DAS Departamento de Automação e Sistemas
CTC Centro Tecnológico
UFSC Universidade Federal de Santa Catarina

A Fixed-wing UAV Capable of Vertical Take-off and Landing for Aerial Mapping and Photogrammetry.

Relatório submetido à Universidade Federal de Santa Catarina

como requisito para a aprovação da disciplina:

DAS 5511: Projeto de Fim de Curso

Willian de Medeiros Galvani

Florianópolis, Fevereiro de 2018

A Fixed-wing UAV Capable of Vertical Take-off and Landing for Aerial Mapping and Photogrammetry.

Willian de Medeiros Galvani

Este relatório foi julgado no contexto da disciplina

DAS 5511: Projeto de Fim de Curso

e aprovada na sua forma final pelo

Curso de Engenharia de Controle e Automação

_____

Prof. Ubirajara Franco Moreno

Orientador

Banca Examinadora:

<div align="right">

João Marcelo Corrêa
Orientador na Empresa

Prof. Ubirajara Franco Moreno
Orientador no Curso

Prof. Hector Bessa Silveira
Responsável pela disciplina

Marcelo De Lellis Costa de Oliveira
Avaliador

Bruno Ferreira Fontana
Debatedor

Wesley André Bortolozo Junior
Debatedor

</div>

Acknowledgements

To my family, for supporting through the years it took for me to get here.

To my girlfriend Mariana, for putting up with me the whole time, as I skipped many fun things to work instead.

To the professors and staff of the Department of Automation and Systems, for the knowledge and for enabling me to get where I am today.

To my friends, for making the journey so far more bearable.

To ProVANT, for enabling me to do what I love, and introducing me to Robota.

To Robota, for being most of the friends mentioned above, and enabling me to develop many projects I wouldn't in other ways. Also for forcing me to take a leadership position eventually, and developing my teamworking and project management skills.

To Novarum Sky, for the opportunity of developing this project.

To Juliano, who got me into Novarum Sky, worked through most people's vacations with me, and helped me through the prototype's testing and development.

To Patrick, for all the projects we developed together, both in ProVANT, Robota, IbexCPS, and more.

To Rafael, for helping me with all the questions regarding aerodynamics.

To my sister, Kamila, for reviewing this document at short notice.

Fly, you fools!

<div style="text-align: right">

J. R. R. Tolkien

</div>

Resumo

Mapeamento aéreo é uma das tarefas que foi revolucionada com a chegada dos drones nos ultimos anos. O trabalho manual de tirar fotos, organizá-las e juntá-las mudou para colocar coordenadas em um software, e as fotos resultantes em outro para o pós-processamento após o voo.

Dependendo da tarefa em questão, o operador pode escolher utilizar multirotores para áreas menores, ou uma aeronave de asa fixa para as maiores. Enquanto multirotores são precisos e podem pousar/decolar de virtualmente qualquer lugar, sua autonomia sofre, uma vez que todo o empuxo para mantê-los em voo é gerado diretamente pelas hélices. Aeronaves de asa fixa, por outro lado, podem cobrir grandes áreas rapidamente com um consumo energético menor, mas são mais difíceis de posicionar e podem requerer dezenas de metros para pouso e decolagem.

Este trabalho propõe o desenvolvimento de uma aeronave entre estes dois mundos. O protótipo projetado é uma aeronave de asa fixa *tail-sitter*, capaz de decolar na vertical como um multirotor e transicionar para o modo de voo asa fixa para maior eficiência, habilitando a cobertura de grandes áreas sem necessitar de aparatos adicionais para pouso e decolagem, nem de amplos espaços.

No teste realizado, foi comprovada a capacidade de decolagem e pousos verticais, no entanto não foi possível testar pousos e decolagens autônomos, tampouco transição e voo horizontal, por limites de espaço e tempo. Apesar dos resultados parciais serem positivos, mais testes serão conduzidos até a finalização do produto.

Palavras-chave: tail-sitter, aerofotogrametria, VANT.

Abstract

Aerial mapping is one task that got revolutionized by the arrival of drones in the latest years. The manual job of taking pictures, printing and assembling them together was changed into putting coordinates into a software, and the pictures into another one after the flight.

Depending of the task at hand, the operator can choose a multirotor for smaller areas, or a fixed-wing aircraft for larger ones. Both categories have their quirks: while multirotors are precise and can take-off/land virtually anywhere, their autonomy suffers as they generate all their lift by using propellers; Fixed-wing aircraft, on the other hand, can cover large areas quickly with lower power consumption, but are harder to position, and require larger areas for take-off and landing.

This work proposed an aircraft in between these two worlds. The prototype designed is a tail-sitting fixed-wing aircraft, able to take-off as a multirotor and transition into fixed-wing mode for more efficiency, enabling it to cover larger areas while needing a small area for take-off or landing and no additional apparatus for take-off.

On the test performed, the VTOL capability was verified, however it was not possible to test autonomous take-offs and landings, nor transition and fixed-wing flight, due to time and space limitations. While the results observed are promising, more tests are required to finish the product.

Keywords: tail-sitter, aerophotogrammetry, UAV.

List of Figures

List of Tables

List of abbreviations and acronyms

VTOL   Vertical Take-off and Landing

UAV   Unmanned Aerial Vehicle

CAD   Computer-Aided Design

GPS   Global Positioning System

GIS   Geographic Information System

RTK   Real Time Kinematics

PPK   Post Processed Kinematics

DIY   Do It Yourself

AVR   Atmel's microontroller architecture

DJI   Comercial UAV retailer

GLONASS   Russian Global Positioning System

UART   Universal Asynchronous Receiver-Transmitter

Contents

1 Introduction

1.1 Novarum Sky

Novarum Sky is a still young company, created in 2014 and based in Florianópolis-Brazil. It develops technologies for both manned and unmanned aerial systems, including long-range digital audio/video transmission solutions, real time kinematics for precise localization during inspections and mapping, and inspections systems themselves.

The company was featured on Web Summit 2017 Lisboa, and has its main partners currently in Europe, with ongoing negotiations with MikroKopter and EDP.

1.2 Motivation

Technology and automation have been changing and improving many tasks in last few decades. One of the tasks is aerial mapping, which started with balloons, then manned airplanes, and now, for smaller areas, is being increasingly done with drones [2].

For the company, this project might mean a new innovative product, as it has both advantages of fixed-wing and rotary-wing aircraft. Such product means there is no need for long landing stripes, nor for relatively expensive equipment such as landing parachutes. The competition is also low, as fixed wings are currently a niche market, corresponding to around 3% of the photogrammetry solution by DroneDeploy [3].

In the context of Automation and Control Engineering, this project entails most of the areas discussed, such as mechanics, electronics, manufacturing, fast prototyping, and control of dynamic systems.

1.3 Objectives

The final objective of the work is to have a working prototype of a VTOL fixed-wing UAV able to autonomously take off vertically, transition into fixed-wing mode, follow a planned path taking pictures, transition back into hover mode, and land autonomously. It is planned to have a smaller prototype to test and tune the hover mode before testing the larger, heavier and more powerful final prototype, for safety and practicality reasons. The possible on-board electronic flight controllers will be briefly described and one of them chosen. An overview will be given of the control systems in place and their tuning. The requisites for the job will be gathered, and the electromechanical structure designed and built around it. It is expected that the prototype fulfills the hole between rotary-wing

and fixed-wing aircraft by being able to land in tight spaces, while having a performance close to that of fixed-wing aircraft.

1.4   What Is Out There?

Right now the photogrammetry and mapping sector seems to be taken by multi-rotors, which compose 97% of the drones registered on DroneDeploy [3], a cloud-based photogrammetry software.

In Brazil, a few companies have fixed-wing aircraft for photogrammetry, such as Horus Aeronaves and Nuvem UAV. The drones however work only in fixed wing mode, and the larger ones (1.7 m on Horus' Verok, 2 m on Nuvem UAV's Batmap) need parachutes for landing.

Internationally, Wingtra seems to be the state-of-the-art solution on tail-sitting VTOL aircraft. Its drone, WingtraOne, is able to take-off, fly and land autonomously, and comes bundled with all the required equipment and software required for the photogrammetry. The Wingtra project was born at the Autonomous Systems Lab, at ETH Zurich in Switzerland, where a lot of research was done lately on such kinds of VTOL aircraft [4]. The goal of this project is to get close to the performance and usability of Wingtra.

1.5   Requisites

For the design, a few conditions have been imposed by the available material and desired performance:

- The flight time should be of atleast 1 hour.

- The cruise speed should be around $15m/s$ (around 50 km/h).

- The batteries used will be 6s lithium-polymer packs of 4500 mAh (2 in parallel) or a 4s pack of 10000 mAh, as they are available on the company.

- The motors should preferably be the ones already in use at the company, MK3538, MK3638, or MK3644

- The UAV must be able to take of and land autonomously.

1.6   Structure

The rest of this document is structured as follows. Chapter 2 describes the fields of aerial mapping and photogrammetry. Chapter 3 delves into the flight mechanics and the

UAV's mechanical design. Chapter 4 shows the electronics involved. Chapter 5 explains the involved software. Chapter 6 shows the control structure and its tuning. Chapter 7 demonstrates the work to build the prototypes. Chapter 8 details the validation process, the tests performed, and the results obtained. Chapter 9 closes the document with the conclusions.

2 Aerial Mapping and Photogrametry

2.1 The need for mapping the land

The first known map (actually a painting of a city) dates back to the 7th millennium BCE, [5], while the oldest surviving world maps are from 9th century BCE Babylonia [6].

In the past, maps were used mostly for localization and navigation, and were made without special tools, mainly by sight. During the Age of Exploration (XV-XVII), new tools such as the sextant and magnetic compass helped improve their accuracy, but they remained as a navigational tool.

In the last centuries, maps began being used to precisely map properties, natural landscapes, and cities, and used as a tool of government [7]. Mapping properties, for example, requires high dimensional accuracy, which is hard to get with regular tools. This is usually the job of land surveyors, professionals who use a multitude of equipment, such as total stations, robotic total stations, GPS receivers, retro reflectors, 3D scanners, radios, handheld tablets, digital levels, subsurface locators, drones, GIS, and surveying software.

2.2 Aerial Mapping

Aerial mapping consists of using photographs taken from the air, usually with the camera facing straight downwards, correcting the perspective transformation, and assembling them into an orthomosaic, as seen in figure 1.

2.3 Aerophotogrammetry

Aerophotogrammetry takes the job one step further. By knowing the camera's intrinsic parameters, software is capable of matching a number of pictures, detecting features on the environment, and locating the point used to take each of the pictures, a process known as multi-view stereo. With this information, it is possible to rebuild in 3D most of the environments, enabling the operator to interact with the area as a 3D mesh. By using precise GPS information(such as RTK/PPK[1] data, or total stations) or known landmarks, it is possible to accurately measure distances, areas, volumes, angles and elevations, simplifying the surveyors' job. Aerophotogrametry can also be used to rebuild in 3D buildings and other structures, enabling precise calculations of volume and distances,

---

[1]   Real-Time Kinematics and Post-Processing Kinematics, two techniques for increasing GPS accuracy.

Figure 1 – Orthomosaic. source: Indonesian Redcross/OpenAerialMap

allowing the use of 3D models on CAD software for faster and easier construction and planning. It allows, for example, the calculation of displaced volume in a quarry, or how much landfill is required to level some terrain.

The results of an open-source multi-view stereo pipeline implementation using openMVS [8] and openMVG [9] can be seen in figures 2 and 3. In figure 2 the software shows the cameras found, and their relative positions on the map. The orange areas are locations not covered by the cameras. It is important to notice that, as the coverage does not catch every angle of the structures, some deformations are expected, especially in hidden areas. Figure 3 shows the rebuilt and textured 3D model.
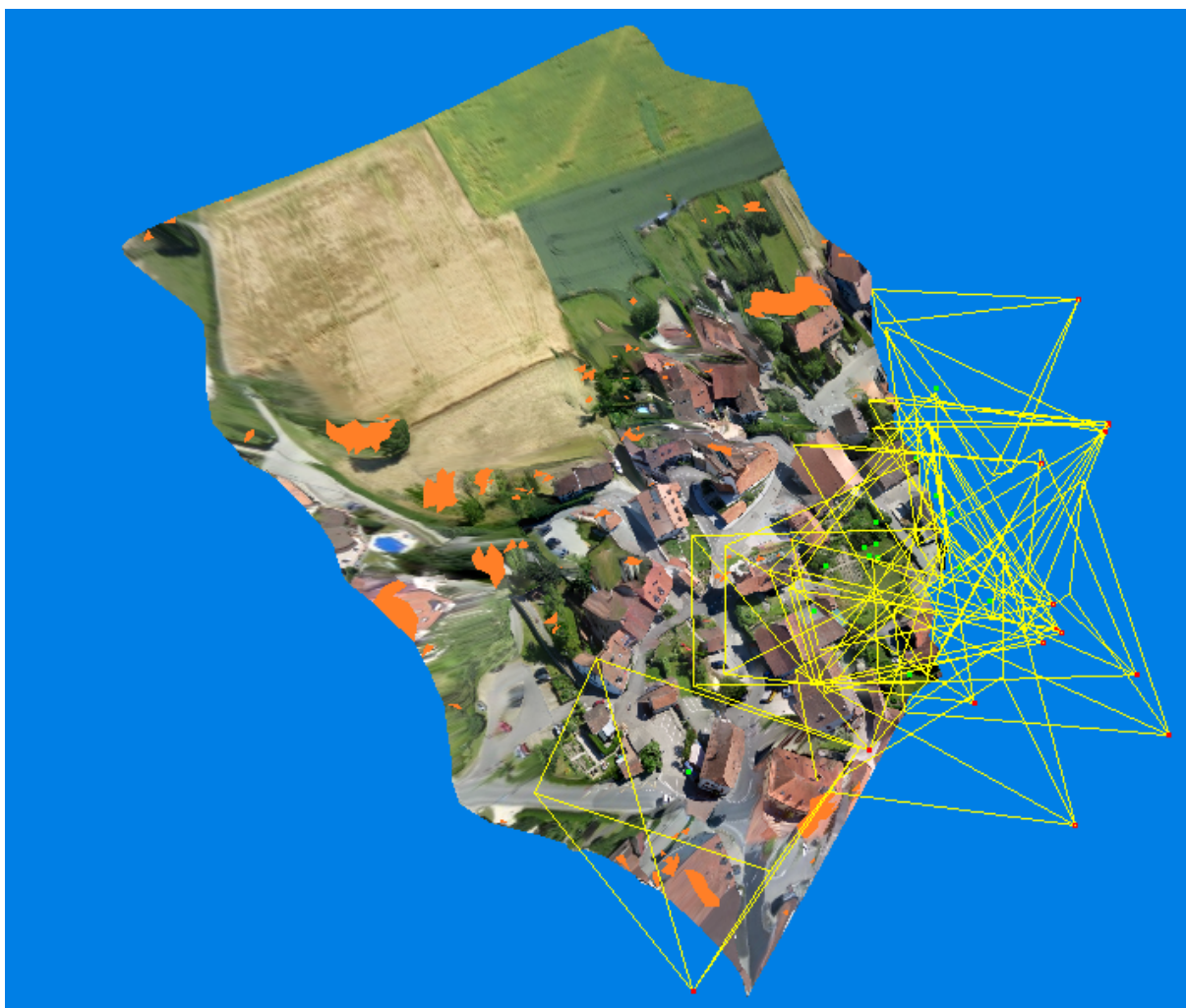
Figure 2 – Identified camera positions on "Oblique mapping of a village" dataset [1]. Source: original
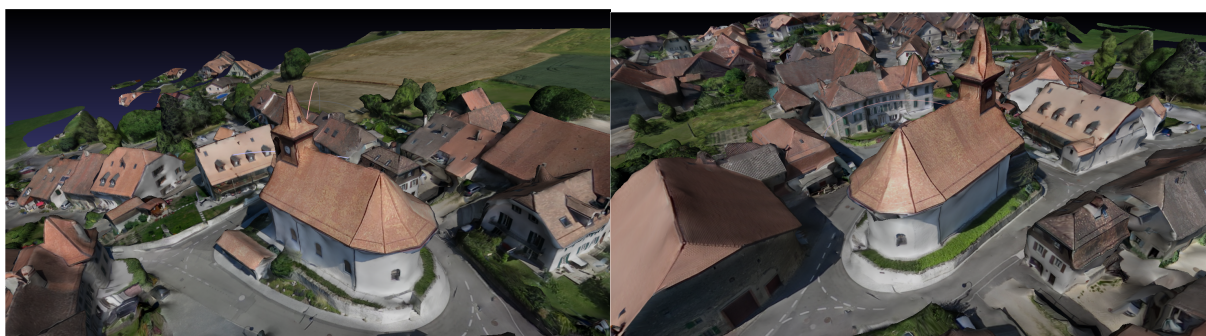
Figure 3 – 3D reconstruction of the "Oblique mapping of a village" dataset [1]. Source: original

3 Flight Mechanics and Design

3.1 Brief Introduction to Flight Mechanics

In order to achieve proper flight, a vehicle needs an upwards force and means of maneuverability. The former is usually generated by the means of a propeller, while the latter can be either the result of spinning propellers or using control surfaces to deflect the passing air's movement, causing a force towards the opposite direction.

Flight mechanics is a field of mechanics which deals with the study of vehicle trajectories, performance, stability, and aerodynamic control.

3.2 Fixed-Wing Mechanics

In fixed-winged aircraft, the air flowing through the wings generates a pressure differential, which lowers the pressure on top of the wing and generates a force called lift. This force is responsible for canceling the gravitational pull and keeping the vehicle aloft in the air.

In a simplified explanation, two main principles are responsible for generating lift:

- Flow deflection and Newton's laws

  Most wings have an angle of attack (to be hereafter called α) such that $\alpha > 0$, which means the air passing through it gets deflected down. According to Newton's second law, an opposite force is necessary on the wing. That force is the generated lift.

- Increased flow speed and Bernoulli's principle

  Bernoulli's principle states that within a steady airflow of constant energy, when the air flows through a region of lower pressure it speeds up and vice versa. This implies there is a direct mathematical relationship between the pressure and the speed, meaning if one knows the speed at all points within the airflow, one can calculate the pressure and vice versa. For a cambered airfoil (where the chord at the top is longer that the chord at the bottom) the air needs to take a longer path, moving faster, thus lowering the pressure on the top, and generating lift.

3.2.1 Airfoil Shape

How much lift is generated depends on the chosen airfoil. A cambered airfoil (longer chord on the upper surface than in the lower one) generates lift even when the angle of attack α is zero. Symmetric airfoils need a positive angle, and the lift is generated by deflecting the air downwards. Other properties that depend on the airfoil shape are the drag (air force pushing against the direction of movement) and angular moment it creates on the aircraft.

3.2.2 The Coordinate System and Nomenclature

The coordinate system, when dealing with the fixed-wing and VTOL modes, is as shown in figures 4 and 5.
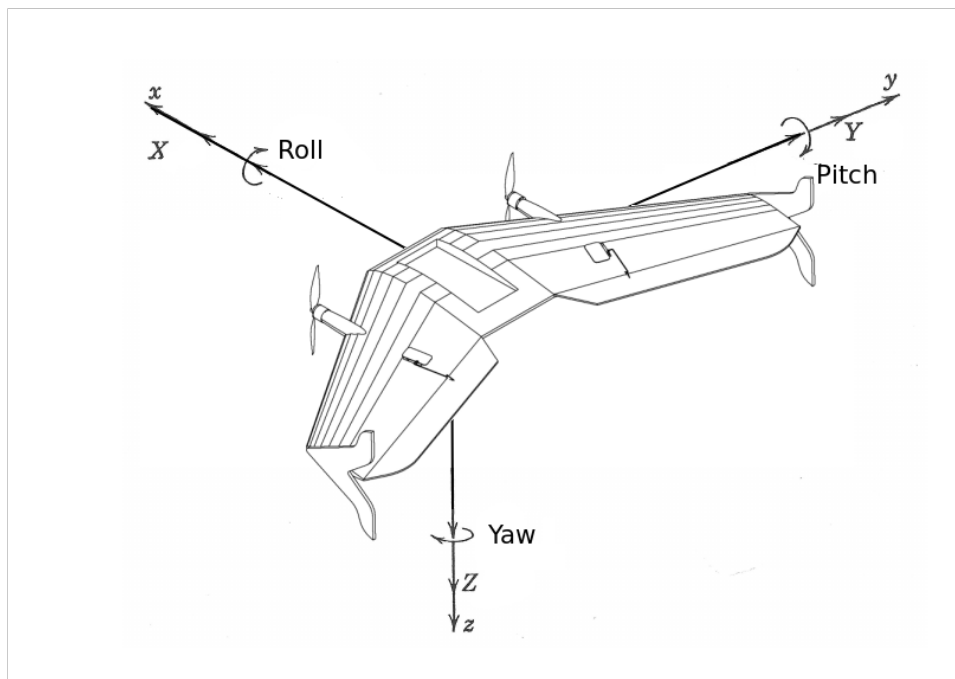
Figure 4 – Body coordinates system and relevant variables in fixed wing mode. Source: original

Where:

- x, y, and z are the coordinates, with the origin in the vehicle's center of mass.

- X, Y, and Z are the components of the aerodynamic force in each of the x, y, and z coordinates, respectively.

- Roll, Pitch and Yaw respectively represent the rotations around the X, Y, and Z axis.

- Between VTOL and Fixed Wing mode, Yaw is Switched with Roll, as the controls switch to Multicopter mode, which usually have the rotors parallel to the ground.
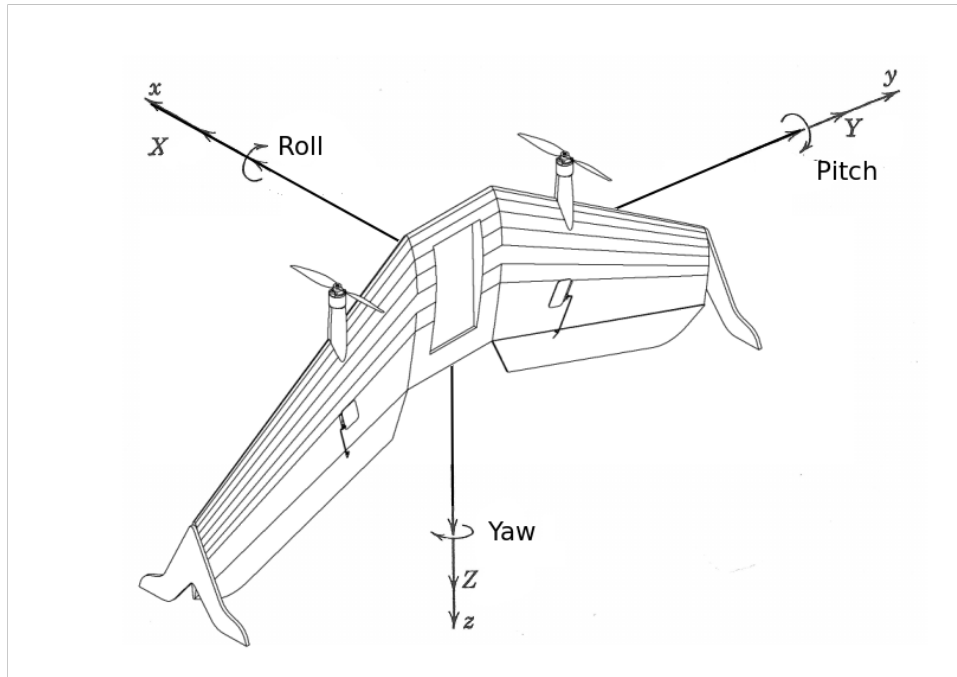
Figure 5 – Body coordinates system and relevant variables in VTOL mode. Source: original

3.3 VTOL Mechanics

When in VTOL mode, the coordinate system used is similar to that of a conventional multirotor, with Z pointing up parallel to the motors' axis, and X going through the fuselage, pointing away from the belly of the aircraft.

The mechanics involved in vertical take-offs and landings are slightly different. The lift generated becomes meaningless, no more than a small perturbation to the system. The generated thrust becomes directly responsible for vertical motion and roll control, while the control surfaces can redirect the airflow allowing control of yaw and pitch.

An approximate model can be seen in [10], where a wind-tunnel test was used to find the parameters. As this work does not focus on the dynamics or control itself, it is not detailed here.

3.4 XFLR5

XFLR5 [11] is an analysis tool for airfoils, wings and planes operating at low Reynolds Numbers. It includes:

- XFoil's Direct and Inverse analysis capabilities;

- Wing design and analysis capabilities based on the Lifting Line Theory, on the Vortex Lattice Method, and on a 3D Panel Method.

This tool enables the iterative design and analysis of multiple aircraft configurations.

3.5 Design

The chosen design is the one of a flying wing, a fuselage-less aircraft made of a wing, propulsion system, and control surfaces. The reasons are a simpler and sturdier mechanical structure, besides the possibility of the VTOL configuration.

3.5.1 Preliminary Design

As a starting point, a wing with a central hub and 2 semi-wings ending in symmetrical winglets was designed. The ZAGI12 airfoil, seen in figure 6, was chosen due to its good soaring capabilities and low stall speed.

With the airfoil chosen, its characteristics were calculated with the aid of XFOIL, an airfoil analysis tool built into XFLR5.
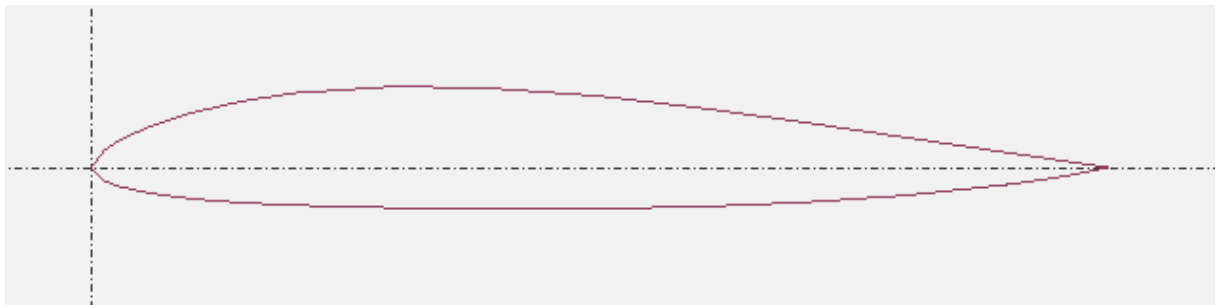
Figure 6 – Zagi 12 airfoil. Source: original

These characteristics plots can be seen in figure 7.

From this figure, the point with the highest Cl/Cd ratio - the theoretical point with the best lift to drag ratio, and therefore best gliding performance - can be found. It can also be seen that the airfoil's moment "pulls" it into this better Cl/Cd ratio, allowing the aircraft to fly into this ideal condition without deflection of the control surfaces, which would cause aerodynamical losses.

With that data, the main body was conceived, as seen in figure 8. With this CAD tool we can then analyze the performance of the aircraft as a whole. This gives us the same data as the airfoils', but for the whole craft, as seen in figure 9.

More data can be inferred from these graphs. From figure 7 it can be seen that the highest C_l, or Lift Coefficient, is obtained around $\alpha = 8$ deg, which, possibly by design of the airfoil, is also the zone with a higher C_l/C_d, or *lift-to-drag ratio* maximizing the gliding distance.
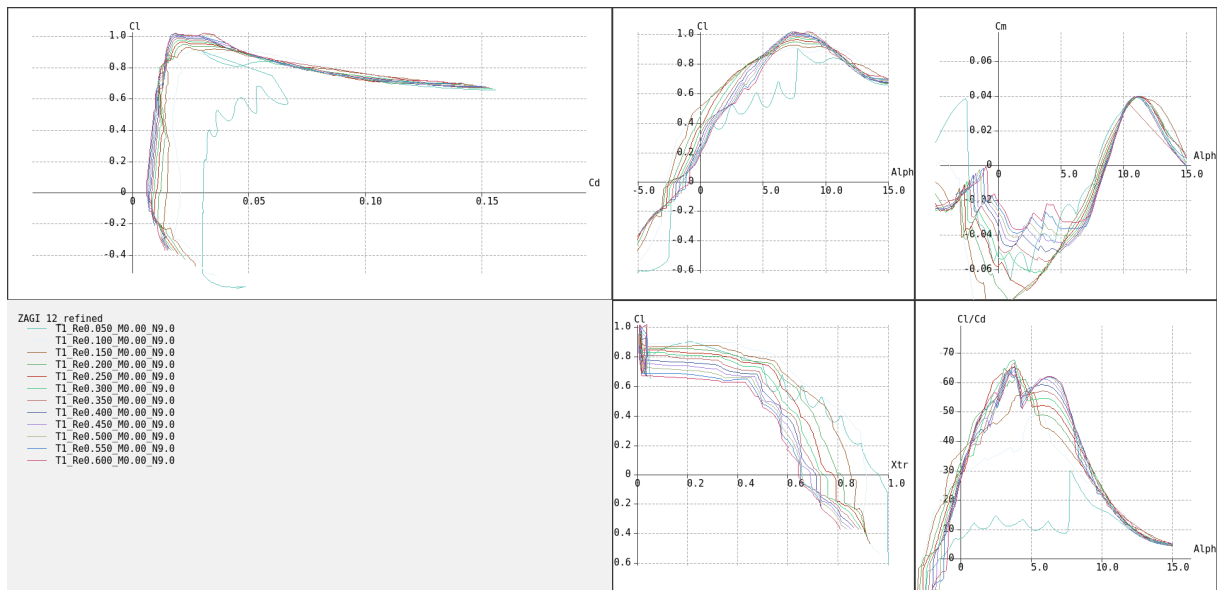
Figure 7 – Zagi 12 characteristics. The different lines are different reynolds numbers. Source: original
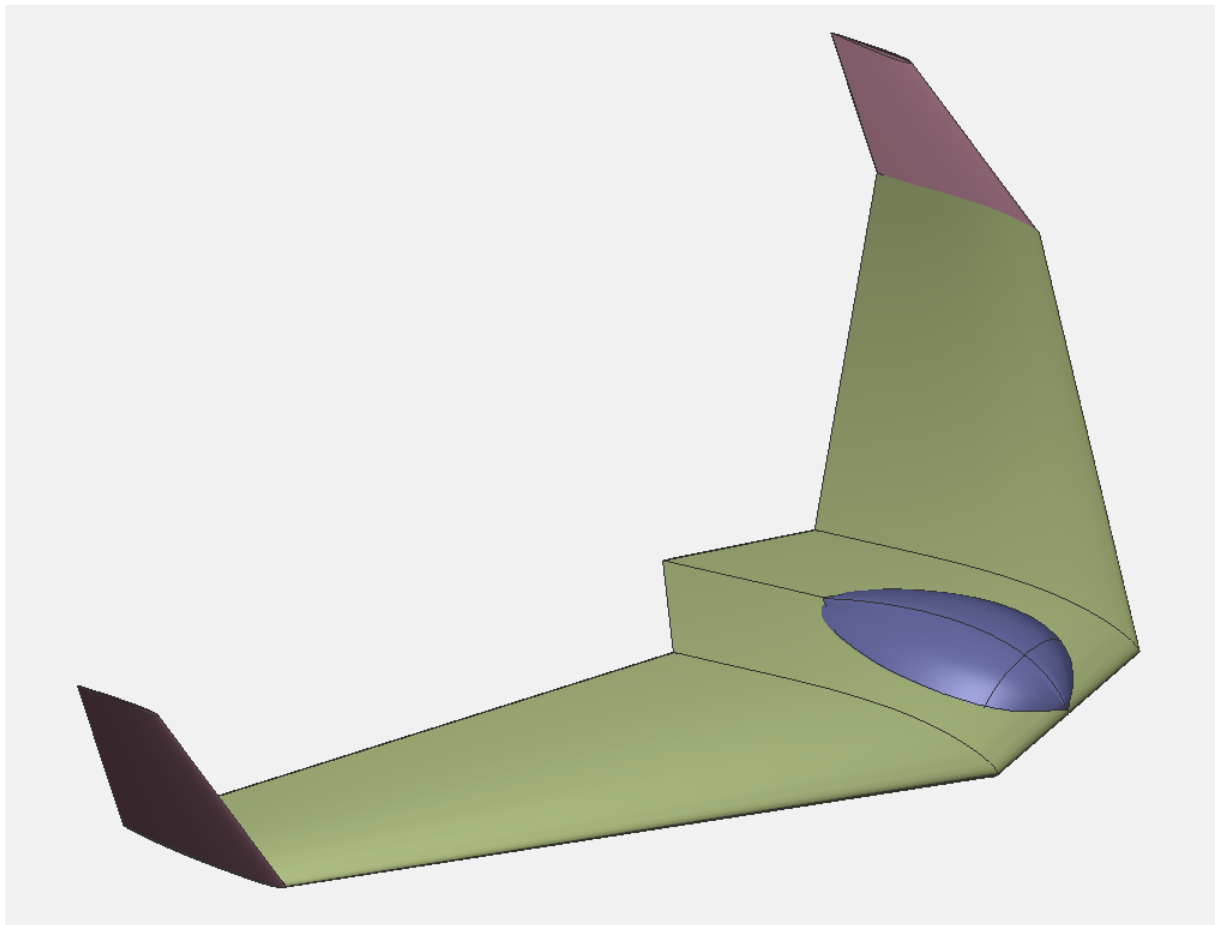
Figure 8 – First concept of the aircraft. Source: original

From figure 9, it is noticeable that without command inputs, the aircraft tends to point the nose down 2°, as its maximum lift-to-drag coefficient is at 8 deg, the center of gravity can be adjusted to make the cm/alpha angular coeficient close to 0, meaning the aircraft is marginally stable, in a way that only a small elevon input is required to turn it towards the best alpha, reaching it's maximum lift-to-drag. This change can, however, decrease the stability of the aircraft and increase the chances of a stall[1]. However, as the system is running on a closed loop, this is not a problem.

The flight time can be estimated by analyzing the motor power consumption during cruise flight.

First, as we assume $3kg$ of mass. We need to find the angle α that, at $15m/s$, produces $3kgf$ of lift.

The Lift formula is the following:

$$L = \tfrac{1}{2}\rho v^2 S C_L \tag{3.1}$$

Where L is the produced lift in Newtons, ρ is the air density in kg/m^3, S is the wing's surface area in m^2, and C_L is the dimensionless lift coefficient, and ρ is the airspeed.

Plugging the data into the equation and solving for C_L, we find that the incidence angle $\alpha = 0.83$, which, looking at the $C_D \times \alpha$ plot, results in a drag coefficient C_D of 0.08.

With the drag coefficient C_D, we can calculate the estimated drag force F_D:

$$F_D = \tfrac{1}{2}\rho v^2 S C_D \tag{3.2}$$

This results in a drag of approximately $8.84N$ ($900gf$).

Assuming there are manufacturing issues and other imperfections, we assume a drag of $1.5kgf$. Such drag would mean that each motor needs to produce around $750gf$ of thrust. According to figure 14, the motor needs approximately $6A$ to produce this thrust on either propellers set, resulting in $\frac{10Ah}{12A}$ hours of flight, or 50 minutes. This is assuming 70% extra drag than simulated.

A slower flying speed should also help with power consumption, as the required alpha will be higher, resulting in a better lift-to-drag coefficient.

3.5.2 Final Design

Due to building issues and the desire to maximize both effective payload and flight autonomy, the design was simplified, extending the chord back on the center of the wings, as seen in figures 10 - 12. The electronics bay was embedded into the main section,

---

[1] A stall is a condition in aerodynamics and aviation wherein the angle of attack increases beyond a certain point such that lift begins to decrease. This can lead an aircraft to crash.
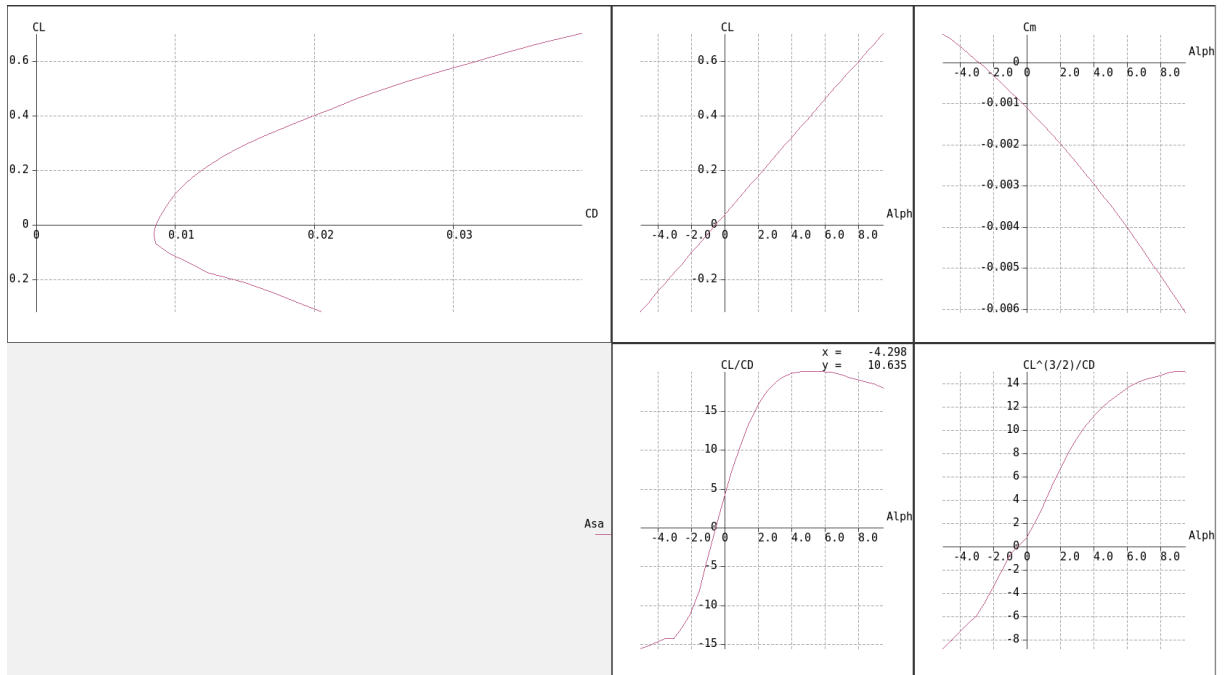
Figure 9 – Flight characteristics of the final aircraft design. Source: original
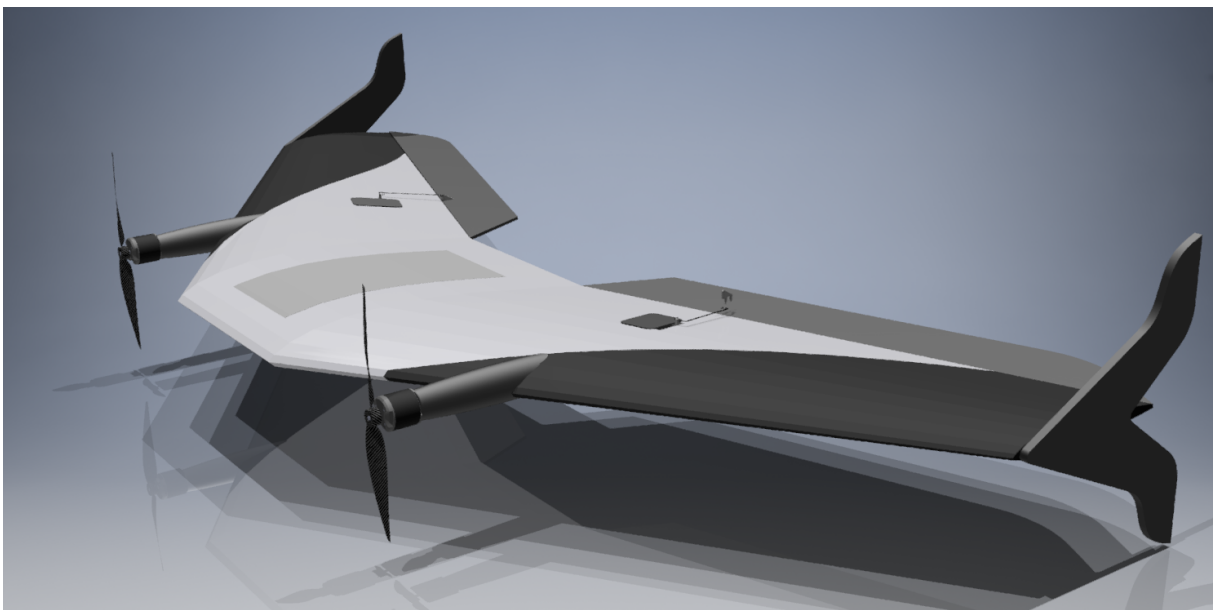
Figure 10 – Final design of the aircraft. Source: original

reducing the aerodynamical cross-section, thus reducing drag. The whole design was then assembled on Autodesk Inventor Professional prior the manufacturing of the prototype.
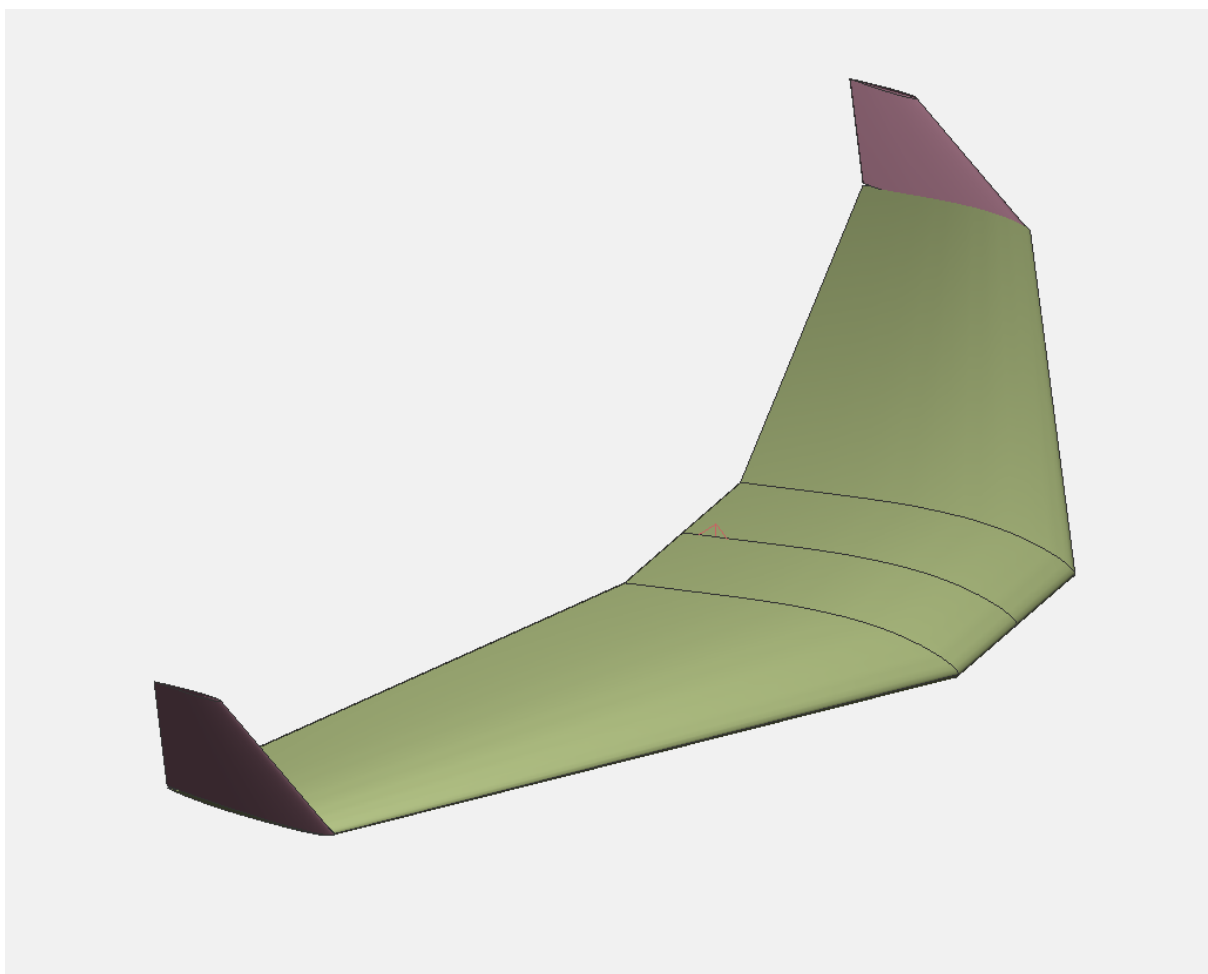
Figure 11 – Final design of the aircraft, on XFLR5. Source: original
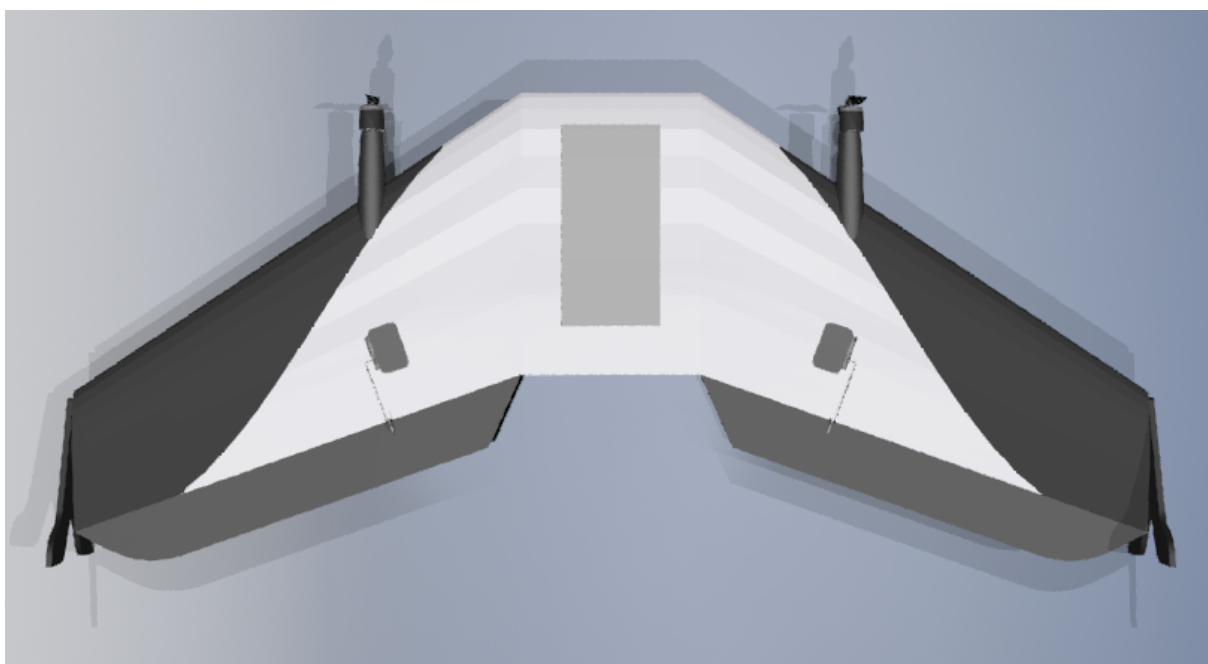
Figure 12 – Final design of the aircraft, top view. Source: original

4 The Electronics

In order for the aircraft to fly and navigate autonomously, onboard electronics are required, for both actuation, power source, and navigation. Some of the used electronics were already available, and were chosen for this reason. Others were chosen taking into account technical specifications and price.

4.1 Propulsion

Due to the familiarity and availability, the Mikrokopter Mk3638 Motor was chosen, paired with E-Max Simon 60A escs. The main characteristics of the motors can be seen on table 1.

Experimental curves for the motor are available at Mikrokopter's website, and the relevant ones are reproduced in figure 14. Each motor should give, on 16 V, around 1.9 kg of static thrust when paired to 12 inches propellers, up to 2.5 kg on 15 inches, while drawing 25 A, or about 560 W eletrical power, or 350 W mechanical power.

Figure 13 – Mikrokopter MK3638 Brushless Motor. Source: Mikrokopter.de
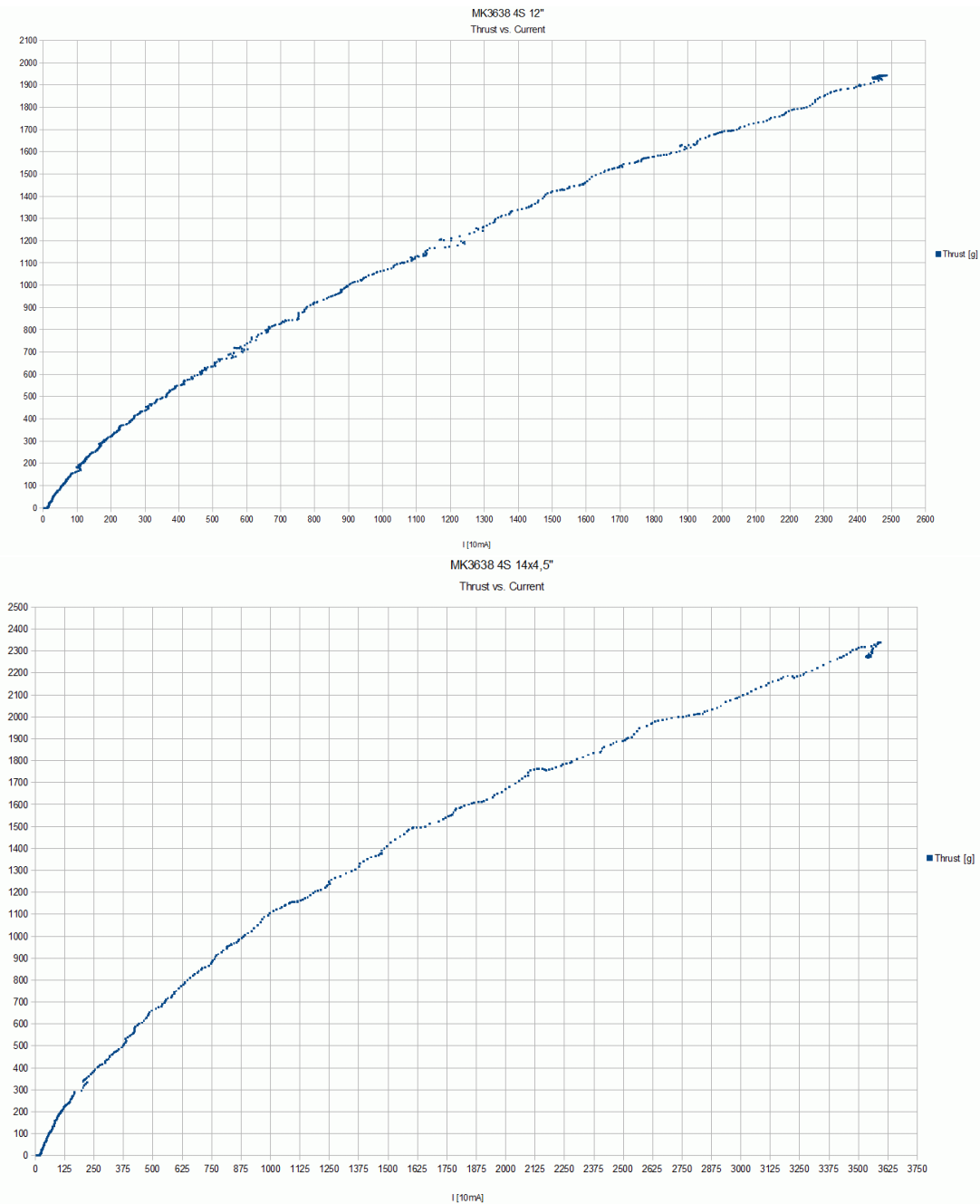
Figure 14 – Motor curves with 12 and 15 inches propellers. Source: mikrokopter.de

Lithium cell count	3-6
Load current	max. 20A
Maximum load current. (60 sec)	25 A
No-load speed	770 rpm / V
Recommended Propeller Size	10" .. 14"
Mechanical power	350 W
Thrust max.	2200 g
Wire length	about 52 cm
Total weight (without cable)	approx 100 gf
Dimensions	h=35mm, d=38mm
Shaft diameter	4 mm

Table 1 – Mk3638 Brushless Motor Technical Specifications Source: original

As the airplane is aimed to weigh around 3 kgf, each motor needs to pull at least 1500gf for hovering, leaving a maneuvering margin of around 1 kgf for each motor.

4.2   Batteries

As each motor can draw up to 25 A, the battery should be able to provide up to 50 A without issues. The Batteries chosen are also the ones already in use by the company, Multistars 10000 mAh 10C, which, at 10 C rating, are able to sustain a constant draw of up to 100 A.

Each battery weighs approximately 800 g and measures 160 x 65 x 36 mm.

Minimum Capacity	10000mAh
Configuration	4S1P / 14.8V / 4Cell
Constant Discharge	10C
Peak Discharge (10sec)	20C
Pack Weight	804gf
Pack Size	160 x 65 x 36mm
Charge Plug	JST-XH
Discharge Plug	XT90

Table 2 – Multistar 4s 10000mAh Battery Technical Specifications. Source: Hobbyking.com

Figure 15 – Multistar 4s 10000 mAh Lithium Polymer battery. Source: Hobbyking.com

4.3 The Servos and Control Surfaces

The control surfaces must be slightly larger than usual for a flying wing, as on a tail-sitter a reasonable amount of air must be deflected on hover situation, while on most wings a steady airflow is assumed. It is suggested to have control surfaces taking up to 30% of the chord of the wings. Since they are easily swappable, it was decided to start with smaller ones, with a 10 cm chord, and replace them if necessary.

The servos chosen were standard servos Savox SV-0220, linked to the elevon horns with a stiff wire.

Figure 16 – Savox SV-0220 servo. Source: Hobbyking.com

Table 3 – SV-0220 Technical Specifications. Source: Hobbyking.com

Torque @ 6v	6.5kgf.cm
Torque @ 7.4v	8.0kgf.cm
Speed @ 6v	0.16 sec/60 deg
Speed @ 7.4v	0.13 sec/60 deg
Dimensions L x W x H (mm)	40.7 x 20.0 x 37.0
Weight	59.0gf

4.4 The Flight Controller

The multirotor configuration had a huge boom in the last 10 years. In 2009 the first hobby-grade flight controller for multicopters was born, Rolf "KaptainKuk" Bakke's

"KK board". Using a simple AVR controller and three gyroscopes, the board could control angular speed on three axis, enabling pilots to control the multirotors. It was programmed in AVR assembly and had individual PID controllers for each axis. Shortly after, Alexinparis noticed the gyros on the Wii Motion + controller, and MultiWii was born. This project grew to support a variety of sensors and boards, and had an active development community, but has now saturated the AVR controller's capability. Shortly after, still in 2010, DIY Drones released the open-source Arducopter, featuring more advanced flight modes, and even autonomous flight. It did still involve compiling code and flashing it to the controller though. In 2011, DJI started to get visibility with the NAZA controller, which showed remarkable stability, and later got upgraded with a GPS allowing the drone to return to home and hold position in the air. The controller was often sold with a standard frame and motors, which improved stability as the board was pre-tuned to the sold equipment.

Shortly after, DJI began to manufacture the DJI Phantom drones, which is now the main player in the market. Nowadays, three major controllers coexist: MultiWii was ported to 32bits architecture processors and lives on as Baseflight and Cleanflight, mostly on quadcopter racing boards; DJI leads the aerial photography market with their phantom quadcopters; and on the autonomous fields, Ardupilot, PX4, Mikrokopter, and DJI are still competing for the better solutions.

The Flight Controller board chosen is a PixHawk. Both PX4 and ArduPilot software stacks support this board. But Ardupilot is a more mature, tested, open, and community-based platform, and thus it was chosen here, running latest release of ArduPlane, where there's experimental support for tail-sitters.

Figure 17 – Pixhawk flight controller and most peripherals. Source: Mrobotics

4.5 The GPS

The used GPS is an U-Blox M8N GPS receiver, shown in figure 18, coupled to an external compass sensor. The external compass is important because the high currents flowing close the Pixhawk affect the readings of the internal compasses. It supports concurrent reception of up to 3 GNSS (Global Navigation Satellite Systems), GPS, Galileo, GLONASS and BeiDou.

It's precision is around 3 m, occasionally getting lower than 1 m [12].

Figure 18 – M8N GPS receiver and external compass. Source: cooltoyz.co.uk

4.6 The Telemetry

The telemetry system provides a serial (UART) connection to the aircraft via a radio system. The one used can be seen on the top right corner of figure 17 and is a 900MHz radio modem. The telemetry allows real-time reading of parameters and attitude, as well as writing them for setup and tuning.

4.7 The Radio Control System

The used Radio System is a 2.4Ghz radio by Turnigy, the Turnigy 9x. This radio uses fast frequency hopping to avoid interference, and has a reported range of up to 3

km [13]. The radio was modded [14] and the firmware was replaced by the open-source OpenTX [15], which provides much more flexibility to the system, such as custom mixes, switches, automatic functions, periodic functions, and telemetry capabilities.

Figure 19 – The Turnigy 9X Radio System. Source: radioc.co.uk

4.8 The Camera

In order to take pictures for the aerophotogrammetry, a camera is needed. As the company is also working on a RTK[1] system, a compatible camera is required.

Sony's DSC HX60V was chosen due to its sensor size, which at 55 mm is able to provide a broad field of view; for its weight, relatively low at 272 gf; presence of Multi protocol for communication with flight controllers; and external flash support, which can be used to read the precise time when a picture was taken. The time can then be synced to the current GPS position and the aircraft's attitude.

---

[1] Real Time Kinematics, a system that improves GPS localization giving millimetric precision.

Figure 20 – Sony DSC-HX60V Camera. Source: sony.pt

5 The Software

Two main types of software are required for the operation of this kind of aircraft. The flight controller runs on the on-board computer and is responsible for controlling the flight itself, and a ground station software, responsible for higher level commands and telemetry. This chapters details the software used in the project.

Figure 21 – PX4 and Ardupilot. Sources: Ardupilot.org and px4.io

5.1 Flight Controller

The flight controller software runs on the on-board computer, and is responsible for controlling the attitude, altitude, and position of the aircraft. In order to achieve this, most flight controller boards come with internal sensors (accelerometer, gyroscope, magnetometer, barometer) and external ones (airspeed, magnetometer, GPS). The acquired data is used to estimate attitude and position, which is then controlled by the control loops.

The possible choices of flight controller software were ArduPilot and PX4. Ardupilot is a community-drive software started by DYIDrones in 2009 [16]. It is GPL licensed, which means all changes made and commercialized must be open-sourced [17]. ArduPilot is a mature software, with a large community of users and testers.

PX4 was also developed since 2008 [18], mostly by the Computer Vision and Geometry Lab of ETH Zurich (Swiss Federal Institute of Technology) [19] and the Autonomous Systems Lab [20] under a more permissive BSD 3-clause license [21]. For a while both projects worked closely, the Pixhawk Flight Controller board is a result of this interaction. Both also joined Dronecode [22], a Linux Foundation [23] initiative started in 2004 as an attempt to grow the UAV ecosystem and reach larger companies.

Dronecode, however, evolved into a flawed model, according to the Ardupilot Dev Team. It required for all projects to hand over all domains, accounts, and trademarks to their control. The projects were also being directed by the so-called Platinum Members, which meant the development would not be in control of the community anymore. By September 9, 2016, a letter was released stating that Ardupilot was leaving Dronecode, and explaining why [24].

The Ardupilot code was chosen for this project due to their larger openness and community.

Ardupilot is split into four main sub-projects:

- ArduCopter is a Flight controller for helicopters and Multirotors.

- ArduPlane is Focused on fixed wing aircraft, but now includes some of the ArduCopter control loops for VTOL capabilities.

- ArduHover is focused on land vehicles and aquatic surface vehicles.

- ArduSub is a controller for submersible vehicles.

While each of them has different features, they share most of the core ArduPilot code.

Arduplane provides all required controllers for features such as automatic vertical take off and landing, fixed-wing flight, waypoint navigation, inverted flight, stall prevention, geo-fencing, and terrain following: more than enough for performing the tasks proposed in this project.

5.2   Software Architecture

The ArduPilot codebase is split into four main projects, they all do however have the common base, and thus, a similar architecture. In figure 22 the high level architecture is shown, presenting an overview of the main interactions between hardware and software components, from the ground station, to the communication protocol, to the low-level sensors. Arduplane was used, as it implements some of Arducopter's capabilities for VTOL capabilities on fixed wing aircraft.

A more detailed diagram, in figure 23, details the interactions on a source-code file level. Additional information is available at the project's website, as well as the source code [25].

5.3   Ground Station Software

For ground station software, there are two major players, ArduPilot's own MissionPlanner, and KDE's QGroundControl.

MissionPlanner is developed in C# and is generally more up-to-date with Arduplane. It does however have performance issues and is not compatible with Linux.
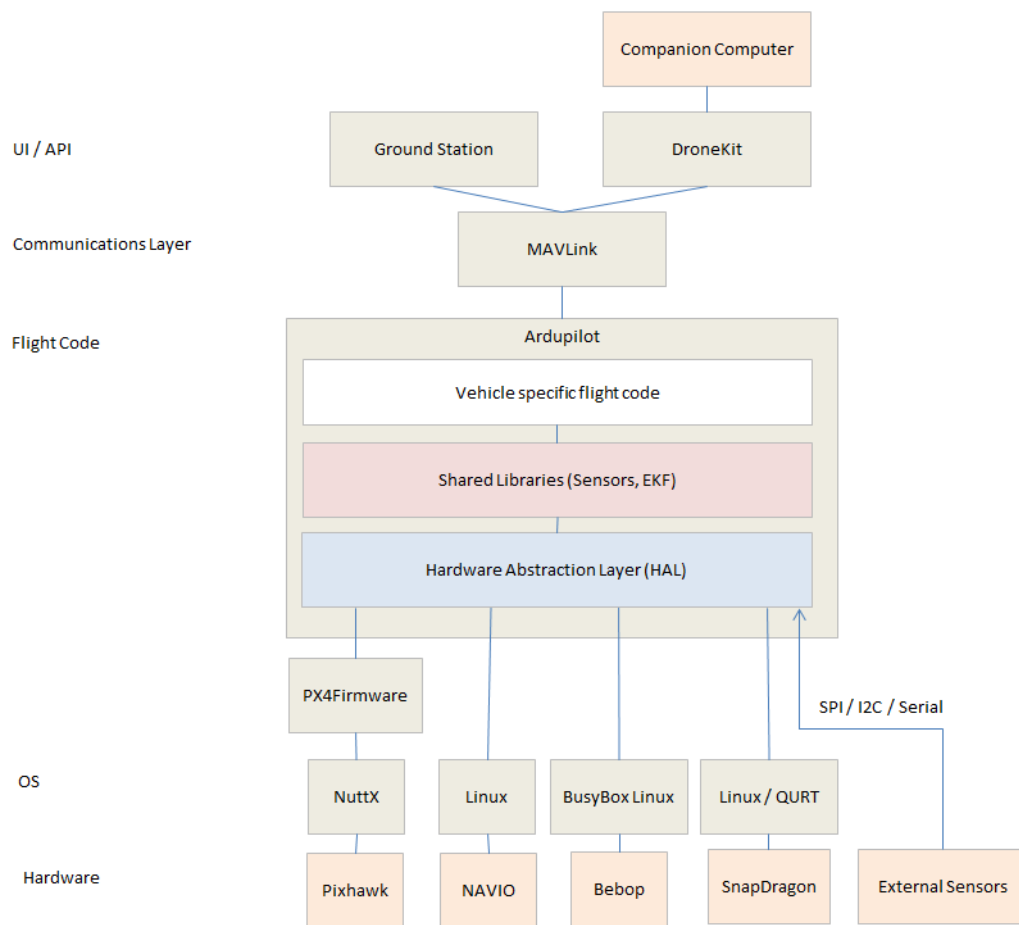
Figure 22 – ArduPilot high-level software architecture. Source: ArduPilot.org

QGroundControl is a C++ and Qt based software, boasting a high performance and well-finished interface, as well as compatibility with windows, Linux, OS-X and Android. But as it attempts to support both ArduPilot and PX4, sporadically the Ardupilot support is not up to date.

Both are able to do the basic aircraft setup, change modes, and setup parameters via wired or wireless connections. As both have upsides and downsides, both were used in the project. QGroundControl was used whenever possible, and, when it could not perform a task, MissionPlanner was used. Their user interfaces can be seen on figure 24.

Additionally, Mavproxy [26] and APM Planner 2 [27] were used to view logs and export them to Google Earth
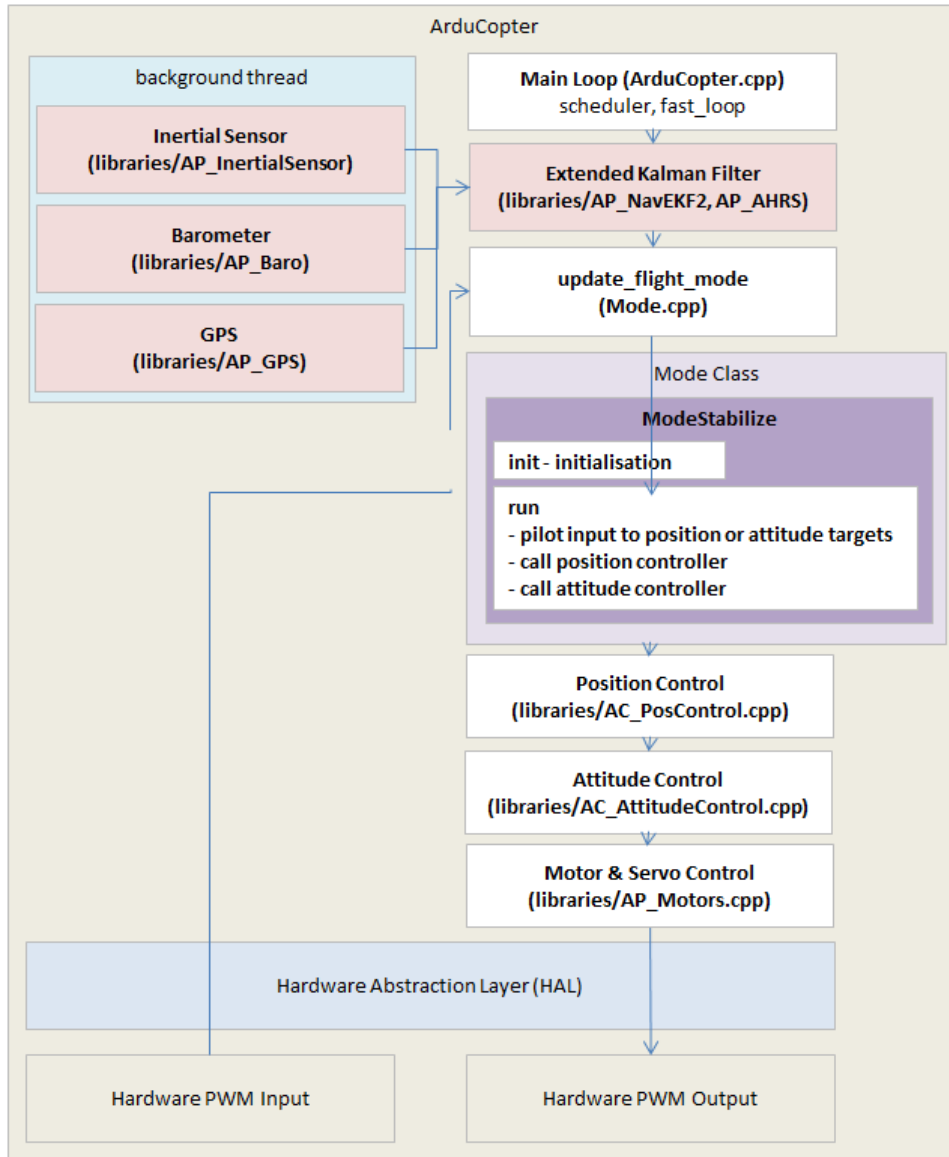
Figure 23 – ArduPilot high-level software architecture. Source: ArduPilot.org
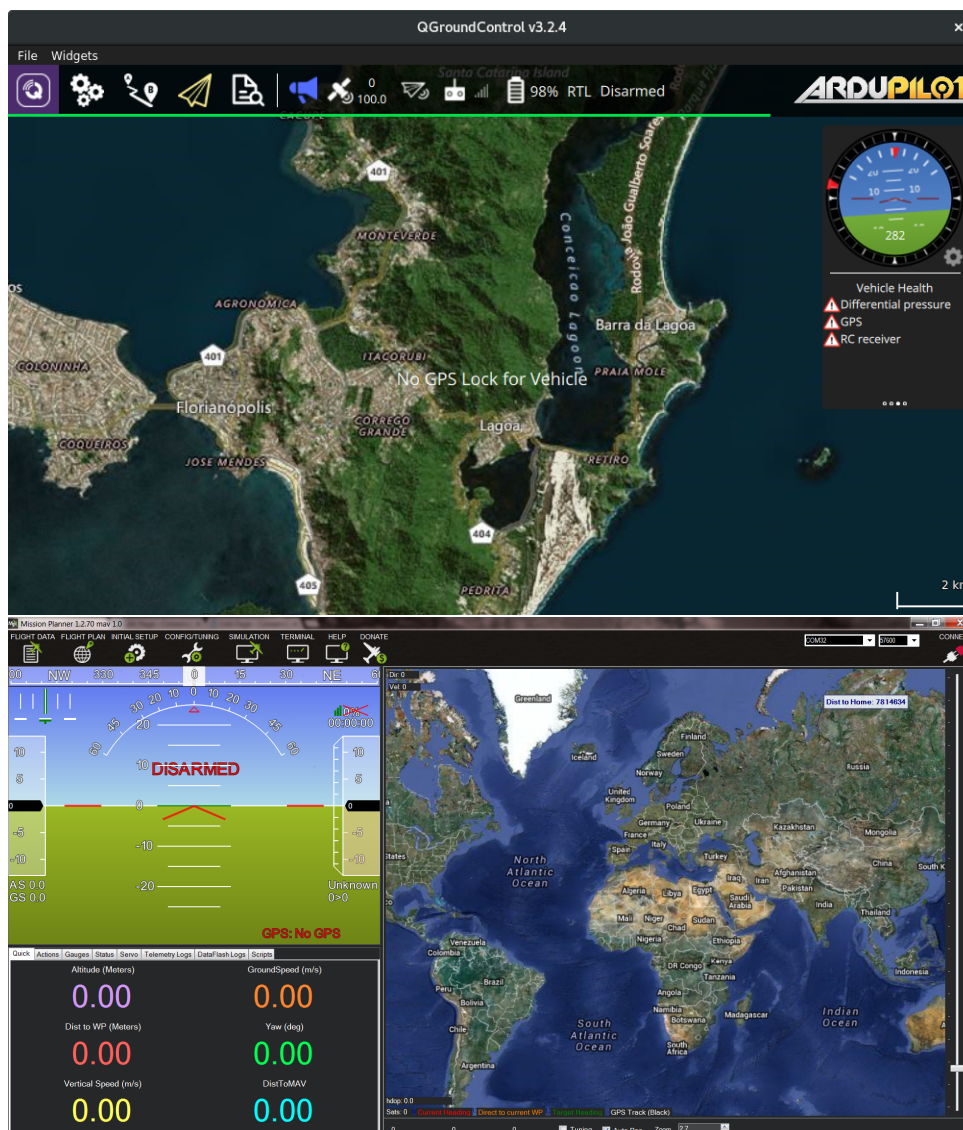
Figure 24 – QGroundControl and MissionPlanner. Source: original

6 The Flight Control Structure

The control Structure used is the one of ArduPlane. In hover, or tail-sitter mode, the ArduCopter stabilization system is used, while in airplane/fixed-wing mode, Arduplanes controllers are used. Both will be discussed and explained in the following sections.

6.1 The Data Acquisition

In order to control all the required variables, they must be available for the controller. The Pixhawk controller provides two redundant IMUs for this reason. Each of them runs an Extended Kalman Filter tracking 22 states [28] [29]. Of the two running Kalman Filters, the one with smaller error estimation is used, and the estimated states are made available for the controllers.

6.2 On Airplane Mode

On Airplane mode, the aircraft is always moving forward, towards the X axis, position control depends on defining a route and pointing the aircraft in order to remain on it.

6.2.1 Roll and Pitch Control

The roll and pitch control loops (seen in figures 25 and 26) are responsible for keeping the aircraft on the desired orientations on the X and Y axis. Usually, pitch is controlled by turning the elevator up and down, while roll is controlled by the deflecting the ailerons. On this aircraft, however, there are only two control surfaces, such that the output of both controllers are summed (mixed, as is usually said in the RC world) in order to control both axis at the same time. While at first they look like a classical P+I+D controller, there are some small changes:

- There's a feedforward controller trying to cancel the current angular rates.

- The Derivative and Integral terms are scaled to the airspeed, and the controller's output as well. This is because as the aircraft moves faster, less deflection is necessary to displace the same amount of air, resulting in the same movement of the body.

The controller outputs are called AileronDemand and ElevatorDemand, and are the theoretical required input on each axis. Ardupilot works like this in order to abstract

the possible airframes. The next code stage mixes these needs to get the proper output for the current airframe. On an airplane, the outputs are only limited to the maximum and minimum possible outputs, on an aircraft like the one in this project, the outputs are combined linearly to get the correct outputs on each elevon.

OMEGA = 1 / RLL2SRV_TCONST

K_P = (RLL2SRV_P - RLL2SRV_I * RLL2SRV_TCONST) * RLL2SRV_TCONST - RLL2SRV_D

K_I = RLL2SRV_I * RLL2SRV_TCONST

Limiter constrains signal between -RLL2SRV_RMAX and +RLL2SRV_RMAX

Figure 25 – Roll control loop. Source: ArduPilot

The output function for RollDemand is as follow:

$$D_{ail}(s) = \left(\frac{A_s(s)K_p s + A_s(s)^2(K_d s + K_i)}{s} \right)(\overline{R}(s) - R(s)) - A_s^2(K_D s + K_i)Rs \quad (6.1)$$

$$R(s) = \Omega sat(R_{raw}(s)) \quad (6.2)$$

Where $R_{raw}(s)$ is the raw roll reading, $D_{ail}(s)$ is the aileron demand, $A_s(s)$ is an airspeed scaling value, $0 < A_s < 1$ (responsible for reducing the control outputs at higher speed, as most aerodynamical forces are proportional to the square of the airspeed), K_d is a derivative term, K_i is an integral term, $\overline{R}(s)$ is the Roll angle setpoint, and $R(s)$ is the current roll angle.

OMEGA = 1 / PTCH2SRV_TCONST

K_P = (PTCH2SRV_P - PTCH2SRV_I * PTCH2SRV_TCONST) * PTCH2SRV_TCONST - PTCH2SRV_D

K_I = PTCH2SRV_I * PTCH2SRV_TCONST

Limiter constrains signal between PTCH2SRV_RMAX_DN and PTCH2SRV_RMAX_UP

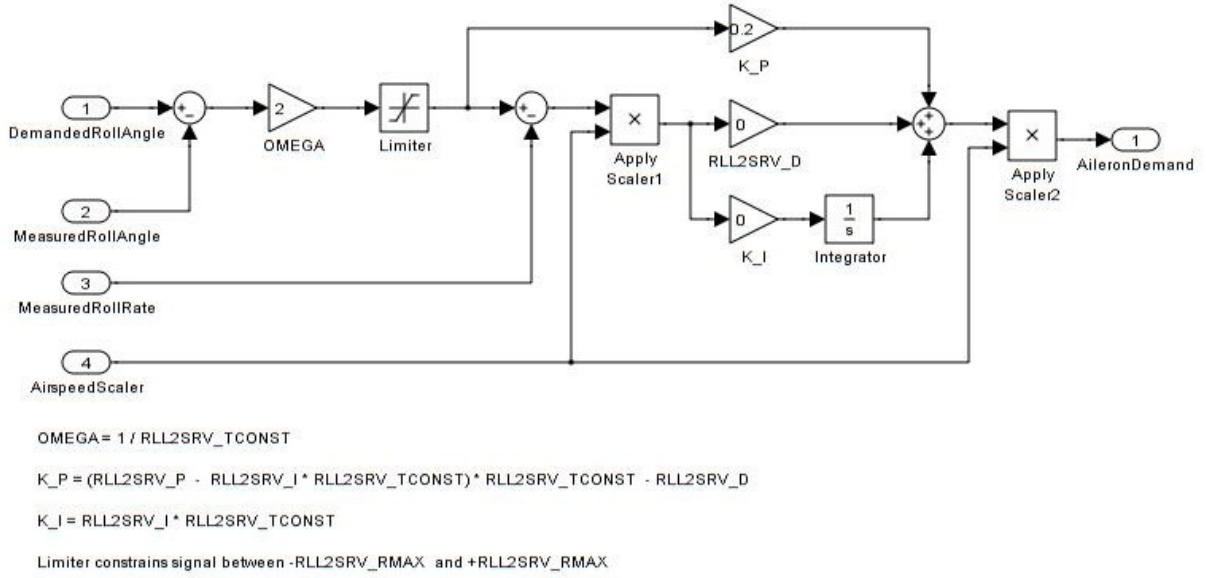Roll Compensation = PTCH2SRV_RLL * gravity / (airspeed * tan(BankAngle) * sin(BankAngle))
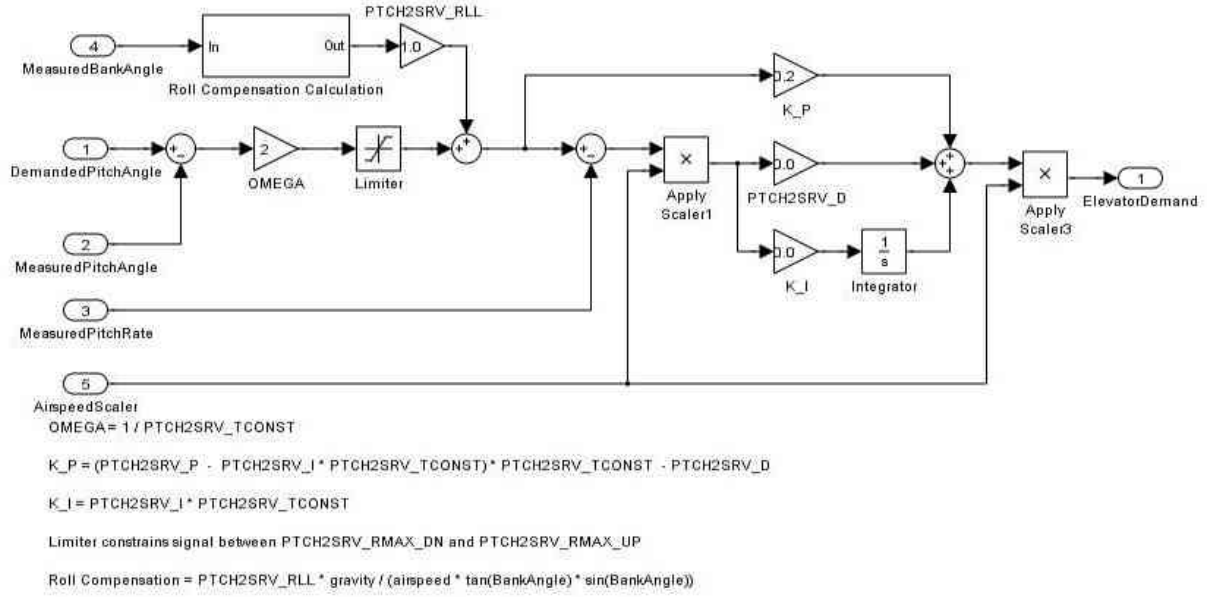
Figure 26 – Pitch control loop. Source: ArduPilot

The output function for PitchDemand is similar:

$$D_{pit}(s) = \left(\frac{A_s(s)K_p s + A_s(s)^2(K_d s + K_i)}{s} \right)(\overline{P}(s) - P(s)) - A_s^2(K_D s + K_i)Ps \quad (6.3)$$

$$P(s) = \Omega sat(P_{raw}(s)) + P_{2R}R(s) \quad (6.4)$$

Where $P_{raw}(s)$ is the raw pitch reading, $D_{pit}(s)$ is the pitch demand, $A_s(s)$ is the same airspeed scaling value, K_d is the derivative term, K_i is the integral term, $\overline{P}(s)$ is the Pitch angle setpoint, and $P(s)$ is the current pitch angle. Additionally, $P_{2R}R(s)$ is a term that attempts to cancel out the effect of banking(rolling) the aircraft affecting the pitch.

6.2.2 Yaw Control

The Yaw Control loop controls the angle around the Z axis. This is usually used for landing only, and is not necessary on this aircraft on airplane mode, as it is most usually changed by a combination of roll and pitch inputs (rolling slightly left with generate a yaw turn leftwise on most aircrafts). It can, however, be seen in figure 27. Like the D and I terms on the roll axis, the controller's output is again scaled with the square of the *AirspeedScaler* factor.
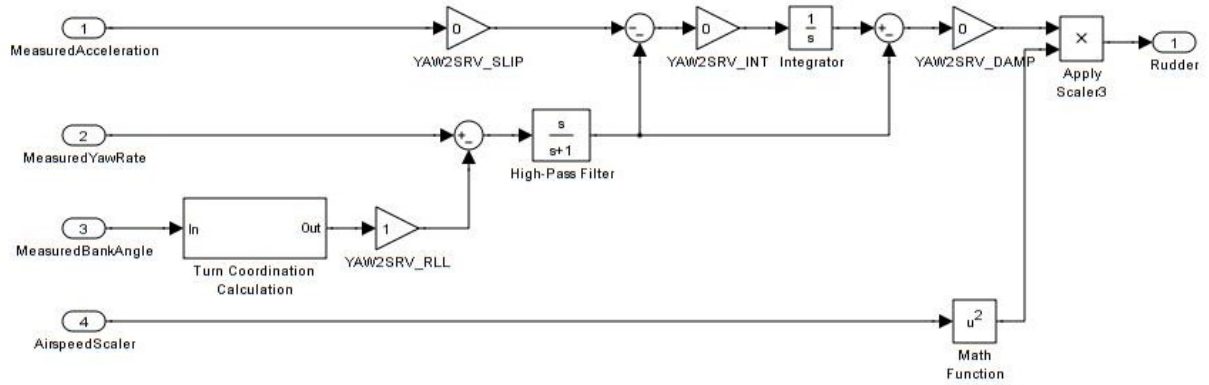
Figure 27 – Yaw control loop. Source: ArduPilot

The output function for RudderDemand is the following:

$$D_{rud}(s) = \underbrace{-\frac{A_s^2 D_y K_i}{s} a_y(s)}_{\text{Anti-Slip}} + \underbrace{\frac{s}{s+1}}_{\text{High-pass filter}} \left(\frac{D_y k_i + 1}{s}\right)(Y(s)s - BY(s)) \tag{6.5}$$

Where D_{rud} is the rudder demand, $A_s(s)$ is the airspeed scaler, D_y is a dampener of translations on the Y axis, K_i is an integral term, $a_y(s)$ is the measured acceleration in Y, and B is a gain scaling the output to the servos. While there's no rudder on this aircraft, it's function can be emulated by applyinf differetial thrust on the motors.

6.2.3   Navigation: Waypoint Circling

The circling algorithm is a simple PD[1] loop forcing the aircraft into the waypoint. The Always-forward nature of airplanes results in the aircraft circling the waypoint. This is usually used when the aircraft is idly waiting for something.

---

[1]   Proportional-Derivative

RTL, GUIDED or LOITER

Waypoint Circle Mode: Use a simple PD controller with feed-forward centripetal acceleration to circle the waypoint.

$$a_y = K_x \cdot x + K_v \cdot \dot{x} + \frac{V^2}{R}$$

$$K_x = \frac{1}{T^2}$$

Waypoint Capture Mode : Aim at the waypoint

Change to Waypoint Circle Mode when acceleration demands from the circle and capture mode calculations cross-over
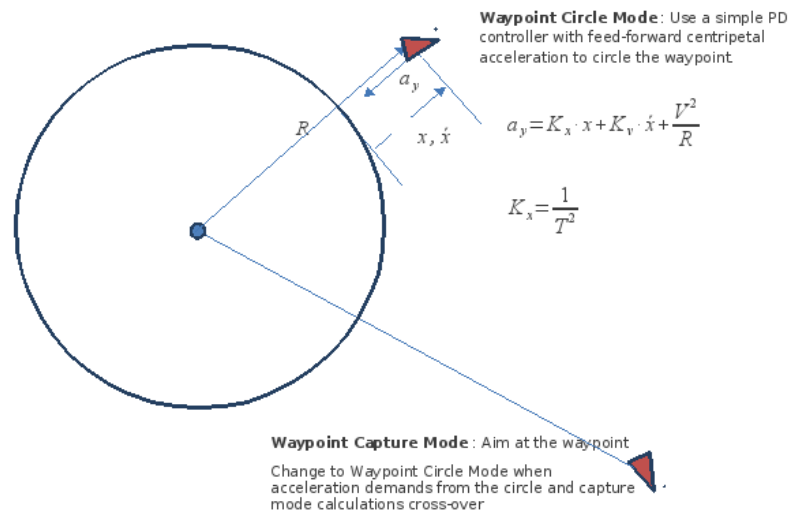
Figure 28 – PD navigation controller, for circling waypoints. Source: ArduPilot

6.2.4   Navigation: Waypoint Following

Since a fixed-wing aircraft usually can't fly in-place, waypoints can be used in two general ways, the aircraft can fly around it in circles, or hit it and then follow to the next one.

In order to circle it, a L1 controller is used. L1 is an adaptative controller designed for trajectory following on aircraft [30]. This technique provides better following of trajectories by taking advantage of the aircraft inertia and geometric properties of the desired path. The controller is briefly explained in figure 29.

Waypoint Following

All tracking modes (with the exception of circling the waypoint) use the following guidance law, where η is the angle between the aircrafts velocity vector and the bearing to the Ll_{point} or the waypoint being captured

$$\zeta = NAVL1_{DAMPING}$$

$$T = NAVL1_{PERIOD}$$

$$K = 4 \cdot \zeta^2$$

$$Ll_{length} = \frac{1}{\pi} \cdot \zeta \cdot T \cdot V$$

Tracking Mode : Aim at the Ll_{point} that is Ll_{length} away from the aircraft

Prev WP   Next WP

Waypoint Capture Mode : Aim at the waypoint. Change to Tracking Mode if distance to waypoint $< Ll_{length}$

Track Capture Mode : Aim at the Ll_{point} that moves along the track so that β = 45 degrees. Change to tracking mode
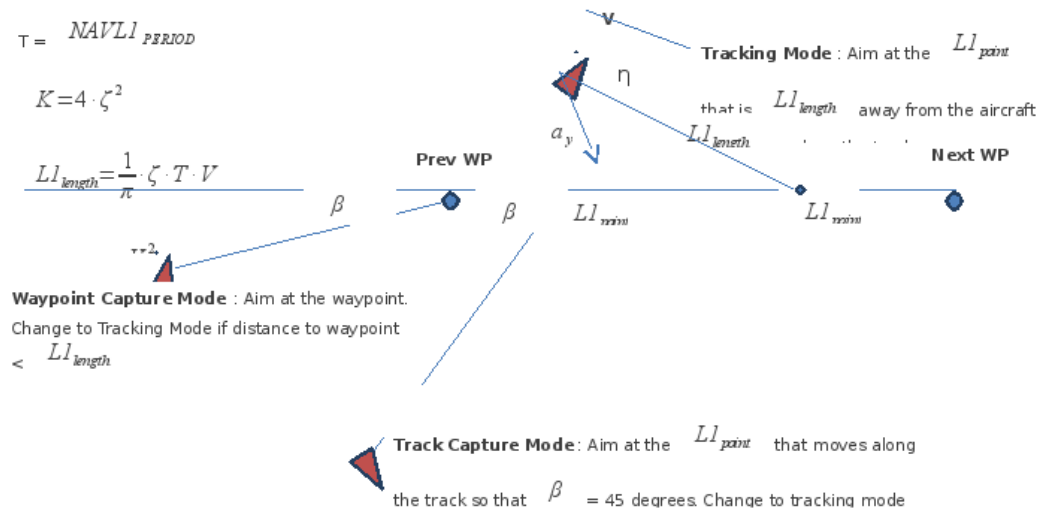
Figure 29 – L1 navigation controller. Source: ArduPilot

6.3   On VTOL Mode

On the VTOL or multirotor mode, cascaded P[2]/PID[3] loops are used. The inner loops control the angular speed, and the outer loops control the attitude, with the outermost loops controlling the altitude and position (again with the L1 controller). These controllers can be seen in figure 30.

---

[2]   Proportional
[3]   Proportional-Integral-Derivative

Figure 30 – Attitude controller. Source: ArduPilot

7 Prototyping

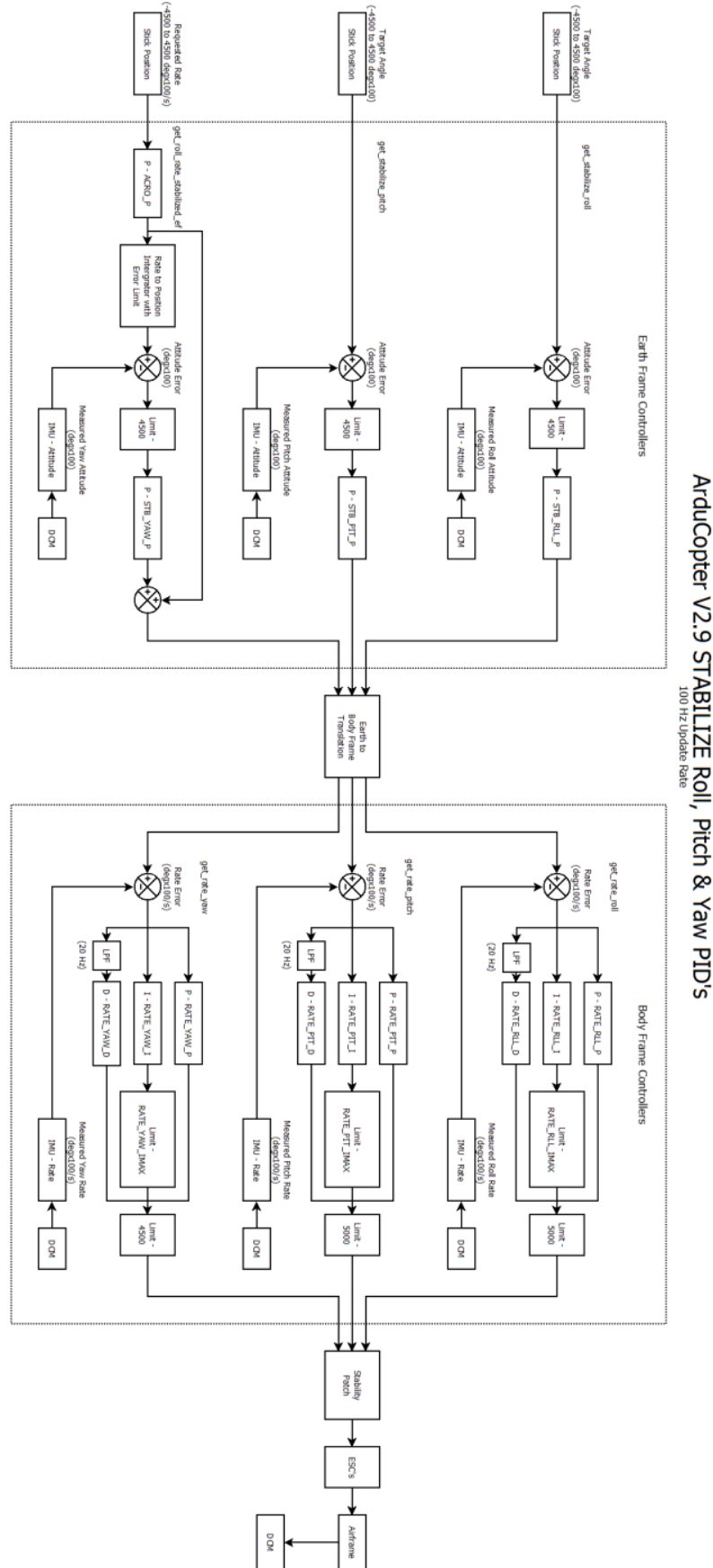As in any product development, a few protoypes were developed. First a smaller, 50 cm wingspan aircraft with no airfoil was assembled to test and tune the flight controllers. The reduced version also enabled testing in close spaces and proximity with people with reduced danger.

With the reduced prototype proven, the larger one, photography-ready was developed. This second prototype is closer to the final desired product, and can be used as such.

Both prototypes are described, as well as their assemblies, in the next sections.

7.1 Reduced Scale Prototype

A reduced prototype was used for preliminary tests of the flight controller and control systems.

Mechanically, this prototype consists of a foam board, two motors, and two control surfaces.

Smaller electronics are used as well. The servos are Turnigy 9 gram servos, the motors are AXN Floater-Jet 2208 2150KV brushless motors, the Escs are HobbyKing's RedBrick 30A ESCs, and the battery a Zippy Compact 3s 1000mah 35C.

The control surfaces were taped to the main body, and linked to the servos by a wire and plastic horn.

The motors had a custom mount 3D-Printed and fitted into the foam.

For the tests and tuning, the prototype had a hook on top, so it could be hanged from the ceiling to avoid hitting the floor and walls during the tests.

The first prototype and its components can be seen in figure 31, and its components are as follow:

1. Motors and 3D-printed mounts

2. HobbyKing RedBrick 30A ESCs

3. Turnigy Pro 9 gram servos

4. Diy OpenLRS 433 MHz receiver

5. Zippy Compact 3s 1000mAh 35C lithium-polymer battery

6. Pixhawk controller
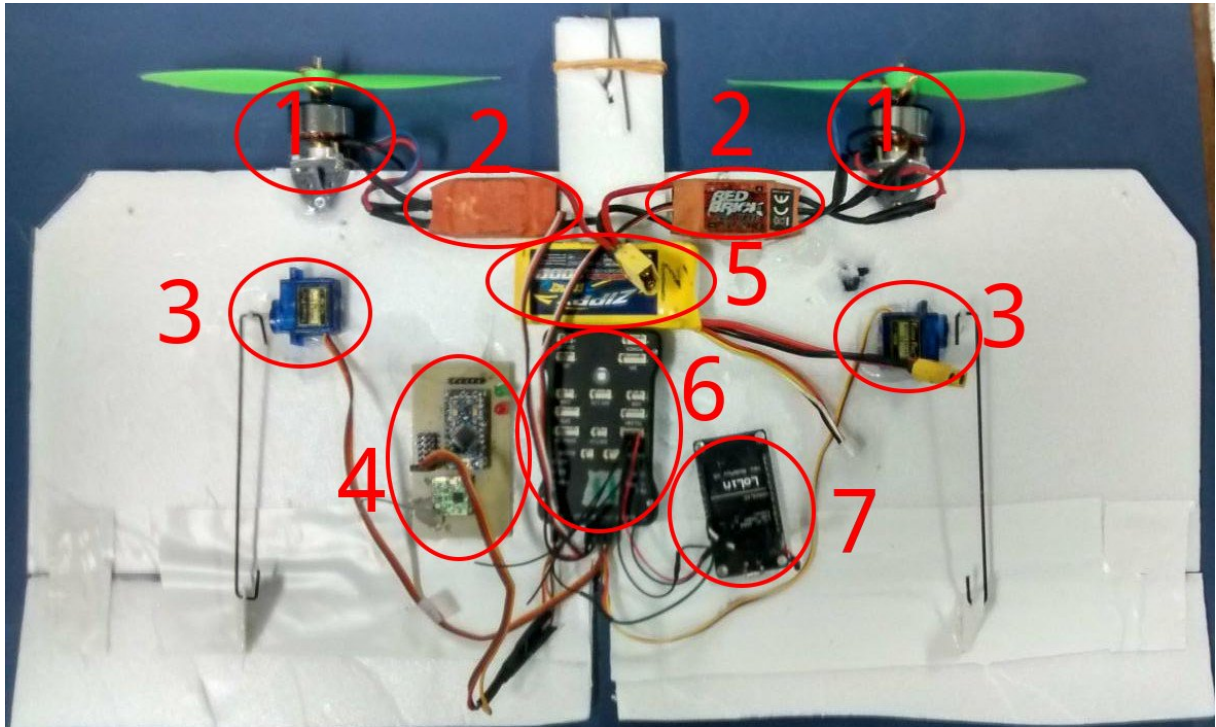
7. ESP-8266 board for telemetry

Figure 31 – Reduced Prototype and parts. Source: original

7.2 Large Prototype

For the larger prototype, standard RC building and fast prototyping technologies were used. The Zag12 airfoil at root was 3D printed in 3 parts (figure 32) then joined and insulated from the hot-wire heat with aluminum foil. For the trapezoidal wings, one side of the wire was tied to a fixed point, in such way that, if the airfoil was a circle, the wire would cut a cone on the foam. This enabled the cut of the trapezoidal wings out of foam. For the center section, two profiles were 3D-printed. their perimeters were then marked with numbers, in such way that two people, one on each side, could coordinate the hot-wire cutting process.

Figure 32 – 3D Printed Airfoil Source: original

This process isn't perfect for the trailing edge, so some of it needs to be removed, which later gets replaced by the elevons.

The cut foam then needs to be sanded down to remove imperfections. The half-wings are then joined with hot glue, and fiber glass spars are used to reinforce the structure.

From this point, the sections can be joined permanently or spars can be used to quickly assemble them.

With the three sections properly cut, they are glued together and sanded, and glass fiber rods were embedded and glued into the structure, two on the top and two on the bottom.

With the main structure assembled, the servos were embedded into the structure. A pocket was carved with hot wire, and two nut-holding 3D-printed parts were embedded deep into the foam and used to screw the top cover, as seen in figure 33.
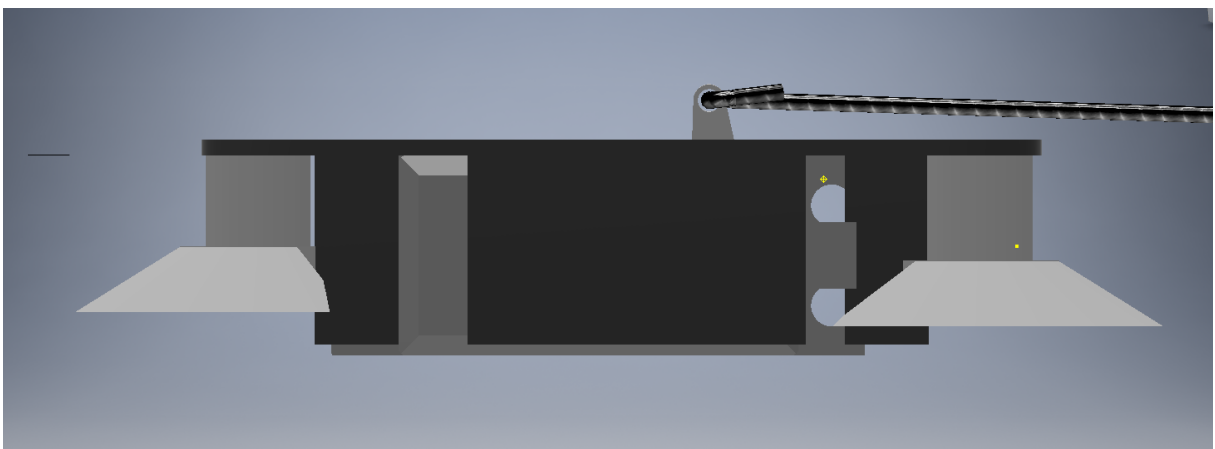

Figure 33 – 3D-printed servo mount structure. Source: original
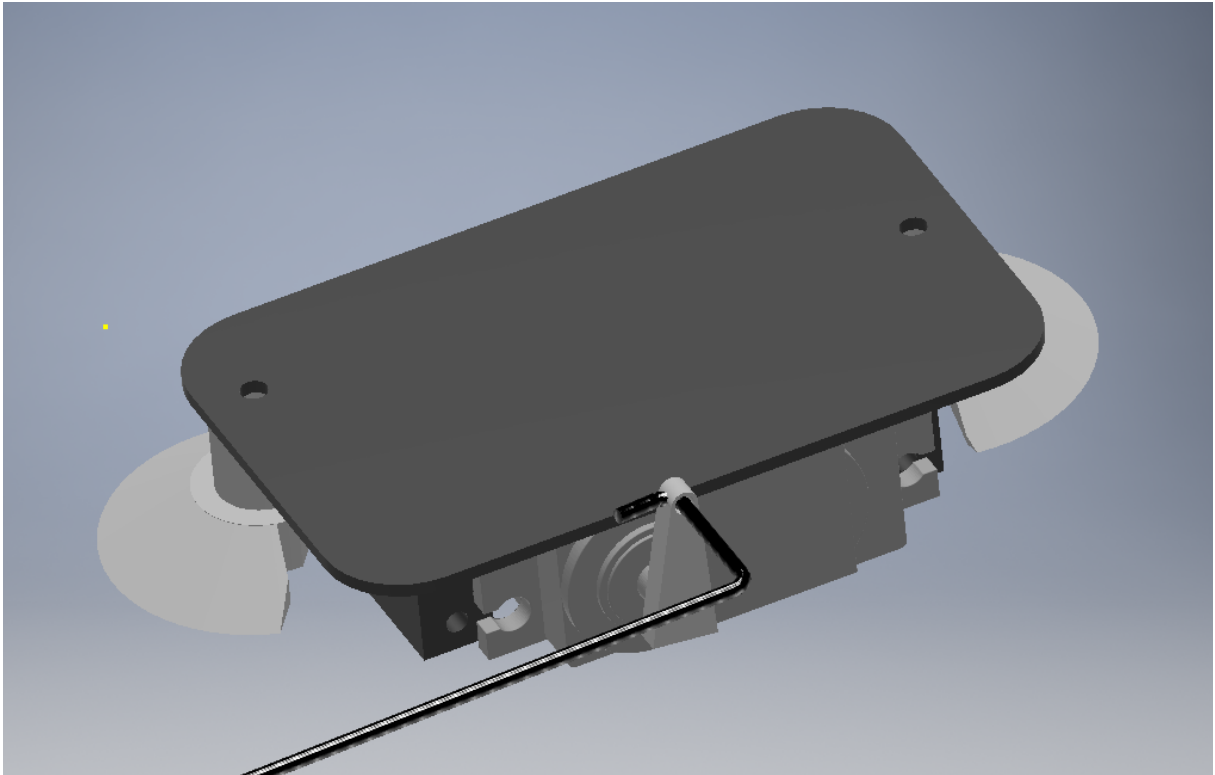
Figure 34 – 3D-printed servo mount structure. Source: original

The main structure was then covered in vinyl, for aesthetical and structural purposes (the tension on the vinyl helps making the structure stiffer). Vinyl is a material that shrinks when heated, which makes it tension itself over its surface.

The motor mounts were designed so they fit perfectly on the wing profile, and 3D-printed, glued and screwed into the main wing. The mounts can be seen in figure 35

Figure 36 – Motor pod design. Source: original

Figure 35 – 3D-printed motor mount structure. Source: original

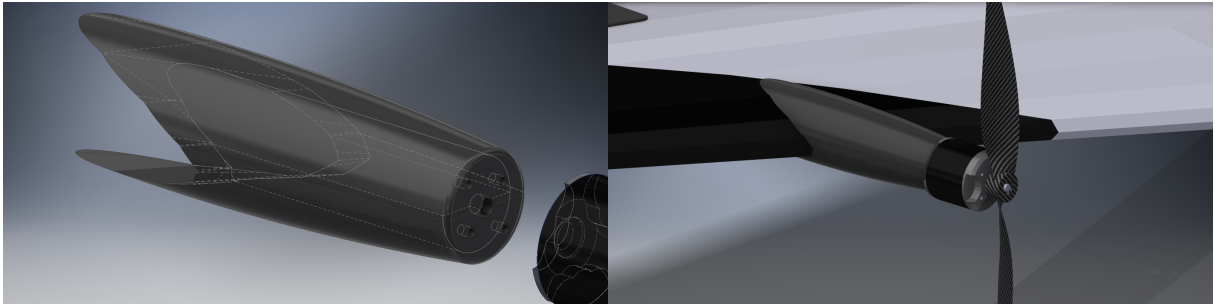The electronics bay was cut using hotwire and carved with a knife. A hot air blower was used to finish the inner surface. The components were placed keeping in mind flexibility to change the camera and batteries without affecting the center of gravity too much, maintaining approximately the same flight characteristics.

The flight controller was glued with vibration-dampening material. The battery was attached with velcro, and the remaining components are either glued or screwed in place. Special care was taken into keeping the magnetometers away from the motor and battery wires, as the induced magnetic field can adversely affect the magnetic readings.

The hinges were made using a type of fibrous tape. The tape was cut into pieces and glued onto itself, in such way that the piece of tape first sticks to the top of one section and to the bottom of the other section, taking care so that it does not stick to either section at their junction. These compound tapes are then glued in pairs, with one piece sticking to the bottom of the wing and the top of the elevon, and the other to the top of the wing and the bottom of the elevon, as illustraded on figure 37.
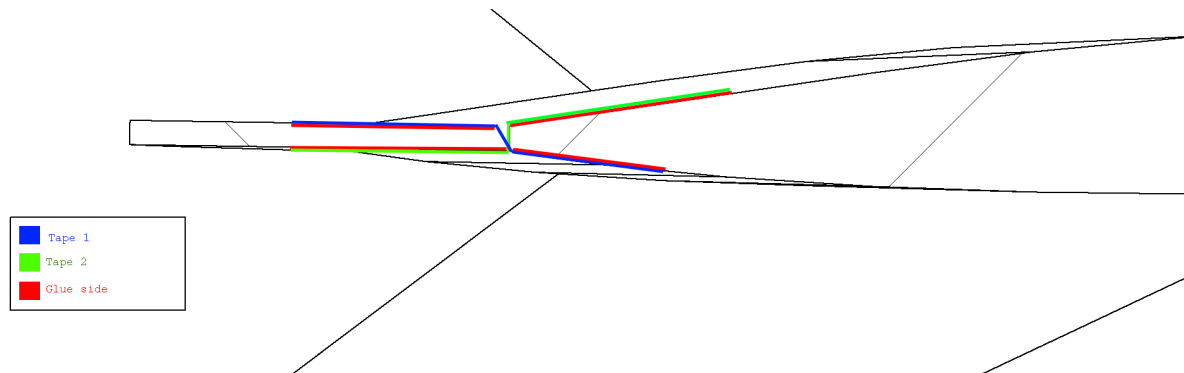

Figure 37 – hinges setup. Source: original

The winglets usually have only an aerodynamic function, as they increase the yaw (Z) stability and help avoid wing tip vortexes, but here they also need to work as a landing gear in VTOL mode.

As they need to let go after certain amount of force is exerted, and need to be removable to aid in transportation, a magnetic system was idealized. On the wingtip there's a 3D-printed profile with slots for the magnets, and the mirrored profile is also present on the winglet. This profile can be seen in figure 38. This profile made sure that four pairs of magnets touch on each winglet.However, this design allows slipping between the profiles, so Velcro was used again to help stiffen the structure without making it too hard. This allows the winglet to absorb impacts and come loose before damaging the rest of the aircraft.
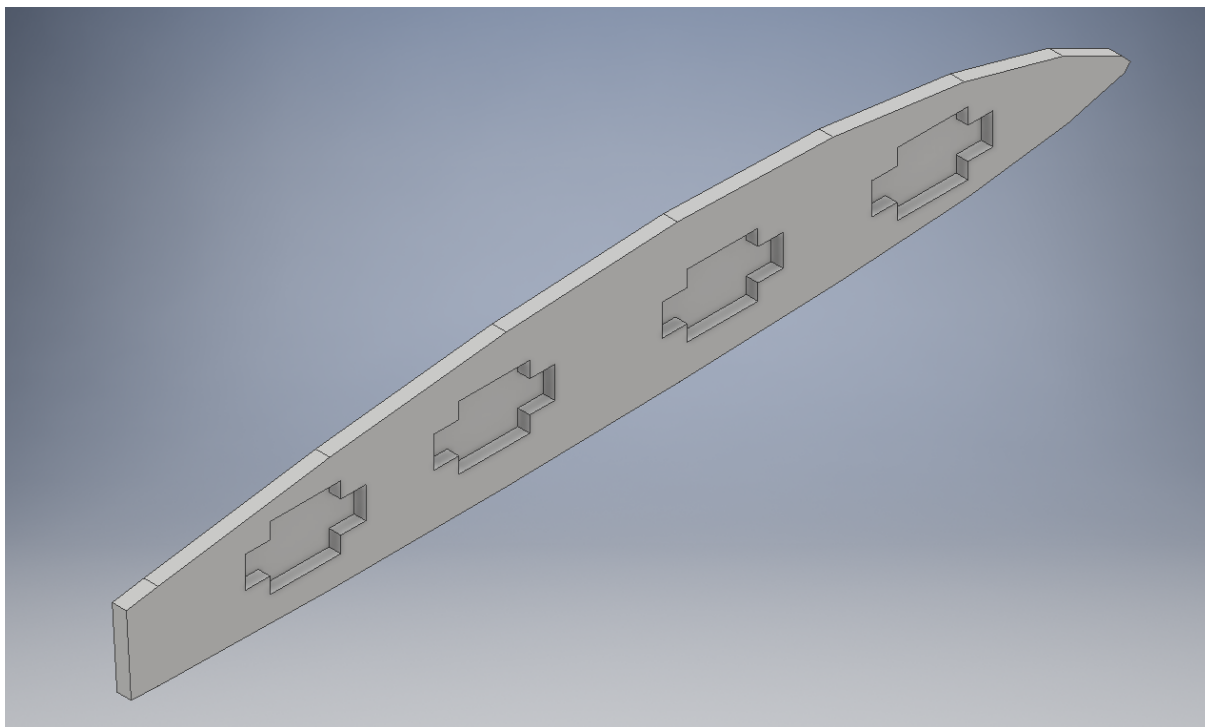
Figure 38 – 3D-printed magnetic coupler. Source: original

Figure 39 – 3D-printed magnetic coupler and winglet assembly. Source: original

7.3   Software Setup

In order to use the ArduPilot software stack to control a tailsitter, some setup is necessary. First the regular ArduPilot setup:

- Frame Type Configuration: The kind of aircraft frame needs to be chosen: in this case it is a tail sitter. This sets up the initial parameters and controllers, as well as output mixers.

- Compass Calibration: This step performs a calibration of the three magnetometers present in the board. Calibrated magnetometers are important for precise heading readings.

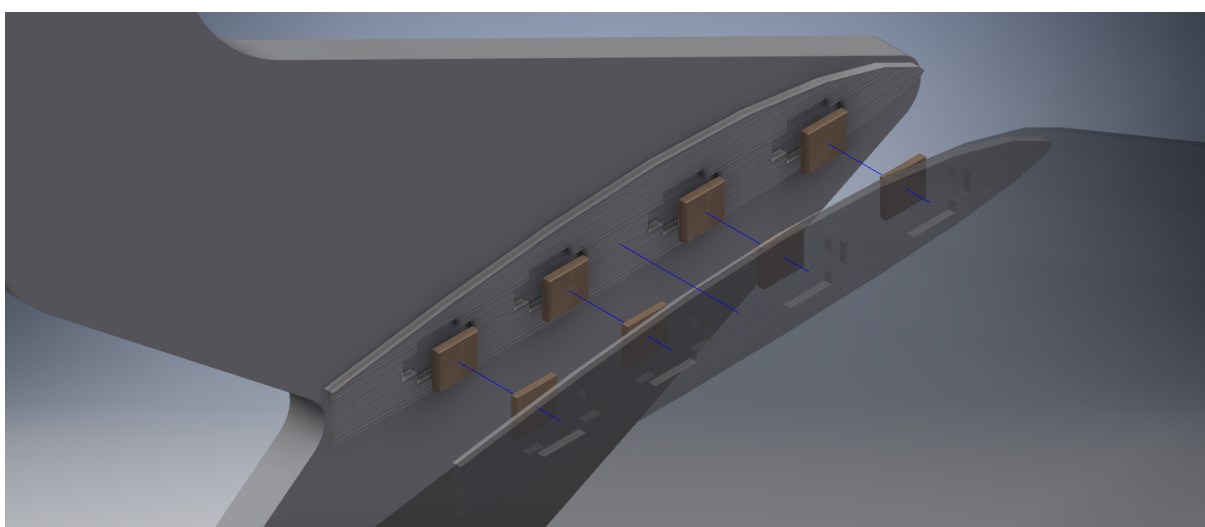- Radio Control Calibration: This recognizes the PWM ranges sent by the radio, so the flight controller knows when to arm/disarm, or apply full throttle.

- Accelerometer Calibration: This step calibrates the accelerometer. As the Pixhawk might not be leveled on the frame, calibrating the accelerometers is important to know the aircraft's real orientation.

- ESC Calibration: Just like the flight controller, the ESCs also need to know the full range of the PWM received from the Pixhawk, and thus need calibration;

After the basic setup, additional changes need to be done on the parameter level:

- AHRS_EKF_TYPE = 3 This makes the flight controller use an extended Kalman Filter that takes into consideration the accelerometer for translations, not only orientations.

- ARMING_CHECK,230 A custom pre-flight check is done, disabling the GPS checks due to the problems reported in 7.4.3.

- SCHED_LOOP_RATE = 300 This makes the Kalman filter update at 300 Hz, important for faster responses on multirotor-like aircraft, like this one in VTOL mode.

- SERVO3_FUNCTION = 73, SERVO4_FUNCTION = 74 This sets outputs 3 and 4 to output the mixers of left motor and right motor on a dual-motor tail sitter aircraft.

7.4 Troubleshooting

This section details some of the problems faced during this work and how each of them was handled.

7.4.1 The Electronic Speed Controllers Do Not Work

With The hardware setup, it was noted that the ESCs did not respond to the flight controllers. This could be due to two main reasons:

- The ESCs are unable to cope with the 400Hz PWM [1] signal generated by the flight controller;

---

[1] Pulse Width Modulation

- The signal voltage was not high enough;

The ESCs did answer properly to a 50Hz signal, so they were working. It was later found on the DiyDrones [31] forum that the ESCs are incompatible with the Pixhawk controller, and two components had to be removed for them to work. Upon further inspection, the components were noticed to be a resistor and a capacitor. This is a strong indication of an RC filter. The presence of an RC filter on the inputs, coupled with the output resistance present in most flight controllers signal outputs, resulted in a resistive divider, as seem in figure 40. This effectively lowered the voltage read on the microcontroller to 2, as seen in figure 42c. Further inspection showed that only one 512 (5100 Ω) resistor was present on the board, and it was, along with a capacitor, bridging a route to ground. The removal of these components was enough to raise the read signal value to 3.4 V, solving the issue, as shown in figure 42d.

With the ESCs accepting their input signals, they needed calibration. The calibration of an RC ESC is a process where it learns the high and low bounds on its input signals. The ESC is turned on with the maximum possible input, it beeps, and the signal can be lowered to the minimum, then it beeps again.
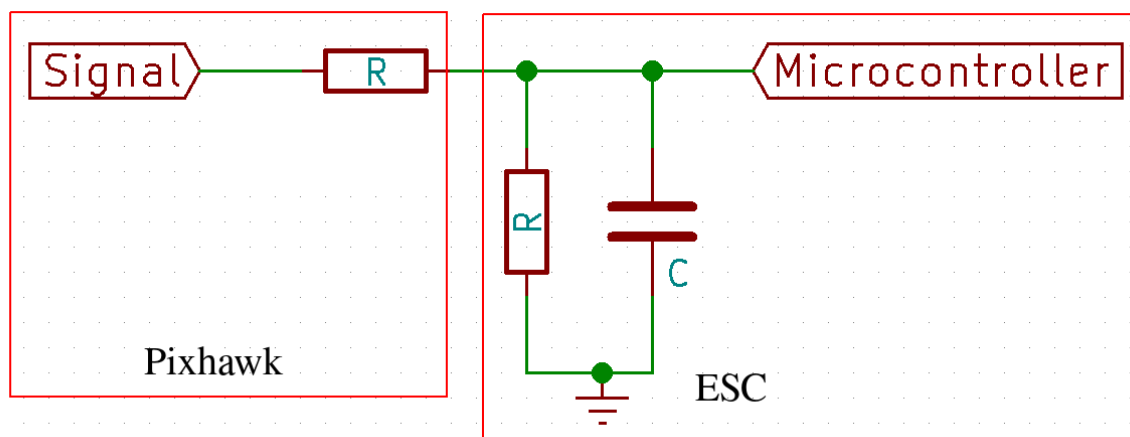

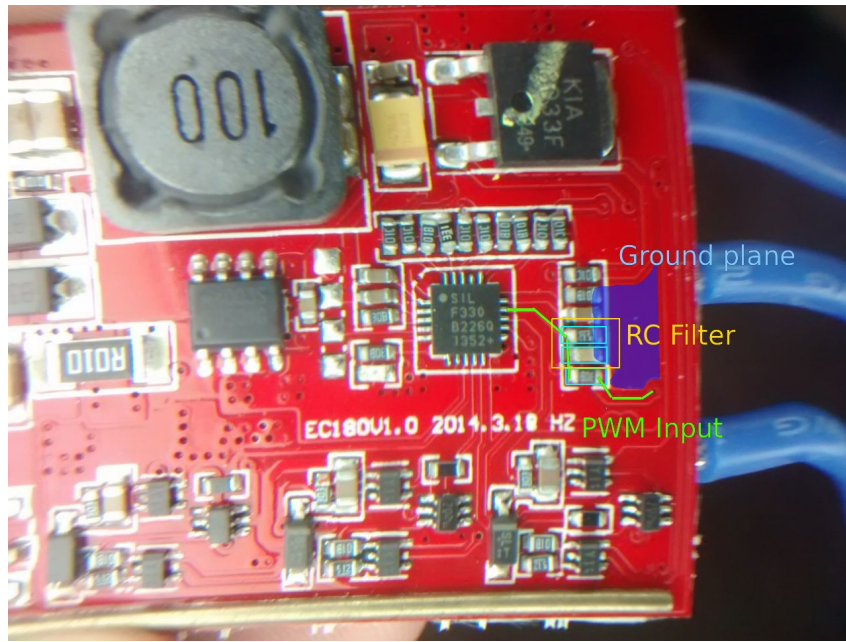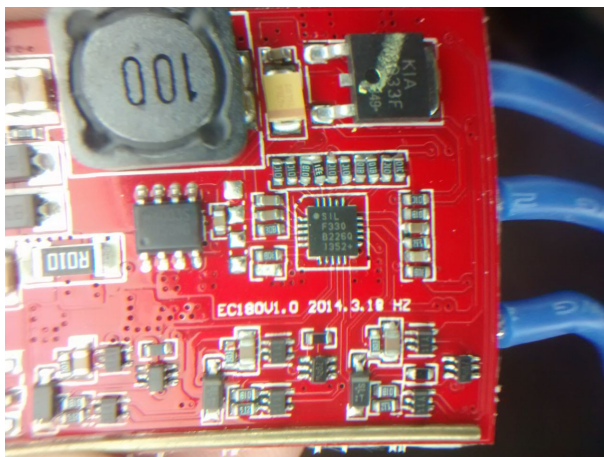Figure 40 – Schematic of signal path between Pixhawk and ESC. Source: original
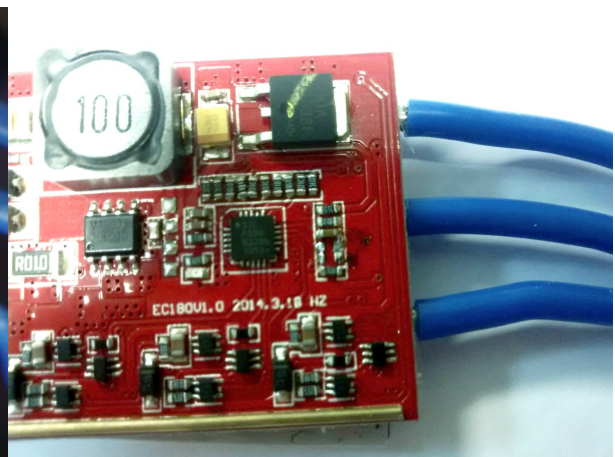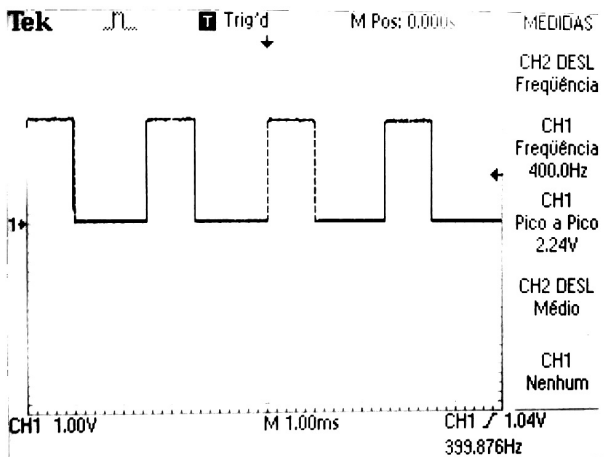
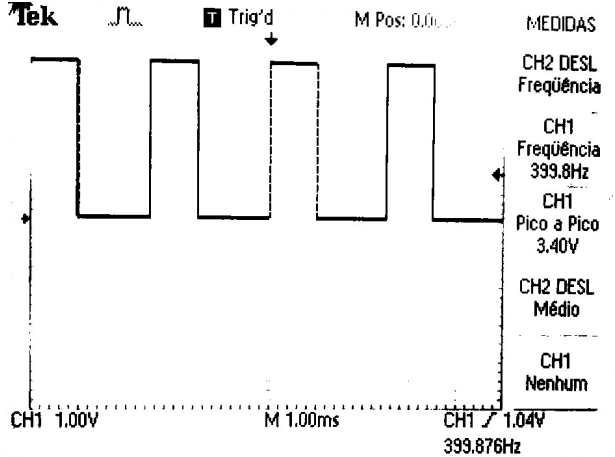Figure 41 – Schematic overlaid on ESC board. Source: original

(a) ESC before modification. Source: original

(b) ESC after modification. Source: original

(c) Signal before modification. Source: original

(d) Signal after modification. Source: original

Figure 42 – Modifications on the ESC.

7.4.2 The Elevons Have a High Frequency Pitch oscillation

Even on the ground, activating the stabilization control resulted in increased high-frequency oscillations of the control surfaces on the pitch direction. Any minor servo correction caused a small movement on the aircraft body, due to moment conservation. This movement is detected and, when trying to compensate, the behavior repeated, until the oscillation peaked with the maximum amplitude reachable by the servos. This behavior was linked to the derivative terms on the pitch controllers. Since, as shown on picture 25, the roll and pitch controls are attenuated with the throttle, this effect is not present during flight. This effect should be handled in software in the future, but was not prejudicial to the tests in this project.

7.4.3 Bad GPS Health

The GPS used, even though recognized by the flight controller, made it show "Bad GPS Health messages". Further research showed that the board was a badly manufactured clone [32], where the wrong version of the EEPROM chip was used, with a different pinout, meaning that while the flight controller was able to communicate and setup the GPS, it was unable to perform a warm start, which is looking for the right satellites using its last known position saved on the EEPROM.

This issue has three possible solutions:

- Unsolder the chip and resolder to the right connections with wires.

- Replace the chip with the correct one.

- Replace the whole GPS with a working one.

The latter was chosen as there was a spare one available, requiring only re-wiring.

8 Assessment

The assessment was planned with incremental tests. First, the VTOL capabilities would be tested, with stabilized flights, then the position hold (loiter), followed by autonomous taking-off, then landing. With the VTOL modes working, it would be time to test the transition to forward flight, the flight itself, and the transition back into VTOL mode.

The tests performed are better described in the following sections, along with their results.

8.1 Tethered Attitude Control Test

To test the attitude control and stabilization, the prototype was hang by a rope, so it's range of movement was restricted, and it was safer to test indoors. The first tests were qualitative. The wing was armed on the QStabilize mode, where the gyroscope and accelerometer are used to try to maintain the aircraft leveled in VTOL mode (propellers spinning parallel to ground). The expected result was that the elevons should move trying to stop the movement, even without propellers on the motors (again, for safety reasons). The aircraft successfully reacted to disturbances on its attitude by moving the control surfaces appropriately.

This test was repeated with propellers, perturbations were applied to the wing by manually trying to turn it on each of the three axis. Each axis was tested at least twice on each direction, and the aircraft reacted re-orientating itself successfully on all occasions.

8.2 Un-tethered Attitude Control Test

For this test, the wing was taken to an open field in the university. For the take-off, it had to be oriented so the wind (of around 13km/h) blew parallel to its surface, so that the wind didn't flip it over. Take-off can't be too slow, as the winglets adhere to the ground and can cause the aircraft to tip over. Once in the air, the controls and stabilization worked well, but once the wind hit the aircraft, it turned perpendicular to the direction of said wind, and the control authority was not enough for both stabilizing flight and turning the yaw axis. While this problem limits the yaw controllability in VTOL mode, the position control is not necessarily affected, as the aircraft can still move in a mixed attitude between VTOL and fixed wing, inclined against the wind and maintaining position.

This could possibly be fixed by increasing the winglets area, however this also increases the area the wind hits, and needs more testing to verify. Another possibility is tweaking the pitch angle limits, which by default are $\pm 30°$, not enough to fight the wind in this case.

The flight path can be seen in figure 43, and an in-flight photo in figure 44. The video is available on youtube [33]. No attempt at transitioning to fixed-wing mode was made at this flight due to the reduced space available, which made the pilot feel unsafe.

Figure 43 – Visualization of first test flight. Source: original

Figure 44 – Photo of first test flight. Source: original

8.3 Position Hold

The next test was taking-off and landing autonomously. Due to the winds and undersized landing gear, the aircraft was positioned with the surface again parallel to the wind flow.

The autonomous take-off however, tried to take-off too slowly, and the winglet/-landing gear grip to the grass limited pitch control prior to taking-off, causing the wing to
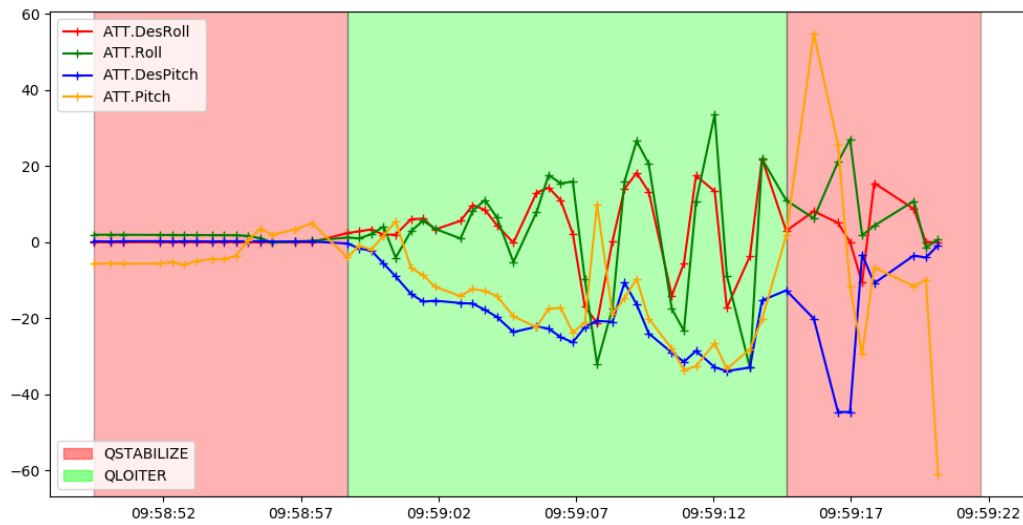
Figure 45 – Visualization of logged attitude control. Source: original

Test #	Flight Time (s)	Position Hold Radius (m)	Maximum Altitude (m)	Notes
1	47	1.5	14.4	
2	47	8.1	12.0	Roll oscillation
3	56	3.0	13.1	
4	37	3.7	4.3	Roll oscillation
5	51	7.5	13.9	Pushed by the wind

Table 4 – Loiter tests summary. Source: original

tip over. The solution was to take-off manually, in QStabilize (Quadplane Stabilize, where the UAV attempts to stay level, but there's no position control) mode, then switching to QLoiter (level plus position control). Five of such flights were attempted. The results are on Table 4. Some of these tests are available online [34].

While most of the results were good, a roll oscillation was present on some of the flights, causing instability and forced landings. The problem appears to be caused by the navigation controller, as the roll("ATT.Roll") actually follows the roll setpoint("ATT.DesRoll"), which is oscillating, as seem in figure 45. This could mean that the navigation controller is oscillating fast around the given point, with increasingly high amplitude, requiring further tuning of its parameters.

8.4   Test Conclusions

Although not all tests could be done on time, the VTOL capabilities were tested and the results were very satisfactory, with only two minor issues. The first issue is that, on an autonomous take-off, the motors speed up too slowly, taking the aircraft off balance

before it gets off the ground, where its control surfaces are not very effective, as the friction holds the winglets in place, impairing the pitch control. The second one is that the winglets/landing gear needs to be a little larger and better attached to the aircraft, otherwise it tips over during landing.

Further tests need to be made on the transition into forward flight and on the forward flight itself.

9 Conclusions

The objective was an UAV system able to autonomously take-off, hover, transition into fixed-wing mode, fly autonomously over a pattern, transition back into hover mode, and land autonomously.

The proposed system was built in foam and tested. The test performed involved only take-off and hovering. The aircraft behaved well in the harsh weather conditions, with 13 km/h winds.

While taking off and hovering are two of the trickiest parts, two crucial parts remain to be tested: landing and transitioning, specially automatically.

The test had promising results, indicating that it is very much possible to have such an aircraft, but more tests are required to verify the remaining aspects.

The results, as they are, are deemed satisfactory.

For a mass-produced aircraft, however, a new landing-gear/winglet system should be thought of, one that is more tolerant to harsh lands, prevents the aircraft from tipping over, but still performs well aerodynamically. The electronics bay should also be made of a stronger material than foam, as even low-speed crashes can loose the components within it. The motors should be better analyzed, and maybe changed for stronger ones, as while they seem to be under their maximum current (the wing currently uses 30 A on hover, meaning each motor is consuming around 15 A, leaving a 10 A margin for maneuverability), they are too hot to touch upon landing, which could be harmful.

Bibliography

1 SENSEFLY Datasets. Available From Internet: <https://www.sensefly.com/drones/example-datasets.html>.

2 SNOW, C. *Drone Analist: The Truth About Drones in Mapping and Surveying.* Available From Internet: <http://droneanalyst.com/2016/07/27/the-truth-about-drones-in-mapping-and-surveying>.

3 DRONEDEPLOY: Commercial Drone Industry Trends. Available From Internet: <https://cdn2.hubspot.net/hubfs/530284/10M_Acre_Report_2017_.pdf?blog>.

4 ETH Zurich - Autonomous Systems Lab. Available From Internet: <http://www.asl.ethz.ch/publications-and-sources/publications.html>.

5 MEECE, S. A bird's eye view of a leopard's spots. the Çatalhöyük 'map' and the development of cartographic representation in prehistory. *Anatolian Studies. 56: 116. JSTOR 20065543*, 2006.

6 RAAFLAUB, K. A.; TALBERT, R. J. A. Geography and ethnography: Perceptions of the world in pre-modern societies. *John Wiley & Sons. p. 147. ISBN 1-4051-9146-5.*, 2009.

7 BUISSERET, D. Monarchs, ministers and maps: The emergence of cartography as a tool of government in early modern europe. *Chicago: University of Chicago Press*, 1992.

8 OPENMVS. Available From Internet: <https://github.com/cdcseacave/openMVS>.

9 MOULON, P. et al. *OpenMVG. An Open Multiple View Geometry library.* Https://github.com/openMVG/openMVG.

10 VERLING, S. et al. Full attitude control of a vtol tailsitter uav. In: *2016 IEEE International Conference on Robotics and Automation (ICRA)*. [S.l.: s.n.], 2016. p. 3006–3012.

11 XFLR5. Available From Internet: <http://www.xflr5.com>.

12 U-BLOX Neo-M8N BN-800 GPS accuracy test. Available From Internet: <https://www.youtube.com/watch?v=eoNqqr3EUWU>.

13 TURNIGY 9x Range? Available From Internet: <https://www.rcgroups.com/forums/showthread.php?1403635-Turnigy-9x-Range>.

14 ER9X Firmware Upgrade for 9X Turnigy Radio. Available From Internet: <https://www.flitetest.com/articles/er9x_Firmware_Upgrade_for_9X_Turnigy_Radio>.

15 OPENTX. Available From Internet: <http://www.open-tx.org/>.

16 HISTORY of Ardupilot. Available From Internet: <http://ardupilot.org/planner2/docs/common-history-of-ardupilot.html>.

17 GNU Lesser General Public License v3 (LGPL-3.0). Available From Internet: <https://tldrlegal.com/license/gnu-lesser-general-public-license-v3-(lgpl-3)>.

18 WAYBACKMACHINE: pixhawk.ethz.ch. Available From Internet: <https://web.archive.org/web/20080901000000*/pixhawk.ethz.ch/>.

19 COMPUTER Vision and Geometry Lab of ETH Zurich (Swiss Federal Institute of Technology). Available From Internet: <http://cvg.ethz.ch/>.

20 AUTONOMOUS Systems Lab. Available From Internet: <http://www.asl.ethz.ch/>.

21 BSD 3-clause. Available From Internet: <https://tldrlegal.com/license/bsd-3-clause-license-(revised)>.

22 DRONECODE. Available From Internet: <https://www.dronecode.org/>.

23 LINUX Foundation. Available From Internet: <https://www.linuxfoundation.org/>.

24 ARDUPILOT and DroneCode part ways. Available From Internet: <http://diydrones.com/profiles/blogs/ardopilot-and-dronecode-part-ways>.

25 ARDUPILOT - Code Overview (Copter). Available From Internet: <http://ardupilot.org/dev/docs/apmcopter-code-overview.html#apmcopter-code-overview>.

26 MAVPROXY. Available From Internet: <http://ardupilot.github.io/MAVProxy/html/index.html>.

27 APM Planner 2. Available From Internet: <http://ardupilot.org/planner2/>.

28 KALMAN, R. E. A new approach to linear filtering and prediction problems. *Transactions of the ASME–Journal of Basic Engineering*, v. 82, n. Series D, p. 35–45, 1960.

29 EXTENDED Kalman Filter Navigation Overview and Tuning. Available From Internet: <http://ardupilot.org/dev/docs/extended-kalman-filter.html>.

30 PARK, S.; DEYST, J.; HOW, J. P. A new nonlinear guidance logic for trajectory tracking. In: *AIAA Guidance, Navigation, and Control Conference (GNC)*. Providence, RI: [s.n.], 2004 (AIAA 2004–4900). Available From Internet: <http://acl.mit.edu/papers/gnc_park_deyst_how.pdf>.

31 GETTING Pixhawk to work with Emax. Available From Internet: <http://diydrones.com/forum/topics/getting-pixhawk-to-work-with-emax?id=705844%3ATopic%3A1912186&page=1#comments>.

32 UBLOX neo 6M connection to APM 2.6. RCGroups.com. Available From Internet: <https://www.rcgroups.com/forums/showthread.php?2065386-Ublox-neo-6M-connection-to-APM-2-6/page2>.

33 TAIL-SITTER VTOL test #1. Available From Internet: <https://www.youtube.com/watch?v=cPDwxvqjEyA>.

34 SECOND Flight Test Playlist. Available From Internet: <goo.gl/P8ZvBS>.