	x	✓	✓	✓	✓

Table 2.4: Comparison of autopilot functions.

2.3.4 Hardware Specifications

In Table 2.5 is presented hardware information. All the autopilots provide GPS support (in some cases it is displayed the operating frequency of the GPS), inertial sensors and telemetry connection. Ardupilot 2.6 provides a compass connection and has a wide range of input/output ports for the external sensors and servos. Openpilot *Revolution* doesn't provide compass support.

	GPS Support	Compass	IMU	Input/Output	Telemetry Connection
VECTOR	✓	-	✓(1KHz)	42 lines (24 servos)	external datalink
MP2028^{g}	1 Hz	-	✓	8/16/24 servos	5 Hz
Kestrel v2.4	✓	-	✓	4 onboard servos+8 external servos+4 serial ports+12 digital	✓
Piccolo SL	4 Hz	-	✓	14 configurable GPIO lines	✓
APM 2.6	✓	✓	✓	4 serial+GPS+Telemetry+Power Module+External Compass+USB	✓
Revolution	✓	x	✓	6 servos+3 four-pin ports+telemetry antenna socket	✓

Table 2.5: Comparison of hardware specifications.

2.3.5 Software Specifications

In Table 2.5 is presented software information. The Open Source autopilots' software provide support for the different operating systems while Micropilot MP2028^{g} can only operate in a Windows environment. The Open Source and Vector autopilots also have the advantage of being configurable while the others do not provide this feature.

	Software/Firmware	Operating System	Configurable
VECTOR	proprietary	-	✓
MP2028^g	proprietary	Windows	x
Kestrel v2.4	proprietary	-	x
Piccolo SL	proprietary	-	x
APM 2.6	opensource	Windows/Linux/Mac	✓
Revolution	opensource	Windows/Linux/Mac	✓

Table 2.6: Comparision of software specifications.

2.3.6 Overall Evaluation and Selection

The classification of each autopilot is presented in Table 2.7. The objective of this classification is to analyse which autopilot best fullfills the requirements stated in Section 2.1. Each autopilot was classified from 1 to 6 in the following decision criteria:

- Dimensions - In order to maximize the payload capability of the UAV it is important that the autopilot has small dimensions;
- Weight - Having an autopilot with lower weight is important as it will reduce the total weight of the UAV;
- Price - The UAV project will benefit from a low cost autopilot;
- Functions - It is required that the autopilot has several functions such as waypoint following and auto take-off and landing;
- Hardware - In this criteria it will be accessed the hardware capabilites of the autopilot presented in Table 2.5;
- Software - The software capabilites of the autopilot are an important part of the autopilot. Being able to operate the software in different operating systems and being able to configure the autopilot will provide the UAV with greater flexibility.

	Dimensions	Weight	Price	Functions	Hardware	Software	Total
VECTOR	1	1	1	6	6	5	3.33
MP2028^g	3	5	2	5	5	4	4.00
Kestrel v2.4	5	6	1	5	5	4	4.00
Piccolo SL	2	2	3	5	5	4	3.50
APM 2.6	5	5	6	6	6	6	**5.67**
Revolution	6	6	6	3	5	6	5.33

Table 2.7: Evaluation of researched autopilot solutions.

The value given in each category takes into account the comparision between the several autopilots. This means that, for example, the price of the open source solutions when compared to the commercial solutions will achieve a much better classification given that both the APM and Revolution have a price that is more than ten times lower than the price of the cheapest commercial autopilot.

Following this decision process, the Ardupilot APM 2.6 is the autopilot with better classification and will be the autopilot used in this work. In fact, the open source solutions achieve better marks than the commercial options given that the project requirements clearly benefit from a cheap and accessible autopilot. These autopilots also benefit from sharing almost all the autopilot functions that the commercial autopilots offer.

Chapter 3

System Design

This chapter aims to give an overview of how the sytem components are connected and configured through the Mission Planner. In Section 3.1, it will be made an overview of the Ardupilot Board. In Section 3.2, it will be explained how the Mission Planner is installed, it will be made a description of the several features of this software, and finally, it will be described how the firmware is loaded into the Ardupilot board. Section 3.3 explains how the Ardupilot Board can be power supplied. From Section 3.4 to 3.8, it is explained how the remaining components are connected to the Ardupilot board and configured in the Mission Planner.

3.1 Ardupilot Board

The Ardupilot 2.6 board has several ports as shown in Figure 3.1. It has an USB port that allows the user to connect the autopilot directly to a computer to upload the firmware. The wireless telemetry port can be either connected to a telemetry radio or to an OSD (On Screen Display) that allows the transmission to the ground station of both telemetry and video. The GPS can be connected to the old style GPS port or to the new style GPS port, depending on the GPS that is being used. The external compass is connect using the External I2C port. This autopilot can be powered by using a Power Module that is connected using the Power port.

3.2 Mission Planner

The APM platform provides an open source software that is used as the ground control station. This software is the Mission Planner and the latest release can be downloaded from `http://ardupilot.com/downloads/?did=82`.

After downloading the software the installation is straightfoward. First, it is required to open the installer file and select **Run** to run the installation utility. Then, simply following the instructions will complete the setup process. The installation utility will install any necessary software drivers.

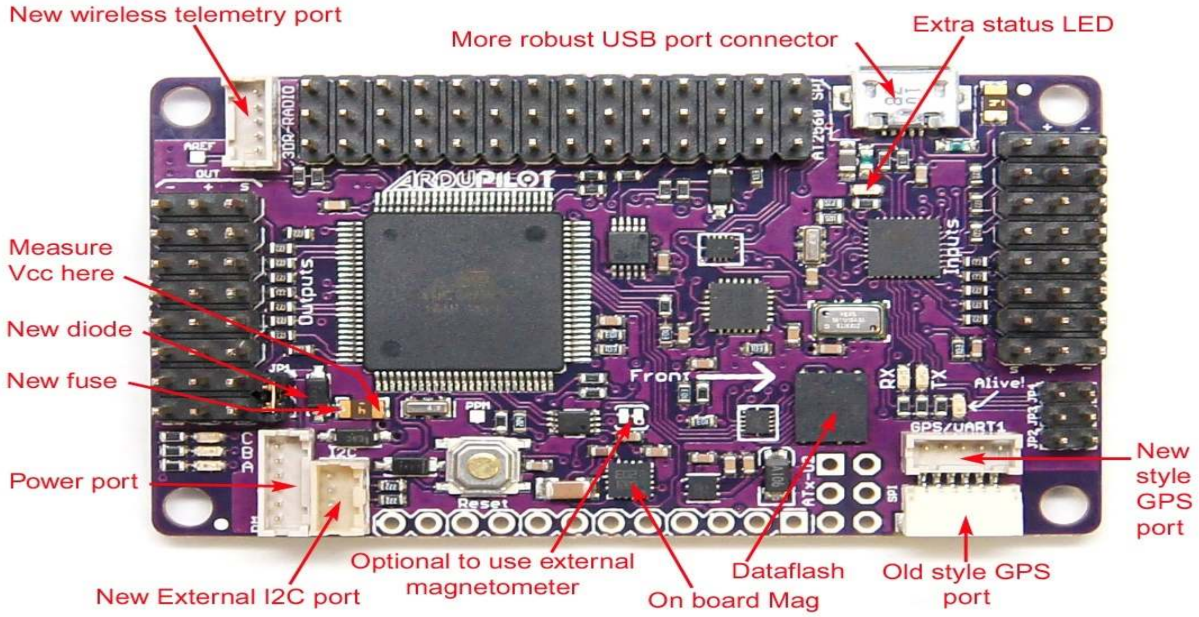


Figure 3.1: Ardupilot board overview [18].

3.2.1 Mission Planner Features

The Mission Planner version that was used is the version 1.3.5. This software has several tabs with different functions. In Figure3.2, it is shown the *Flight Data* tab which is where it is possible to view live data that is being trasmitted through telemetry from the autopilot to the ground station. It has an artificial horizon in the upper-left corner of the screen that shows the orientation of the vehicle, and other important information such as the groundspeed or altitude. The information displayed in the artificial horizon can be configured by right-clicking on it and selecting **Items**.

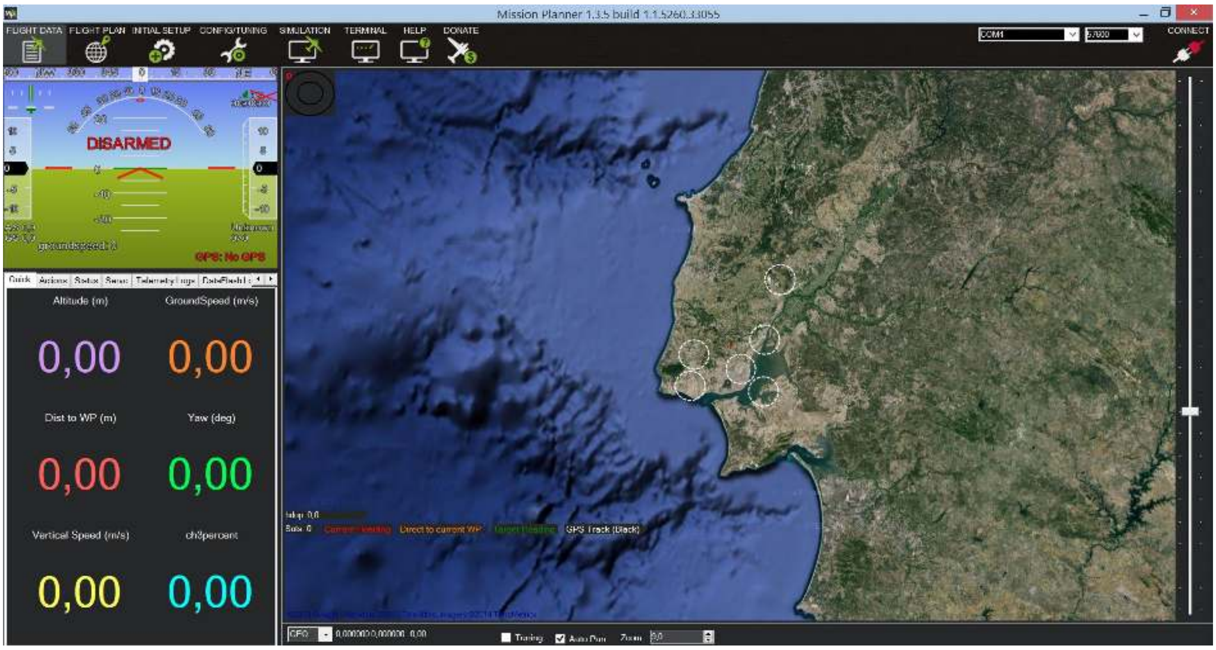
Bellow the artificial horizon there are several tabs with different functions. In *Quick* it can also be seen the same information of the artificial horizon. By double-left clicking on any field, it can be configured the information that is going to be displayed in that field. In the *Actions* tab, the user can give commands to the autopilot, and in the *Status* tab it is displayed the values of the several parameters that are being transmitted by the autopilot.

The current position of the vehicle will be displayed in the map when the autopilot has acquired a GPS lock. By right-clicking on the map, it is also possible to give commands to the vehicle.

An important feature of the Mission Planner is that it allows the creation of missions by setting waypoints for the vehicle to follow. This is done in the *Flight Plan* that is shown in Figure 3.3.

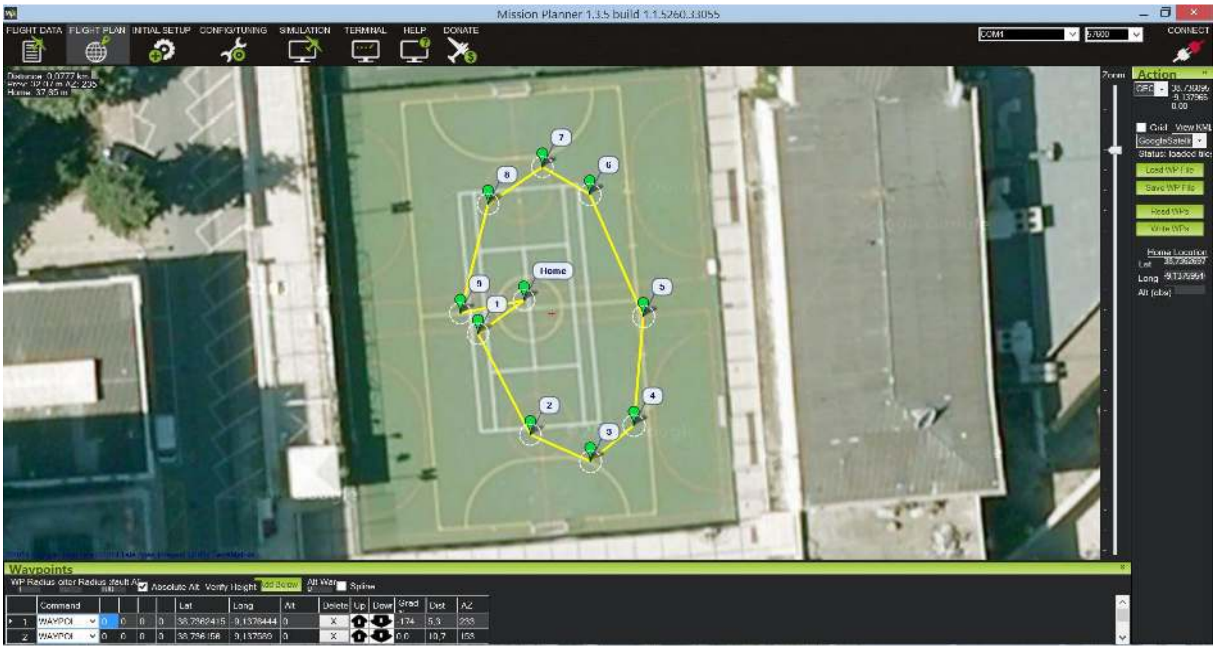
The waypoints can be manually created by right clicking on the map and selecting **Insert Wp**. The waypoints list will appear in the pannel bellow the map. This list can be saved to a file by clicking on **Save WP File** on the right pannel and can be sent to the autopilot, when connected to the Mission Planner, by clicking on **Write WPs**.

The waypoints can also be recorded by the autopilot and then loaded into the Mission Planner. In this case, when the autopilot has recorded the desired waypoints clicking on **Read WPs** will load the waypoints list into the Mission Planner.



Altitude (m) 0,00
GroundSpeed (m/s) 0,00
Dist to WP (m) 0,00
Yaw (deg) 0,00
Vertical Speed (m/s) 0,00
ch3percent 0,00

Figure 3.2: Mission Planner - flight data.



WP Radius oiter Radius efault Al ☑ Absolute Alt Verify Height Add Below Alt War ☐ Spline

	Command					Lat	Long	Alt	Delete	Up	Dowr	Grad	Dist	AZ
1	WAYPOI	0	0	0	0	38,7362415	-9,1378444	0	X			-174	5,3	233
2	WAYPOI	0	0	0	0	38,736158	-9,137589	0	X			0,0	10,7	153

Figure 3.3: Mission Planner - flight plan.

The *Initial Setup* tab is used for the installation of the firmware of the autopilot and configuration of the several components of the Autopilot such as the GPS, Power Module, Sensors, Telemetry Radios and the OSD. The detailed explanation about these components will be made in the following sections.

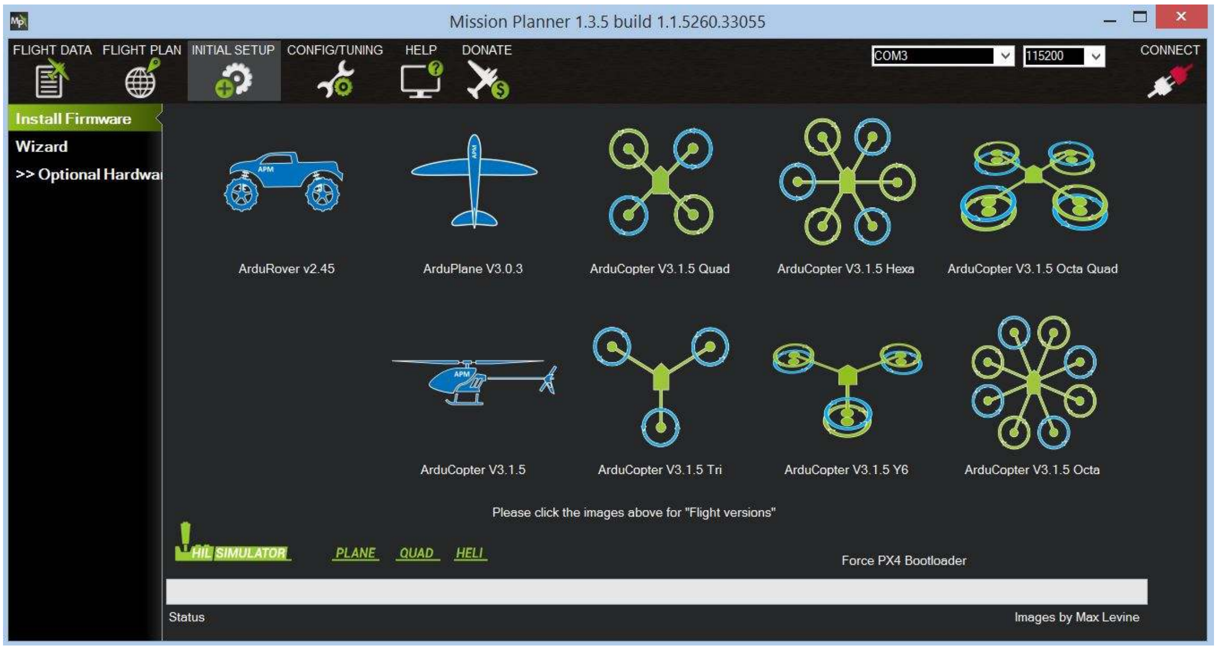
In the *Config and Tuning* tab the several parameters of the Ardupilot can be configured to tune the performance of the autopilot. This will be accessed in Chapter 4.

3.2.2 Loading Firmware into Ardupilot 2.6

After installing the Mission Planner on the ground station computer, the Ardupilot board can be connected to the computer using a micro USB connector for the Ardupilot micro USB port and a direct USB port on the computer. If the operating system does not automaticaly installs the Ardupilot board drivers, they can be manually downloaded from `http://ardupilot.com/downloads/?did=19`.

After connecting the Ardupilot board to the computer, in the Mission Planner upper-right corner drop down menus, the correct port where the Ardupilot is connected needs to be selected. This will appear as **Arduino Mega 2560**. The Baud rate has to be set to **115200**.

Depending on the vehicle configuration, there are several options that can be selected for the Ardupilot firmware. This is shown in Figure 3.4. The installation of the desired configuration firmware is done by clicking on the corresponding icon. Then, the Mission Planner will prompt to confirm the firmware download and installation.



Please click the images above for "Flight versions"

Figure 3.4: Firmware selection.

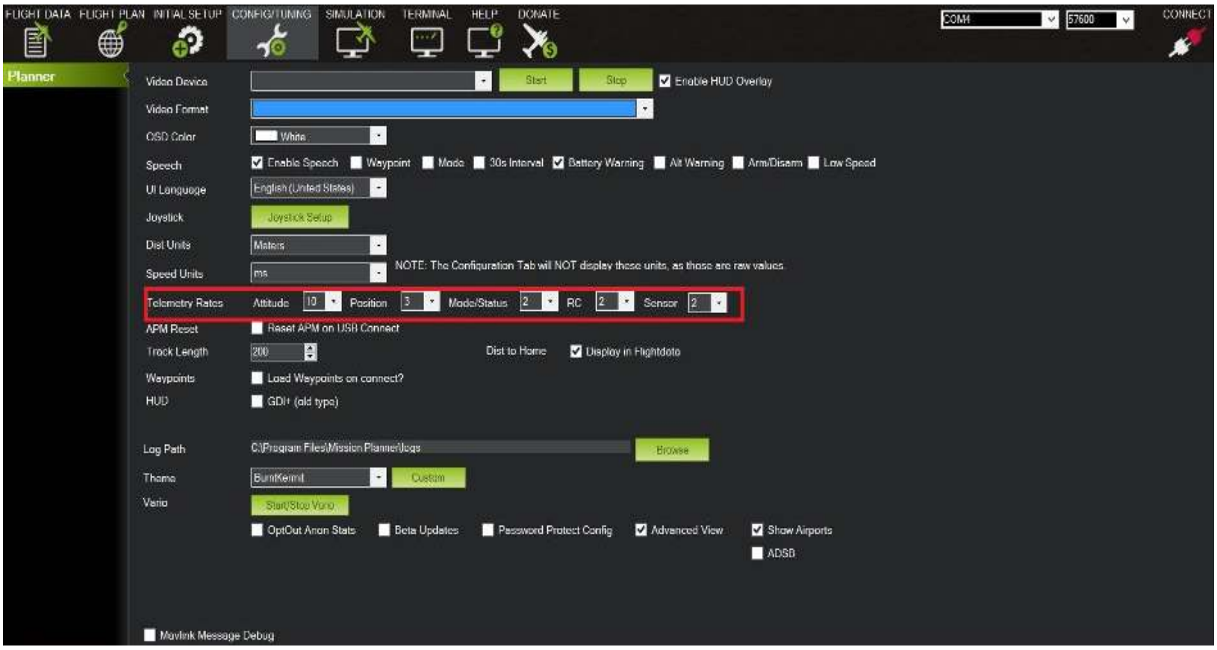
After installing the desired firmware on Ardupilot, the MavLink parameters can be loaded by clicking **Connect** on the upper-right corner.

3.2.3 Telemetry Logs

The flight data can be recorded in two different ways:

- Dataflash logs - These use the autopilot onboard flash memory and can be downloaded after use;
- Telemetry logs (tlogs) - These are automatically recorded by the Mission Planner when the autopilot is connected using the telemetry link. The tlog file will be saved in the ”logs” subfolder in the Mission Planner installation folder.

Figure 3.5 shows how the desired rate at which data sent by the autopilot to the ground station can be controlled in the Mission Planner. The values that were used are also presented in Table 3.1.



Video Device ☑ Enable HUD Overlay
Video Format
OSD Color White
Speech ☑ Enable Speech □ Waypoint □ Mode □ 30s Interval ☑ Battery Warning □ Alt Warning □ Arm/Disarm □ Low Speed
UI Language English (United States)
Joystick
Dist Units Meters
Speed Units ms NOTE: The Configuration Tab will NOT display these units, as those are raw values
Telemetry Rates Attitude 10 Position 3 Mode/Status 2 RC 2 Sensor 2
APM Reset □ Reset APM on USB Connect
Track Length 200 Dist to Home ☑ Display in Flightdata
Waypoints □ Load Waypoints on connect?
HUD □ GDI+ (old type)
Log Path C:\Program Files\Mission Planner\logs
Theme BurntKermit
Vario
□ OptOut Anon Stats □ Beta Updates □ Password Protect Config ☑ Advanced View ☑ Show Airports
□ ADSB
□ Mavlink Message Debug

Figure 3.5: Logs datarate.

Parameter	Telemetry Rate (Hz)
Altitude	10
Position	3
Flight Mode	2
RC	2
Sensor	2

Table 3.1: Telemetry rates.

The Mission Planner can also be used to play back missions from tlog files. This can be done in the Flight Data screen and selecting the Telemetry Logs tab. Then the tlog can be loaded by clicking on **Load**, as shown in Figure 3.6. The slider can be used to jump to any point in time of the log file. When the log is playing, the HUD will move and show all the available information as it was shown during the real flight. By clicking on the tuning checkbox, at the bottom of the screen, any individual data value can be displayed in a graphic.

These logs can also be used to create 3D profiles of the flight paths. This is achieved by clicking on the **tlog to Kml or Graph** button. It will open a new menu, shown in Figure 3.7, where by clicking on **Create KML+GPX** and selecting the desired log will create a kmz and a kml file on the log folder. This can be opened in Google Earth to interactively view the 3D flight path.

In this menu there are also several other options. Selecting **Extract Params** will cause a param file to be created along the tlog file. This file contains a full list of parameters along with their values during flight. **Extract WPs** will create a text file containing any missions uploaded to APM. This file can be used in the Mission Planner to load the mission.

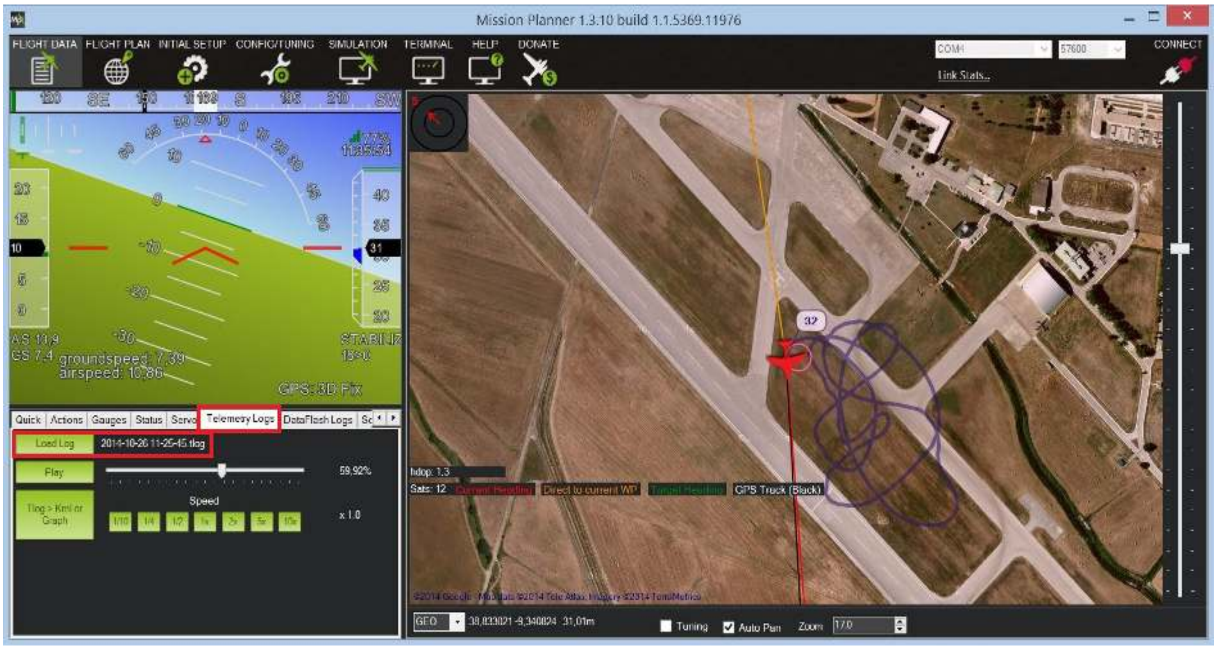


Figure 3.6: Logs playback.

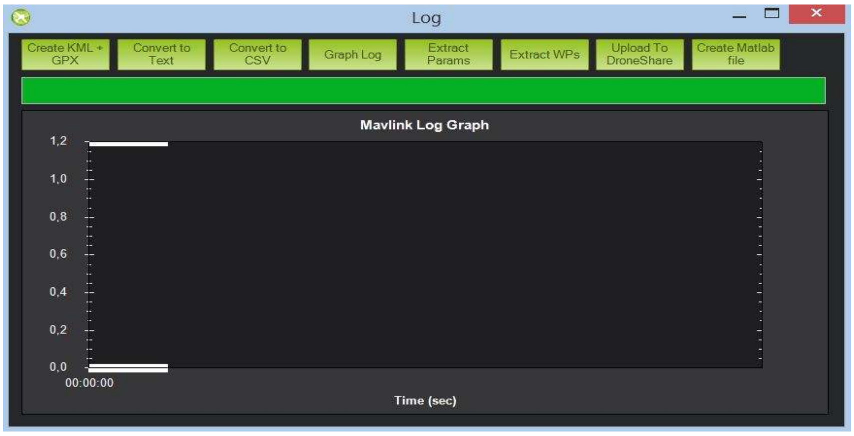


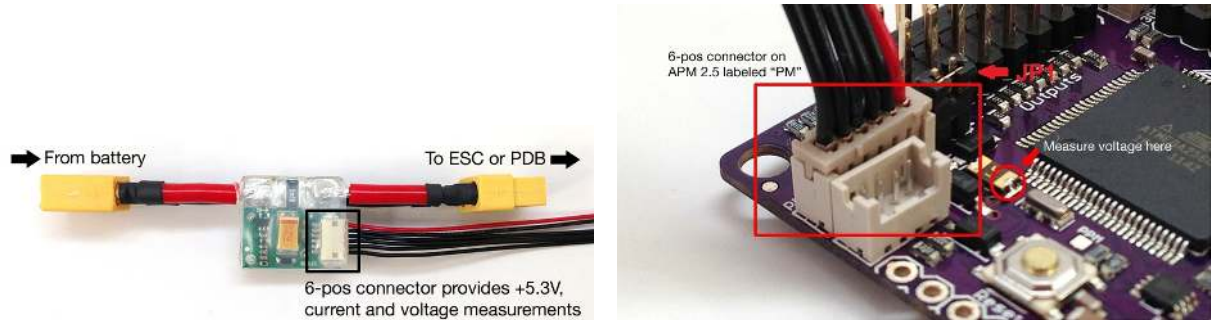
Figure 3.7: Logs menu.

Any parameter recorded in the log can be displayed in graphical form by clicking on the **Graph Log** button and selecting the desired log. This will open a graphic with a list of all the parameters that can be displayed. To display a parameter it needs to be selected.

3.3 Power Supply to the Board

The Ardupilot board needs to be supplied by a power source and this can be achieved by using a 3DR Power Module [18], shown in Figure 3.8. This is usefull as it can be connected to the battery that is used to power the servos of the UAV. The Power Module will allow the monitoring of the battery voltage level while also supplying the Ardupilot board with the required power.

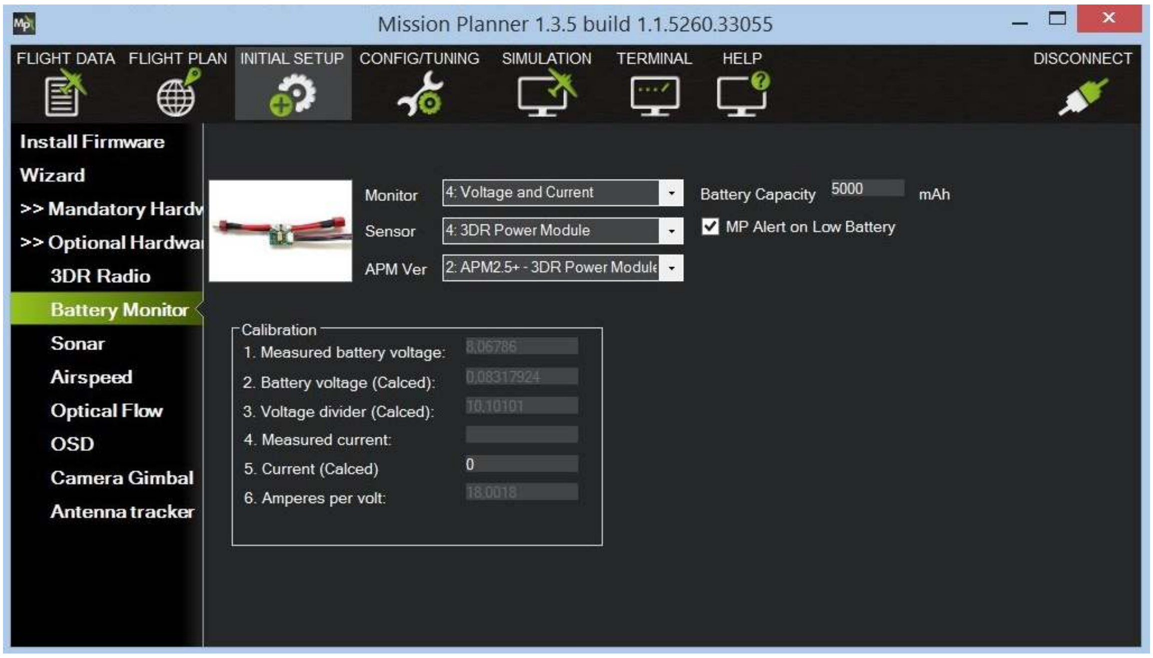
In order to supply the board and the components connected to it with the Power Module, the user has to ensure the the JP1 jumper is not connected. The power module is directly connected to a battery with a maximum voltage of 18 V and will supply the ardupilot board with a stable 5.3 V and 2.25 A current throught the Power port.


From battery
To ESC or PDB
6-pos connector provides +5.3V, current and voltage measurements
6-pos connector on APM 2.5 labeled "PM"
JP1
Outputs
Measure voltage here

(a) Power Module (b) Board Connection

Figure 3.8: Power module connection [18].

After connecting the Power Module to the Ardupilot Board, it needs to be correctly configured throught the Mission Planner. This is shown in Figure 3.9.



Monitor: 4: Voltage and Current
Sensor: 4: 3DR Power Module
APM Ver: 2: APM2.5+ - 3DR Power Module
Battery Capacity: 5000 mAh
☑ MP Alert on Low Battery

Calibration
1. Measured battery voltage: 8.06786
2. Battery voltage (Calced): 0.08317924
3. Voltage divider (Calced): 10.10101
4. Measured current:
5. Current (Calced): 0
6. Amperes per volt: 18.0018

Figure 3.9: Power module configuration.

In order to enable voltage and current sensing the user has to select the following options:

- **Monitor** - 4: Voltage and Current
- **Sensor** - 4: 3DR Power Module
- **APM Version** - 2: APM2.5+ - 3DR Power Module
- **Battery Capacity** - Battery Capacity in mAh

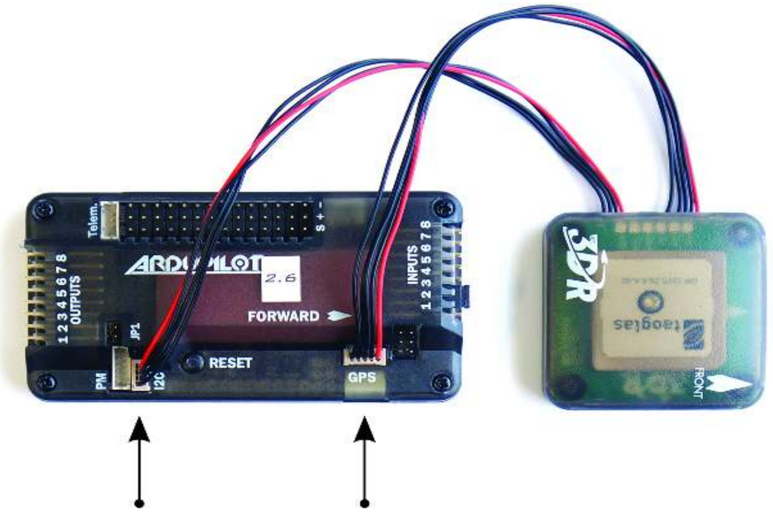
The Mission Planner can be configured to alert verbally when the battery is low. This can configured by checking the *MP Alert on Low Battery* box. The user will be prompted to enter the warning that he wishes to hear, the voltage level and finally the percentage of the remaining current.

The voltage of the battery can be checked if it is being correctly measured by the Ardupilot by comparing the voltage reading on the Mission Planner to a reading from a handheld voltage meter. If it is found that the Ardupilot is not reading the correct voltage this can be corrected by doing the following:

- Set the ”Sensor” field to ”0: Other”;
- Enter the voltage measured using the voltage meter in the ”Measured battery voltage” field.

3.4 GPS and Compass Connection

The 3DR GPS Module [18] provides both the GPS and Compass data. This module is easily connected to the ardupilot board using the GPS and the I2C ports. The connection of this component is explained in Figure 3.10.



Connect GPS MAG port to APM I^{2}C port using 4-position cable.

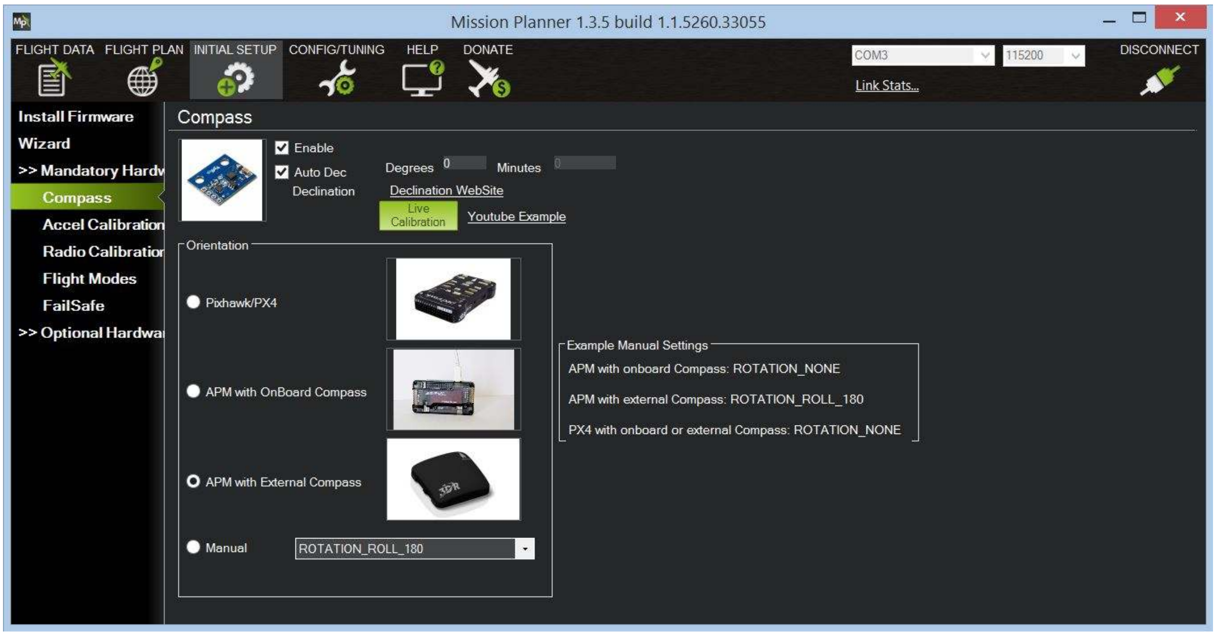
Connect GPS port to APM GPS port using 5-position-to-6-position cable. Connect 6-position connector to GPS and 5-position connector to APM.

Figure 3.10: GPS and compass connection [18].

In order to configure the compass readings, the Ardupilot needs to be connected to the Mission Planner. Then, as shown in Figure 3.11, the compass needs to be enabled by checking the **Enable** box. The declination of the location of the UAV can be manually input in the **Degrees** and **Minutes** boxes, or it can be set to Auto by checking the **Auto Dec** box. The **APM with External Compass** box has to be selected.

3.5 Airspeed Sensor

Ardupilot supports the use of an airspeed sensor, shown in Figure 3.12, which can help in windy conditions, slow flight and autonomous landings. It has a pressure sensor that is connected to a pressure



Compass

☑ Enable

☑ Auto Dec Declination

Degrees 0 Minutes 0

Orientation

○ Pixhawk/PX4

○ APM with OnBoard Compass

● APM with External Compass

○ Manual ROTATION_ROLL_180

Example Manual Settings

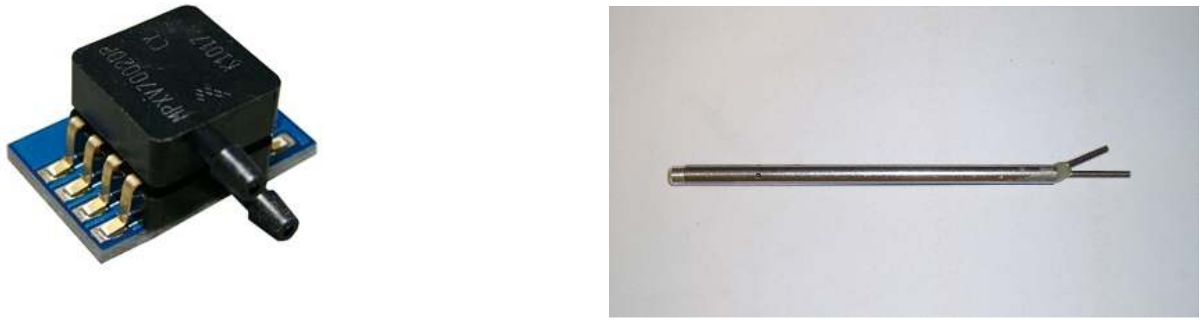
APM with onboard Compass: ROTATION_NONE

APM with external Compass: ROTATION_ROLL_180

PX4 with onboard or external Compass: ROTATION_NONE

Figure 3.11: GPS and compass configuration.

measurement unit. This unit has a pitot tube and a static pressure port. The pitot tube measures the total pressure of the air flow, and the static pressure port measures the static pressure of the air. The top tube that exits the unit is connected to the pressure sensor top tube using a silicon tube. The angled tube of the pressure measurement unit is connected to the bottom port of the pressure sensor using a silicon tube.



(a) Pressure sensor [18].

(b) Pressure measurement unit.

Figure 3.12: Airspeed sensor.

The airspeed sensor is connected to the ardupilot board using an analog port as shown in Figure 3.13.

After the installation of the Airspeed Sensor, it needs to be configured as shown in Figure 3.14. First, the **Enable** and **Use Airspeed** boxes need to be checked. Then, the correct analog pin is selected in the **Airspeed Pin** drop-down menu. In this case the analog pin 0 is being used for the Airspeed sensor.

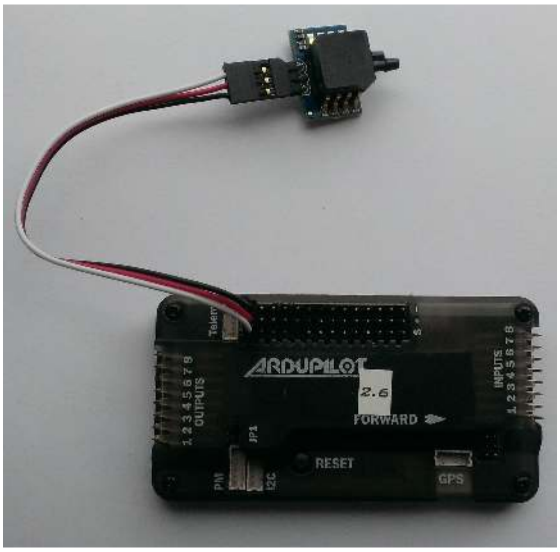
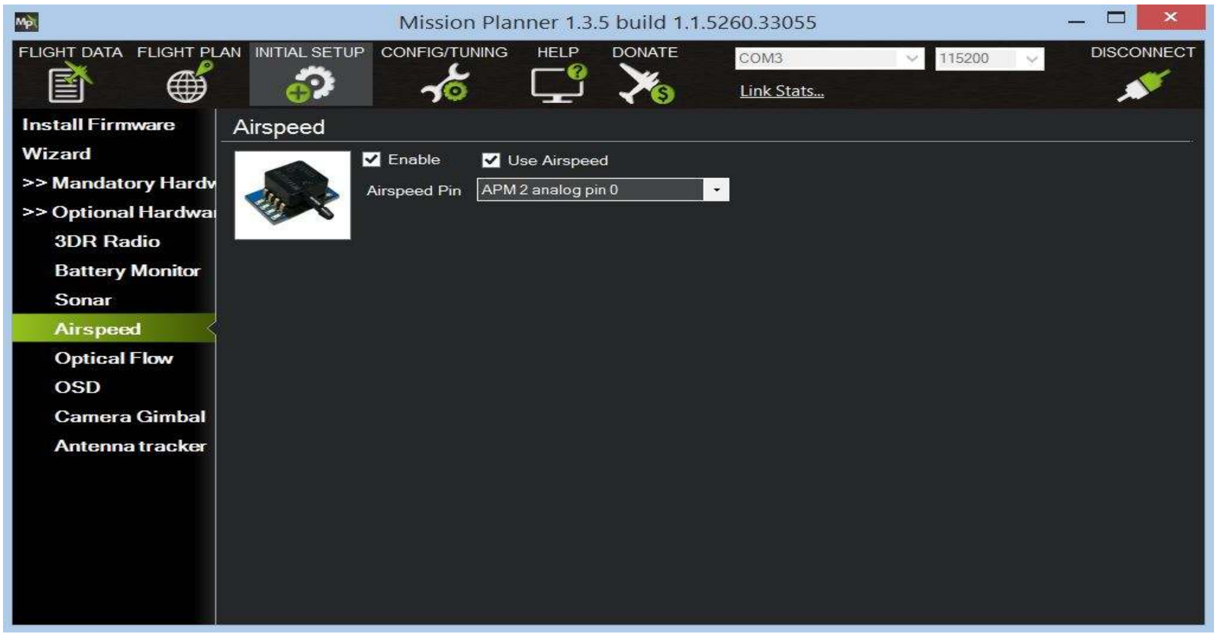


Figure 3.13: Airspeed sensor connection.



☑ Enable ☑ Use Airspeed

Airspeed Pin: APM 2 analog pin 0

Figure 3.14: Airspeed sensor configuration.

3.6 Sonar Sensor

The Sonar sensor [18] provides obstacle avoidance functionality for the Ardupilot. It measures the distance between the sensor and an obstacle in front of it. To connect the sensor to the Ardupilot board, it is required to connect three wires to the sensor's GND (black wire), V+ (red wire) and ”3” (white wire), as shown in Figure 3.15. Then, the connector is pluged into the A0 port similar to the airspeed sensor (with the white wire connected to the ”S” pin and the black wire connect to the ”-” pin. This sensor must be mounted in the front of the vehicle.

The configuration of the Sonar sensor is done on the Advanced Parameters under *Config and Tunning*, as shown in Figure 3.16. The *SONAR_ENABLE* parameter must be set to **Enabled**. The

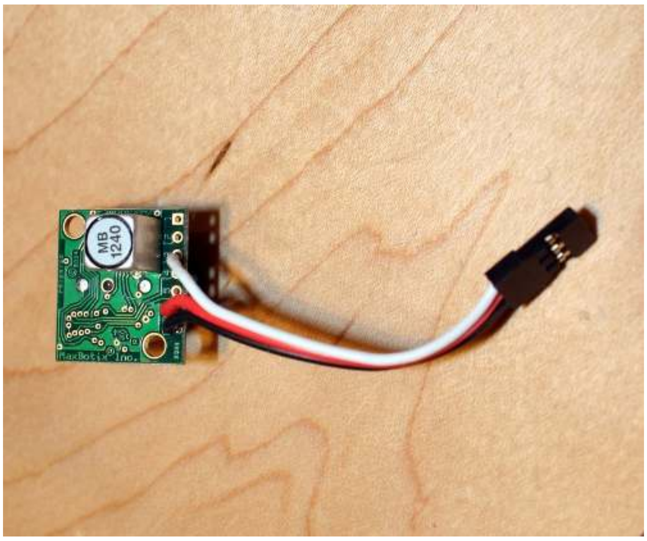
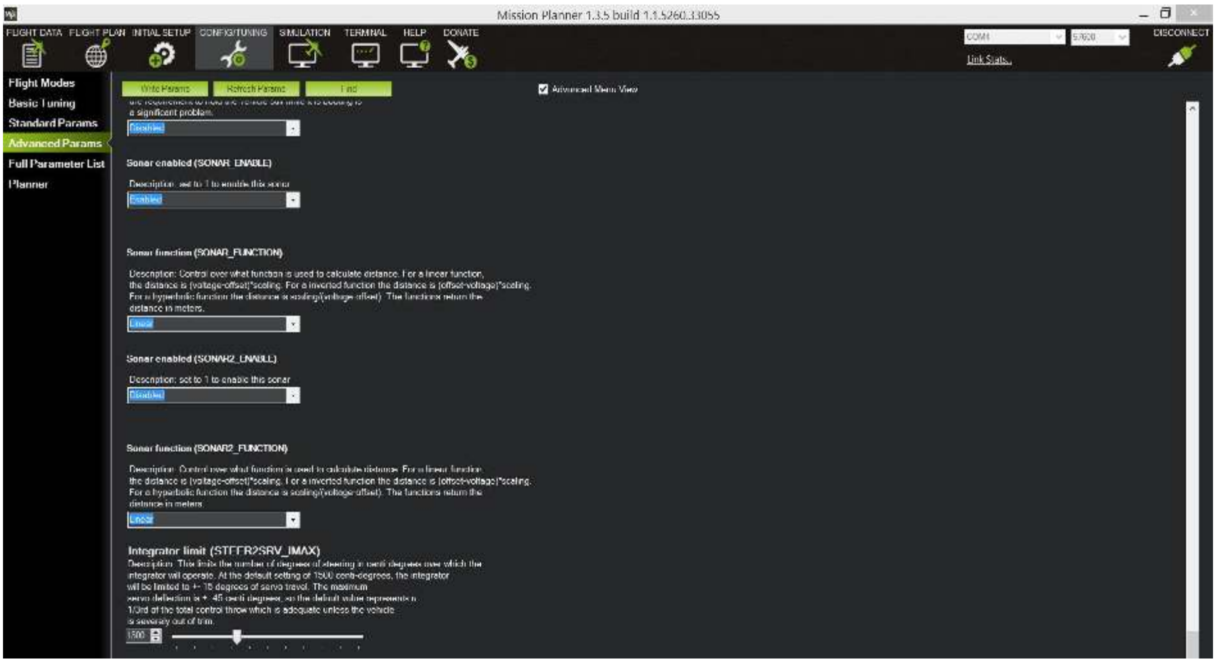


Figure 3.15: Sonar sensor [18].

SONAR_FUNCTION parameter is used to specify the function that is used to calculate the distance, it can be linear, inverted or hyperbolic. If two sensors are used the *SONAR2_ENABLE* must be set to **Enabled** and *SONAR2_PIN* must be set to the correct analog pin where the second sonar is connected.



a significant problem.

Disabled

Sonar enabled (SONAR_ENABLE)

Description: set to 1 to enable this sonar

Enabled

Sonar function (SONAR_FUNCTION)

Description: Control over what function is used to calculate distance. For a linear function, the distance is (voltage-offset)*scaling. For a inverted function the distance is (offset-voltage)*scaling. For a hyperbolic function the distance is scaling/(voltage-offset). The functions return the distance in meters.

Linear

Sonar enabled (SONAR2_ENABLE)

Description: set to 1 to enable this sonar

Disabled

Sonar function (SONAR2_FUNCTION)

Description: Control over what function is used to calculate distance. For a linear function, the distance is (voltage-offset)*scaling. For a inverted function the distance is (offset-voltage)*scaling. For a hyperbolic function the distance is scaling/(voltage-offset). The functions return the distance in meters.

Linear

Integrator limit (STEER2SRV_IMAX)

Description: This limits the number of degrees of steering in centi-degrees over which the integrator will operate. At the default setting of 1500 centi-degrees, the integrator will be limited to +- 15 degrees of servo travel. The maximum servo deflection is +- 45 centi-degrees, so the default value represents a 1/3rd of the total control throw which is adequate unless the vehicle is severely out of trim.

1500

Figure 3.16: Sonar sensor configuration.

3.7 Telemetry Connection

The telemetry connection between Ardupilot and the Ground Station is achieved by using 3DR Radios [18], shown in Figure 3.17. These use antennas with 10 centimeters long. There are two 3DR radios, one for the Ardupilot board and another that connects directly to the ground station computer. The

Ardupilot's radio has a FTDI 6-pin header, allowing it to be directly connected to the telemetry port on the board. The ground station model can be connected to the computer using an USB connector. The required drivers of the 3DR radios should be automatically installed to the computer when the radio is first connected. If, for some reason, they are not automatically installed, they can be manually downloaded and installed from `http://www.ftdichip.com/Drivers/D2XX.htm`.

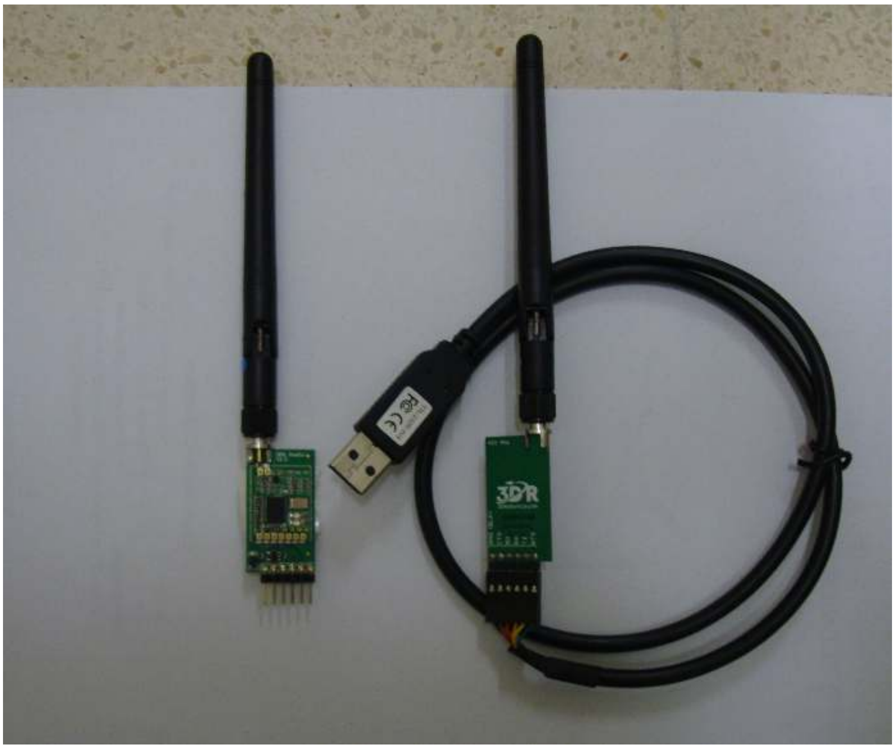


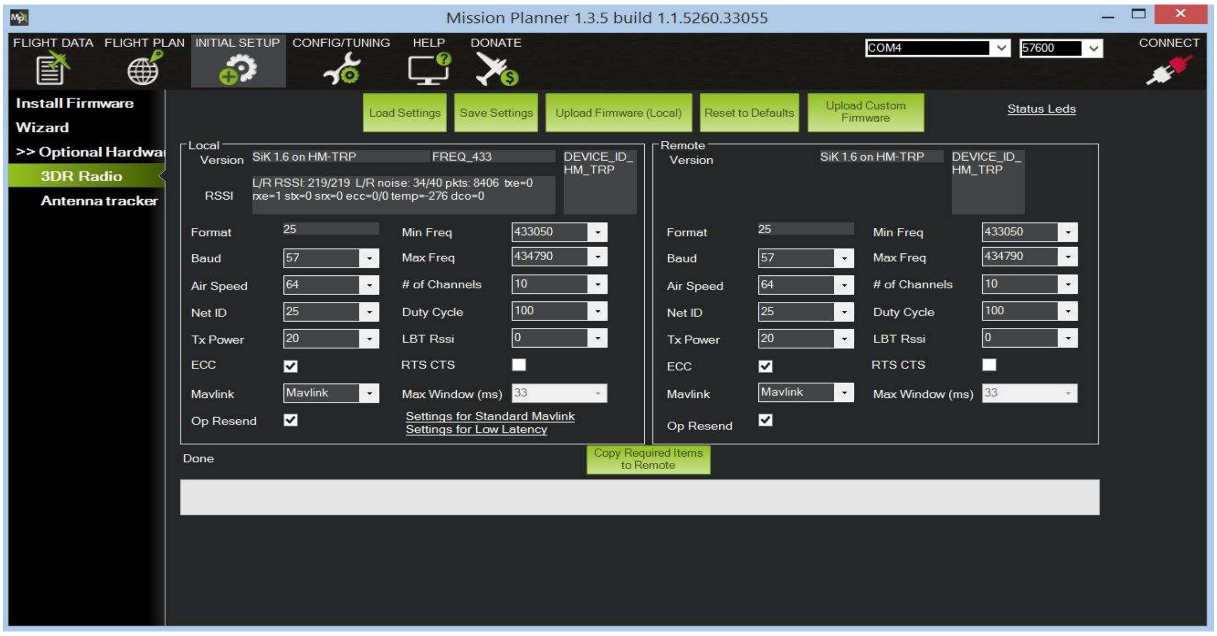
Figure 3.17: 3DR radios.

The 3DR radios have two status LEDs, one green and another red. The meaning of the different state of the LEDs is the following:

- green LED blinking - searching for another radio;
- green LED solid - link is established with another radio;
- red LED blinking - transmitting data;
- red LED solid - in firmware update mode.

To configure the telemetry radios using the Mission Planner, first it is required that in the upper-right corner of the Mission Planner the port where the 3DR radio is connected to the computer is selected. The **Baud rate** must be set to **57600**.

Then, in the *Initial Setup* menu, on the *3DR Radio* tab, under*Optional Harware*, and selecting **Load Settings**, it is necessary the the several boxes have the values shown in Fig. 3.18. After this step, it is required to select **Save Settings**. Now, it is possible to press on the **Connect** to establish the connection between the ground station and the autopilot using the telemetry radios.



Local

Version: SiK 1.6 on HM-TRP FREQ_433 DEVICE_ID_HM_TRP

RSSI: L/R RSSI: 219/219 L/R noise: 34/40 pkts: 8406 txe=0 rxe=1 stx=0 srx=0 ecc=0/0 temp=-276 dco=0

Format	25	Min Freq	433050
Baud	57	Max Freq	434790
Air Speed	64	# of Channels	10
Net ID	25	Duty Cycle	100
Tx Power	20	LBT Rssi	0
ECC	☑	RTS CTS	□
Mavlink	Mavlink	Max Window (ms)	33
Op Resend	☑		

Remote

Version: SiK 1.6 on HM-TRP DEVICE_ID_HM_TRP

Format	25	Min Freq	433050
Baud	57	Max Freq	434790
Air Speed	64	# of Channels	10
Net ID	25	Duty Cycle	100
Tx Power	20	LBT Rssi	0
ECC	☑	RTS CTS	□
Mavlink	Mavlink	Max Window (ms)	33
Op Resend	☑		

Done

Figure 3.18: Telemetry radios configuration.

3.8 OSD Connection

An OSD (On Screen Display) board is used to fly an UAV in FPV (First Person View) mode. It overlays the telemetry stream data on a video stream from an onboard camera. To correctly set up the OSD, the UAV must be equipped with an onboard camera and a wireless video transmitter, as schematically shown in Figure 3.19.

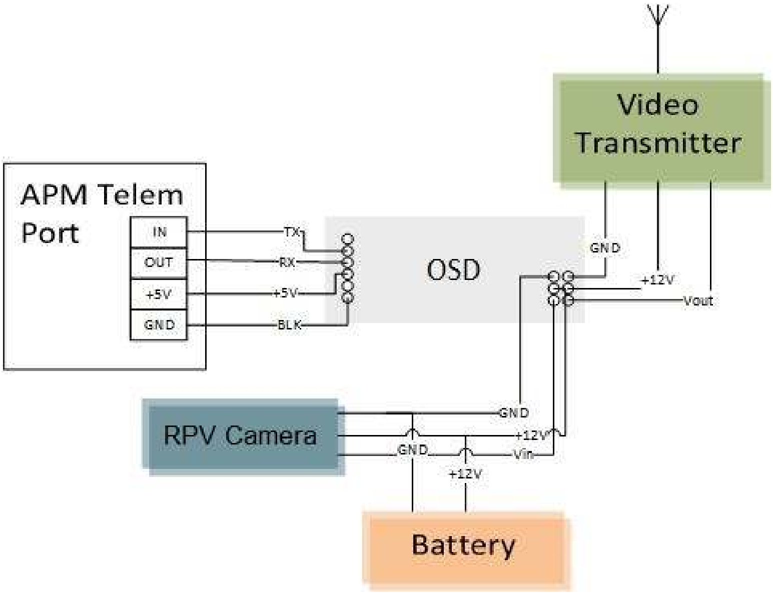


Figure 3.19: OSD connection.

To receive the data from the OSD and the video camera, it is required a wireless video receiver connected to a display such as a computer with the required software or directly to a monitor. This is schematicaly shown in Figure 3.20.

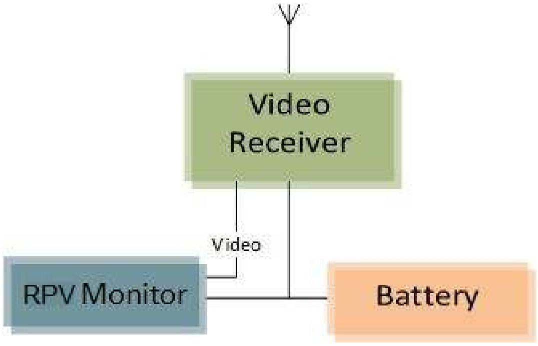


Figure 3.20: OSD Receiver Connection

The OSD and telemetry radio can work simultaneously. This will allow the user to control the autopilot by receiving wireless telemetry data in the ground station while also receiving video and OSD information in a separate monitor. To correctly connect both components in parallel, there needs to be an Y cable connected to the telemetry port and to both the telemetry radio and the OSD, but in this case the OSD TX port must not be connected.

3.8.1 OSD Configuration

The *Arducam OSD Config* is a configuration tool that not only allows the firmware update of the OSD board but also the configuration of the several panels of information displayed on the FPV monitor. It can be downloaded from `https://code.google.com/p/arducam-osd/downloads/list`. After downloading and extracting the files to the desired folder, the program is run by executing *OSD_Config.exe*.

The OSD can be connected to the computer with the same USB cable used to connect the telemetry radio to the ground station. Then, the proper Serial COM port must be selected on the configuration tool.

The latest OSD's firmware update can be downloaded from `http://ardupilot.com/downloads/?did=40`. To upload this firmware, first the OSD must be connected to the configuration tool. Then, on the **Options** tab, selecting **Update Firmware** will require the firmware file to be selected.

The information displayed by the OSD can be managed in the **Panel 1** and **Panel 2** tabs. The functions displayed can be selected and positioned as required. After rearranging the information, the panels can be written to the OSD by clicking on *Save current tab to OSD*.

There is a comprehensive list of available information that can be displayed on the OSD screen. The more important parameters are the following:

- Vehicle attitude such as pitch and roll, climb rate, heading, airspeed, velocity;
- Current altitude, Home altitude, Home direction, Home distance, Waypoint distance, Waypoint direction;
- Flight mode, throttle, battery voltage, time, warnings.

3.8.2 Video Transmitter and Receiver

The video receiver used is shown in Figure 3.21. This receiver works in 1.3 GHz and has 12 channels that can be switched by pressing one of the channel switch buttons. It has a RCA video output and a RCA audio output. The power supply can be from 7 V to 12 V and it has a current consumption of 245 mA [21].

The video transmitter is shown in Figure 3.21. It has 8 channels and works in 1.3 GHz. The channel can be switched by pressing the channel switch button. The transmitter requires a power supply of 12 V and has a current consumption of 235 mA. An important feature of this transmitter is that is weights 21 g [22]. It has a 4 cable connector with the following colour meaning:

- Red - 12 V Power supply;
- Black - Ground;
- Yellow - Video in;
- White - Audio in.

The video receiver and transmitter use omnidirectional 3 dBi antennas, and they need to be working in the same channel for the connection to be properly set up.

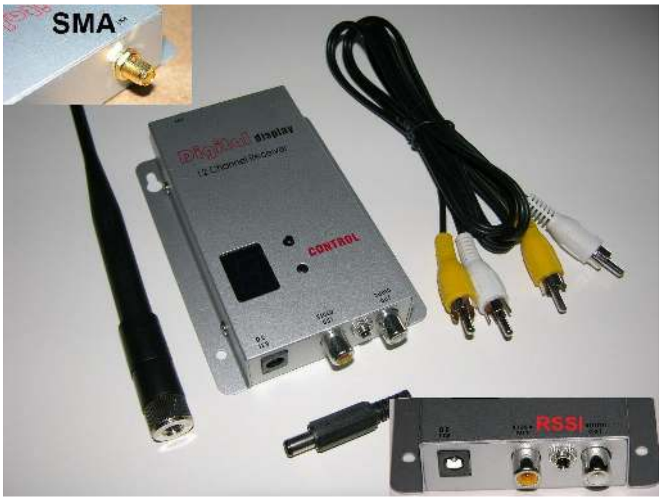


Figure 3.21: Video receiver [21].

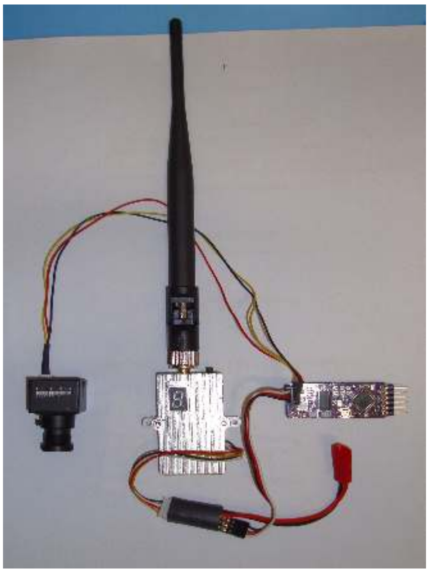


Figure 3.22: Video transmitter, onboard camera and OSD.

Chapter 4

Rover Ground Testing

In order to test and analyse the performance of the Ardupilot, it is first installed on a Radio Controlled (RC) model car, allowing it to be used as a rover. This is an autonomous ground vehicle that can be remotely controlled similarly to an UAV. The main advantage of installing the autopilot on the RC car is that it will allow the ground testing phase to be conducted in a more safe and controlled environment than if it was installed in the UAV, with a lower risk of damaging the vehicle. Furthermore, it will allow the testing of several functions of the autopilot while also giving a more detailed insight of its installation and configuration.

The Ardupilot installation on the rover test platform will be explained in Sections 4.1. The configurations that needed are explained in Section 4.2. Section 4.3 explains how the steering and navigation are tunned. The ground tests and functioning of the autopilot will be shown in Section 4.4.

4.1 Ardupilot Rover Setup

To convert an RC car to a Rover the Ardupilot APM autopilot is added between the RC receiver and the car steering servo and Motor Speed Controller (ESC). The RC system must be connected to the Ardupilot as shown in Tables 4.1 and 4.2.

RC Rx	APM Input	Function
1	1	Steering
2	3	Throttle
5	8	Mode
6	7	Record Waypoint

Table 4.1: Input channels.

APM Output	Car	Function
1	Servo	Steering
3	Motor/ESC	Throttle

Table 4.2: Output channels.

The RC car with the autopilot system is shown in Figure 4.1. The several components of the autopilot system were mounted onto a wooden board to facilitate the connection to the RC car.

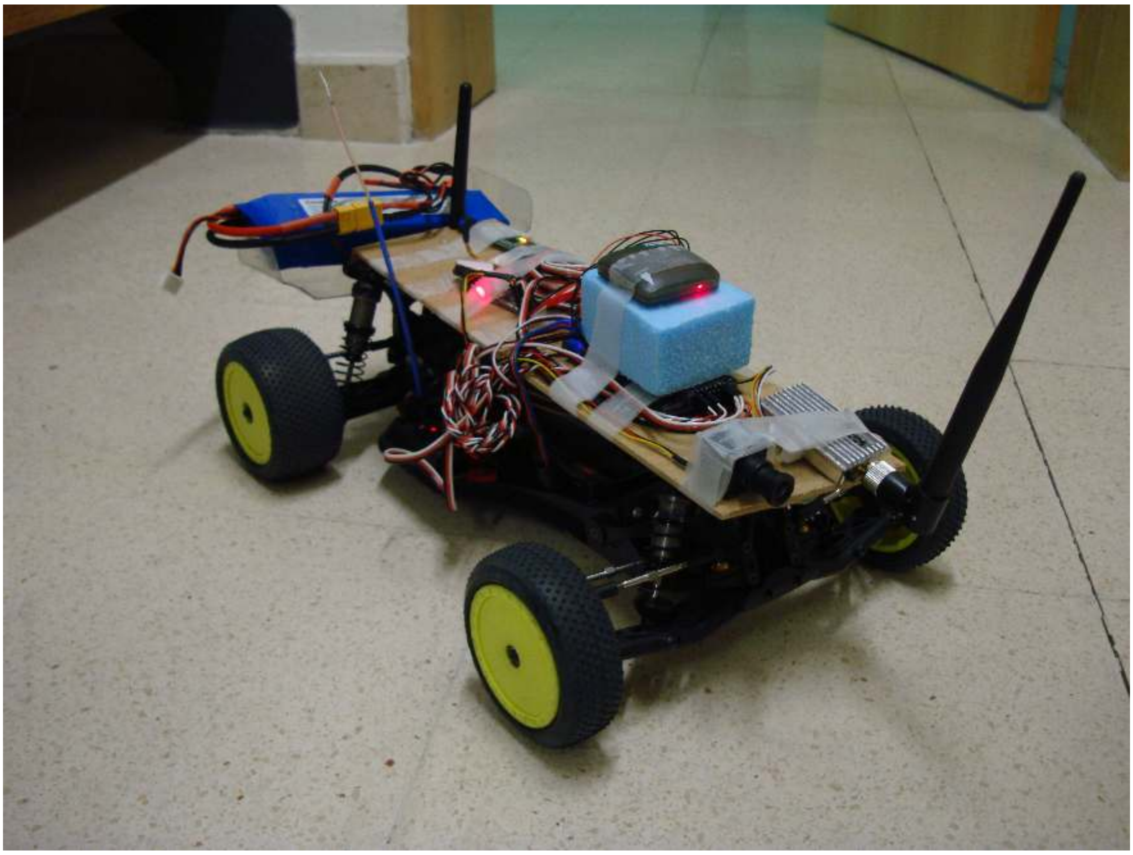


Figure 4.1: Rover test platform.

4.2 Ardupilot Rover Configuration

After correctly connecting the autopilot to the RC system there are some configurations that need to be made. The first is to load the correct firmware, as explained in Section 3.2.2. The firmware that allows the use of the Ardupilot in a Rover is the *ArduRover v2.45*. Then the RC radio settings need to be calibrated. This is achieved in the *Radio Calibration* tab under *Initial Setup*. If the green bars move when the RC transmitter sticks move then by clicking on **Calibrate Radio** and pushing the sticks as far as they will go will define the range of PWM values where the RC transmitter works.

The next configuration is done in the *Flight Modes* section. This is where each flight mode is assigned to a given position for the channel 8 toggle switch. For each position, there are several modes that can be selected. Only mode 6 is allways MANUAL mode. After selecting the desired modes, the configuration has to be saved by clicking on **Save Modes**. The different modes meaning are the following:

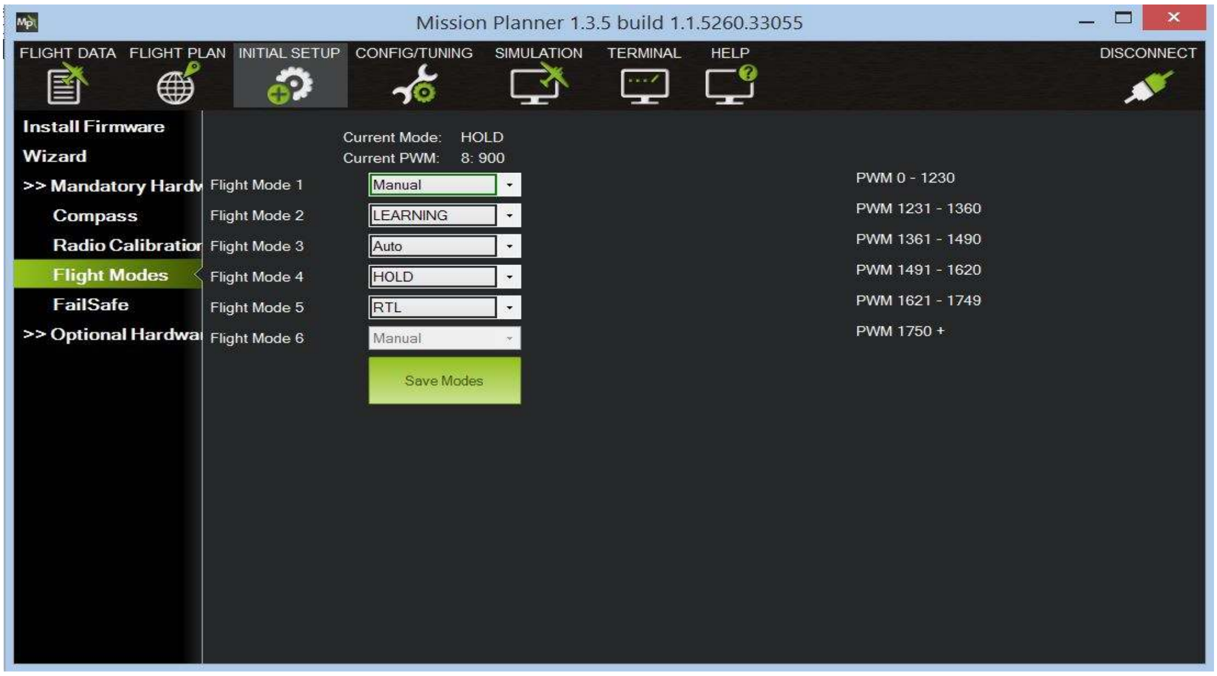
- MANUAL mode - Basic radio pass through drive the vehicle with the RC transmitter
  - Steering and Throttle are manually controled;

  - Only mode that does not require a GPS lock and Home position to control the vehicle. The Home position location is set when the vehicle's GPS achieves the first lock;
  - Mommentary toggle of channel 7 switch clears the waypoints list;
  - Steering hard right when channel 7 is toggled resets the Home location to the current vehicle's position.

- LEARNING mode - Also radio pass through drive the vehicle but waypoints can be recorded for use in Auto mode
  - Steering and Throttle are manually controled;
  - Mommentary toggle of channel 7 switch stores the current GPS position as a new waypoint.
- AUTO mode - Follow a course of waypoints generated in LEARNING Mode or in the Mission Planner
  - The autopilot will loop the waypoints lists stored in memory;
  - Mommentary toggle of channel 7 switch triggers immediate RTL (Return to Launch), the vehicle returns to Home position and enters HOLD mode;
  - To regain manual control of the vehicle, the mode must be switched to MANUAL or LEARNING;
  - If the RC connection is lost the vehicle will automatically enter in HOLD mode.
- STEERING mode - LEARNING mode with obstacle avoidance
  - This mode works similar to the LEARNING mode but the controls use the same code for steering, throttle and obstacle avoidance used in AUTO mode;
  - Driving the vehicle at an obstacle will trigger the vehicle to avoid it, provided that the required sensors are in use;
  - Mommentary toggle of channel 7 switch stores the current GPS position as a new waypoint.
- HOLD mode - the vehicle comes to a full stop
  - The vehicle stops with the wheels straight ahead.

The *Flight Modes* were configured as shown in Figure 4.2.

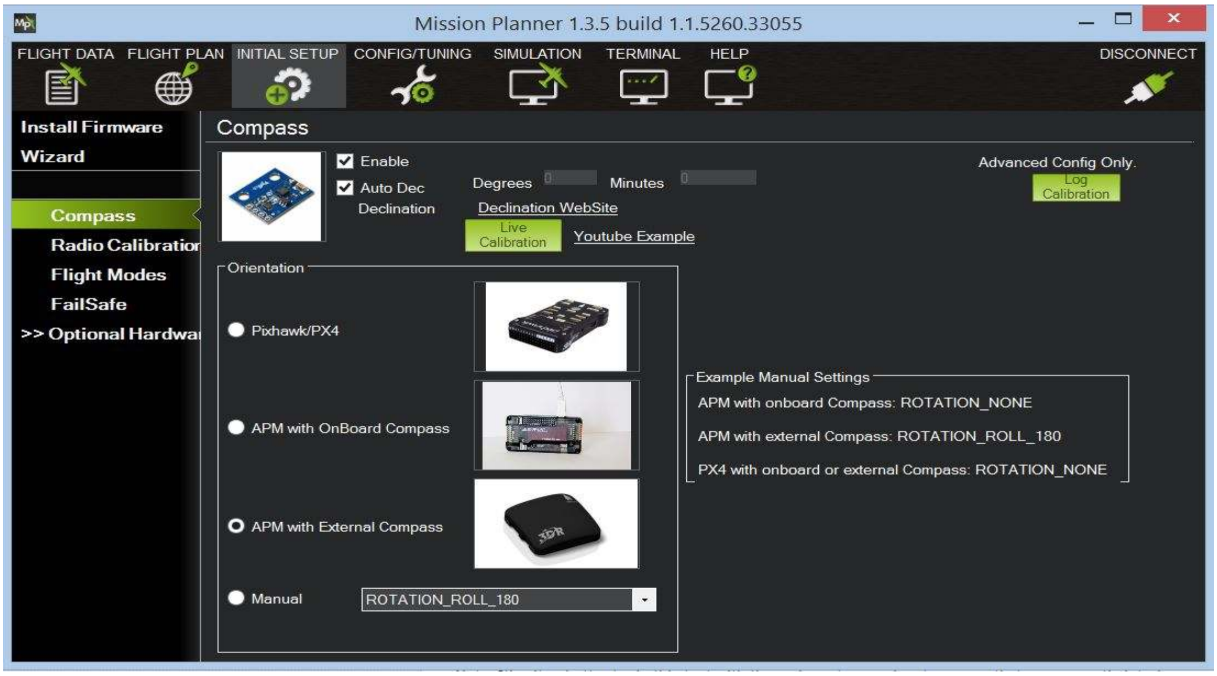
Another important step is the configuration and calibration of the compass of the Autopilot because the Rover does most of its steering based on the compass information. This is done in the *Compass* section as shown in Fig. 4.3. The **Enable** and **Auto Declination** boxes must be checked and the orientation must be set to *APM with External Compass*. To calibrate the compass readings, it is necessary to click on the **Live Calibration** button and follow the instrutions.



Current Mode: HOLD
Current PWM: 8: 900

Flight Mode 1	Manual	PWM 0 - 1230
Flight Mode 2	LEARNING	PWM 1231 - 1360
Flight Mode 3	Auto	PWM 1361 - 1490
Flight Mode 4	HOLD	PWM 1491 - 1620
Flight Mode 5	RTL	PWM 1621 - 1749
Flight Mode 6	Manual	PWM 1750 +

Figure 4.2: Flight modes configuration.



☑ Enable
☑ Auto Dec Declination
Degrees 0 Minutes 0

○ Pixhawk/PX4
○ APM with OnBoard Compass
● APM with External Compass
○ Manual ROTATION_ROLL_180

Example Manual Settings
APM with onboard Compass: ROTATION_NONE
APM with external Compass: ROTATION_ROLL_180
PX4 with onboard or external Compass: ROTATION_NONE

Figure 4.3: Compass configuration.

4.3 Steering and Navigation Tunning

For the Rover to function properly, there are six key parameters that need to be correctly configured. These are:

- STEER2RSV_P - This parameter tells the rover code what the turning circle, as a diameter in meters, for the rover is. This is important because it is used to calculate the steering angle for a desired turn rate.

- TURN_MAX_G - This parameter tells the rover the maximum G force that the rover can handle while remaining stable. Setting it too high will cause the rover to flip over or skid, while setting it too low and the rover may not turn sharply enough.

- NAVL1_PERIOD - This controls the navigation agressiveness in AUTO mode. A smaller number means more agressive turning while larger numbers mean larger, smoother turns.

- SPEED_TURN_GAIN - This controls how much the rover should slow down while turning, as a percentage of the current target speed. Setting it to 100 and the rover will not slow down in turns, while reducing this value will improve handling in tight courses.

- CRUISE_SPEED - This parameter controls the target speed in m/s in AUTO mode.

- CRUISE_THROTTLE - This sets the initial guess at what the throttle should be to achieve CRUISE_SPEED when driving straight ahead. This needs to be right for the rover to achieve good speed control.

To start tunning these parameters, they need to be set to the initial conservative values presented in Table 4.3. These parameters can be changed in the *Config and Tunning* tab, under **Advanced Params** (after checking the *Advanced Menu View* box, in **Standard Params**).

Parameter	Value
STEER2RSV_P	2
TURN_MAX_G	1
NAVL1_PERIOD	6
SPEED_TURN_GAIN	100
CRUISE_SPEED	2
CRUISE_THROTTLE	20

Table 4.3: Steering and navigation initial parameters.

The steps presented bellow were followed to configure the steering and navigation parameters. The final values are presented in Table 4.4.

1. To configure the CRUISE_THROTTLE value it is required to know what throttle percentage will give a speed of 2 m/s. This was achieved by driving the car in MANUAL mode and noting the value of throttle presented in the ground station for a speed of 2 m/s.

2. The STEER2RSV_P is set up by driving the rover in MANUAL mode and turning hard to one side allowing the rover to make a complete circle. Then, measuring the diameter of this circle in meters will give the correct value for this parameter.

3. The TURN_MAX_G is tuned by driving the rover in STEERING mode and turning at maximum turn rate and maximum throttle, while adjusting this parameter to the highest level that the rover remains stable without starting to roll over or skid.

4. The NAVL1_PERIOD is used to tune the steering in AUTO mode. First it is necessary to create a course for the rover to follow in this mode, a rectangular course is best to start tuning this parame-

ter. Then if the rover weaves along the straights or turns too late in the corners the NAVL1_PERIOD should be raised. If the rover doesn't turn sharply enough then the parameter should be lowered.

After configuring the parameters listed above, the CRUISE_SPEED may be raised for higher driving speeds. If so, it is necessary to set the correct value for the CRUISE_THROTTLE as explained.

The Ardupilot has a vast list of parameters that can be configured. A full list of these parameters with a complete description can be found in `http://rover.ardupilot.com/wiki/apmrover2-parameters/`.

Parameter	Value
STEER2RSV_P	1
TURN_MAX_G	1
NAVL1_PERIOD	9
SPEED_TURN_GAIN	100
CRUISE_SPEED	2
CRUISE_THROTTLE	100

Table 4.4: Steering and navigation parameters.

4.4 Ground Testing

The ground testing phase main objectives were first to check if the overall system was correctly assembled, to analyse and improve, if possible, the behavior of the autopilot in the several flight modes and to correct any problem that was not expected.

The test of the Ardupilot functioning was conducted at the sports field in *Instituto Superior Técnico*, Lisbon. This was an ideal place as it provided a wide flat ground with a clear view to the sky which was necessary for the correct functioning of the autopilot GPS. It also provided some landmarks that help to identify the precise location of the rover when comparing it to the position displayed on the Mission Planner map.

First, the system was tested using the wireless telemetry connection but with the OSD disconnected. When the system is started up it takes a couple of minutes to acquire a GPS lock. By comparing the real position of the rover with the position displayed on the Mission Planner map, it was possible to see that the error was relatively small, especially when the GPS acquired six or more sattelites.

After analysing the GPS performance, the flight modes functioning were accessed. First, the rover was driven in MANUAL mode and it proved to be working correctly. The rover would respond as expected to the inputs sent by the RX controller, allowing it to be driven from a remote location. After testing the MANUAL mode, the rover was driven in LEARNING mode in order to record waypoints files to test the AUTO mode later on. The rover was driven to a certain position and that position was recorded as a waypoint by clicking on the RX controller button that was configured as the channel 7 switch. First it was recorded simple rectangular tracks and later the tracks shown in Figure 4.4 and Figure 4.6. Having a waypoint list recorded in the autopilot memory and switching the flight mode to AUTO will make the rover follow the waypoints from the beginning and stopping at the last waypoint. This mode proved to be working correctly after increasing the CRUISE_THOTTLE parameter that was set to low at the beginning which did not allow the rover to move.

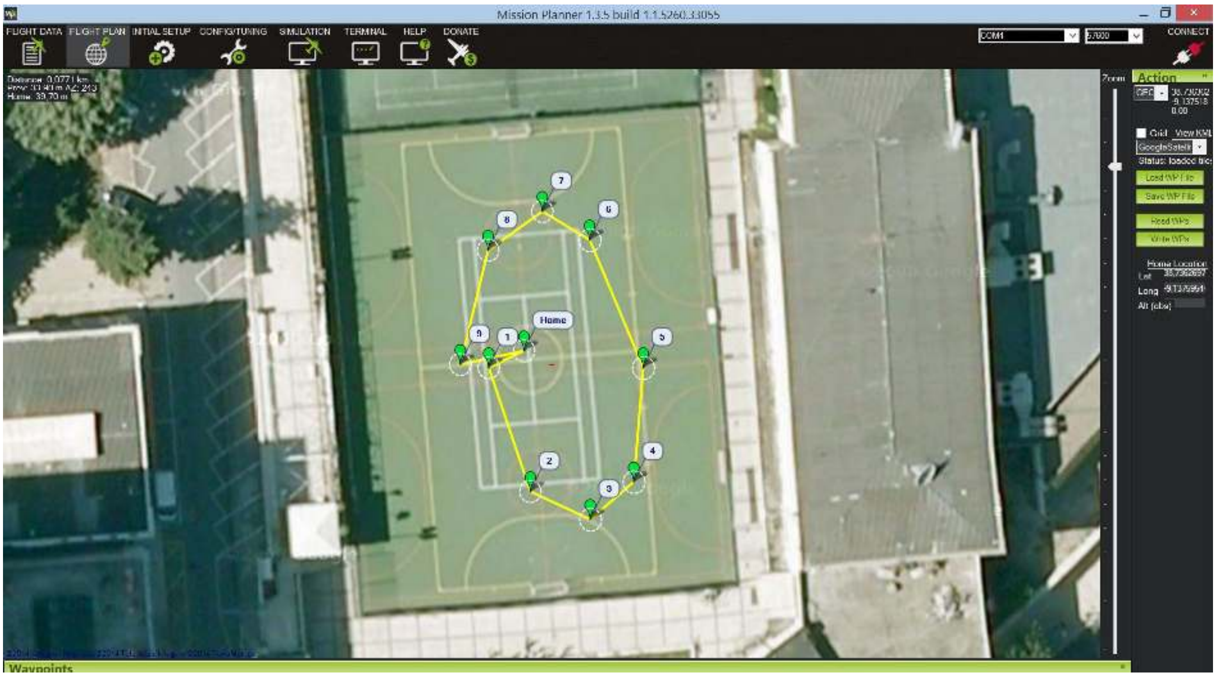


Figure 4.4: Oval track.

Other flight modes that were tested were the HOLD and the RTL (Return to Launch). The first one proved to stop the rover immediately when activated. The RTL mode drives the rover automatically to the HOME location that is first set when the system is started up. Both modes proved to be working correctly.

When all the tests of the autopilot were concluded, the performance of the OSD was analysed. The video and OSD data was correctly received at the ground station computer, as shown in Figure 4.5.

However, there were some problems as the video transmitter antenna interfered with the reception of the RX controller signal, causing the rover to lose the controller connection at a distance of around 10 meters. Furthermore, the video transmitter overheated when turned on for some time, but this was corrected by installing an heat dissipation plate on the transmitter.

The ground test phase proved to be helpfull due to severall reasons. The most important is that it gave a detailed insight of how the system is correctly connected and configured. Furthermore, it was possible to identify some problems such as the interference of video transmitter antenna with the receiver controller signal and the heating of the video transmitter. The behavior of the autopilot in the severall flight modes was also accessed and it proved to be working correctly with no faults.

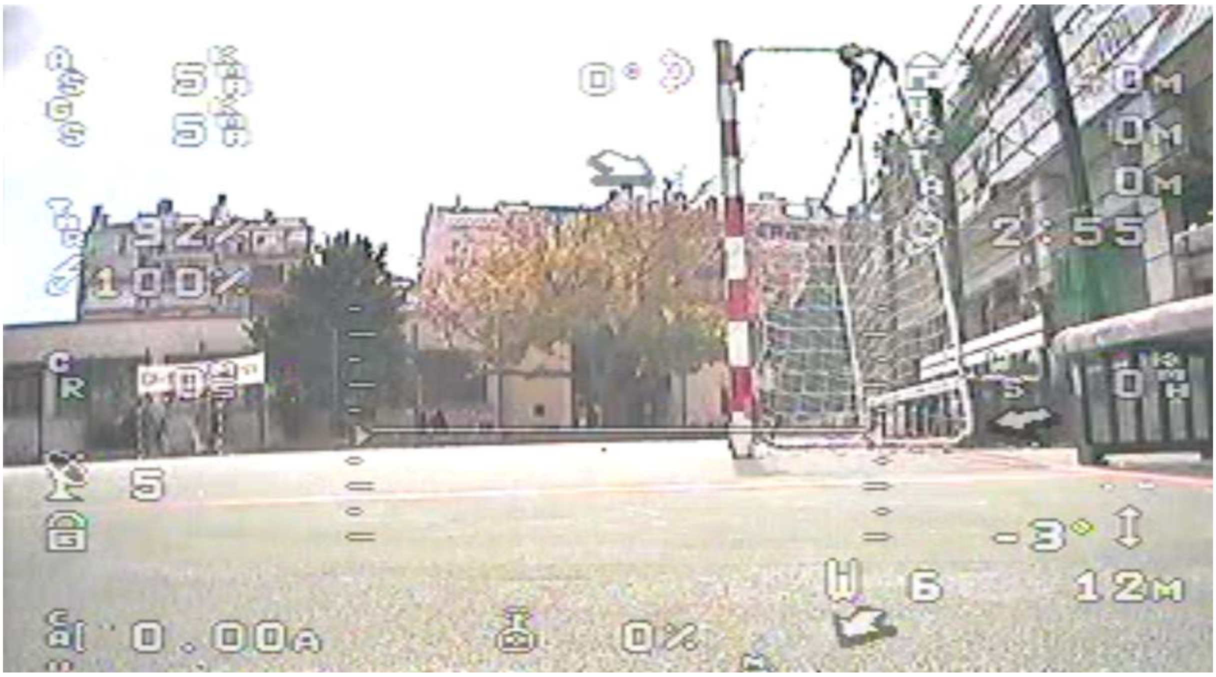


Figure 4.5: OSD display.

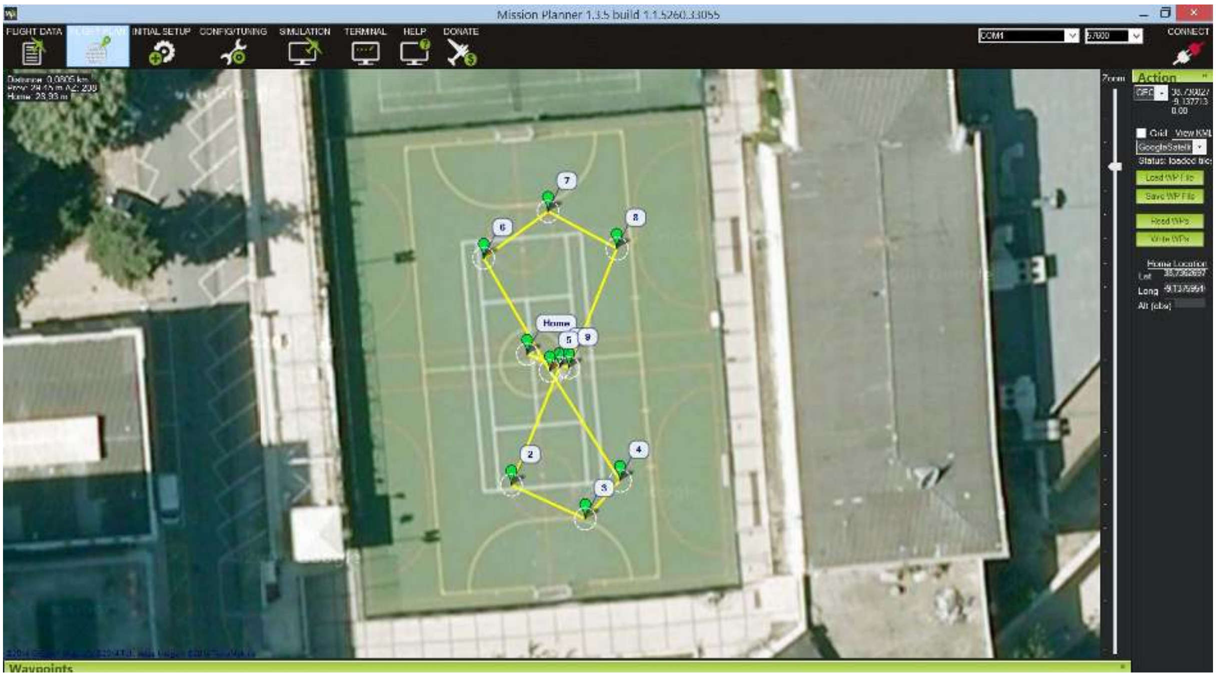


Figure 4.6: Figure 8 track.

Chapter 5

UAV Flight Testing

The objective of this chapter is to give a detailed description of how the autopilot system is installed and configured on the aircraft, so it can operate as an UAV.

In Section 5.1, it is explained how the autopilot is installed on the plane. How the autopilot is configured in the Mission Planner is explained in Section 5.2. Section 5.3 is a checklist that should be followed prior to the first flight of the aircraft. Section 5.4 gives a complete description of how the autopilot should be tunned for further performance improvements. Finally, in Section 5.5 the flight test results are presented.

5.1 Ardupilot Installation

To install the Ardupilot autopilot on the plane, it is first required that the board inputs and outputs are correctly connected. These connections are similar to those explained in Section 4.1. The RC receiver channels must be connected to the APM inputs as presented in Table 5.1. The APM outputs must be connected to the UAV servos as shown in Figure 5.1.

RC Rx	APM Input	Function
2	1	Roll/Aileron
3	2	Pitch/Elevator
1	3	Throttle
4	4	Yaw/Rudder
6	8	Aux4(Mode Switch)

Table 5.1: Input channels.

Before installing the autopilot on the plane, it is best to load the correct firmware into the Ardupilot board. In this case the firmware is the *Arduplane v3.1.0*. When installing the APM board on the plane it is very important that the board is correctly oriented, with the GPS connector facing forward and the servo cables facing back. It should also be as close as possible to level when the plane is in its flying orientation. Ideally, the board should be mounted as close to the center of gravity as possible and mounted on a foam plate onto a solid platform.

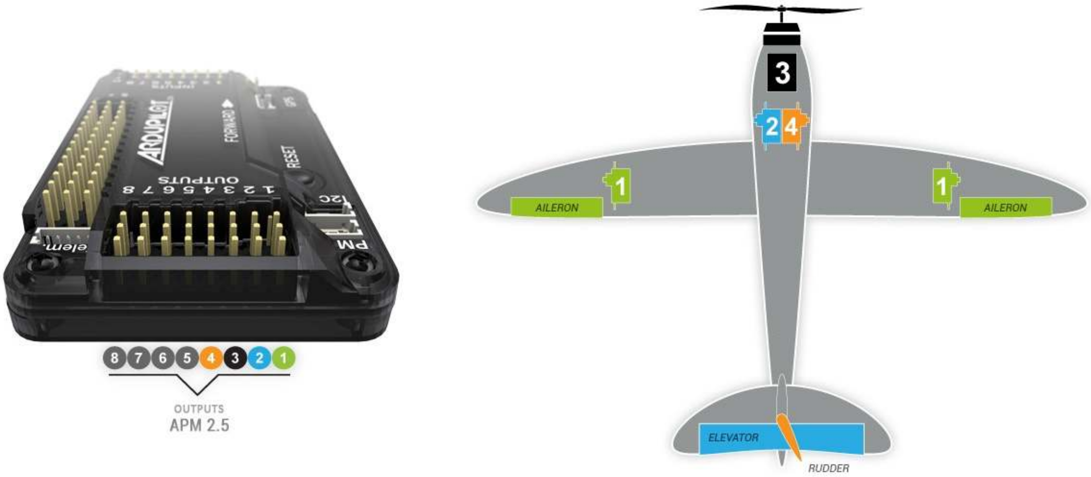


Figure 5.1: APM outputs [18].

Regarding the GPS module, there are also some considerations to take into account. It should be mounted with the square antenna side facing up, regardless of its orientation. It should also be as far as possible from radio transmission equipment and, ideally, it should have a clear view to the sky.

Figure 5.2 shows the aircraft with the autopilot mounted on top of the fuselage. The Airspeed sensor is mounted on the left wing. The GPS module is mounted on top of the autopilot platform. The telemetry antenna was mounted on top of the fuselage behind the autopilot.

5.2 Ardupilot Configuration

After loading the firmware into the Ardupilot board and connecting it to the RC controller, it is necessary to calibrate the RC inputs. This is achieved by doing what was explained in the Rover testing in Section 4.1.

It is also necessary to configure the *Flight Modes* section, where it is possible to assign different flight modes that can be switched with the controller. The configuration process is equal to the Rover flight modes configuration, but in this case there are different flight modes, which have the following meaning:

- MANUAL mode - This mode provides full manual control of the plane without providing any stabilization.

- STABILIZE mode - Provides RC control with simple stabilization. The autopilot levels the plane if no control inputs are applied. In this mode the throttle is limited by the THR_MIN and THR_MAX parameters.

- FLY BY WIRE_A (FBWA) - In this mode the autopilot will hold the roll and pitch specified by the control inputs. The roll limit is specified in the LIM_ROL_CD parameter and the pitch limits in LIM_PITCH_MAX and LIM_PITCH_MIN parameters. The throttle is manually controlled but limited by the THR_MIN and THR_MAX parameters.

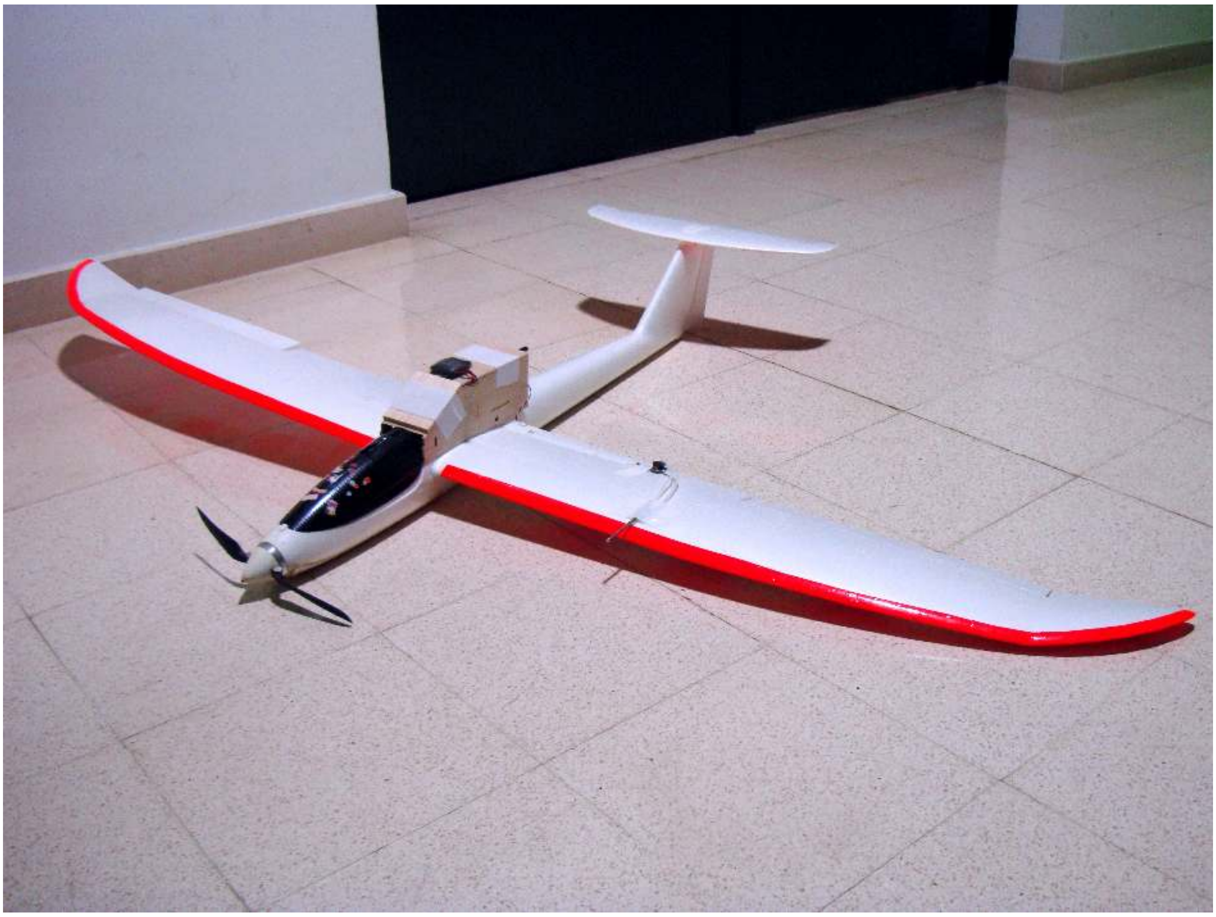


Figure 5.2: UAV test platform

- FLY BY WIRE_B (FBWB) - This mode is similar to FBWA but the autopilot will hold the altitude as well. In this mode, the altitude is controlled with the elevator. If the elevator is left centered, the plane will hold its current altitude. If the elevator is moved, the plane will gain or lose altitude in proportion to how much the elevator was moved. The elevation gain for a full elevator deflection can be set in FBWB_CLIMB_RATE. The target airspeed is controlled using the throttle and it is limited by the ARSP_FBW_MIN and ARSP_FBW_MAX parameters.

- AUTOTUNE - This mode flies similar to FBWA but it does automatic tunning of roll and pitch control gains. A detailed description of using the automatic tunning can be found in `http://plane.ardupilot.com/wiki/automatic-tuning-with-autotune/`.

- TRAINING - This mode is ideal for teaching beginners manual RC control. If the roll is less than the LIM_ROLL_CD parameter than the pilot has manual roll control. If the pilot tries to go past that limit the autopilot will held the roll at that limit. The same applies to the pitch where the limits are specified in the LIM_PITCH_MIN and LIM_PITCH_MAX parameters.

- ACRO - Is a mode for advanced users that provides rate based stabilization with attitude lock. This mode is good to use for acrobatic maneuvers such as rolls or loops. A more detailed description of this mode can be found in `http://plane.ardupilot.com/wiki/flying/flight-modes/`.

- CRUISE - It is similar to FBWB but provides a ”heading lock”. It is ideal for long distance FPV flight, as the plane can be directed to a distant object and it will accurately track to that object, automatically controlling altitude, distance and heading.

- Return to Launch (RTL) - The plane will return to the launch point, (the point where the plane acquired its first GPS lock) and loiter there until manual control is regained. The target altitude for RTL mode is set in the ALT_HOLD_RTL parameter.

- LOITER - In this mode the plane will circle around the point where it was ordered to loiter, holding its current altitude. The radius of the circle is set by WP_LOITER_RAD parameter, but is also limited by the NAV_ROLL_CD limit and the NAVL1_PERIOD navigation tuning.

- CIRCLE - It is similar to loiter but it does not attempt to hold the position. It is meant to be used as a failsafe and when a failsafe occurs the aircraft will enter by default this mode for 20 seconds, before switching to RTL.

- GUIDED - This mode is used when it is desired that the aircraft flies to a specific point on the map without setting up a mission. Using the ground control station ”Click to Fly” feature, the operator can click on the map to send the plane to the desired location, it will then enter LOITER mode at that location. It is also used in GeoFencing. When the geofence is breached, the plane will enter GUIDED mode to return to a predefined geofence return point. A detailed description about GeoFencing can be found in `http://plane.ardupilot.com/wiki/geofencing-3/`.

- AUTO - This mode works similar to the AUTO mode presented in the Rover section 4.1. It allows the plane to follow a mission set by the ground control station. It allows the pilot to influence the flight of the plane if the STICK_MIXING option is enabled. If it is enabled, the pilot control inputs will override the autopilot control commands.
  Within this flight mode there are also two sub-modes which are TAKEOFF and LAND. These are both set by the mission control scripting only.

  - TAKEOFF - The takeoff mission specifies a target pitch and target altitude, and during takeoff the aircraft will use the maximum throttle set by the THR_MAX parameter. When the aircraft reaches its target altitude the takeoff is considered complete. It is important that the aircraft is pointing to the wind direction and aligned with the runway before takeoff. The autopilot will hold the heading during takeoff with the initial heading set by the direction of the plane when the takeoff starts. It is recommended that a compass is enabled and correctly configured for auto-takeoff.

  - LAND - In LAND mode the throttle and altitude are controlled by the autopilot. After getting close to the LAND_FLARE_ALT from the target altitude or LAND_FLARE_SEC seconds from the target landing point the plane will flare to the LAND_PITCH_CD and will hold heading for the final approach.

The UAV flight modes selected are presented in Table 5.2. There are six flight modes in Mission Planner, but the sixth mode is automatically set to MANUAL and this can not be changed.

Mode Number	Mode Chosen
1	MANUAL
2	STABILIZE
3	FBWA
4	GUIDED
5	AUTO

Table 5.2: UAV flight modes.

5.3 First-time Flight Checks

1. **Ground Calibration**

   It is best to set the transmitter mode switch to MANUAL, which is a safe mode to start the system. When the board is powered up at the field, the plane should be left montionless on the ground until the three LEDs stop flashing. This will ensure that the gyros calibrate and the plane is ready to fly after acquiring a GPS lock. After the first flight, the autopilot will have learned the gyros offsets and the need to do this for subsequent flights can be avoided by setting the SKIP_GYRO_CAL parameter to 0.

2. **Setting the Home Position**

   Before flying it is necessary that the autopilot acquires a GPS lock. When the boards is powered up and first acquires the GPS lock, it will set that location as Home Position. This is the location that the plane will return to if it executes an RTL. So it is advisable that the plane should only be powered up at the location where it should return to.

3. **Checking the Control Surfaces**

   The correct functioning of the control surfaces should be checked before every flight. This is done by holding the plane in the hands and switching the flight mode to STABILIZE. Then by pitching and tilting the plane, the control surfaces should move the correct way to stabilize the aircraft. This will ensure that a channel was not accidentally reversed.

4. **First Flight**

   For the first flight, the takeoff should be made in MANUAL mode. Then, during the flight, the flight mode can be switched to STABILIZE or FBWA in order to observe the control surfaces behaviour. They should move and return the plane to level when the plane it pitched or rolled. A detailed explanation of how to tune the gains of the autopilot can be found in 5.4.

5. **Second Flight**

   For the second flight the navigation can be tested by switching the flight mode mid-flight to RTL. The aircraft should return to the HOME location and orbit overhead at a fixed altitude. If it does not circle overhead in a near-perfect circle, the roll parameters need to be tuned. A detailed description of how to tune the roll, pitch and yaw controllers can be found in section 5.4.1. Once all of this has been checked out, a waypoint mission can be programmed and tested in AUTO mode.

5.4 Navigation and Performance Tunning

With the default settings, the autopilot will fly the majority of hand launched RC airframes safely, but for a better navigation and a more reliable performance it is necessary to tune the autopilot. This is achieved in the *APM:Plane Pids* under the *CONFIG/TUNNING* tab. In this screen the Servo Roll and Servo Pitch PID gains can be tunned to enable flight in FBWA mode, and also the L1 Control gain can be tunned for a more accurate horizontal positioning.

It is also possible to load and save configuration files in the *Advanced Parameters* screen. In order to send changes made in parameters to the autopilot it is necessary to press **Write Params**. Pressing the **Read Params** will load the current saved parameters on the board into the Mission Planner.

In Section 5.4.1 will be made a detailed description of how to tune the roll, pitch and yaw controllers gains. Section 5.4.2 explains how the speed and height controller is tunned. Finally, Section 5.4.3 explains how the horizontal navigation controller can be tunned.

5.4.1 Roll, Pitch and Yaw Controllers Tunning

Before tunning the controllers it is required that the following has been completed:

1. The aicraft is trimmed;

2. The radio calibration has been done;

3. The airspeed sensor is calibrated;

4. The autopilot is leveled;

5. FBWA has been selected has one of the available flight modes;

6. The pitch and roll angle displayed in the HUD match the rotation of the plane.

Initial Assessment

The aircraft should takeoff in MANUAL mode and the trims and throttle should be adjusted to a cruise position so that the plane flies straight and level at a speed that should correspond to somewhere between 30% to 60% throttle.

Then the flight mode should be switched to FBWA. The aircraft should continue to fly with wings level and at a fairly constant height. If it wants to roll or pitch more than a small amount then there is a problem with the models trim, autopilot level or radio calibrations, and it needs to fixed before proceeding further.

If the plane starts to wag its wings, then it should be immediately switched to MANUAL mode and the RLL2SRV_P parameter should be halved before switching back to FBWA.

If the plane starts to porpoise, then it should be immediately switched to MANUAL mode and the PITCH2SRV_P parameter should be halved before switching back to FBWA.

Roll Control Tunning

A simple method to tune the roll controller is the following:

1. With the airplane in FBWA it should be put in a rapid bank angle demand by pushing the aileron stick all the way over, holding it for a couple of seconds and then releasing it. This should be done in both directions. It is desired that the airplane rolls quickly and smoothly to the new bank angle and back again without any overshoot or wing oscillations. If the roll response is too slow then the RLL2SRV_P parameter should be increased in increments of 0.1 until the plane has the desired response;

2. If there are bank angle oscillations or overshoot, the RLL2SRV_P parameter should be reduced;

3. In order for the controller to cope better with the wind the RLL2SRV_I parameter can be slowly increased. Usually, a value of 0.05 will work in several models. If overshoot or oscilliations start to occur then this parameter should be halved.

Pitch Control Tunning

The method to tune the pitch controller is similar to the method present above.

1. With the airplane in FBWA it should be put in a rapid pitch angle demand, holding it for a couple of seconds and then releasing it. This should be done in both directions. It is desired that the airplane pitches quickly and smoothly to the new pitch angle and back again without overshoot or purpoise. If the pitch response is too slow then the PTCH2SRV_P parameter should be increased in increments of 0.1 until the plane has the desired response;

2. If there are bank angle oscillations or overshoot, the PTCH2SRV_P parameter should be reduced;

3. The airplane should be rolled in the maximum bank angle in each direction. The nose should stay fairly level during the turns without a significant gain or loss of altitude. If, during the turns, the plane gains height then the PTCH2SRV_RLL should be reduced in small increments of 0.05. If the plane descends immediately when the plane banks then the PTCH2SRV_RLL should be increased in small increments of 0.01.

Yaw Control Tunning

The yaw controller can be configured either as a simple yaw damper or as a combined yaw damper and side-slip controller. Given that control of side-slip uses measured lateral acceleration, it will only work for aicrafts that have enough lateral fuselage side area to produce a measurable lateral acceleration when they side-slip. The method to tune the yaw damper controller is the following:

1. The YAW2SRV_SLIP, YAW2SRV_INT and YAW2SRV_DAMP parameters must be initially set to zero. The YAW2SRV_RLL must be set to 1.0;

2. The plane should be rolled several times from the maximum bank angle in one direction to the maximum bank angle in the other direction in order to observe the yawing motion. If as the wings pass through level the nose is yawed in the opposite direction to the roll then the KFF_RDDRMIX should be increased until the yaw disappears;

3. The YAW2SRV_DAMP should be increased in increments of 0.05 until the yaw angle starts to oscillate. When this happens, the value of the parameters should be halved;

4. The plane should be rolled into and out of turns in both directions. If it has a tendency to turn the nose to the outside of the turn then the YAW2SRV_RLL should be increased in increments of 0.05. If it has a tendency to turn the nose to the inside of the turn, then this parameter should be reduced in increments of 0.01.

After tunning the yaw damper, the side-slip controller can be also tuned by doing the following steps:

1. In the Mission Planner, the lateral acceleration ay should be ploted in the tuning graph window;

2. The plane must be rolled rapidly from the maximum bank angle in one direction to the maximum bank angle in the other direction in order to observe the lateral acceleration. If the lateral acceleration value is around zero and doesn't change when the aircraft is rolled then no side-slip control is required, and the steps 3 and 4 can be ignored;

3. The YAW2SRV_INT must be set to 1.0. If this value causes the yaw angle to oscillate then the parameter must be halved from the smaller value that causes oscillaton;

4. If the y acceleration is offset or spikes up during turns, then the YAW2SRV_SLIP must be progressively increased in steps of 0.5 until the error disappears or the yaw angle starts to oscillate. If the yaw angle oscillates then the value of this parameter should be halved.

5.4.2 Speed and Height Controller Tunning

The Speed and Height Controller is employed in the TECS (Total Energy Control System) algorithm. Before tunning this controller the Airspeed sensor needs to be correctly calibrated and the pitch controller needs to be tunned as explained in Section 5.4.1. The calibration of the airspeed sensor is explained bellow.

Airspeed Sensor Calibration

The first step to calibrate the Airspeed sensor is to make sure that it is working properly and correctly zeroed. When the APM is started on the plane the electronics should warm up for 1 minute and then a pre-flight calibration of the sensor can be done. The Mission Planner has a menu called *Preflight Calibration*, as shown in Figure 5.3. The airspeed measurement unit needs to be covered to prevent any wind influence and then the **Do Action** button can be pressed and the calibration should take a couple of seconds.

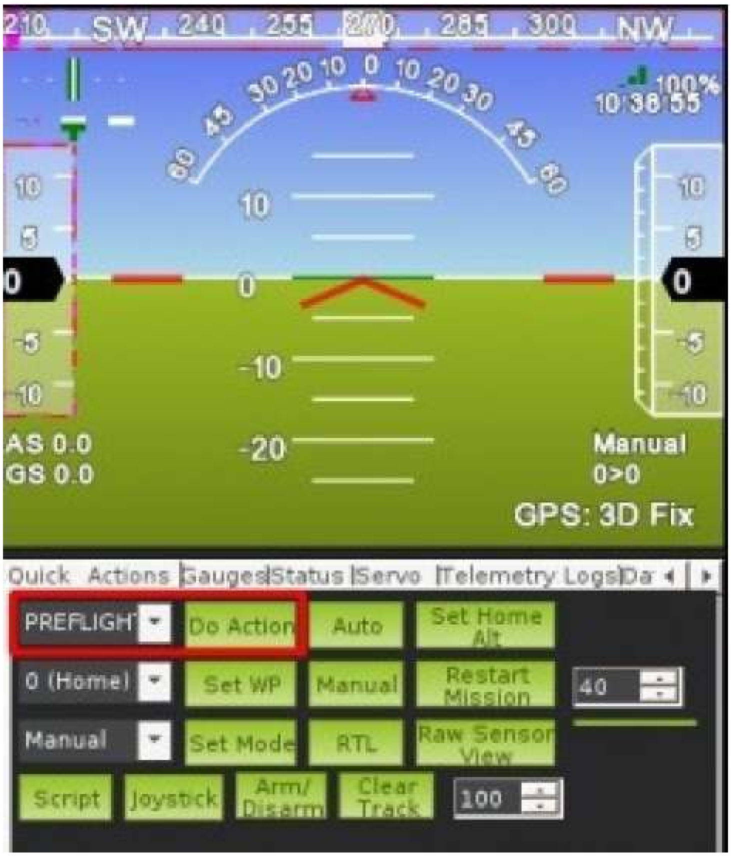


Figure 5.3: Preflight calibration menu.

Next, the airspeed sensor needs to be checked if it is working correctly by blowing in the pitot tube and checking that the AS value in the HUD changes accordingly.

The APM autopilot has an automatic calibration feature that determines the value of the ARSPD_RATIO parameter. This determines how the autopilot maps the differential pressure measured by the airspeed sensor into an airspeed value. To automatically calibrate this value the following steps need to be followed:

1. The ARSPD_AUTOCAL needs to be set to 1;

2. The plane should fly a repeated circuit or loiter for 5 minutes. This can be done in any flight mode. The change in the ARSPD_RATIO can be monitored by refreshing the parameters list;

3. After completing the step 2, the plane can be landed and the ARSPD_AUTOCAL should be set to 0 to prevent any further changes;

4. The ARSPD_RATIO can be checked again and if the value is outside the range between 1.5 and 3.0 then, if possible, the pitot tube should be realocated to reduce the aerodynamic interference from adjacent fuselage or wings.

Setting Initial Parameters

After having a properly calibrated sensor and a well tunned pitch controller, it is necessary for the to set throttle, pitch angle, airspeed and vertical speed limits for getting the TECS algorithm to perform well. In the following tests, the aircraft should be flown at a speed between FWB_ARSPD_MIN and FWB_ARSPD_MAX.

1. The maximum throttle percentage needs to be set in THR_MAX. This should be set to a value that enables the aircraft to climb at its maximum pitch angle at TRIM_ARSPD;
2. The throttle percentage required to fly level at TRIM_ARSPD should be set in the TRIM_THROTTLE parameter. This can be done by testing different throttle settings in FBWA;
3. The maximum and minimum airspeed limits should be set in the ARSPD_FBW_MAX and ARSPD_FBW_MIN parameters. The ARSPD_FBW_MAX should be the the maximum speed that the aircraft flies at THR_MAX. The ARSPD_FBW_MIN should be the slowest speed at which the aircraft can safely turn at a maximum bank angle;
4. The maximum picth angle at which the aircraft can fly with the throttle set to THR_MAX should be set in LIM_PITCH_MAX. This can be done by performing maximum pitch angle climbs in FBWA with the throttle set to the maximum and checking the airspeed during climb. If the airspeed rises above TRIM_ARSPD during the climb, then either the LIM_PITCH_MAX can be increased or the THR_MAX can be reduces. If it falls below ARSPD_FBW_MIN during climb then either the LIM_PITCH_MAX can be reduced or the THR_MAX can be increased;
5. The minimum pitch angle can be set in LIM_PITCH_MIN which is the pitch angle at which the aircraft can fly with the throttle set to THR_MIN without over-speeding;
6. The maximum climb rate can be set in TECS_CLMB_MAX. This is the best climb rate that the aircraft can achieve with the throttle set to THR_MAX and flying at TRIM_ARSPD. This can be measured in FBWA mode by performing climbs to height with the throttle set to THR_MAX;
7. The minimum sink rate can be set in TECS_SINK_MIN. This is the sink rate of the aircraft with the throttle set to THR_MIN and flown at TRIM_ARSPD. This can be measured by closing the throttle in FBWA and gliding the aircraft down from height;
8. The maximum sink rate can be set in TECS_SINK_MAX. If this value is too large, the aircraft can overspeed when descending. This should be set to a value that can be achieved without exceeding the lower pitch angle limit and without exceeding ARSPD_FBW_MAX.

Flight Testing

1. With the aircraft loitering around a waypoint in eitheir AUTO, GUIDED or RTL mode, the height should remain constant without noticeable pitching or height changes greater than 10 m. If the planes oscillates in height then the TECS_TIME_CONST should be increased in increments of 1;

2. The THR_MAX, LIM_PITCH_MAX and TECS_CLMB_MAX can be checked if they are correctly set by commanding a positive altitude change of no less than 50 m in either LOITER, RTL or GUIDED mode. The objective is to set these parameters such that the throttle required to climb is about 80% of THR_MAX, the aircraft is maintaining airspeed, and the demanded pitch angle is about 5 degrees below LIM_PITCH_MAX.
    - If speed drops below the desired value, and the throttle increases to and stays on THR_MAX, then either TECS_CLMB_MAX should be reduced or THR_MAX increased.
    - If the demanded pitch angle is constantly at the limit set by LIM_PITCH_MAX, then either the pitch angle limit LIM_PITCH_MAX needs to be increased or the maximum climb rate TECS_CLMB_MAX needs to be reduced.
3. The LIM_PITCH_MIN and TECS_CLMB_MIN can be verified by commanding a negative altitude change of no less than 50m in either LOITER, RTL or GUIDED mode. The objective is to set these parameters such that the throttle is on THR_MIN, the airspeed is below ARSPD_FBW_MAX, and the demanded pitch angle is about 5 degrees above LIM_PITCH_MIN.
    - If the speed is too high, then TECS_SINK_MAX should be reduced .
    - If the demanded pitch angle is constantly at the limit set by LIM_PITCH_MIN, then either the pitch angle limit LIM_PITCH_MIN needs to be reduced or the maximum sink rate TECS_SINK_MAX needs to be reduced.
4. If the height response oscillates the value of TECS_PTCH_DAMP can be increased in increments of 0.1 and the value of TECS_TIME_CONST can be increased in increments of 1.0;
5. If the airscraft uses an airspeed sensor, the value of TRIM_THROTTLE can be adjusted so that it matches the average amount of throttle required by the controller during constant height loiter.

There is a comprehensive list of parameters that can be used to further fine tune the TECS algoritm. A description of these parameters can be found in [23].

5.4.3 Horizontal Navigation Tunning

Ardupilot uses a navigation controller called L1 Controller [24] that produces accurate flight paths both for waypoint following and loiter. Tunning this controller requires that the NAVL1_PERIOD is correctly adjusted. A smaller value of this parameters will lead to a more agressive navigation, while a larger value will lead to a more gentle navigation. The NAVL1_DAMPING and WP_RADIUS can also be adjusted for further tunning.

To tune the L1 Controller the following steps should be followed:

1. The Roll and Pitch controller and the Height controller need to be previously tunned;
2. The LIM_ROLL_CD should be set to an appropriate value for the bank angle at which the aircraft can fly without stalling;

3. In the Mission Planner, a rectangular mission should be planned. This rectangle should be small enought so that the aircraft can be seen;
4. The NAVL1_PERIOD should be inittially set to 25, the NAVL1_DAMPING should be set to 0.75 and the WP_RADIUS to the distance that the aircraft would fly in 2 seconds at cruise speed;
5. The aircraft should takeoff in MANUAL mode, changing to AUTO mode so that it flies the previously programmed mission;
6. If the plane is turning too slowly then the NAVL1_PERIOD should be reduced by 5. If it is “weaving” after a turn then the NAVL1_PERIOD can be increased by 1 or 2;
7. To get the maximum performance possible, after completing the tuning of NAVL1_PERIOD, it can be increased 1 and then the NAVL1_DAMPING can be increased in increments of 0.05 to get the wanted response.

To control whether the autopilot flies the aircraft throught a waypoint, then turns afterwards or before the waypoint so that it lines up for the next leg of the mission, the waypoint radius should be adjusted in the WP_RADIUS parameter. In order for the aircraft to fly through the waypoint, this parameter should be set to a smaller value. Setting it to a large value will make the aircraft turn before passing through the waypoint, allowing the plane to line up for the next leg of the mission.

5.5 Flight Test Results

The flight testing phase was separated in two parts. The first part consisted of UAV ground tests and the objective was to check if the system was correctly assembled. The second part consisted of flight tests with the objective of analysing the autopilot behaviour and improving its performance.

The first part of the tests were conducted at *Instituto Superior Técnico* in Lisbon. It was made an overall check regarding the stability of the autopilot module, shown in Figure 5.4.

The pitot tube was initially mounted on this platform, but since it would be directly behind the propeller of the UAV, it would have erroneous readings. This was corrected by installing the pitot tube and the pressure sensor in the left wing of the UAV, as shown in Figure 5.5.

After having the system correctly installed on the aircraft, the next ground tests conducted were to check if the control surfaces responded to inputs from the radio controller. To check this, the system was connected to the Mission Planner and the autopilot flight mode was set to MANUAL. Then, all the UAV servos were individually inspected and checked to see how they responded to simple control inputs. It was verified that all the control surfaces and the throttle responded correctly to the control commands sent by the radio controller.

The last step of the ground tests were to verify how the autopilot would stabilize the plane when it was pitched or rolled. To verify this, the autopilot flight mode was set to STABILIZE and the aircraft was rolled and pitched several times. It was verified that the control surfaces would react in order to stabilize the plane. When the aircraft was rolled, the autopilot would use the ailerons to bring the plane back to

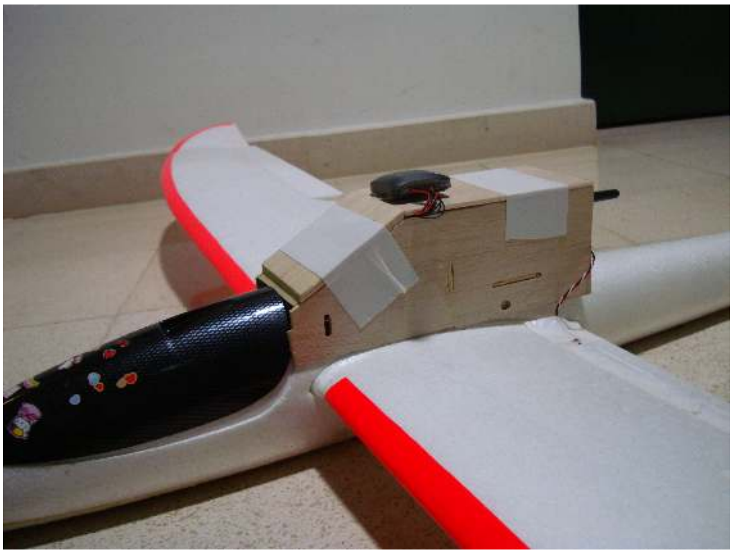


Figure 5.4: UAV autopilot module.

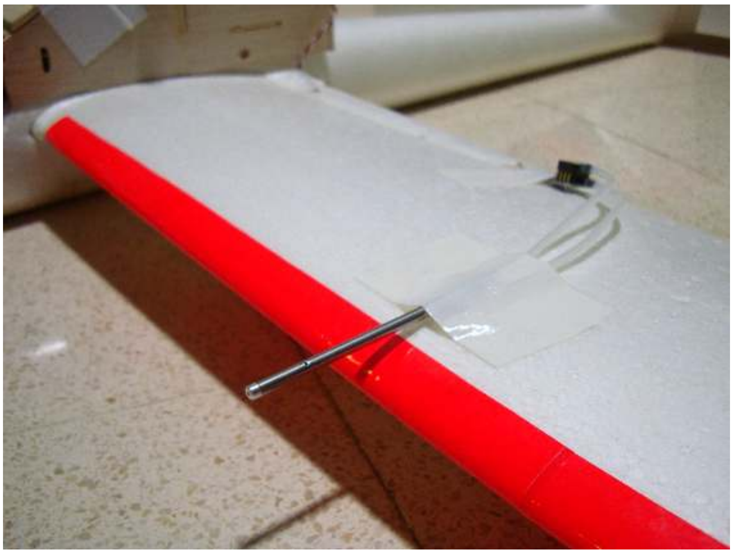


Figure 5.5: UAV pitot tube.

the initial position. When the aircraft was pitched, the autopilot would use the elevator to correct its pitch angle. If there was any sideslip angle, the rudder would be activated.

These tests ensured that the autopilot was correctly installed and that the UAV was ready for the flight test phase.

The flight test phase was conducted at *Base Aérea Nº1* in Sintra. This provided a large runway where the flight tests could be safely conducted. The goal was to test if the autopilot was working correctly in the several flight modes. Before flying the UAV, the autopilot system was properly secured to the aircraft as shown in Figure 5.6.

The first test conducted was to check if there was any interference between the radio controller of the aircraft and the telemetry radio link. The aircraft was manually conducted to a distance of around 200 meters and the control surfaces were activated using the radio controller. It was verified that the aircraft was correctly responding to any control input and that the telemetry information was being correctly

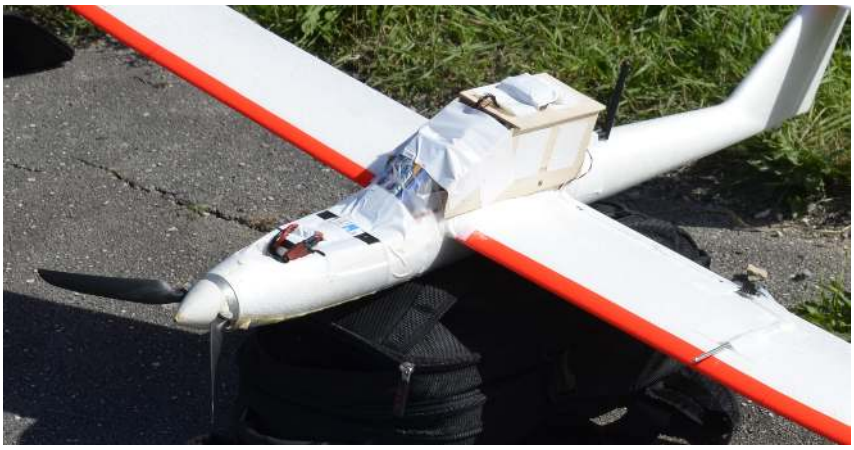


Figure 5.6: UAV test system.

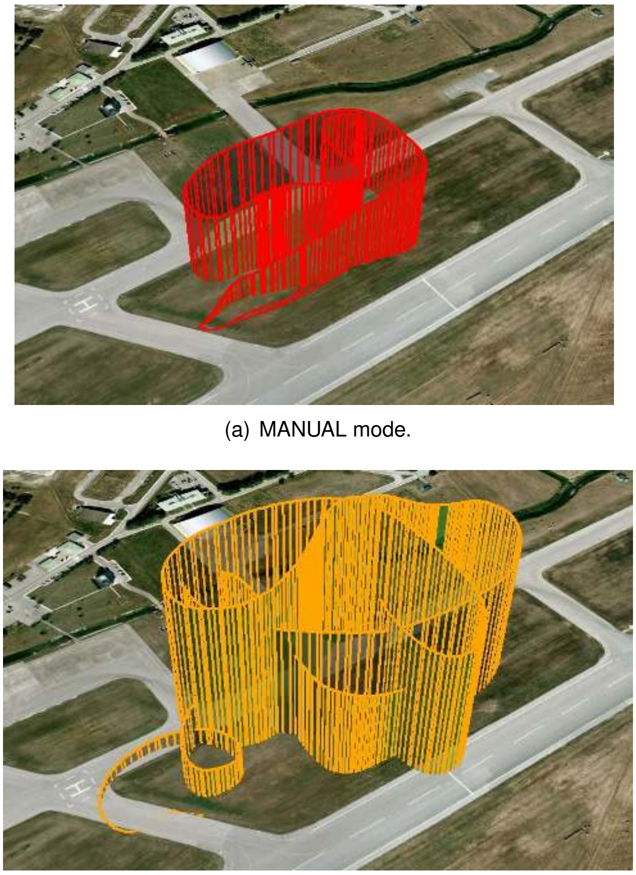
received at the ground station computer.

The aircraft was then flied for the first time. The first flight objective was to analyse the performance of the autopilot in MANUAL mode and in STABILIZE mode. The aircraft was hand lanched in MANUAL mode and it was flown for some time to check its behaviour. The flight path in MANUAL mode is shown in Figure 5.7(a). The autopilot provided full manual control of the aircraft as expected. The flight mode was then changed to STABILIZE. The flight path in this mode is shown in 5.7(b). When there where no control inputs applied to the aircraft, the autopilot automatically leveled the plane. This mode provided a stable flight while also allowing manual control of the UAV.

To test the autopilot AUTO mode it was created a simple rectangular mission shown in Figure 5.8. The aircraft would fly the waypoints at an altitude of 100 meters relative to the ground.

The UAV was hand launched in MANUAL mode and when it was at the correct altitude the flight mode was changed to AUTO. The flight path is shown in Figure 5.11. The yellow flight path is the segment flown in MANUAL while the green flight path is the AUTO mode segment. When the AUTO mode was triggered the aircraft immediately flew to the first waypoint of the mission. It then followed the mission without any issue. When the aircraft completes the mission it returns to its HOME position and it will loiter above that position. Figure 5.9 displays the distance to the next waypoint of the mission. The aircraft starts at around 95 meters from the first waypoint and when it reaches a distance of around 20 meters of the waypoint it will compute the distance to the next waypoint. This happens for the four waypoints of the mission and for the Home position too. At the third waypoint, the autopilot changed its course to the last waypoint at a distance of 35 meters. In Figure 5.10 it is shown the distance of the UAV relatively to the waypoint mission. It only starts to compute this distance when it passes the first waypoint because that is where the track starts. It then deviates from the track reaching a track error of more than 60 meters. This was expected as the UAV only had to pass near the waypoints and not follow the rectangular course. It can be seen that when it passes near the second waypoint the position error is smaller than five meters. When it passes through the third and fourth waypoints the track error is very small.

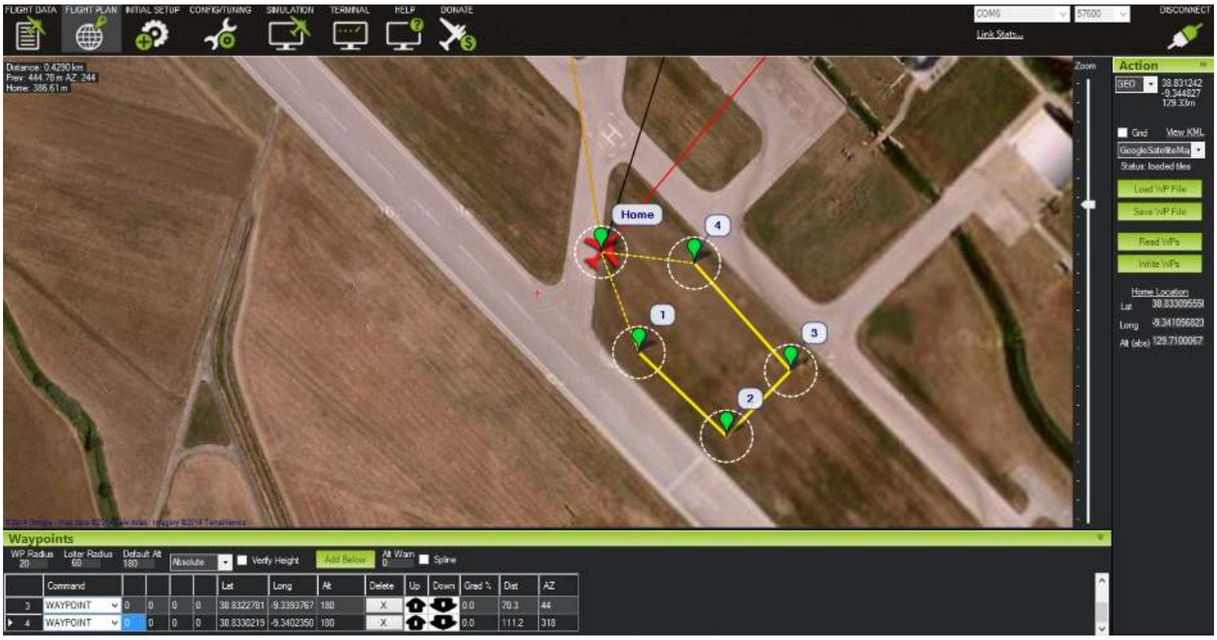
The last test flight was to analyse the behaviour of the autopilot in LOITER and in GUIDED mode.



(a) MANUAL mode.

(b) STABILIZE mode.

Figure 5.7: Flight paths - first flight.



Distance: 0.4290 km
Prev: 444.78 m AZ: 244
Home: 386.61 m

GEO 38.831242
-9.344827
129.33m

□ Grid

GoogleSatelliteMa

Status: loaded tiles

Home Location
Lat 38.83309555l
Long -9.341056823
Alt (abs) 129.7100067

WP Radius 20 Loiter Radius 60 Default Alt 180 Absolute □ Verify Height Alt Warn 0 □ Spline

	Command					Lat	Long	Alt	Delete	Up	Down	Grad %	Dist	AZ
3	WAYPOINT	0	0	0	0	38.8322781	-9.3393767	180	X	⬆	⬇	0.0	70.3	44
▶ 4	WAYPOINT	0	0	0	0	38.8330219	-9.3402350	180	X	⬆	⬇	0.0	111.2	318

Figure 5.8: UAV flight test mission.

The aircraft was hand launched in MANUAL mode and when it was at the desired altitude and at the desired location it was changed to LOITER mode. The flight path in LOITER mode is shown in Figure 5.12(a). The UAV loitered around the initial point where it was set to loiter. The radius of the loiter was

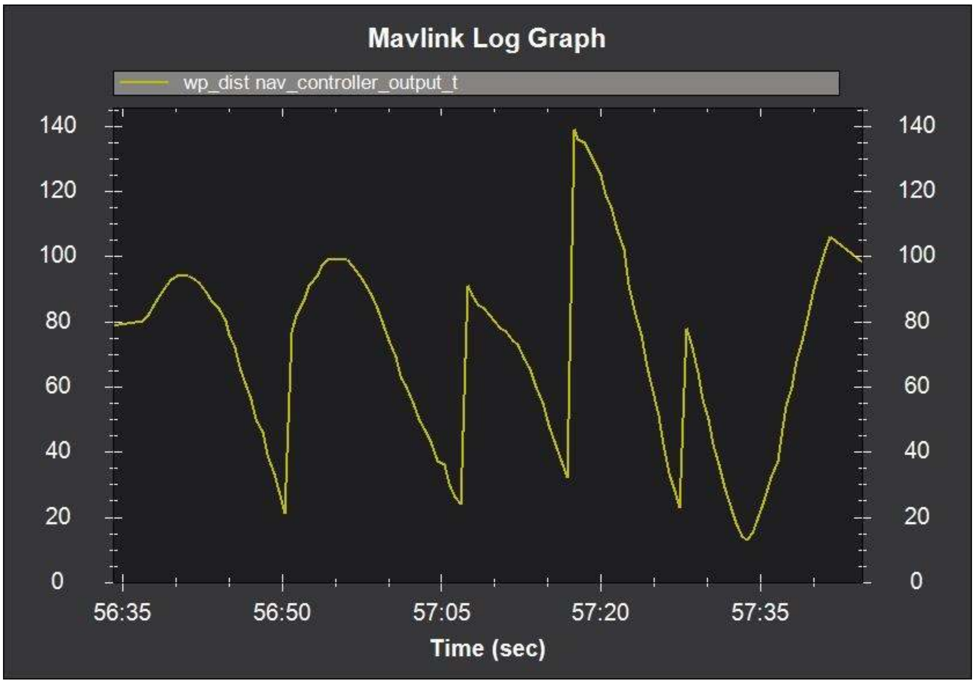


Figure 5.9: Waypoint distance (in meters).

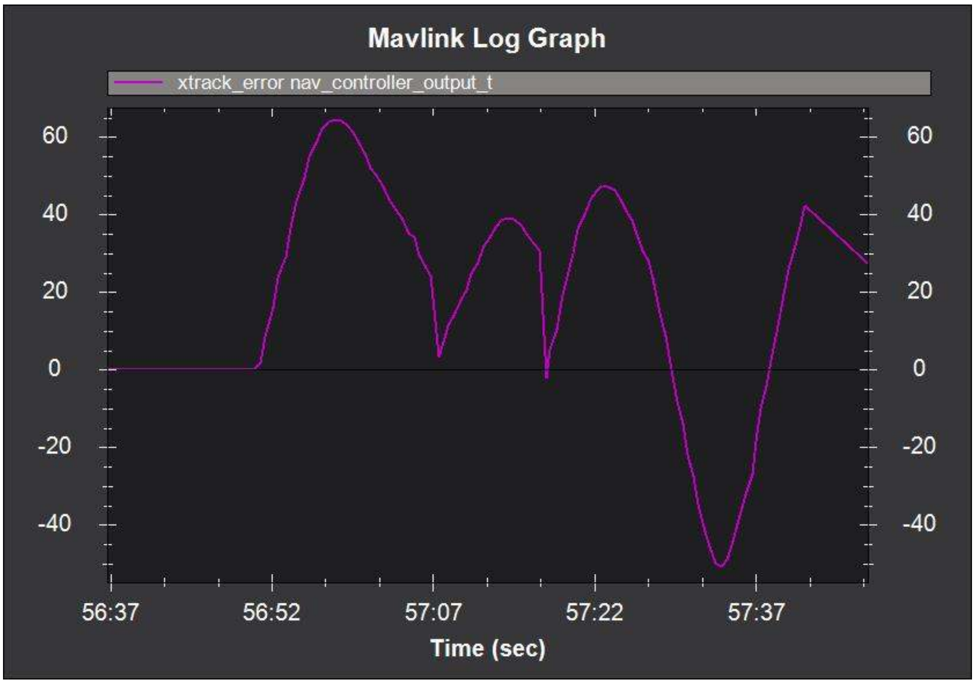


Figure 5.10: Track position error (in meters).

set in the autopilot parameters to 40 meters. The flight mode was then changed to GUIDED. This mode allowed the aircraft to be controlled with the ground station computer. The aircraft was sent to a location and it was seen that it would loiter around that position as soon as it reached it. This is shown in Figure 5.12(b).

There are several flight parameters that can be analysed using the Mission Planner log playback feature. Figure 5.13 shows the absolute alitude of the aircraft for each test flight. In the first flight the

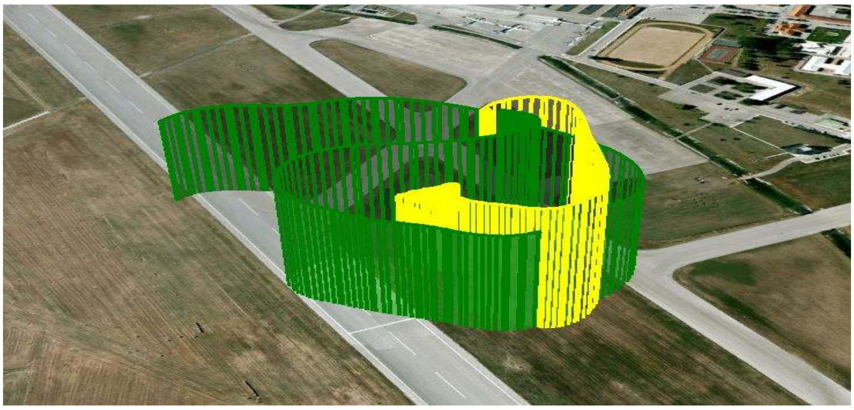
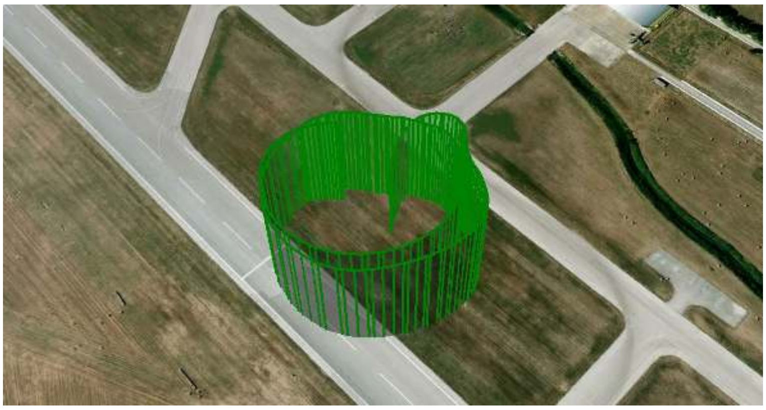
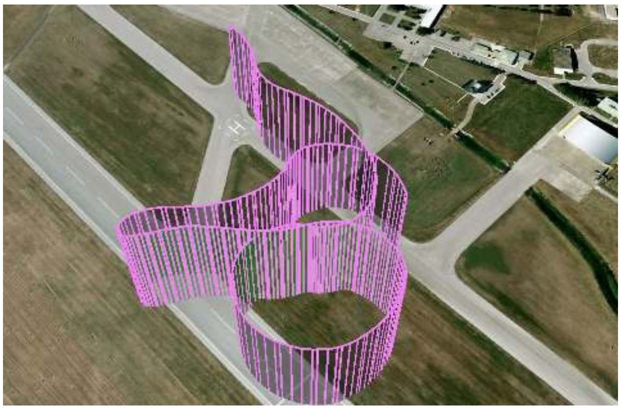


Figure 5.11: AUTO mode flight path.



(a) LOITER mode.

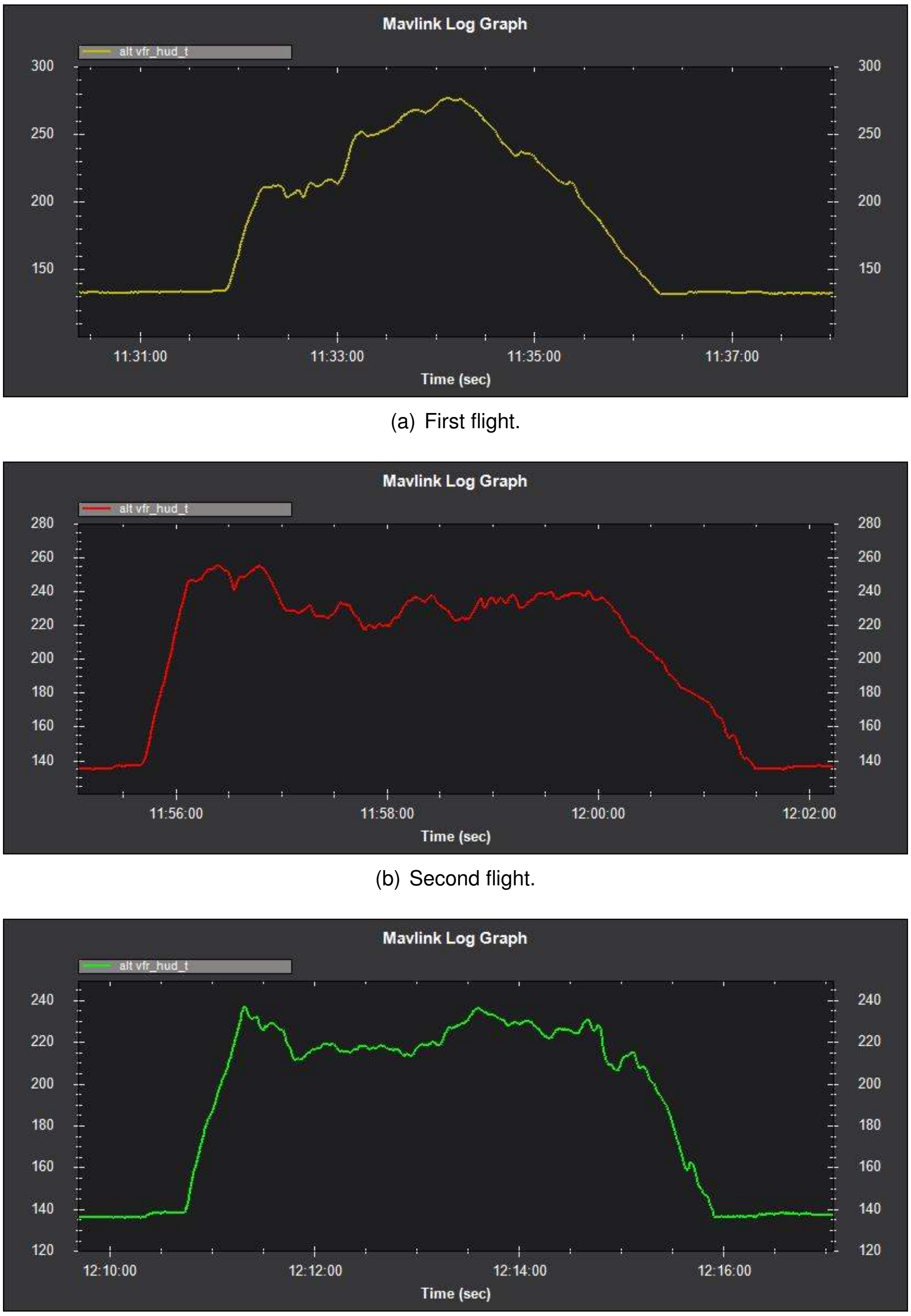


(b) GUIDED mode.

Figure 5.12: Flight paths - third flight.

flight altitude varies throughout the flight due to the aircraft being flown in MANUAL mode. On the other two test flights, it can be seen that the aircraft maintains its altitude when flying in AUTO, LOITER and GUIDED mode. This is remarkable as the autopilot was using the default parameters settings.

Figure 5.14 displays the airspeed, groundspeed and windspeed profiles for each flight. The autopilot estimates the groundspeed based on the GPS information. The airspeed is measured using the pitot tube and the windspeed is then calculated based on the direction and difference of these two speeds. By analysing the three flights it can be seen that the airspeed remains relatively constant throughout each flight. The groundspeed has sudden variations due to the direction of the flight relatively to the



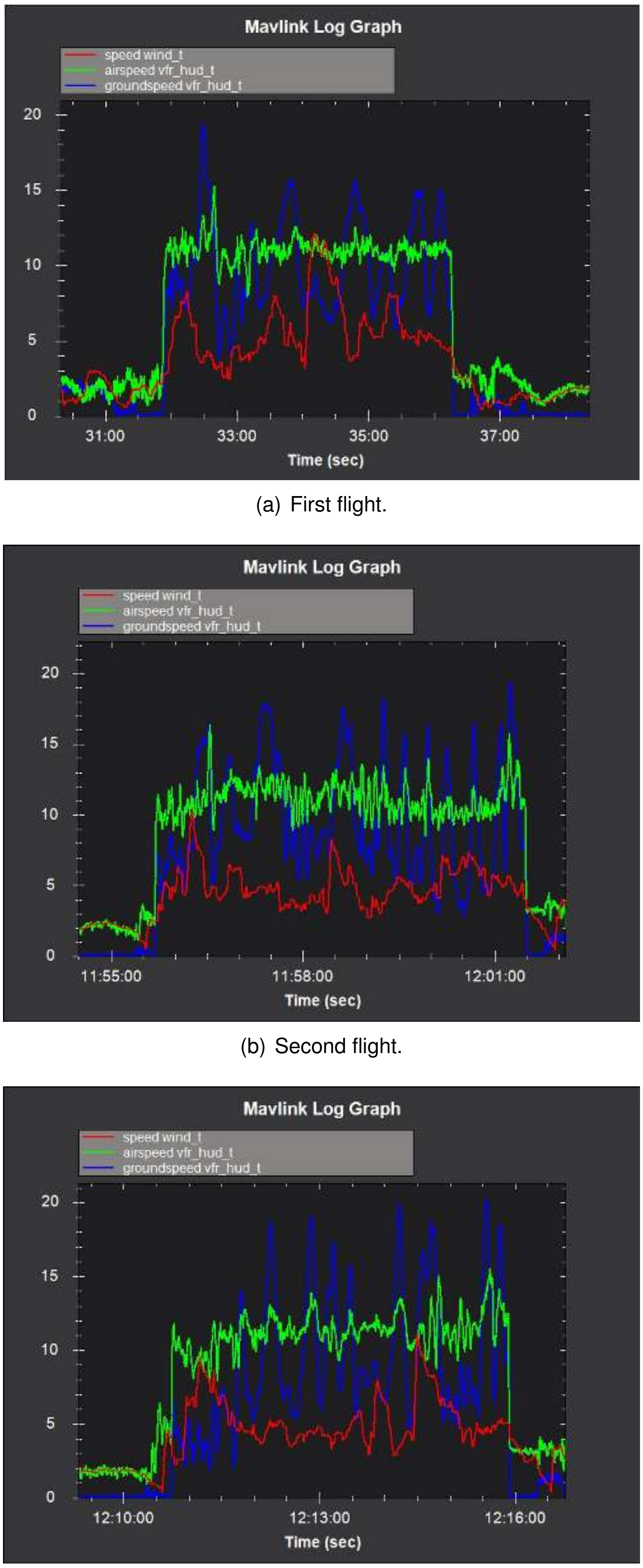
(a) First flight.

(b) Second flight.

(c) Third flight.

Figure 5.13: Test flights altitude profile. The alitude is displayed in meters.

wind.



(a) First flight.

(b) Second flight.

(c) Third flight.

Figure 5.14: Test flights airspeed (green), groundspeed (blue) and windspeed (red) profile. The speeds are displayed in m/s.

Chapter 6

Conclusions

In this chapter, a summary of the achievements of this work is presented in Section 6.1. In Section 6.2 some future works are considered.

6.1 Achievements

In this thesis, it was analysed and implemented an autopilot solution for an UAV. It was found that an open source autopilot can be used to transform a simple commercial radio controlled vehicle into an autonomous, remotely piloted vehicle at a low cost. It was also analysed the ground control station software that is provided by this autopilot. A RPV system was successfully implemented. This system was usefull as it provides realtime video from the perspective of the vehicle.

Ardupilot APM 2.6 showed its great flexibility between different vehicle platforms. This allowed the autopilot to be tested in a rover before being tested in the UAV. The Mission Planner software provided a complete ground control station. It displayed detailed information about the vehicle data, the creation of mission profiles was easilly accomplished, and the configuration of the autopilot parameters was straightforward. It also allowed the analysis of flight data by using the recorded flight logs.

The rover ground test phase proved to be an important step of the work. It provided a more detailed knowledge of how the system was set up. This was a more secure way to test the several system components before installing them on the UAV. After correctly tunning the autopilot, the rover was capable of automatically following missions that were created in the Mission Planner. This phase also showed that the RPV system interfered with the rover radio receiver.

The flight test phase showed that the autopilot was correctly configured and installed on the aircraft. It proved to be working correctly in the several flight modes even with default parameters. This test showed that the autopilot was ready to be installed in the UAV that is being developed.

6.2 Future Work

Naturally, the next phase of this work, is to install the autopilot on the LEEUAV, once the project is complete, and tune it according to what is described in this thesis.

It would be interesting to further develop the RPV system of the UAV. It could be either by developing a platform that integrates the video feed in the Mission Planner, or by using a gimbal that would allow the user to control the camera position.

It would also be interesting to study and analyse the autopilot performance regarding other vehicle platforms such as rotorcraft.

Bibliography

[1] C. Dubrin. UAV market worth $8.3 b by 2018. *Microwave Journal*, page 37, August 2013.

[2] J. T. K. Ping, A. E. Ling, T. J. Quan, and C.Y. Dat. Generic unmanned aerial vehicle (uav) for civillian application. *IEEE Conference on Sustainable Utilization and Development in Engineering and Technology*, October 2012.

[3] J. R. Wilson. Uav roundup 2013. *Aerospace America*, pages 26–36, July 2013.

[4] Us drone flights hit technical and military turbulence, July 2014. `http://rt.com/usa/169624-drones-us-faa-issues`.

[5] A. C. Marta. Long Endurance Electric UAV. Technical report, Instituto Superior Técnico, 2013.

[6] US Department of Defense. *Unmanned Systems Integrated Roadmap FY2013-2038*, December 2013.

[7] R. Austin. *Unmanned Aircraft Systems: UAVs design, development and deployment*. Wiley, 1^{st} edition, 2010. ISBN:9780470058190.

[8] General Atomics Aeronautical. Predator XP brochure. `http://www.ga-asi.com/products/aircraft/pdf/PredatorXP.pdf`.

[9] Joe Mcnally. `http://ngm.nationalgeographic.com/2013/03/unmanned-flight/mcnally-photography?source=podrelated#/1`.

[10] S. Cheng. Rapid deployment uav. *Aerospace Conference, IEEE*, March 2008.

[11] Historic Wings. George The Autopilot, August 2012. `http://fly.historicwings.com/2012/08/george-the-autopilot`.

[12] K. Fitchard. Consumers Tell Cisco They're Comfortable With the Driverless Car, May 2013. `http://www.businessweek.com/articles/2013-05-15/consumers-tell-cisco-theyre-comfortable-with-the-driverless-car`.

[13] H. Chao, Y. Cao, and Y. Chen. Autopilot for small unmanned aerial vehicles: A survey. *International Journal of Control, Automation and Systems*, 2010. doi:10.1007/s12555-010-0105-z.

[14] UAV Navigation. Vector Datasheet, 2013.

[15] Micropilot. Mp2028^{g} datasheet.

[16] Procerus Technologies. Kestrel Autopilot v2.4, April 2014.

[17] CloudCap. Cloud Cap Technology Piccolo SL, January 2014.

[18] APM Open Source Autopilot, March 2014. http://ardupilot.com/.

[19] 3D Robotics. APM 2.6. http://3drobotics.com/wp-content/uploads/2013/07/APM-2.6-web-version.pdf.

[20] Openpilot, April 2014. http://www.openpilot.org/.

[21] Video Receiver - Electronica RC, . http://electronicarc.com/catalogo/product_info.php?cPath=43_49_70&products_id=454&osCsid=33bee22c4cfbfa05d1d1ca2a42be156d.

[22] Video Transmitter - Electronica RC, . http://electronicarc.com/catalogo/product_info.php?cPath=43_49_70&products_id=420&osCsid=33bee22c4cfbfa05d1d1ca2a42be156d.

[23] APM Speed and Height Controller Tuning Guide, . http://plane.ardupilot.com/wiki/tecs-total-energy-control-system-for-speed-height-tuning-guide/.

[24] APM Navigation Tuning Guide, . http://plane.ardupilot.com/wiki/navigation-tuning/.