PEOPLE'S DEMOCRATIC REPUBLIC OF ALGERIA

MINISTRY OF HIGHER EDUCATION AND SCIENTIFIC RESEARCH

University of Mohamed El-Bachir El-Ibrahimi – Bordj Bou Arreridj

Faculty of Science and Technology

Departement of Electromechanics

Discuss the graduation thesis in order to obtain the degree of MASTER

Fields: Automatic and Industrial computer science

Subject:

Design and Implementation of Firefighting Drones

Presented by: **Mebarki Brahim & Hichem Bendjbel & Abadou Walid**

Jury members:

First name & Last name	*Grade*	*Quality*	*Establishment*
Mr.Toufik Madani Layadi	**MCA**	**Advisor**	***BBA University***
Mr.Issam Meghlaoui	**MCB**	**Co-advisor**	***BBA University***
Pr.Moustfai Messaoud	**PR**	**President**	***BBA University***
Mrs.Djamila Zehar	**MCB**	**Examiner**	***BBA University***

College year 2023/2024

Thanks

*First of all, we would like to thank **ALLAH** who has empowered us from the first year of primary school to this day of graduation, while asking him to bring us more to continue.*

The most distinguished thanks go to our supervisor Dr. LAYADI

who enabled us to carry out this project and supported us with his advices and guidance.

A special thanks to Dr.KHANFER for every support he has given to us.

Special thanks to the scientific research laboratory of the University of Bordj Bou Arreridj and all those responsibles for it especially Dr. MOUSTFAI.

Also a special thanks to Mrs. REFFAD AICHA an MCA at Setif university.

We would also like to take this opportunity to thank our teachers who contributed to our training, and all the staff at the Bordj Bou Arreridj University

We would like to thank the members of the jury who agreed to do us the

honor of judging our work.

Dedication

I dedicate this modest work to :

To my dear mother.

To my dear father.

To my grandmother.

To all my family.

To all my friends.

To all my dearest colleague

For everyone who knows BRAHIM.

To the entire class of Automatic.

MEBARKI BRAHIM

Dedication

I dedicate this modest work to :

To the soul of my dear mother.(Allah rest her soul)

To my father.

To all my borthers and sisters.

To all my friends.

To all my dearest colleague

To the entire class of Automatic.

ABADOU WALID

Dedication

I dedicate this modest work to :

To my mother.

To my father.

To all my family.

To all my friends.

To all my dearest colleague

To everyone helps us to do this work.

To the entire class of Automatic.

BENDJBEL HICHAM

TABLE OF CONTENTS

Abstract

Drones have recently occupied an important place in human daily life. These Drones are characterized by advanced features, depending on demand. They can provide multiple services for making human life easy. Therefore, these systems can be used in public places such as food delivery and postal services. These systems also allow for the completion of difficult tasks such as military applications. These aircraft can be controlled at a distance of up to 42 km via radio communications. In general, electric motors are used in aircraft, and these systems also use lithium-polymer batteries and electronic speed regulators for power.

The system developed in this work is a multi-model drone and are designed to extinguish fires in buildings. Each model consists of two basic parts. The first part represents the aircraft structure with its parts, which is a mixture of materials. The second part represents the controler and electric motors. The cockpit designed allows to control the three different models to be controlled at the same time. After designing the models, applied experiments were carried out in front of the laboratory. As for future work, integration of new control technologies into aircraft is suggested.

Résumé

Les systèmes d'avions sans pilote ont récemment accédé à une place importante dans la vie quotidienne. Les utilisations les plus fondamentales aux plus délicates de ces systèmes sont disponibles. Ces systèmes englobent une large gamme de services destinés à faciliter la vie des personnes. Par conséquent, ils peuvent être employés dans des fonctions gouvernementales. Ces systèmes permettent d'accomplir des tâches sophistiquées et difficiles dans des applications militaires. Les avions sans pilote peuvent être pilotés à différentes distances via des communications sans fil. Généralement, les moteurs électriques sans balais sont utilisés dans les petits drones. Des batteries au lithium-polymère et des régulateurs de vitesse électroniques assurent son alimentation. Augmenter l'autonomie de ce type de drones présente une difficulté. La tâche actuelle consiste à développer et à mettre en œuvre un drone quadrirotor, un VTOL et un avion d'entraînement RC. Un système de radiocommande contrôlant chacun de ces véhicules séparément. Avant la réalisation, de nombreuses études, simulations et tests ont été réalisés. Comme des travaux de futur des techniques modèrnes de commande-contrle ont été envisagées.

ملخص

احتلت أنظمة الطائرات بدون طيار مؤخرًا مكانًا مهمًا في الحياة اليومية للإنسان. تتميزهذه الانظمة بخصائص متطورة و ذلك حسب الطلب. تستطيع هذه الأنظمة بتوفير خدمات متعددة مما يجعل حياة الإنسان سهلة. ولذلك، يمكن استخدام هذه الأنظمة في الأماكن العامة مثل توصيل الطعام والخدمات البريدية. كما تسمح هذه الأنظمة بإنجاز المهام الصعبة مثل التطبيقات العسكرية. ويمكن التحكم في هذه الطائرات بمسافة تصل الى 42 كلم و ذلك عبر الاتصالات اللاسلكية. بشكل عام، يتم استخدام المحركات الكهربائية في الطائرات، كما تستعمل هذه الأنظمة بطاريات من نوع الليثيوم بوليمر ومنظمات السرعة الإلكترونية للطاقة.

النظام الذي تم تطويره في هذا العمل هو عبارة عن طائرة بدون طيار بعدة نماذج موجهة لإطفاء الحرائق في البنايات مع قمرة قيادة. كل نموذج يتكون من جزئين أساسيين. الجزء الأول يمثل هيكل الطائرة بأجزائه والذي هو عبارة مزيج من المواد. أما الجزء الثاني فيمثل الميكرومعالج و محركات كهربائية. قمرة القيادة المصممة تسمح بالتحكم في ثلاث نماذج في نفس الوقت. بعد تصميم النماذج تم القيام بتجارب تطبيقية أمام المخبر. أما فيما يخص العمل المستقبلي تم اقتراح دمج تقنيات تحكم في الطائرات.

LISTE OF FIGURES

LISTE OF TABLES

ACRONYMS

UAV	**U**nmanned **A**erial **V**ehicle
VTOL	**V**ertical **t**ake-**o**ff and **L**anding
RC	Radio Control
PID	Proportional, Integral, Derivative
HMI	Human-Machine Interface
HALE	**H**igh-**A**ltitude **L**ong-**E**ndurance
MALE	**M**edium-**A**ltitude **L**ong-**E**ndurance
MUAV	**M**ini **U**nmanned **A**erial **V**ehicle
TUAV	**T**actical **U**nmanned **A**erial **V**ehicle
UCAV	Unmanned Combat Aerial Vehicle
DOF	**D**egrees **O**f **F**reedom
APM	**A**rdu**P**ilot **M**ega
DC	**D**irect **C**urrent
ESC	**E**lectronic **S**peed **C**ontroller
GPS	**G**lobal **P**ositioning **S**ystem
IDE	**I**ntegrated **D**evelopment **E**nvironment
Li-Po	**L**i-**P**olymer
MEMS	**M**icro **E**lectronic **M**echanical **S**ystems

PWM	**P**ulse **W**idth **M**odulation
RCS	**R**adio **C**ontrol **S**ystem
SPI	**S**erial **P**eripheral **I**nterface
USB	**U**niversal **S**erial **B**us

General Introduction

Due to many benefits, drones have known large interest from the robotics community over the past ten years. Nowadays, robotics becomes necessary to realize dificult tasks in different filelds .

A vertical take-off and landing aerial vehicle known as a quadcopter has gained a lot of attention recently and is now a popular study topic for numerous teams and labs. Four propellers are employed in these types of vehicles for lifting and propulsion. By creating new parts for the quad copter design, especially the flight controller, which acted as the quad-copter's brain, it significantly helped to simplify the setup. This project aims to create a drone that incorporates several control techniques.

This work is organized as follow:

In the first chapter, drone technology is presented. Also numerous definitions, terms, and concepts relevant linked to the subject are given.

The second chapter concerns the modeling and control of the quad-copter. A case study model that we created enables the analysis of the dynamic evolution of angle, position, rotational speed, and four-wheel drive translation. Then , implementation of PID controller according to an arduino programming code is performed.

The third chapter details the steps of the mechanical design of a quadri-rotor, VTOL, RC Trainer plane, and how we built the radio commande station including the HMI interface, and the configuration and calibration of the autopilot board. The outcomes of simulations, the drone's practical implementation, and viewpoints are provided. Finally, conclustion and perspectives are given.

CHAPTER 1

STATE OF THE ART ABOUT DRONES

1 CHAPTER 1 : STATE OF THE ART ABOUT DRONES

1.1 Introduction:

The miniaturization of onboard sensors has sparked interest in automating aircraft capable of vertical takeoff and landing. These drones are capable of carrying out tasks more or less autonomously. As such, they are ideal potential solutions for indoor or outdoor applications where there is currently a risk to the safety of people or equipment, such as the inspection of works of art such as bridges or power lines high voltage, exploration of dangerous environments, etc..

In this chapter, we are going to present a general overview of drones, including their definition and historical development. Then we will give classifications of UAVs, their advantages and disadvantages. Also, we will give a description of the realized drone.

1.2 Drone definition

A drone or UAV (Unmanned Aerial Vehicle) is an aircraft without a human pilot on board that uses aerodynamic forces to produce vertical flight. It can be controlled remotely, autonomously or semi-autonomously. It carries a payload, making it capable of carrying out specific tasks, for a flight duration which may vary depending on its capabilities[1].

The size and mass depend on the operational capabilities sought. Automatic piloting or piloting from the ground makes it possible to envisage very long flights, several dozen hours.

1.3 Drone Classifications

The classification of drones depends on the country. However, drones can be classified into several categories according to altitude, durability which is the time the aircraft can spend in flight, size or even their wing. In this context, drones can be classified into three families, which are given in the following subsections (Sallah & Babou, 2018).

1.3.1 Fixed wings UAVs :

UAVs in this family consist of a pair of wings (figure 1.1) providing lift, with propulsion provided by one or more propellers.

This family comprises the following categories:

- **High-altitude, long-endurance (HALE) drones**: These are large drones, most often with fixed wings. They are characterized by their very long flight times and collect information over very long periods (12 to 48 hours) at high altitude.It can fly at altitudes of up to 20 km for a range of several thousand kilometers[3].
- **Medium-altitude long-endurance (MALE) drones**: Characterized by their long flight times at medium operational altitude, having great autonomy, it can fly at altitudes between 5 km and 12 km for a range of up to 1000 km [3].
- **UCAV combat drones**: These are equipped with weapons or intelligence-gathering systems, and are designed for reconnaissance, attack and fire missions. They can carry a lethal payload.
- **TUAV tactical UAVs**: Which can fly at altitudes of 200 to 5,000 meters, with a range of around ten hours. They can be classified into two categories:
- **MAV micro-UAVs**: Less than 15 cm in size, weighing from a few dozen to a few hundred grams, they can fly up to ten kilometers for twenty minutes or so, and perform tasks that larger UAVs cannot. they are generally equipped with propellers driven by electric motors. by electric motors.
- **MAV mini-UAVs**: They are about one meter in size and can fly up to fly up to a ceiling of 300 meters, with a range of a few hours and a very light a very light payload [4].

Figure 1-1 Fixed-wing UAVs

1.3.2 Rotary-wing UAVs:

This family of UAVs is characterized by vertical take-off and landing, capable of hovering at low speed and low altitude, and therefore not requiring a take-off or landing runway. They use one or more rotors, as shown in figure 1.2, and can carry out missions that fixed-wing vehicles cannot. This family encompasses several categories:

- **Mono rotor**: consists of a main rotor with a stabilizer bar for lift and propulsion.
- **Birotors**: Two rotors rotating in opposite directions at the same speed, providing lift and translation.
- **Quadri rotors**: consisting of a symmetrical cross-shaped frame with a motor at the end of each arm.
- **Convertibles**: these are machines fitted with a rotor tilting mechanism that enables them to hover, take off and land vertically in restricted and difficult areas. The main drawback of these machines is their instability during the transition phase between flight in airplane mode and helicopter mode.

Figure 1-2 Rotaty wing UAVs

1.4 Type of control:

UAVs are generally piloted from a distance in three different ways:

- **Visual piloting**: horizontal distance of aircraft less than 100 meters from the pilot, who retains a direct view of the aircraft.
- **Out-of-sight piloting**: distance greater than 100 meters, using video feedback to guide the aircraft.
- **Autopilot**: pre-flight recording of aircraft navigation parameters. which evolves according to these predefined parameters.

1.5 The use of drones

The use of drones was first known in military applications, such as surveillance and reconnaissance and as a target designation platform or weapon. Then, several civil applications became competing, notably in the observation of natural phenomena (Avalanches, volcanoes, etc.), the spraying of pesticides on agricultural areas, environmental monitoring (example: pollution measurements) and networks roads, infrastructure maintenance, etc...[2].

1.5.1 Military use of drones

Drones have been used in the military since the Second World War for observation, intelligence, terrain reconnaissance for ground and air troops, and as a combat weapon. A drone can be piloted from locations thousands of kilometers away from the aircraft.

Figure 1-3 Military use of drones

1.5.2 Civilian use of drones

The transfer of UAVs to the civil sector has opened up a significant number of civil applications, (as shown in figure 1.4), opening up considerable growth potential in a variety of innovative sectors, such as:

- Study of the atmosphere, soils (geology) and oceans.
- Forest fires and avalanches.
- Crop monitoring and agricultural spraying.
- Search and rescue (sea, mountains, desert...).
- Dropping supplies and rescue equipment in hostile areas.
- Surveillance of road traffic and transport of hazardous materials.

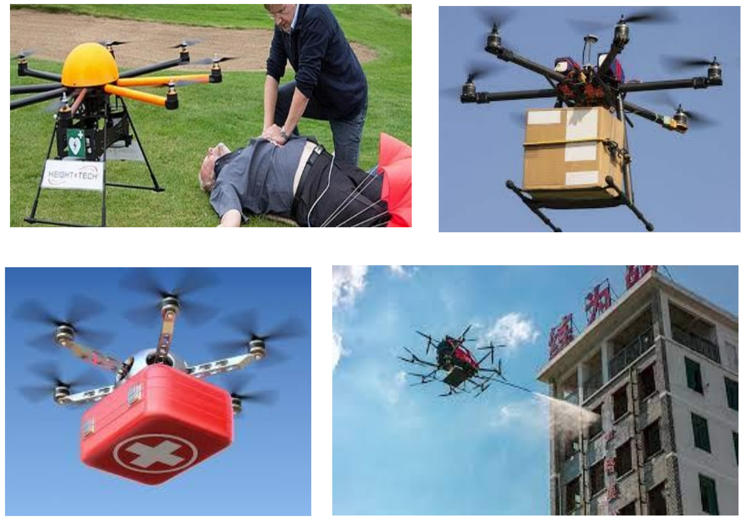


Figure 1-4 Civilian use of drones

1.6 The advantages and disadvantages of drones

Table 1.1 summarizes the advantages and disadvantages of each drone family.

Table 1.1 Advantages and disadvantages of each drone family**.**

Family	Advanteges	Disadvantages
Fixed-wing UAVs	- Ability to carry more weight - Ability to fly with greater range and altitude	- No hovering - No low-level or low-altitude flying - Take-off and landing runway required
Rotary-wing UAVs	- Allows vertical take-off and landing - - Allows low-speed, low-altitude and low-altitude flight - - Allows maneuvering in the air. - - More stability and to control.	- Complex maintenance and repair. - Less range, speed and altitude in flight.

1.7 Regulations concerning the use of drones

The use of an unpiloted aircraft is subject to a number of regulations that are set out in articles of law such as the orders of 11 April 2012, the orders of 2016 and those of 2017, and which differ from one country to another. By way of example, here are the essential rules laid down by the French DGAC(Direction Générale de L'aviation Civile), which must be complied with when using a civil drone (Sallah & Babou, 2018):

- The use of a drone must respect the safety of people.
- It is forbidden to fly a drone just above a person because the drone's propellers are dangerous and they can injure.
- A drone must not reach a height of more than 150m.
- Never fly over a drone at night, and the person flying the drone must never lose sight of the aircraft.
- It is forbidden to fly a drone over public areas in built-up areas, near airfields and over sensitive sites such as military zones, nuclear power stations, electricity distribution boxes, railway lines, etc.
- Permission must always be sought from the people in the shots before they can be used, and it is forbidden to distribute the shots for commercial purposes.

Other new rules will come into force concerning the obligation to declare drones weighing more than 800 grams and the need for them to be equipped with light and sound signals so that they can be easily identified in the air.

The FAA also introduced new regulations in the United States on 21 June 2016 concerning the categories of civil drones and, in particular, those weighing less than 25 kg and flying at an altitude of less than 120 metres:

- It is forbidden to exceed altitudes of 122 metres and speeds of 160 Km/h.

- Drone users must be over 16 years of age and pass a 3-hour test at an approved centre.

- The drone must be registered with the FAA before it can take off.

- It is forbidden to fly at night, but it is possible to fly 30 minutes before sunrise and 30 minutes after sunset.

It is forbidden to fly from an aircraft, but it is possible to do so from a moving car.

- It is forbidden to attach a payload to the drone that would reduce its capacity.

1.8 Conclusion

Drones have a lot of fields of use, so there are rules and regulations to use them. The realization of this kind of project mainly requires a modeling of its dynamics to study the different movements before implementation phase.

In the next chapter we will present the dynamic modeling of a quadrotor type UAV system, as well as the corrector used to eliminate the disturbances is used.

CHAPTER 2

MODELING AND CONTROL OF UAVs

2 CHAPTER 2 : MODELING AND CONTROL OF UAVS

2.1 Introduction :

The first step to develop a system of control is to establish its mathematical model. This model must be as faithful as possible to the dynamics of the system to be controlled. The system we are trying to control here is a quad-rotor UAV. It is supposed as a complex system because of the number of physical effects that affect it. In this chapter, we will concentrate on the modeling of quad-rotor UAVs.

2.2 Quadrorotor functioning

The quadrotor has four rotors which are defined in space by six degrees of freedom (DOF), three rotational movements and three translational movements. The four rotors are generally placed at the ends of a cross, and the control electronics are generally placed in the center of the cross. Opposite propellers rotate in one direction, while the other two rotate in the opposite direction to prevent the aircraft from spinning on its axis. [5].

The quadrotor is an under-actuated system (the number of actuators is less than the number of DOFs to be achieved) and its operation is quite specific. By cleverly varying the rotation speeds of the motors, it can be raised/lowered, tilted left/right (Roll) or forwards/backwards (Pitch) or turned on itself (Yaw) [6].

Figure 2-1 General Structure of quadrirotor.

2.3 Controlling the drone on the three axes of yaw, pitch and roll

The quadrotor is a multidimensional MIMO (Multiple Input, Multiple Output) system. In other words, it is a system with several inputs and several outputs. Its outputs are the actuators and they are usually made up of four brushless motors. Each motor is controlled by an ESC and is fitted with a propeller. As for the inputs, these are mainly the data recovered from the sensors, namely the angular rotations relating to the variables to be controlled, namely throttle, roll, pitch and yaw. Their stabilization is necessary for stable flight, which is why we are using four PID controllers for each of these variables.

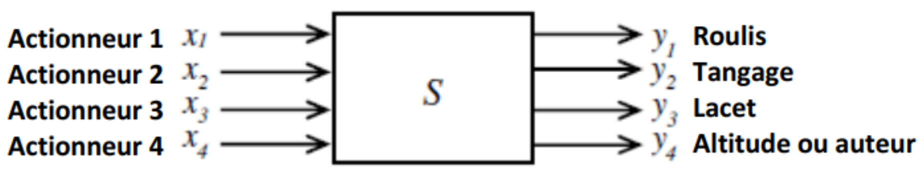


Figure 2-2 Drone as a MIMO system

The relationships between the controller and the variables to be controlled can be represented by:

$$Y(p) = S(p).X(p) \qquad \text{(II.0)}$$

Where, p is the Laplace operator.

$$X(p) = |X_1(p) \quad X_2(p) \quad X_3(p) \quad X_4(p)|^T$$

$$Y(p) = |Y_1(p) \quad Y_2(p) \quad Y_3(p) \quad Y_4(p)|^T$$

$$S(p) = \begin{vmatrix} S_{11}(p) & S_{12}(p) & S_{13}(p) & S_{14}(p) \\ S_{21}(p) & S_{22}(p) & S_{23}(p) & S_{24}(p) \\ S_{31}(p) & S_{32}(p) & S_{33}(p) & S_{34}(p) \\ S_{41}(p) & S_{42}(p) & S_{43}(p) & S_{44}(p) \end{vmatrix}$$

2.4 Quadrotor movements

The basic movements of the quadrotor are made by varying the speed of each rotor so as to modify the thrust produced. These movements are coupled, meaning that the quadrotor cannot move without performing a roll (ϕ) or pitch (θ) movement. The quadrotor has five main movements: The vertical movement (Thrust), the roll movement (Roll), the pitch movement (Pitch), the yaw movement (Yaw) and the translation movement.

2.4.1 Vertical movement

The vertical movement simply corresponds to the ascent and descent of the quadrotor. Ascent is obtained by increasing the speed of the four motors to equal values, which has the effect of cancelling out the torque generated. Descent is achieved by reducing the speed of the motors. Figure 2.3 illustrates the vertical movement.

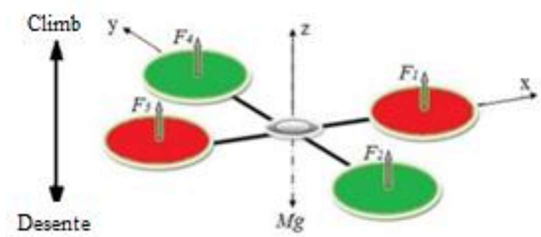


Figure 2-3 Illustration of vertical movement.

2.4.2 Roll movments

To obtain a rolling movement, a torque is applied around the axis, which means a difference in thrust between the rotor (02) and the rotor (04). This movement (rotation about the x axis) is coupled to a translational movement along the y axis.

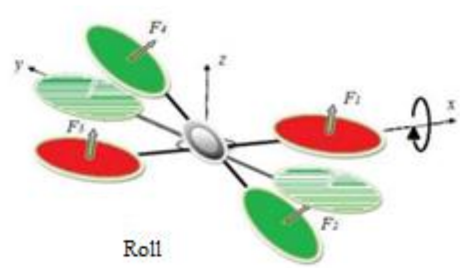


Figure 2-4 Illustration of roll movement.

2.4.3 Pitch movement

Similarly, by applying a torque about the y axis, which results in a difference in thrust between the rotor (01) and the rotor (03), a pitching motion is obtained. This is coupled to a translational movement along the x axis.

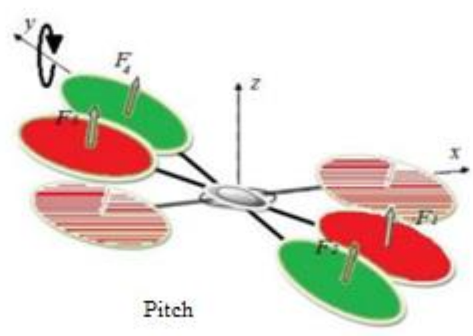


Figure 2-5 Illustration of Pitch movement.

2.4.4 Yaw movement:

Yaw is used to turn the quadrotor on itself. This is achieved by increasing the speed of the rotors (1 and 3) and proportionally decreasing the speed of the rotors (2 and 4).

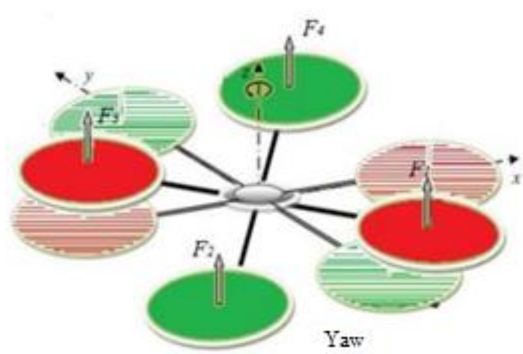


Figure 2-6 Illustration of Yaw movement.

2.4.5 Transnational movement

To obtain translational movement along the X or Y axis, all you have to do is roll or pitch, because these movements depend directly on the dynamics of the attitude.

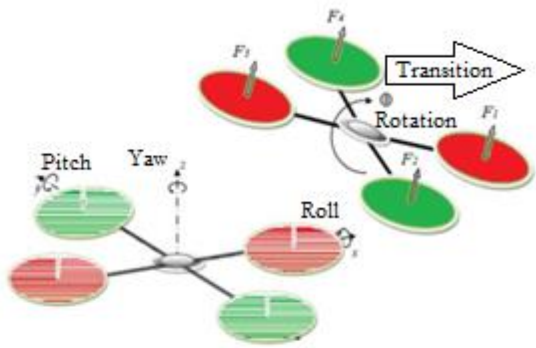


Figure 2-7 Illustration of translational movement.

2.5 Quadrirotor flight mechanics :

Quadrotors are fairly complex mechanical systems. Their movements are governed by several effects, both mechanical and aerodynamic. The quadrotor model must take into account all the effects that affect its movement, including gyroscopic effects.

Modelling aerial robots is a delicate task, since the dynamics of the system are highly non-linear and fully coupled. In order to better understand the dynamic model developed below, here are the various working hypotheses:

- The quadrotor structure is assumed to be rigid and symmetrical, which means that the inertia matrix will be assumed to be diagonal.
- The propellers are assumed to be rigid in order to neglect the effect of their deformation during rotation.
- The center of mass and the origin of the frame of reference coincide…..
- The lift and drag forces are proportional to the squares of the rotational speed of the rotors, which is a very close approximation to aerodynamic behaviour.

To evaluate the mathematical model of the quadrotor we use two reference points, a fixed reference point linked to the earth $R^b$and a mobile reference point R^m joined to the center of mass of the quadrotor body and located in the intersection of the two bars. The transition between the moving frame of reference and the fixed frame of reference is given by a matrix known as the transformation matrix T, which contains the orientation and position of the moving frame of reference relative to the fixed frame of reference. The following axis convention is chosen [7].

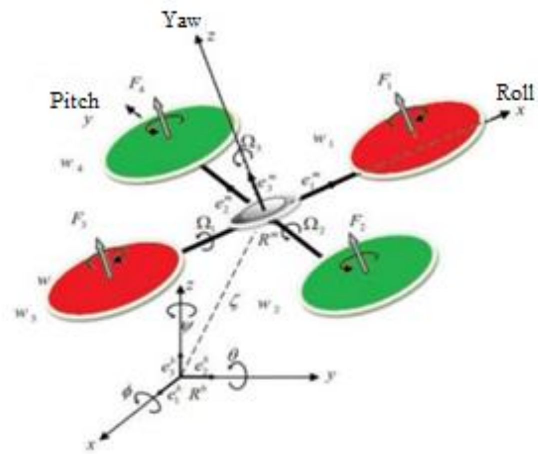


Figure 2-8 The quadrotor structure and relative coordinate systems.

$$T = \begin{bmatrix} R & \xi \\ 0 & 1 \end{bmatrix} \tag{II.1}$$

With R is the rotation matrix (describes the orientation of the moving object), $\xi = [x \quad y \quad z]^T$ is the position vector. To determine the elements of the rotation matrix R, we use the Euler angles.

2.5.1 Euler angles

At the beginning, the moving frame of reference is coincident with the fixed frame of reference, after which the moving frame of reference makes a rotational movement about the x axis of a roll angle$\left(-\frac{\pi}{2} < \phi < \frac{\pi}{2}\right)$, followed by a rotation about the y axis of a pitch angle $\left(-\frac{\pi}{2} < \phi < \frac{\pi}{2}\right)$ followed by a rotation about the z-axis of a yaw angle (-π < ψ < π).

So we have the formula for the rotation matrix R :

$$R = Rot_z(\psi) * \ Rot_y(\theta) * \ Rot_x(\phi)$$

$$R = \begin{bmatrix} c\psi & -s\psi & 0 \\ s\psi & c\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c\theta & 0 & s\theta \\ 0 & 1 & 0 \\ -s\theta & 0 & c\theta \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & c\phi & -s\phi \\ 0 & s\phi & c\phi \end{bmatrix} \tag{II.2}$$

$$R = \begin{bmatrix} c\psi c\theta & s\phi s\theta c\psi - s\psi c\phi & c\phi s\theta c\psi + s\psi s\phi \\ s\psi c\theta & s\phi s\theta s\psi + c\psi c\theta & c\phi s\theta s\psi - s\phi c\psi \\ -s\theta & s\phi c\theta & c\phi c\theta \end{bmatrix} \tag{II.3}$$

With: s is sinus function and c is cosine function.

2.5.2 Angular velocity

The rotational speeds $\Omega1, \Omega2, \Omega3$ in the fixed frame of reference are expressed as a function of the rotational speeds $\dot{\phi}, \dot{\theta}, \dot{\psi}$ in the moving frame of reference, we have :

$$\Omega = \begin{bmatrix} \Omega_1 \\ \Omega_2 \\ \Omega_3 \end{bmatrix} = \begin{bmatrix} \dot{\phi} \\ 0 \\ 0 \end{bmatrix} + \mathrm{Rot}_x\ (\phi)^{-1} \begin{bmatrix} 0 \\ \dot{\theta} \\ 0 \end{bmatrix} + \left(\mathrm{Rot}_y\ (\theta)\mathrm{Rot}_x\ (\phi)\right)^{-1} \begin{bmatrix} 0 \\ 0 \\ \dot{\psi} \end{bmatrix} \tag{II.4}$$

The roll rotation takes place when the reference frames are still the same. Then, as far as pitching is concerned, the vector representing the rotation must be expressed in the fixed reference frame: it is therefore multiplied by Rotx $(\phi)^{-1}$. Similarly, the vector representing the yaw rotation must be expressed in the fixed reference frame, which has already undergone two rotations. The rotational speed becomes as follow:

$$\Omega = \begin{bmatrix} \Omega_1 \\ \Omega_2 \\ \Omega_3 \end{bmatrix} \quad = \begin{bmatrix} \dot{\phi} \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ \dot{\theta} c\phi \\ \dot{\theta}\theta s \end{bmatrix} + \begin{bmatrix} -\dot{\psi} s\theta \\ \dot{\psi} s\phi c\theta \\ \dot{\psi} s\phi c\theta \end{bmatrix} = \begin{bmatrix} \dot{\phi} - \dot{\psi} s\theta \\ \dot{\theta} c\phi + \dot{\psi} s\phi c\theta \\ \dot{\psi} s\phi c\theta - \dot{\theta} s\phi \end{bmatrix} \tag{II.5}$$

$$\Omega = \begin{bmatrix} 1 & 0 & -s\theta \\ 0 & c\phi & s\phi c\theta \\ 0 & -s\phi & c\phi c\theta \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \tag{II.6}$$

When the quadrotor makes small rotations, we can make the following approximations:

$$c\phi = c\theta = c\psi = 1 \text{ Et } s\phi = s\theta = s\psi = 0$$

So the angular velocity will be:

$$\Omega = [\dot{\phi} \quad \dot{\theta} \quad \dot{\psi}]^{\mathrm{T}} \tag{II.7}$$

2.5.3 Linear velocity

Linear speeds $v_x^b; v_y^b; v_z^b$ in the fixed frame in function to linear speeds $v_x^m; v_y^m; v_z^m$ in movable frame are given as follow:

$$v = \begin{bmatrix} v_x^b \\ v_y^b \\ v_z^b \end{bmatrix} = R * \begin{bmatrix} v_x^m \\ v_y^m \\ v_z^m \end{bmatrix} \quad \text{(II.8)}$$

2.6 Physical effects acting on the quadrotor

In our case, we will only consider the forces and moments applied to the quadrotor generated by aerodynamic effects, propeller rotation and gyroscopic precession.

2.6.1 Forces :

The forces acting on the quadrotor system are detailed in following sub-sections:

2.6.1.1 The weight of the quadrotor

This force is due to the mass of the object. It is always perpendicular to the surface of the earth. It is given in the inertial reference frame by:

$$P = m \cdot g \cdot \vec{k} \quad \text{(II.9)}$$

With: m is the total mass and g is gravity.

2.6.1.2 Thrust forces

The forces caused by the rotation of the motors are perpendicular to the plane of the propellers. These forces are proportional to the square of the engine speed:

$$F_i = b\omega_i^2 \quad \text{(II.10)}$$

With $i = \overline{\overline{1:4}}, b$: is the coefficient of lift, which depends on the shape and number of blades and the density of the air.

2.6.1.3 Drag forces:

The drag force is the coupling between a pressure force and the viscous friction force, in this case we have two drag forces acting on the system which they are :

Drag in propellers: acts on the blades, is proportional to the density of the air, the shape of the blades and the square of the speed of rotation of the propeller, it is given by the following relationship:

$$T_h = d.\omega_i^2 \tag{II.11}$$

With d The drag coefficient depends on the manufacture of the propeller.

Drag along the axes (x , y , z): this is due to the movement of the quadrotor body.

$$F_{tT} = K_{ftT}.v \tag{II.12}$$

$$F_{tR} = K_{ftR} \cdot \Omega \tag{II.13}$$

With : $\boldsymbol{K_{ftT}}$**.** The translational drag coefficient and v: the linear velocity.

$\boldsymbol{K_{ftR}}$ The coefficient of rotational drag and **Ω**: the angular velocity.

2.6.2 Moments

There are several moments acting on the quadrotor. These moments are due to the forces of thrust and drag and to gyroscopic effects.

2.6.2.1 Moments due to thrust forces:

Rotation about the x-axis: this is due to the moment generated by the difference between the lift forces of rotors 2 and 4. This moment is given by the following relationship:

$$M_x = l(F_4 - F_2) = lb(\omega_4^2 - \omega_2^2) \tag{II.14}$$

With l is the length of the arm between the rotor and the quadrotor's centre of gravity.

Rotation about the y-axis: this is due to the moment generated by the difference between the lift forces of rotors 1 and 3. This moment is given by the following relationship:

$$M_y = l(F_3 - F_1) = lb(\omega_3^2 - \omega_1^2) \tag{II.15}$$

2.6.2.2 2.6.2.2. Moments due to drag forces:

Rotation about the z axis: this is due to a drag torque in each propeller, which is given by:

$$M_z = ld(\omega_1^2 - \omega_2^2 + \omega_3^2 - \omega_4^2) \quad \text{(II.16)}$$

Moment resulting from aerodynamic friction, given by:

$$M_a = K_{fa}\Omega^2 \quad \text{(II.17)}$$

2.6.3 Gyroscopique effect:

The gyroscopic effect is defined as the difficulty of modifying the position or orientation of the plane of rotation of a rotating mass. The gyroscopic effect is named after the way in which the gyroscope, a motion control device used in aviation, works (from the Greek gyro meaning rotation and scope, to observe).

In our case there are two gyroscopic moments, the first is the gyroscopic moment of the propellers, the other is the gyroscopic moment due to quadrotor movements (SATLA, 2018).

Propeller gyroscopic moment: given by the following relationship:

$$M_{gh} = \sum_1^4 \Omega \wedge J_r[0 \quad 0 \quad (-1)^{i+1}\omega_i]^T \quad \text{(II.18)}$$

With J_r : is the inertia of the rotors.

Gyroscopic moment due to quadrotor movements: this is given by the following relationship:

$$M_{gm} = \Omega \wedge J\Omega \quad \text{(II.19)}$$

With J is the inertia of the system.

2.7 Development of the Newton-Euler mathematical model:

The Quadrotor is modelled as a rigid body subject to actuating and external forces. A well-known result of mechanics is that the dynamics of rigid bodies can be described using the Newton-Euler approach (based on the forces and moments acting on the body) and the Euler-Lagrange approach (based on energy assumptions). With the Newton-Euler approach, the dynamics is initially formulated in terms of the coordinates of the moving frame of reference (linked to the rigid body), then expressed in terms of the coordinates of the inertial frame of reference using kinematic transformations. Lagrange's approach, on the other hand, directly requires the use of generalised coordinates (coordinates of the inertial frame of reference) and this requires a much heavier symbolism. As a result, the final result is the same, but obtained using different notations. In this section, a Newton-Euler approach is adopted in order to derive the rigid body dynamics of the Quadrotor, as it represents the simplest approach for modelling (Yacef, 2018). Based on the previous equations of the forces applied to the quadrotor equations (II.9,...II.12), and the moments acting on the quadrotor equations (II.14,.......II.18). And to summarise the set of equations describing the complete quadrotor model, we use the Newton-Euler formulation, and the dynamic system model is present in the following form (SATLA, 2018):

$$\begin{cases} \dot{\zeta} = v \\ m\ddot{\zeta} = F_f + F_t + F_g \\ \dot{R} = RS(\Omega) \\ J\dot{\Omega} = -\Omega \wedge J\Omega + M_f - M_a - M_{gh} \end{cases} \tag{II.20}$$

With:

ζ : is the quadrotor position vector.

m: The total mass of the quadrotor.

Ω : The angular velocity expressed in the fixed reference frame.

R : The rotation matrix.

A: The vector product.

$$J = \begin{bmatrix} I_x & 0 & 0 \\ 0 & I_y & 0 \\ 0 & 0 & I_z \end{bmatrix} \tag{II.21}$$

$S(\Omega)$: is the antisymmetric matrix; for a velocity vector $\Omega = [\Omega_1; \Omega_2; \Omega_3]$, it is given by:

$$S(\Omega) = \begin{bmatrix} 0 & -\Omega_3 & \Omega_2 \\ \Omega_3 & 0 & -\Omega_1 \\ -\Omega_2 & \Omega_1 & 0 \end{bmatrix} \tag{II.22}$$

F_f : is the total force generated by the four rotors, given by:

$$F_f = R * [0 \quad 0 \quad \sum_1^4 F_i]^T \tag{II.23}$$

$$F_i = b \cdot \omega_i^2 \tag{II.24}$$

F_t : The drag force along the axes (x,y,z) is given by:

$$F_t = \begin{bmatrix} -K_{ftTx} & 0 & 0 \\ 0 & -K_{ftTy} & 0 \\ 0 & 0 & -K_{ftTz} \end{bmatrix} . \dot{\zeta} \tag{II.25}$$

$K_{ftTx}, K_{ftTy}, K_{ftTz}$: Translation drag coefficients.

F_g: The force of gravity is given by:

$$F_g = \begin{bmatrix} 0 \\ 0 \\ -mg \end{bmatrix} \tag{II.26}$$

M_f : Moment caused by the forces of thrust and drag. It is given by:

$$M_f = \begin{bmatrix} l(F_4 - F_2) \\ l(F_3 - F_1) \\ ld(\omega_1^2 - \omega_2^2 + \omega_3^2 - \omega_4^2) \end{bmatrix} \tag{II.27}$$

M_a: Moment resulting from aerodynamic friction, given by:

$$M_a = \begin{bmatrix} K_{fax}\dot{\phi}^2 \\ K_{fay}\dot{\theta}^2 \\ K_{faz}\dot{\psi}^2 \end{bmatrix} \tag{II.28}$$

$K_{fax}, K_{fay}, K_{faz}$: Aerodynamic friction coefficients.

2.7.1 Translational equations of motion:

Having presented the equations of the forces in the previous sections, we can now move on to the complete model of the quadrotor; using Newton's second law in the case of linear motion, we have the following formula:

$$m\ddot{\zeta} = F_f + F_t + F_g \tag{II.29}$$

Replace each force by its formula, and we find:

$$m\begin{bmatrix}\ddot{x}\\ \ddot{y}\\ \ddot{z}\end{bmatrix} = \begin{bmatrix}c\phi c\psi s\theta + s\phi s\psi\\ c\phi s\psi s\theta - s\phi c\psi\\ c\phi c\theta\end{bmatrix}\sum_1^4 F_i - \begin{bmatrix}K_{ftTx}\dot{x}\\ K_{ftTy}\dot{y}\\ K_{ftTz}\dot{z}\end{bmatrix} + \begin{bmatrix}0\\ 0\\ -mg\end{bmatrix} \tag{II.30}$$

This gives us the differential equations that define translational motion:

$$\begin{cases}\ddot{x} = \frac{1}{m}(c\phi c\psi s\theta + s\phi s\psi)(\sum_1^4 F_i) - \frac{K_{ftTx}}{m}\dot{x}\\ \ddot{y} = \frac{1}{m}(c\phi s\psi s\theta + s\phi c\psi)(\sum_1^4 F_i) - \frac{K_{ftTy}}{m}\dot{y}\\ \ddot{z} = \frac{1}{m}(c\phi c\theta)(\sum_1^4 F_i) - \frac{K_{ftTz}}{m}\dot{z} - g\end{cases} \tag{II.31}$$

2.7.2 Equations of rotational motion:

Applying Newton's principle to the case of rotation, we find the following formula:

$$J\Omega = -\Omega \wedge J\Omega + M_f - M_a - M_{gh} \tag{II.32}$$

If we replace each moment by the corresponding expression, we find:

$$\begin{bmatrix}I_x & 0 & 0\\ 0 & I_y & 0\\ 0 & 0 & I_z\end{bmatrix}\begin{bmatrix}\bar{\phi}\\ \ddot{\theta}\\ \bar{\psi}\end{bmatrix} = -\begin{bmatrix}\phi\\ \dot{\theta}\\ \dot{\psi}\end{bmatrix} \wedge \left(\begin{bmatrix}I_x & 0 & 0\\ 0 & I_y & 0\\ 0 & 0 & I_z\end{bmatrix}\begin{bmatrix}\dot{\phi}\\ \dot{\theta}\\ \dot{\psi}\end{bmatrix}\right) - \begin{bmatrix}J_r\bar{\Omega}_r\dot{\theta}\\ -J_r\bar{\Omega}_r\theta\\ 0\end{bmatrix}$$
$$-\begin{bmatrix}lb(\omega_4{}^2 - \omega_2^2)\\ K_{fax}\dot{\phi}^2\\ K_{fay}\dot{\theta}^2\\ K_{faz}\dot{\psi}^2\end{bmatrix} + \begin{bmatrix}lb(\omega_3{}^2 - \omega_1^2)\\ ld(\omega_1^2 - \omega_2^2 + \omega_3^2 - \omega_4{}^2)\end{bmatrix} \tag{II.33}$$

We then obtain the differential equations defining the rotational movement:

$$\begin{cases} I_x\ddot{\phi} = -\dot{\theta}\dot{\psi}\left(I_z - I_y\right) - J_r\bar{\Omega}_r\dot{\theta} - K_{fax}\dot{\phi}^2 + lb(\omega_4^2 - \omega_2^2) \\ I_y\ddot{\theta} = \dot{\phi}\dot{\psi}(I_z - I_x) + J_r\bar{\Omega}_r\dot{\theta} - K_{fay}\dot{\theta}^2 + lb(\omega_3^2 - \omega_1^2) \\ I_z\ddot{\psi} = \dot{\phi}\dot{\theta}\left(I_y - I_x\right) - K_{faz}\psi^2 + ld(\omega_1^2 - \omega_2^2 + \omega_3^2 - \omega_4^2) \end{cases} \tag{II.34}$$

With:

$$\bar{\Omega}_r = \omega_1 - \omega_2 + \omega_3 - \omega_4 \tag{II.35}$$

En conséquence, le modèle dynamique complet qui régit le quadrotor est donne par le système des équations suivant :

$$\begin{cases} \ddot{\phi} = -\dot{\theta}\psi\,\frac{(I_z - I_y)}{I_x} - \frac{J_r}{I_x}\bar{\Omega}_r\dot{\theta} - \frac{K_{fax}}{I_x}\phi^2 + \frac{lb}{l_x}(\omega_4{}^2 - \omega_2{}^2) \\ \ddot{\theta} = \phi\dot{\psi}\,\frac{(I_z - I_x)}{I_y} + \frac{J_r}{I_y}\bar{\Omega}_r\dot{\theta} - \frac{K_{fay}}{I_y}\dot{\theta}^2 + \frac{lb}{I_y}(\omega_3{}^2 - \omega_1{}^2) \\ \ddot{\psi} = \phi\dot{\theta}\,\frac{(I_y - I_x)}{I_z} - \frac{K_{faz}}{I_z}\psi^2 + \frac{ld}{I_z}(\omega_1{}^2 - \omega_2{}^2 + \omega_3{}^2 - \omega_4{}^2) \\ \ddot{x} = \frac{1}{m}(c\phi c\psi s\theta + s\phi s\psi)(\sum_1^4 F_i) - \frac{K_{ftTx}}{m}\dot{x} \\ \ddot{y} = \frac{1}{m}(c\phi s\psi s\theta + s\phi c\psi)(\sum_1^4 F_i) - \frac{K_{ftry}}{m}\dot{y} \\ \tilde{z} = \frac{1}{m}(c\phi c\theta)(\sum_1^4 F_i) - \frac{K_{ft\tau z}}{m}\dot{z} - g \end{cases} \tag{II.36}$$

2.7.3 Force/moment relationship and motor speed

From the equations cited above (II.10, II.14, II.15, II.16), we can calculate the speed of the motors from the forces and moments applied to the quadrotor. This relationship is very important for the implementation of the controller. We can therefore rewrite the equations in matrix form as follows:

$$\begin{bmatrix} F \\ M_x \\ M_y \\ M_z \end{bmatrix} = \begin{bmatrix} b & b & b & b \\ 0 & -bl & 0 & bl \\ -bl & 0 & bl & 0 \\ d & -d & d & -d \end{bmatrix} \begin{bmatrix} \omega_1^2 \\ \omega_2^2 \\ \omega_3^2 \\ \omega_4^2 \end{bmatrix} \tag{II.37}$$

By inverting the matrix, we obtain the relationship between the speed of the motors:

$$\begin{bmatrix} \omega_1^2 \\ \omega_2^2 \\ \omega_3^2 \\ \omega_4^2 \end{bmatrix} = \begin{bmatrix} \frac{1}{4b} & 0 & \frac{1}{2bl} & -\frac{1}{4b} \\ \frac{1}{4b} & -\frac{1}{2bl} & 0 & \frac{1}{4b} \\ \frac{1}{4b} & 0 & -\frac{1}{2bl} & -\frac{1}{4b} \\ \frac{1}{4b} & \frac{1}{2bl} & 0 & \frac{1}{4b} \end{bmatrix} \begin{bmatrix} F \\ M_x \\ M_y \\ M_z \end{bmatrix} \tag{II.38}$$

2.8 The PID controller

The PID controller, or PID (proportional, integral, derivative) corrector, is a control algorithm used to improve the performance of a closed-loop system or process. It is the most widely used controller in many fields, where its corrective qualities apply to a wide range of physical quantities. The advantages of this controller include [37] :

- Simple structure.
- Good performance in many processes.Reliable, even without a specific model of the control system.

The general principle of a controller is to compare the SP (Set Point) setpoint and the PV (Process Value) system state or measurement, in order to correct it effectively. Regulation (or constant-set-point control) consists in acting in such a way that a measurement is equal to a set point (in reality, it just has to be as close as possible). If the aim is to reach a setpoint that can evolve over time according to an appropriate mathematical law, we speak of tracking or enslavement; if the aim is to eliminate disturbances so that a value remains constant, we speak of regulation [37].

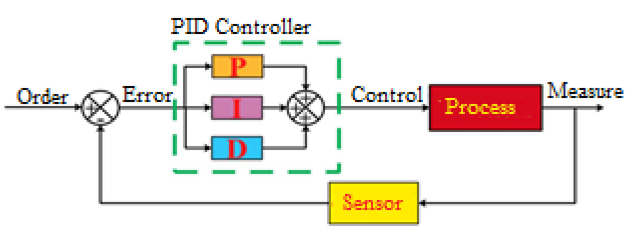


Figure 2-9 Architect of the the PID controller

The control loop variables can therefore be defined according to the following table:

Table 2.1 Control loop variables.

Ordered size	Drone rotation angle, altitude level
Control size	The speed of each motor
Disturbance magnitude or disturbance	The wind
Setpoint value	The altitude level, angle = 0°.
Order error or deviation	The difference between the desired value and the measured value

2.8.1 Implementing a PID corrector on the Arduino board

Our setup uses a single sensor (MPU 6050) that contains both an accelerometer and a gyroscope. This sensor is capable of calculating rotation angles along the three axes x, y, z. So we use 3 PID controllers for the internal loop (φ, θ, ψ).

Due to its inherent instability, the implementation of a control law is imperative for the flights of our quadrirotor. The PID controller plays a crucial role in providing the necessary commands to the motors, depending on the quadrirotor's orientation (which is determined by sensor feedback) and the specific instructions it receives.

The effects of each PID controller parameter on our drone are as follows:

Proportional action P: This is the most important factor, defining the level of correction. To set it, we increase the value of P at a low I until we obtain vibrations. Finally, we retain the value of P before vibration occurs.

Integral action I: The quadrotor naturally glides on its axes according to the last setpoint, and this is where the integral term comes into play. In other words, when the error is small and lasts over time. To adjust this parameter, we increase the value of I until the quadrirotor's behavior is stable. The higher the I term, the slower the flying robot's behavior, so it will be less responsive.

Derivative action D: This term is reduced to soften the quadrirotor's reactions; if, on the other hand, you want the robot to behave nervously, increase its value. If you increase D too much, you may have the same effect as increasing P: you'll end up with vibrations [14].

2.9 Conclusion

In this chapter, we have presented the operating principle of the quadrotor. This type of drone contains four rotors, two of which turn in one direction and the other two in the opposite direction. By varying the rotational speeds of these rotors, the quadrotor can make different movements in both translation and rotation. Then we presented the movements and flight mechanics of the quadrotor using the Newton-Euler formalism, which enabled us to establish the dynamic model of the quadrotor in order to get as close as possible to the real dynamics of the quadrotor. The latter is subject to perturbations from the external environment which influence the rotary behaviour of the quadrotor, and we have used the classic linear PID corrector. The next chapter will be devoted to the production of a quadrotor UAV.

CHAPTER 3

CONCEPTION AND OPERATING TESTS

3 CHAPTER 3 : CONCEPTION AND OPERATING TESTS

3.1 Introduction

This chapter is divided into two parts. The first part explains the main components of the drone, and gives more detail about each one. In addition, the role of each component in the global system is given. The second part demonstrates the final prototype of the drone and some operating tests. Also, the main parameters of control are discussed to show the system performances.

3.2 General concept of the project

The proposed project that we are going to realize, it consists of three main models of UAVs which are a Quadri-rotor, RC Trainer plane and the VTOL. The quadrirotor is powered by four motors, these motors are controlled by an arduino flight controller (Atmega318p microcontroller) via electronic speed controllers (ESC). This drone contains a small camera which is powered by a battery.

The RC Train:r plane is for training on flying and controlling the planes. First, we will design the RC Trainer Plane's structure. Then we will place three servo motors to control the wings of this airplane, and we'll place a brushless motor with an ESC on the back of the plane.

The VTOL (vertical take-off and landing) is hybridization of the RC Trainer Aireplane's structure with the functioning of the Quadri-rotor. This hybridization is more important to ensure the firefighting objective. We will make a support that can carry numbers of Rockets contains the Fire balls that extinguish the fire. The fire balls can be activated by an electric detonator.

The control station is developed by using a Human Machine Interface (HMI), an arduino card with radio frequency wireless communication module. The interface allows following the drone movements in the space. Figure 3.1 shows the general conception of the project.

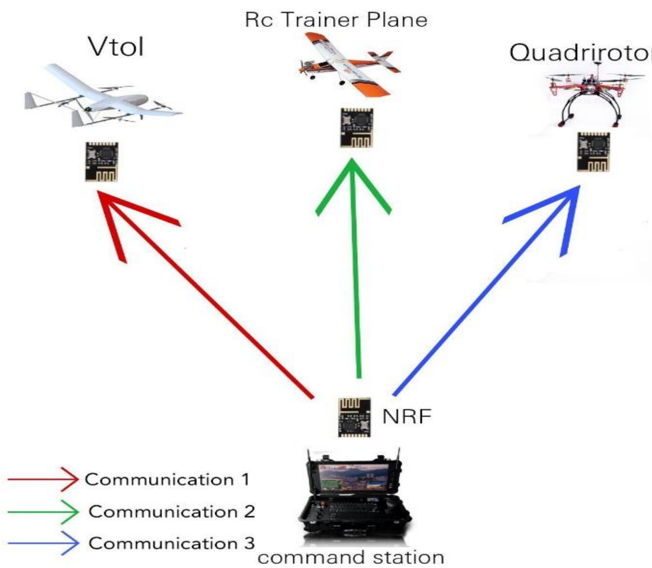


Figure 3-1 General conception of the project.

A.PART ONE : The Drone Components

3.3 UAV Quadri-rotor components

The quadcopter mainly consists of a chassis and a propulsion system. A complete propulsion system includes motors, propellers, electronic speed controllers (ESC), and a battery. Almost all small multi-rotor drones are electric, and almost none are gas-powered. For this reason, we will focus on the implementation of electric propulsion using brushless DC motors [8].

3.3.1 The frame

The choice of the frame is essential because it determines many other components of the quadricopter. If you opt for a large chassis, you will have to adapt all the other components to suit it. So the chassis is the first thing to consider when building a quadricopter.

A quadricopter is made up of four arms arranged in a '+' or 'x' shape, or even in an 'H' shape. The only difference between these models is that the cross-shaped chassis offers a better unobstructed view to the front and rear. This makes it particularly suitable for on-board shooting and capturing images or video.

The characteristics to be taken into account for the chassis are weight, which will be linked to the construction materials and its resistance to impact; the lighter the chassis, the more power you retain and the more flight time you gain [22]. Figure 3.2 shows an example of a frame.

Figure 3-2 example of Frame DJIF450

3.3.2 Brushless motors

Figure 3-3 Example of Brushless motors

Brushless motors have many advantages over direct current motors; the absence of brushes results in better energy performance and increased reliability. First, it is necessary to define the type of application of the drone supporting these engines and the final weight of the device.

Parameters to be defined are desired endurance, drone size, total weight, payload, maximum speed, number of engines (4 in our case).

Generally, for a small drone we need 8000 rpm maximum "stable" flight (fire), 10,000 rpm "normal" flight (versatile) or 12,000 rpm maximum flight "nervous" (addiction, speed).

So if you want to fly with a 3S battery (11.1V) it is recommended to use 8,000 rpm/11.1V 720 KV, 10,000 rpm/11.1V 900 KV or 12,000 rpm/11.1V 1100 KV.

Table 3.1 Main parameters of a small drone

Function	Versatile, FPV, shooting ...
Drone's Size	450mm
Total weight	2 Kg
Payload	0.8 Kg
Maximum speed	50Km
Number of engine	4

3.3.3 The propellers

Figure 3-4 Example of Propellers

When designing a quad-copter, the choice of engine and propeller type is important and affects its dynamics. The propellers are available in different types (large pitch, small pitch), in different materials (plastic, carbon, fiberglass) and in different sizes. The size, weight, power and flexibility required for the machine must be considered when selecting a combination propeller engine. A propeller is a rotating device consisting of a number of blades with an aerodynamic profile that creates depressions and overpressures on either side of the propeller. The propeller used has two blades and measures 10 x 4.5 cm. The larger the propeller, the more power it takes to rotate it. But the bigger it is, the more lift it generates, and therefore the more it hovers and flies. Conversely, with a small propeller, you'll need less power, so you'll have less lift, but you'll fly more aggressively.

The propellers have a great influence on a drone by choosing between a size and a pitch greater or less can vary the endurance. Table 3.2 demonstrates examples of propeller & engine combinations.

Table 3.2 examples of propeller & engine combinations

Motors	Propellers	Results
Kv High	Large propellers	No → Pull hard on the battery and risk burning the engines.
Kv High	Small propellers	Yes → for acrobatic flying.
Kv Low	Large propellers	Yes → for stable flight.
Kv Low	Small propellers	No → not enough lift, the drone won't take off.

3.3.4 Electronic Speed Controller (ESC)

Figure 3-5 Example of an ESC

An electronic speed control (ESC) is an electronic circuit that controls and regulates the speed of an electric motor. It may also provide reversing of the motor and dynamic braking. Miniature electronic speed controls are used in electrically powered radio controlled models. Full-size electric vehicles also have systems to control the speed of their drive motors.

A variable speed drive allows several operations in the form of start-up and adjustment procedures or operations for programming several parameters such as: timing mode, control frequency, starting force....

Throttle rank setting is a procedure we've already used in the project to set the rpm margin, the protocol is as follows:

- ✓ Switch on transmitter.
- ✓ Move throttle lever to highest position.
- ✓ Connect battery to ESC.
- ✓ The sound of two beeps should be heard this means that the point representing the highest rpm has been confirmed and recorded.
- ✓ Move the throttle lever to the lowest position (in 2 seconds), a long beep should be heard to announce the detection of the point representing the lowest rpm.
- ✓ This is followed by the emission of several beeps indicating the number of battery cells used.
- ✓ Immediately after the test is completed, a melody is heard, making the throttle ready for use.

Table 3.3 The specifications of th ESC :

ESC	HW 30A
DC current	30A
Burst current	40A
Li-xxbatterie (Cells)	3

dimensions	47 x 24 x 9
weight with wires	25
BEC	2 A

3.3.5 Flight controller

A flight control is one of the most important elements of a UAVs. it is in charge of stabilizing the aircraft, ensuring accurate flight manoeuvres and providing data to the pilot. There is many Flight Controllers like Ardupilot APM, KK2,QQ...

some tests have been realized using Ardupilot APM, also by using Flight Controller based on Arduino UNO . Its embedded system contains principal electronic components.

3.3.5.1 Ardupilot APM Mega 2.6

APM 2.6 is a complete, open-source autopilot system that transforms any radio-controlled machine into a fully autonomous vehicle, capable of carrying out programmed (GPS) missions with waypoints.

Figure 3-6 Ardupilot APM Mega 2.6

3.3.5.2 Flight controller based on Arduino

3.3.5.2.1 Arduino UNO

Arduino Uno is a microcontroller board based on the ATmega328P . It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz quartz crystal, a USB connection, a power jack, an ICSP header and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with an AC-to-DC adapter or battery to get started.

Figure 3-7 Arduino UNO

Table 3.4 The specifications of the Arduino UNO:

Microcontroller	ATmega328P
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Inout Voltage (limit)	6-20V
Digital I/O Pins	14 (of which 6 provide PWM output)
PWM Digital I/O Pins	6
Analog Input Pins	6
DC Current per I/O Pin	20mA
DC current for 3.3V Pin	50mA
Flash Memory	32KB (Atmega328p)
SRAM	2KB (Atmega328p)
EEPROM	1KB (Atmega328p)
Clock Speed	16MHz
Length	68.6 mm
Width	58.4 mm
Weight	25

3.3.5.2.2 Gyroscope sensor MPU6050

The MPU6050 module is a micro electromechanical system (MEMS) that consists of a 3-axis accelerometer and a 3-axis gyroscope inside and a Digital Motion Processor (DMP), all in a small 4x4x0.9 mm package (DMP). This helps us measure acceleration, velocity, orientation, displacement and many other parameters related to the motion of a system or object[4].

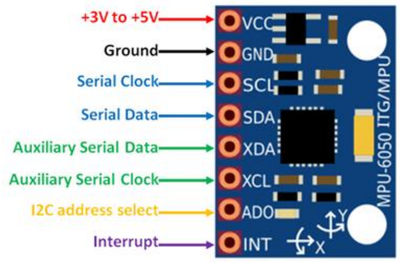


Figure 3-8 Gyroscope sensor MPU6050

3.3.5.2.3 Battery Batterie Lipo 2200 mAh

Lithium polymer batteries known as LiPo batteries are a new type of battery now used in many consumer electronics devices. They have gained popularity in the radio control industry in recent years, and are now the most popular choice for anyone looking for long life and high power. LiPo batteries offer a wide range of advantages. But each user must decide whether the advantages outweigh the disadvantages.

A LiPo cell has a nominal voltage of 3.7 V. For the 7.4 V battery, this means that there are two cells in series (which means that the voltage adds up). This is sometimes why you'll hear it referred to as a "2S" battery - it means there are 2 cells in series. So, a two-cell pack (2S) is 7.4 V, a three-cell pack (3S) is 11.1 V, and so on.

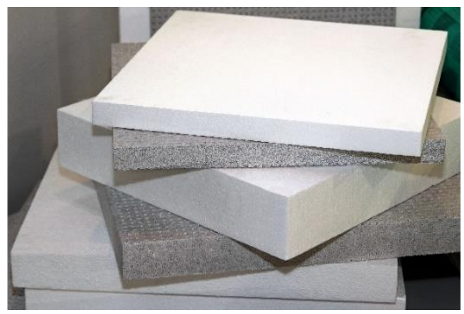
Figure 3-9 Example of Lipo-Battery

- **Specifications :**
    - ✓ Capacity: 2200mAh.
    - ✓ Nominal voltage: 3.7 V per cell.
    - ✓ Charge cut-off voltage: 4.2 V per cell.
    - ✓ Discharge cut-off voltage: 3.0 V per cell.
    - ✓ Charging technology: Direct Current (DC), Direct Voltage (DV).

- ✓ Life cycle: ≥ 500 times per cell (Once i.e. charge then discharge 4.2V, 3V respectively).
- ✓ Max current to be withdrawn: 35C i.e. 35 * 2.2 = 77A.
- ✓ Number of cells: 3S.
- ✓ Balancing wires.

3.4 RC Trainer Aireplane and the VTOL

3.4.1 The Structure compnents :

Polysterine	
Kraft paper	
Carbon fiber tubes	

Fibre de glass	
Resine	
Wheels	

3.4.2 Servo motors :

MG90S is a micro servo motor with metal gear. This small and lightweight servo comes with high output power, thus ideal for RC Airplane, Quadcopter or Robotic Arms.

Figure 3-10 Example of MG90S Servo motor

- **Specifications**
  - ✓ Operating Voltage: 4.8V to 6V (Typically 5V)
  - ✓ Stall Torque: 1.8 kg/cm (4.8V)
  - ✓ Max Stall Torque: 2.2 kg/cm (6V)
  - ✓ Operating speed is 0.1s/60° (4.8V)
  - ✓ Gear Type: Metal
  - ✓ Rotation : 0°-180°
  - ✓ Weight of motor : 13.4gm
  - ✓ Package includes gear horns and screws

3.4.3 Relay Sheild Module

This is a 5V 4-channel relay interface board, and each channel needs a 15-20mA driver current. It can be used to control various appliances and equipment with large current. It is equiped with high-current relays that work under AC250V 10A or DC30V 10A. It has a standard interface that can be controlled directly by microcontroller.

Figure 3-11 Example of Relay Sheild module

3.4.4 Electrical converter 3.3 volts to 400.000 volts

Its a dangerous device, that converts 3.3 volts to 400,000 volts . The high voltage unit uses the principle of the Tesla coil (coil) to raise the voltage. It produces a high voltage pulse, which is small in size and the efficiency is high, (the discharge intensity is violent).

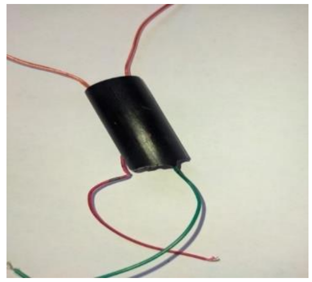


Figure 3-12 Example of Electrical converter

3.4.5 Fire ball :

Fire extinguisher balls are small shaped automatic fire extinguishers that can be thrown to the fire and dispense fire-fighting substances within 3 seconds. They are effective both passively and actively, i.e. mounted on the wall and/or thrown or rolled into flames. The loud noise that the fire extinguisher ball makes is useful in ridding the area of smoke. It also reduce the risk of fire by providing additional protection in areas where there is a high risk of Class B fires[10].

Figure 3-13 Example of Fire ball

3.5 Command Station

3.5.1 Arduino NANO

The Arduino Nano is a small, complete, and breadboard-friendly board based on the ATmega328 (Arduino Nano 3.x). It has more or less the same functionality of the Arduino Duemilanove, but in a different package. It lacks only a DC power jack, and works with a Mini-B USB cable instead of a standard one. [1]

Figure 3-14 Arduino NANO

Table 3.5 . specifications of the Arduino NANO

Microcontroller	Atmega328
Architecture	AVR
Operating Voltage	5 V
Flash Memory	32 KB of which 2 KB used by bootloader
SRAM	2 KB
Clock Speed	16 MHz
Analog IN pins	8
EEPROM	1 KB
DC Current per I/O Pins	40 mA
Input Voltage	7-12V
Digital I/O Pins	22 (6 which are PWM)
PWM Output	6
Power Consumption	19 mA
PCB Size	18 x 45 mm
Weight	7g

3.5.2 HMI Schneider Magelis (HMIGTO23/10)

A Human-Machine Interface (HMI) is a user interface for connecting a person to a machine, system or device. This term broadly defines any device allowing a user to interact with a device in an industrial environment [9].

Figure 3-15 HMI Schneider Magelis

3.5.3 RS458 To TTL Modbus module

RS485 is a standard defining the electrical characteristics of drivers and receivers for use in serial communications systems, widely used in industrial field. multi-point systems are supported. It's used to convert the TTL standard to the RS485 standard. If the outside serial device is RS485 standard, you can attach this unit onto M5stack, therefore, to implement the communication with RS485 device by TTL protocol.

It has an input DC of 12V and it can be programmed by th Arduino and python.We use it to make a connection between HMI and the Arduino.

Figure 3-16 RS485 Module

3.5.4 Joysticks

The joystick is a position sensor which returns two analogue values representing its X,Y position. We use it to control the direction or speed.

It consists of two mechanically coupled potentiometers positioned to detect the horizontal and vertical components of the joystick. Thus the resistance values of the potentiometers vary independently depending on the position of the joystick.

3.5.5 Switches

The switches is a fundamental component in electrical engineering, responsible for breaking or completing an electric circuit, in this RSC we need them for activating and desactivating some features like security and the option of throwing the fireball.

3.6 Software

3.6.1 Arduino IDE

The Arduino Integrated Development Environment - or Arduino Software (IDE) - contains a text editor for writing code, a message area, a text console, a toolbar with buttons for common functions and a series of menus. It connects to the Arduino hardware to upload programs and communicate with them [11].

Figure 3-17 Arduino IDE software logo

3.6.2 Vijeo Designer

Vijeo Designer is state-of-the-art software for creating operator screens and configuring the operational parameters of Human Machine Interface (HMI) devices. It provides all the tools needed to design an HMI project, from data acquisition to the creation and visualization of animated synoptics [12].

Figure 3-18 Vijeo Designer software logo

3.6.3 Proteuse

The Proteus software is an electronic card production software that also allows the simulation of electronic assemblies.

Figure 3-19 Proteuse software logo

3.6.4 Mission planner

Mission Planner is a ground control station for Plane, Copter and Rover. It is compatible with Windows only. Mission Planner can be used as a configuration utility or as a dynamic control supplement for your autonomous vehicle.

Figure 3-20 Mission Planner software logo

3.7 Partial conclusion :

In this part, we provided an overview of the system that we will implement, and we described each component with mentioning its characteristics in order to allow us to accomplish it in the right and successful way.

In the second part, we will explain how we will install and program each vehicle, while conducting the necessary experiments.

B.PART TWO : Realization and experience

3.8 Radio Commande Station RCS) based on RF Arduino and HMI :

There are some RF control devices as FlySky i6 with its reciver that makes you able to control your drone or any electronic vehicle. And you can also build your own radiocontroller device.

In our case we build a commande station based on Arduino NANO and NRF24L01 module to transmite the signals of the joystics and the switches to another NRF reciver that has an unobstructed transmission range of 1 km which placed in the vehicles that we want to control and we can control it through the Arduino nano board through the SPI port.

First of all the commande station contains two joystics which made from four potentiometers to control the quadrirotor through the transmission channels predefined on the Arduino Nano board, using only four channels we can control the six degrees of freedom our quadrirotor has. And it can control the servo motors which placed in the wings of the RC Trainer plane and the VTOL.

We makee an interface to the HMI through which we can specify which vehicle we want to control, each one her own interface and control panel.

Programming of RCS system is based on two codec’s for the transmitter and receiver. We relied on the C programming language available on the Arduino IDE software. And the HMI Interface programmed using Vijeo Designer Software, it sends the state of the variables to RCS via a serial communication protocol RS485.

3.8.1 Realization of the printed circuit boards (PCB) :

In this Commande station we have two main PCB.

The 1st board placed on the back of the HMI, contanis an Arduino Nano and RS485 module, it made for a serial communication (Modbus protocol) between the HMI and the board of Commande station.

We make it to choose which vehicle we want to control and to commande some extra features of each vehicle via the HMI.

First of all we designe the scheme by Proteuse software and we make the PCB layout

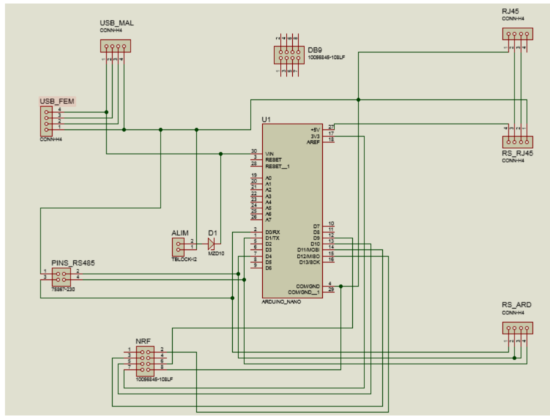
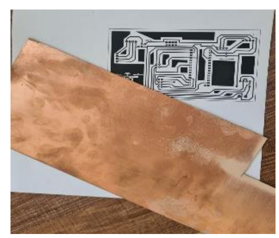
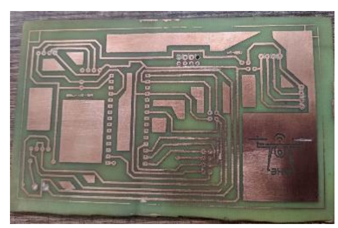


Figure 3-21 Scheme of the serial communication Arduino/RS485.

Table 3.6 The steps how we print the PCB :

1	Impression → Superposition ; Cutting → Superposition	We print the circuit on magazine paper then we cut the Epoxy/copper plate to the dimensions of the circuit.	
2	Superposition	We put the paper on the plate.	
3	Engraving operation	We put the electronic iron on the plate and we leave it for 5 to 7 mins.	
4	Cooling the card	Carefully we put the iron aside and rinse the plate in cold water. leave it for 10 mins.	
5	Paper peeling	Just peel it off completely. The operation does not present any difficulty because the water will have largely taken effect.	
6	Cleaning	At the end we obtain a copper plate carrying the printed circuit in black.	

7	Engraving	We wipe the card gently to eliminate the drops of water, the operation will not take more than thirty seconds. **Note**: one must operate gently so as not to erase the circuit.	
8	Rinsing	We place the card in a container filled with ferric chloride and we shake it. The perchloride will decompose the copper which is not protected, it will be eliminated on the whole plate except on the tracks drawn on the artwork. **Note**: During this operation, the safety instructions must be followed.	
9	Cleaning	We rinse the plate in water, we must be careful because the perchloride is always active, toxic and dirty. The printed circuit is cleaned using a cloth soaked in alcohol until the photosensitive layer disappears.	

3.8.1.1. Result of the first boared

Figure 3-22 The printed circuit placed on the back of the HMI

The second board is for the the commande station , we place it in a plat of Forex then we place it in a bag. This board contains an Arduino Nano and NRF module and pins to connect the wires of the switches and the joysticks.

We follow the same steps of the 1st board.

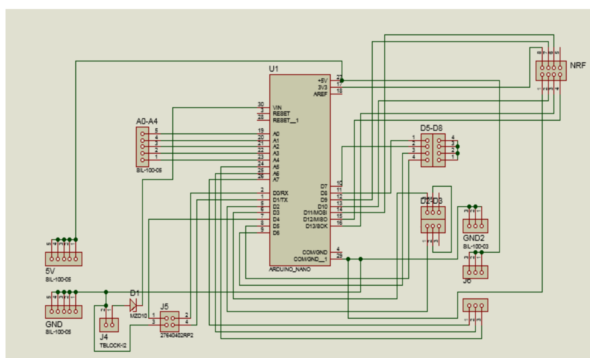


Figure 3-23 Schematic of the Commande station board

3.8.1.2. Result of the second board :

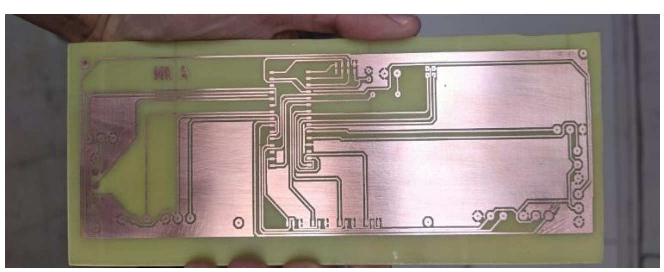


Figure 3-24 Commande station printed circuit

This is the result after the wiring :

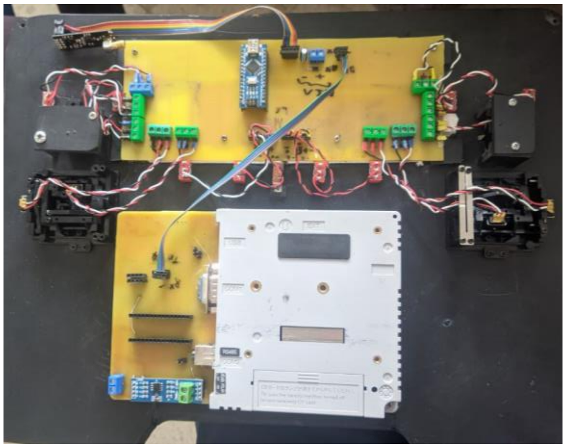


Figure 3-25 Commande station circuit with HMI

And this is the final result :

Figure 3-26 Commande station's bag

3.8.2 Programming the HMI

Vijeo designer is the software which we can designing the interface of Schneider's HMI.

So we open the software and we creat a new project, and select the appropriate HMI model.

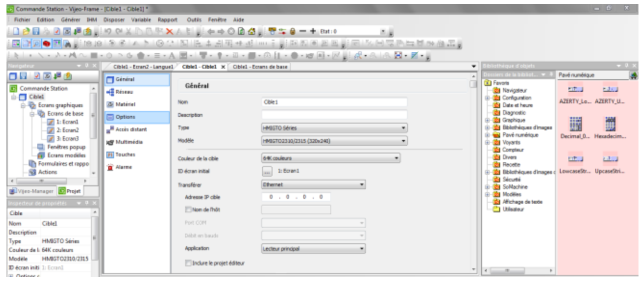


Figure 3-27 Vijeo designer interface

According to the used variables the interface has been designed

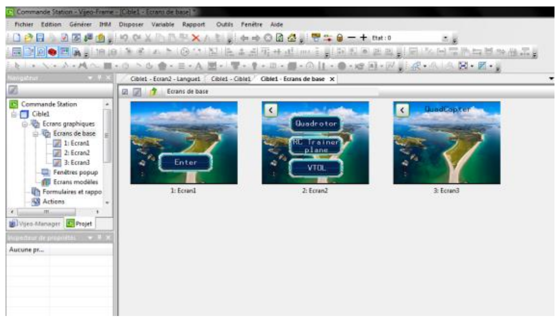


Figure 3-28 Designe the interfaces of the system

We save the project and send it to a USB disk, and we connect it the HMI and automatically the project will be installed in the HMI.

3.8.1.3. Comments and notes :

Concerning the alimenatation, The HMI must be alimentated by 24V, and the PCB by 5V. there is a usb port in HMI has 2 pins of power (VCC/GND) of 5V, so we aliment the PCB from those pins.

Concerning the wires, we use the Terminal screw blocks (Figure 3.27) in place of the jumpers (Figure 3.26), because if gaves us better connection and performance.

Figure 3-29 Jumpers

Figure 3-30 Terminal screw blocks

3.9 Realization of the Flight controller of the Quadrirotor

The flight controller based on an arduino microcontroller (Atmega328p) and MPU6050 sensor, and the recived signals from the commande station and the ESC actuators.

The sensor and the command station sends the data to the Arduino microcontroller to control the speed and the synchronization of the motors, and from this we achieve the stability.

The aim of this experiment is to install a program on the quadrotor that will enable us to raise it to a certain altitude and fly at a specific speed. To do this, we've assembled all the necessary electronic components into the circuit shown in figure (3.28).

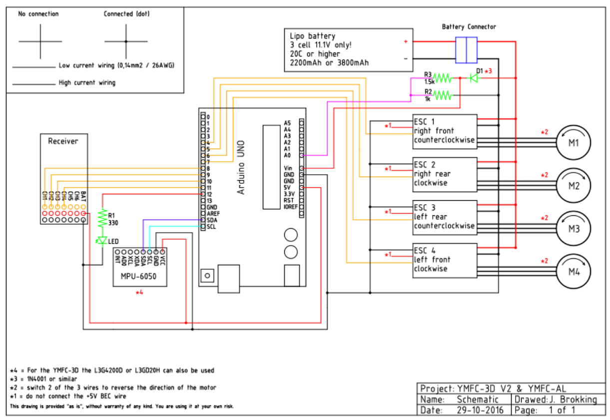


Figure 3-31 Flight controller's schematic[13].

To configure our X-shaped drone, we need to install two motors clockwise (CW) and two counter-clockwise (CCW). To achieve this, simply reverse the wire connections between the ESCs and motors 1 and 3, arranged diagonally (Figure 3.29).

Figure 3-32 Assembly of drone motors

we used a PCB to build our circuit (Figure 3.30).

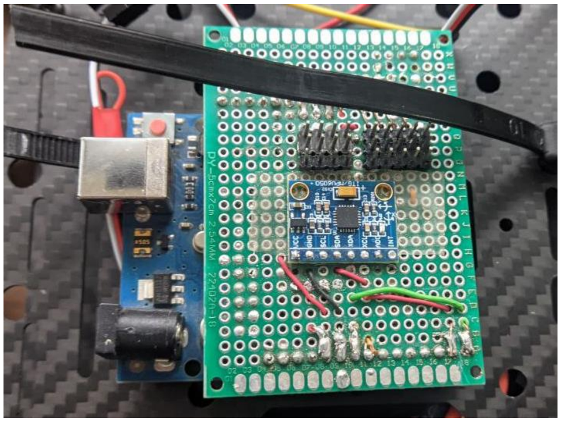


Figure 3-33 Arduino Flight controller

Finally the quadrotor based on the Arduino UNO board has been obtained, shown in figure (3.31).

Figure 3-34 The final assembly of flight controller based on Arduino UNO

3.9.1 Programming

The flight controller is programmed using Arduino IDE. The following steps shows the main codes.

Step one: A Setup code, here we configurate the transmitter and calibrate the MPU6050, and all the date which are PWM signals will be saved on the EEPROM.

```
#include <Wire.h>                //Include the Wire.h library so we can communicate with the gyro
#include <EEPROM.h>              //Include the EEPROM.h library so we can store information onto the EEPROM

//Declaring Global Variables
byte last_channel_1, last_channel_2, last_channel_3, last_channel_4;
byte lowByte, highByte, type, gyro_address, error, clockspeed_ok;
byte channel_1_assign, channel_2_assign, channel_3_assign, channel_4_assign;
byte roll_axis, pitch_axis, yaw_axis;
byte receiver_check_byte, gyro_check_byte;
volatile int receiver_input_channel_1, receiver_input_channel_2, receiver_input_channel_3, receiver_input_channel_4;
int center_channel_1, center_channel_2, center_channel_3, center_channel_4;
int high_channel_1, high_channel_2, high_channel_3, high_channel_4;
int low_channel_1, low_channel_2, low_channel_3, low_channel_4;
int address, cal_int;
unsigned long timer, timer_1, timer_2, timer_3, timer_4, current_time;
float gyro_pitch, gyro_roll, gyro_yaw;
float gyro_roll_cal, gyro_pitch_cal, gyro_yaw_cal;
```

Figure 3-35 Arduino Setup code

Step two: ESC Calibration, here we calibrate the ESC as we explained earlier (Sub-subsection 3.3.4).

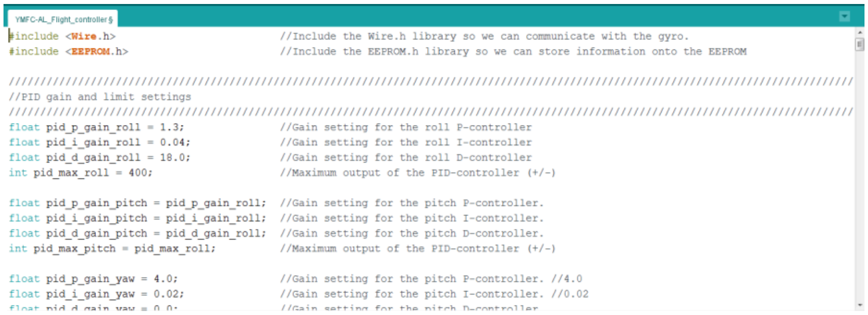
```
#include <Wire.h>                          //Include the Wire.h library so we can communicate with the gyro.
#include <EEPROM.h>                        //Include the EEPROM.h library so we can store information onto the EEPROM

//Declaring global variables
byte last_channel_1, last_channel_2, last_channel_3, last_channel_4;
byte eeprom_data[36], start, data;
boolean new_function_request,first_angle;
volatile int receiver_input_channel_1, receiver_input_channel_2, receiver_input_channel_3, receiver_input_channel_4;
int esc_1, esc_2, esc_3, esc_4;
int counter_channel_1, counter_channel_2, counter_channel_3, counter_channel_4;
int receiver_input[5];
int loop_counter, gyro_address, vibration_counter;
int temperature;
long acc_x, acc_y, acc_z, acc_total_vector[20], acc_av_vector, vibration_total_result;
unsigned long timer_channel_1, timer_channel_2, timer_channel_3, timer_channel_4, esc_timer, esc_loop_timer;
unsigned long zero_timer, timer_1, timer_2, timer_3, timer_4, current_time;

int acc_axis[4], gyro_axis[4];
double gyro_pitch, gyro_roll, gyro_yaw;
```

Figure 3-36 Arduino ESC-calibrate code

Step three: The final code is the result of collection of all those to control the quadrirotor.



```
#include <Wire.h>                     //Include the Wire.h library so we can communicate with the gyro.
#include <EEPROM.h>                   //Include the EEPROM.h library so we can store information onto the EEPROM

///////////////////////////////////////////////////////////////////////////////////////////////////////
//PID gain and limit settings
///////////////////////////////////////////////////////////////////////////////////////////////////////
float pid_p_gain_roll = 1.3;               //Gain setting for the roll P-controller
float pid_i_gain_roll = 0.04;              //Gain setting for the roll I-controller
float pid_d_gain_roll = 18.0;              //Gain setting for the roll D-controller
int pid_max_roll = 400;                    //Maximum output of the PID-controller (+/-)

float pid_p_gain_pitch = pid_p_gain_roll;  //Gain setting for the pitch P-controller.
float pid_i_gain_pitch = pid_i_gain_roll;  //Gain setting for the pitch I-controller.
float pid_d_gain_pitch = pid_d_gain_roll;  //Gain setting for the pitch D-controller.
int pid_max_pitch = pid_max_roll;          //Maximum output of the PID-controller (+/-)

float pid_p_gain_yaw = 4.0;                //Gain setting for the pitch P-controller. //4.0
float pid_i_gain_yaw = 0.02;               //Gain setting for the pitch I-controller. //0.02
float pid_d_gain_yaw = 0.0;                //Gain setting for the pitch D-controller
```

Figure 3-37 Arduino Flight controller code

3.9.2 Tests and Results

3.9.2.1 Transmitter and reciver

To make sure that the communication between RCS and the Flight controller is succed and fairly good we made a test.

After charging the arduino reciver code to the flight controller we open the serial monitor through Arduino IDE,then we move all the joysticks and we see the result.

Table 3.7 Tests of the reciver on Arduino IDE

 Placing the sticks on their meduim position.	Roll:-+-1500 Nick:-+-1500 Gas:-+-1500 Yaw:-+-1500 Roll:-+-1500 Nick:-+-1500 Gas:-+-1500 Yaw:-+-1500 Roll:-+-1500 Nick:-+-1500 Gas:-+-1496 Yaw:-+-1504 Roll:-+-1500 Nick:-+-1500 Gas:-+-1500 Yaw:-+-1500 Roll:-+-1500 Nick:-+-1500 Gas:-+-1500 Yaw:-+-1500 Roll:-+-1500 Nick:-+-1500 Gas:-+-1496 Yaw:-+-1504 Roll:-+-1500 Nick:-+-1500 Gas:-+-1496 Yaw:-+-1504 The sticks values should be in the meduim value.
 Moving the Roll stick to the right	Roll:<<<1132 Nick:-+-1504 Gas:-+-1496 Yaw:-+-1500 Roll:<<<1080 Nick:-+-1500 Gas:-+-1496 Yaw:-+-1500 Roll:<<<1036 Nick:-+-1504 Gas:-+-1496 Yaw:-+-1504 Roll:<<<1000 Nick:-+-1500 Gas:-+-1496 Yaw:-+-1504 Roll:<<<1000 Nick:-+-1500 Gas:-+-1496 Yaw:-+-1504 Roll:<<<1000 Nick:-+-1504 Gas:-+-1496 Yaw:-+-1504 Roll:<<<1000 Nick:-+-1504 Gas:-+-1496 Yaw:-+-1504 Roll:<<<1004 Nick:-+-1500 Gas:-+-1496 Yaw:-+-1504 Roll values changing to the lowest value.
 Moving the Roll stick to the left.	Roll:>>>1996 Nick:-+-1500 Gas:-+-1500 Yaw:-+-1500 Roll:>>>2000 Nick:-+-1500 Gas:-+-1496 Yaw:-+-1504 Roll:>>>2000 Nick:-+-1500 Gas:-+-1496 Yaw:-+-1500 Roll:>>>1996 Nick:-+-1500 Gas:-+-1496 Yaw:-+-1500 Roll:>>>1996 Nick:-+-1500 Gas:-+-1496 Yaw:-+-1500 Roll:>>>1996 Nick:-+-1504 Gas:-+-1496 Yaw:-+-1500 Roll:>>>2000 Nick:-+-1500 Gas:-+-1500 Yaw:-+-1500 Roll:>>>1996 Nick:-+-1500 Gas:-+-1500 Yaw:-+-1500 Roll values changing to the maximum value.
 Moving up the Pitch stick	Roll:-+-1500 Nick:^^^1000 Gas:-+-1500 Yaw:-+-1504 Roll:>>>1540 Nick:^^^1004 Gas:-+-1496 Yaw:-+-1500 Roll:>>>1560 Nick:^^^1000 Gas:-+-1496 Yaw:-+-1504 Roll:>>>1588 Nick:^^^1000 Gas:-+-1500 Yaw:-+-1504 Roll:>>>1540 Nick:^^^1000 Gas:-+-1496 Yaw:-+-1504 Roll:-+-1500 Nick:^^^1000 Gas:-+-1496 Yaw:-+-1504 Roll:-+-1500 Nick:^^^1000 Gas:-+-1496 Yaw:-+-1504 Roll:-+-1496 Nick:^^^1004 Gas:-+-1496 Yaw:-+-1504 Pitch values changing to lowest value.
 Moving down the Pitch stick	Roll:-+-1500 Nick:vvv1988 Gas:-+-1500 Yaw:-+-1500 Roll:-+-1504 Nick:vvv1996 Gas:-+-1500 Yaw:-+-1500 Roll:-+-1504 Nick:vvv2000 Gas:-+-1500 Yaw:-+-1500 Roll:-+-1504 Nick:vvv2000 Gas:-+-1500 Yaw:-+-1500 Roll:-+-1504 Nick:vvv2000 Gas:-+-1496 Yaw:-+-1504 Roll:-+-1504 Nick:vvv2000 Gas:-+-1496 Yaw:-+-1504 Roll:-+-1504 Nick:vvv2000 Gas:-+-1496 Yaw:-+-1504 Roll:-+-1504 Nick:vvv2000 Gas:-+-1496 Yaw:-+-1500 Pitch values changing to the maximum value

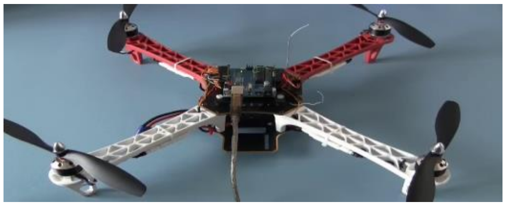
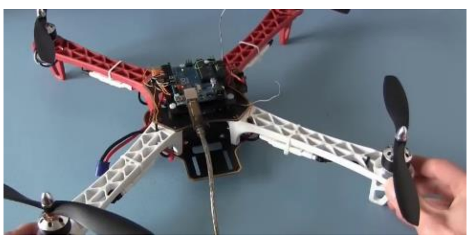
 Moving up the thrust stick	Roll:-+-1504 Nick:-+-1500 Gas:^^^1752 Yaw:-+-1504 Roll:-+-1504 Nick:-+-1500 Gas:^^^1840 Yaw:-+-1500 Roll:-+-1500 Nick:-+-1500 Gas:^^^1940 Yaw:-+-1500 Roll:-+-1504 Nick:-+-1500 Gas:^^^2000 Yaw:-+-1500 Roll:-+-1500 Nick:-+-1504 Gas:^^^2000 Yaw:-+-1500 Roll:-+-1500 Nick:-+-1504 Gas:^^^2000 Yaw:-+-1504 Roll:-+-1500 Nick:-+-1500 Gas:^^^2000 Yaw:-+-1504 Roll:-+-1500 Nick:-+-1500 Gas:^^^2000 Yaw:-+-1504 Roll:-+-1500 Nick:-+-1500 Gas:^^^2000 Yaw:-+-1500 Gas values changing to the maximum value.
 Moving down the thrust stick	Roll:-+-1500 Nick:-+-1500 Gas:vvv1352 Yaw:-+-1500 Roll:-+-1500 Nick:-+-1500 Gas:vvv1160 Yaw:-+-1504 Roll:-+-1500 Nick:-+-1500 Gas:vvv1028 Yaw:-+-1504 Roll:-+-1500 Nick:-+-1500 Gas:vvv1004 Yaw:-+-1500 Roll:-+-1504 Nick:-+-1500 Gas:vvv1004 Yaw:-+-1500 Roll:-+-1504 Nick:-+-1500 Gas:vvv1004 Yaw:-+-1500 Roll:-+-1504 Nick:-+-1500 Gas:vvv1004 Yaw:-+-1500 Roll:-+-1504 Nick:-+-1500 Gas:vvv1004 Yaw:-+-1500 Gas values changing to the minimum value.
 Moving the Yaw stick to the Left.	Roll:-+-1500 Nick:-+-1504 Gas:-+-1500 Yaw:<<<1344 Roll:-+-1500 Nick:-+-1500 Gas:-+-1496 Yaw:<<<1192 Roll:-+-1500 Nick:-+-1500 Gas:-+-1496 Yaw:<<<1100 Roll:-+-1500 Nick:-+-1504 Gas:-+-1492 Yaw:<<<1032 Roll:-+-1504 Nick:-+-1500 Gas:-+-1496 Yaw:<<<1004 Roll:-+-1504 Nick:-+-1500 Gas:-+-1496 Yaw:<<<1004 Roll:-+-1500 Nick:-+-1500 Gas:-+-1496 Yaw:<<<1004 Roll:-+-1504 Nick:-+-1500 Gas:-+-1492 Yaw:<<<1004 Roll:-+-1504 Nick:-+-1500 Gas:-+-1492 Yaw:<<<1004 Yaw values changing to the minimum value.
 Moving the Yaw stick to the Right.	Roll:-+-1500 Nick:-+-1500 Gas:-+-1500 Yaw:>>>1560 Roll:-+-1500 Nick:-+-1500 Gas:-+-1496 Yaw:>>>1704 Roll:-+-1504 Nick:-+-1500 Gas:-+-1496 Yaw:>>>1796 Roll:-+-1500 Nick:-+-1500 Gas:-+-1500 Yaw:>>>1916 Roll:-+-1500 Nick:-+-1504 Gas:-+-1496 Yaw:>>>1980 Roll:-+-1500 Nick:-+-1500 Gas:-+-1500 Yaw:>>>2000 Roll:-+-1500 Nick:-+-1500 Gas:-+-1500 Yaw:>>>2000 Roll:-+-1500 Nick:-+-1500 Gas:-+-1496 Yaw:>>>2000 Yaw values changing to the minimum value.

The ESCs and servo motors are controlled by Pulses from 1000ms to 2000ms (which are a PWM signals) so the pulses changes when we change the stick's position.

3.9.2.2 Gyroscopic calibration

We make this test to know the performance and the sensibility of the MPU6050 gyro sensor, which is on the main components of the stability of the drone.

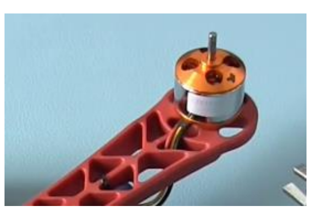
Table 3.8 Test of the calibration of the MPU6050 on Arduino IDE.

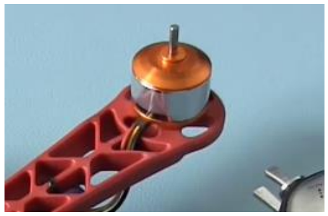
	Starting calibration...................... done! Pitch:-0 --- Roll:+0 --- Yaw:-0 --- Pitch:-0 --- Roll:+0 --- Yaw:-0 --- Pitch:-0 --- Roll:+0 --- Yaw:-0 --- Pitch:+0 --- Roll:+0 --- Yaw:-0 --- Pitch:-0 --- Roll:+0 --- Yaw:-0 ---
	Pitch:-0 --- Roll:-11 RwU Yaw:-0 --- Pitch:-0 --- Roll:-12 RwU Yaw:-0 --- Pitch:-0 --- Roll:-11 RwU Yaw:-0 --- Pitch:+0 --- Roll:-12 RwU Yaw:-0 --- Pitch:-0 --- Roll:-10 RwU Yaw:-0 --- Pitch:-0 --- Roll:-7 RwU Yaw:+0 --- Pitch:-0 --- Roll:-7 RwU Yaw:+0 ---
	Pitch:-8 NoD Roll:-0 --- Yaw:+0 --- Pitch:-7 NoD Roll:-0 --- Yaw:-0 --- Pitch:-8 NoD Roll:-0 --- Yaw:-0 --- Pitch:-5 NoD Roll:-0 --- Yaw:+0 --- Pitch:-4 NoD Roll:+0 --- Yaw:+0 --- Pitch:-3 NoD Roll:+0 --- Yaw:+0 --- Pitch:-1 --- Roll:-0 --- Yaw:-0 --- Pitch:-0 --- Roll:+0 --- Yaw:-0 ---
	Pitch:+0 --- Roll:+0 --- Yaw:-9 NoL Pitch:+0 --- Roll:+0 --- Yaw:-11 NoL Pitch:+0 --- Roll:+0 --- Yaw:-7 NoL Pitch:-0 --- Roll:-0 --- Yaw:-5 NoL Pitch:+0 --- Roll:+0 --- Yaw:-2 --- Pitch:-0 --- Roll:+0 --- Yaw:+0 --- Pitch:-0 --- Roll:+0 --- Yaw:+0 ---

The quad-rotor must be in a flat surface to do the calibration. And the MPU sensor must be placed in an appropriate and flat place on the drone, to see the Roll, Pitch, Yaw in 0 value.

Effect of the motor's vibration on the gyro sensor MPU6050

Table 3.9 Tests of the motor's effect on the MPU6050

Motor OFF		2 3 3 1 2 2 1 1 Autoscroll

Motor ON		17 16 16 19 18 17 18 15 15 16
Motors ON with propellers		85 92 105 109 109 100 97 92 91 81 78 75

There is a lot of gyro sensors, and each one has his specifiction, here we will aim a differnce between MPU6050 and L3GD20H cencerning the effect of the vibration shown on figure(3.28)

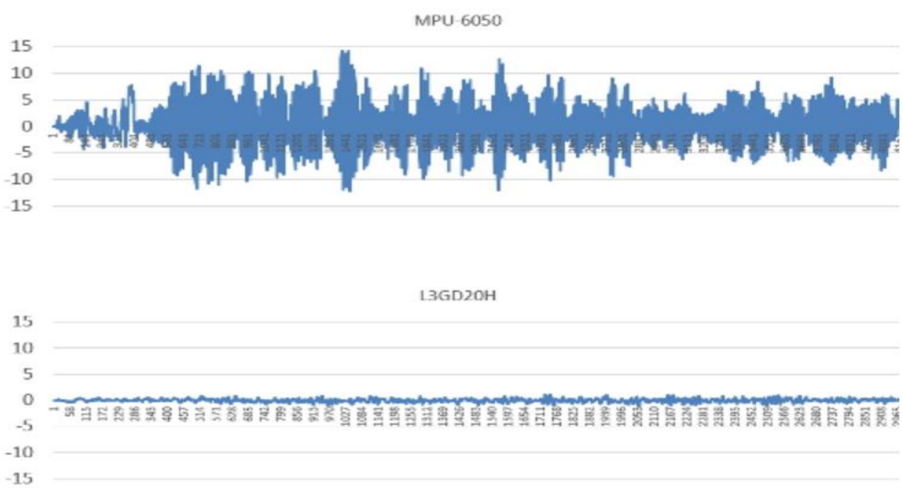


Figure 3-38 Effect of the vibration between MPU6050 and L3GD20H [13].

As we seen in the previous figure, the L3GD20H has a filter of vibration, so it gives better performance than MPU6050.

3.10 Result and explainations

When we fly the drone we got a successful fly and an acceptable performance.

Figure 3-39 Testing the fly of the drone

3.11 RC Trainer plane

This plane is designed specifically to learn how to fly an RC plane and to familiarized the orientation, and for testing the performance of the RCS.

3.11.1 Design of the structure

We mentioned in the first part of this chapter the used components, so according to plans the structure has been designed and installed.

Figure 3-40 RC trainer plane structure

3.11.2 Reciver board

This board for controlling our plane through a transmitter which is the RCS that we realizate before.

It contains an Arduino Nano, and the NRF module.

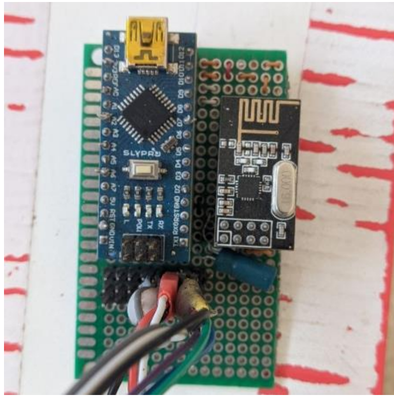
 

Figure 3-41 Reciver of the RC plane. **Figure 3-42.** The BLDC placed of the back of the RC trainer plane

Now we place a BLDC motor on the back present the Thrust, makes the plane move forward.

After this we place two servo motors represents the Pitch movement.

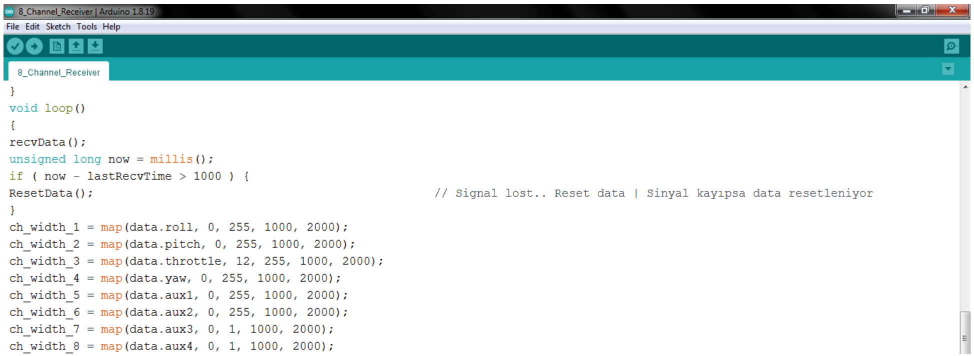
Figure 3-43 Pitch servo motors.

And one servo motors represent the Yaw movement.

Figure 3-44 Yaw Servo motor

3.11.3 Programmtion

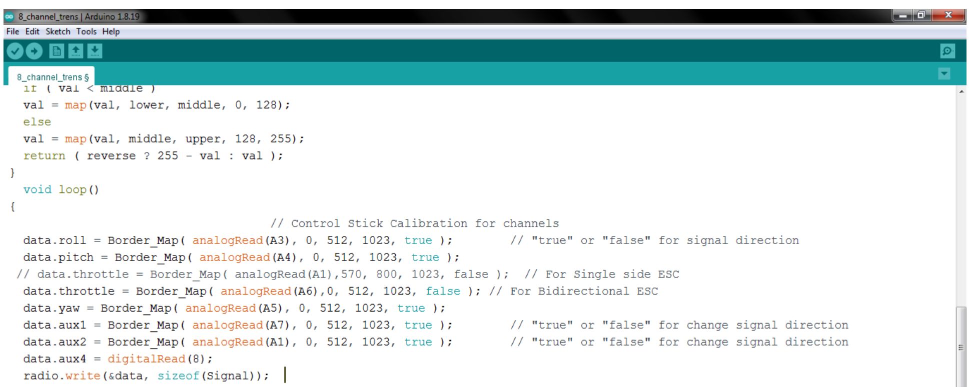
Now we programme the reciver's Arduino according to the servo motors and the BLDC motor, by charging the code in the Arduino IDE.




```
}
void loop()
{
recvData();
unsigned long now = millis();
if ( now - lastRecvTime > 1000 ) {
ResetData();                                    // Signal lost.. Reset data | Sinyal kayıpsa data resetleniyor
}
ch_width_1 = map(data.roll, 0, 255, 1000, 2000);
ch_width_2 = map(data.pitch, 0, 255, 1000, 2000);
ch_width_3 = map(data.throttle, 12, 255, 1000, 2000);
ch_width_4 = map(data.yaw, 0, 255, 1000, 2000);
ch_width_5 = map(data.aux1, 0, 255, 1000, 2000);
ch_width_6 = map(data.aux2, 0, 255, 1000, 2000);
ch_width_7 = map(data.aux3, 0, 1, 1000, 2000);
ch_width_8 = map(data.aux4, 0, 1, 1000, 2000);
```

Figure 3-45 RC trainer plane reciver code

At the same time, we programe the transimtter (RCS), according to this plane.




```
  if ( val < middle )
  val = map(val, lower, middle, 0, 128);
  else
  val = map(val, middle, upper, 128, 255);
  return ( reverse ? 255 - val : val );
}
  void loop()
{
                                    // Control Stick Calibration for channels
  data.roll = Border_Map( analogRead(A3), 0, 512, 1023, true );        // "true" or "false" for signal direction
  data.pitch = Border_Map( analogRead(A4), 0, 512, 1023, true );
 // data.throttle = Border_Map( analogRead(A1),570, 800, 1023, false );  // For Single side ESC
  data.throttle = Border_Map( analogRead(A6),0, 512, 1023, false ); // For Bidirectional ESC
  data.yaw = Border_Map( analogRead(A5), 0, 512, 1023, true );
  data.aux1 = Border_Map( analogRead(A7), 0, 512, 1023, true );        // "true" or "false" for change signal direction
  data.aux2 = Border_Map( analogRead(A1), 0, 512, 1023, true );        // "true" or "false" for change signal direction
  data.aux4 = digitalRead(8);
  radio.write(&data, sizeof(Signal));
```

Figure 3-46 RC trainer plane transimmter code

3.11.4 Test :

Figure 3-47 Tests of RC Trainer plane

3.12 VTOL Drone

We can say on this plane that it is a mix of the RC trainer plane and the quad-rotor drone. We controll it via the RCS, it carry the fire extinguishing system to carry out its mission.

3.12.1 Designe of the structure

We mentioned in the first part of this chapter the used components, so according to plans we designed and installed the structure, this drone is distinguished by its large size and heavy weight compared to previous drones.

Figure 3-48 VTOL's structure

3.12.2 Placing the motors

Since this is a mix of quad-rotor drone and RC trainer plane, so we place the four BLDC motors as a X shape, but this drone has a heavy weight, so we try to double the motors and that is by placing another motor under the motor is already placed. to make the motors can carry the plane and make it fly as shown in figure 3.46.

Figure 3-49 Doubling the BLDC motors

We place another BLDC motor in the back of the VTOL. So after the vertical take-off, this BLDC motor has a major role for horizontal flight.

Figure 3-50 BLDC motor for the thrust

3.12.3 Flight Controller

The flight controller is for controlling the plane and makes it stable, we used the APM

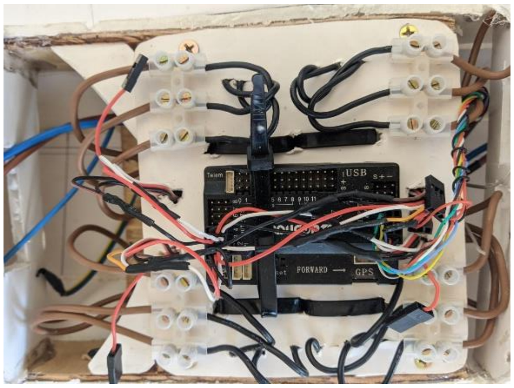


Figure 3-51 VTOL's Flight controller

3.12.3.1 Programming and calibration

The APM flight controller is already programmed and we should calibrate it using MISSION PLANER software.

For the first time, the accelerometer could not determine the direction of the quad-copter. Fortunately, there is an arrow at the top of the APM plate to easily identify the orientation of the plate as well as facilitate the calibration steps.

Figure 3-52 Accelerometer Calibration

Compass calibration is necessary because the developed code relies on a sensors fusion algorithm that uses the compass and GPS to determine attitude and altitude. Like the accelerometer, the compass must be calibrated the first time through the APM mission planning software.

Figure 3-53 Internal Compass Calibration

The APM is connected to the channels of the RC receiver. The Mission Planner software provides the necessary steps for RC calibration. During calibration, the joysticks must be placed in their maximum and minimum positions in order to record the endpoints of the PWM signal. At the end of the calibration, the APM mission planning software provides a table with all the channel endpoints reached. This table can be used to check the operation of the RC.

Figure 3-54 RC Calibration

3.12.4 Fire extinguishing system

This system placed under the VTOL, used for extinguishing the fire and achieve the mission of the aircraft. It composed of a rocket structure shows in figure 3.55 and a fire ball, and voltage amplifier and gunpowder.

Figure 3-55 Rocket structure

3.12.4.1 Operating principal

From the HMI interface, we press a button to open the relay to activate the voltage amplifier, the latter it burns the gunpowerwhich gives momentum to the fire ball for the purpose of launching it towards the fire in order to extinguish it.

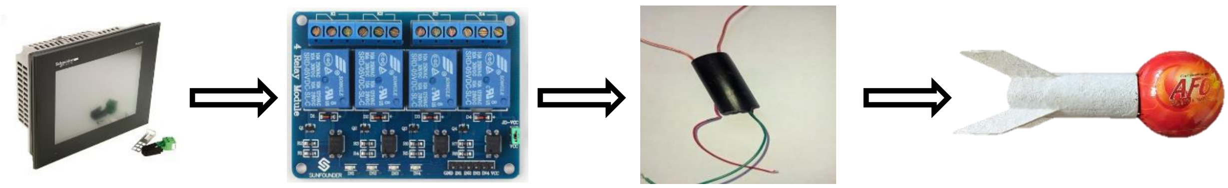


Figure 3-56 A diagram showing the working principle of the fire extinguishing system

3.13 Conclusion

This chapter discusses the realization of the Radio command station, the drone, VTOL, and the RC trainer plane, and its results and simulations , after which the elements used in designing the quadrotor and the VTOL and the RC Trainer plane are introduced. Then the design of the radio control is explained. Next, we have provided details about the basic assembly that must be completed in advance, so that everyone can communicate. We then discussed how to calibrate a quad-copter, which we were able to test and found to be stable in flight.

General conclusion

Conception and implementation of drones are realized in this project. For this kind of projects, specific materials and advanced programming are required.

Development of Quad-copter systems need to master several fields such as mechanical design, aviation, information technology, automation, electronics, and communications. In this project, we have developed a mathematical model of an aerobic vehicle then we turned it in reality.

This work has been divided into two parts: the first is a literature review about drone control techniques, and the second concernes hardware implementation using particular electrical components.

Realization of this work needs hard work in terms of theoretical study and in terms of application. Despite the limited time and lack of capabilities, we were able to achieve the desired goal, which is to embody a network of aircraft that can extinguish fires. The obtained results demonstrate taht the drone keeps its stability during the flight testing. As a result, various tests were carried out in the laboratory under precise conditions.

As a future work, adding the GPS module to facilitate the drone displacement from a place to another. Also we suggested to use the STM32 module in place of Arduino because it integrates GPS module.

Implementing an obstacle avoidance system employing a distance sensor, such as an infrared sensor, sonar, or vision system, is another significant choice to be mentioned.

Concerning the Fire extinguishing VTOL, adding thermal camera to preview the apartment where there was a fire before extinguishing.

Using LoRa module instead the NRF for increasing the communication distance.

The ability to send the drone on a mission after specifying the route represents significant advancement.

BIBLIOGRAPHY

BIBLIOGRAPHY

[1]. C. Delmotte : « La thermographie infrarouge », Laboratoire Qualité de l'Air et Ventilation CSTC - Centre Scientifique et Technique de la Construction, 1998

[2]. H. González-Jorge, J. Martínez-Sánchez, M, Bueno: « *Unmanned aerial systems for civil applications: A review* », Drones, 1, p.2, 2017.

[3]. Bouabdallah, S: « PID vs LQ control techniques applied to an indoor micro quadrotor ».2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) , Vol. 3, pp. 2451-2456, 2004

[4]. Chen, X., Zhang, G., Lu, C. and Cheng, J: « Quadrotor aircraft attitude control algorithm based on improved UKF". In IOP Conférence Séries : Earth and Environmental Science (Vol. 233, No. 4, p. 042037), 2019.

[5]. H. Khebbache, Tolérance aux défauts via la méthode backstepping des systèmes non linéaires, Thèse de magister,Université de sétif,2012.

[6]. L. Gautier, B. Hamilton, J. Hazebrouck, and T. Tourrette "Pilotage d'un quadri rotor via un FPGA", Tuteur ESIEE engineering, Journées pédagogiques, 2008

[7]. T.Perez. *Ship Motion Control :Course Keeping and Roll Stabilisation using Rudder and Fins.Springer-Verlag*, 2005.

[8]. Robotshop Platforme, Robotshop website, Manuel étulisation.

[9]. Randal Beard, "Quadrotor Dynamics and Control Rev 0.1," 2008.

[10]. A. Chen, «Fire extinguisher balls,» 2023. [En ligne]. Available: https://afoball.com/fire-extinguisher-balls/.

[11]. Arduino.cc, «Arduino Integrated Development Environment (IDE) v1,» Arduino, 19 09 2023. [En ligne]. Available: https://docs.arduino.cc/software/ide-v1/tutorials/arduino-ide-v1-basics.

[12]. Vijeo Designer Didacticiel Version 5.1 , 01/2010

[13]. J. Brokking, «Brokking.net,» Project YMFC-3D, 30 4 2017. [En ligne]. Available: http://www.brokking.net/ymfc-3d_main.html.

[14]. Mahony, R., Kumar, V., & Corke, P: « *Multirotor aerial vehicles: Modeling, estimation, and control of quadrotor* ». IEEE Robotics and Automation magazine, *19*(3), 20-32, 2012

APPENDIX

APPPPENDIX A

Rotation Matrix

First rotation

The first rotation is a rotation of ψ around the *Z0* axis which makes the *Y0* axis coincides with the *Y'* axis and the *X0* axis with the *X'* axis. The rotation matrix is given by the following equation.

$$R\psi = \begin{bmatrix} cos(\psi) & -sin(\psi) & 0 \\ sin(\psi) & cos(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Second rotation

The second rotation is a rotation of θ around the Y' axis which makes the X' axis coincide with the X1axis and the Z0 axis with the Z' axis. The rotation matrix is given by the following equation.

$$R\theta = \begin{bmatrix} cos(\theta) & 0 & sin(\theta) \\ 0 & 1 & 0 \\ -\,sin(\theta) & 0 & cos(\theta) \end{bmatrix}$$

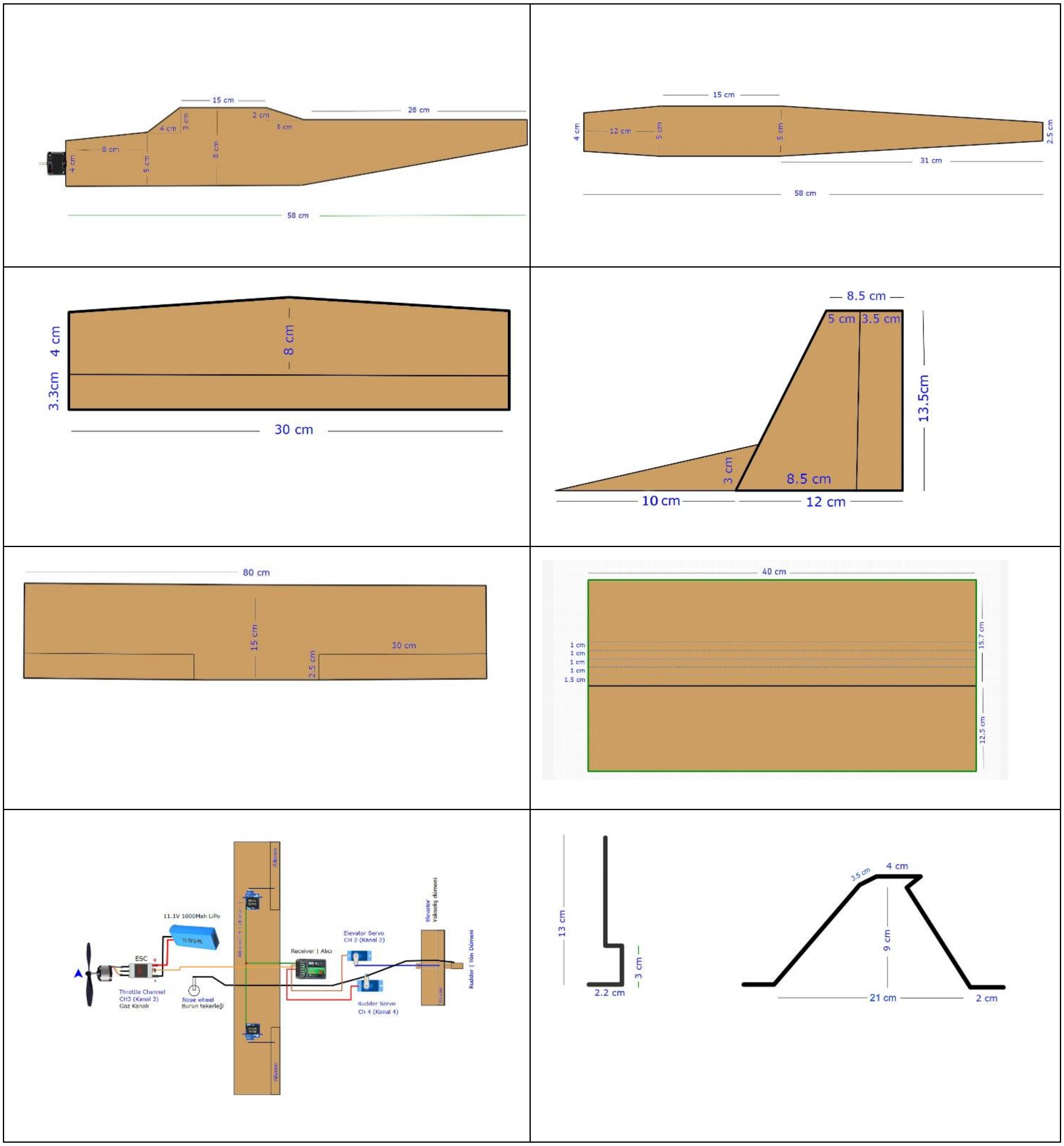
Third rotation

The third rotation is a rotation of φ around the X1 axis which makes the Z' axis coincides with the Z1axis and the Y' axis with the Y1 axis. The rotation matrix is given by the following equation.

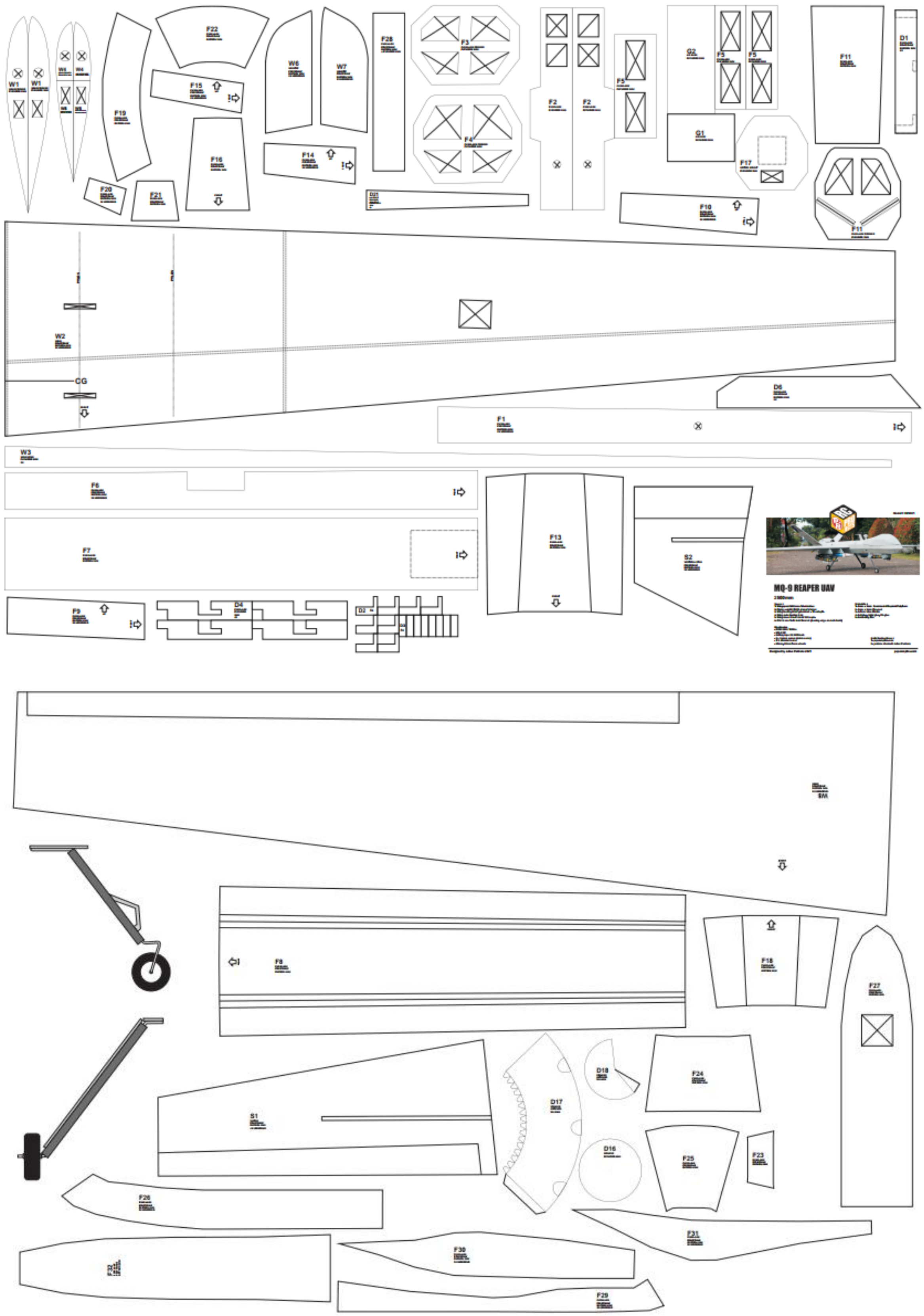
$$R\varphi = \begin{bmatrix} 1 & 0 & 0 \\ 0 & cos(\varphi) & -sin(\varphi) \\ 0 & sin(\varphi) & cos(\varphi) \end{bmatrix}$$

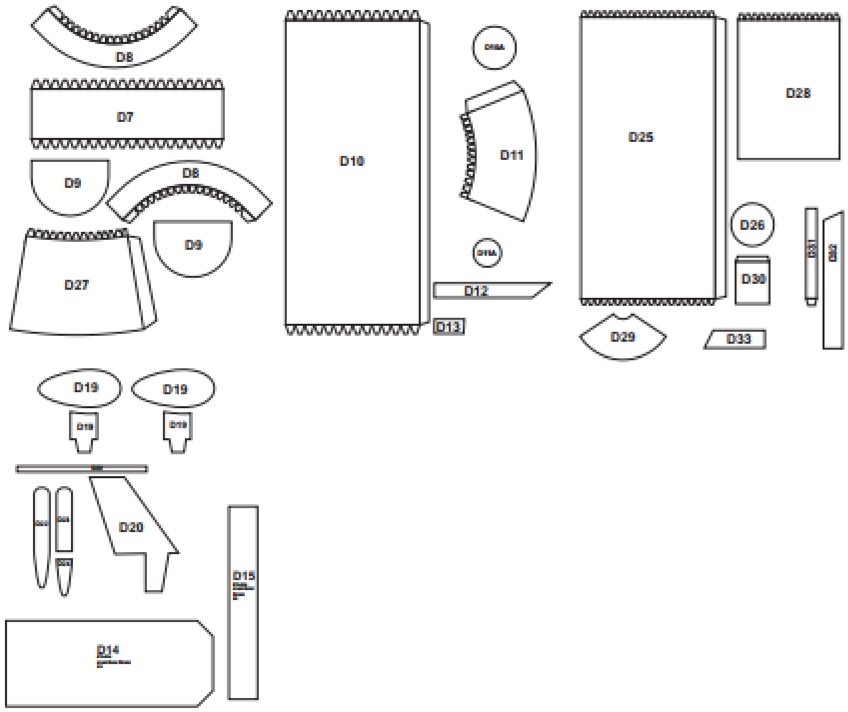
APPPPENDIX B

RC Trainer plane Structure's plane and demensions



VTOL Structure’s plan




D8
D7
D9
D8
D9
D27
D10
D11
D12
D13
D25
D28
D26
D30
D29
D33
D19
D19
D20
D15
D14