University of Sidi Mohamed Ben Abdellah

Faculty of Sciences and Technologies Fez

Industrial Engineering Department

Internship thesis

Prepared by:

Zarzouri Ayoub

Specialty : Mechatronics Engineering

Entitled

Autonomous Navigation and Flying for fixed wings aircraft: Small Unmanned Aerial Vehicles

Location: TUBITAK UZAY Ankara Turkey

Ref: /IMT15 –Internship 2

Presented the: .../ 11/ 2016 in front of:

- Pr.Lhabib Hamedi
- Pr.
- Pr.

Abstract

The Unmanned Air Vehicle sector has become the most dynamic growth sector in the aerospace industry. There are more and more civilian 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 TÜBİTAK UZAY (Space Technologies Research Institute) 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 2.5, 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 different 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 simulator. In this test, the autopilot and the ground control station system worked correctly. Next, the system was installed on a small aircraft. 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 fixed wings aircraft. This autopilot is ready to be used in the UAV that is being developed.

Résumé

Le secteur véhicule aérien non habité (UAV's en anglais) est devenu le secteur de la croissance la plus dynamique dans l'industrie aéronautique. Il y a des applications de plus en plus de civils qui profitent grandement de l'utilisation de ces appareils.

L'objectif de ce travail est de tester et de mettre en œuvre un pilote automatique et une station de contrôle au sol pour une avion qui est parrainé par TÜBİTAK UZAY(Institution de la recherche et les technologies d'univers) Il est également prévu de mettre en œuvre un système Personne de vue à distance avec le pilote automatique .

Le pilote automatique qui a été choisi pour être mis en œuvre est le ArduPilot APM 2.5, qui est une source ouverte, à faible coût pilote automatique. Cela permettra de réduire le coût total du projet, tout en offrant des performances similaires à autopilotes commerciales. Ce pilote automatique peut être utilisé dans différents types de véhicules. Pour la station de contrôle au sol, il est utilisé le Mission Planner.

Afin de tester en toute sécurité le pilote automatique sans risque d'endommager tout composant, il a d'abord été installé et testé sur un simulateur. Dans cet essai, le pilote automatique et le système de surveillance travaillé correctement. Ensuite, le système a été installé sur un petit avion. Plusieurs essais en vol ont été effectués et il a été vérifié que le pilote automatique fonctionnait correctement, même avec les paramètres par défaut.

Ce travail a permis la mise en œuvre d'un pilote automatique à faible coût qui a de bonnes performances de qualité. Ce pilote automatique a prouvé sa flexibilité lors de leur mise en œuvre dans différentes plates-formes. Ce pilote automatique est prêt à être utilisé dans l'avion qui est en cours d'élaboration.

DEDICATION

I dedicate all my efforts and struggles of the educational life to my dear parents, without them I'm meaningless. Also I devote the work of this internship report to respectable and honorable professors who taught and supported me in developing my personality as a competent professional.

ACKNOWLEDGEMENT

First of all I thank Allah, who is the holder of my breaths, without his order nothing is possible.

I am highly thankful to my respectable brothers, teachers, friends and family members who were my supporter throughout my educational career, further I thank all the employees of the TUBITAK UZAY, where I have done my internship.

I have set light, an ever-burning flame of gratitude and deep sense of obligation to my honorable teachers for their valuable guidance, constructive criticism and inspiring attitude during my studies, especially Mrs.Lhabib Hamedi responsible of the Mechatronics engineers.

I appreciate and thank all the members of the faculty starting by its head master Mrs.M.IJJAALI and so on.

Not forget, great appreciation goes to all members of TUBITAK, my supervisor Mrs. Samat Akcay , Mrs.Osman Akgun project manager of his strong believe on me ,Mrs.AbdRahman Aydimir for his help to get this position ,Mrs. Murat ,Mrs. Omar for their technical support and the rest of Aerodynamic Department ,who helped me to get my project done and to write my internship report from time to time during the project. The whole program really brought us together to appreciate the true value of friendship and respect of each other.

Last words are lacking to express my feelings and indebtedness to my affectionate Family and Friends for their love, appreciation, which always stand by me mentally and spiritually during all years of my study.

Table of contents

List of Figures

List of Tables

List of Abbreviations

AoA	Angle of Attack.
AHRS	Attitude and Heading Reference System.
CG	Center of Gravity.
DoF	Degree of Freedom.
ESC	Electronic Control Speed
FPV	Flight Person View
FDM	Flight Dynamic Model
GCS	Ground Control Station
GNC	Guidance, Navigation and Control
GPS	Global Positon Satellite
GUI	Graphical User Interface
HIL	Hardware in the Loop
IMU	Inertial Measurement Unity
LiPo	Lithium Polymer
MAVLink	Micro Air Vehicle Link
MP	Mission Planner
NED	North-East-Down
PID	Proportional-Integral-Derivative.
PWM	Pulse Width Modulation
RC	Radio Controller.
RPV	Remotely Piloted Vehicles.
UAV	Unmanned Aerial Vehicle.
UDP	User Datagram Protocol
SITL	Software in the Loop

General Introduction

Unmanned Aerial Vehicles (UAV's) are and have been prominent within both military and government operations during the most recent years. With the rapidly advancing technology, these drones are becoming increasingly effective and significantly less costly. The development has piqued the interest of many organizations with intentions to explore and adapt UAV's as one of their modern solutions. Mapping, surveying and search/rescue missions are some of the applications where the attributes of an UAV are appealing. These are functions not only valuable in modern warfare, but also for nonmilitary purposes. There is a lot to gain by replacing the pilot with an automated system, it is cheaper, always available and can be customized. The most important aspect is that it makes the airspace an accessible medium where airborne vehicles can be used to aid in numerous operations. Using UAVs for search and rescue has advantages in both response time and need for manpower compared to piloted aircrafts. These are qualifications that attract organizations and countries such as Turkey.

Our work done in Aerodynamics department of TÜBİTAK UZAY in Turkey aims to implement and analyze the performance that can be achieved with an open source Autopilot that is cheaper than commercial autopilots used in modern UAVs.

The ArduPilot system, as an open source solution, is design to fit a wide range of airframes including fixed wings we want to build in this project. Also, it allows the user to reprogram the code to fit a particular aircraft and tune the PID gains to achieve the desired performance.

The main objective of this project is to develop a set of tools and instructions that the institution would use to flight test its own UAVs.

To accomplish the main objective it will be necessary to:

- Gain understanding on the ArduPilot system (Code and Ground Control Station).
- Simulate and test the Ardupilot using real flight simulators before real flight.
- Develop an Autopilot module to be integrated into the MATLAB Simulink for numeric simulation.
- Install the ArduPilot system in a commercially available airframe.
- Flight test the system to gain knowledge on its features and behavior.

To achieve this we divided our rapport to the following chapters:

Chapter 1: Consisted of description of the institution support the project.

Chapter 2: Has the role of explaining the technology and describing the problems and solutions proposed.

Chapter 3: Aims to implement solutions had chosen (Ardupilot Mega 2.5) and droop light of their prices.

Chapter 4: Showed different ways to simulate the Ardupilot technology.

Chapter 5: Discussed the results, failed and future work.

Key words: UAV, Autopilot, Ardupilot Mega, Ground Control Station.

Chapter 1: Description of the Company

1. Introduction

In this chapter we will try to give a general overview of TÜBİTAK UZAY describing history, vision, number of employees and their duties and organization structure as it is nowadays.

2. Description of the Company

1.1. Company name

Space Technologies Research Institute (TÜBİTAK Uzay Teknolojileri Araştırma Enstitüsü).

1.2. Company Location

ODTÜ Yerleşkesi, 06800, Ankara/ Turkey (Figure 1).

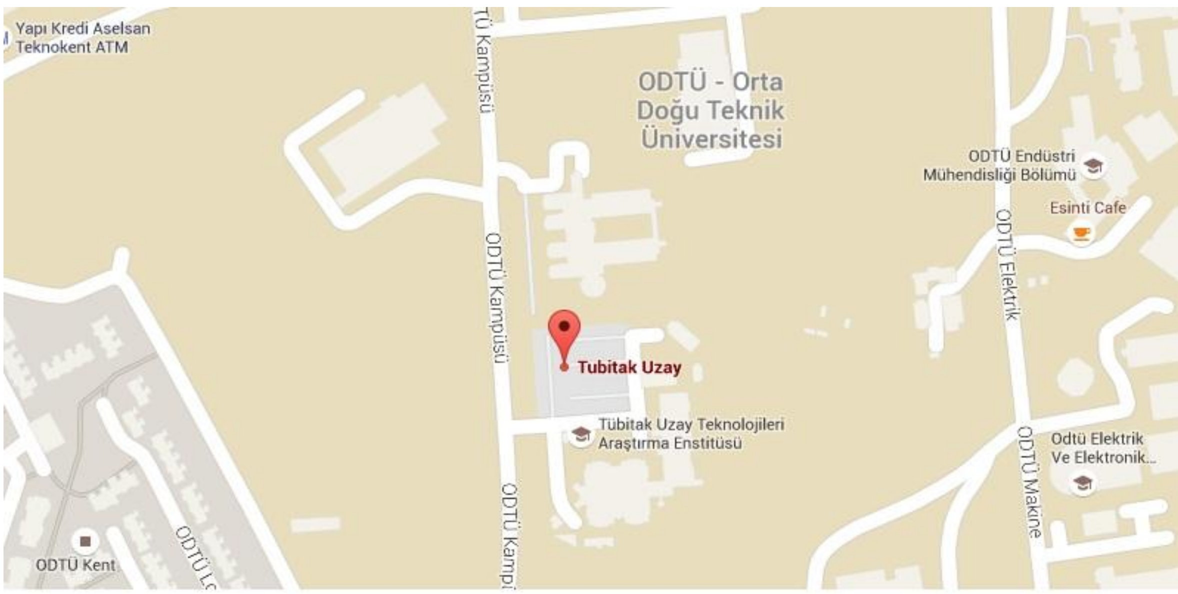


Figure 1: Company Location

3. TÜBİTAK UZAY: Vision and Interest

TÜBİTAK Space Technologies Research Institute (TÜBİTAK UZAY), (Figure 2), is an applied research, technology development, and application-oriented institute. Its core competence is in satellite technologies in system and subsystem levels, however, it's diversified its interest into communication systems, remote sensing and data processing, and aerospace technologies.

Its interest in remote sensing is focused on post-processing of aerial and satellite images and developing algorithms and software tools to provide usable information for decision makers in areas such as precision agriculture and disaster management.

Recently, it has expanded its interest into aerospace technologies such as small unmanned systems to support Turkey's precision agriculture studies, high altitude long endurance unmanned systems for large area surveillance and mapping, aero-structures, and aero-thermal systems.

TÜBİTAK UZAY has its roots in Ankara Electronics Research and Development Institute, which were founded in 1985 as a publicly funded research institute. Later in 1991, the institute expanded its research areas into the information technologies. Along with this change of focus, the institute was renamed as Information Technologies Research Institute (BİLTEN) in 1995. Starting in 2001, it gravitated more to space domain in following years. Due to increasing focus on satellite technologies, in 2006, Institute's name was changed to the current name.

Figure 2: TÜBİTAK UZAY Ankara

Furthermore, TÜBİTAK UZAY is a member of the Committee on Earth Observation Satellites (CEOS) and a supporting member of the ISPRS (International Society for Photogrammetric and Remote Sensing). It also represents Turkey in the Asia-Pacific Space Cooperation Organization (APSCO).

4. Organizational Structure of the Company

Schematic organization :(Figure 3)

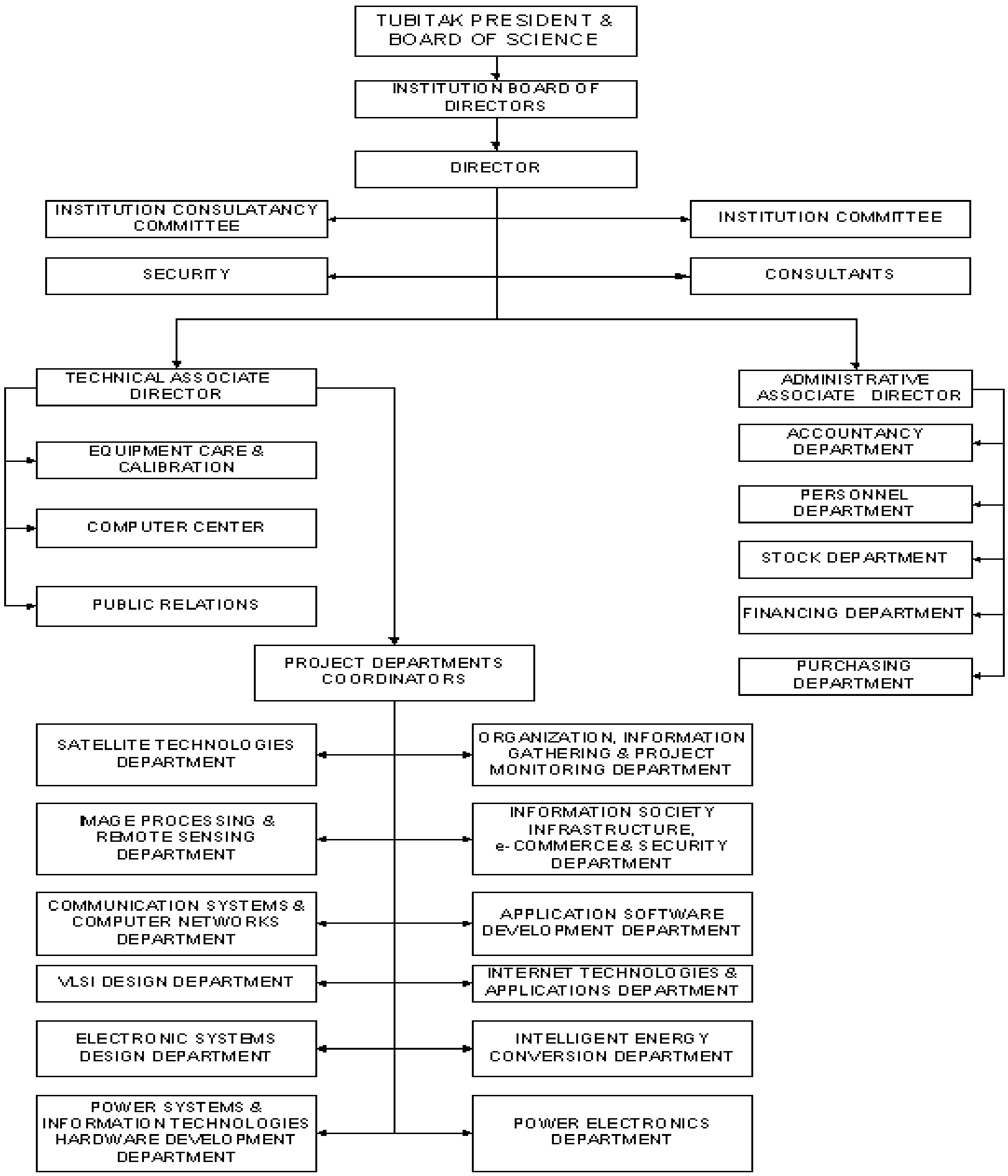


Figure 3: Company Organizational Structure

5. Number and Duties of Engineers Employed

The statistical data for the exact numbers of engineer employees is not found. However, the approximate values for the number and duties of the engineers are:

Electronics Engineer 165

Computer Engineer 60

Industrial Engineer 30

Other Researchers 25

Technician 15 Staff 20

TOTAL 315

6. Brief History of TUBITAK UZAY

UZAY was founded in 1985, with the name of Ankara Electronics Research and Development Institute, by a protocol signed by Middle East Technical University (METU) and the Scientific and Technological Research Council of Turkey (TUBİTAK). The Institute is located in the campus area of Middle East Technical University.

1984 TUBITAK-METU protocol signed.

1985 Ankara Electronics Research and Development Institute (TAEAGE).

1995 Information Technologies and Electronics Research Institute (BİLTEN)

1998 New building in METU Campus.

2000 TSE ISO 9001

7. Areas of Business

- Satellite Technologies.
- Communication Systems.
- Data and Image processing.
- Air platform design and Reliability Services.

8. Conclusion

In this chapter we described a general view of our project and we put it in its proper frame, besides, we presented the organizational structure of the company and some general information, in the next chapter we are going to introduce next step of our project which is the analyze functional .

Chapter 2: Description of the project

1. Introduction

This chapter deals with the fundamental physical components and properties of conventional fixed-wings aircraft. Describing its different part, defining some basic aerodynamic concept which are directly related to our project, Furthermore explaining the UAV's (Unmanned Aerial Vehicles) system and its types, also highlighted problems and solutions proposed.

2. Definition of an Aircraft

Figure 4 illustrates a conventional fixed-wing aircraft that is the basic flight vehicle of interest in this project. The key physical components, or subsystems, that define the aircraft are the fuselage, the wings, the horizontal tail, the vertical tail, and the propulsion system.

The fuselage provides working volume for passengers, cargo, and aircraft subsystems that are internal to the aircraft. The fuselage is important in terms of achieving particular flight missions, but it is not especially important from a flight performance perspective. The two wings are crucial for flight, since their main purpose is to generate lift.

The aircraft illustrated in Figure 4, and all aircraft considered hereafter, are fixed-wing aircraft, since the wings are rigidly attached to the fuselage. This is in contrast with helicopters or other rotary wing flight vehicles that generate lift using rotating blades.

Other important flight subsystems, illustrated in Figure 4, are the horizontal tail, the vertical tail, and the engines. The horizontal and vertical tails are rigidly attached to the fuselage as indicated. The horizontal tail provides longitudinal stability and control capability, while the vertical tail provides directional stability and control capability. The engines are crucial flight subsystems, since they generate the thrust force that acts on the aircraft.

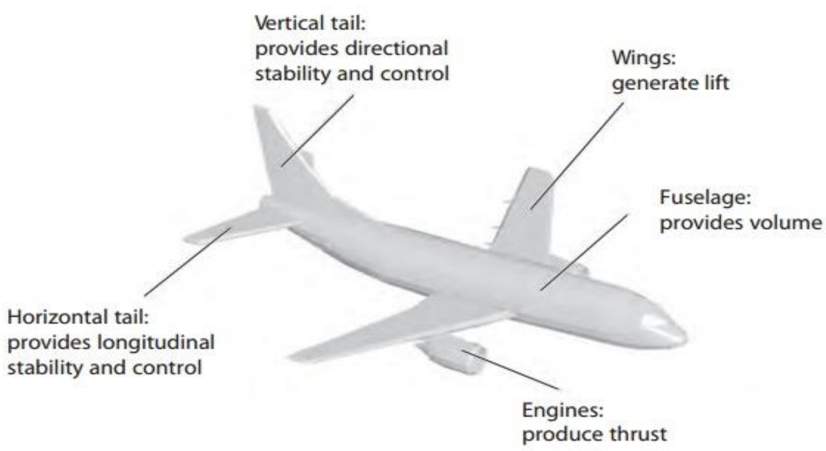


Figure 4: Conventional Aircraft

3. Aerodynamic Control Surfaces

An airplane has three fundamental control surfaces (Figure 5): ailerons, elevators and a rudder. Within the cockpit, two controls operate the control surfaces. The control stick controls the ailerons and elevators. The rudder pedals control the rudders.

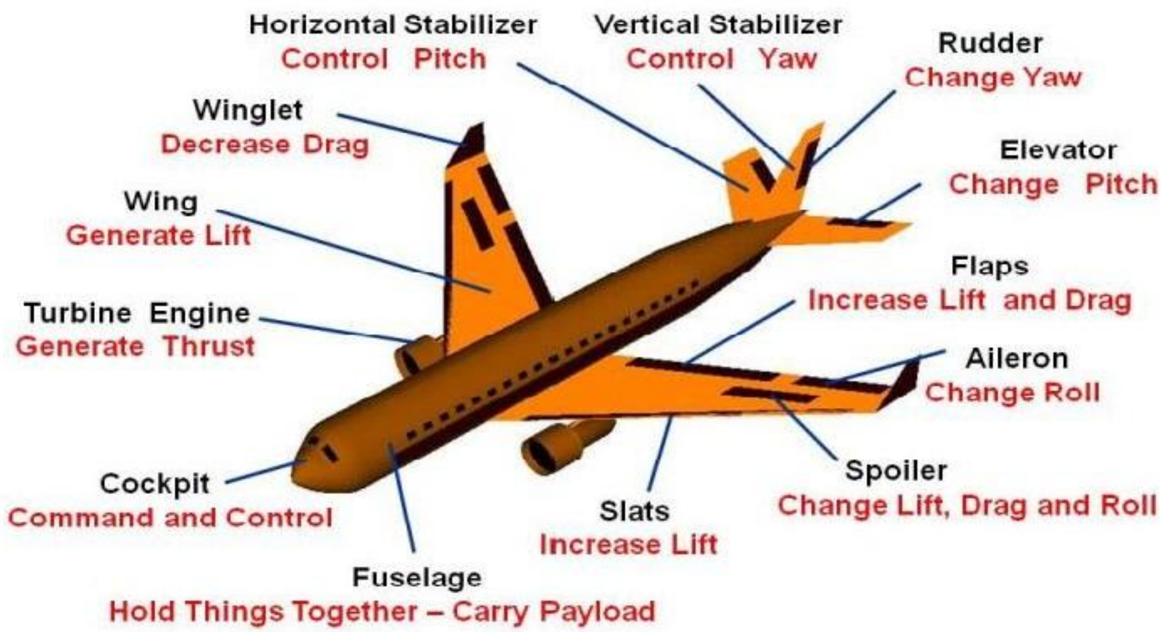


Figure 5: Airplane parts and function

The table 1 below explains flow from pilot controls to control surface to motion.

Control in cockpit	Control Surface	Motion
Control Stick (right and left) ailerons roll	Control stick (front and back) elevators pitch	Rudder pedals rudder yaw

Table 1:control surface to motion

3.1. The Ailerons

The ailerons (Figure 6) are flap-like structures on the trailing edge of the wings -one on each side. When the pilot moves the control stick to the right, the right aileron will tilt up and the left aileron will tilt down. This will cause the airplane to roll to the right. When the pilot moves the control stick to the left, the left aileron tilts up, the right aileron tilts down and the airplane rolls to the left. This happens because as the aileron tilts downward (effectively increasing camber) more lift is created and the wing rises. As it tilts upward, less lift will be created and the wing will descend. If the wing on one side of the airplane rises and the other descends, the airplane will roll towards the side of the decrease in lift.

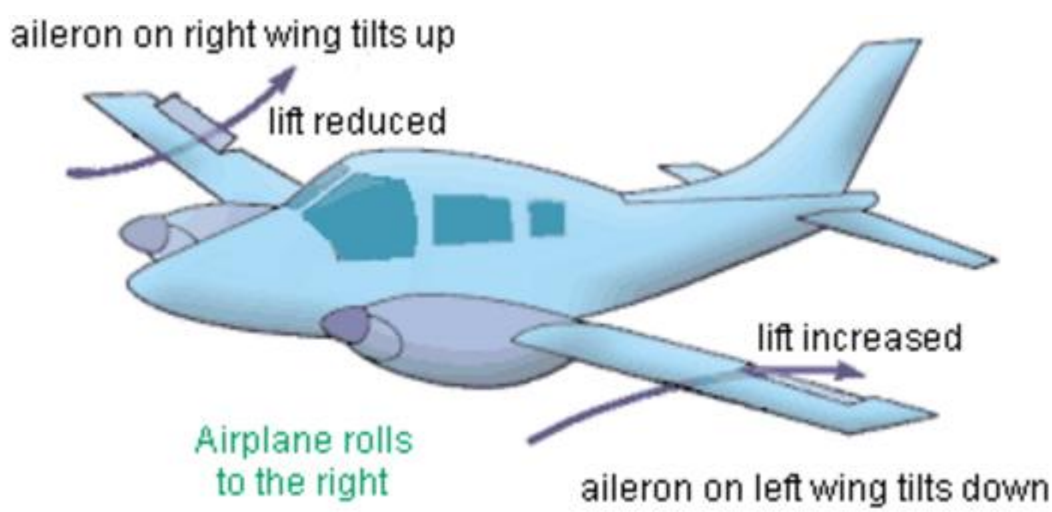


Figure 6 : Aileron in the aircraft fixed wings

3.2. The elevators

The elevators (Figure 7) are also flap-like structures that are mounted on each side of the horizontal stabilizer. When the pilot pushes the control stick forward, the elevators tilt downward (both in the same time). This causes the tail of the airplane to rise and the fuselage to tilt down - this is called pitching down. When the pilot pulls the control stick back, the elevators tilt upward, the tail goes down and the fuselage pitches nose-up. When the elevator tilts downward more lift is created (like ailerons) and the tail rises. When the elevator tilts upward, less lift is created and the tail descends.

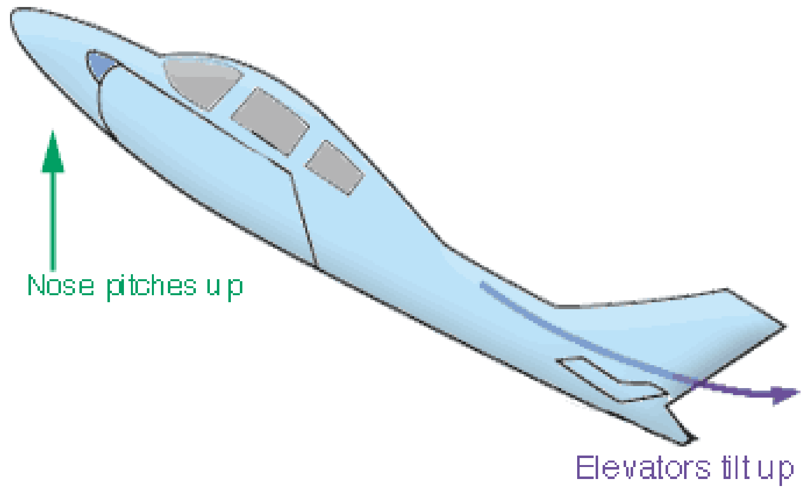


Figure 7: Elevator in the aircraft fixed wings

3.3. The rudder

The rudder (Figure 8) is located on the fin. The two rudder pedals are located at the pilot's feet. When the pilot pushes on the right rudder pedal, the rudder tilts to the right and the airplane yaws "nose-right." When the pilot pushes on the left rudder pedal, the rudder tilts to the left and the airplane yaws "nose-left." Again this is due to lift. However, the direction of this lift force is different than the lift force that causes the airplane to rise. When the rudder tilts to the right, more lift is created on the right, which "lifts" or pushes the vertical stabilizer to the left.

This, in turn, causes the airplane to yaw nose-right. The opposite motion occurs when the rudder tilts to the left.

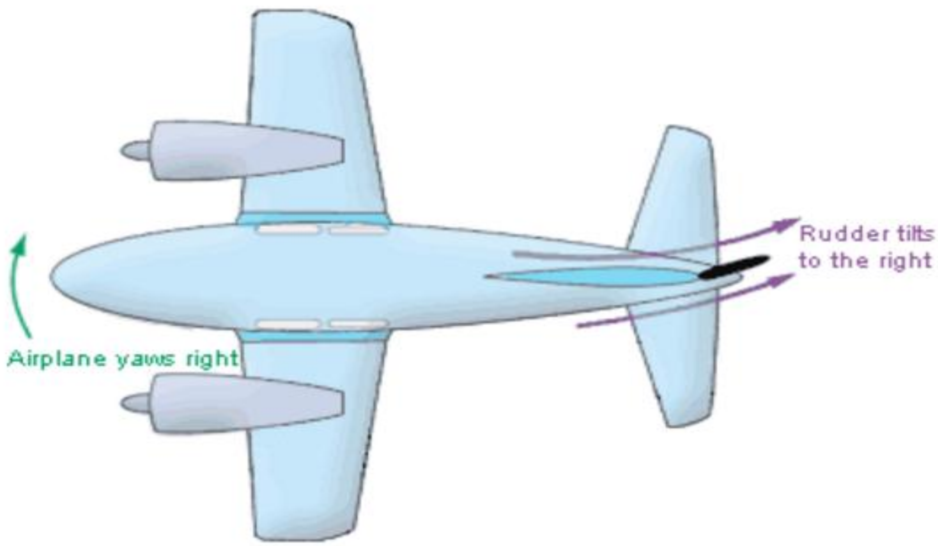


Figure 8: Rudder in the aircraft fixed wings

4. Principle of Flight Control

The four basic forces acting upon an aircraft during flight are lift, weight, drag and thrust as shown in Figure 9.

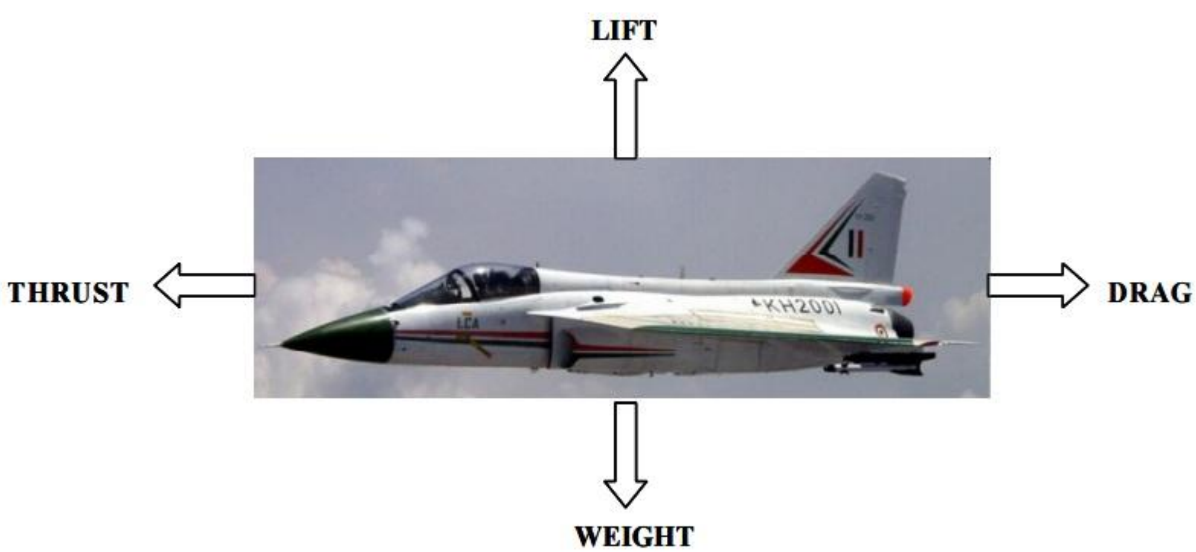


Figure 9: Forces acting in an aircraft

4.1. Lift

Lift is caused by the flow around the aircraft. Lift is the upward force created by the wings, which sustains the airplane in flight. The force required to lift the plane through a stream of air depends upon the wing profile. When the lift is greater than the weight then the plane raises.

4.2. Weight

Weight is the downward force created by the weight of the airplane and its load; it is directly proportional to lift. If the weight is greater than lift then the plane descends.

4.3. Drag

The resistance of the airplane to forward motion directly opposed to thrust. The drag of the air makes it hard for the plane to move quickly. Another name for drag is air resistance. It is created or caused by all the parts.

4.4. Thrust

The force exerted by the engine which pushes air backward with the object of causing a reaction, or thrust, of the airplane in the forward direction.

5. Unmanned Ariel Vehicles: UAV’s

5.1. Definition

The UAV is an acronym for Unmanned Aerial Vehicle, which is an aircraft with no pilot on board. UAV’s can be remote controlled aircraft (e.g. flown by a pilot at a ground control station we called them drones) or can fly autonomously based on pre-programmed flight plans or more complex dynamic automation systems. UAVs are currently used for a number of missions, including reconnaissance and attack role.

In few words, a UAV’s (figure 10) is a vehicle that is capable of travel without requiring a human to be onboard. They may be operated remotely by humans or may be capable of autonomous operation using programmable technologies.

Figure 10: Real UAV's system (NASA)

5.2. A brief History of UAV's

The first appear of the name UAV's was in the beginning of 20th century, and it's continue used nowadays.

The table below (Table 2) shows a history of UAV's.

Year	Historical event
1903's	Pre-aviation days the use of primitive UAV technology was used for combat and surveillance.
1910's During world war I	The U.S made test flight of UAV and recognized the potential of the combat before Armistice arrived
19030's by the mid-to-late- 1930's	The new emerging UAV's were used as a combat training tool to train pilots
1940's during world war II	German developed V-1 and V-2 rockets that were used to bomb England on a regular bases .There was no defense against this weapons, Until the U.S develop a targeting system that could track and shoot down the rocket.
1960's	The new UAV took on a new combat role during the Vietnam War as a stealth surveillance tool.
1970's	The success of the Fire Bee UAV continued to the end of Vietnam War .Other countries began to develop their own UAV and U.S began to develop more advance UAV.
During the late of 1970's throughout of 1980's	Israeli Air force was aggressive in designing and developing a new version of UAV's ,which were integrated into many UAV fleets around the world and including U.S.
1990's till nowadays ….	UAV's served a critical combat role of keeping pilots away from harm and allowing the UAV's to perform the same mission, they are also playing peaceful role of monitoring weather and Earth's environmental.

Table 2: History of UAV's

As we can see from the table above the UAV's are deeply began so earlier and keeping developing from U.S and other countries while others start to understand them and develop their own ones which is the case of Turkey and its military.

5.3. The principle of operation

The main vocation aerial drones is the observation and monitoring, Vocation until now mainly used for military purposes (currently 90%global market for drones). Thus, all the drones, whether independent (we are talking here about UAV's) or not (That's main a drone have its radio for command), require the presence on the ground of at least one operator to collect real-time the benefits of the mission: it receives, analyzes and stores information transmitted by the drone.

Today, progress, both in performance and their drone's equipment, they confer a very wide potential for use in the civilian field.

Therefore, UAV's system have two essential parts, the drone itself –Aircraft fixed wing in our case – and the control system.

As shown in Figure 11 the plane is controlled from the ground by pilots or programmers during the mission, and also might be just watched and analyzed by them in case the UAV works well.

The surveillance data is sent to satellite from which is received by the ground base.

Figure 11: Principal Working of UAV's

5.4. Advantages and Use of UAV's

UAVs come in many shapes and sizes. Each of these have their own unique pros and uses. It is these characteristics which ultimately leads to the operator's decision in which platform will best fit the application. It is understanding these key attributes and acting on them will ensure that your mapping mission is a success.

Besides, military uses of UAVs, humanity began to explore them in public safety as searching and rescuing people in disasters as floods or earthquakes, in addition, small UAVs used for law enforcement traffic monitoring and crowd control, finally UAVs become more popular nowadays in agriculture filed reducing time and increasing quality of job (figure 12).

Figure 12: Use of UAVs in Agriculture

5.5. Types of UAV's

The most logical classification of UAVs adopted for main criteria of size or performance (mainly the scope and operation altitude), thus we have so many types of UAVs around the world (Figure 13).

Starting by the miniature drones which are generally covers all the drone with a wingspan of less 50 centimeters, it could be reduced to only a few cm (Nano-drones in some cases).

Furthermore, drones short-range, they are used to "see the other side of the hill", or few kilometers. Also we find Tactical UAVs to medium range which can be equipped with low-speed (150 km / h) or faster (700 km / h), these drones represent the intermediate category, with different performances. With a takeoff weight, which remains below one tone, their range extends from 30 to 500 km, their flight altitude, from 200 to 5000 meters, and endurance, 2 to 8 hours.

Maritime tactical drones are also a type of UAV's have autonomy large enough (at least 5 hours), and be able to put down in strong winds on a narrow platform, partially surrounded by barriers and subjected to movements of large amplitude, roll and pitch in rough sea.

Besides, Rotary wing UAV's which is distinguished by its form of sustenance but covers sizes varied UAV performance, also, long-endurance drones with flight times of between 12 and 48 hours, they are in the category of "large" drones, whose size is mainly dictated by heavy payload and a high amount of fuel needed for the mission. This category is itself divided into two parts, depending on the flight altitude machines: as for airplanes, the more flying high, the faster we go the more we travels distance : Thus, there is the so-called drones "MALE" (Medium Altitude Long Endurance) and "HALE" (High Altitude Long Endurance).

Finally, Combat drones, or UCAV (Unmanned Combat Aerial Vehicle)Those are of course the attack-minded drones, whose payload includes arms - usually, missiles - to perform attack mission ,the ground or in the longer term air defense and air policing. They are true fighters unmanned but the remote management weapons greatly increases the technological complexity.

Figure 13: Types of Drones

6. Problems

The nation of Turkey has the biggest army after U.S in the world (In term of number), and to keep it strong enough to defeat any adventure must follow the new technology invented and used in the world as possible as they can .In this framework the institution of research where I had my internship was charge to learn and prototype a small UAV system for fixed wings aircraft for military usage, the controlling part was my task in this period of work.

The hard parts in the project is how to understand and modify the open source code used for UAV’s, besides, how to simulate all the work before start real experience and to control your flight from GCS (ground control Station) software existed and to adapt them to your own desire and missions launched.

7. Solutions

To do so, a deep study should be launch in order to understand what behind the scene. After a long research and study hold about UAV’s, we came to the solutions and taches below:

7.1. Conception and test of hardware

- Modeling and design a Small UAV.
- Build a small prototype of fixed wing.
- Launch a safe, stabilize and auto flight.

7.2. Coding, simulating, Testing and development.

- Understand and use open sources for Ground Control Station (GCS) already existed.
- Understand and simulate the open source code for Ardupilot Mega 2.5 used for UAV’s.
- Build Simulink Blocks for autopilot for numerical simulation.

8. Conclusion

This chapter has been reorganized to present the theory of aerodynamic as well the principle of flight and drones in a more practical manner.

Besides, this chapter has allowed as to discuss problems and solutions of our project, the following chapter will cover several solution had chosen and materials used.

Chapter 3: Implementing the solutions

1. Introduction

Chapter 3 introduces solutions had chosen starting by the hardware and chose of material, used to build the prototype and the characteristics of each one, furthermore the introduction of software used for Simulation, conception ,supervision and control from GCS.

Due to accuracy and precision demanded in this area of work, controlling and supervision are a must, using them to analyze data in order to improve the quality of missions launch and avoid errors made, to achieve this we fixed the following objectives:

- Chose good hardware for prototyping respecting the budget and the needs of the customer.
- Collect and wire the hardware.
- Chose appropriate software for conception and design a Small UAV and supervisions from base ground control.
- Simulate several flights using appropriate software.

2. Prototype conception

Prototypes are fully functioning parts and they are used for low volume applications. Usually for single pieces and replacement parts, prototyping is a cost effective way to create short run tooling or even disposable tools. This includes:

2.1. Mechanical hardware

In order to be able to flight and test the materials used for UAVs system, we should pick up a small flight model form the stores respecting many characteristics such as weight, size, volume and other characteristics.

After checking up a huge catalog existed, we choose Volantex RC Ranger EX 753 as shown in Figure 14 model which has the following characteristics (Table 3):

Characteristic	Specification
Wingspan	1980mm (77.9 in)
Overall Length	1170mm (46 in)
Flying Weight(without battery)	1500g
Prop Size	1060 Propeller
Motor Size	3715 powerful out runner brushless motor
Speed Control	Easy-Plug 40A Switch-mode BEC brushless ESC
Recommended Battery	11.1V/14.8V 3300mAh ~ 10000mAh LiPo
Flaps	Yes
Retracts	No
Recommended Environment	Outdoor
Assembly Time	Less than half hour
Minimum Age Recommendation	14 years

Table 3: Aircraft's Specification

Figure 14: Volantex RC Ranger EX 753 model

Volantex RC Ranger EX 753 module Price

This module is available in the Get Hop Shop in UK and we bought it) for 249.99 €.

2.2 Electronic Hardware

To control and run the code, an electronic part is crucial for that, hopefully a package (figure 15) recently designed for UAVs systems is developed and still going on to achieve goals .

Figure 15: APM 2.5 package

Several component made this package are:

2.2.1 Ardupilot Mega 2.5 cart

The flight controller is the brain of your drone. The flight controller reads all of the sensor data and calculates the best commands to send to your drone in order for it to fly, in our project APM plays this role

Ardupilot Mega (APM) is a professional quality IMU (Inertial Measurement Unity) autopilot that is based on the Arduino Mega platform. This autopilot can control fixed-wing aircraft (our need in this project), multi-rotor helicopters, as well as traditional helicopters. It is a full autopilot capable for autonomous stabilization, way-point based navigation and two way telemetry with Xbee wireless modules. Supporting 8 RC channels with 4 serial ports. ArduPilot Mega consists of the main processor board (red one above) and the IMU (Combination of Accelerometer sensor and Gyroscope 6 degree of freedom sensor) shield which fits above or below it (shown mounted together below).

Figure 16: Ardupilot Mega 2.5

APM Features

- Free open source autopilot firmware that supports planes (which is our project), multi-copters (tri, quad, hex, oct, etc.), traditional helicopters and ground rovers.
- Simple setup process and firmware loading via a point-and-click utility. No programming required, But if you do want to fiddle with the code, you can with the easiest embedded programming toolkit available: Arduino, which is the next step in our project, modify the code following our needs.
- Full mission scripting with point-and-click desktop utilities can support hundreds of 3D waypoints
- Two-way telemetry and in-flight command using the powerful MAVLink protocol.
- Choice of free Ground Stations, including the state-of-the-art HK GCS, which includes mission planning, in-air parameter setting, on-board video display, voice synthesis, and full data logging with replay.
- Autonomous takeoff, landing and special action commands such as video and camera controls
- Supports full "hardware-in-the-loop" simulation with XPlane and Flight Gear.

- 4MB (Mega Byte) of onboard data-logging memory. Missions are automatically data logged and can be exported to KML (Google earth is a good example for this, we will cover it in the next chapters).
- Built-in hardware failsafe processor, can return-to-launch on radio loss.

Accelerometer and Gyroscope and Barometer

These are the inertial sensors on our APM cart. The accelerometer measures acceleration forces, and the gyro measures rotational forces. By combining these measurements the flight controller is able to calculate the drone current attitude (angle it is flying at) and perform necessary corrections.

A barometer is a pressure sensor that you use to measure the aircrafts altitude. These pressure sensors are so sensitive that they can detect the change in air pressure when your plane moves a few centimeters.

APM price

This cart is available in the Unmanned Tech Shop in UK and we bought it (airspeed sensor and APM power module are not included in the price) for 48.00€.

2.2.2 Telemetry

Telemetry is the automatic measurement and wireless transmission of data from remote sources which is GCS (Ground Control Station) in our case. In our project we choose to use it for send and receive information from the plane when it’s in the sky, for example to change the mission.

Our telemetry used the Xbee wireless module to communicate RX and TX each other with a frequency of 433MHz used legally in Europe and in Turkey as well and a baud of 57600.

We can use the MAVLink support in the telemetry to monitor the link quality while flying, because our ground station supports it.

Note that for only the United State the communication frequency is 915 MHz

On APM 2.5(inside the aircraft)

We use the dedicated telemetry port and supplied cable, as shown in figure 17:

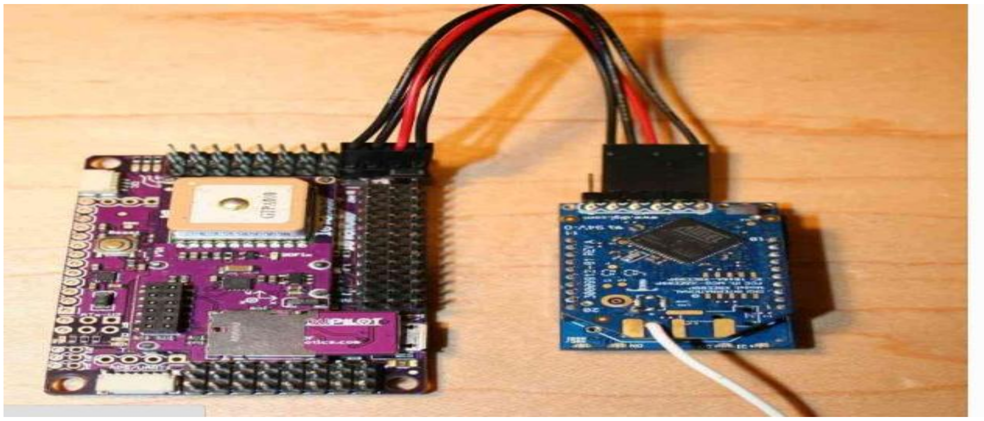


Figure 17: Telemetry in the plane connected with APM

On the ground station

If you're using a USB adapter, simply connect it via a USB cable as show

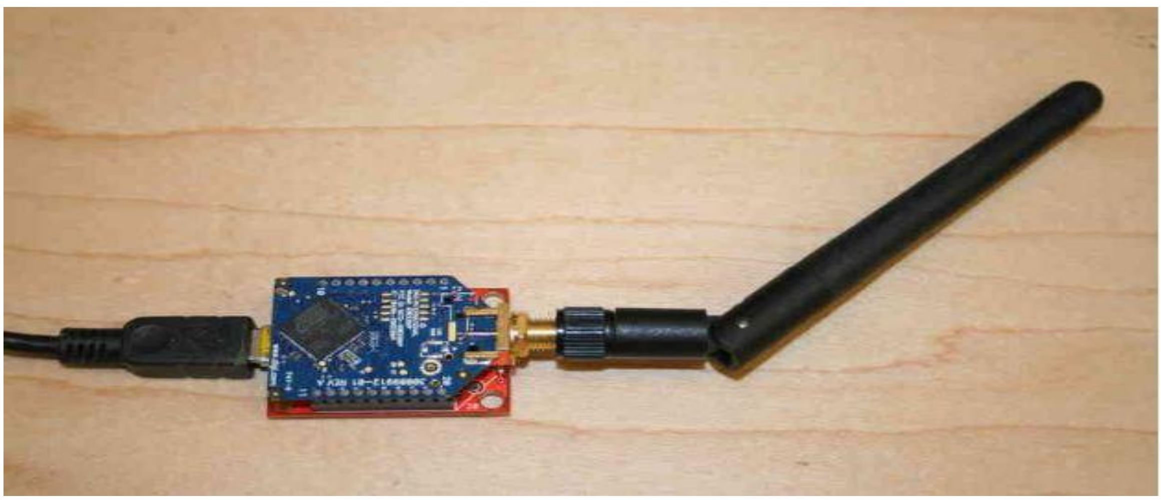


Figure 18: Telemetry connected to the computer (BGC) by USB port

Telemetry's price

This device is available in the Unmanned Tech Shop in UK and we bought it for 59.87€.

2.2.3 GPS model

Whether the vehicles are guided autonomously, or guided by ground based pilots (GCS), GPS (Global Positon Satellite) in UAV plays an important role. As long as sufficient satellites signals (4 satellites) can be accessed during the entire UAV mission, GPS navigation techniques can offer consistent accuracy. Often, GPS is used in conjunction with Inertial Navigation Systems (INS), to offer more comprehensive UAV navigation solutions.

The most common use of GPS in UAV is navigation. A central component of most navigation systems on a UAV, GPS is used to determine the position of the vehicle. The relative positioning and speed of the vehicle are also usually determined by the UAV GPS. The position provided by the receiver can be used to track the UAV, or, in combination with an automated guidance system which is APM cart in current project, steer the UAV.

Autonomous UAV usually rely on a GPS position signal which, combined with inertial measurement unit (IMU) data, provides highly precise information that can be implemented for control purposes. In order to avoid accidents in an area heavily populated by other UAV or manned vehicles, it is necessary to know exactly where the UAV is located at all times. Equipped with GPS, a UAV can not only provide location and altitude information, but necessary vertical and horizontal protection levels.

For better use in our prototype we choose a module who combine a GPS and compass in one device as shown in Figure 19 below. (More information about the principle in Appendices.)

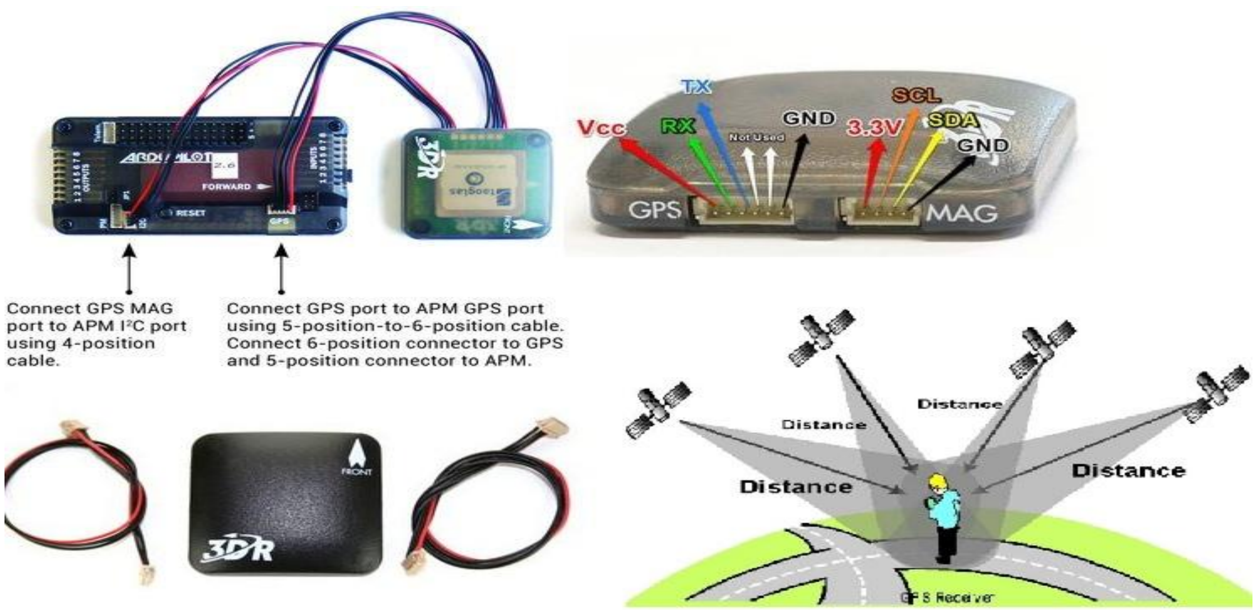


Figure 19: GPS module 3DR +Compass inside

<u>GPS price</u>

This device is available in the 3D Robotic Shop in UK and we bought it for 81.15€.

2.2.4 Airspeed sensor: Pitot tube

This is only used on fixed wing plane. An airspeed sensor (figure 20) is another form of a pressure sensor, but instead of measuring altitude, this measures for fast the air is passing the aircraft. It does this by comparing the dynamic and static pressure via a pitot tube. Airspeed is important for fixed wing aircraft as the airflow over the wings is what generates lift, and if you are flying too slowly you could end up stalling your plane and crashing.

Figure 20: unmanned speed sensor kit - MPXV7002DP

MPXV 7002DP's price

This device is available from the Unmanned Tec shop in UK and we bought it for 25.14€.

2.3 Electrical Hardware

2.3.1 Battery

A battery powers all of the electronics, and the motor on your plane, The difference with this battery we are going to use and the ones you would use in your TV remote is the chemicals used in the battery. Because those batteries are based on Lithium Polymer chemistry (hence the name LiPo) which allow these batteries to have a very high energy density compared to other types of batteries. A battery with a higher energy density will be able to hold more energy compared to another battery of the same weight which is why LiPo batteries are commonly used with RC aircraft and drones.

There are other types of batteries such as nickel cadmium (NiCad) or nickel–metal hydride (NiMH), however these are seen as older battery technology and are not really used anymore because they are very heavy and don't hold much energy compared to LiPo's.

For the reasons above we choose to work with a LiPo battery in our project due to the advantages gives us.

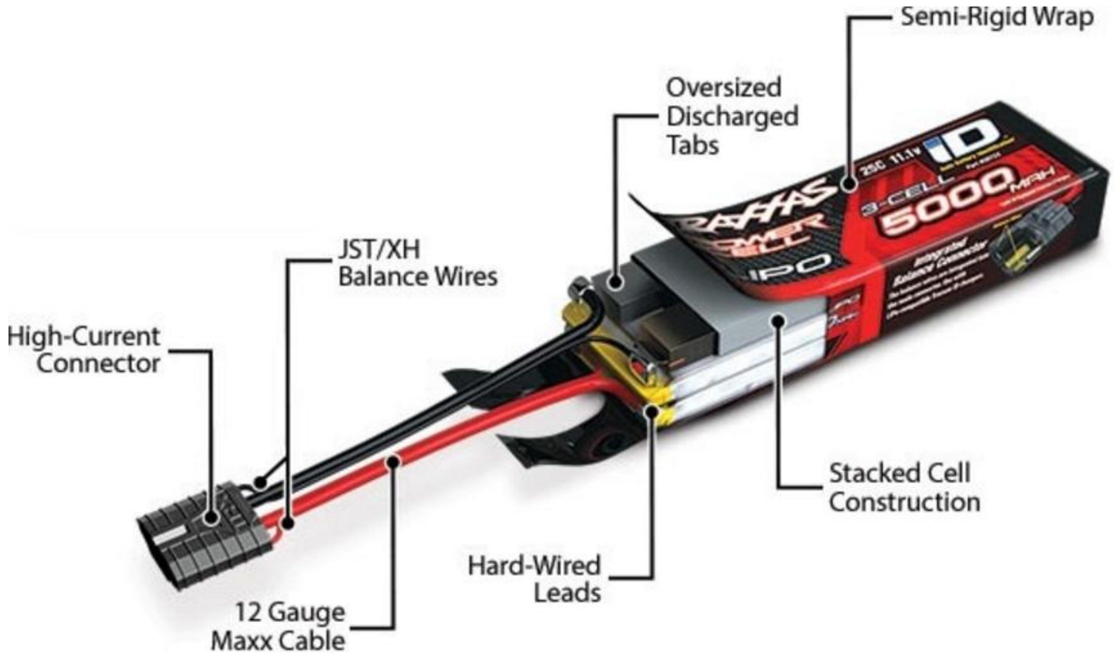


Figure 21: LiPo battery

In order to get the best flight time and performance it important to know how to choose the best Lipo battery.

Specification

Note: More details about this type are discussed in Appendices.

Net Weight	Brand	Capacity	Discharge Rate	Max continuous current	Max burst current	Parallel	Length	Width
120g	Tattu	1300mAh	45C	58.5A	117 A	1P	71mm	32mm
Wire gauge	Connector Type	Thickness	Voltage	Max burst discharge rate	Balance Connector Type			
No	XT60	22mm	3S(11.1V)	90C	JST-XH			

Table 4: technical Specification of LiPo battery

TATTU 1300MAH 11.1V 45C 3S1P LIPO Battery's price

This device is available in the Unmanned Tec shop in UK and we bought it for 16.15€.

2.3.2 Power Module

Because your autopilot is a sensitive electronic device, it is important that it receives a clean power supply. A power module (figure 22) is used to convert the battery voltage from your drone battery down to a low voltage that your autopilot uses (often 5v).

However the other benefit of using a power module is that it gives you the ability to measure your battery voltage and capacity. This is useful because if your flight controller can measure the battery it knows when your battery is running low so it can warn you to land. Other autopilot systems also have some failsafe functions build in which will automatically take over and bring your drone back home when the battery level gets too low.

High voltage APM power module with 3A UBEC's price

This device is available in the Unmanned Tec shop in UK and we bought it for 23.94€.

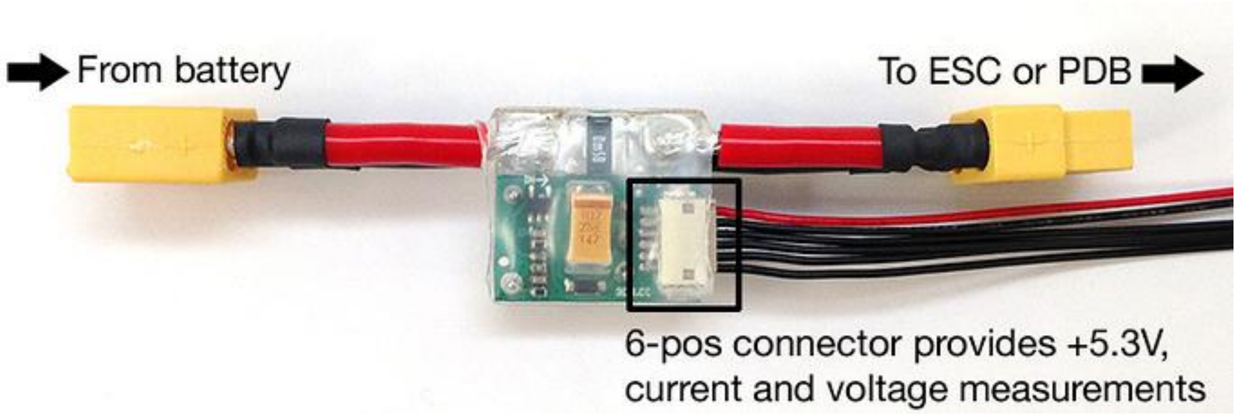


Figure 22: Power Module

2.3.3 Electronic Speed Control (ESC)

An electronic speed controller (or ESC) controls how fast your airplane's motor spins. Is a little device that control the speed of your brushless motor? And it's an interface between the airplane's radio receiver and the power plant.

An ESC will have three sets of wires, one lead will plug into your airplane's main battery. The second lead will have a standard servo wire that plugs into the throttle channel of your receiver. And finally, the third set of wires actually power the motor.

The first thing to consider when buying your ESCs is what size of ESC you need for your setup. Of course, when we talk about 'size' of ESC, we do not mean the physical size of the module, but of the amount of amperage it supplies to your motor. The standard range of ESC sizes is 12A-40A ESCs.

To choose the right ESC for our needs, we also have to consider what motor we are going to use, and what propellers. The choice for these three essential pieces of equipment are entirely linked to one another .Thus in order to make the right decision we looked to table of the throttle in Appendix 2.

After analyze our needs we choose the ESC with the specifications following for control our motor’s speed.

- Size: 20A ESC for the Vortex 250 Pro.
- A high-end 32-bit processor using the latest FET and driver technology.
- Wide: only 9.4mm.

<u>ESC price</u>

This device is available in the Unmanned Tec shop in UK and we bought it for 19.70€.

Figure 23: ESC for Brushless motor

2.3.4 Brushless motor

The motor of your drone is what your propellers are connected to, which cause them to spin around and generate thrust to enable your drone to fly.

In our case - fixed wings aircraft-, the motor generate a forward thrust to push the plane forwards through the air.

In order to choose the right motor for our plane, many factors must take into consideration such as global weight of the plane including all the components.

In general the following points should be knowing to choose your motor:

1. The ready-to-fly weight of your model.
2. The model's wing span.
3. The model's wing area. This is usually in the manual, or on the plans.
4. What sort of flying you intend to do with the model (scale, aerobatics, 3D)
5. What sort of motor will you use? (Out runner, in runner, brushed, etc).
6. What sort of battery will you use?

Using the open source software@Calc (figure 24) to estimate our calculation for the appropriate motor (more details in Appendix 3) we choose

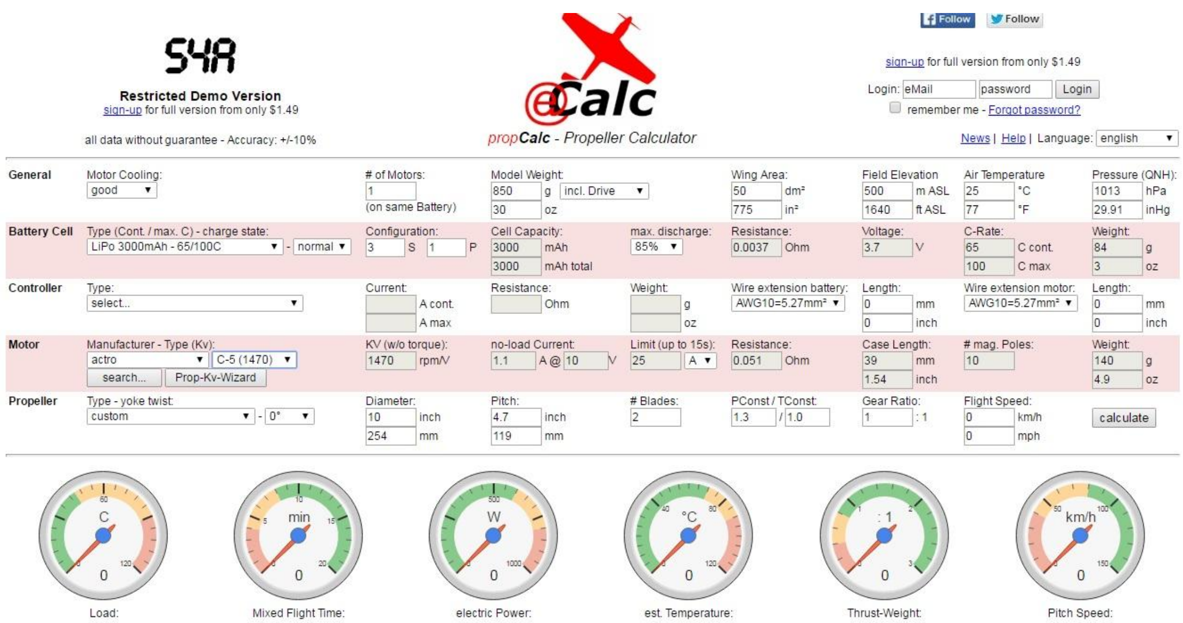


Figure 24: Estimation for brushless motor

Specifications

KV	Stator Diameter	Stator Length	Shaft Diameter	Configuration	Motor Dimension	Weight	N of Cells	Internal resistance
2100	23mm	6mm	3mm	NP	28.5x21.5mm	35.8g	3-4S	0.05Ω
Max Cnt Current	Max Cnt Power							
27.4A	405.5W							

Table 5: Motor's MR2306 specifications

<u>MR2306 2100 KV Motor's price</u>

This device is available in the Unmanned Tec shop in UK and we bought it for 14.95€.

2.3.5 Servos

In simple words, as we already knew, Remote Control (RC) airplane servos, like the one shown in Figure 25, reliably move to a desired angular position, given the appropriate input.

Choosing the right RC servos for our airplane is much easier than you might think. Servos are literally the "muscles" that control every moving part of our airplanes.

We want a servo than can handle the torque required to move our control surfaces quickly and precisely. A servo's characteristics are defined by Torque, Speed, Dimensions and Weight. (More details shown in Appendix 4).

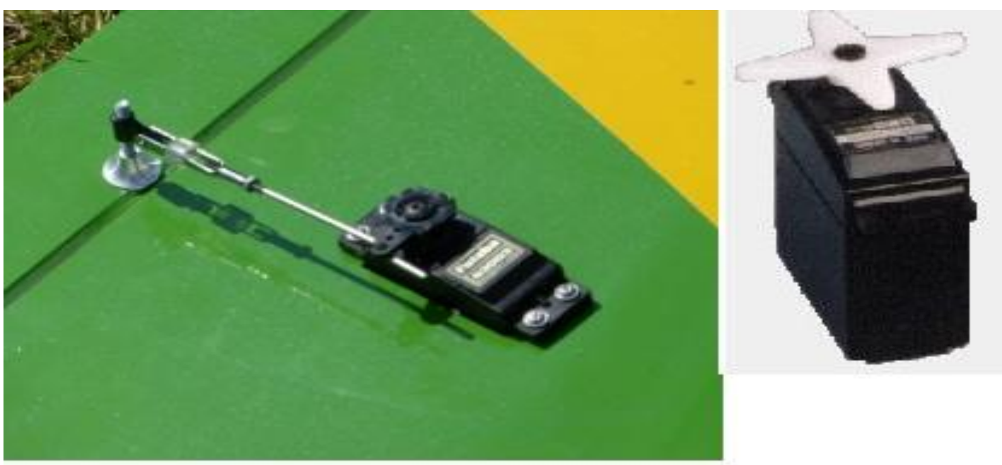


Figure 25: controlling Ailerons by a Servo

The appropriate servos we need in our control surfaces had the specifications shown in the table 6 below.

Specifications

Weight	Dimensions (mm)	Lead length	Mounting hole spacing	Mounting depth	Operating voltage	Stall Torque	Operating Speed
2.5g	L18.02 x W7.91 x H16.8	135 mm	21.9mm	10.15mm	4.0V~6.0V	0.27kg/cm	0.08sec/60deg

Table 6:EMAX ES9251 2.5G DIGITAL SERVO specifications

EMAXES9251 2.5G digital Servo's price

This device is available in the Unmanned Tec shop in UK and we bought it for 4€.

In our plane we use five servos for all control surfaces (Ailerons, Ruder, and Elevators), thus the price is 4*5=20€.

3. Hardware Wiring

After chosen all the components we need in this project and bought them from the stores, the next step is to collect this materials and wire them in a right way to avoid any damage, to do so, reading the data sheet for each component is a must.

After a deep analyze all the communications type should be use such as I^{2}C, PWM….Those communication protocols will be discussed in the end of this chapter.

We came to the wiring shown in the figures 26, 27, 28.

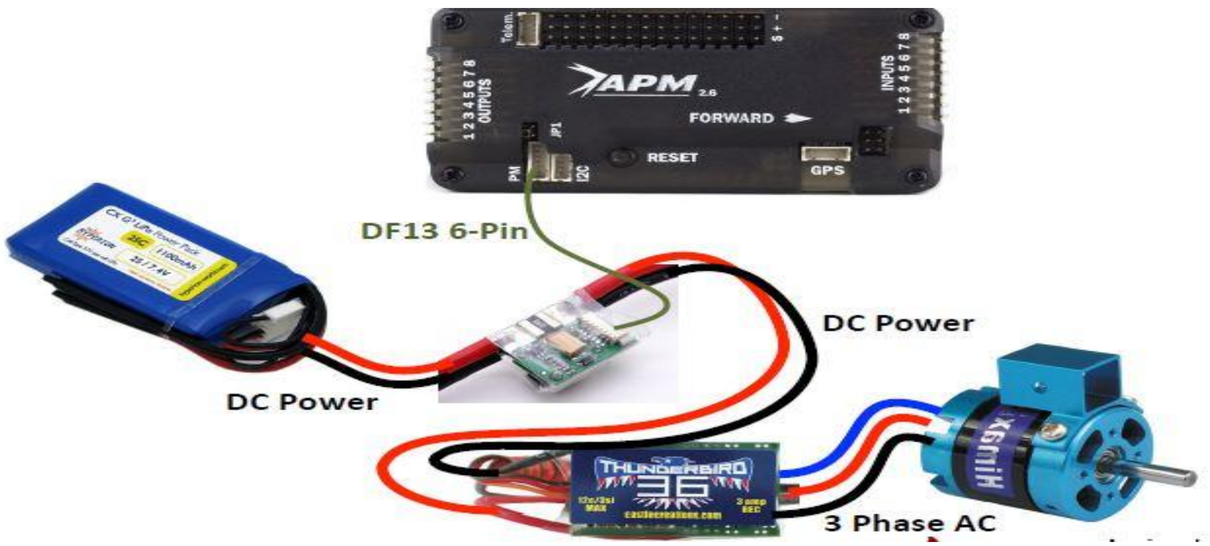


Figure 26: ESC, Motor, Battery and Power Module Wiring Method

Figure 26 shows the wiring between battery, power module, ESC and Brushless motor used for generate the throttle.

The Power Module connected to APM via PWM technic explained in the paragraph below.

DC power comes AC trough ECS in order to turn the motor.

Order of Motor phases is not a matter, that’s mean we can inverse them (two of them) to find the right direction of motor’s spin.

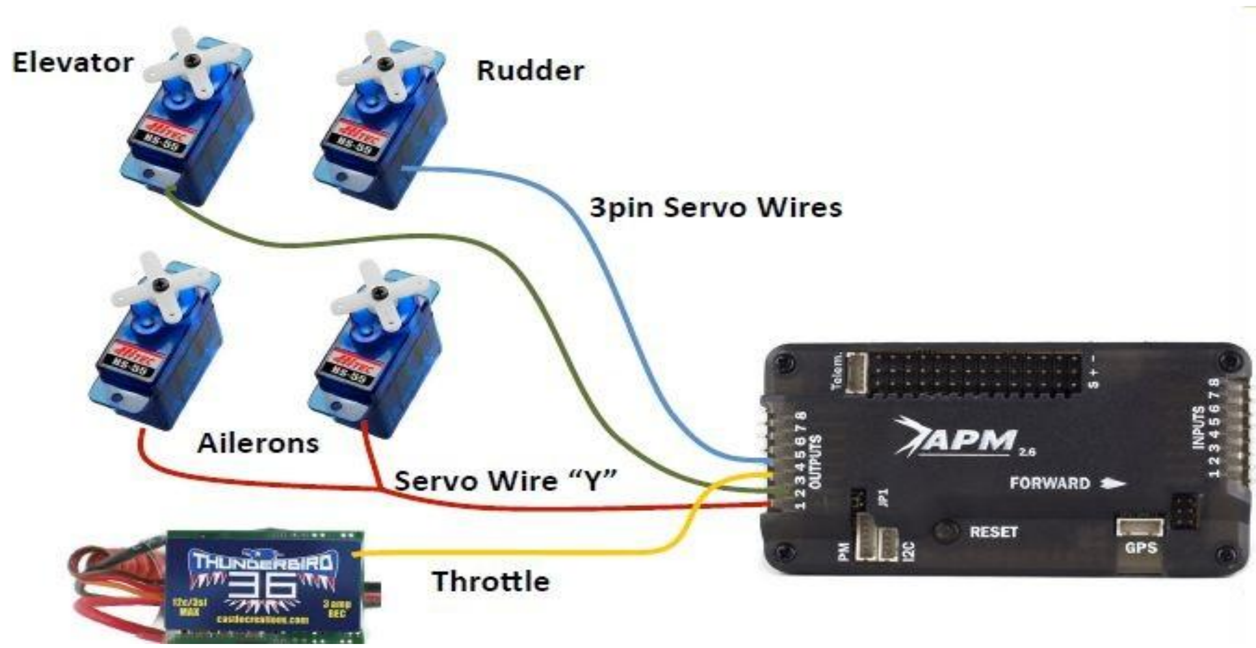


Figure 27: Servos Wiring to APM2.5

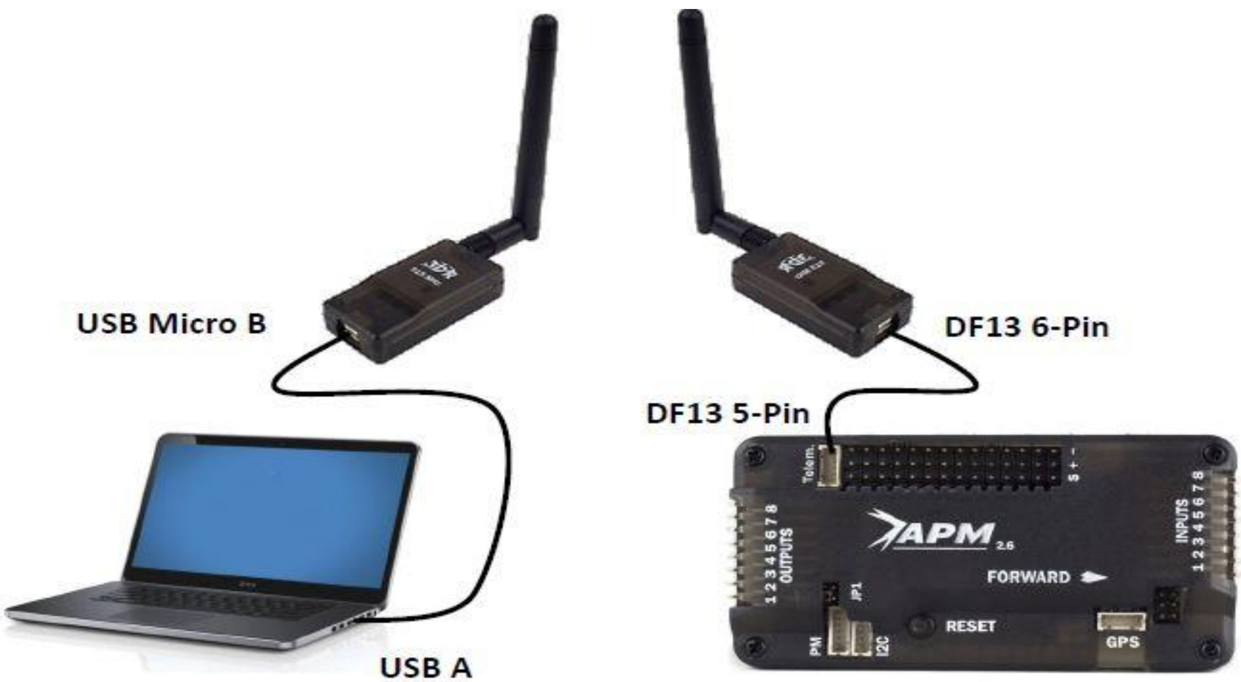


Figure 28: Telemetry connection to APM and Computer

4. Supervision and Controlling (GCS).

4.1.Ground Control Station

Although the autopilot is responsible for controlling the plane, the ground station is the software that gives the commands. Every flight path is generated by the ground control station and transmitted as a reference trajectory for the autopilot to follow. It is the software that allow users to interact with the aircraft, specify the fight path and track its movement.

A ground control station is a standard tool for controlling UAVs. However, they are often tailored for certain autopilots. For the APM autopilot Mega cart, the Mission Planner (MP) station is the compatible control stations.

Mission Planner station is specialized for small UAVs that uses the MAVlink (Micro Air Vehicle Link) protocol, for communicating with the autopilot. Other popular ground stations used for the same application are QGroundControl, APM Planner and Droid Planner.

We choose Mission Planner as CGS for our project due to its compatibility total with Windows 7.

The main features of Mission Planner are presented below:

- Open-source MAVLink Micro Air Vehicle Communication Protocol
- 2/3D aerial maps with drag-and-drop waypoints
- In-fight manipulation of waypoints and onboard parameters
- Real-time plotting of sensor and telemetry data
- Logging and plotting of sensor logs
- Support for UDP,TCP and serial (radio modem)

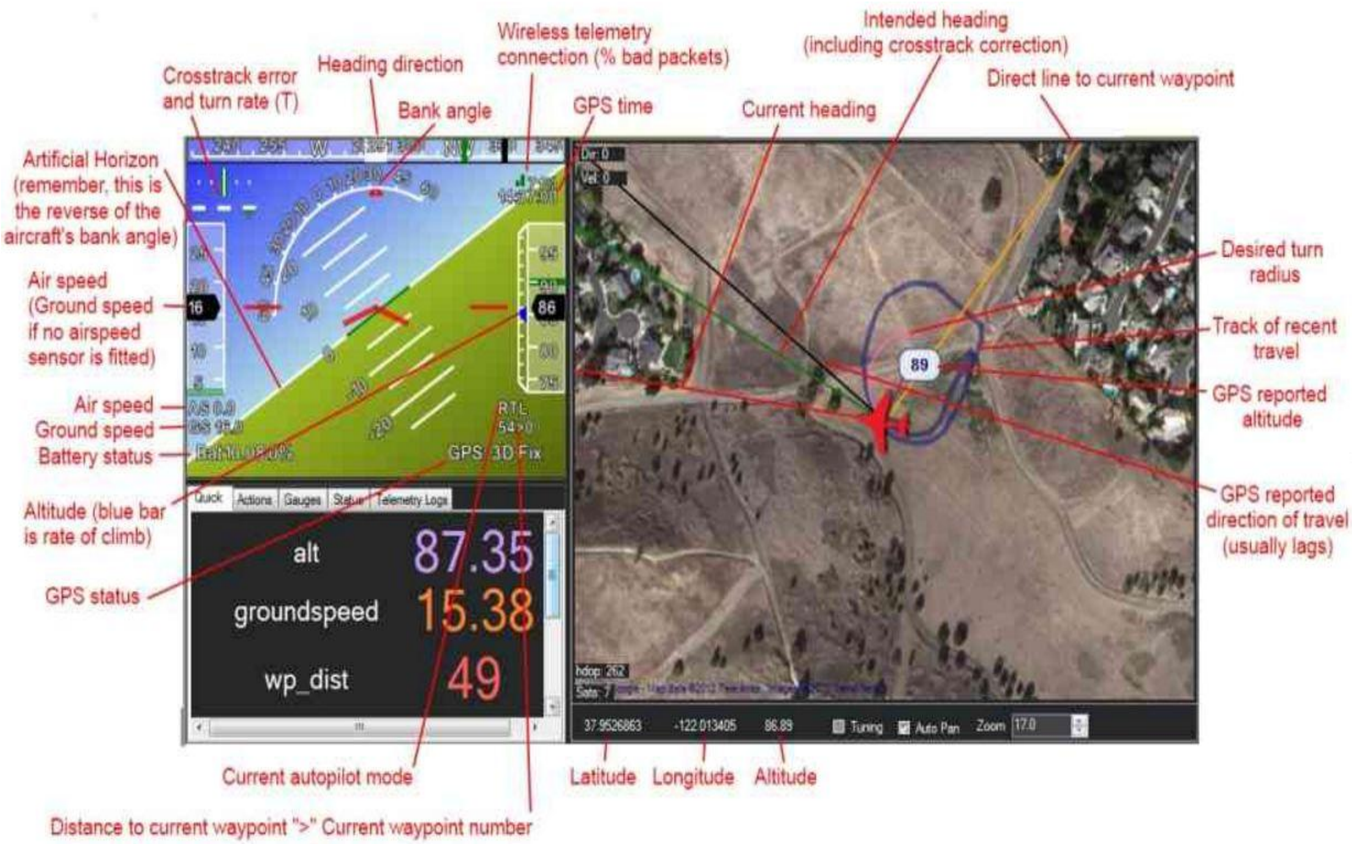


Figure 29: Graphical User Interface for MP

In Figure 29 the Graphical User Interface (GUI) for Mission Planner is displayed. The ground station provides information about location, altitude, attitude, warnings and other relevant data about the plane. The main features of this program is the ability to generate fights paths by setting way-points. After a way-point is set, the shortest route is uploaded to the autopilot and an autonomous mission can be initiated. It is the tuning of the autopilot that determines the aircraft's ability to follow the reference trajectory. Besides being used as a tool for generating fight paths, the control station is also used for sensor calibration and controller tuning.

4.2.X-Plane Simulator

X-Plane is the world most comprehensive and powerful flight simulator for personal computers, and it offers the most realistic flight model available.

X-Plane is not a game, but an engineering tool that can be used to predict the flying qualities of fixed and rotary-wing aircraft with incredible accuracy. Because X-Plane predicts the performance and handling of almost any aircraft, it is a great tool for pilots to keep up their currency in a simulator that flies like the real plane, for engineers to predict how a new airplane will fly, and for aviation enthusiasts to explore the world of aircraft flight dynamics.

X-Plane gave us the chance to simulate our module using simulation software in the loop (SITL) and simulation hardware in the loop (HIL) and analyze the behavior of our craft.

Figure 30: X-Plane simulator

4.3.Plane Maker Software

In order to flight and simulate our own model to get the exact approximation of parameters used to tune PIDs values and other parameters, we are going to use Plane Maker to build a near model of Volantex RC Ranger EX 753 used in this project.

Plane Maker is a program bundled with X-Plane that lets users design their own aircraft. Using this software, nearly any aircraft imaginable can be built. Once all the physical specifications of the airplane have been entered (e.g., weight, wing span, control deflections, engine power, airfoil sections, etc.), the X-Plane simulator will predict how that plane will fly in the real world; it will model the aircraft's performance just like it does for X-Plane's built-in aircraft.

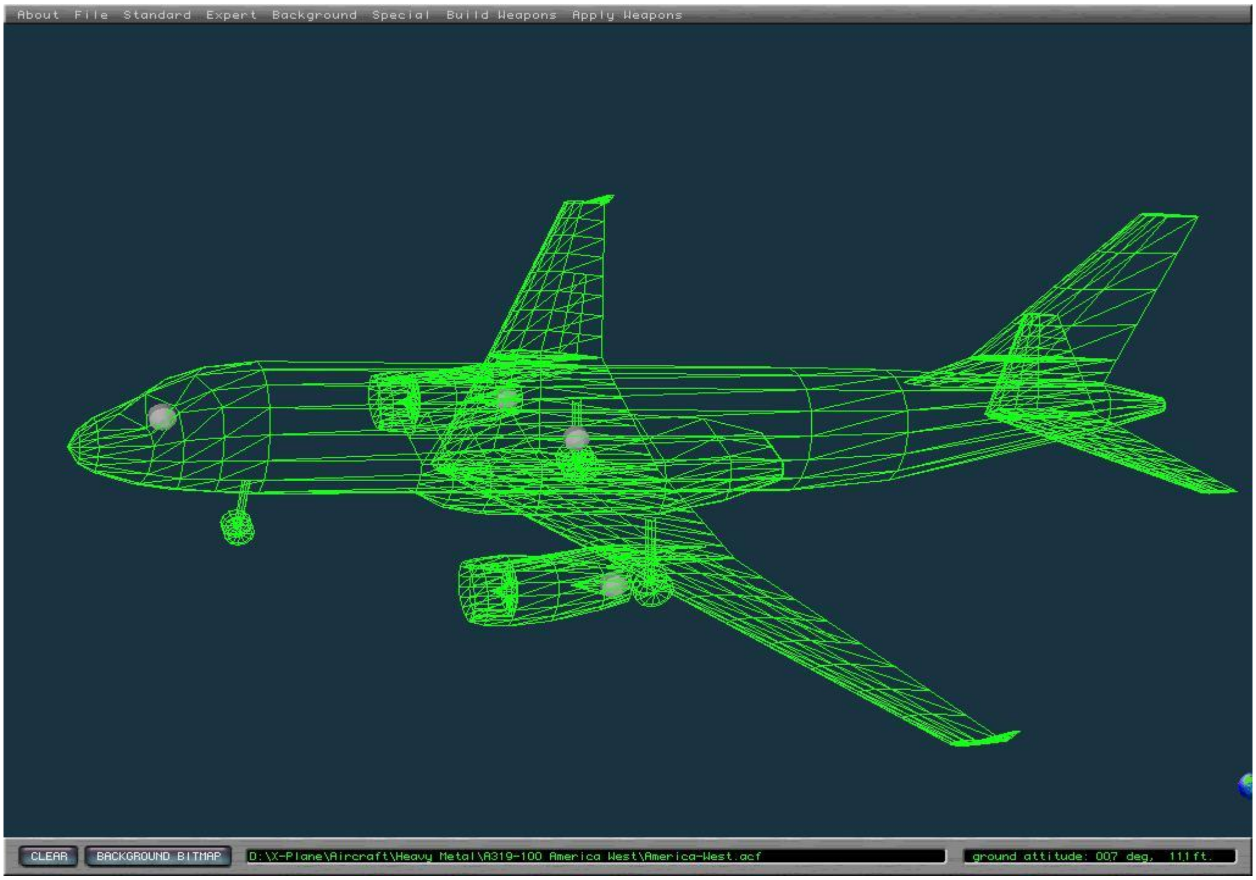


Figure 31: Plane Maker Software

5. Communication Protocols and technics:

Although to communicate APM and the components in one side and the APM and the software used for simulation in the other side , and to communicate the GCS and X-Plane several communication protocols and technics are used ,in the following lines we will explain briefly same of this protocols

5.1. I^2C protocol

The Inter-integrated Circuit (I2C) Protocol is a protocol intended to allow multiple “slave” digital integrated circuits which are sensors, IMU in our case, to communicate with one or more “master” chips which is APM Mega in this project. Like the Serial Peripheral Interface (SPI), it is only intended for short distance communications within a single device. Like Asynchronous Serial Interfaces (such as RS-232 or UARTs), it only requires two signal wires to exchange information.

.

.

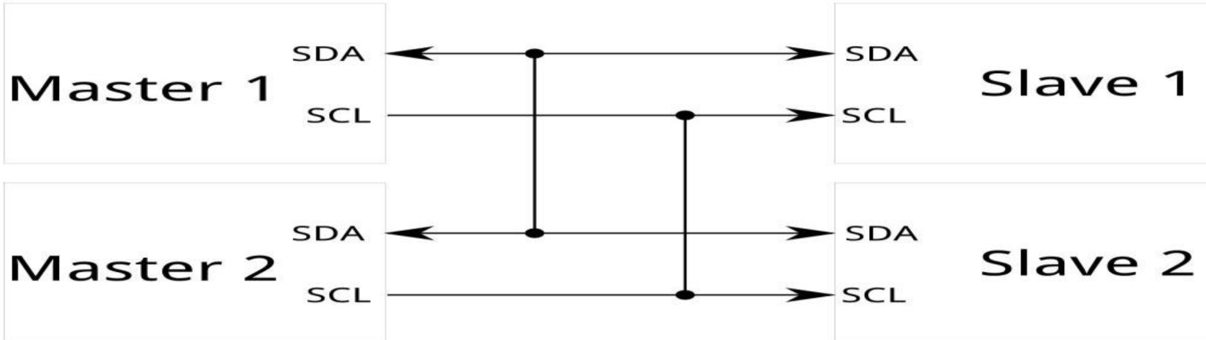


Figure 32: I2C protocol Architecture

I^2C requires a mere two wires, like asynchronous serial, but those two wires can support up to 1008 slave devices. Also, unlike SPI, I2C can support a multi-master system, allowing more than one master to communicate with all devices on the bus, although the master devices can't talk to each other over the bus and must take turns using the bus lines.

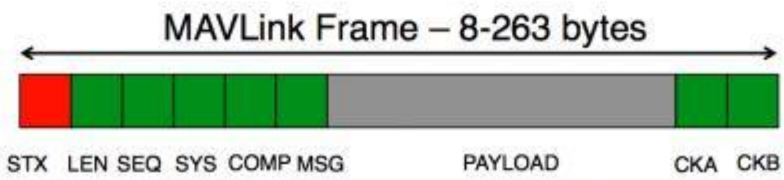
Data rates fall between asynchronous serial and SPI, most I^2C devices can communicate at 100 kHz or 400 kHz. There is some overhead with I^2C, for every 8 bits of data to be sent, one extra bit of Meta data (the "ACK/NACK" bit, which is discussed in Appendix 4) must be transmitted.

The hardware required to implement I^2C is more complex than SPI, but less than asynchronous serial. It can be fairly trivially implemented in software.

5.2.MavLink protocol

MAVLink or Micro Air Vehicle Link is a protocol for communicating with small unmanned vehicle. It is designed as a header-only message marshaling library. MAVLink was first released early 2009 by Lorenz Meier under LGPLlicense.

Micro Air Vehicle Link (MAVLink) is the header only, message-marshalling library used as the communication protocol between the ground station and plane. The main components of a MAVLink message are the header, system ID, message ID, and payload (main message). The header is used to classify the message as a MAVLink packet. The system ID identifies the system sending the message while the message ID identifies the type of message being sent. For example, the most common message to send is the heartbeat (ID = 0) which is constantly sent to ensure that the plane and ground station are properly connected and communicating. The payload of the message is the content inside it. The payload can contain fields such as the vehicle type, flight mode, positioning data, or commands to execute. These messages are sent as data packets between the ground station and plane to properly fly the plane.



Byte Index	Content	Value	Explanation
0	Packet start sign	v1.0: 0xFE (v0.9: 0x55)	Indicates the start of a new packet.
1	Payload length	0 - 255	Indicates length of the following payload.
2	Packet sequence	0 - 255	Each component counts up his send sequence. Allows to detect packet loss
3	System ID	1 - 255	ID of the SENDING system. Allows to differentiate different MAVs on the same network.
4	Component ID	0 - 255	ID of the SENDING component. Allows to differentiate different components of the same system, e.g. the IMU and the autopilot.
5	Message ID	0 - 255	ID of the message - the id defines what the payload "means" and how it should be correctly decoded.
6 to (n+6)	Data	(0 - 255) bytes	Data of the message, depends on the message id.
(n+7) to (n+8)	Checksum (low byte, high byte)	ITU X.25/SAE AS-4 hash, **excluding packet start sign, so bytes 1..(n+6)** Note: The checksum also includes MAVLINK_CRC_EXTRA (Number computed from message fields. Protects the packet from decoding a different version of the same packet but with different variables).	

Table 7: MavLink segment structure

There are so many important messages send by MavLink protocol, the most important is MAVLINK_MSG_ID_HEARTBEAT described below, and the other messages you can look for them in appendix 4 for better understanding.

MAVLINK_MSG_ID_HEARTBEAT

This is the most important message. The GCS keeps sending a message to APM or PX4 to find out whether it is connected to it (every 1 second). This is to make sure the MP is in synchronization with APM when you update some parameters. If a number of heartbeats are missed, a failsafe is triggered and plane lands, continues the mission or Returns to launch (also called, RTL). The failsafe option can be enabled/disabled in MP under Configure/Setup Failsafe options. But you can't stop heartbeats.

5.3.UDP protocol

In simple words, User Datagram Protocol (UDP) is part of the Internet Protocol suite used by programs running on different computers on a network. UDP is used to send short messages called datagrams but overall, it is an unreliable, connectionless protocol. We used this protocol to connect two computers, MP was running in one whereas the X-plane run in the second, In fact it's used to simplify the simulation in hardware in the loop using separate computers.

UDP is widely used in video conferencing and real-time computer games. The protocol permits individual packets to be dropped and UDP packets to be received in a different order than that in which they were sent, allowing for better performance. UDP network traffic is organized in the form of datagrams, which comprise one message units. The first eight bytes of a datagram contain header information (figure 30), while the remaining bytes contain message data. A UDP datagram header contains four fields of two bytes each:

- Source port number
- Destination port number
- Datagram size
- Checksum

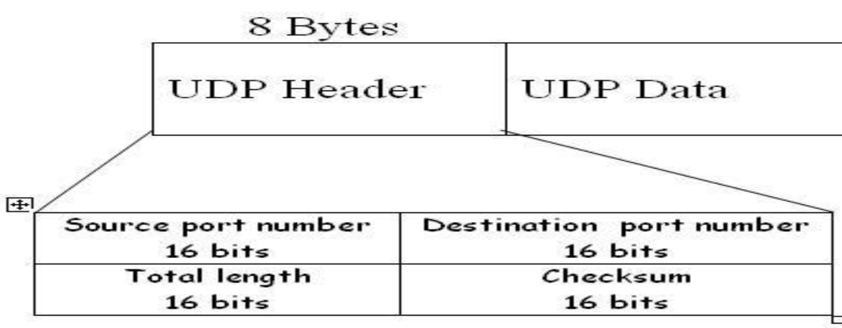


Figure 33: UDP segment structure

5.4.TCP protocol

Transmission Control Protocol, and pronounced as separate letters. TCP is one of the main protocols in TCP/IPnetworks. Whereas the IP protocol deals only with packets, TCP enables two hosts to establish a connection and exchange streams of data. TCP guarantees delivery of data and also guarantees that packets will be delivered in the same order in which they were sent.

We used this protocol to run X-plane and Mission planer in purpose to simulate software in the loop (SITL), via IP.

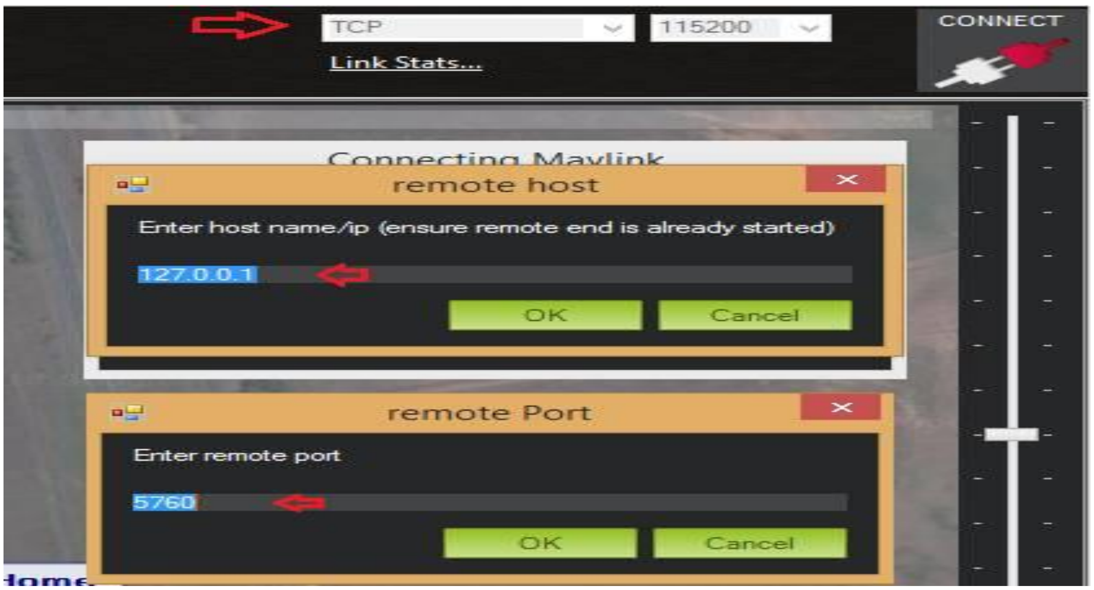


Figure 34: TCP protocol in MP

5.1. PWM technique

Pulse Width Modulation or PWM is a technique for supplying electrical power to a load that has a relatively slow response. The supply signal consists of a train of voltages pulses such that the width of individual pulses controls the effective voltage level to the load. Both AC and DC signals can be simulated with PWM.

One of the advantages of PWM is that the signal remains digital all the way from the processor to the controlled system, no digital-to-analog conversion is necessary. By keeping the signal digital, noise effects are minimized. Noise can only affect a digital signal if it is strong enough to change a logical-1 to a logical-0, or vice versa. We used this to connect our power module to The APM cart.

6. Global budget of the project

The estimation for the global budget gave to this project described in the table 8, please bear in mind that the price of the component are relatives ,so at writing time of this lines the prices are as shown below.

Component	Price in Euro	Number	Total Price in Euro
APM Mega	48	1	48
Telemetry	59.87	1	59.87
GPS	81.15	1	81.15
Pitot tube	25.14	1	25.14
Battery	16.15	1	16.15
Power Module	23.94	1	23.94
Brushless Motor	14.95	1	14.95
Servo	4	5	20
Aircraft Module	249.99	1	249.99
R/C radio receiver	140	1	140
		Total Price	679.19

Table 8: Budget estimation

Note that the software are not included in the budget because they are all open sources and are totally available in the internet.

7. Conclusion

This chapter has received high focus and time due to its rule to describe the way we choose the hardware and the aircraft module, also it’s highlighted software used in the simulation, the price study has been added to this chapter .The next chapter will put under light the algorithm and code used in UAVs, besides, a part of simulation will be discussed.

Chapter 4: Algorithm and Simulation of Autopilot

1. Introduction

In this chapter we will drop light on the general algorithm how the autopilot works and steps should be done before go out for the first flight in real time. Also we will see steps for create your own model in plane maker then simulate it in X plane 10 simulator using simulation software in the loop (STIL) and simulation hardware in the loop (HIL).

2. General Autopilot steps

Before to simulate the open source we are using in this project for autopilot (Ardupilot Mega 2.5) is good to know the normal steps to follow for a normal UAV's fixed wings as we have of course.

Those steps we can collect them in five big steps everyone has so-steps bellow it.

The five steps are:

1. Modeling.
2. Guidance.
3. Navigation.
4. Control.
5. Simulation.

Note: Some source call step 2,3and 4 GNC (Guidance Navigation Control), also others they name only step 4 autopilot.

Thus, let's explain big axes for each step.

2.1 Modeling

In this step all your skills should focus on building a plane model from your imagination (I mean which you imagine to flight with), to do this you should know that there is to axes to stand by. First Flight Dynamic Model (FDM) you can find it by Newton laws and System identification.
To find this model there is three option as far as I know first to use a 6-DOF model from Aerodynamic Simulink toolbox already exist but the problem here you cannot change parameters, lack of flexibility, second is to use models from open sources like JSBSim but is hard to connect them with Simulink (our goal here), third is to build your own model in Simulink.
Table 6 below shows the open sources exist for this mission.

Open source FDM	Class	Software	User community	Documentation
JSBSim	Free & OS	C/C++	Big	Excellent
YaSim	Free & OS	C/C++	So big	Good
AeroSim	Free	Simulink	Average	So good

Table 9: Open Sources for FDM

Second thing you should focus on in this step is to find coefficients, I won't stop here a lot, you can use physics laws to find them and there are open source can do this job for you .I highly recommend DATCom (only if you are building fixed wings) but you can chose your own from the list (Autodesk, Tornado, AVl, LinearPro, Open Foam, ANSYSFluant …).

Please note that a lot of mathematics behind this (is not easy like it appears).For more details books in the reference page will be enough.

2.2 Guidance

What we mean by guidance is the ability of the plane to follow waypoints in order to achieve a mission, simple enough is how the plane will act to move from point 1 to point 2 (you might find instead of point leg don't surprised is just a technical word), the famous guidance algorithm is proportional Navigation used in fusels but there is two other ways for guidance L^{+}_{2} or what is knowing by LOS (Line of sight) and L1 controller.

Notice that in all loops we are going to show in this paper the inner loop is for control and the outer loop is for guidance ,also L1 and L^{+}_{2} are different systems .

The goal of guidance is to reach desire path, first focus on path following and then in advance future try precise waypoint.

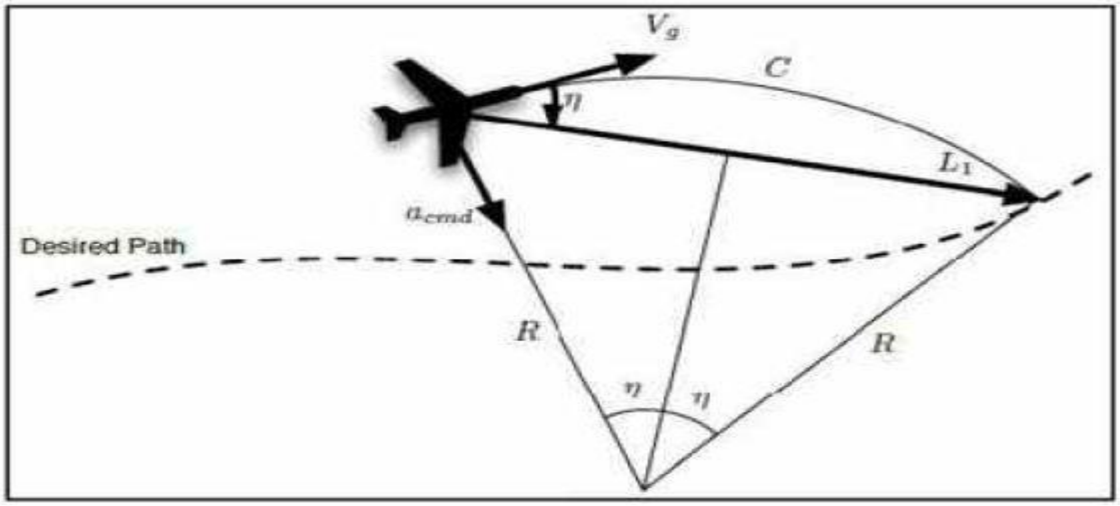


Figure 35: Plane trying to follow the path

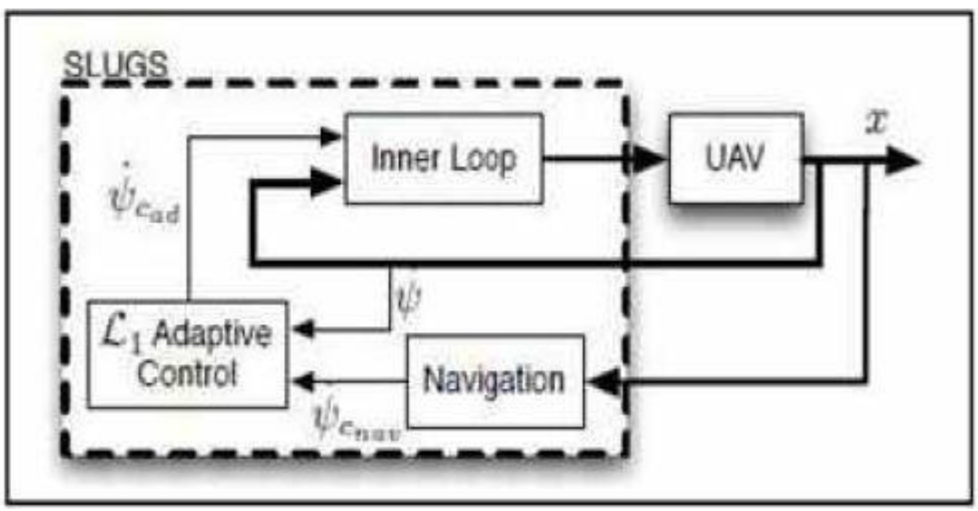


Figure 36: L1 controller as explain in SLUGS project

So many details to understand will be in the reference page, now let’s move to navigation step.

2.3 Navigation

Navigation process stand to help the program to find the position and the attitude of the plane in order to define the rotation rate for each axe, and acceleration rate forward for each axe. We have two technologies to combine between them to find satisfactory navigation result the first one is Global Navigation Satellite System which has only two systems GPS (Global position satellite)controlled by US or GLONASS controlled by Russia. The second technology to combine is INS (Inertial Navigation System).

Simple enough GPS gives us position, INS gives us Attitude, thus good navigation.

The best way to fusion GPS (I noticed GLONASS but I will not use it because it is not famous) and INS is Extended Filter Kalman invented in 1960 and used stage method and has cons, one of them is the price (so expansive).

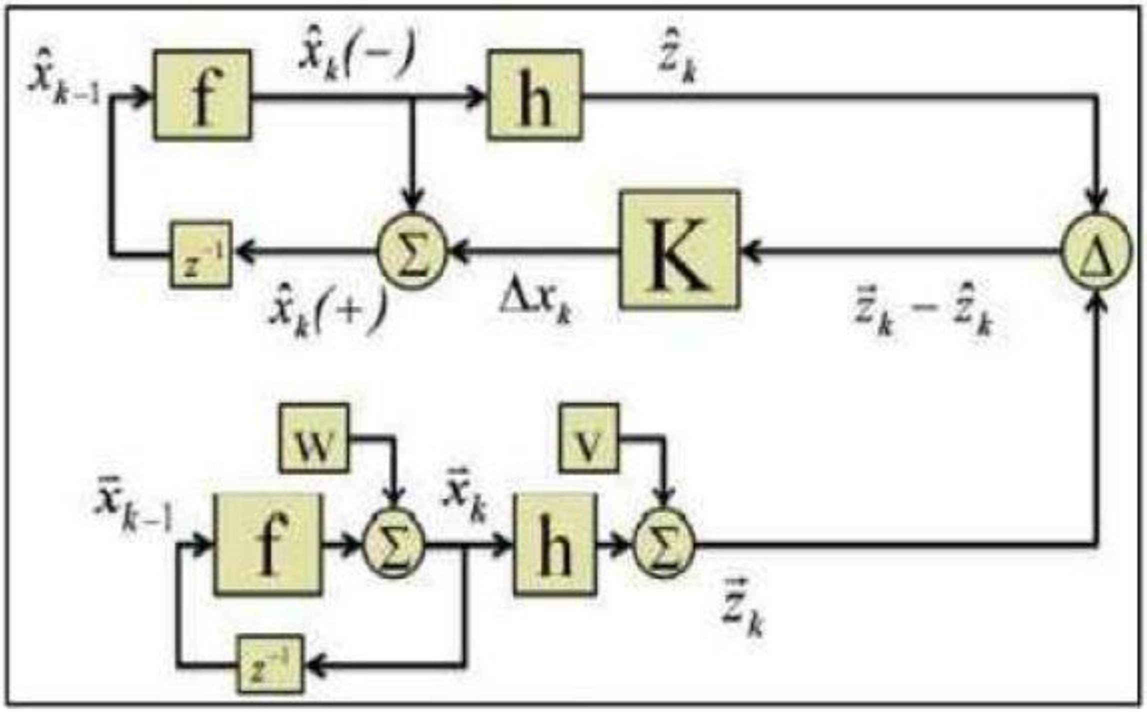


Figure 37: General view of filter Kalman

Clearly the output is nearly the same of desire result.

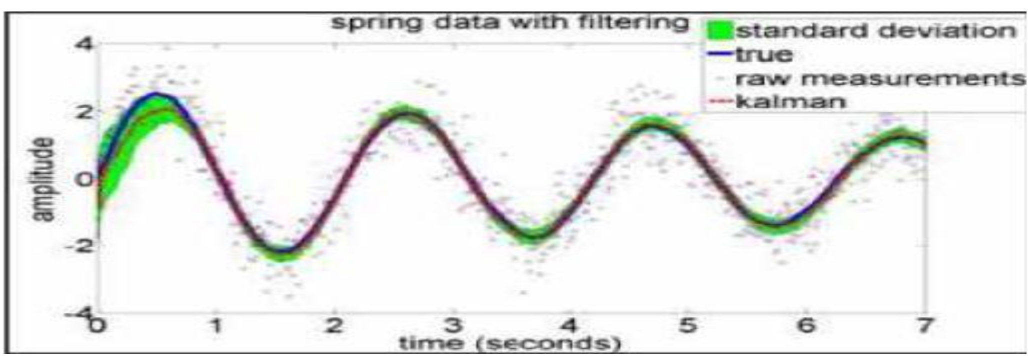


Figure 38: Kalman filter experience

2.4 Control

This step uses data collected in 2 and 3.I noticed before that control uses inner loop and guidance uses outer loop .as the following picture shows.

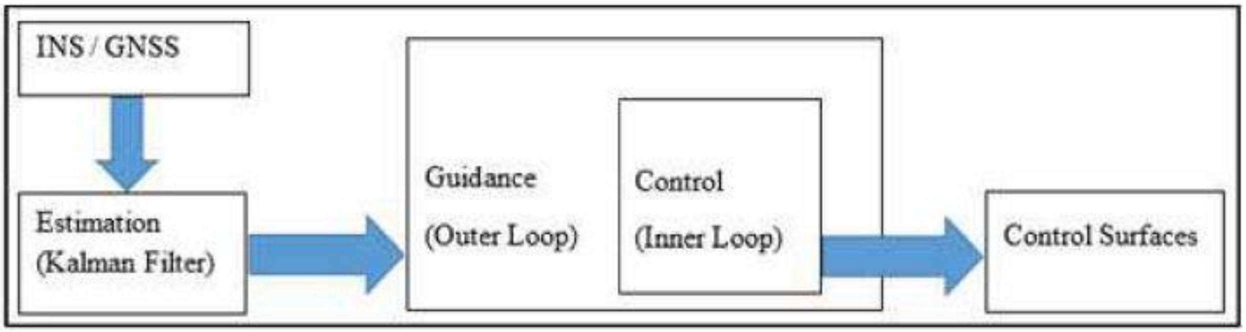


Figure 39: loops and their job

In control point view there are two channels for the plane should be controlled. Lateral channel (Ailerons and rudder) and longitudinal channel (Elevator and throttle).

In the following loops I am going to explain the principle for both of channels.

First let's start with lateral channel which is controlled by ailerons and rudder, we get turn rate and we do coordinate turn simply.

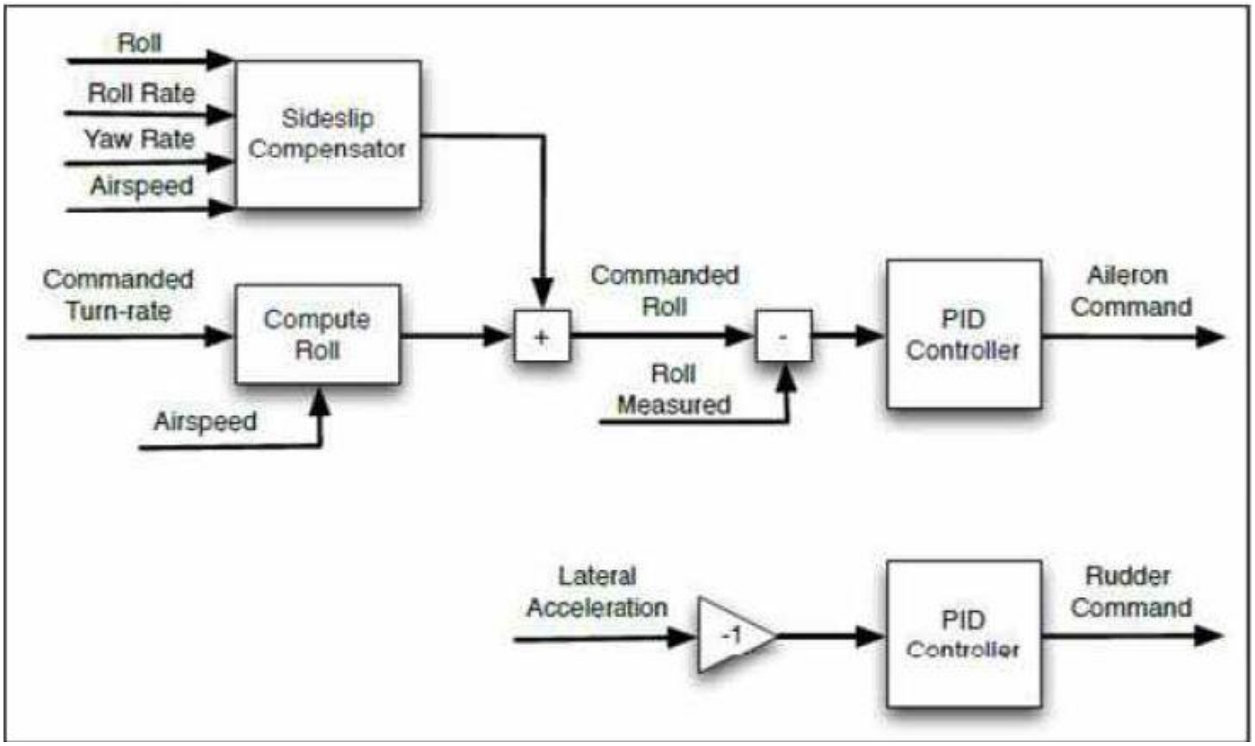


Figure 40: Lateral channel control

We are trying here to control the bank angel, to do this we will use ailerons (roll) but in the same time , we should do turning by rudder (yaw /lateral acceleration) in order to compensate adverse yaw coming from negative bank affect.

As shown in Figure 40PID (Proportional-Integral-Derivative) takes negative lateral Acceleration and gives rudder a command.

Then moving to compute roll using airspeed and turn rate information and subtract this value from measured one the result send to PID to control Ailerons as any other feedback loop works.

Now we finished with lateral channel.

For the longitudinal channel the things are similar but here our goal is to control pitch using elevator and speed using throttle as shown in figure below.

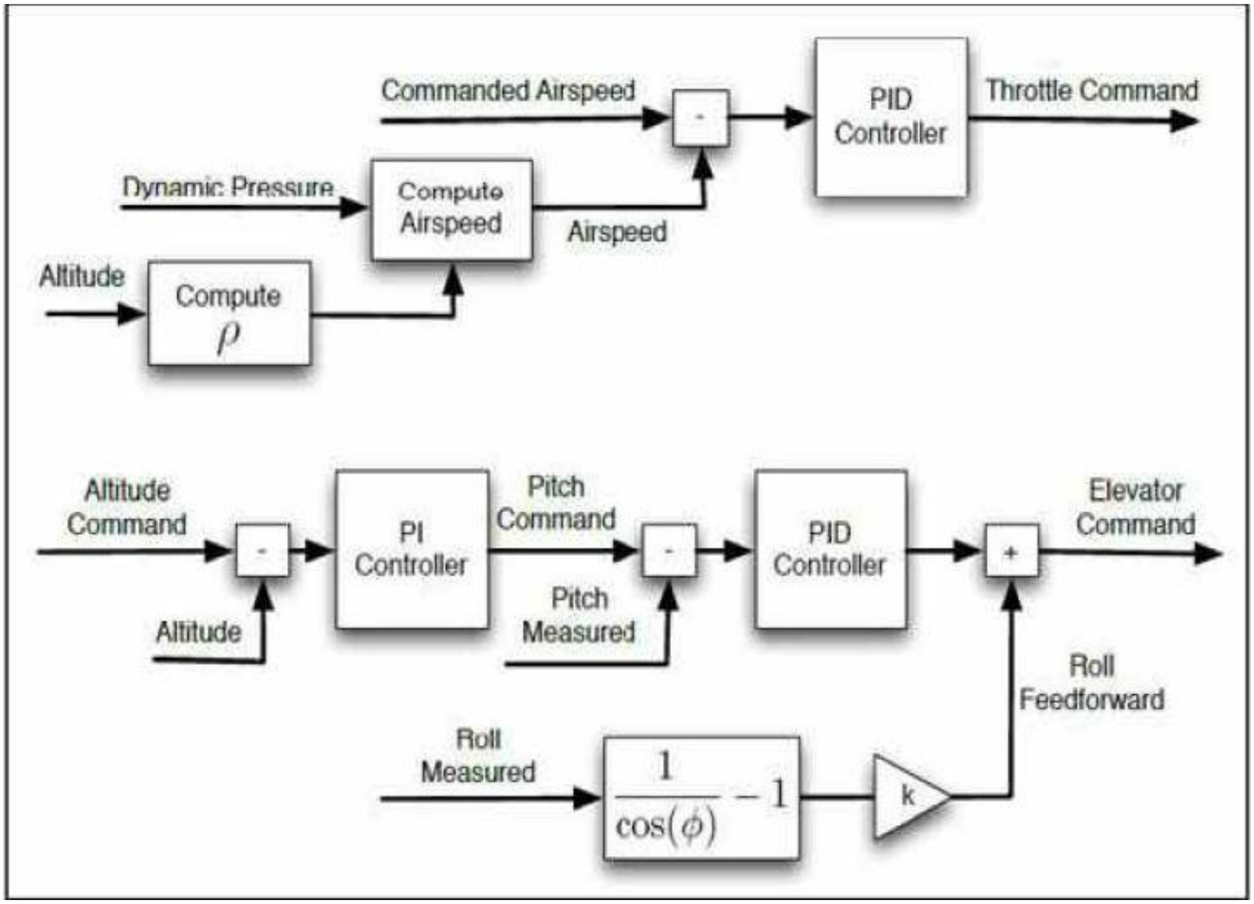


Figure 41: Longitudinal channel control

Though, command altitude comes and we subtract it from actual altitude and then send to PI controller to find pitch command again subtract this value from measured one and send the result to PID controller. But here before elevator command there something you should notice it.

We fixed in lateral control first negative affect which was compensate adverse yaw now is time to fix the second negative affect created by coordinate turn method which is decreasing altitude level.

Is it clear now that's why we are using the below block and adding this value to PID out value to command the Elevator.

For throttle, using intensity of air and dynamic pressure we can find airspeed for the plane comparing this value with commended one and give it to PID controller to have your speed command transmit to the motor (electric one or other type doesn't matter for us).

In this point we finished step number four, move to see our work and visualize it.

2.5 Simulation

In this step we want to show you how to see your work in order to develop or modify it you got out for real flight (you risk to lose your aircraft if you skip this step).
There's two ways either Simulation in the loop or Hardware in the loop and you can use one laptop running Simulink and Control Ground station or Simulink, Control Ground Station and Real flight Simulator like X-Plane or Gear Fighter.(You can use two computer is better).

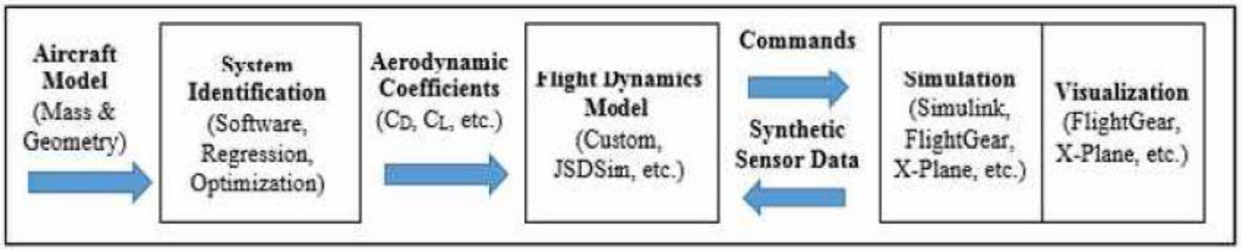


Figure 42: Simulation structure

Hardware in the loop Simulator generate synthetic data like if the plane is in the air and the it send to avionics which gives order to control surfaces too and the loop is closed. Notice that if the plant model (I mean your FDM and Simulink model) is precise the tuning of the real coefficients especially PID will not take a lot of time in real aircraft (close value generated).If you are using two computer please follow figure 43 bellow for communication issues.

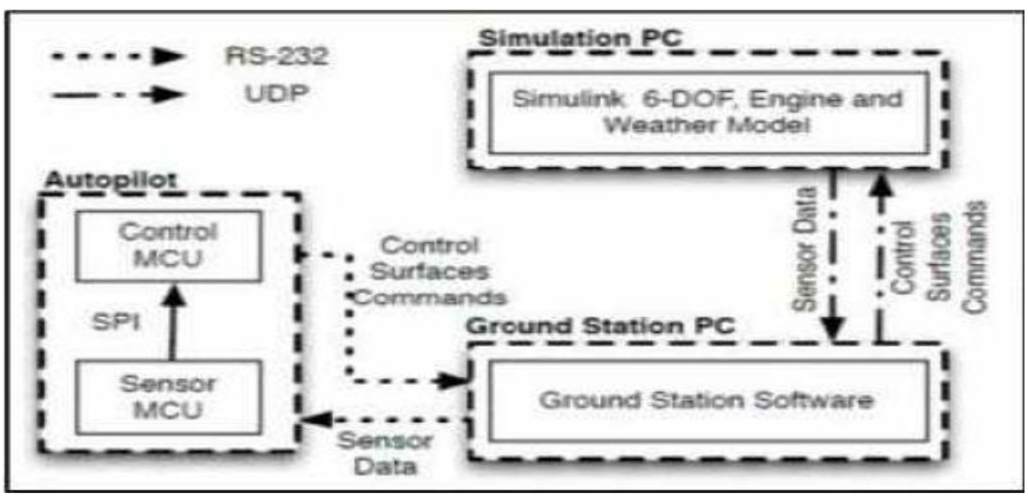


Figure 43: Communication issues

The last thing is visualization you can see your model in X plane or Gear fighter while you are running Simulink and you can watch you mission flight like you are outside (X-Plane is not a game is an engineering tool for simulation).

3. Plane Maker

There are as many different ways to go about working in Plane Maker as there are aircraft designers. The following steps, though, serve as a good workflow sequence to start from when modeling in Plane Maker:

1. Decide on a design.
2. Create the fuselage, wings, and tail of the aircraft
3. Create secondary objects, such as landing gears and engine nacelles.
4. Set up the systems and internal properties, including the engines, electrical systems, weight and balance, and viewpoints.
5. Set up any additional features of the aircraft, such as added weapons or special controls.
6. Create a 2-D instrument panel.
7. Test-fly the aircraft in X-Plane and fine-tune the features of the aircraft from steps 1-6 as needed.
8. Add textures, 3-D objects, extra liveries, etc.

After struggling to learn this software we came to the model shown in figure 44 as an approximation to the model we want to fly by.

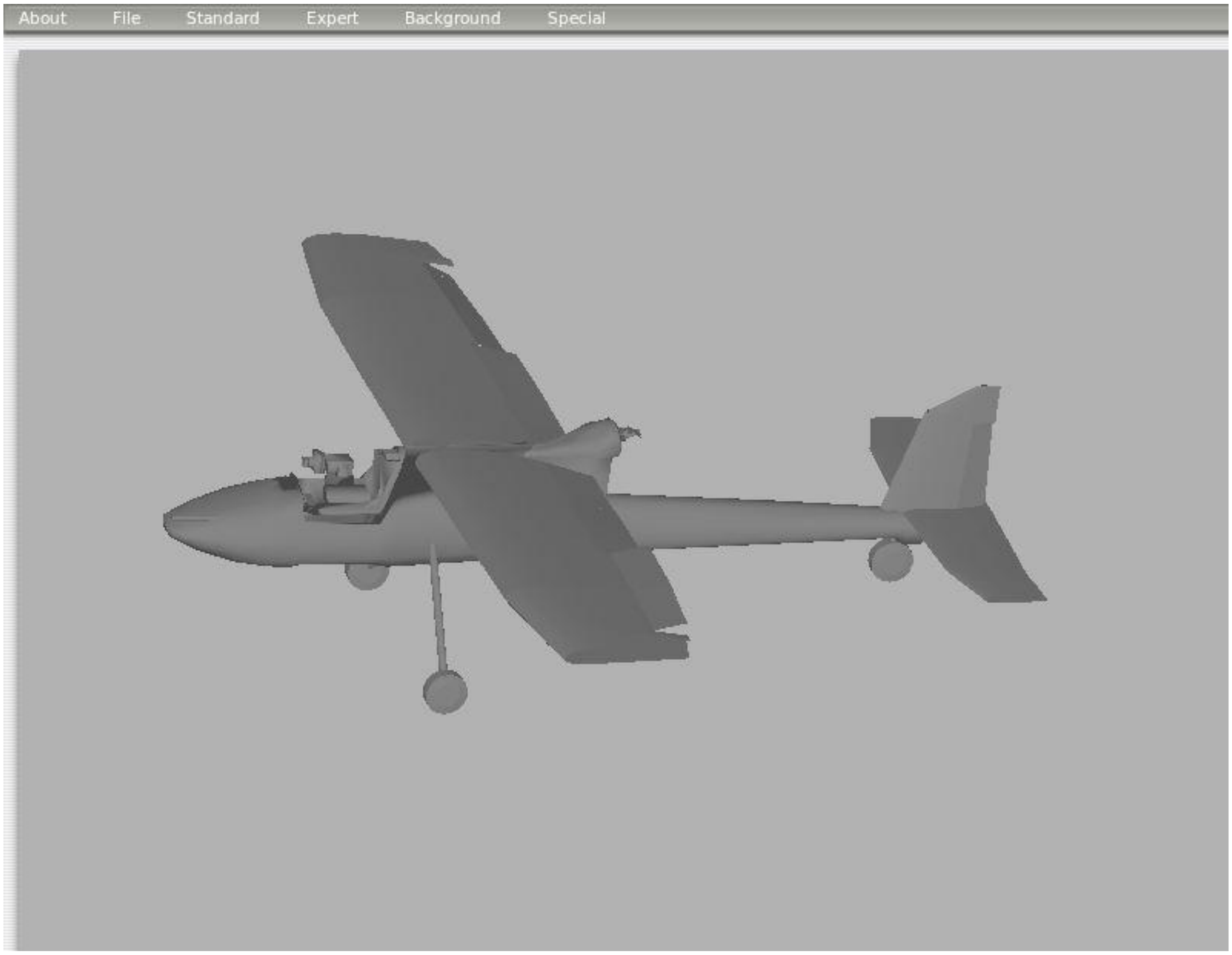


Figure 44: Approximation of Volantex RC Ranger EX 753 model in Plane Maker

This is a model we built in Plane Maker in purpose to use it for the X-Plane simulation, behind the picture there's several parameters should be adjust like power and type of the engine, also empty and maximum weight, control surfaces and others discussed in appendix 6.

4. Simulation hardware in the loop (HIL)

Hardware in the loop simulation (HIL or HITL) means to feed simulated sensor data to the autopilot and to return its control outputs into the simulation.

The HIL simulator consists of three main parts:

- Sensors
- Process
- Actuators

The Sensors part simulates the sensor's output data in the airframe. This data will be processed by the UAV autopilot hardware as input. The sensor output data that should be produced by HIL simulator are position data (speed, altitude, latitude and longitude) and attitude data (roll, pitch and yaw). This can be accomplished by reading data from the simulator. The Actuators part simulates how the UAV autopilot hardware can change the surface control of the airframe (aileron, elevator and rudder) and throttle position. In real world application this will be done by controlling the servos put in the corresponding control surface or throttle engine. In HIL simulator this is done by writing data to the X-Plane that will affect the control surface of the simulated airframe. The Process part simulates how the airframe will react to the input given by the UAV autopilot hardware. So basically this part is where we should put the system dynamic model (transfer function). Generally this is the most complex part of the HIL simulator, but fortunately this part is already provided by the X-Plane (using its blade element approach). The HIL simulator plug in communicates with the UAV autopilot hardware through RS232 serial communication.

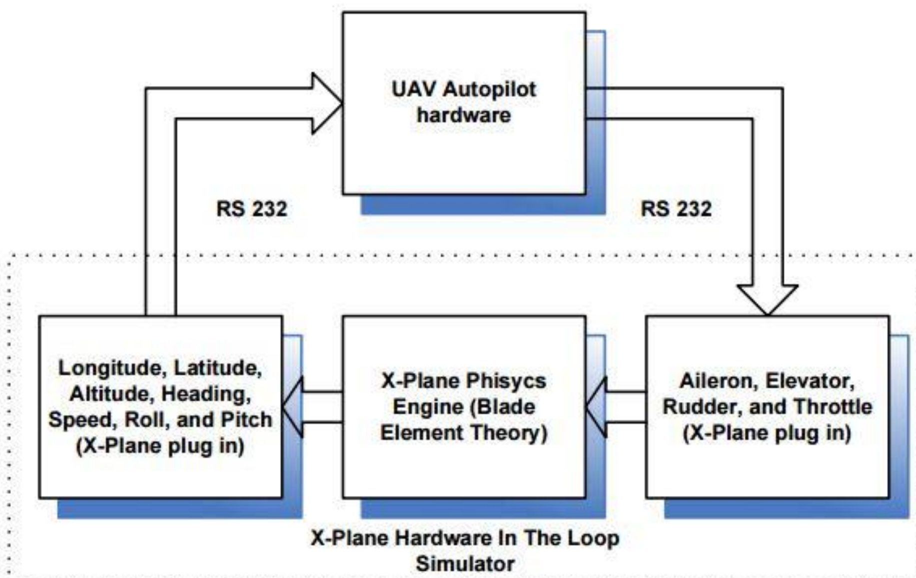


Figure 45:X-Plane as HIL simulation platform

4.1 Data Inputs and Output Settings

X-Plane needs a number of communication settings. Open **Settings → Data Input & Output**. For X-Plane 10 users: X-Plane 10 data settings. Check all the checkboxes below:

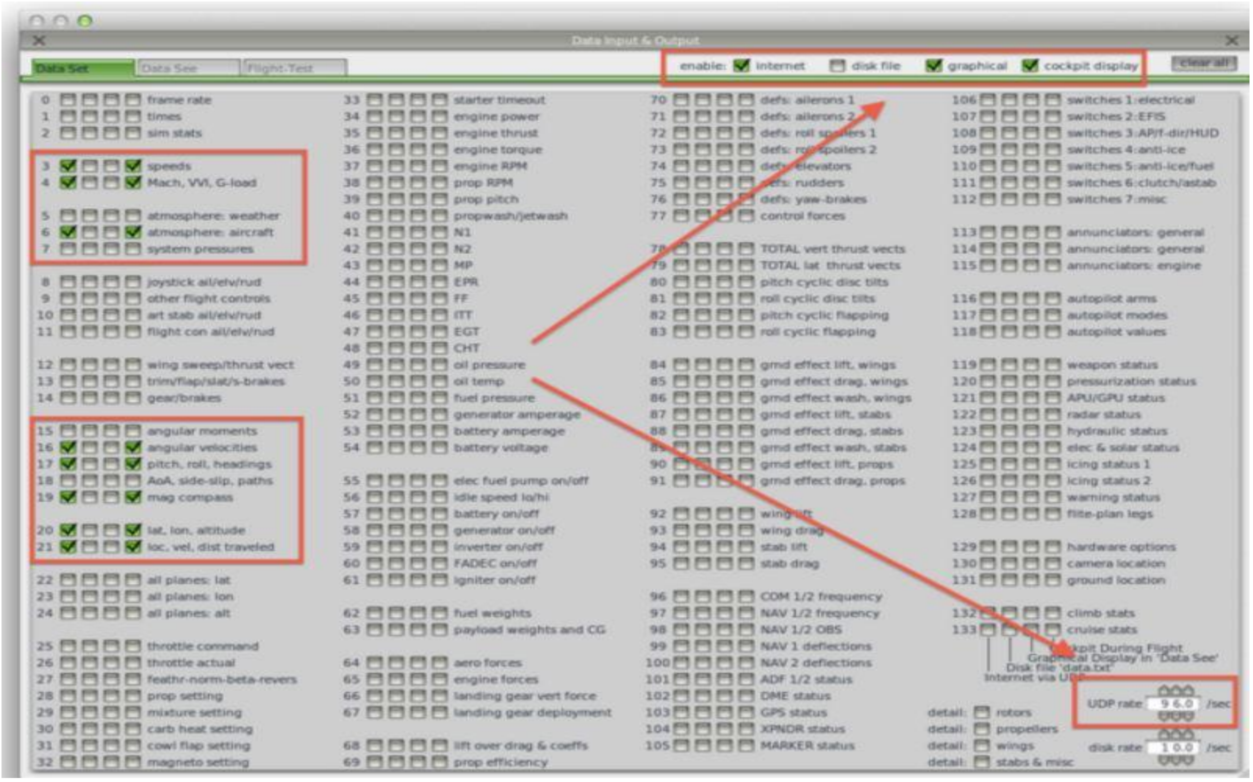


Figure 46:Data Inputs and Output Settings

4.2 IP Settings

On the IP screen of X-Plane (**Settings → Net Connections**), set the UDP data port to 49005 as in the picture below. QGroundControl uses port 49000 to send/receive data. The complete port setup is:

Port 49000: X-Plane port (from X-Plane to QGC: 49000 → 49005)

Port 49005: QGroundControl port (from QGC to X-Plane: 49005 → 49000)

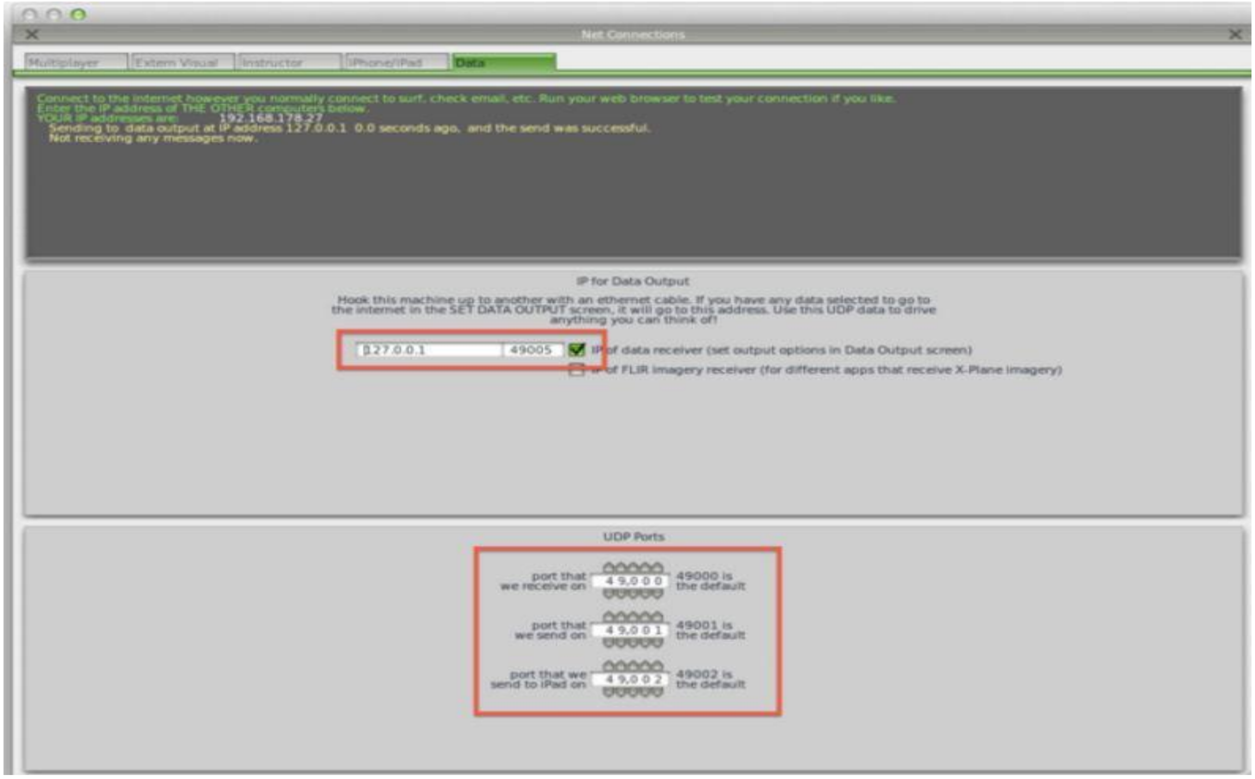


Figure 47: IP Settings

4.3 Simulation Update Rate Settings

The X-Plane simulation (physics) usually runs at the graphical frame rate. For small models like RC aircraft higher rates are needed. Open the menu at **Settings → Operations & Warnings** and increase the rate as shown below:

Figure 48: **Simulation Update Rate Settings**

4.4 Hardware in the loop simulator 3D visualization

Total time spent in the HIL simulation is comparable to 150 flight hours in real world field trial. After successful test in the HIL simulator the autopilot hardware is installed into UAV test bed prototype. This UAV prototype achieved its first successful autonomous flight, the mission is 6 waypoint autonomous navigation with loitering and cross tracking.

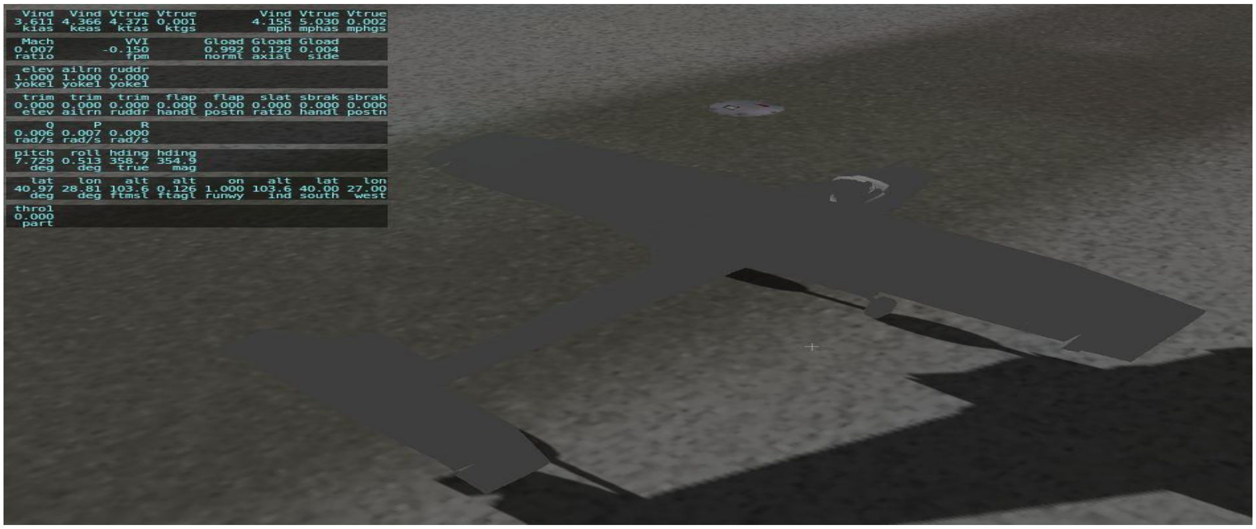


Figure 49: **Hardware in the loop simulator 3D visualization**

The utilization of HIL simulator in the UAV development life cycle is proved to be very valuable. It enables the developer to test many aspects of the UAV autopilot hardware, finding the problem, refine the firmware, test the reliability, fine tune system parameter, and many others. All of those iterations are done inside development environment without risking the valuable airframe and payload. The HIL simulator also has the potential for low cost training tools for UAV operators.

5. Simulation software in the loop (SITL)

One of the disadvantages of HIL is you need in every simulation all hardware to connect them together, thus, we saw that is not practical enough for a new project needs a lot of attempts to learn it.

In this purpose to have a smooth way for simulation is a priority for an engineer, SITL could fix this problems for us, and we could have a simulation without any hardware and in a fast way.

Using X-Plane with SITL is a good way to get some experience flying ArduPilot and learning how to use the ground control station (Mission planner in this project). It can also be used to see how ArduPilot handles unusual aircraft and to develop support for aircraft features that may not be available in other simulator backend.

TO do this we followed the architecture in figure 50 below:

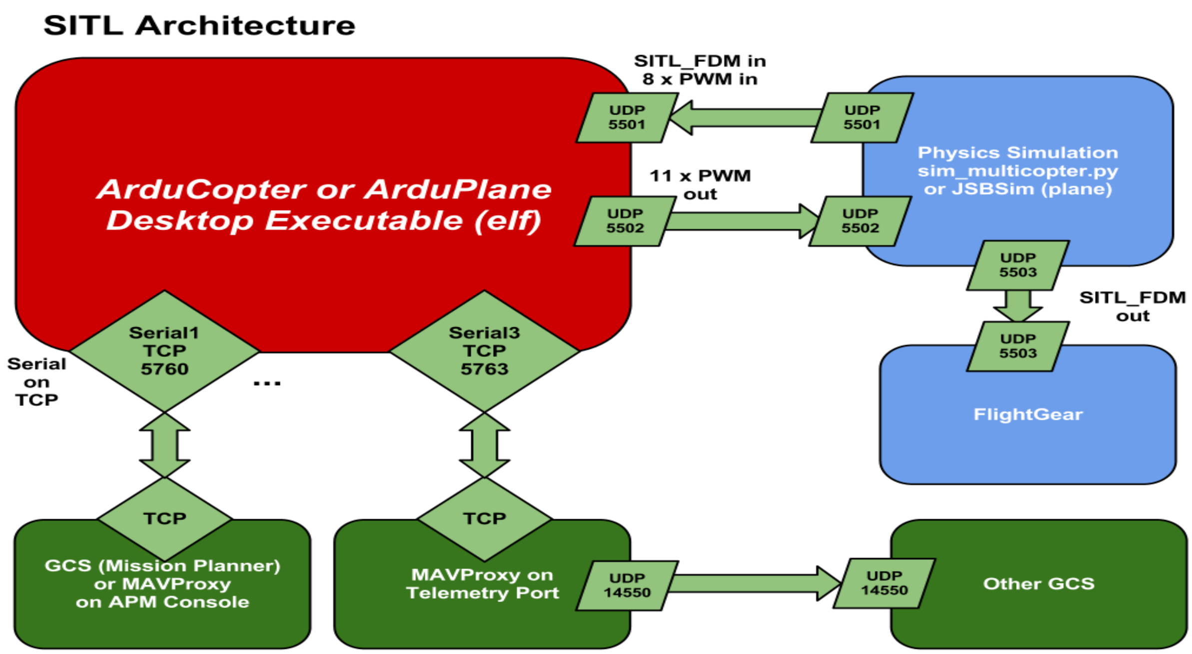


Figure 50: SITL architecture

5.1 Setup of X-Plane 10

Before starting SITL the only thing you need to setup on X-Plane is the network data to send the sensor data to the IP address of the computer that will run ArduPilot. This can be the same computer that is running X-Plane (in which case you should use an IP address of 127.0.0.1) or it can be another computer on your network.

To do that go to the Settings -> Net Connections menu in X-Plane and then to the Data tab. Set the right IP address, and set the destination port number as 49001. Make sure that the receive port is 49000 (the default), as shown in Figure 51 below.

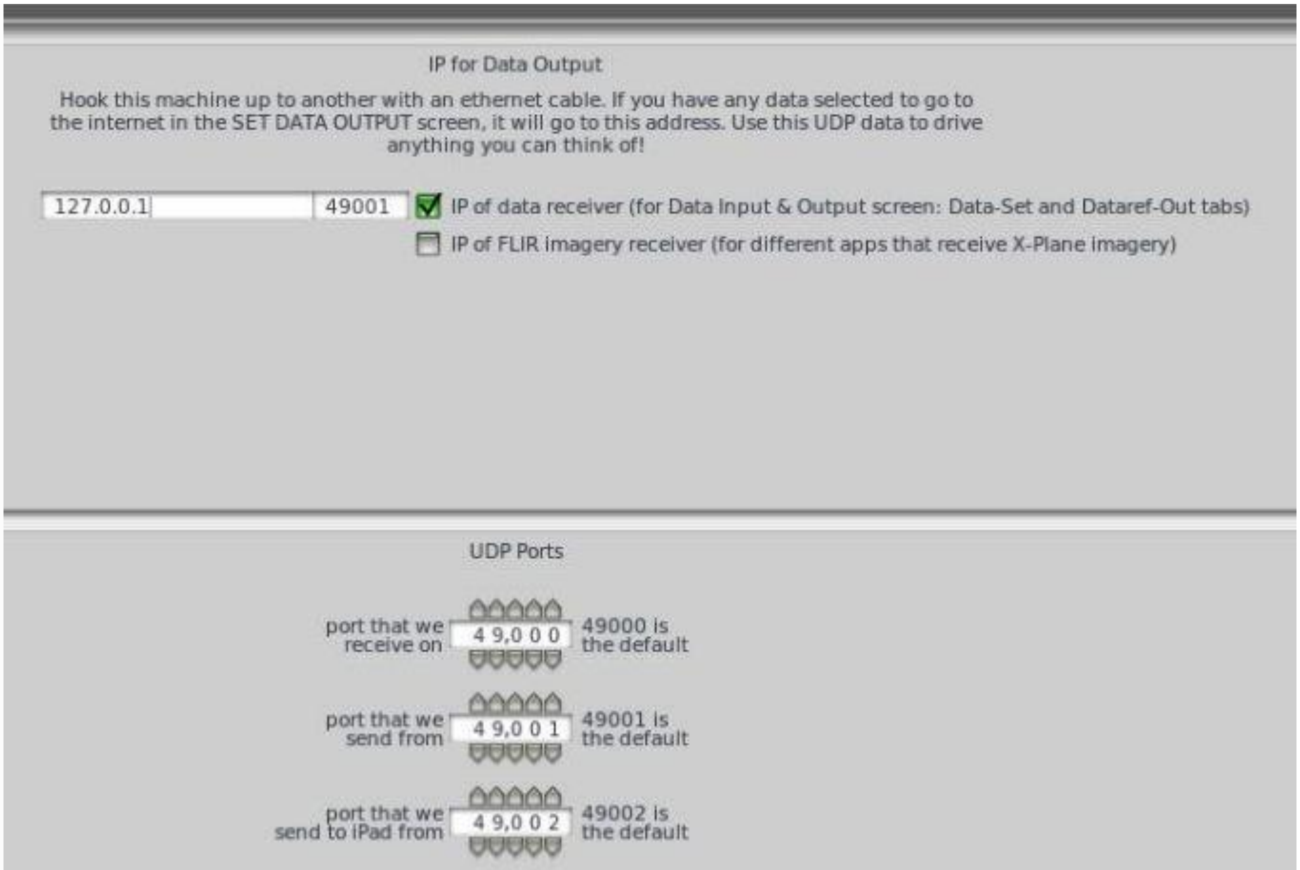


Figure 51: IP and ports setting for X plane 10

5.2 Running SITL within Mission Planner on Windows:

This approach is good if you just want to test ArduPilot with SITL but you don't want to make changes to the code (our first aim in this project).

Mission Planner will download a build of ArduPilot SITL for Windows that is built each night from git master.

To start SITL directly from Mission Planner we need to have a very recent version of Mission Planner which is 1.3.39 available in network for free.

In the SIMULATION tab select X-plane and X-plane 10. Then select Advanced IP Settings and click through the IP addresses, set them to 127.0.0.1, with the default network ports, as shown in figure 52.

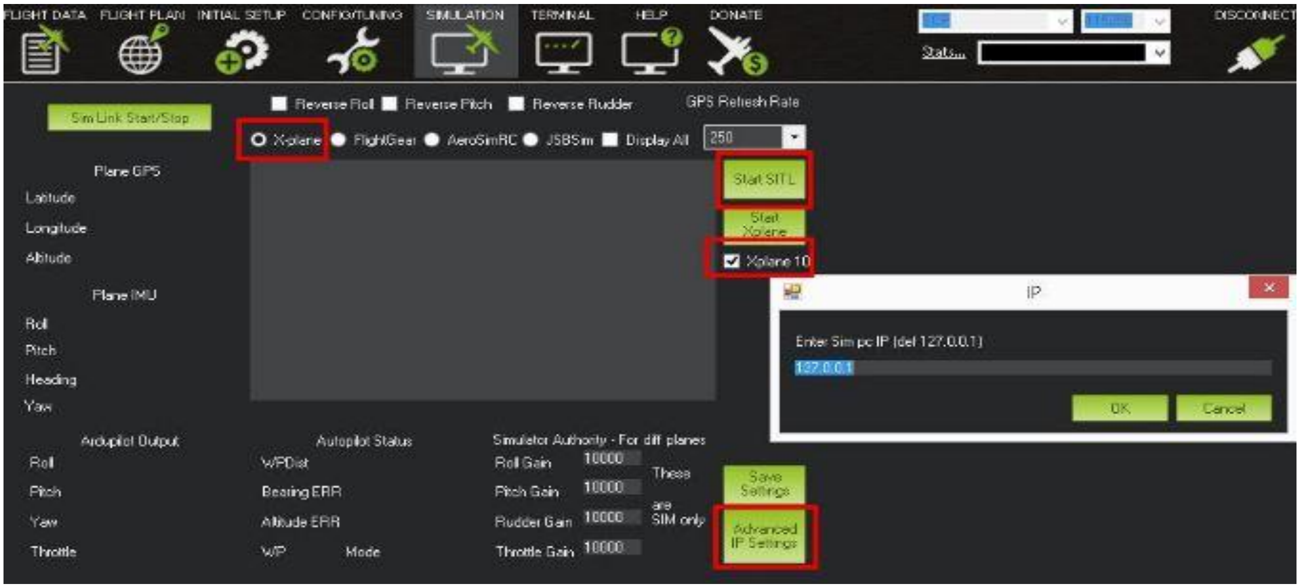


Figure 52: Mission planner with X plane communication

After that we can easily start the SITL and launch a flight mission chosen plane frame in MP for fixed wings aircraft (Autopilot code by default), and setting all parameters for our model.

Figure 53 shows a full mission launched in Ataturk Airport with a normal take off and normal landing.

As we can notice the mission combine several waypoint the aircraft should navigate them with a determined altitude for each one.

The take-off had an angle of pitch of 15 °C (for the small aircraft we use angle between 10 and 15 °C), and altitude of 30 meter when the aircraft reach it we consider that the take-off is finish and the plane should start navigate to the first waypoint, after completing the mission the plan should land automatically with an altitude of zero meter.

Using SITL allowed us to switch between different flight modes while the plane in the sky such as Stabilize mode or RTL.

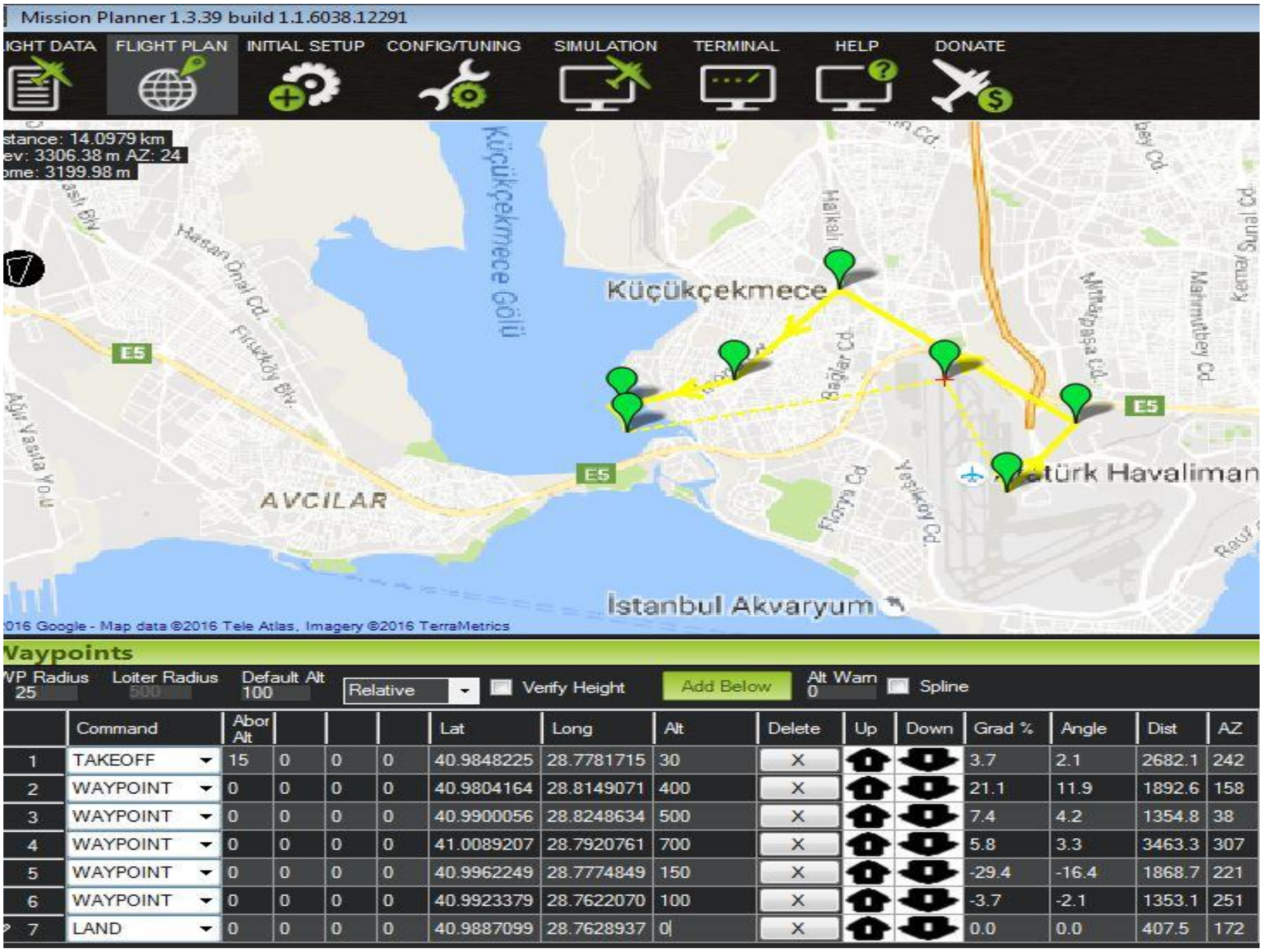


Figure 53: Full auto mission launched

Opening the X-Plane simulator we can visualize our flight exactly like its real, and take notes of problems facing the aircraft.

6. Autopilot Using MATLAB/Simulink

6.1. UAV dynamic model (FDM)

It would be reasonable to minimize the risk associated with control system implementation, by running it through simulation and analyzing its behavior. Matlab/Simulink provide a rather powerful tool for engineers in various discipline to simulate and analyze mathematical models and control systems. Special set of blocks and functions are available as part of the 'Matlab Aerospace Toolbox/Blockset', 'Matlab Control System Toolbox'. They are widely used by engineers to simulate aircraft dynamics and control. The focus of this lines is on simulating UAV model and controller developed for this project. Simulation results will be covered in the next chapter.

To go forward in our project we were looking to simulate the whole part of the project using Simulink tools offered by Matlab especially aerospace library which allowed us to build the dynamic model of our aircraft (the details are not presented here due to the confidentiality of the project).

Using the geometry and mathematical models of the plane (offered by the team) we came to the UAV dynamic model presented in the figures 54, 55, 56 and 57 below.

The total force acting on an aircraft is comprised of three elements: Thrust, Gravitational and Aerodynamic force.

Thrust force is directly produced by the aircraft propulsion system. The task of this block is to compute contribution of thrust force in the UAV total force and moment. Aerodynamic forces and moments are functions of aerodynamic coefficients. These coefficients are obtained from Datcom for different AoA (Angle of Attack) and control surface deflections.

After evaluating coefficients at angle of Attack and control surfaces deflections, Simulink block entitled "Aerodynamic Forces and Moments" receives these coefficients and calculate forces and moments due to aerodynamics. Gravitational force is calculated based on mathematical model.

Finally, there is the "6 DoF Euler angle" block. This is a built in Simulink block (Part of aerospace blockset) that receives forces and moments and returns 6 DoF Euler angle representation of equations of motion. It basically provides information on aircraft position (Xe, Ye, Ze), orientation and velocity (u; v; w; p; q; r).

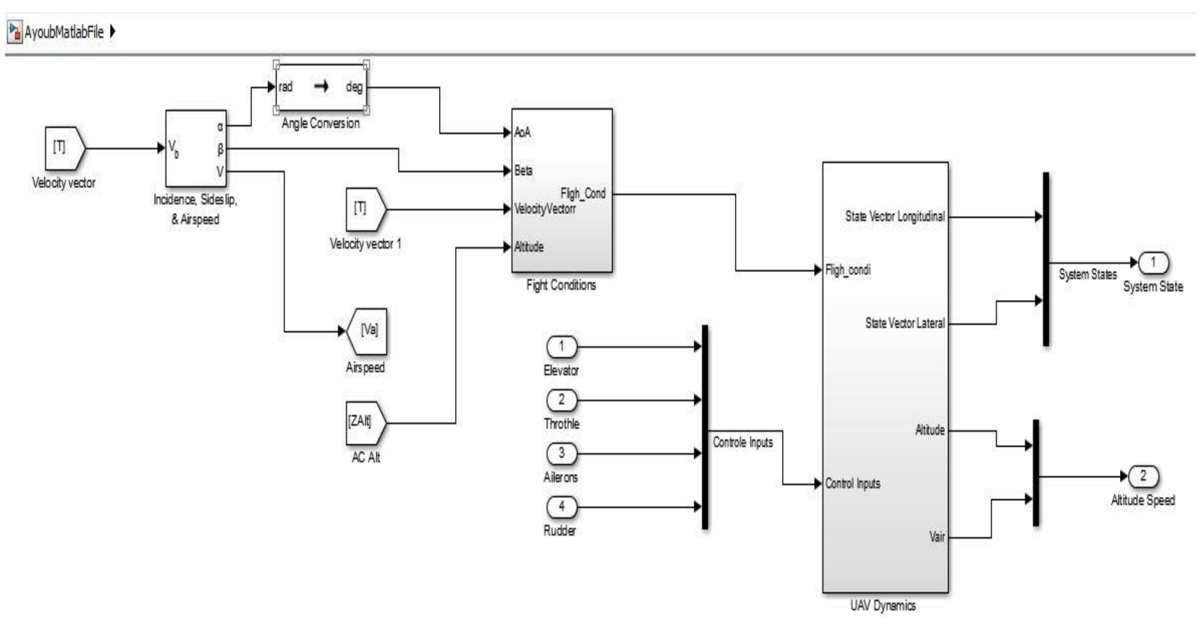


Figure 54:Non-linear System

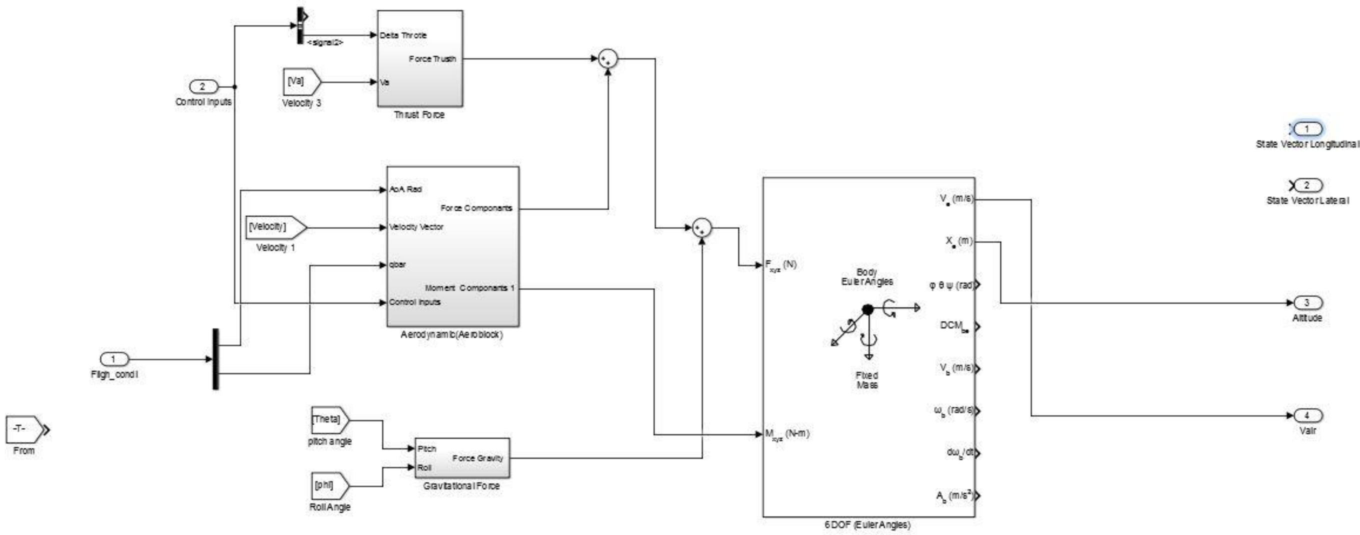


Figure 55: UAV dynamics

6.2.Simulation of control loops

Controllers developed above in this chapter are longitudinal channel controller and lateral channel controller.

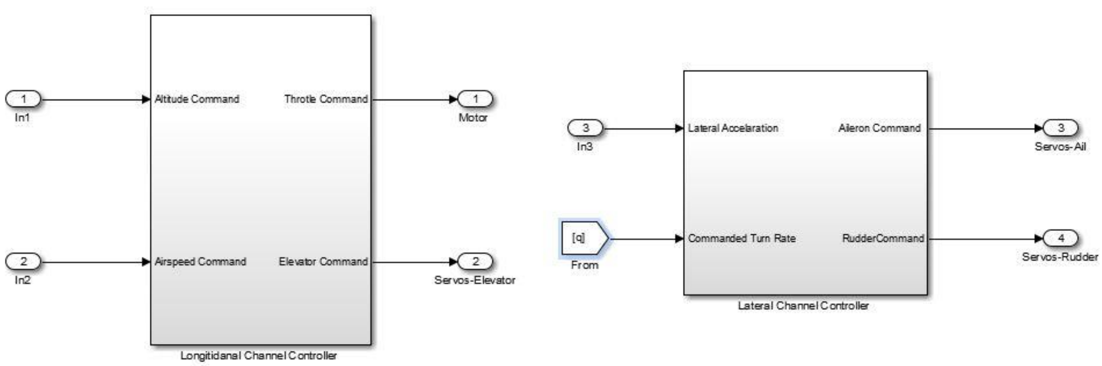


Figure 56: Controllers in UAV

6.2.1 Longitudinal Controller in Simulink

Now let us examine the longitudinal controller block. Overview of the simulated longitudinal controller is illustrated in Figure 57. The block takes two set of inputs: Input Commands and Longitudinal States. This control the throttle and the altitude (Engine and Elevator).

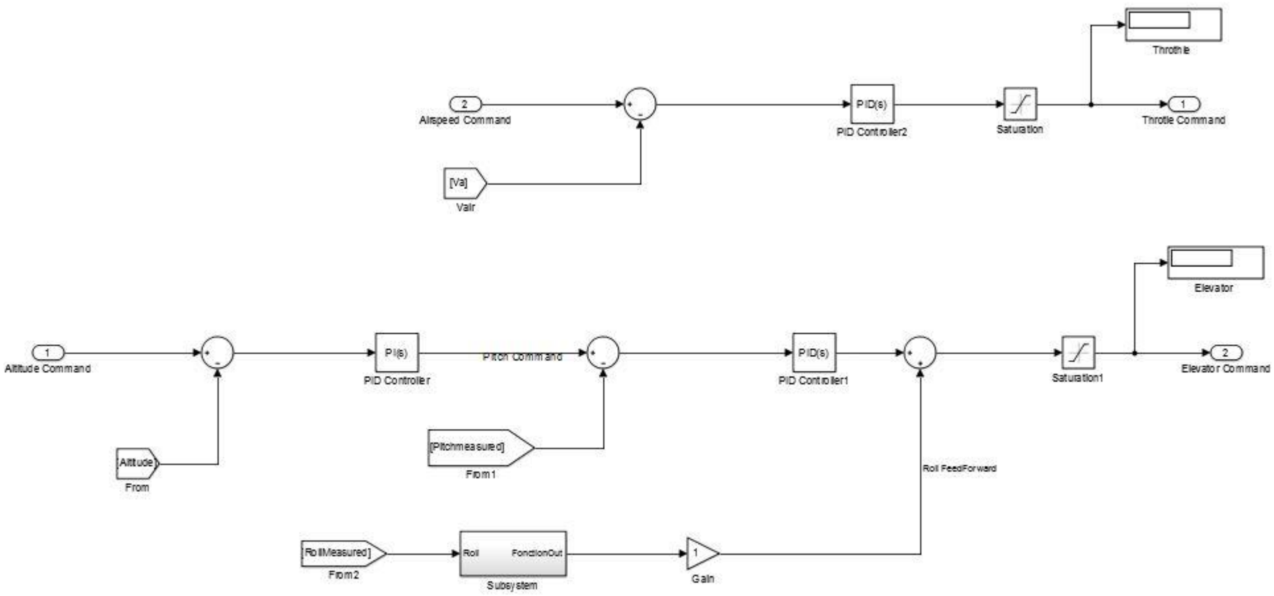


Figure 57: Longitudinal controller

Altitude command is subtracted from the current altitude to produce an error signal. This error signal is fed through the PI controller. Same goes with airspeed command and the integrator. Output of both controllers are subtracted from control signals generated through L_2^+feedback (Guidance controller). This produces final, combined control signals that are further passed through saturation blocks which represent limitation in aileron deflection angle and throttle opening.

6.2.2 Lateral Controller in Simulink

Lateral Controller block is illustrated in figure 58. Same as before, the error signal (in this case the difference between commanded roll angle and actual roll angle)is fed through the PID controller and is subtracted from lateral states to create control command for ailerons servos.

Here we control the roll and the yaw for the plane that’s mean the direction (right or lift) using ailerons and rudder.

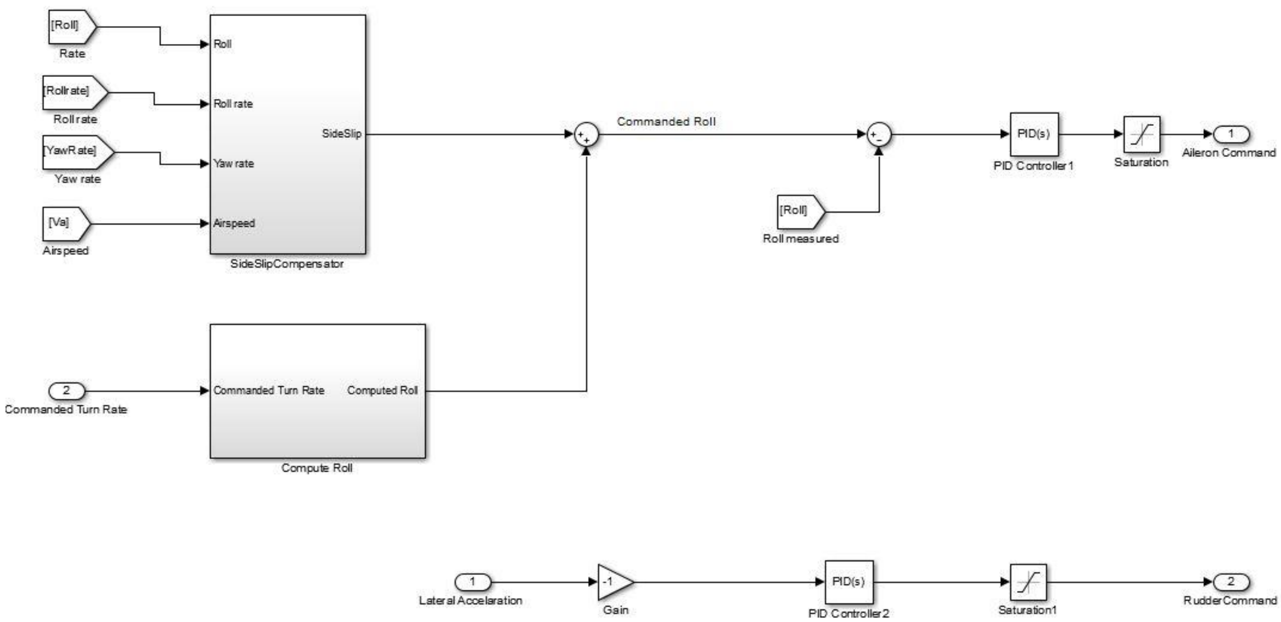


Figure 58: Lateral controller

Note that all the blocks are built base on mathematical models are not presented in this paper due to confidentiality of the project as mentioned above.

7. Conclusion

This chapter has been reorganized to present different ways we choose to simulate the autopilot chosen to be used in this project, also it gives us an overview of the algorithm use in UAV's system. In the end of it we presented several blocks built in Simulink such a UAV dynamics and controllers.

THIS PAGE INTENTIONALLY LEFT BLANK

Chapter 5: Discussion and Future work

1. Discussion

1.1 Moral and Ethics

While unmanned aerial vehicles are being deployed and explored at an increasing pace, the moral dilemma and debate of their use has grown. There are a variety of situations where UAVs are useful. As previously stated; their application within military, civilian and industrial areas are being explored and is often regarded as an effective solution.

Consequently, as with most new technology, the ethics and morals in their establishment becomes a relevant topic of discussion. The use of UAVs Beyond-Visual-Line-of-Sight (BLOS) are currently illegal in several countries, Turkey being one of these. The reason behind these laws will be highlighted and also the benefits of removing them.

1.2 Safety

The obvious downside with regard to safety for UAVs is that it introduces a risk of falling out of the sky and hurting people, animals or property. Almost any malfunction or other unanticipated situation will result in a crash for a UAV. If a drone stop functioning correctly it will, unlike cars or other vehicles, gain speed instead of slowing down. It is also difficult to see and have time to react to an object falling from above, since it is never expected or within a human's visual line of sight. Moreover, since an UAV does not have a pilot onboard or no pilot at all, it can be difficult to execute an evasive action if an unexpected object presents itself.

Besides the safety of people on the ground, there is also the decreased safety for pilots in manned aircrafts sharing the same airspace as drones. A collision risk is introduced between manned and unmanned vehicles, which can pose significant danger to the pilots and other passengers. A drone is often small and can be difficult for pilots to spot. Even though UAVs would incorporate transponders to be easily detected and have a sophisticated avoidance system, other pilots would not be able to communicate with these if necessary.

However, for the pilots controlling these drones or have set an autonomous mission the safety is drastically increased and even ensured. Using an UAV poses no threat at all to the one controlling it. It can be used for otherwise dangerous reconnaissance missions in a hazardous or unfriendly environment without risk for the pilot. Also, their ability to be deployed instantly have numerous uses for search and rescue missions. Information about a scene can be obtained quickly and be used for critical decisions.

1.3 Simulation and controller tuning

Simulation (STIL and HIL) is the only proof obtained that shows that the autopilot is working correctly, but might not be optimally tuned. In the simulation environment a virtual model is used that have the same or similar attributes to the physical plane. The size, weight, balance and center of gravity is the same or very close to the physical plane. However, there are approximations due to limitations in the simulation software. These are although expected and the achieved tuning from the simulation is not assumed to be optimal, but at least be able to keep the plane airborne when testing. Evidently, the controller could not stabilize the physical plane. It did however operate the virtual plane with precision and speed in simulations. Some of the possible causes are presented below:

- The approximations of the virtual model and physics in the simulation environment were too large and not similar to the physical plane.
- The simulated weather conditions in the simulation were not as harsh as the weather conditions during physical testing, hence, the controller is not tuned to be robust against heavy disturbances.

One approximation in the virtual model that could have caused deviating flight characteristics are the airfoils. In order for the simulation to calculate the correct forces acting on the plane it needs to know the shape of the wings, i.e. the airfoils. Since the airfoils are determining the lift, drag and other aerodynamics they are essential for designing an exact virtual model. However, the exact airfoils for the VolanteX RC Ranger EX 757-3 FPV is not known and could not be obtained. Instead an approximate airfoil was chosen based on being similar to the cross-section of the VolanteX RC Ranger EX 757-3 FPV and also being recommended as an appropriate airfoil for smaller flaying wings. If this approximation have made a major difference in flight characteristics is not obvious since the subject is complex and requires extensive testing and knowledge to answer.

Besides, while switching to Simulink model several problems started to appear such how to model complex matrices plane uses them for stabilization and to calculate vector state for longitudinal and lateral channels, so a dilemma appears which way is more efficient to build a block function in Simulink or to write a Matlab code in workspace and relate it with Simulink file.

1.4 Autopilot

Two autopilots were considered for this project. The software comparison was mainly focused between the PX4 and Ardupilot Mega, being the more popular autopilots. Moreover, the viable hardware alternatives was the APM board and Pixhawk. The different software are optimized or only compatible with certain hardware, which created restrictions in combining hardware and software.

Ardupilot is implementable on both the APM board and the Pixhawk, but does not have all the functions when running on a Pixhawk. The Pixhawk/PX4 combination is a newer and faster autopilot. The Ardupilot is older and runs on an 8-bit architecture, i.e. it does not have much room for improvement regarding speed and additional functions. The Pixhawk runs a 32-bit architecture and is therefore significantly faster, but is constantly being developed and is not as thoroughly tested as the Ardupilot. It was preferable to choose an autopilot that could be improved, but as it is still in development, the APM has been behaving erratically since the beginning.

The PX4 board was tested as well, as an alternative to compare the performance and robustness. The PX4 board is more robust and efficient than APM which mean we can carry on the project with it in the future stages.

2. Future work

An interesting recommendation for future work would be to use a platform connected with camera to help the UAV to analyze data and change the waypoints mission according to facing obstacles using C++ and Visual Studio that could be really interesting.

Another interesting point for future research is the following: Adapting the open source used for GCS by adding bottoms to start stored missions immediately that's will help to reduce time and errors.

Finally, although lateral and longitudinal controller was developed here (please refer to Simulink files in DVD), there are limitations on the time period that the plane can roll or pitch. To somehow overcome this problem would be another open area for further research.

General Conclusion

The objective of this project was to choose and to validate an autopilot system for a small UAV (Unmanned Arial Vehicle) in order to implement it later on in a real one. This objective was achieved by determining the major functions of an Ardupilot Mega system available as an open source, and results were sufficient to show that we can launch a real flight with it.

This thesis brings together knowledge and advancements in many scientific areas and engineering disciplines, namely: mathematical modeling, aerospace engineering, control system, Mechatronics engineering and computer science. Large part of the work focused on choosing the right Autopilot system –Hardware and Software- for new UAV being developed and modeling the controllers of the system. A nonlinear model was developed and simulated in Simulink. Static and dynamic stability analysis had also been accomplished in Datcom, which is one of the most widely used software for aircraft aerodynamic analysis.

After modeling the chosen plane, various simulations were accomplished such STIL and HIL to analyze performance of Autopilot chosen under system uncertainty. Particularly, the influence of variations in the UAV mass and center of gravity (CG) location were studied. Based on the results obtained, we can conclude that the Ardupilot functions properly with acceptable robustness. Furthermore is being tremendous even with default parameters.

We can also conclude that using Matlab Simulink for simulate autopilot loops is better than just used the default code because is offering the ability to change any parameter you think has an effect on plane behavior.

One thing that hereby must be told is that for me it has been a great experience to be in a foreign country and in a big company that has an engineering experience gained over decades.
Speaking a different language to discuss technical issues and to communicate during the work is itself a valuable experience for an engineering career.

Bibliography:

- Conférence : Mieux connaître les drones (in Frensh)
- Modeling, Autopilot Design, and Field Tuning of a UAV with Minimum Control Surfaces: PhD thesis by Ming Liu, Greg K. Egan, and Fendy Santoso.
- Modelling Simulation and Control of Fixed-wing UAV: CyberSwan: Master degree's by Jon Bernhard Høstmark.
- Stanford APM: Plane Overview conference by Trent Lukaczyk April 7, 2014
- Autopilot Design Comparison and Flight Experiments for a Small UAV
- By Peter Ortner, Luigi del Re in "Proceedings of the 7th Asian Control Conference, Hong Kong, China, August 27-29, 2009"
- Modeling, Simulation and Control System Design for Civil Unmanned Aerial Vehicle (UAV) Master's Thesis By: Shahriar Bagheri
- Plans without pilot for TARIK Arawi (Book was in Arabic)

Webography:

https://en.wikipedia.org/wiki/Autopilot

https://www.youtube.com/watch?v=aap-U9tGQUE

http://virtualskies.arc.nasa.gov/aeronautics/5.html

www.githob.com >> site web for open source codes

www.missionplanner.com >> official site web for BCG mission planer

http://www.dronetrest.com/t/lipo-batteries-a-guide-to-using-and-looking-after-your-batteries/1278

https://learn.sparkfun.com/tutorials/serial-communication

www.DYIdrone.com

www.ardupilot.com/plane

Appendices:

Appendix 1

The voltage (V)	Paddle size	current (A)	thrust (G)	power (W)	efficiency (G/W)	speed (RPM)	Working temperature (° C)
11	EMAX8045	1	110	11	10.0	3650	
		2	200	22	9.1	4740	
		3	270	33	8.2	5540	
		4	330	44	7.5	6200	
		5	390	55	7.1	6700	
		6	440	66	6.7	7150	
		7.1	490	78.1	6.3	7400	36
	EMAX1045	1	130	11	11.8	2940	
		2	220	22	10.0	3860	
		3	290	33	8.8	4400	
		4	370	44	8.4	4940	
		5	430	55	7.8	5340	
		6	480	66	7.3	5720	
		7	540	77	7.0	5980	
		8	590	88	6.7	6170	
		9	640	99	6.5	6410	
		9.6	670	106	6.3	6530	43

MT.

Motor 239 Thrust table for a 3S Lipo battery

How to calculate the maximum continuous current output for your battery

In order to know what the total current draw of your drone system is, we cab calculated it based on this simple formula:

Max continuous Amp draw (A) = Battery capacity (Ah) x Discharge rate (C)

For an example, we have a 5100mAh 3 cell Lipo battery with a 10C rating. To find the maximum continuous amp draw, we first convert the 5100mAh to 5.1Ah, and multiply that number by 10C, to give a total continuous output of (5.1 x 10) = 51A

How to find the optimum C rating
As choosing the battery is often the last step to building your own drone, we will already know what motors and ESC we are using. Since the motors will draw the most amount of energy from your battery we can base our calculation around this.

The battery C rating depends on the capacity
There is no fixed C rating that you will need to use as the maximum current output of a battery depends on the capacity and C rating. Typically the smaller the capacity of a battery, the higher the C rating needs to be, this is why for many high capacity multi-rotor batteries you will find very low C ratings in the range of 10-15C.

How much capacity do I need?
Now that you know the required current draw from your battery, the capacity and C rating can be found. In general it's best to get the highest possible capacity battery that you can, which still keeping the total weight of your quadcopter including the battery and other equipment at around 50-70% of the maximum motor thrust.

Battery Voltage (cell count)

The battery voltage, or cell count is another important decision that you will need to make. Higher voltage batteries allow your motors to produce more power, however the higher voltage batteries are heavier since they contain more cells.

There is no golden rule to follow when it comes to battery voltage, but the way you can find the best voltage for your drone is to look through your motor thrust data tables and compare the efficiency. You will find that motors are generally more efficient and powerful when using higher cell count lipos (higher voltage), but some of the efficiency bonus is negated by the increase in weight and cost of the battery. So depending on how many motors you are using you will need to choose what is best for your current setup. One thing to bear in mind is to also make sure that your motors/ESC and other electronics are able to support the voltage of your battery. Some motors will only support a specific cell count lipo, or a specific range of voltages which might make the decision easier.

Appendix 2

Data used for ESC chose

Motor type	The voltage (V)	Paddle size	current (A)	thrust (G)	power (W)	efficiency (G/W)	speed (RPM)
MT1806-2280KV	7.4	5030 Carbon Fibre Prop	4.4	210	32.6	6.4	13530
		APC 6*4	6.8	280	50.3	5.6	12030
		5*4.5 three-blade prop	6.2	240	45.9	5.2	12330
	11.1	5030 Carbon Fibre Prop	8	380	88.8	4.3	18510
		APC 6*4	11.3	460	125.4	3.7	15160
		5*4.5 three-blade prop	10.6	410	117.7	3.5	15910

Appendix 3

Data used for motor chose

Table of Thrust :

The voltage (V)	Paddle size	current (A)	thrust (G)	power (W)	efficiency (G/W)	speed (RPM)	Working temperature (° C)
11	EMAX8045	1	110	11	10.0	3650	
		2	200	22	9.1	4740	
		3	270	33	8.2	5540	
		4	330	44	7.5	6200	
		5	390	55	7.1	6700	
		6	440	66	6.7	7150	
		7.1	490	78.1	6.3	7400	36
	EMAX1045	1	130	11	11.8	2940	
		2	220	22	10.0	3860	
		3	290	33	8.8	4400	
		4	370	44	8.4	4940	
		5	430	55	7.8	5340	
		6	480	66	7.3	5720	
		7	540	77	7.0	5980	
		8	590	88	6.7	6170	
		9	640	99	6.5	6410	
		9.6	670	106	6.3	6530	43

MT:

Motor 239 Thrust table for a 3S Lipo battery

kV Rating

The Kv rating of a motor. When a motor is supplied with a voltage, it spins. As the voltage increases, so does the rate of the spin; logical. This rate of spin is quantified using revolutions per minute i.e. rpm, and this Kv rating gives you the rpm a motor will spin at full throttle, given a certain voltage. This value is also assuming that the copter is unloaded. The Kv rating gives you the rpm of the motor in the following way:

$$rpm = Kv \times voltage.$$

Magnet Configuration

The configuration number tells you how many electromagnets there are on the stator, and the number of permanent magnets there are on the rotor. The number before the letter N shows the number of electromagnets there are in the stator. The number before the P shows how many permanent magnets there are in the rotor. Most outrunner brushless motors follow the 12N14P configuration. There are some specialist low KV multirotor motors which have a higher number of electromagnets and permanent magnets which allow the motor to create more torque more efficiently, however due to the added number of magnets these motors are more expensive.

Some other terminology

When investigating and looking into which motors to buy, there are a few terms that are useful to know and recognise. This text takes you through some of these with brief explanations.

Efficiency

The next important feature for anyone considering motors is efficiency. It is intuitive why high efficiency is desired in any motor as they can run for longer, and higher efficiency results in less heat produced. If this is the case, the motors can be pushed harder at any given moment, or over a long period of time.

The most basic and general equation for calculating efficiency is the following:

$$efficiency = \frac{power\ out}{power\ in}.$$

The power in is calculated using the formula V × I where V is voltage and I is current. Intuitively, the power out of the motor is the power put into it, minus the energy that is necessarily lost i.e. power out = power in− power lost. There are two main ways that a motor can lose efficiency; copper loss and iron loss.

Copper Loss. Copper loss is the loss of energy through heat. To calculate this value, we need to incorporate the electrical resistance of the motor, which is referred to as the winding resistance and denoted by Rw. This quantity is measured in Ohms and once we have this value, we can go on to easily calculate the copper loss with

$$copper\ loss = I^2 \times R_w.$$

This is measured in Watts.

Iron Loss. Iron loss is a little less intuitive but nonetheless important. This quantity explains the power lost due to the fluctuating magnetic fields in the motor. To calculate this term, we need to know the amount of current the motor needs to run when it is unloaded, denoted by I0. With this term, we calculate the iron loss as follows

$$Iron\ loss = V \times I_0.$$

Conclusion. Combining these two losses together, we obtain the total power lost, and so we can now calculate the efficiency of a motor. This is given by

$$efficiency = \frac{VI - (I^2 R_w + VI_0)}{VI}.$$

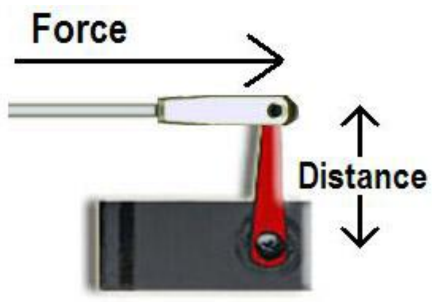
Choosing the right motor

Referring back to the note earlier in the text, choosing the right motor for you depends on what you would like out of your copter. You need to balance efficiency against thrust produced depending on the use. For acrobatic copters, you may need loads of rpm but efficiency may not be at the forefront of your mind. If you're looking for longer flights where your copter can retain stability, you may favor efficiency over thrust. For more information on this please see this post below.

Appendix 4

The majority of servos are powered by 4.8V. Upgrading to a 6.0V receiver battery is a great way to increase the performance of your servos and possibly extend flight times. Make sure you take the added weight of a larger batter into consideration...

Torque



Torque is a measurement of the **strength** of a servo. Torque is determined by multiplying the force acting on the servo arm by the distance from the center of the servo.

The air passing by the airplane is always trying to make the control surfaces move. The servos must "fight" this air resistance in order to move or hold the control surface neutral.

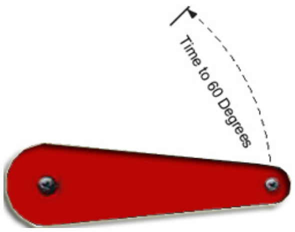
As the size of an airplane increases, the wind resistance acting on the control surfaces increase. This is why larger airplanes need RC servos with more torque than smaller airplanes require.

However, the size of the airplane isn't the only factor that determines how much torque you need. The type of airplane you have as well as your flying style will ultimately determine how much torque is required of the servos.

3-D airplanes have very large control surfaces which require much more torque than a non 3D sports planes of the same size. Also, if your flights tend to be gentle and laid back, you can by with less torque than someone who likes to jerk the sticks around to perform crazy violent maneuvers.

Servo Speed

The speed of an RC servo is defined as the minimum amount of time it takes the servo arm to rotate 60 degrees at maximum rated torque.



Obviously if the servo is being used on a control surface, you want the speed to be as fast as possible. If the RC servo is used on the landing gear, you'll probably want the speed to be a bit slower. Don’t be deceived by the speed rating. The "top speed" of a servo isn't an accurate measure of how quickly the servo responds to your stick movements.

Speed is a very important specification, but there are other factors that we will discuss below that will ultimately determine how quickly and precisely the servo responds to your input.

Servo Dimensions and Weight

The dimensions and weight of a servo is directly proportion to the torque requirements. Micro RC Airplanes use a small servo whereas Giant Scale Airplanes require very large servos. This is fairly self-evident and is more or less common sense.

There are RC servos specifically designed for just about any size RC aircraft out there. The first step is to find a servo that meets your torque requirements, then make sure the weight and dimensions are suitable for your airplane.

RC Servo Bearings

The shaft of an RC servo is supported by either a ball bearing or a bushing. A servo with bearings will operate much more smoothly and will last much longer than a bushed servo.

Most standard size servos these days will have either 1 or 2 bearings on the output shaft where it exits the servo case. Many micro servos do not have bearings because the loads are too small to justify the added weight of bearings. In contrast, most large scale servos need two bearings to support the large loads of the control surfaces.

Kits are available that supply the ball bearings and case tops in order to replace the bushing with bearings. These kits provide a very economical way to upgrade your servos without having to replace them.

Gears inside RC Servos

The motors inside of our servos turn much faster than we would ever need the servo to move. For that reason, every servo has a set of gears that reduces the speed of the output shaft. As with any gear reducer, this also causes the torque of the output shaft to be much higher than the torque of the motor.

Most servo gears are made from a type of plastic called nylon. Sometimes under excess torque these plastic gears will strip out. If you're lucky, you will get some warning signs before having a catastrophic failure that destroys your airplane. But there have been a many of crashes due to stripped servo gears.

Many large scale or 3D pilots will buy RC servos with metal gears. These are bit more expensive and they weigh a little more. But they will withstand a lot more torque than standard plastic gears. The only downside, besides the added weight, is that metals gears will wear over time and this leads to slop in the servo. This wear happens slowly and gives you plenty of warning, as opposed to set of nylon gears that has the potential of stripping any time it is overloaded.

Replacement gears sets are readily available for both metal and plastic. So when you destroy the gears in your servo, it is cheaper to replace the gears rather than buy a new servo.

Digital vs Analogue RC Servos

The physical working portion of a digital servo is exactly the same as a standard RC servo. Both have the same plug and both can be used by a standard receiver. The only difference is the digital servo has a microprocessor that modifies the frequency of the control pulses to the motor.

If you're not familiar with how a standard servo operates, you may want to read in order to have a better understanding of the following discussion.

Let’s say you only want to move the servo a *tiny* bit, so you *barely* move the transmitter stick off center. A standard servo will receive a series of extremely short full voltage pulses 50 times a second. These very short length instantaneous voltage pulses occur so infrequently that the motor doesn't even react. The amount of transmitter stick movement from center position that is required to make a servo start moving is called the servo's **dead band.**

If you're flying a trainer or sports RC airplane, the dead band isn't a huge deal. But if you're a helicopter or 3D pilot you need every bit of the stick movement to count. Too much dead band could lead to a dead plane!

A digital servo's microprocessor will convert the signal to shorter length pulses at a frequency of 300 times per second as opposed to 50 times per second of the original PPM signal. The motor is able to respond to smaller stick movements because it is receiving those very short pulses 6 times more often. This also gives a digital servo much stronger holding power because it reacts much more quickly when pushed out of position.

Digital servos react to your stick movements much more accurately than standard RC servos. You can think of this like the frame rate of a video. A video will appear much smoother if the frames are shorter and appear more often, as opposed to longer frames that occur less often. This same logic can be applied for understanding why digital servos give you much more precise control over your RC airplane.

Motors That Drive Servos

If you've been shopping around for a servo you have probably noticed the different types of servo motors available. You will see 3P, 5p, coreless and brushless. So which one should you choose?

3P and 5P

Most RC servos on the market are either 3 pole **(3p)** or 5 pole **(5p)** motors. All DC motors have a set of permanent magnets that the electromagnets **(windings)** are attracted to. These permanent magnets are called poles.

The torque of the motor can be reduced if the electromagnets happen to be precisely in between two poles when the servo is trying to hold position. In other words, 5P motors will have more holding torque with smoother operation than 3P motors. If you're a beginner, there probably isn't enough difference that you would even be able to tell.

Coreless Motors

In a conventional motor, windings are wrapped around a metal core to form the electromagnets. A coreless motor does not have a core. The windings consist of a wire mesh that rotates around the outside of the permanent magnets.

This eliminates the problem with reduced torque at certain positions as discussed above with standard "pole" motors. A coreless motor will also respond much quicker to stick inputs because it doesn't have to overcome the momentum of a metal core when changing directions.

Brushless Motors

Futaba has recently developed a brushless servo motor. So you get all the benefits of a brushless motor incorporated into your servo. This translates into longer servo life, faster response time, smoother operation and resistance to vibrations.

Servo Plugs and Wires

All RC servos will have three wires. One wire is positive lead **(+)** from the battery. One wire the ground **(-)** from the battery. The third wire is the control input from the receiver.

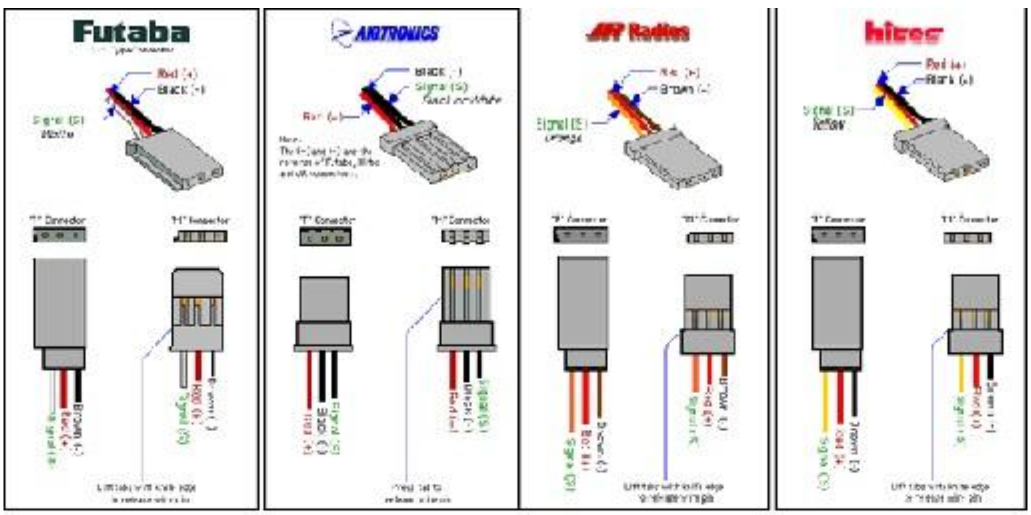
It is important to note that you can't just swap the pos.**(+)** and neg.**(-)** to reverse the servo. Doing so may damage your servo and receiver. You can buy connectors that act as a servo reverser by reversing the control signal. A **Y connector** is commonly used with two ailerons servos that share the same receiver output. The **Y connector** reverses the signal to one servo so that the two aileron servos act opposite of one other.

Futaba's "J" connector is a plug with a little plastic tab, or key, that prevents you from plugging it in backwards.

Futaba's J plugs, Airtronic's Z plugs, Hitec's S plugs, Hobbico's U plugs, and JR's connectors are all compatible as long as you make sure the wires are in the correct order within the plug. If you want to plug a male Futaba plug into another brand you will have to shave the safety tab off.

If your servo is a different brand than your receiver, it is critical that you have the appropriate wires lined up before plugging the connectors together. The wires can easily be popped out of the connector for rearrangement.

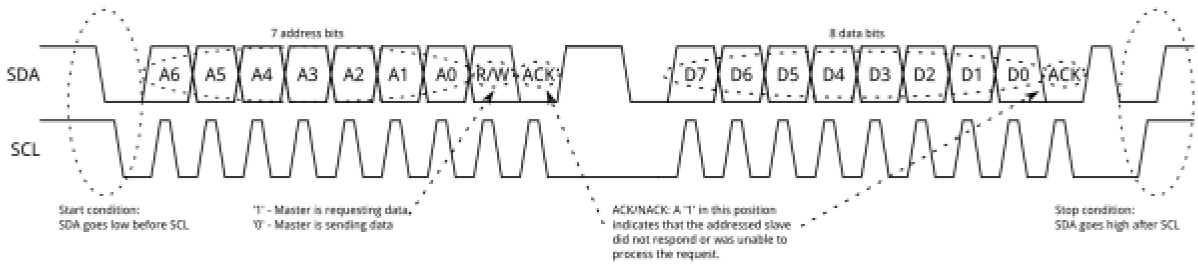
The images below show you the wiring schematic for the plugs of the four major servo manufacturers.



Appendix 5

Communication via I^2C is more complex than with a UART or SPI solution. The signaling must adhere to a certain protocol for the devices on the bus to recognize it as valid I^2C communications. Fortunately, most devices take care of all the fiddly details for you, allowing you to concentrate on the data you wish to exchange.

Basics



Messages are broken up into two types of frame: an address frame, where the master indicates the slave to which the message is being sent, and one or more data frames, which are 8-bit data messages passed from master to slave or vice versa. Data is placed on the SDA line after SCL goes low, and is sampled after the SCL line goes high. The time between clock edge and data read/write is defined by the devices on the bus and will vary from chip to chip.

Start Condition

To initiate the address frame, the master device leaves SCL high and pulls SDA low. This puts all slave devices on notice that a transmission is about to start. If two master devices wish to take ownership of the bus at one time, whichever device pulls SDA low first wins the race and gains control of the bus. It is possible to issue repeated starts, initiating a new communication sequence without relinquishing control of the bus to other masters.

Address Frame

The address frame is always first in any new communication sequence. For a 7-bit address, the address is clocked out most significant bit (MSB) first, followed by an R/W bit indicating whether this is a read (1) or write (0) operation.

The 9th bit of the frame is the NACK/ACK bit. This is the case for all frames (data or address). Once the first 8 bits of the frame are sent, the receiving device is given control over SDA. If the receiving device does not pull the SDA line low before the 9th clock pulse, it can be inferred that the receiving device either did not receive the data or did not know how to parse the message. In that case, the exchange halts, and it’s up to the master of the system to decide how to proceed.

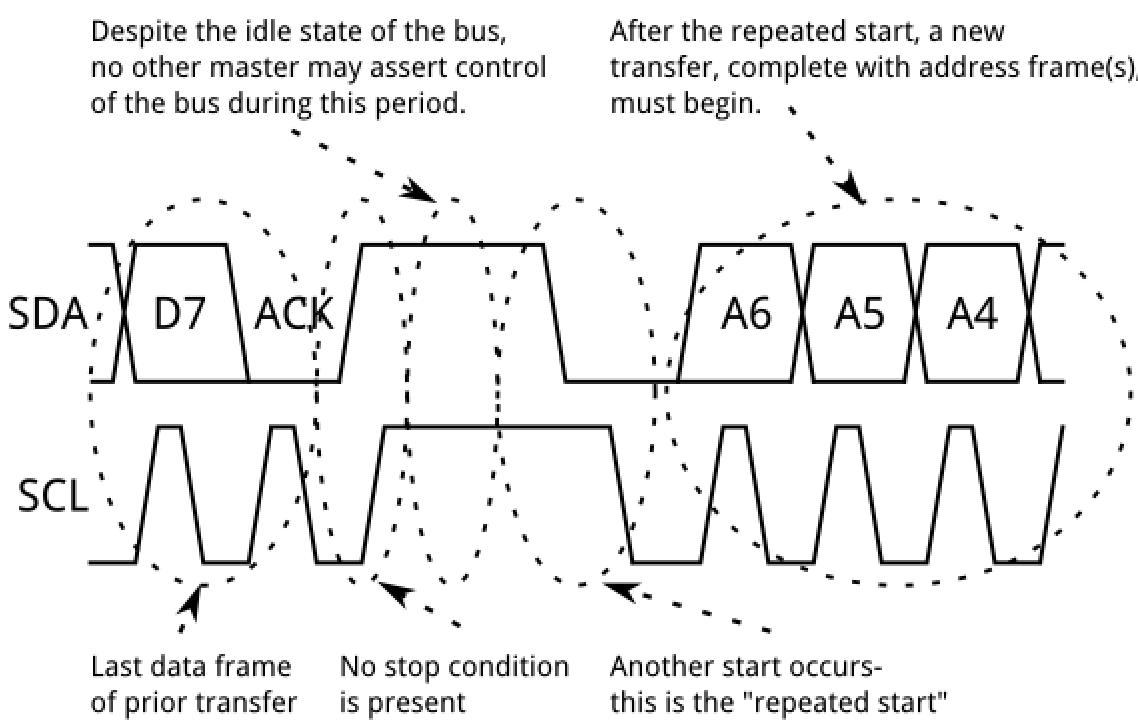
Data Frames

After the address frame has been sent, data can begin being transmitted. The master will simply continue generating clock pulses at a regular interval, and the data will be placed on SDA by either the master or the slave, depending on whether the R/W bit indicated a read or write operation. The number of data frames is arbitrary, and most slave devices will auto-increment the internal register, meaning that subsequent reads or writes will come from the next register in line.

Stop condition

Once all the data frames have been sent, the master will generate a stop condition. Stop conditions are defined by a 0->1 (low to high) transition on SDA *after* a 0->1 transition on SCL, with SCL remaining high. During normal data writing operation, the value on SDA should **not** change when SCL is high, to avoid false stop conditions.

Repeated Start Conditions

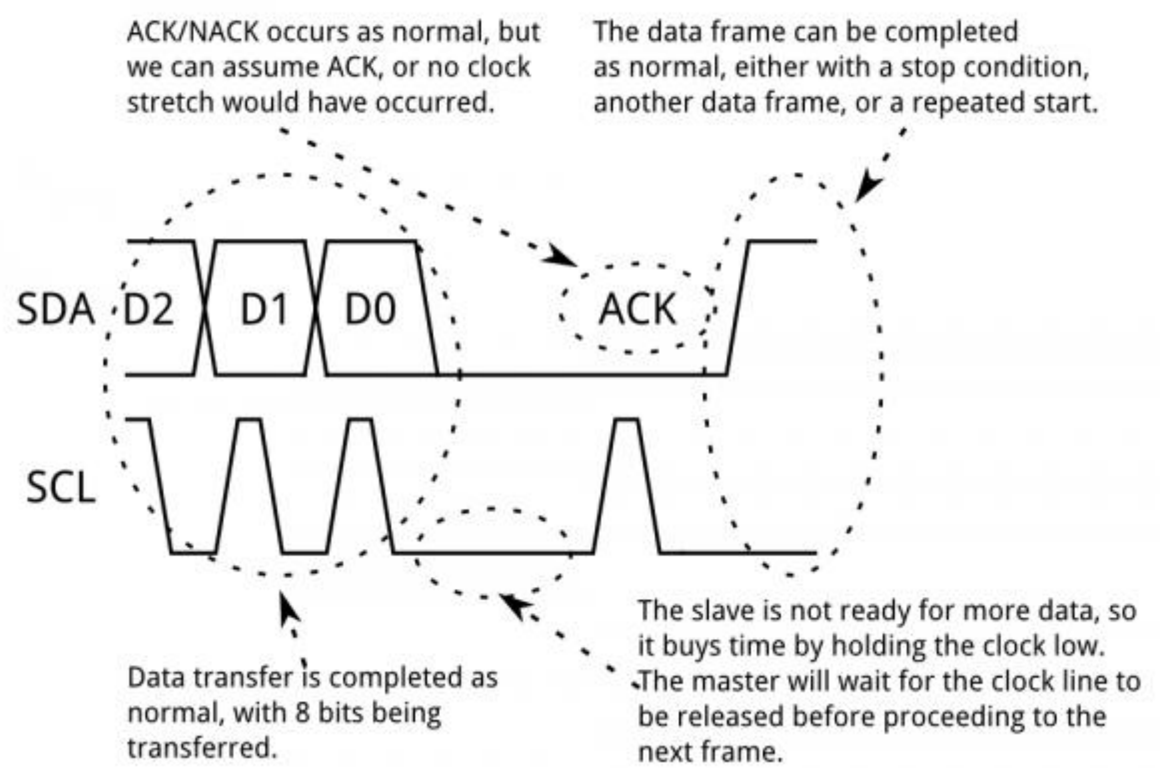


Sometimes, it is important that a master device be allowed to exchange several messages in one go, without allowing other master devices on the bus to interfere. For this reason, the repeated start condition has been defined.

To perform a repeated start, SDA is allowed to go high while SCL is low, SCL is allowed to go high, and then SDA is brought low again while SCL is high. Because there was no stop condition on the bus, the previous communication wasn't truly completed and the current master maintains control of the bus.

At this point, the next message can begin transmission. The syntax of this new message is the same as any other message–an address frame followed by data frames. Any number of repeated starts is allowed, and the master will maintain control of the bus until it issues a stop condition.

Clock stretching



At times, the master's data rate will exceed the slave's ability to provide that data. This can be because the data isn't ready yet (for instance, the slave hasn't completed an analog-to-digital conversion yet) or because a previous operation hasn't yet completed (say, an EEPROM which hasn't completed writing to non-volatile memory yet and needs to finish that before it can service other requests).

In this case, some slave devices will execute what is referred to as "clock stretching". Nominally, **all** clocking is driven by the master device–slaves simply put data on the bus or take data off the bus in response to the master's clock pulses. At any point in the data transfer process, an addressed slave can hold the SCL line low after the master releases it. The master is required to refrain from additional clock pulses or data transfer until such time as the slave releases the SCL line.

MavLink protocol

MAVLINK_MSG_ID (Block of main messages):

1) MAVLINK_MSG_ID_HEARTBEAT: //0 a. This is the most important message. The GCS keeps sending a message to APM/PX4 to find out whether it is connected to it (every 1 second). This is to make sure the MP is in sync with APM when you update some parameters. If a number of heartbeats are missed, a failsafe (can be) is triggered and copter lands, continues the mission or Returns to launch (also called, RTL). The failsafe option can be enabled/ disabled in MP under Configure/Setup Failsafe options. But you can't stop heartbeats, can you? The name is very justified!!

2) MAVLINK_MSG_ID_REQUEST_DATA_STREAM: //66

a. Sensors, RC channels, GPS position, status, Extra 1/2/3

3) MAVLINK_MSG_ID_COMMAND_LONG: // 76 a. Loiter unlimited, RTL, Land, Mission start, Arm/Disarm, Reboot

4) SET_MODE: //11

a. E.g. set_mode(packet.custom_mode)

5) MAVLINK_MSG_ID_MISSION_REQUEST_LIST: //43

a. Total waypoints: command_total variable of parameters. This stores the total number of waypoints that are present (except home location, for multi-copters)

6) MAVLINK_MSG_ID_MISSION_REQUEST: //40

a. Set of MAV_CMD value enum members, such as: (MAV_CMD_)CHANGE_ALT, SET_HOME, CONDITION_YAW, TAKE_OFF, NAV_LOITER_TIME

7) MAVLINK_MSG_ID_MISSION_ACK: //47

a. // turn off waypoint send

8) MAVLINK_MSG_ID_PARAM_REQUEST_LIST: //21

a. count_parameters (Count the total parameters)

9) MAVLINK_MSG_ID_PARAM_REQUEST_READ: //20

a. Receive and decode parameters (Make sense of Param name and Id)

10) MAVLINK_MSG_ID_MISSION_CLEAR_ALL: //45

a. When you use mission planner flight data screen and say, Clear Mission with the mouse menu, this is where it goes. It clears the EEPROM memory from the APM/PX4. 11) MAVLINK_MSG_ID_MISSION_SET_CURRENT: //41

a. This is used to change active command during mid mission. E.g. when you click on MP Google map screen and click ‘Fly To Here’, as an example.

12) MAVLINK_MSG_ID_MISSION_COUNT: // 44

a. Save the total number of waypoints (excluding home location) -> for Multicopters.

13) MAVLINK_MSG_ID_MISSION_WRITE_PARTIAL_LIST: //

a. Just keeping a global variable stating that APM is receiving commands now. This is to avoid other MavLink actions while important parameters are being set.

14) MAVLINK_MSG_ID_SET_MAG_OFFSETS: //151

a. Set the mag_ofs_x, mag_ofs_y, mag_ofs_z, say after compass calibration to EEPROM of APM/PX4. Mission Planner (MP) automatically does this or you can do this too by going to Full Parameters List under SOFTWARE CONFIGURATION.

15) MAVLINK_MSG_ID_MISSION_ITEM: //39

This is an interesting part. This message contains sub messages for taking real-time action. Like setting waypoints and advanced features... This is how it works, as below: a. Receive a Waypoint (WP) from GCS and store in EEPROM of APM/PX4. b. Sends 4 params (e.g. Delay, HitRad, -, and Yaw Angle) for LOITER_TIME (as ID) + (Lat, Long, Alt: Defines the 3D position of object in space). These parameters are defined as an Enum in code + Options (1 = Store Altitude (Alt) relative to home altitude). Each Command (or ID) may have different parameters of interest. Mission Planner shows ‘blank’ header for column, because there is no parameter defined for this ID. As a summary, the interesting parameters sent with every action are: 4 params + ID (of action) + (Lat, Long, Alt) defines 3D position of copter. Note the 4 params can be some sort of custom action as for camera setting, camera trigger, Loiter time, etc. c. As in figure below from Mission Planner, each ID defines a waypoint (AFAIK). LOITER_TIME, LOITER_UNLIMITED, WAYPOINT all are waypoints that send along with other parameters (LATITUDE, LONGITUDE and ALTITUDE) because each is saved as a waypoint in APM/PX4. Period. d. Keep in mind: ALTITUDE is always relative to home altitude (Always!) with the current design. e. You can define ‘these actions’ in Common.xml and use Python GUI generator to generate the code that APM/PX4 will use. Let me cover this later or ask me in forum on how to. You can add your own interesting parameters for the (4 params), I mentioned. f. When APM receives this ‘main’ command (MAVLINK_MSG_ID_MISSION_ITEM), it reads the ID from the MavLink packet and performs switch (case) on the ID. E.g. i. Loiter turns, Set home, Loiter time, Repeat servo, Set servo, etc. 16) MAVLINK_MSG_ID_PARAM_SET: //23

a. Set the parameter. Remember, we can set a value to a parameter in the Mission Planner (say, 'Full Parameter List'); this is where it goes when we do this. APM sends the data value set for confirmation. Have you seen MP, saying 'value set failed'? Also, APM/PX4 logs simultaneously, this value, for our analyses offline.

17) MAVLINK_MSG_ID_RC_CHANNELS_OVERRIDE: //70

a. Override RC channel values for HIL (Hardware-In-Loop-Simulation) OR complete GCS control of the switch position via GUI (I have not tried this, though)!

18) MAVLINK_MSG_ID_HIL_STATE: //90

a. Used for HIL simulation. It is a virtual reality with your copter/ plane.

19) MAVLINK_MSG_ID_DIGICAM_CONFIGURE: //

20) MAVLINK_MSG_ID_MOUNT_CONFIGURE: //

21) MAVLINK_MSG_ID_MOUNT_CONTROL: //

22) MAVLINK_MSG_ID_MOUNT_STATUS:// a. So far, as the name suggests, configures the appropriate command settings as set by the user.

23) MAVLINK_MSG_ID_RADIO, MAVLINK_MSG_ID_RADIO_STATUS: //

a. Studies the packet rate of the telemetry/ USB and auto adjusts the delay between sending receiving packets if the signal strength is lower than expected or errors are getting higher. It is like an adaptive software flow control. Look at __mavlink_radio_t in C++ (APM code) OR mavlink_radio_t (C#, Mission Planner) code. Both are defined as structs.

Appendix 6

Plan maker steps:

1. Decide on a design.

2. Create the fuselage, wings, and tail of the aircraft

3. Create secondary objects, such as landing gears and engine nacelles.

4. Set up the systems and internal properties, including the engines, electrical systems, weight and balance, and viewpoints.

5. Set up any additional features of the aircraft, such as added weapons or special controls.

6. Create a 2-D instrument panel.

7. Test-fly the aircraft in X-Plane and fine-tune the features of the aircraft from steps 1-6 as needed.

8. Add textures, 3-D objects, extra liveries, etc.