Master thesis : Building a drone from scratch: Conception, modelling and optimization of a hexadrone

Auteur : Harmel, François
Promoteur(s) : Redouté, Jean-Michel
Faculté : Faculté des Sciences appliquées
Diplôme : Master : ingénieur civil électricien, à finalité spécialisée en "signal processing and intelligent robotics"
Année académique : 2021-2022
URI/URL : http://hdl.handle.net/2268.2/14397

University of Liège
School of Engineering and Computer Science

ATFE0014-1 Master thesis

Building a drone from scratch: Conception, modelling and optimization of a hexadrone.

Author
François HARMEL

Master's thesis carried out to obtain the degree of Master of Science in Electricity Engineering

Academic year 2021-2022

Abstract

Building a hexadrone from scratch to meet certain constraints is not an easy task, especially for people who are not initially trained in this field. However, this is the crazy bet that has been followed in this master thesis.

The first main goal of this paper is to explain both from a theoretical and a practical point of view what choices were made throughout the year from a design and a construction point of view.

The second goal of this work was to verify if it was possible to determine beforehand what would be the final behaviour of the UAV (e.g. flight time) based on data provided by the manufacturers. Therefore, it was a question of mixing the theoretical with the experimental as well as the practical in order to disentangle the true from the false.

The third and last goal of this work was to realize a collaboration between 2 students: Luca Delfino, electromechanic at Henallux and François Harmel, electrical engineer at ULiège in order to increase the competence of the team

The results obtained after one year of work are satisfactory the practical realization meets quite well the theoretical behaviors that were planned, it is thus a total success

List of Figures

Contents

Nomenclature

Abbreviations

AC	**A**lternative **c**urrent
BCM	**B**roadcom **c**hip-specific pin nu**m**bers
BEC	**B**attery **e**liminator **c**ircuit
BLDC	**B**rush**l**ess **d**irect **c**urrent
CCW	**C**ounter-**c**lock**w**ise
CW	**C**lock**w**ise
DC	**D**irect **c**urrent
DSM	**D**igital **s**ystem **m**ultiplexer
DSSS	**D**irect-**s**equence **s**pread **s**prectrum
EES	**E**ngineering **e**quation **s**olver
ESC	**E**lectronic **s**peed **c**ontroller
FC	**F**light **c**ontroller
FHSS	**F**requency-**h**opping **s**pread **s**prectrum
FPV	**F**irst **p**erson **v**iew
GPIO	**G**eneral **p**urpose **i**nput/**o**utput
GPS	**G**lobal **p**ositioning **s**ystem
HX	**H**ybrid **x**
IBUS	**I**ntelligent input **bus**
IMU	**I**nertial **m**easurement **u**nit
LiHV	**Li**thium-polymer **h**igh **v**oltage
LiPo	**Li**thium **po**lymer
MCU	**M**icro **c**ontroller **u**nits
MOSFET	**M**etal **o**xide **s**emiconductor **f**ield-**e**ffect **t**ransistor
OSD	**O**n-**s**creen **d**isplay
P	**P**arallel
P	**P**roportional
PDB	**P**ower **d**istribution **b**oard
PID	**P**roportional **i**ntegrative **d**erivative
PLA	**P**oly**l**actic **a**cid
PPM	**P**ulse **p**osition **m**odulation
PWM	**P**ulse **w**idth **m**odulation
PX4 mini	**P**i**x**hawk **4 mini**
QGC	**QG**round**C**ontrol
RPi	**R**aspberry **pi**

RPM	**R**otation **p**er **m**inute
RTF	**R**eady **t**o **f**ly
RTL	**R**eturn **t**o **l**aunch
RX	**R**eceive
S	**S**eries
SBUS	**S**erial **bus**
TX	**T**ransmit
UART	**U**niversal **a**synchronous **r**receiver/**t**ransmitter
UAV	**U**nmanned **A**erial **V**ehicle
Wi-Fi	**Wi**reless **fi**delity

Variables

δ	Duty cycle of PWM command	$[-]$
δ_{motor}	Duty cycle of the motor	$[-]$
δ_{PWM}	Normalized duty cycle of the software PWM signal	$[-]$
η_{esc}	Efficiency of the ESC	$[-]$
η_{motor}	Efficiency of the motor (without considering rotor)	$[-]$
η_{rotor}	Efficiency of the rotor (without considering motor)	$[\mathrm{g} \cdot \mathrm{W}^{-1}]$
η_{syst}	Efficiency of pair motor + rotor	$[\mathrm{g.W}^{-1}]$
κ	Induced power correction factor	$[-]$
$\omega_{motor,no-load}$	Rotational speed of the non-loaded motor	$[\mathrm{RPM}]$
ω_{motor}	Rotational speed of motor	$[\mathrm{RPM}]$
Ω_{rotor}	Rotational speed of the rotor	$[\mathrm{s}^{-1}]$
ρ_{air}	Air density	$[\mathrm{kg} \cdot \mathrm{m}^{-3}]$
ρ_{cf}	Carbon fiber density	$[\mathrm{g} \cdot \mathrm{cm}^{-3}]$
ρ_{fluid}	Fluid mass density	$[\mathrm{kg} \cdot \mathrm{m}^{-3}]$
ρ_{PLA}	PLA density	$[\mathrm{g} \cdot \mathrm{cm}^{-3}]$
σ_{rotor}	Rotor solidity	$[-]$
τ	Torque produced by motor	$[\mathrm{N} \cdot \mathrm{m}]$
A	Rotating disk area	$[\mathrm{m}^2]$
C_{bat}	Battery capacity	$[\mathrm{mAh}]$
c_b	Chord length of the blade	$[\mathrm{m}]$
C_d	Drag coefficient	$[-]$
$C_{P,drag}$	Power coefficient with drag effect	$[-]$
$C_{P,id}$	Ideal power coefficient	$[-]$
$C_{P,non-id}$	Power coefficient with non ideal effect	$[-]$
C_T	Thrust coefficient	$[-]$

CR_{bat}	Battery c-rating	[h^{-1}]
E	Motor voltage	[V]
F_{WL}	Wind loading	[$\mathrm{N} \cdot \mathrm{m}^2$]
I	Current	[A]
I_0	Idle current	[A]
I_b	Battery current	[A]
I_{esc}	Current at the output terminals of the ESC	[A]
I_m	Motor current	[A]
I_{safe}	Maximum safe current draw of a battery	[mA]
K_e	Back-EMF constant of motor	[$\mathrm{V} \cdot \mathrm{RPM}^{-1}$]
K_t	Torque constant of motor	[$\mathrm{V} \cdot \mathrm{RPM}^{-1}$]
K_v	Velocity constant of motor	[$\mathrm{RPM} \cdot \mathrm{V}^{-1}$]
K_v^*	Velocity constant of motor, considering η_{motor}	[$\mathrm{RPM} \cdot \mathrm{V}^{-1}$]
$length_i$	length of component $\in$ bottom plate (theoretical design)	[mm]
N_b	Number of blades	[−]
N_{cells}	Number of cells allowed for the LiPo battery pack	[−]
N_{motor}	Number of motor	[−]
nb_{arms}	Number of arms (theoretical design)	[−]
P_b	Battery power	[W]
P_{drag}	Power provided by the rotor, considering drag effect	[W]
P_{elec}	Electrical power	[W]
P_{id}	Power provided by the rotor, in ideal case	[W]
P_{mech}	Mechanical power	[W]
P_{motor}	Power consumed by motor	[W]
P_m	Power developed by the motor	[W]
P_{non-id}	Power provided by the rotor, considering drag and non-ideal effects	[W]
$pulse_{duration}$	Duration of a pulse width signal	[ms]
R	Rotor radius	[m]
R_c	Connection resistance	[Ω]
R_c	Winding resistance	[Ω]
R_m	Motor resistances	[Ω]
S_f	Security factor (thrust to weight ratio)	[mm]
s_{rotor}	Speed of rotor	[$\mathrm{m} \cdot \mathrm{s}^{-1}$]
s_{sound}	Speed of sound	[$\mathrm{m} \cdot \mathrm{s}^{-1}$]
$size\_plate$	Size of the bottom plate (theoretical design)	[mm]
$T_{all_{r}otor}^{(g)}$	Thrust, in g, provided by all rotors (theoretical design)	[g]
$T_{rotor}^{(g)}$	Thrust provided by rotor, expressed in g	[g]

$T_{rotor}^{(N)}$	Thrust provided by rotor, expressed in N	[W]
$t_{discharge}$	Discharge time of the battery	[h]
t_{fly}	Flying time of the drone	[min]
$t_{h\_to\_min}$	Conversion factor from hour to minutes	[−]
T_{PWM}	Time period of the software PWM signal	[ms]
V	Voltage	[V]
V_{arm_design}	Volume of one single arm (theoretical design)	[cm^3]
V_b	Battery voltage	[V]
V_{esc}	Voltage at the output terminals of the ESC	[V]
V_{frame_hx4}	Volume of frame, considering 4 motors (theoretical design)	[cm^3]
V_{frame_hx6}	Volume of frame, considering 6 motors (theoretical design)	[cm^3]
V_{plate_design}	Volume of the top and bottom plate (theoretical design)	[cm^3]
$W_{drone\_component}$	Weight of the components composing the drone, without frame	[g]
$W_{payload}$	Weight of the payload (theoretical design)	[−]
W_{uav}^{g}	Weight of drone	[g]
W_{uav}^{N}	Weight of drone	[N]
$width_i$	Width of component $\in$ bottom plate (theoretical design)	[mm]
DUTY_INC	Increment between 2 consecutive duty cycle values of the software PWM signal	[−]
DUTY_MAX	Maximum duty cycle value of the software PWM signal	[−]
DUTY_MIN	Minimum duty cycle value of the software PWM signal	[−]
DUTY_MOTOR	General duty cycle of the software PWM signal	[−]
FREQUENCY	Frequency of the software PWM signal	[Hz]

Part I

General introduction and global structure

1 Context and description of the project

Who has never dreamed of picking up a remote control and flying through the air with a drone? Drones are becoming more and more a part of our daily lives and are playing an increasingly important role in sectors such as safety and agriculture. For example, Brussels airport tested last year the deployment of drones on its runways in order to determine whether they could be used to control the air traffic [1].

Over time, since the first drone appeared in 1839 [1], technology has continued to evolve and revolutionize existing inventions. Today, engineers design drones, equipped with various sensors, to meet various tasks such as autonomous flight or obstacle detection and avoidance in order to make them more efficient. Drones are now able to perform very diversifying tasks. For instance, drones can be used to optimize the management of agricultural plots[3]. Indeed, the cultivated plots can be very large, which complicates the task of the farmer as it becomes difficult for him to be aware of everything that happens on his plot. Drones, thanks to sensors mounted on them, can offer him an aerial view of his plot from which he will be able to draw maps of the vegetation's growth. Basically, regardless of whether it is for recreational, commercial or military purposes, drones are starting to become a full-fledged player in our modern society.

The main goal of this master thesis was to build a drone capable of taking off and hoovering with a robust control system. The idea behind it was to check if one could model beforehand the characteristics of the drone (here, mainly the flight time and the payload) and bring a design that would allow to reach the predicted objective. To achieve this, there were several key steps, listed below:

- Learning about the world of drones (bibliographical research) in order to answer questions such as: What are the key components in a drone? How do they all interact? What are the equations that govern the flight time of the drone?
- Building a small prototype able to control a motor. It was necessary to understand what was the role played by the ESC [2] as well as the hidden protocol behind that allows it to control the motor.
- Carrying out experimental measurements using the prototype previously built in order to collect useful data on the engine such as the power consumed or the thrust it could generate. These experimental data will be compared to the theoretical data (datasheet provided by the producer as well as a theoretical model representing the interaction of the ESC with the motor). The goal was to verify if there was a difference between the theory and the experimental data in order to guarantee that the specifications envisaged for the drone were respected.
- Sectioning a "kit" drone in order to have a first practical experience (learning by doing). Once the kit was built, it was necessary to be able to answer the following questions: How to set up the remote control in order to be able to successfully control the drone? What were the differences between the different flight modes available? How the different parameters of the control system behave?
- Theoretically designing the final drone that was to be built based on the experience gained on the kit. Once the theoretical design was finalized and some constraints were defined (flight time and payload), the plans of the drone had to be drawn in order to machine the different parts of the frame (i.e. the main body of the drone). The final design has been enforced by several theoretical calculations and practical experiments.
- Assembling the frame and integrating the electronics on the drone. The aim was to optimize the distribution of weight and space within the drone in order to balance it as much as possible, which then allows a better control of the drone.

---

[1] "In 1989, Austrian soldiers used unmanned balloons filled with explosives to attack the city of Venice", as stated in [2].

[2] "ESC" stands for Electronic Speed Controller.

- Launching the drone and measuring the flight time in order to check if the theory met the practice in order to conclude the whole work.

2 Structure of this paper

In this paper, the structure that has been chosen is conceptual more than chronological.

First, a global description of a drone will be given. In this part, all the main components will be listed and, for each of them, most of the possible aspects will be discussed, as well as the advantages and disadvantages that it implies.

Second, the experimental measurement will be explained: the experimental prototypes built and the importance of making experimental measurements will be described. Of course, we will also expose the collected data and will explain what can be drawn from it.

Third, we will expose all the experience gained on the learning kit and discuss control system and flight mode.

Fourth, we will describe the theoretical design that has been made based on the specifications previously decided.

Fifth, the goal of this project was not to just develop hard skills. Indeed, this large project required to learn the basics of project management and teamwork. A word will therefore be given about this, which includes the project planning and the organisation during the year.

Finally, a final word will summarise the learning outcomes of this project (hard skills and soft skills) and suggest further improvements.

NB: In order to get the meaning of any abbreviations or variables (and their units), please refer to the nomenclature.

Part II

Drone description

This part is focused on the complete drone description (general state of the art). It starts by giving a general definition of drones. Then, starting from a picture of a drone, each component will be discussed on several insightful aspects.

1 Definition

Before going into the components description, it would be interesting to look at the following definition of a drone [4].

Definition 1. Unmanned aerial vehicle

An unmanned aerial vehicle (UAV), commonly known as a drone, is an aircraft without any human pilot, crew, or passengers on board. UAVs are a component of an unmanned aircraft system (UAS), which includes adding a ground-based controller and a system of communications with the UAV. The flight of UAVs may operate under remote control by a human operator, as remotely piloted aircraft (RPA), or with various degrees of autonomy, such as autopilot assistance, up to fully autonomous aircraft that have no provision for human intervention.

From this definition, we can understand that the flight system that will be built will not have a pilot on board. The only interactions that we will have with the drone will be made through a remote control. So, we ought to make sure that the system is safe before making it fly (what would happen if the communication with the drone is lost after it has taken off?).

2 Components

The main components of a drone can be seen in Figure 1. The following components can be found (alphabetical order):

- Control receiver,
- Electronic speed controller (ESC),
- Flight controller,
- Frame,
- Motor,
- LiPo battery,
- Power distribution board (PDB),
- Propellers/rotor,
- Remote control.

Each of the elements listed above is essential for the proper functioning of the drone: providing that only one of these elements has a problem, then the whole drone can suffer from a malfunction, a failure or even an accident!

We have to know what roles these different components play in order to fully understand how a drone works in its entirety.

The following sections present each of these elements, without any order defined beforehand. The differences between the models that can be found for the same component will also be discussed.

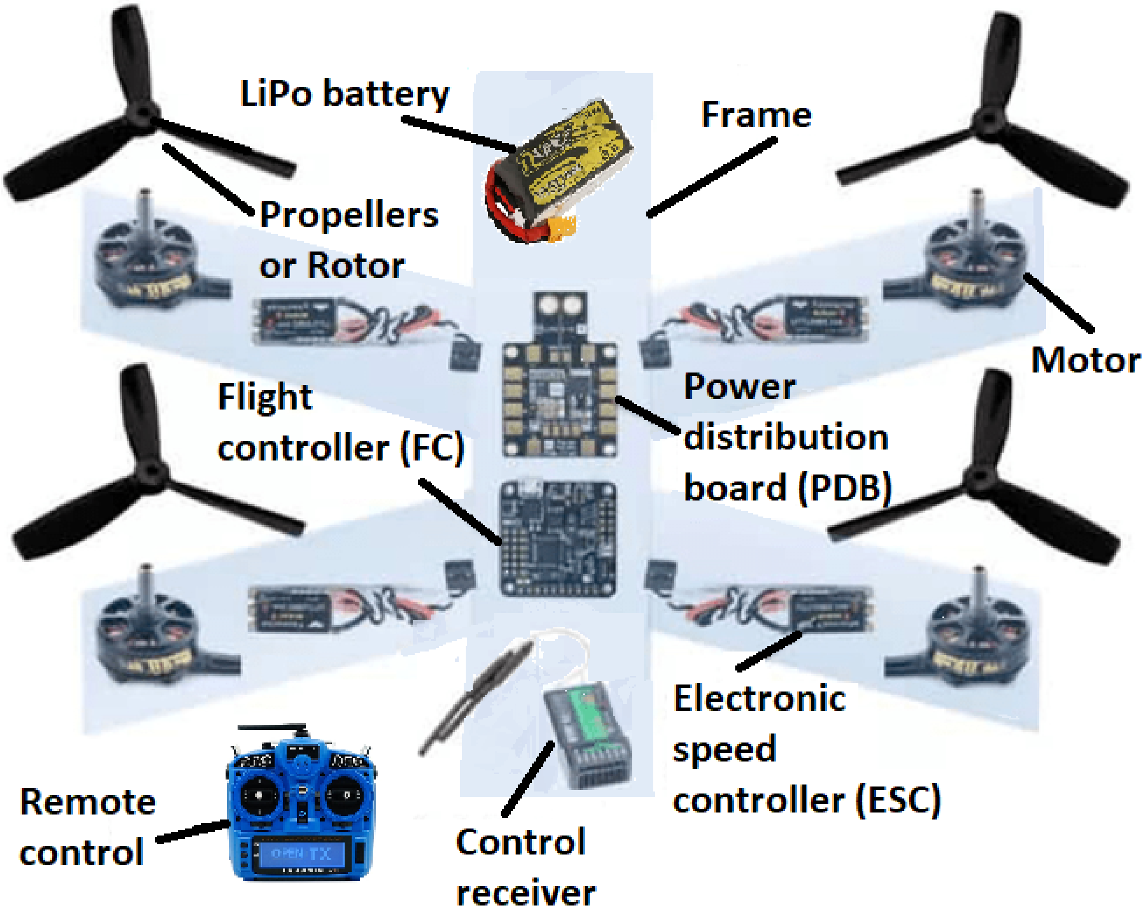


Figure 1: Overview of the main components of a drone.
Sources: https://ebrary.net/182376/health/needs, https://tinyurl.com/2kxhcby5, https://tinyurl.com/2hu3pytp

2.1 Frame

In this section, the frame of the drone will be discussed. A home-made definition of the frame is written below and a typical frame is illustrated in Figure 2.

Definition 2. Drone frame

The frame is generally composed of three elements:

- A top plate
- A bottom plate
- Arms

Note that, in some designs, the two plates can be merged into a single plate.

The main goal of the frame is to hold all the other components together. It also allows to protect some critical components from vibrations (see section 2.8.4). In summary, it acts as the drone's skeleton.

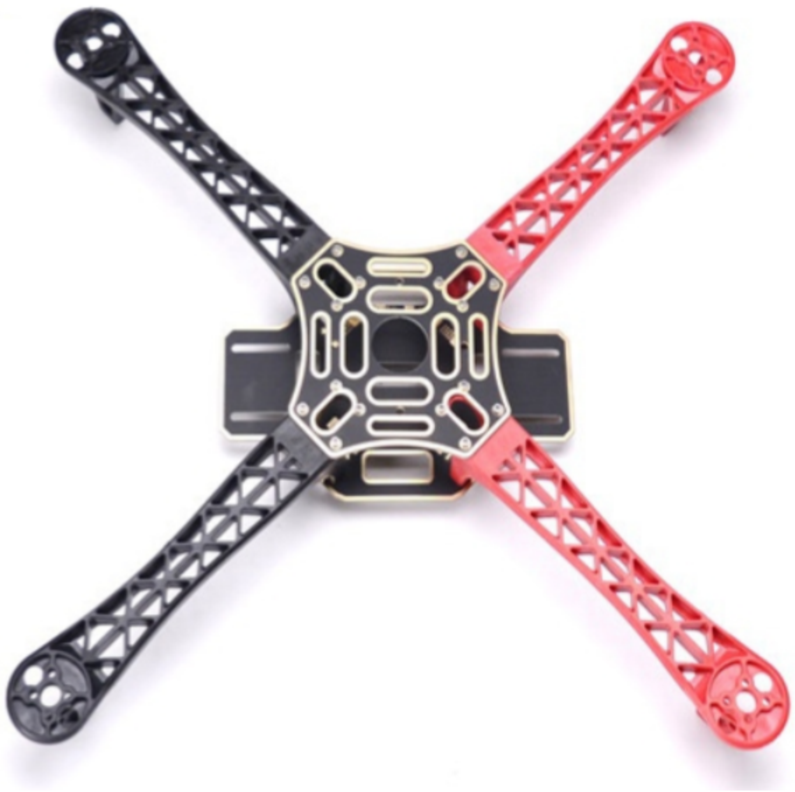


Figure 2: Typical frame of a drone: it is composed of a top plate, a bottom plate and independent arms.
Sources: https://tinyurl.com/44vu2z68

In the following, the frame composition, shape, size and the materials that can be used to build it will be discussed. They must be carefully chosen because they directly impact the drone's performances.

2.1.1 Composition

As said in Definition 2, the frame is composed of two plates and several arms[3].
The electronics onboard is fixed thanks to the plates. They usually contain holes that allow the passage of cables or straps (fixing elements).
The arms may be part of the bottom plate (in that case, it is known as unibody design) or they can be independent. The main advantage of the unibody design is that the frame will be more resistant because the frame will have fewer junctions (therefore less weak spot). However, if one of the arms is damaged in a crash, then the whole frame has to be rebuilt instead of just changing the defective arm. The goal of the arms is to hold motors (or motor mounts) and sometimes ESC.

2.1.2 Material

The frame must be strong enough to withstand any impact. Knowing that the frame of the drone is often the largest part of the drone, it is necessary to try to make it as light as possible. So a first compromise appears between solidity and weight. Choosing the right material is therefore critical to achieving the desired properties.
The Table 1 shows a comparison between the usual materials used to build a frame.

By looking at the Table 1, we can conclude that drones made of carbon fibre will have better performance in terms of stiffness/density, but will also cost more. It is important to know, however, that this material

---

[3] The number of arms gives its name to the drone. For instance, a quadcopter has four arms, while a hexacopter has six arms.

	Carbon fiber [5]	Plastic (pla) [6]	Wood [7] [8]
Rigidity (E in Gpa)	Very rigid: 240	Flexible: 0.037	Variable: Balsa: 3.71 Cherrybark oak: 15.7
Density (ρ in g/cm^3)	Light: 1.8	Very light: 1.25	Variable: Balsa: 0.15 Cherrybark oak: 0.78
Cost	Expensive	Cheap	Cheap
Machining	More difficult to work: requires some tooling (grinder)	3D Printing	Easy to work with woodworking tools (saw)
blocks radio signal	Does	Does not	Does not
Additional remark	Conducts electricity (insulate wire)	/	Good vibration dampener

Table 1: Comparison of the properties of the three most commonly used frame materials.

conducts electricity, so the wires must be well insulated. In addition, the communication antennas on the board must be well positioned because carbon fibre also blocks radio waves. Carbon fibre can be used to make fairly light but very robust drones.

Wood is a material that is often used by beginners as it is better known to the general public. The cost of the raw material is affordable for the average person. The main difficulty lies in the type of wood you choose. The physical properties of the drone will vary greatly from one type of wood to another. Both lightweight and rigid drones can be made from wood.

3D printing PLA plastic is a very good alternative that can be used, provided you have a 3D printer of course. Knowing about technical drawing is needed but otherwise it is an easy material to work with as the printer does all the work. Very light but not very strong drones can be made with PLA.

NB: In this section, only the three most common materials used to make frames have been discussed, but this is a non-exhaustive list. Almost any material can be used to make a frame. Aluminium, fiberglass, and other materials could also be mentioned for instance.

2.1.3 Shape

Any shape can be given to the frame but there are standard configurations, which will be discussed in this section.

The shape of the frame influences the dynamic performance of the drone [9]. Indeed, the center of gravity affects the dynamics of the drone and its control [9]. To better understand this concept, the best scenario is illustrated in Figure 3. In this picture, the center of gravity C_G (see definition 3) corresponds to both the center of mass C_M (see definition 4) and the center of thrust C_T (see definition 5). In this scenario, all the motors (two in this case) act equally when generating the global thrust, resulting in a perfect control. Controversially, in another scenario where the frame is not perfectly symmetrical, C_M would differ from C_G, meaning that the motors would not have the same authority/contribution in the global thrust.If all the motors share the same thrust level, then the macroscopic effect of this scenario would be that the drone flips, resulting in a general loss of control of the drone. Of course, if the gap between C_M and C_M remains small, then the integrative term of the PID controller should be able to compensate for this effect by tracking the error over time. In conclusion, it is good practice to come up with a design where $C_M = C_G = C_M$.

Definition 3. Center of gravity [10]

Centre of gravity is the point at which the distribution of weight is equal in all directions, and does depend on the gravitational field.

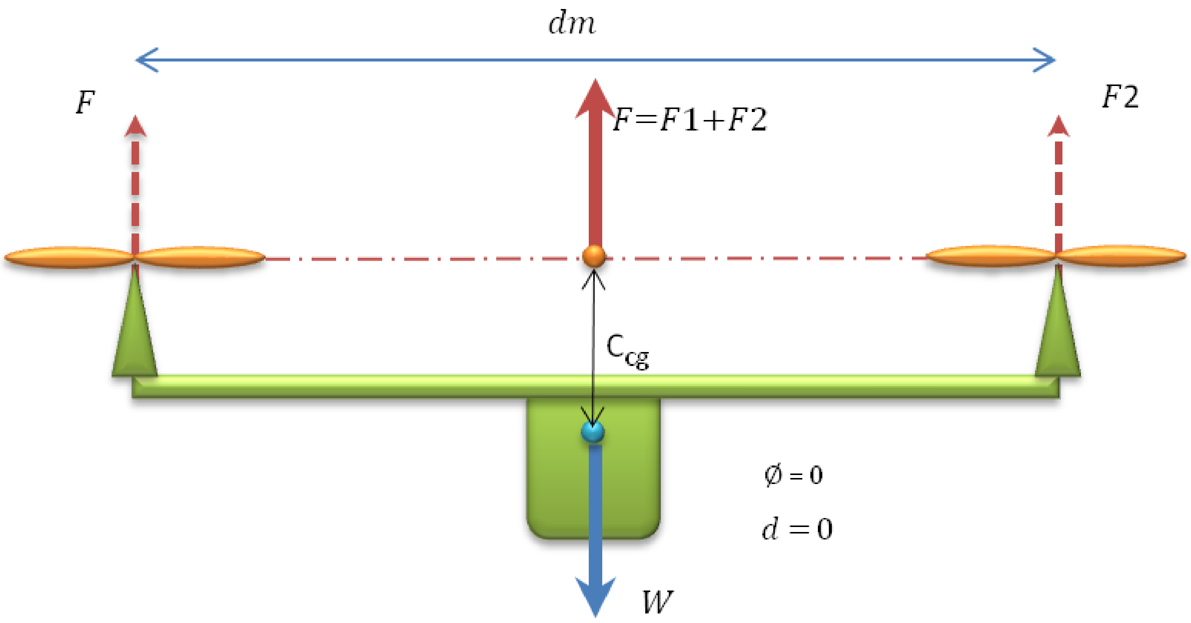


Figure 3: Representation of the position of the center of gravity C_G, the center of mass C_M and the center of thrust C_T.
Sources: https://drones.stackexchange.com/questions/1094/how-does-the-center-of-gravity-affect-a-quadcopter

Definition 4. Center of mass [10]

Centre of mass is the point at which the distribution of mass is equal in all directions, and does not depend on the gravitational field.

Definition 5. Center of thrust [11]

The center of thrust) is the midpoint where thrust from all a craft's reaction engine balances and the direction in which a craft's thrust is acting.

Note that the wind resistance also plays an important role in the drone's dynamics. That is why the more aerodynamic the frame shape is, the more the drag effect will be reduced.

Now that we have understood the importance of the symmetry in the frame shape, five popular frame models will be presented in the framework of a quadcopter [12][4]

They are illustrated in Figure 4 and a comparison is available in Table 2.

	body section	dynamics
H	long	less agile (pitch inertia)
True X	small	optimal: centralized mass
Hybrid X	long	between H and True X
Stretched X	medium	high speed/performance
Square	small	not optimal: increased weight and drag

Table 2: Frame shape comparison of two criteria: the body section (i.e working area) available on the frame and its dynamics.

---

[4] This can be generalized whatever the number of motors.

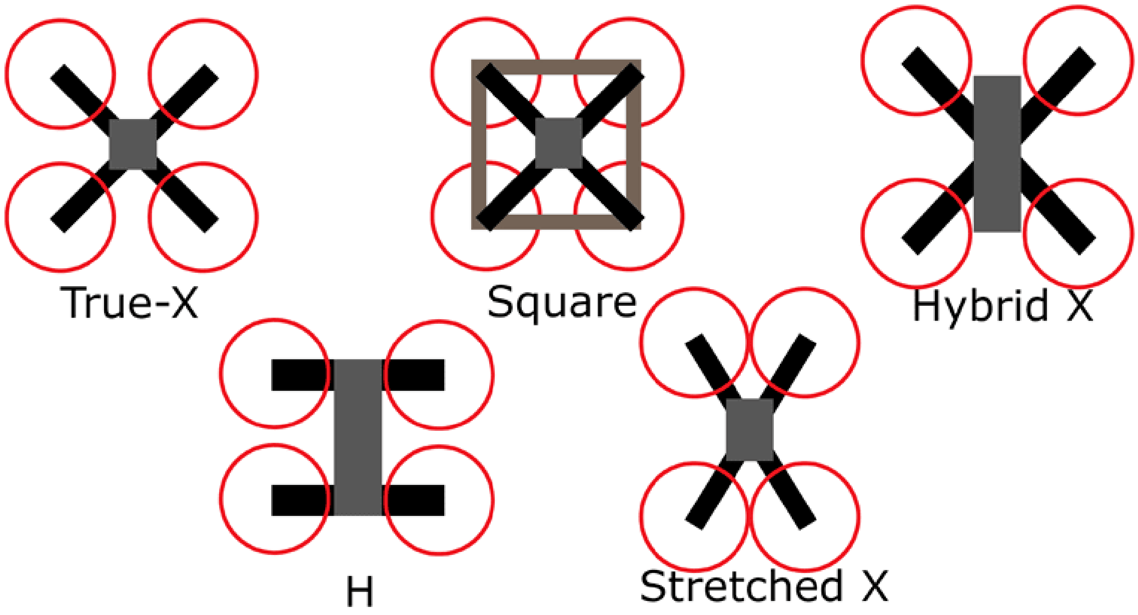


Figure 4: Common frame shapes, in the framework of a quadcopter.
Sources: https://dronenodes.com/drone-frame-racing-freestyle/

- H: Not common for experts, this shape is often used by beginners since its ensures a long body area (due to motor spacing) for material disposal. However, the performances in flight are not optimal since the long arms increase the inertia of the whole frame.
- True X: The four motors are equally distant from the center of gravity, ensuring very good flight performances. The body area can be made very small since the motors are separated by the natural geometry of "X".
- Hybrid X: Compromise between H and True X shapes. It allows a long body while ensuring that all motors are equally spaced from the center, leading to balanced flight performances.
- Stretched X: The goal of this shape is to reduce the interactions between front and rear propellers, leading to high speed and flight performances. However, it is most advanced and requires a good tuning.
- Square: It is a reinforced version of the True X model; it allows to increase the strength of the whole frame while increasing the total weight and the drag (since the surface area is bigger than a classical True X).

Of course, this is not an exhaustive list. There are a lot of other designs such as Dead Cat (improves camera field of view) or Z (less turbulent air directed toward rear motors) [13].

2.1.4 Size

The size of the frame represents the diagonal distance (units: mm) between two opposite motors. There is a link between the size of the frame and the size of the rotors. There are drones of all sizes, ranging from 1 cm (Piccolissimo [14]) to 1.8 m (Divine Eagle [15]). The size of the frame influences the flight behavior of the drone [16]: a small frame allows to reach higher speeds while reducing the overall weight. At this scale, the wind resistance is negligible for small frames but the drone will be carried by the wind. On the other hand, larger frames ensure much more stable flights. The additional weight ensures more inertia which allows a better control in windy area. Note that they need to compensate the wind resistance with bigger motors. There is a counterbalancing effect between size and weight variables. The global flight behavior is dictated by the size/weight ratio.

2.2 Propeller/Rotor

In this section, the topic will be rotors/propellers. Note that both names are correct but the term "propellers" is more used in aviation; so in the following text, the term rotor will be preferred. First, a description of the role between the flight movement of the drone and the rotor will be given. Then, the influence of the size, the number of blades or the type of material used to build a rotor will be explained.

2.2.1 Physical explanation

Each rotor is powered by a dedicated motor and the main goal of the rotor is to transform rotary motion into linear thrust, allowing the drone to be directed along any trajectory. Indeed, by spinning, the propeller creates an airflow which results in a pressure difference between the top and bottom surfaces of the rotor. Therefore, it accelerates an air mass upwards (lift) which opposes gravity. Note that there is a physical limit: the speed of the rotor, s_{rotor}, is limited to the speed of the sound s_{sound}. Therefore, we have:

$$s_{rotor} \leq s_{sound} = 330 \text{ m.s}^{-1} \tag{1}$$

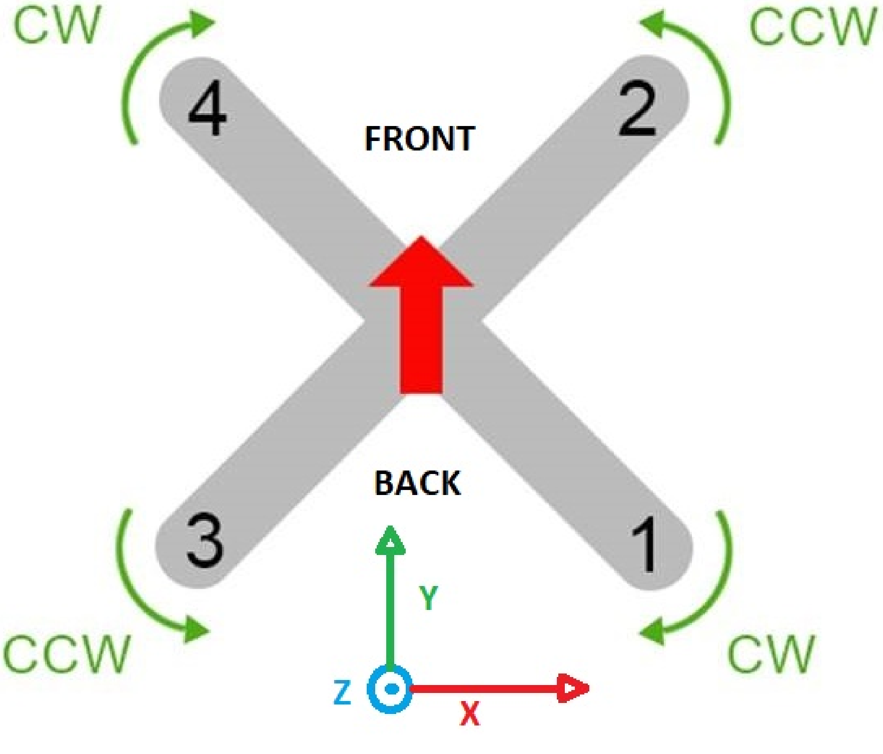


Figure 5: Illustration of clockwise CW and counter-clockwise CCW rotors on a classical quadcopter. Sources: https://dronenodes.com/drone-motors-brushless-guide/

Basically, The Figure 5 represents the typical rotor arrangement for a quadcopter. Theses rotors are arranged in pairs, spinning in clockwise (CW) or counter-clockwise (CCW). The CW and CCW movements of the rotors balance each other, which means that the net angular velocity is zero (ensures stability): the UAV is in hovering flight. To move the UAV in any direction, it is necessary to have a non-zero angular velocity, and therefore to modify the rotations of each of the rotors (i.e the motor supplied voltage, using ESC), according to the desired movement: it ascends, descends or affects its yaw, pitch and roll[5]. These are notions that come from the aeronautical field. They describe rotational movements around the three main axes of the system. The Figure 6 shows these three axes.

---

[5] In French, yaw means "lacet", pitch means "tangage" and roll means "roulis".

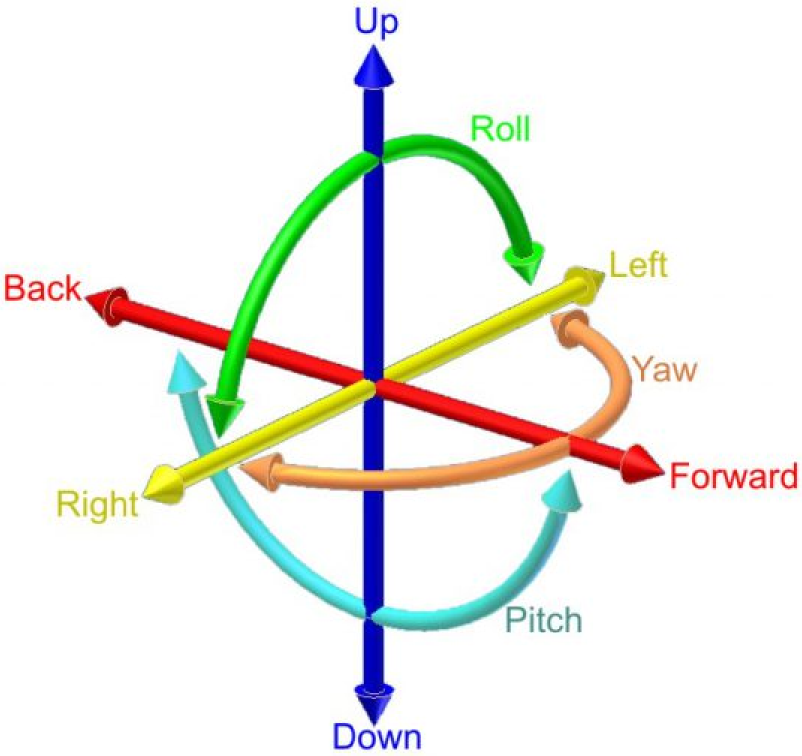


Figure 6: Description of the three main rotation axis.
Sources: https://emissarydrones.com/what-is-roll-pitch-and-yaw

Now, we will illustrate how it is possible for a drone to perform these three types of rotation/movement by modifying intelligently the rotational speed of its CW and CCW rotors (refers to Figure 5) [17] [18].

- The yaw axis, also called the vertical axis, allows the drone to move its nose on the left or right, as shown in Figure 6. In order to perform this maneuver, the UAV must increase the rotational speed of the CW (resp. CCW) rotors, corresponding to rotors 1 and 4 (resp. 2 and 3), to turn on its right (resp. left).
- The pitch axis, also called the lateral/transverse axis, describes an axis connecting the left side with the right side. It allows the drone to move its nose up and down, as shown in Figure 6. In order to perform this maneuver, the UAV must increase the rotational speed of the rear (resp. front) rotors, corresponding to the rotor 1 and 3 (resp. 2 and 4), to move forward (resp. backward) (refers to Figure 5).
- The roll axis, also called the longitudinal axis, describes an axis connecting the front to the back. It allows the drone to rotate itself on its left or right, as shown in Figure 6. In order to perform this maneuver, the UAV must decrease the rotational speed of the left (resp. right) rotors, corresponding to the rotors 3 and 4 (resp. 2 and 1), to move on its left (resp. right).

NB: When talking about the elements that allow to steer a drone, there is sometimes mention of the throttle. Strictly speaking it is not a steering movement but an element to control the altitude of the drone. For example, to increase (resp. decrease) the altitude, it is enough to increase (decrease) the speed of rotation of all the rotors.

2.2.2 Dimension

Two measurements are often given to characterize rotors:

- Total rotor length from end to end,
- Rotor pitch.

The first one informs the size of the rotor while the second describes the attack angle. The attack angle represents the distance that can be travelled in one complete rotation of the rotor. Therefore, it depends on the blades composing the rotor and their orientations.

The effect of the size of the rotor can be easily understood [19]: small rotors need less energy to reach a certain speed because they are less influenced by inertia than large rotors. They are therefore more reactive and easier to control. On the other hand, at a fixed RPM, large rotors provide more lift than small ones. In addition, they generally provide more stability.

The effect of the pitch can be understood as follows: the higher the pitch is, the more distance the rotor covers at a fixed RPM. So more lift is obtained but at the cost of a more energy-consuming motor.

In summary, a drone equipped with small rotors with maximum pitch will reach high speeds but will have a very limited flight autonomy, unlike a drone equipped with large rotors with low pitch which will require stability rather than speed.

2.2.3 Blade

The rotors can be equipped with a variable number of blades: there are some with two, three or even four blades. There is a relationship between the efficiency $\frac{T^{(g)}_{rotor}}{P_{motor}}$ (see section 2.3) of a drone and the surface area of the rotor:

$$\frac{T^{(g)}_{rotor}}{P_{motor}} = \sqrt{\frac{2 \cdot \rho_{air}}{F_{WL}}} \tag{2}$$

where:

- $T^{(g)}_{rotor}$ represents the thrust provided by the rotor,
- P_{motor} is the power consumed by the motor,
- ρ_{air} designs the air density,
- F_{WL} represents the wind loading (i.e the distribution of force on the rotor's surface).

As it can be seen from Equation (2), to increase the efficiency ratio $\frac{T}{P}$, it is enough to increase the number of blades: in this way, W will be reduced because the forces will be better distributed instead of being concentrated on a small number of blades. The more blade there is, the more lift is created at fixed RPM but the more drag is experienced (which increases according to the rotor surface) so, at the end of the day, it is difficult to predict whether adding a blade will do more good than harm. Indeed, the effect of the number of blades on the performance of the drone in flight is still not well known because it depends on many external factors such as drag losses [20], motor losses, Reynolds number, .. These interactions will not be studied in this work.

2.2.4 Material

As for the frames (see section 2.1.2), the classic materials found in the rotors are plastic and carbon fiber. The Table 3 summarises their pros and cons.

	Carbon fiber	Plastic
Price	Costly: supplier's dependency	Cheaper: can be self-made
Mechanical properties	Stiffer and lighter: better flight performances	Flexible: better absorption during impact

Table 3: Comparison between two materials mainly used to make rotors, namely plastic and carbon fiber.

2.3 Motor

Now that more is known about the rotors and the different maneuvers that a drone can perform, it is time to describe the component that really allows the drone to move (i.e. generating the rotation and speed of the rotor), namely the motor. This section begins with a reminder of the two existing technologies for motors (brushes vs. brushless) [21]. Then, the different factors to be taken into account when comparing two motors will be presented.

2.3.1 Brushed vs brushless

The first technology to be developed was the so-called brush motors. These motors had two wires (positive and negative terminal) and no electronics: they were therefore very easy to control (no need for an ESC) and could work in extreme conditions/environment.

Then appeared the technology of brushless motors. This technology improves the previous one on many points: since these motors don't use brushes, they don't require any maintenance or replacement in this respect. Moreover, these motors heat up less (better heat dissipation) and there is no more brush friction involved: more efficient motor and higher speed/range achievable. On the other hand, all these advantages do not come for free: these brushless motors are composed of three wires (positive, negative terminal and control signal) and require an ESC to operate (see section 2.4).

2.3.2 Selection criteria

There are three main factors to consider when comparing two brushless motors:

- Thrust/Weight ratio
- Motor efficiency (i.e thrust/power ratio)
- K_v value

2.3.2.1 Thrust/Weight ratio

The first notion to take into account is the thrust/weight ratio $\frac{T^{(N)}_{rotor}}{W^N_{uav}}$(see Definition 6).

Definition 6. Thrust to weight ratio [22]

Thrust-to-weight ratio $T^{(N)}_{rotor}/W^N_{uav}$ is a dimensionless ratio of thrust to weight of a rocket, jet engine, propeller engine, or a vehicle propelled by such an engine that is an indicator of the performance of the engine or vehicle. The thrust-to-weight ratio is calculated by dividing the thrust (in newtons) by the weight (in newtons) of the engine or vehicle.

As explained in the definition, this ratio reflects the vehicle's ability to propel itself through the air, overcoming the force of gravity on its own weight. Obviously, the higher this ratio is, the more easily the vehicle will be able to move (remain stable in windy conditions).

In practice (i.e. thumb's rule), the aim was a 2:1 ratio, which translates as follows: the vehicle must be able to hover when its throttle (altitude control) is at 50%. In order to select motors with enough thrust, the ratio is calculated by taking into account the whole drone for the weight calculation: weight of the frame, weight of the onboard electronics, weight of the engines, ... Once the weight of the whole aircraft is calculated, one must choose engines that are two times more powerful (i.e. able to produce twice the capacity of the weight in thrust). Once again, a trade-off has to be considered: the better the ratio, the better the drone's agility but at the expense of stability (indeed, if the 1% increase in throttle increases the total thrust generated by the motors, it is clear that it will be difficult to stabilize). To give some number [23], some achieved 13:1 ratio but, at this state, motor spin limitation occurs.

2.3.2.2 Motor efficiency

The second notion to take into account is the efficiency of the motor because it is linked to the global efficiency of the system. To be able to calculate the efficiency of the drone, it is necessary to consider that the motor and the rotor are not coupled before considering their combination [24].

First, the motor efficiency is modelled thanks to the dimensional ratio $\frac{P_{mech}}{P_{elec}}$, which is calculated thanks to the following equations:

$$P_{mech} = \tau \cdot \omega_{motor} \tag{3}$$

$$P_{elec} = V \cdot I \tag{4}$$

$$\eta_{motor} = \frac{P_{mech}}{P_{elec}} = \frac{\tau \cdot \omega_{motor}}{V \cdot I} \tag{5}$$

Then, the rotor efficiency is modelled thanks to the ratio η_{rotor}, so that:

$$\eta_{rotor} = \frac{T^{(g)}_{rotor}}{P_{motor}} = \frac{T^{(g)}_{rotor}}{P_{mech}} \tag{6}$$

where $P_{mech} = P_{elec}$ since the motor is coupled to the rotor.

Finally,the overall efficiency can be calculated using the following equation:

$$\eta_{syst} = \eta_{rotor} \cdot \eta_{motor} = \frac{T^{(g)}_{rotor}}{P_{mech}} \cdot \frac{P_{mech}}{P_{elec}} = \frac{T^{(g)}_{rotor}}{P_{elec}} = \frac{T^{(g)}_{rotor}}{V \cdot I} \tag{7}$$

The Equation (7) expresses the link between the thrust generated by the drone and its power consumption. This ratio varies when changing the motor, the rotor or the ESC (not considered here). It is therefore necessary to look at the specs of the engine that we wish to buy because it could reduce the overall efficiency of the drone.

2.3.2.3 $\mathbf{K_v}$ value

Last but not least, it is important to have a look at the K_v value (see Definition 7), also called velocity constant of the motor.

Definition 7. $\mathbf{K_v}$ value - Velocity constant [25]

“K_v” refers to the constant velocity of a motor (not to be confused with “kV,” the abbreviation for kilovolt). It is measured by the number of revolutions per minute (rpm) that a motor turns when 1V (one volt) is applied with no load attached to that motor. The K_v rating of a brushless motor is the ratio of the motor’s unloaded rpm to the peak voltage on the wires connected to the coils.

From that Definition 7, the K_v value indicates how fast the motor turns when a voltage is applied to it. To give a concrete example, consider a 1000 K_v motor, like the one shown in Figure 7. By connecting this motor with a 11.1V supply voltage, the motor would spin at: $11.1 \cdot 1000 = 111000$ RPM, with no load. By adding rotor on top of the motor, the measured speed (RPM) will be smaller because of air resistance applying to the rotor. Generally, small (resp. large) K_v motors are used with large (resp. small) rotors since the focus is put on stability (resp. speed) rather than speed (resp. stability).

There is a link between K_v velocity constant and K_t torque constant [26] [27]. To highlight it, we must first introduce the back EMF constant K_e, which is an inverse measure of K_v:

$$K_e = \frac{1}{K_v} \tag{8}$$

K_e expresses the link between the back-propagated EMF E and the rotational speed ω_{motor} of the motor, so that:

$$E = K_e \cdot \omega_{motor} \tag{9}$$

K_t expresses the link between the torque τ generated by the motor and the current drawn I:

$$\tau = K_t \cdot I \tag{10}$$

Figure 7: A brushless DC electric motor, whose velocity constant K_v is equal to 1000 RPM/V.
Sources: https://www.graylogix.in/product/bldc-a2212-13t-1000-kv-brushless-dc-motor

Applying conservation of electrical and mechanical power (neglecting losses) and using equations (9) and (10), we have the following relationship:

$$\underbrace{E \cdot I}_{\text{Conservation of electrical power}} = \underbrace{\tau \cdot \omega_{motor}}_{\text{Conservation of mechanical power}} \tag{11}$$

$$K_e \cdot \omega_{motor} \cdot I = K_t \cdot I \cdot \omega_{motor} \tag{12}$$

$$K_e = K_t \tag{13}$$

Finally, using the fact that K_v is an inverse measure of K_e (see equation (8) in equation (13), we end up with:

$$\frac{1}{K_v} = K_t \Longleftrightarrow K_v \cdot K_t = 1 \tag{14}$$

From the last equation, the following behavior can be derived: K_v and K_t are inversely proportional, which means that a motor with a large velocity constant K_v has a small torque constant K_t.

This relationship between the two constants can be used to explain why motors driving large rotors have a small K_v value: they prefer torque to speed in order to be strong enough to drive the rotor.

The relation can also be used to make a quick link between the motor losses and the size of the motor: For a fixed torque, if we consider a motor with a small K_v (thus large K_t), then the relation tells us that the current drawn by the motor will be lower, thus the motor losses ($R \cdot I_2$) will be lower too. However, the rotors will be larger (i.e. heavier) in order to guarantee a large K_t. Thus, there is a compromise to consider.

2.4 Electronic Speed Controller

Now that the motor/ rotor pair is well understood, it is time to describe the component which allows to control our BLDC motor, namely the ESC. Indeed, as explained in part 2.3.1, the BLDC motor is composed of three wires: a positive terminal, a negative terminal and the third wire which is used by the ESC to communicate the control instructions coming from the flight controller, often in the form of PWM signal. It also takes the DC power from the battery to deliver it to the motor (requiring AC power). Each ESC is connected to one single motor (A quadcopter would therefore require four ESCs).

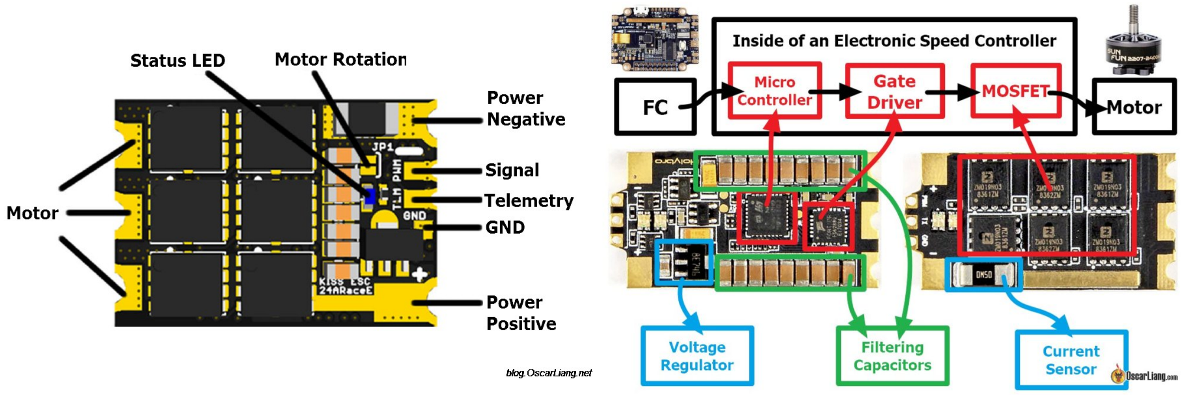


Figure 8: Illustration of the different elements composing a typical Electronic Speed Controller.
Sources: https://oscarliang.com/kiss-24a-esc-re/ https://oscarliang.com/choose-esc-racing-drones/

In this section, a description of the components inside the ESCs will be made. Then, the ESC functioning will be discussed. After that, we will give a word about the main differences between individual and "4-in-1" ESCs. Finally, a rating of the ESC will be given to conclude with the different firmware that can be found inside an ESC.

2.4.1 Composition

The typical insight of an ESC is illustrated in Figure 8.

By looking at the Figure 8, several elements can be observed, so that:

- Positive and negative pads to solder the battery,
- Three pads to connect the three-phase BLDC motor,
- Input PWM signal (control signal coming from FC),
- Status led indicating the current state of the ESC (initialization, error, ..),
- Filtering capacitors whose goal is to filter out the noise and ripples coming from the battery voltage,
- A switching network, composed of MOSFETs, used to drive the motor
- Gate Driver, used to drive the switching network.
- A microcontroller, sending information to change the speed of the switching network.

NB: Note that some ESCs have voltage regulator (typically 5V) directly embedded, which allows to power other components (though the battery) such as servo motors for instance.

2.4.2 Working principle

As a reminder, a BLDC motor is composed of two different parts, as illustrated in Figure 9:

- A moving part, named rotor, which is a permanent magnet,
- A static part, named stator, which consists of windings.

Current goes through the windings, allowing the generation of magnetic field lines. These magnetic field lines attract or repulse the permanent magnet. By activating the coils one after the other, the BLDC motor will continuously spin.
In order to increase the performances, opposite coils are often connected together, in such a way that they generate opposite magnetic field. In that way, the force suffered by the permanent magnet is double, since two windings are implicated instead of only one.

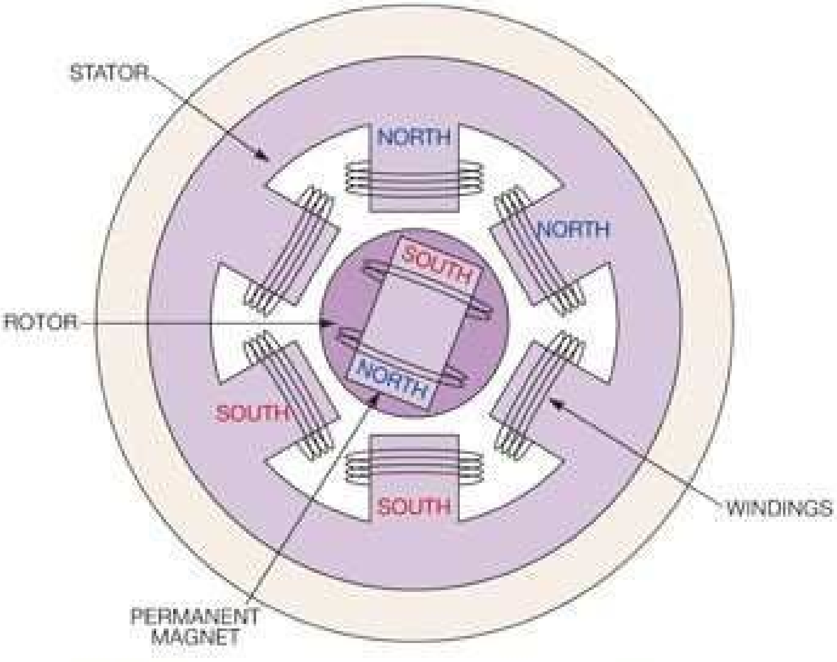


Figure 9: A brushless DC electric motor, composed of a permanent magnet and windings.
Sources: https://www.researchgate.net/figure/BLDC-hub-motor-have-rotor-with-a-permanent-magnet-containing-north-and-south-pole_fig2_349722826

Another trick to improve the efficiency is to activate two adjacent windings at the same time, with opposite magnetic field. Indeed, the first winding will attract (resp. repulse) the rotor, while the second one will repulse (resp. attract) it.
Therefore, it is necessary to activate the auspicious windings at correct timing, in order to control the speed of the motor. [28]
ESCs are used for this purpose, as illustrated in Figure 10

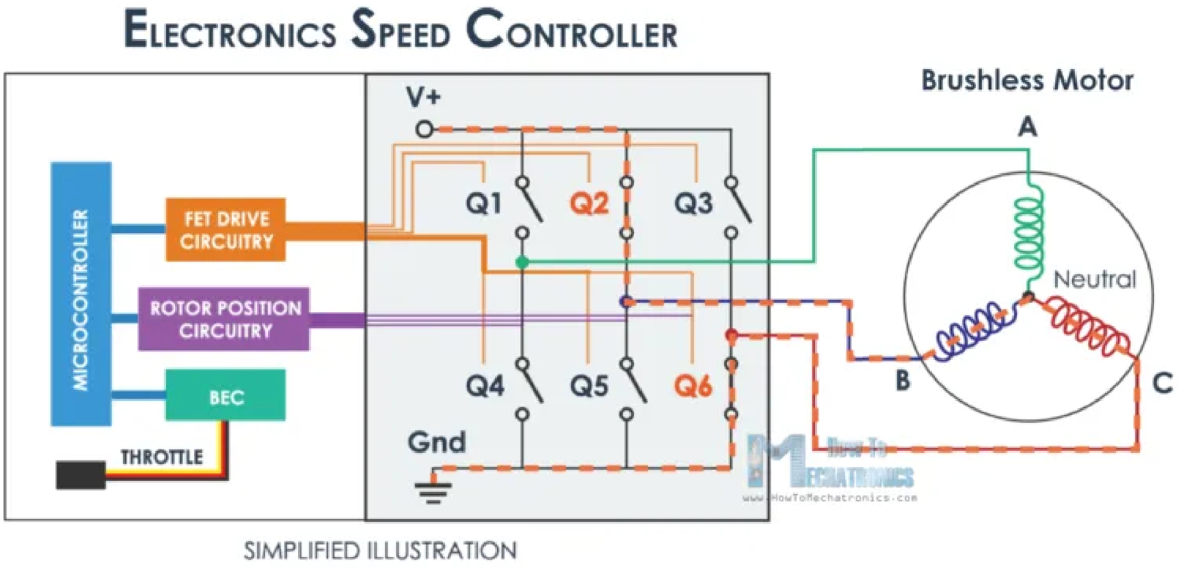


Figure 10: Switching network (MOSFET), used by the ESC to control the activation of each winding.
Sources: https://howtomechatronics.com/how-it-works/how-brushless-motor-and-esc-work/

In Figure 10, MOSFET 2 and 6 are switched ON, resulting in current passing through coil B and C. Since all coils are in star configuration, there is the presence of a neutral between coil B and C. Therefore, magnetic field lines generated within coil C are opposite sign compared to the ones generated within coil B and coil A is in "rest" mode, as explained above. Note that the speed of the motor is linked to the switching frequency of the MOSFET network. A phase is composed of one winding with positive current, one winding with negative current and one winding turned off. Thus, a complete cycle is composed of six different phases. To detect the rotor's position and therefore turn on the correct MOSFETs, the ESC can use Hall's sensor to track magnetic field or sensors to detect the back EMF. As a summary, the ESC tracks a speed reference and adjusts the speed of the motor by changing the speed of the switching network (MOSFETs) present in the motor. It follows very precise timing to transmit the current to the different windings [29].

2.5 Individual vs 4-in-1

As explained in the introduction about ESCs (see section 2.4, one ESC is required for each controlled motor. Therefore, some resellers decided to combine all the ESCs together into one circuit board. This circuit board is named "x-in-1", where x denotes the number of ESCs combined[6]. The Table 4 displays the pro and cons of "4-in-1" ESCs, while the Figure 11 illustrates individual and 4-in-1 ESCs.

4-in-1 ESC	
Advantages	Disadvantages
Lighter than four individual ESCs	Loss of four ESCs in one crash
Compact design	Not reliable if one ESC breaks
Embedded voltage regulator	
Cheaper	

Table 4: Pro and cons of 4-in-1 ESCs.

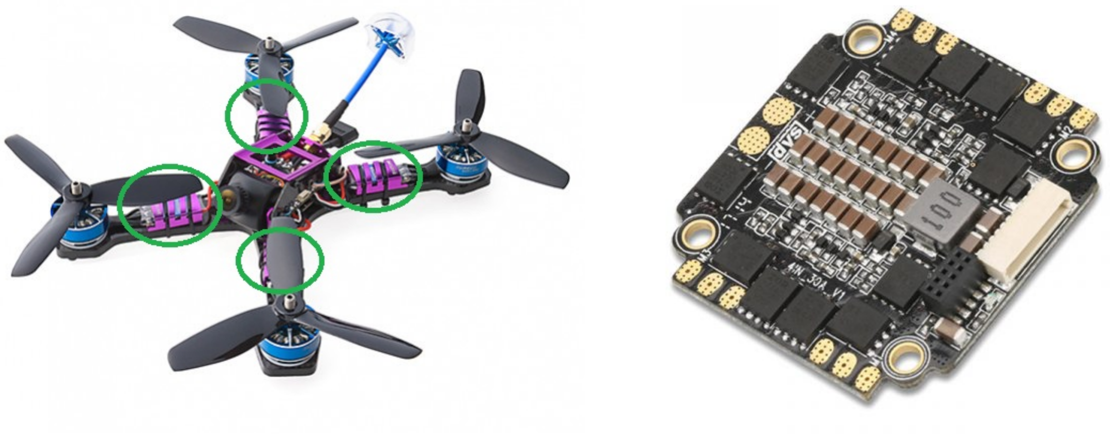


Figure 11: On the left, four individual ESCs (circled in green) while on the right, 4-in-1 ESCs soldered on a circuit board.
Sources: https://hobbyking.com/nl_nl/diatone-2017-gt200s-fpv-racing-drone-pnf-violet.html?___store=nl_nl
https://www.myfpvstore.com/fpv-electronics/esc/dys-f30a-4-in-1-30a-esc-blhelis_s-dshot-d300-600/

---

[6] For instance, in the case of a quadcopter, the combined ESCs will be named "4-in-1".

4-in-1 ESC offers a great and compact design compared to individual since there are less wires as the ESCs are soldered on the circuit board. There is also an advantage in terms of weight since less wires reduce the overall weight of the drone. The circuit board of the 4-in-1 ESC can also embed additional features such as voltage regulator for external components. Buying a 4-in-1 ESC is often cheaper than buying the corresponding quantity of individual ESC. However, if only one ESC in the circuit is affected, then all the ESCs are affected and the whole module is no longer usable and must be replaced, unlike the individual ESCs which can be replaced without affecting the other ESCs.

2.5.1 Rating

ESC manipulates current from battery pack to power motors. Depending on the motor, the amount of current required to be drawn from the battery pack can differ, so some rating must be indicated on the packaging:

- Amperage: This value represents the sustained current that the ESC can handle. Moreover, if the current passing though it is bigger[7] than the value indicated, it must be for a short amount of time in order to prevent accident.
- Power: It is linked to the type of battery (see section 2.7) that can be paired with the ESC. Higher voltage battery provide the same motor output power at lower current withdrawal.

Therefore, selecting a appropriate ESC requires knowledge of the following two factors:

- Motor size: the bigger the motor, the higher the required current. This current must be smaller than the sustained current that the ESC can handle.
- Battery cells: the battery voltage must be compatible with the ESC.

2.5.2 Firmware

As illustrated in Figure 8, there is a microcontroller inside the ESC. Its job is to correctly interpret the signals sent to it by the FC [8] and to activate the switches at the right time in order to correctly control the BLDC motor. This control is done through an inbuilt program known as firmware. Concretely, the firmware indicates which protocols are supported by the ESC as well as what type of configure interface can be used with it. The protocol, on the other hand, allows the ESC and the CF to agree on the rules to follow when they want to communicate and understand each other,e.g. for instance, the maximal frequency indicating that the motor must be spinning at full power.

There are several firmware available on the market [9]. They all have their specificity but they can be classified in two groups, according to the types of signals they use in their protocol: analog signals (Analog or standart PWM, Oneshot, Multishot) and digital signals (D-shot, Proshot). As in other fields, digital signals are beginning to take the lead over analog signals because digital signals are much more robust to noise than analog signals. Moreover, the calibration step is no longer necessary.

2.6 Power distribution board

Now that the drone's components allowing its motion have been described, it is time to look at the component that allows the current to flow within the circuit between all components (ESCs and FC). Historically, the main goal of PDB was to deliver power from the battery to the ESCs. Since technology improves over time, PDB can also deliver power to other peripherals such as servo motors, FPV camera, FC, .. In order to spare some space, some resellers integrated the PDB within the FC. Even if the role of the PDB does not seem that important (one could imagine a design where the ESCs would be directly connected to the battery) it is not! Indeed, the PDB allows to reduce the stress (i.e. the workload) of the FC which concentrates only on the piloting and not on the distribution of the current within the drone. Moreover, the PDBs are equipped to better filter the electrical noise as well as to handle bigger currents and heat.

---

[7] In that case, the term current burst is used.

[8] The full chain is the following: signal from the remote control (e.g. throttle) is received by the FC and then sent as a PWM signal to the ESC.

[9] To cite some, there are BLHeli,BLHeli_S, BLHeli_32, SimonK, Kiss.

In this section, the power distribution board (PDB) will be described. First of all, a list of all the elements that can be soldered on the PDB will be established. Then, the ratings that are present on the PDB will be discussed. Finally, a great feature of the PDB, the voltage regulator, will be mentioned as well as other additional features that can be found on the PDB.

2.6.1 Nomenclature

A typical PDB and its soldering pads is represented in Figure 12.

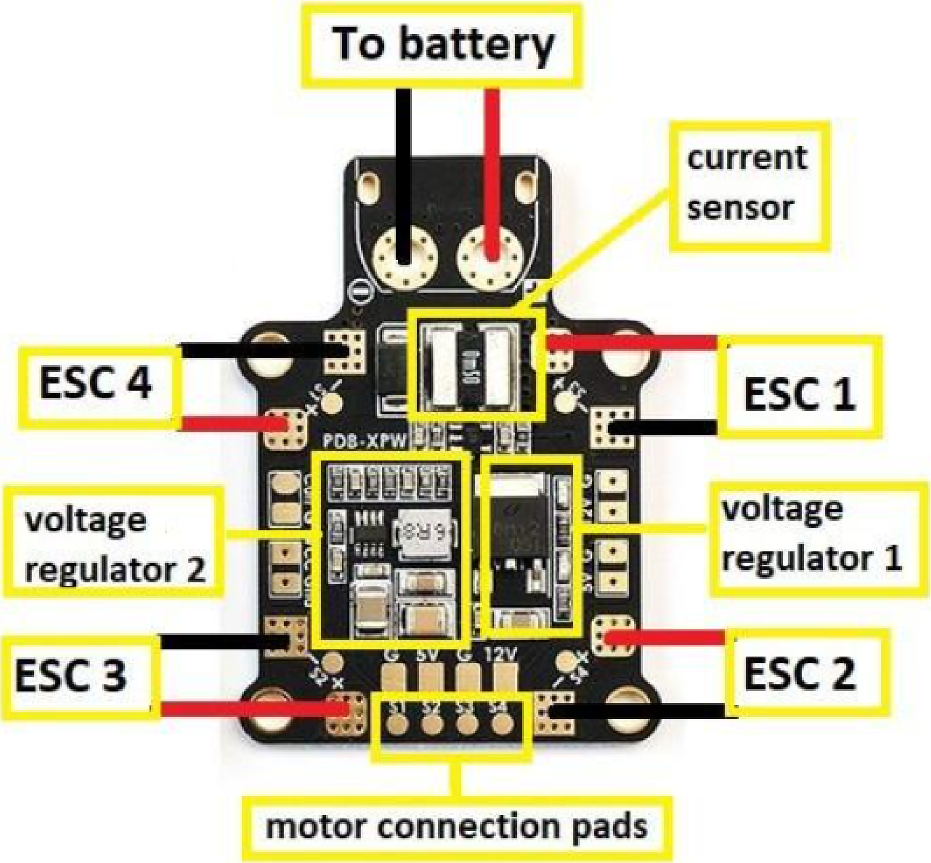


Figure 12: This picture shows a typical PDB for a quadcopter. Several soldering pads are present in order to be able to connect battery and ESCs. It also embeds additional elements such as voltage regulator and current sensor.
Sources: https://dronenodes.com/pdb-power-distribution-board/

In Figure 12, several elements can be retrieved:

- Two soldering pads on the top, allowing to connect the battery to the PDB.
- Four sets of two soldering pads on the extreme left and right, allowing to connect the 4 ESCs. It is through this connection that the ESCs (and therefore the motors) receive power from the battery.
- Four soldering pads on the bottom, one for each ESC. They are used as a communication link between FC and ESC (see section 2.5.2).

- A current sensor, near the battery pads. It allows to control at any time the battery charge level and to send the information back to the FC, which will be able to take decisions if the battery level approaches a critical threshold[10].
- Two voltage regulators, at the center of the circuit board. They can be used to power external peripherals (which requires stable 5V for instance) such as FC, FPV camera or even servo motors.

2.6.2 Voltage regulator

The main goal of the voltage regulator is to maintain a fixed output voltage, regardless of the input voltage. Two families exist: linear voltage regulator and switching voltage regulator.

We often retrieve linear voltage regulator on FC while PDB is composed of switching voltage regulator. The voltage regulators which are arranged on the PDB (i.e switching) are more efficient than the voltage regulators embedded on the FC (i.e linear) in general:

- Switching voltage regulators: those regulators contain switches that are turned on and off to charge an inductor. They do not emit excessive heat but the design is more complicated (requires external elements as capacitors, resistors) than linear voltage regulator. Those external elements must be well tuned to make it work efficiently.
- Linear voltage regulators: those regulators work in the following way: they take the difference between input and output voltage and burn the excess in heat. Therefore, they are less efficient but cheaper and easier to design.

2.6.3 Rating

One PDB differs from another by many factors such as size, additional embedded components, number of ESCs that can be connected, shape, .. Therefore, it is important to check that the PDB can handle/provide the requested current from the motors[11].

2.6.4 Additional features

As said in the introduction (see section 2.6), technology is evolving every day, so manufacturers are taking advantage of this to improve PDBs by adding features. For instance, there are PDBs that contain video transmitter onboard (for FPV camera) or even a combination of PDB and FC on the same circuit board. In that case, it allows the FC to estimate the consumed power (and therefore the battery capacity) by receiving information about battery voltage and current.

The general idea behind those improvements is to save space on the drone by combining the tasks of different components inside one circuit board. These two designs are illustrated in Figure 13.

2.7 Battery

Now that more is known about the PBD that distributes electrical power to the various equipment, it is time to look at the battery that powers this PDB. The characteristics of the battery influence the flight time of the drone (current delivered) and limit the speed of the motors (cell voltage). The weight of the battery accounts for a large part of the final weight of the drone. It is easy to understand that the heavier the drone is, the more difficult it is to get it to take off and the shorter the flight time is. Thus, doubling the number of batteries does not mean that the flight time will be doubled as well. Therefore, there is a compromise between the weight of the battery and the flight time. Also note that the batteries remain a dangerous component if badly maintained or if they are used in bad conditions.

In this section, the two main battery technologies, LiPo and LiHV, will be compared. Then, the cells composing the battery pack will be discussed. After that, the battery specifications will be described and a point of honor will be put on the "C-rating". A word about the connector will be given before concluding with the charging station.

---

[10] Typically, it will trigger RTL or land immediately.

[11] Remember that the larger the motor, the bigger the drawn current.

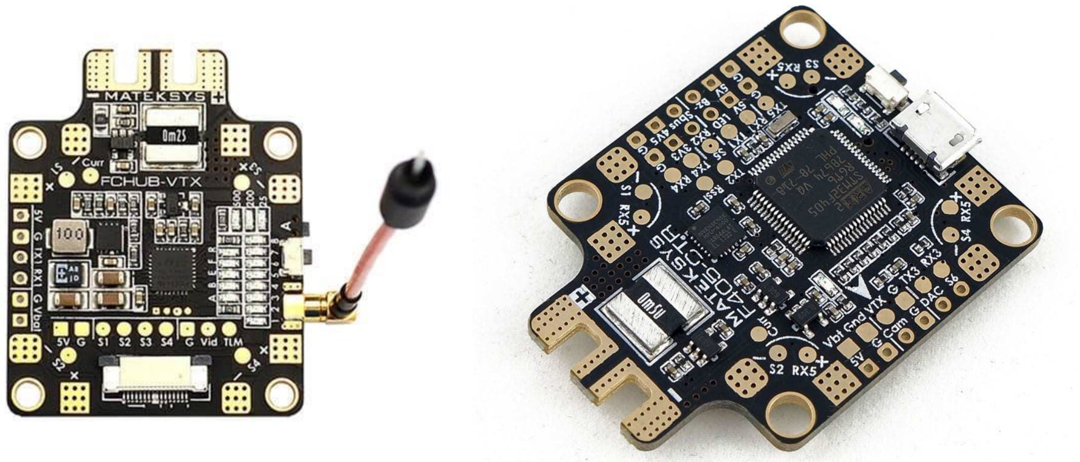


Figure 13: On the left, PDB and video transmitter combo; on the right, combination of PDB with FC. Sources: https://dronenodes.com/pdb-power-distribution-board/

2.7.1 LiPo vs LiHV

The motors being very greedy, powerful batteries capable of delivering high currents are needed. This is the reason why lithium batteries are used to power drones. There are two main categories of lithium batteries, based on different battery chemistry, namely:

- Lithium polymer (LiPo),
- Lithium polymer high voltage, (LiHV).

The main differences between those two categories are shown in Table 5.

	LiPo	LiHV
Fully charged voltage	4.2V	4.35V
Nominal voltage	3.7V	3.8V
Discharge behavior	linear	abrupt drops
Weight	bigger	smaller
Lifespan	300+ cycles	30-40 cycles
Drone application	common quadcopters	micro drones

Table 5: Main differences between lithium polymer and lithium polymer high voltage batteries.

As it can be seen from Table 5, a LiHV battery has a higher voltage than a LiPo battery. This means that at a fixed current, a LiHV battery will deliver more power than a LiPo battery. In addition, thanks to the chemistry used to make LiHV batteries, they are lighter when compared to their counterpart with the same charge.

Of course, there is a price to pay for those advantages: the lifespan of a LiHV battery is much shorter than that of a LiPo battery (six times fewer cycles here [30] [31]). Moreover, there is a non-linear discharge behavior, which does not make easy the estimation of the remaining flight time on the basis of the battery charge.

In terms of applications, LiHV batteries are used for dynamic drones, which need to achieve high performance in terms of speed. LiPo batteries are more classic and used by beginners.

2.7.2 Cell

A battery is a stack of identical cells, mounted in series (like a chain) on top of each other. The voltage of a battery is therefore determined/represented by the number of cells it contains. This is the reason why a battery is often denoted by αS [12], where:

- α is a number representing the number of LiPo cells in the battery pack.
- S indicates that all those cells are soldered in series.

The voltage provided by a battery pack is an important factor since, at fixed current, the power delivered by that battery grows linearly with the voltage. Note that the weight of the battery pack also increases with the number of cells. This means that there is a compromise between the power delivered by and the weight of the battery pack. The Table 6 indicates voltage and typical applications for 1S to 6S batteries[33].

Number of LiPo cells	Nominal battery voltage	Common quadcopter applications
1S	3.7V	Indoor micro brushed
2S	7.4V	30-70mm micro brushless
3S	11.1V	100-220mm brushless
4S	14.8V	220mm brushless race/freestyle
5S	18.5V	220mm+ brushless race/freestyle
6S	22.2V	220mm+ brushless

Table 6: This table shows the conversion between the number of cells (S) and the corresponding battery voltage. The typical use of each type of battery is also indicated.
Sources: https://www.getfpv.com/learn/new-to-fpv/all-about-multirotor-fpv-drone-battery/

2.7.3 Specification

On a typical battery pack, three specifications are always displayed:

- Voltage/Number of cells (see section 2.7.2)
- Capacity
- C-Rating

2.7.3.1 Capacity

The capacity of a battery, measured in mAh, expresses the amount of current that the battery is able to deliver within an hour. For instance, a 1000 mAh battery would provide 1A for one hour or 2A during 30 min or even 4A during 15 min. The higher this value is, the higher the power delivered by the battery is on a charge cycle, and therefore the flight time increases.

However, a higher capacity very often means that the battery is heavier, and therefore that the flight time is reduced. There is therefore an optimization/compromise to be made between the flight time to be achieved and the weight of the battery (which impacts the final weight of the drone).

2.7.3.2 C-Rating

The C-Rating (notation C) of a battery pack informs about the maximum current that this battery is able to deliver continuously. This is a maximum limit that should never be exceeded, otherwise the cells may be damaged or the battery may catch fire!

---

[12] It is also possible to find another notation βP, where P indicates that the β cells are put in parallel to reach a certain capacity[32].

The maximum current delivered by the battery is calculated according to the following formula [33]:

$$I_{safe} = C_{bat} \cdot CR_{bat} \tag{15}$$

where:

- I_{safe} is the maximum safe current draw [mA],
- C_{bat} represents the battery capacity [mAh],
- CR_{bat} denotes the C-Rating [h^{-1}].

For instance, a 1000 mAh 100C battery could deliver a maximum current equal to: $I_{safe} = 1000 \cdot 10^{-3} \cdot 100 = 100$A. The C-rating is therefore a measure of the time required for the battery to discharge so that [34]:

$$t_{discharge} = \frac{1}{CR_{bat}} \tag{16}$$

where $t_{discharge}$ represents the discharge time, expressed in hour. For instance, a 1000 mAh 100C battery ($I_{safe} = 100$A) is able to deliver full power (i.e 100 A) during $t_{discharge} = \frac{1}{100} = 0.01$ hour. Indeed, $0.01 \cdot 100 = 1\text{A} = 1000\text{mAh}$ and the initial battery capacity is retrieved.

The C-Rating of the battery can also be used to check the motors/ESCs compatibility with the battery pack. Indeed, dividing I_{safe} by the number of motors gives the maximum current draw by the motors. This maximal value must be supported by the ESCs (see section 2.5.1).

2.7.4 Connector

There is a whole range of battery connectors, varying mostly in size (they must be able to handle large currents). As explained in the battery specifications part (see section 2.7.3), the greater the battery capacity is; the greater is the amount of current that will flow. Hence, the greater the battery size is, the bigger the connector size is. The Figure 14 illustrates some connectors.

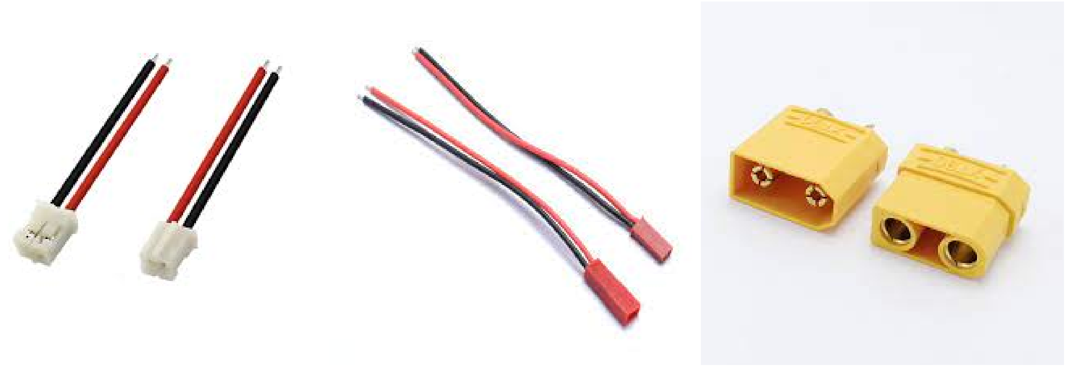


Figure 14: From left to right: JST-PH, JST and XT90 connectors used with 1S, 2S-3S and 3S-6S battery respectively
Sources: https://www.getfpv.com/learn/new-to-fpv/all-about-multirotor-fpv-drone-battery/

2.7.5 Charge station

Charging a LiPo battery is no small task. Indeed, it is necessary to ensure that the voltage of each cell of the battery remains around the nominal voltage of 3.7V at all times. This is why smart chargers, as shown in Figure15, are used.

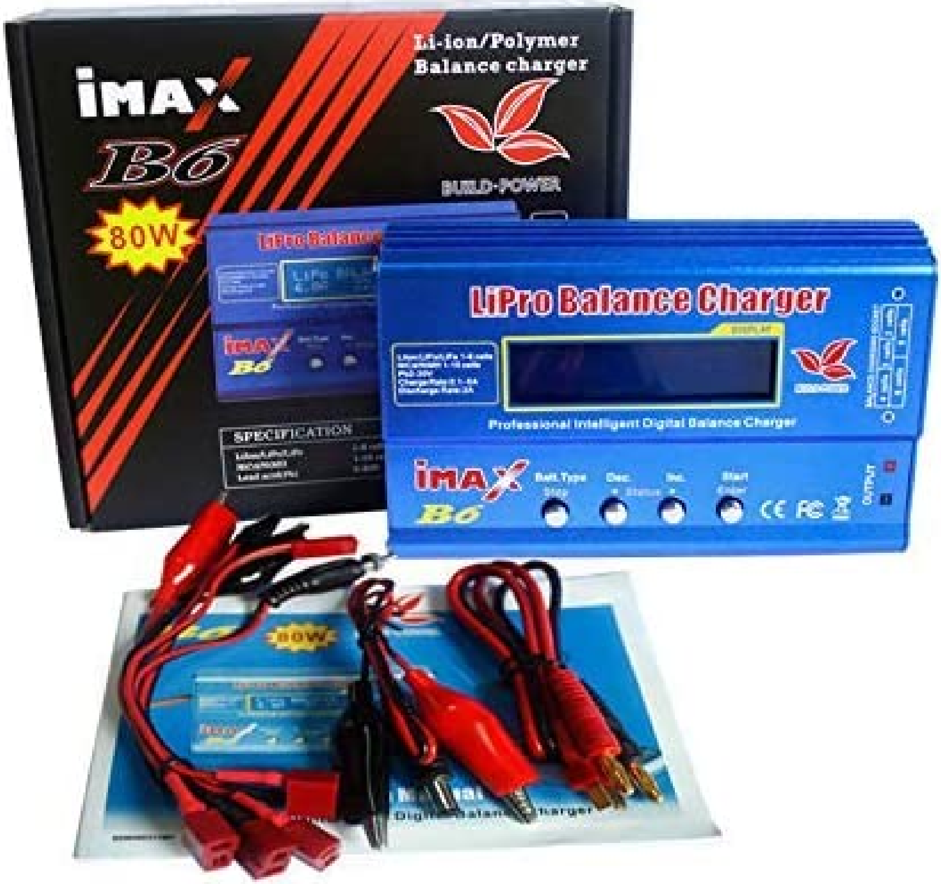


Figure 15: Illustration of an intelligent charging station, used to recharge LiPo batteries.
Sources: https://tinyurl.com/2p96hxkc

These chargers have two inputs:

- The first input is used to connect the positive and the negative terminal of the battery, like a "classic" charger.
- The second input is used to connect a special connector that takes the voltages between all the individual cells. In this way, the charger can control the voltage of each cell at any time. By looking at the number of wires of this connector, one can easily determine the number of cells in the battery [13]. Those connectors are illustrated in Figure 16.

2.8 Flight controller

Now that the description of the drone's "legs" is finished (i.e. all the components allowing the drone to move), it is time to move on to the description of its brain (i.e. the components responsible for making decisions, acting on the actuators), according to the values returned by the sensors. The brain in question is the flight control (FC), while the actuators correspond to the speed of each motor. This FC is equipped with various sensors in order to be able to send the correct signals to the ESCs,which will drive the motors as expected from the FC. Flights controllers have evolved with time and technology. Nowadays, they can be connected to various devices, such as GPS, FPV antenna, LED systems, telemetry systems, ..

This section will begin with the description of the sensors within the FC. Then, the firmware, processor and UART will be discussed to conclude on the combination of FC with other components.

---

[13] Number of cells = number of wires - 1.

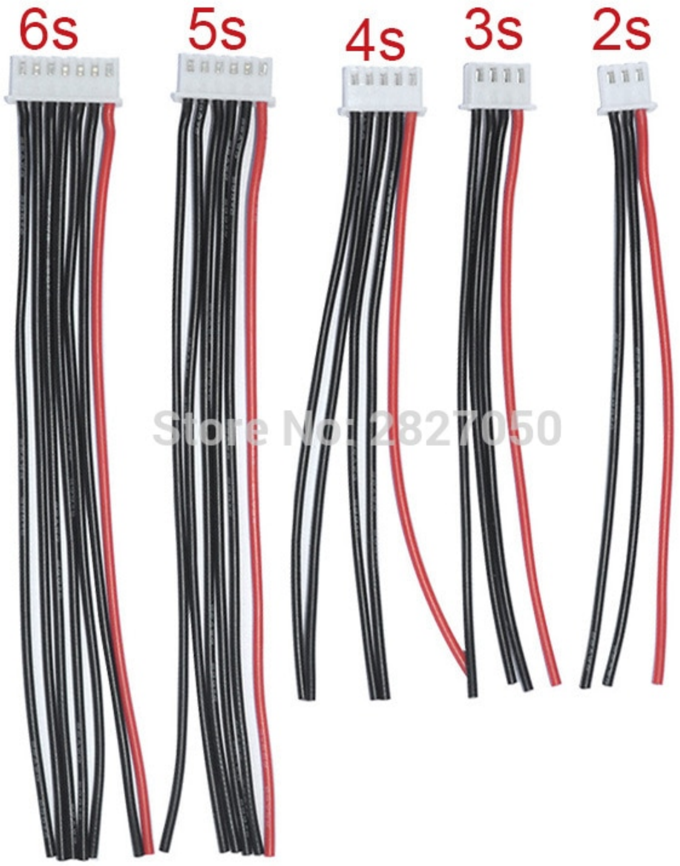


Figure 16: Illustration of the special connectors used with a LiPo charger to control cell voltage.
Sources: https://tinyurl.com/4au2rjsv

2.8.1 Firmware

All FC contains firmware (i.e a program) that is responsible for taking actions, based on sensors inputs or received radio signals:

- received radio signals: the firmware will transform commands sent from a remote controller into instructions for the ESCs, to control the speed of the motors in order to execute the manoeuvre dictated by the remote controller.
- sensor inputs: the firmware will analyse the values provided by various sensors in order to perform automatic manoeuvres, such as autonomous take-off, flight or landing.

Bunch of firmware for FC exist on the market, each having its own user interface and specificity. It is really application-dependant since some firmware have been designed for racing/freestyle flights while others are able to perform autonomous flights. Note that it is necessary to set some parameters before attending flying with the drone. Fortunately, it is not really difficult since they can be configured through a computer or even a smartphone. Here is a non-exhaustive list of firmware that are currently available on the market [35]:

- Betaflight: racing/freestyle flights for mini quadcopters, supporting a wide range of FC.
- KISS: one of the quicker setup/configuration processes.
- iNav: designed to perform tasks such as flying autonomously past landmarks (entered by the user) on a map or enabling RTL (return to home).

- Ardupilot: the most popular open-source firmware for GPS autopilot.
- Multiwii: firmware created using the IMU sensor from the Nintendo Wii, combined with an Arduino board.

2.8.2 Processor

The firmware codes are stored on processors, which are named microcontroller units (MCU). The MCUs are also responsible for handling complex mathematical computations. Nowadays, among the different MCU, five types are widely spread in FCs: F1, F3, F4, F7 and H7. They only differ in calculation speed and memory size, as shown in Table 7 [36].

	F1	F3	F4	F7	H7
Speed	72 MHz	72 MHz	168 MHz	216 MHz	480 MHz
Memory	128 KB	256 KB	1 MB	1 MB	128 KB

Table 7: Comparison in terms of calculation speed and memory between the five main types of MCU used in FCs.

Nowadays, F1 and F3 are no longer supported by most firmware since they use a lot of memory space. Note that some processors are able to handle inverted signals, while others cannot.

2.8.3 UART

To exchange information with external components (e.g. telemetry), the FC relies on the universal asynchronous receiver/transmitter (UART). Therefore, the external devices, such as serial radio receivers or Race transponders are connected to the FC by this hardware serial interface. In fact, the external peripherals are connected on the TX pin (transmit data) of the UART while the RX pin (receive data) of the UART is connected on the FC. The number of available UART depends on the design but this is also often linked to the number of processors, as shown in Table 8 [36].

Processor type	F1	F3	F4	F7
Number of UART	2	2-5	2-5	3-7

Table 8: Relationship between processor type and number of available UART in FC.

2.8.4 IMU

The control signals sent by the FC greatly depend on values returned by sensors. In order to be able to successfully control the movement of the drone, the FC needs information about its current environment. This is the purpose of the inertial measurement unit (IMU). The IMU is responsible for measuring force, angular velocity and attitude, thanks to the following sensors:

- Accelerometer: it measures the drone's acceleration (horizontally and vertically) but also provides the acceleration forces suffered by the drone in 3D space (useful to determine the tilt angle).
- Barometer: it measures the atmospheric pressure. This can be useful to approximate drone's altitude [14].
- Gyroscope: it is used to get a measurement of the rate of rotation of the drone. It is essential to provide stability in flight. To reduce errors from motor vibrations, it is often soft mounted on foam.
- Thermometer: it checks the drone's temperature and triggers fail-safe (see Definition 8) if the IMU is too warm for instance.

---

[14] As altitude increases, atmospheric pressure decreases. The atmospheric pressure can be calculated at a given altitude [37].

Definition 8. Fail-safe [38]

Designed to return to a safe condition in the event of a failure or malfunction.

The main differences between IMUs are in the accuracy of the measuring instruments, e.g. the maximum sampling frequency, the resistance to electrical noise or the communication protocol used. Note that all the sensors must be properly calibrated, in order to give consistent data and avoid crash.

2.8.5 Design

As mentioned in section 2.6.4, other components can be integrated with FC such as PDB or even 4-in-1 ESCs. The main goal of these designs is to save space and ensure compatibility between components.

FC can also embed other additional features such as black box (record flight logs), battery eliminator circuit (BEC) which provide regulated 5V, boot button (manually flash firmware), ..

2.9 Control receiver and remote control

Now that we have described all the components allowing the drone to plan (see section 2.8) and to physically carry out (see sections 2.2, 2.3, 2.4) its trajectory in flight as well as components allowing power to flow in the drone's circuit (see 2.7, 2.6), it is time to describe the radio control system, which is composed of:

- A control receiver (drone side)
- A remote control (user side)

The operation of the radio control system is as follows: the remote control recovers the inputs sent (stick and switch) and sends them into the air. The control receiver located at the drone level has to recover its information in real time and to make them circulate until the brain of the drone, namely the FC, which, thereafter, will decide of the control signals to be sent to the ESCs in order to move the drone in compliance with the data entered at the remote control by the user.

In this section, the binding between receiver and transmitter will be discussed. Then, the scope of the communication will be illustrated for different technologies. After that, some communication protocol will be described. Finally, the number of required transmitter channels will be explained.

2.9.1 Binding

As introduced in this section, the radio control system must establish a point-to-point contact between the remote and the corresponding control receiver in order for the information entered on the remote control to reach its destination. The communication does not have to be picked up by all the drones in the vicinity. It is the role of the binding to establish this link control receiver-remote. The binding operation can differ from one remote to another but always behaves more or less the same way: once the control receiver is powered, it waits for a remote in the neighborhood to send it a binding instruction. Usually, the binding instruction is sent by pressing a button on the remote. Once the binding instruction is accepted, they are linked. During the rest of the communication, they will be synchronized in terms of the frequency to be used (this will vary throughout the communication but will always be in the authorized band: 2.4 GHz).

2.9.2 Range

The communication range is defined as the maximum distance during which the control receiver still clearly perceives the signals sent by the remote. The maximal range will vary accordingly to those factors [39] [40]:

- The output power of the transmitter: the greater the power emitted by the remote control is, the further the signal will travel. There is a limit for the power emission, depending on the type of transmission.
- The sensitivity of the receiver: the higher the sensitivity is, the more the receiver will be able to separate the useful signal, whose power decreases with distance, from the noise.
- The quality of antennas at both ends: the larger the antenna the better the signal quality sent/received. Furthermore, its placement is important to get a good communication.
- transmission standards:
  - Bluetooth: 100m. Could have Wi-Fi or microwave oven interferences.
  - Wi-Fi: 200-300m[15]. Depends on geographical conditions (mountains vs open fields) and on the occupation of the frequency band.
  - DSSS or FHSS: ≈ 2km. standards for expensive radio remote controller. The same remote controller can be used for different drones.

2.9.3 Communication protocols

Once the binding is done (see section 2.9.1), it means that the receiver understands the language used by the remote control. Once the link is established, the receiver should tell the FC that it is now able to communicate with the remote and also informs it of the protocol used by that specific remote. Here is a non-exhaustive list of the standard protocols used by the remote controls:

- PWM: Slow analog signal using one separate wire for each channel. Nowadays, it is not used anymore.
- PPM: Improved version of PWM: no longer need to have one specific wire dedicated to only one channel. All channels are sent through the same wire, as a series of timed pulses.
- Digital protocols(SBUS, IBUS, DSM2/X): do not rely on timed pulses signals anymore since the signal is digitized. This gives perfect accuracy and the communication is quicker.

2.9.4 Transmitter channels

It is important to understand that each command is attributed to one and only one transmitter channel. Indeed, it allows to send simultaneous commands to achieve a more complex movement [16]. From section 2.2.1, yaw, pitch, roll and throttle have already been mentioned. In fact, these are the four essential controls for any drone. Since a different channel is assigned to each command, a minimum of four channels is needed to properly control the drone.

In order to enhance the piloting experience, it is always nice command "Flight mode" to switch from one piloting style to another. In addition, it is recommended to assign a "Kill switch" command to block the flow of power within the drone and an "arming" command to arm/disarm the motors. These improvements lead to six channels.

In conclusion, before buying a control receiver and a remote, be sure that:

- Both are compatible (see section 2.9.1)
- They can handle as many channels as needed for the current application (and its future evolution).

2.9.5 Programming

Before flying a drone with a remote, it is necessary to set up the remote control correctly by programming it. "Programming" should be understood as:

- Binding the remote with the adequate receiver (see section 2.9.1),

---

[15] Some remotes can reach 2km.

[16] For instance, sending throttle up and yaw command make the drone rotating on itself while going up in the sky.

- Attributing a radio channel to each stick and switches, as well as their minima and maximal values.
- Linking the radio channels attributed to a command, such as throttle, roll, kill switch, ..

Once all those steps are done, the remote control is set up and ready to fly a drone.

2.10 Additional information

remettre toutes les parties

Part III

Comparison between theoretical and practical

This part is focused on measurements and model validation. More concretely, the methodology as well as the prototype allowing to control a motor will be explained as a first step. In a second step, the experimental setup and the measurements obtained with it will be presented. In a third step, the equations behind the theoretical model will be written. The data provided by the manufacturers will be transcribed as a forth step. The final step will be a comparison between experimental and theoretical measurements.

1 Methodology

One of the objectives of this final study was to answer the following questions:

- How does the FC communicate with the ESC ? More precisely, what is the communication protocol[17] used?
- Is it possible to rely on data provided by datasheets or suppliers regarding the performance of electronic components (ESC and motor)?
- Can the theoretical model, representing the interaction between the ESC and the motor, be validated on the basis of experimental measurements?

The answer to these various questions would allow us to answer the final question: from the data provided by the data sheets alone, would it be possible to build a UAV and be able to predict precisely what its performance would be (in terms of flight time or payload that can be sustained)? Moreover, the answer to this last question will allow to validate the second objective of this master thesis, namely the construction of a drone. Indeed, a first design will be drawn from technical specifications. These specifications can be imagined on the basis of the knowledge that can be gained from the theoretical data.

In order to be able to answer all these questions and validate all these objectives, the method to be followed has been divided into three main steps:

- The first step was to understand how it is possible to control a BLDC engine. For this purpose, a prototype (comprising a BLDC motor, an ESC and a raspberry pi) was assembled and computer code was written to communicate with the ESC and ultimately control the motor.
- The second step was to take a series of experimental measurements, using the prototype built in the previous step. These measurements were used to characterise the engine.
- The third and last step was to combine the experimental with the theoretical (theoretical model and data sheets) and try to find a relationship between the two.

Note that each step will be further detailed, in chronological order, in the following sections.

2 BLDC motor control

In this section, the control of a BLDC motor, using an ESC, will be explained. As a reminder, BLDC motors and ESCs were described in sections 2.3 and 2.4 respectively.

---

[17] The communication protocol represents the book of rules that have to be followed.

2.1 Presentation of the final test bench

First of all, the main goal of this section was to understand how a general BLDC motor could be controlled in order to get some measurements later on (see section 3), using the same setup, represented in Figure 17.

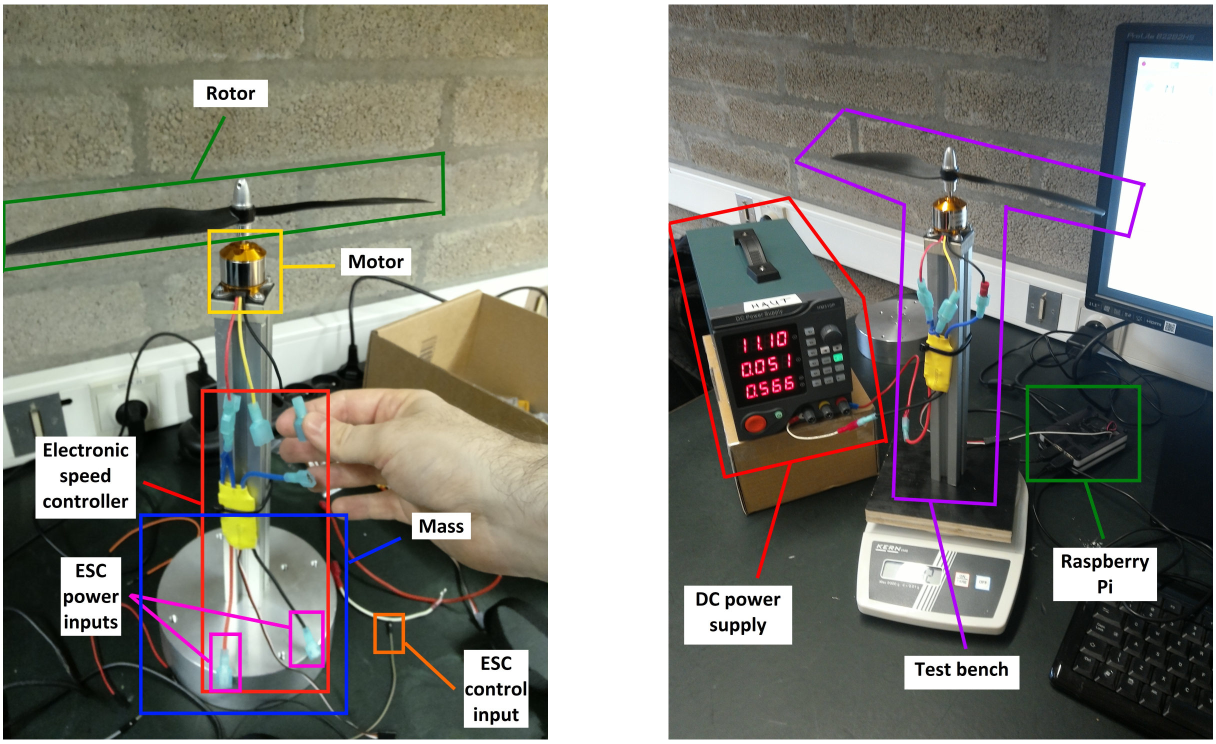


Figure 17: On the left, zoom on the test bench connecting motor and ESC together. On the right, prototype integrated with raspberry pi and power generator.

As it can be seen from the Figure 17, the test bench is composed of the following elements:

- A 5kg mass,
- An ESC composed of two power chords and one control wire,
- A BLDC motor, connected to the ESC thanks to blue connectors,
- A rotor on top of the BLDC motor,
- A DC power supply, injecting power into the ESC by its power inputs,
- A raspberry pi, generating control signals sent through the ESC control input.

The test bench works in the following way: The raspberry pi (RPi) runs a python code (available in Appendix B.1) which outputs a PWM control signal on a specific general-purpose input/output (GPIO) pin of the RPi. The ESC, powered by a DC power supply, receives that PWM control signal on its control input and spins the motor and its rotor. The mass is a safety measure preventing the test bench to take off. A video link of the test bench is available in the Appendix A.1.1.

2.2 Test bench component data

8 The data sheet of the mentioned motor is available in Appendix A.3.1. This one will be named A2212 motor hereafter. This motor is controlled by an ESC whose data sheet is also available in Appendix A.3.2. In the same way, this ESC will be named HW30 ESC in the continuation of the presentation. Those two components are displayed in Figure 18.

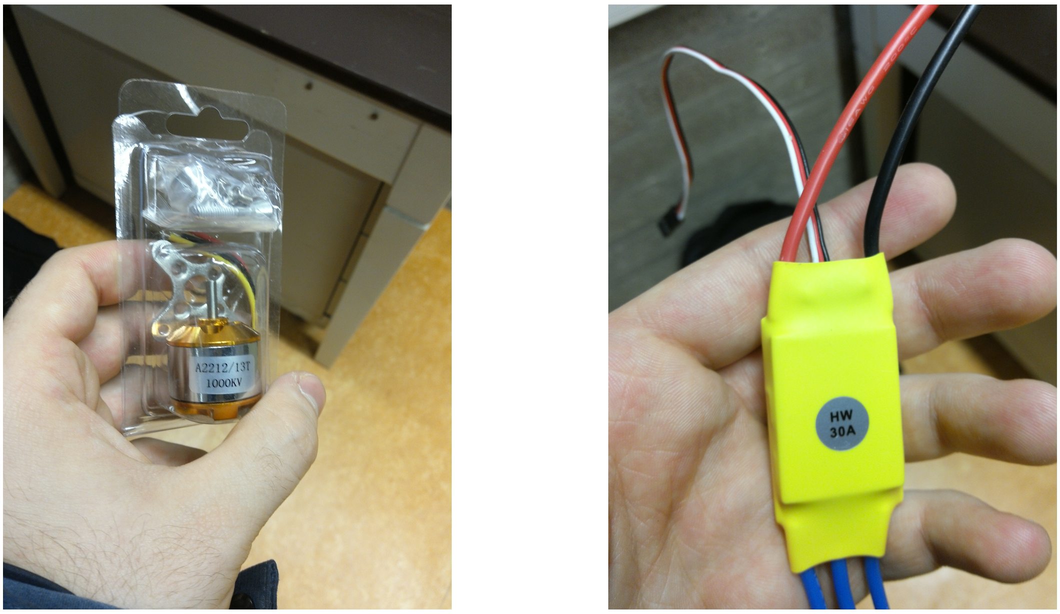


Figure 18: On the left, A2212 motor while on the right HW30 ESC. They are part of the test bench.

2.3 Raspberry pi goal

You could, at this stage of the presentation, wonder about the usefulness of the raspberry pi. Indeed, wouldn't it be possible to do without the control signal and simply feed the ESC to make the motor turn (even at a speed not controlled by the user)? The answer to this question is no because the ESC first goes through a calibration phase! Indeed, as it was explained in the first chapter (see 2.5.2), the ESC controls the power input to the motor (and therefore its speed). The control comes from upstream, the FC in general, which sends it a control signal by means of a well-defined protocol. The ESC must be able to decipher the signals sent by the FC, in particular which signal means that the motor must run at full speed (duty cycle = 100%) or stop completely (duty cycle = 0%)[18]. Therefore, before being able to run an motor, each ESC communicates with the FC in order to receive from it the information allowing it to understand what will be the representation of a high or low signal. In the test bench, the FC is replaced by the raspberry pi.

2.4 Code description

In this section, the python code (see Appendix B.1) will be described in broad outline. This description will be divided logically according to the different functions and the main body of the computer code. Note that the computer code evolved as the protocol was understood. One of the key factors in finalizing the code was to move from hardware to software signal generation. The development of the computer code was based in part on the work of others, found on the internet [41][42][43][44][45][46].

2.4.1 *MAIN* body

The main body of the code has to initialize the pins of the raspberry that will be used to send the logic signal and to create a PWM object that will allow to generate PWM signals of controllable amplitude in the *calibration* and *manual* functions. It will also implement an infinite loop in which the user can choose one of these modes, until *stop* mode is selected.

The first thing to do is to declare the type of numbering that will be used when designating pin numbers. In fact, there are 2 different numbering schemes for the raspberry pi, namely BOARD (physical numbers written on the card) and BCM (electronic numbering of the chip - useful for working with the RPi.GPIO library), as shown in Figure 19.

---

[18] The duty cycle is a parameter indicating what fraction of the battery power is at the motor terminals.

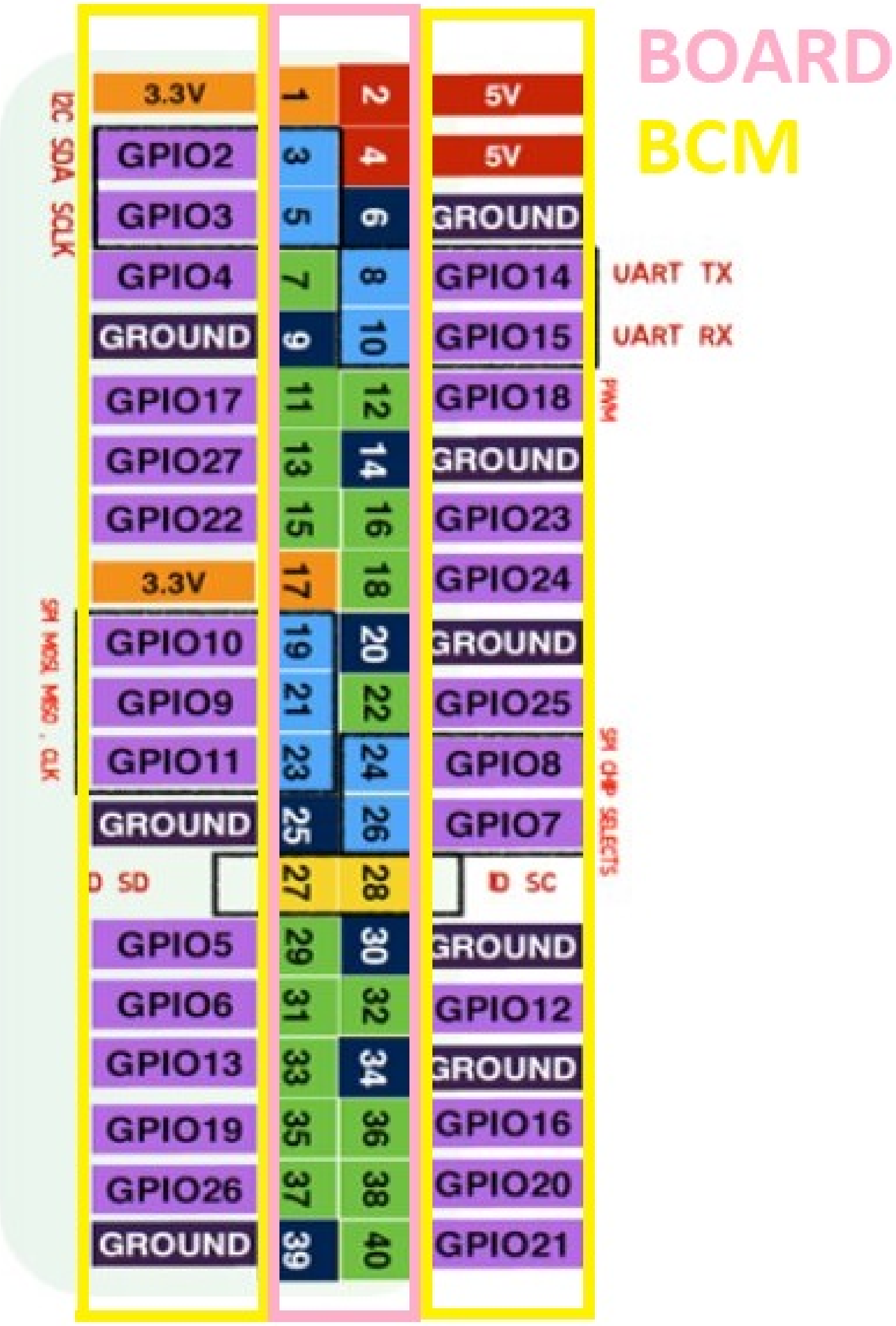


Figure 19: Raspberry pinout: BOARD numbering (Pink) vs BCM numbering (Yellow)
Sources: https://iot4beginners.com/difference-between-bcm-and-board-pin-numbering-in-raspberry-pi/

Here, we will use the board numbering and assign the keyword *control_pin* to pin 12 (the sixth one on the top right, according to the Figure19). It is through this pin that our future control signals will come out.

In order for this pin to output signals, it must be declared as an output pin. This is done by calling the *setup* function of the GPIO library. It is also made sure that its initial value is null (to avoid triggering communication with the ESC at this stage).

Once this is done, it remains to create a PWM object which will be used to generate PWM signals on *control_pin*. This is done by using the *pwm* function of the GPIO library [47].

Once the initialization is done, a loop is entered where the user has the choice between several modes:

- calibration, must be executed first, only once
- manual, can be executed after the calibration is done
- stop, can be executed at any time, allows to quit the program

These modes are implemented by functions with the same name. Note that if the user misspelled (unintentionally or not) the mode that they want to enter, the program will automatically stop.

2.4.2 *calibration()* function

The goal of the calibration is to send two different PWM signals to the ESC, the first indicating the maximum value, the second indicating the minimum value. After receiving these two signals, the ESC will have the range of possible values of the future control signals and will be able to translate them into duty cycle.

Calibration is carried out as follows: first, the PWM signal, generated at all times on the *control_pin*, is set to zero. In addition, the voltage source supplying the ESC must be turned off, which will be validated by the program user by pressing the "Enter" key on their keyboard. Once this step is validated, the PWM max signal, *DUTY_MAX*, is generated on *control_pin* and then the user is asked to turn on the voltage source, to wait for the validation beep indicating that the ESC has correctly interpreted the PWM signal and then to validate this step by pressing "Enter" again.
Once the PWM max signal is received, the PWM min signal,*DUTY_MIN*, is generated on *control_pin* and the "Enter" button is pressed once the three confirmation beeps are emitted, meaning that the calibration has been successful.

It is important to realize that the protocol is methodical. Indeed, the first signal received by the ESC must be the PWM max signal, and not an electrical noise, for example. This is the reason why the maximum signal is generated while the ESC is disconnected and it is only fed once the control signal is on the right output pin. Therefore, this ensures that the first signal received by the ESC is the PWM max signal.

Note that the program is put in blocking mode each time the user must press Enter. The program can only move on to the next step if the Enter button is actually pressed.

2.4.3 *manual()* function

The goal of manual is to allow the user to run the motor at a specific duty cycle. Note that, as mentioned above, the calibration phase must have been successful beforehand, otherwise running the manual mode will not allow the motor to run.

The manual function works as follows: the user is prompted to enter the desired duty cycle value (percentage), which must be in the range [0, 100]. After verification, if the duty cycle entered is valid, a value will be computed (accordingly to the selected duty cycle, see Equation (21) below) and will be used by the *start* function of the PWM object. Then, the motor will run as long as the user does not press "Ctrl" and "c" simultaneously. Once this combination of keys has been entered by the user, they have the choice between restarting the manual function directly and entering a new duty cycle by pressing "c", setting off the PWM signal on *control_pin* and going back to the main loop by pressing "q" or quitting the program (and therefore starting the stop function) by pressing another key than "c" and "q".

2.4.4 *stop()* function

The goal of the stop function is quite simple: it allows to stop the program: the PWM signal on *control_pin* is turned off and the all the pins of the raspberry pi are cleaned (set to default value). After that, the program closes due to the break instruction in the main loop.

2.4.5 Parameters values and duty cycle computation

Along the whole program, four parameters are used:

- *FREQUENCY*: frequency of the software PWM control signal.
- *DUTY_MIN*: minimum duty cycle of the software PWM control signal.
- *DUTY_MAX*: maximum duty cycle of the software PWM control signal.
- *DUTY_INC*: increment between two consecutive available (duty cycle) values of the software PWM signal.

The HW30 data sheet, in the control signal part, gives some useful information to initialise those parameters. Regarding the frequency, it is stated that: "30A BLDC ESC requires standard 50-60Hz PWM signal from any remote control as throttle input". The *FREQUENCY* parameter is therefore set to 50Hz. Note that it corresponds to a time period of 20ms.

Regarding the minimum and maximum signals, it is stated that: "Throttle speed is proportional to the width of the pulse. Maximum throttle position is user programmable. In general throttle is set at zero for 1ms pulse width and full at the 2ms pulse width". Based on the pulse width and frequency, it is possible to determine the duty cycle value of the software PWM signal as follows:

$$\delta_{PWM} = \frac{pulse_{duration}}{T_{PWM}} \cdot 100 \tag{17}$$

where:

- δ_{PWM} is the duty cycle (percentage) of the software PWM signal.
- $pulse_{duration}$ is the duration of the pulse, in ms.
- T_{PWM} is the period of the software PWM signal, in ms.

Based on this Equation (17), it is possible to compute the minimum and maximum duty cycle values for the software PWM signal:

$$DUTY\_MIN = \frac{1}{20} \cdot 100 = 5\% \tag{18}$$

$$DUTY\_MAX = \frac{2}{20} \cdot 100 = 10\% \tag{19}$$

It is important to note that the software values of the duty cycle do not correspond directly to the real duty cycle values of the motor. In fact, the software value of 5% (resp. 10%) assigned to DUTY_MIN (resp. DUTY_MAX) corresponds to the minimum (resp. maximum) real duty cycle value of the motor, i.e. 0% (resp. 100%).

Now that the limits of the range of possible values for the duty cycle are known, it is necessary to divide this range into 100 equal parts. This allows the user to choose a duty cycle ranging from 0% to 100%, with a resolution of 1%. This resolution of 1% is represented by the parameter *DUTY_INC*, whose value can be computed from *DUTY_MAX* and *DUTY_MIN* as follows:

$$DUTY\_INC = \frac{DUTY\_MAX - DUTY\_MIN}{100} = \frac{10\% - 5\%}{100} = 0.05\% \tag{20}$$

Note that this parameter allows to make the link between software and motor duty cycle values.

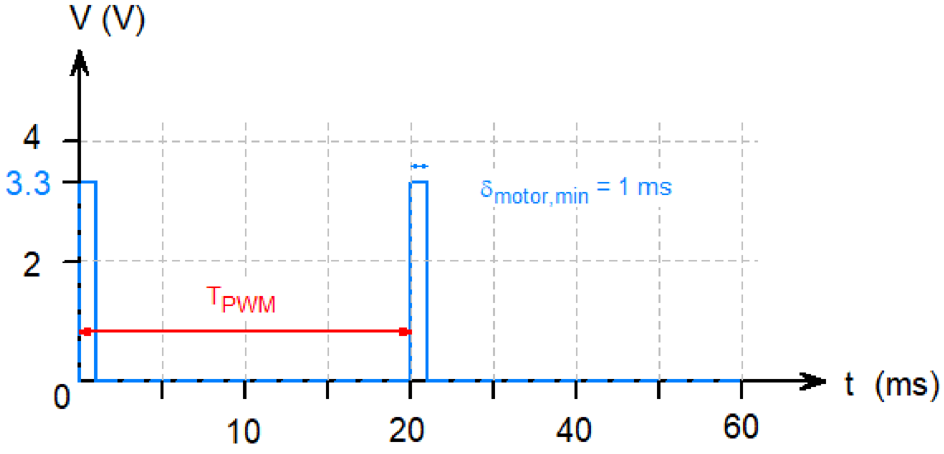
Based on Equations (18), (19) and (20), it is now possible to match the duty cycle selected of the motor δ_{motor} and the duty cycle of the software PWM signal DUTY_MOTOR:

$$DUTY\_MOTOR = DUTY\_MIN + \delta_{motor} \cdot DUTY\_INC \tag{21}$$

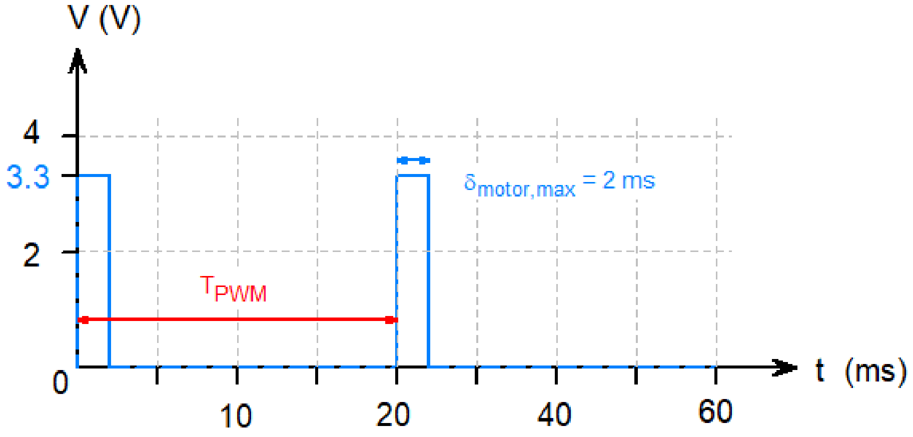
where:

- δ_{motor}: it is the duty cycle (percentage) of the running motor.
- DUTY_MOTOR: it is the software conversion of δ_{motor}.

For instance, using Equation (21), a duty cycle δ_{motor} of 30% corresponds to a software duty cycle value DUTY_MOTOR of $5 + 30 \cdot 0.05 = 6.5\%$.



(a) PWM signal corresponding to $\delta_{motor,min} = 0\%$, i.e. to a pulse of 1ms.



(b) PWM signal corresponding to $\delta_{motor,max}$=100%, i.e. to a pulse of 2ms.

Figure 20: Extreme values of δ_{motor} and the corresponding PWM signals, represented in the software by the variable PWM_MIN and PWM_MAX.

2.5 Hardware vs Software PWM signal

In the previous section, it was quickly mentioned that the PWM signal used was of software origin and not hardware. In fact, like most other microcontrollers, the raspberry pi is able to generate analog signals, such as PWM signals. Unlike digital signals that can be generated on any GPIO pin, analog PWM signals can only be generated on pins 12, 32, 33 and 35 as explained in [48]). Moreover, this requires the use of the pigpio library which is more difficult to handle than the GPIO library for digital signals. However, it should be noted that analog PWM signals are much more accurate than digital PWMs, but the latter are still accurate enough to communicate with the ESC, as stated in [49]. The digital PWM signals have therefore been preferred in this application for their ease of use.

3 Experimental measurements

In this section, the test bench presented in section 2 will be used with different measuring devices.

First of all, the measuring devices will be introduced as well as the data that can be obtained from them. After that, these experimental measurements will be combined in order to get the characteristic values of the motor-rotor pair.

3.1 Measuring devices

In order to be able to characterize the motor-rotor pair later (see section 3.2), it is essential to quantify the thrust that the motor can provide and the speed at which the rotor can turn. The thrust will be measured with a scale while the speed measurements will be carried out with a tachometer. The integration of these two measuring instruments in the test bench is shown in Figure 21. The current as well as the voltage are also important parameters to acquire. This will be carried out by the laboratory power supply, which can display the current provided as well as the power consumption for the selected voltage.

3.1.1 Scale

The scale, displayed in Figure 22, is a high-precision scale whose specifications can be found in Appendix A.3.3). The maximum weight that the scale can support without going out of adjustment is limited to 2kg, which is precisely why a wooden plate replaces the previous 5kg aluminum mass.

3.1.2 Tachometer

The tachometer,displayed in Figure 23, is a non-contact digital one whose specifications can be found in Appendix A.3.4).

The device has two different measurement modes:

- counting each rotation,
- averaging over a minute to give the RPM value (the one of interest here). It is limited to 99,999 RPM but this is sufficient in the case of this experiment.

For a correct measurement, it is normally necessary to place the tachometer to a location 50-200 mm from the rotating device, on which a reflective strip has been placed.
The reflective strip being not available at the time of the experiment, an aluminium strip, polished on one side, was used instead. The purpose was to have a material reflective enough to reflect the laser emitted by the tachometer. However, it has been shown during experiments that the tachometer had to be placed very close to the rotating device in order to acquire valid measurements.

3.1.3 Data and experiment process

The goal was to acquire data about thrust and speed, according to the duty cycle of the motor. Knowing that at low duty cycle, the motor produces almost no thrust and that the ones used by UAVs usually run with a duty cycle varying from 50% to 100%, only this range of interest has been studied.
It could be interesting to complete the experimental values by studying the motor in the [0%, 50%] duty cycle range in order to verify if the behaviors that can be extracted from the studied range are also verified in this new range. This could be the subject of a later study.

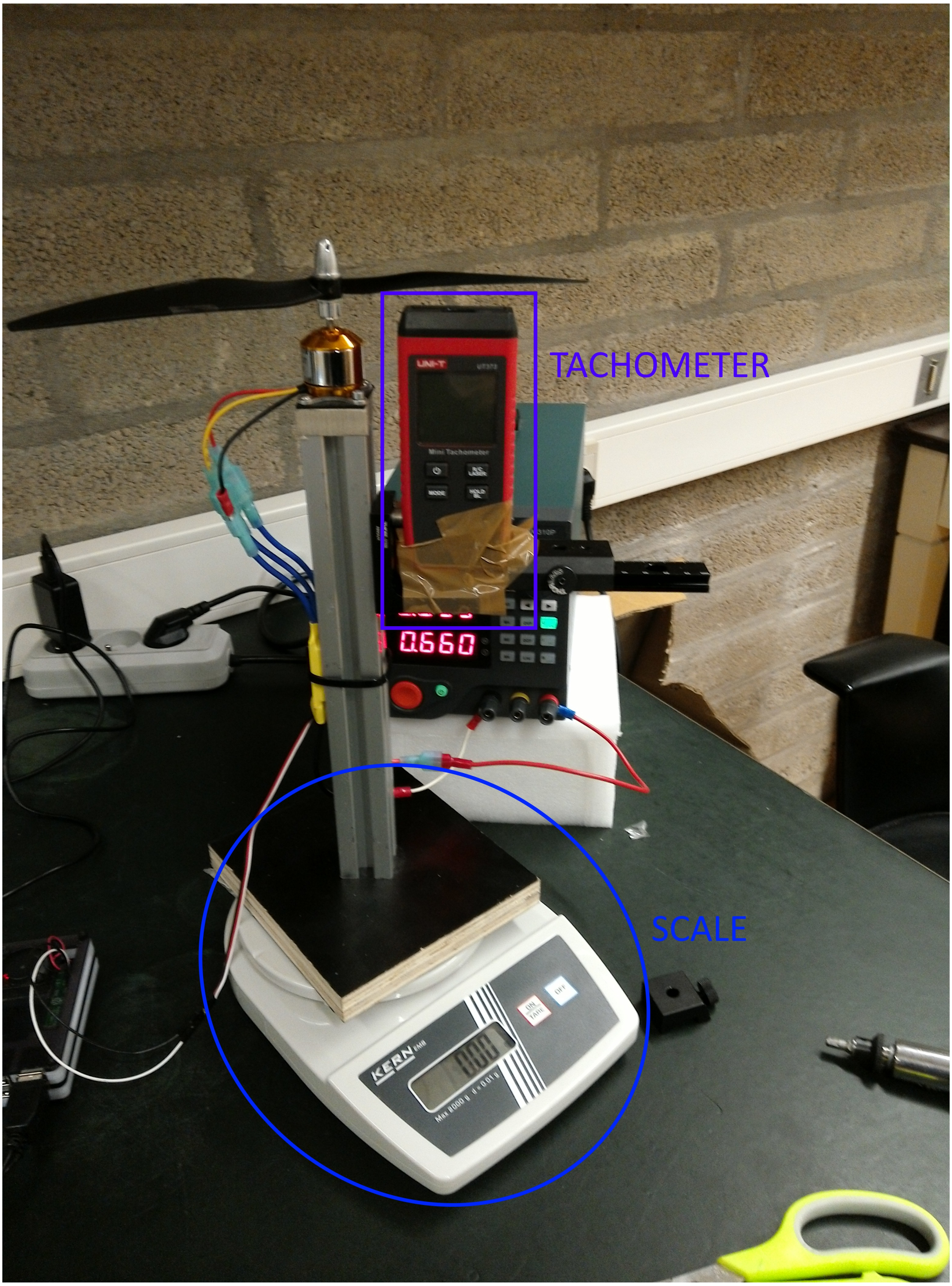


Figure 21: The scale, in blue, and the tachometer, in purple, integrated in the test bench

Figure 22: High precision scale, KERN EMB 2000-2 model.
Sources: https://www.kern-sohn.com/fr/EMB

Figure 23: Mini non-contact digital tachometer, UT373 model.
Sources: https://www.amazon.com/UNI-T-UT373-Non-Contact-Digital-Tachometer/dp/B07P5S1BGV

Since the ESC accepts supply voltages in the range [2S, 4S][19] as indicated in its data sheet, 3S (i.e 11.1V) was designated as the reference voltage for the experiments. However, under normal circumstances, the voltage supplied to the ESC comes from a battery that discharges over time and not from a laboratory power supply voltage, which remains constant over time. In order to take into account the effect of battery discharge, three cases were studied during the experiments:

- Case 1 - low battery : the cells of the battery are considered to be at their minimum safe charge. In the case of a LiPo battery, a cell is said to be at minimum safe charge when its voltage is around 3V. Knowing that the reference voltage for the experiments is 3S, a voltage of: $3 \cdot 3 = 9$V will be provided to the ESC. However, since the ESC has a low-voltage protection threshold (it checks automatically if the voltage of the battery is normal, according to the number of cells. If not, the communication is directly aborted), this scenario could not be tested. Therefore, an intermediate case between low battery and normal battery has been tested. In this one, the cells have a voltage between the low and normal level, i.e. ≈ 3.3V. The ESC is thus supplied with a voltage of ≈ 10V.
- Case 2 - normal battery : the cells of the battery are considered to be at nominal voltage. In the case of a LiPo battery, a cell is said to be at nominal voltage when its voltage is around 3.7V. Knowing that the reference voltage for the experiments is 3S, a voltage of: $3 \cdot 3.7 = 11.1$V will be provided to the ESC.

---

[19] Remember from section 2.7.2 that voltage is described in terms of cells, where 1S = 3.7V for a LiPo battery. This means that the ESC accepts supply voltages in the range [7.4; 14.8V] here.

- Case 3 - high battery : the cells of the battery are considered to be fully charged. In the case of a LiPo battery, a cell is said to be fully charged when its voltage is around 4.2V. Knowing that the reference voltage for the experiments is 3S, a voltage of: $3 \cdot 4.2 = 12.6$V will be provided to the ESC. In practice, a voltage of 12V has been selected.

The Tables 9, 10 and 11 contain all the data measured on the test bench, considering only the relevant duty cycle interval and the three cases described above. These data were obtained using the manual mode (after calibration) of the python code already described in a previous section (see section 2.4). Note that the Google spreadsheet, containing original data and graphs, is available, in reading mode, through a link in Appendix A.2.1.

CASE 1: Low battery (10V)				
Duty Cycle [%]	Current [A]	Power [W]	Thrust [g]	Blade Speed [RPM]
40	1.35	13.5	120	-
45	1.73	17.3	150	-
50	2.20	22	190	-
55	2.70	27	235	-
60	3.30	33	265	-
65	4.30	43	310	-
70	4.75	47.5	354	-
75	5.60	56	394	-
80	6.50	65	435	-
85	7.50	75	475	-
90	8.20	82	505	-
95	9.30	93	545	-
100	9.90	99	570	-

Table 9: Experimental data acquired for case 1. Note that RPM measurements have not been acquired due to time constraints.

CASE 2: Normal battery (11.1V)				
Duty Cycle [%]	Current [A]	Power [W]	Thrust [g]	Blade Speed [RPM]
40	1.58	17.5	147	-
45	1.99	22.1	180	-
50	2.28	25.3	204	-
55	2.52	28	217	3800
60	2.93	32.5	245	4060
65	3.64	40.4	304	4300
70	4.41	49	353	4550
75	5.14	57	388	4835
80	6.04	67	437	5003
85	6.94	77	479	5270
90	7.84	87	523	5470
95	8.83	98	558	5673
100	9.64	107	587	5840

Table 10: Experimental data acquired for case 2. Note that RPM for duty cycles $\leq 50\%$ have not been acquired because the tachometer did not detect anything for those values.

CASE 3: High battery (12V)				
Duty Cycle [%]	Current [A]	Power [W]	Thrust [g]	Blade Speed [RPM]
40	1.83	22	179	3445
45	2.33	28	218	3787
50	2.92	35	270	4000
55	3.58	43	315	4395
60	4.33	52	365	4721
65	5.17	62	415	5020
70	6.17	74	465	5300
75	7.25	87	515	5523
80	8.33	100	560	5740
85	9.17	110	590	6130
90	9.75	117	625	6200
95				
100				

Table 11: Experimental data acquired for case 3. Note that there are no acquired parameters for duty cycles $\geq 95\%$ because the power supply had a current limit of 10A.

Based on all those measurements data, three interesting graphs can be drawn for the three considered cases. These graphs will be analysed in the following sections. All of these graphs will refer to thrust and power, but provide different views of the problem, for a better understanding of the situation.

3.1.3.1 Evolution of power and thrust, regarding the duty cycle

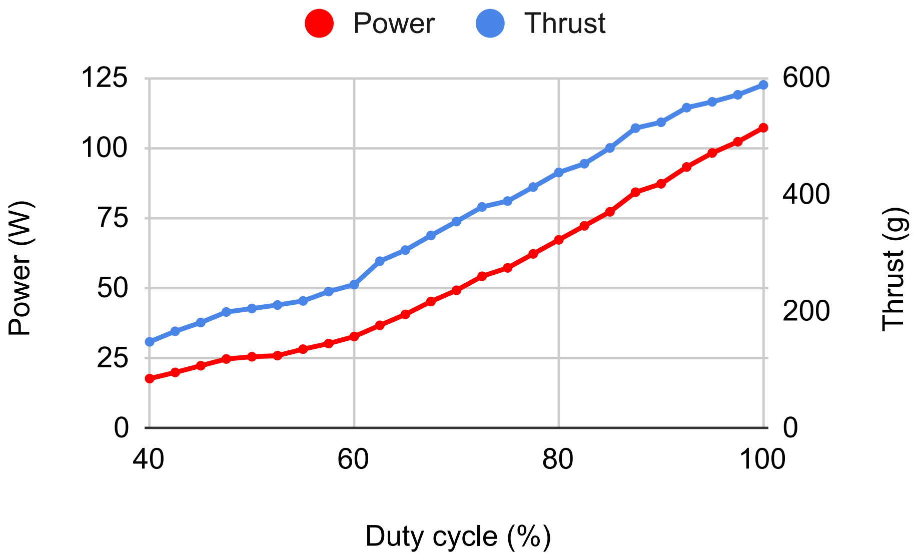
The graphs corresponding to the three cases are illustrated in Figure 24, from which the following conclusions can be drawn:

- The supply voltage does not seem to modify the global evolution of the thrust and the consumed power, i.e. the same behavior is found in the three cases.
- The evolution of the thrust as well as that of the power according to the duty cycle is not linear but rather quadratic. Indeed, doubling the duty cycle (from 40% to 80%) increases the thrust by a factor 2.97, 3.62 and 3.13, while the power consumed increases by a factor 3.83, 4.81 and 4.54 for case 2, 1 and 3 respectively.
- Cases 1 and 2 do not differ a lot since the input voltage is nearly the same. A bigger difference can be observed for duty cycle $\leq 60\%$.
- Case 3 differs from the two other cases as the duty cycle increases. Indeed, at duty cycle $= 40\%$ (resp. 90%), there is a difference of $179 - 147 = 32$g (resp. $625 - 523 = 102$g) for the thrust and $22 - 17.5 = 4.5$W (resp. $11 - 87 = 30$W) for the consumed power between cases 3 and 2. The difference factor, between the two cases, was therefore multiplied by ≈ 3 (resp. ≈ 6) regarding the thrust (resp. power consumed). The efficiency of the motor being calculated as the ratio $\frac{T}{P}$, it is expected that the efficiency of case 3 will be lower than for case 2 (see Figure 26), with the difference becoming more and more pronounced as the duty cycle increases.

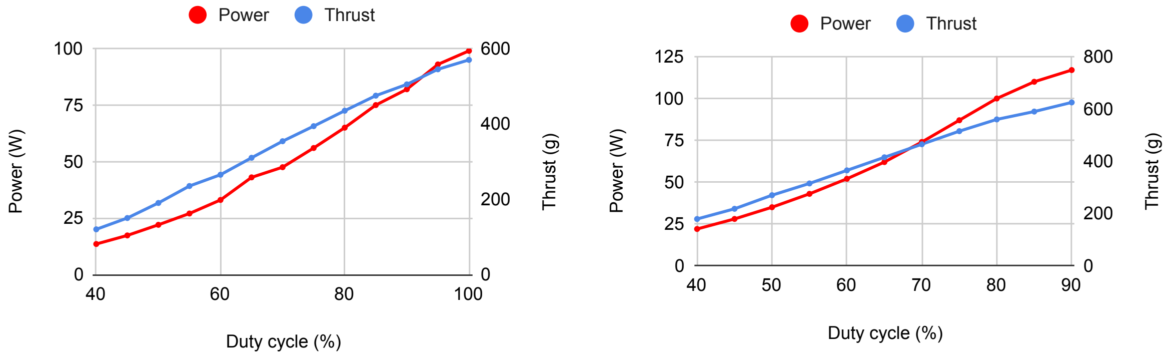
3.1.3.2 Evolution of the thrust, regarding the power

The graphs corresponding to the three cases are illustrated in Figure 25. From these graphs, the following conclusions can be drawn:

- The battery charge does not seem to modify the global evolution of the thrust and the consumed power, i.e. the same behavior is found in the three cases.



(a) Case 2: Normal battery charge.



(b) Case 1: Low battery charge.

(c) Case 3: High battery charge.

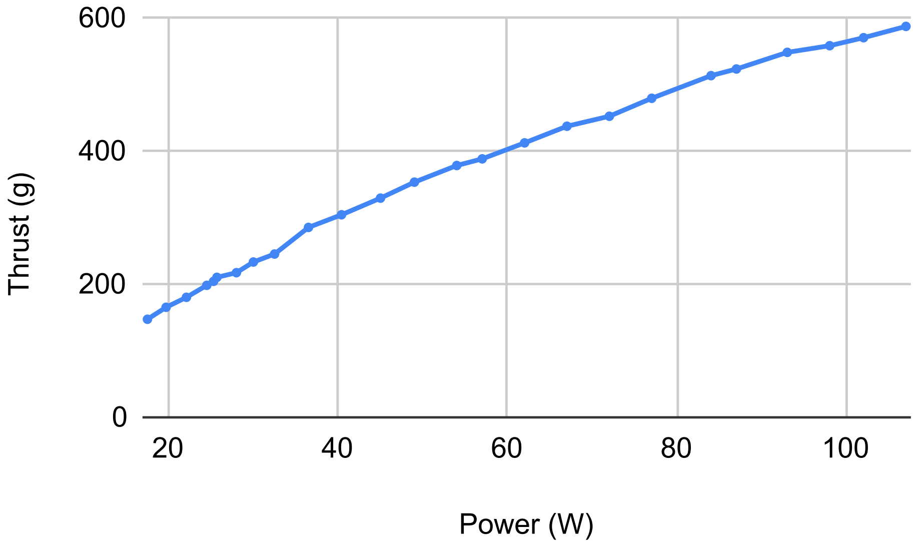
Figure 24: Evolution of power consumed P (red) and thrust T (blue) in function of the duty cycle

- The relation between the power consumed and the thrust generated is not linear. The slope of the graph being less than 45 degrees, one can conclude that the power consumed increases more rapidly than the thrust generated, i.e. the more thrust that must be generated, the more it will cost in terms of power.
- The main purpose of these graphs is to calculate what would be the power consumed by the motor by setting the thrust it must generate. In other words, by setting the thrust to 400g, the engine consumes ≈57W, ≈59.5W and ≈59W for case 1, 2 and 3 respectively. It allows to conclude that, for a thrust of 400g, case 1 has the best efficiency, since it consumes less than the two others.

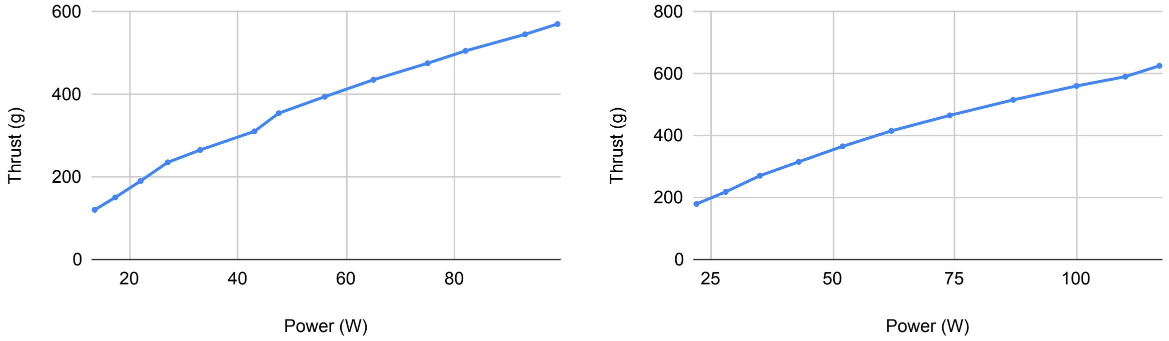
3.1.3.3 The evolution of the efficiency, $\frac{T}{P}$, regarding the thrust

The graphs corresponding to the three cases are illustrated in Figure 26. From these graphs, one can draw the following conclusions:

- The supply voltage does not seem to modify the global evolution of the thrust and the consumed power, i.e. the same behavior is found in the three cases.
- The more the thrust increases, the more the efficiency decreases, which can be linked to what has been said before, i.e. the power consumed increases more rapidly than the thrust generated.



(a) Case 2: Normal battery charge



(b) Case 1: Low battery charge

(c) Case 3: High battery charge

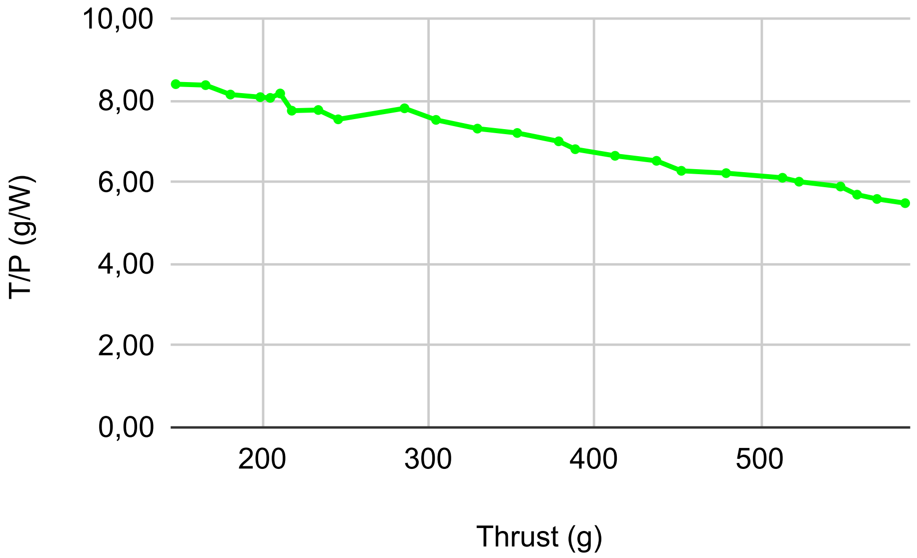
Figure 25: Evolution of thrust T (blue) in function of the power consumed P

- One can notice that the efficiency is almost constant (i.e. ≈ 8) for thrust $\leq$ 300g, which means that running three motors producing a thrust of 100g each will consume exactly the same as running only one motor generating a thrust of 300g. On the other hand, if the aim is to generate 600g of thrust, it is better to have two motors producing 300g of thrust each (and thus consuming 39.5W/unit so 79W in all) than to have a single motor providing 600g while consuming more than 107W (motor only produces 587g of thrust at 100%). Note that these calculations were made on the basis of data from case 2, but the conclusions drawn are still valid for the other cases.

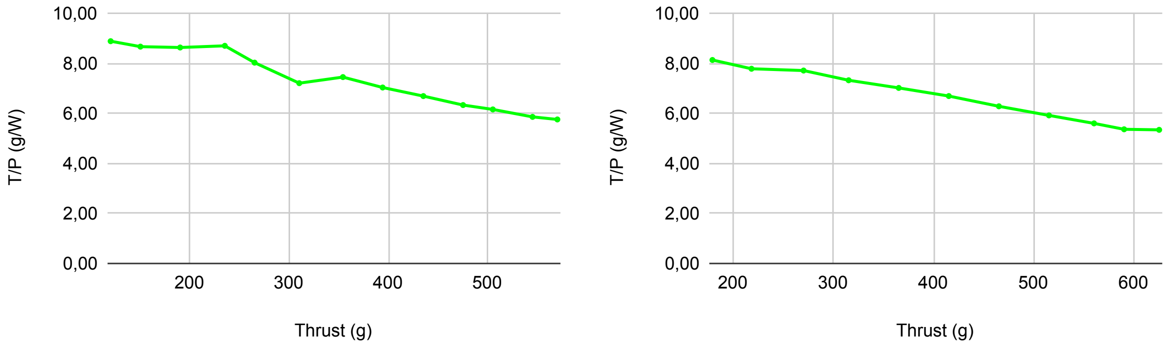
The efficiency $\frac{T}{P}$ is given as information in Table 12 for all three cases.

Efficiency T/P													
Duty cycle	40	45	50	55	60	65	70	75	80	85	90	95	100
Case 1: 10V	8.89	8.67	8.64	8.70	8.03	7.21	7.45	7.04	6.69	6.33	6.16	5.86	5.76
Case 2: 11.1V	8.40	8.14	8.06	7.75	7.54	7.52	7.20	6.81	6.52	6.22	6.01	5.69	5.49
Case 3:12V	8.89	8.67	8.64	8.70	8.03	7.21	7.45	7.04	6.69	6.33	6.16	5.86	5.76

Table 12: Evolution of the motor efficiency, $\frac{T}{P}$, in function of the duty cycle for all cases.



(a) Case 2: Normal battery charge



(b) Case 1: Low battery charge

(c) Case 3: High battery charge

Figure 26: Evolution of efficiency $\frac{T}{P}$ (green) in function of the thrust T

3.2 Characterisation of the motor-rotor pair

A motor rotor pair is fully qualified by two parameters known as:

- The thrust coefficient C_T
- The power coefficient C_P

Knowing the parameters related to the design of the rotor and the fluid in which it moves, it is possible to find the thrust generated (resp. the power required) by the motor-rotor pair from the thrust (resp. power) coefficient C_T (resp. C_P).

In the following, the aim is to calculate these two coefficients using the data collected in the previous section. The mathematical relations will be linked to refinements, which will evolve over time in order to model reality as closely as possible[20]. The mathematical equations modelling these coefficients as well as their notations (under small modifications for convenience) have been taken from a master thesis [50].

---

[20] In this work, this will be done only for the power coefficient.

3.2.1 General case

Without considering the rotor tilt, the thrust $T^{(N)}_{rotor}$ and the ideal power P_{id} required by the rotor can be modelled by the following equations:

$$T^{(N)}_{rotor} = C_T \cdot \rho_{fluid} \cdot A \cdot (R \cdot \Omega_{rotor})^2 \tag{22}$$

$$P_{id} = C_{P,id} \cdot \rho_{fluid} \cdot A \cdot (R \cdot \Omega_{rotor})^3 \tag{23}$$

where:

- $T^{(N)}_{rotor}$: the thrust [N],
- C_T: the thrust coefficient [−],
- ρ_{fluid}: the fluid mass density [kg · m^{-3}],
- A: the rotating disc area [m^2],
- R: the rotor radius [m],
- Ω_{rotor}: the rotational speed of the rotor (i.e. angular speed) [s^{-1}],
- P_{id}: the ideal power [W]^{21},
- $C_{P,id}$: the ideal power coefficient [−].

Note that both coefficients are dimensionless. Indeed, isolating the coefficients from Equations (22) and (23), then rewriting the relation in terms of units gives:

$$C_T = \frac{T^{(N)}_{rotor}}{\rho_{fluid} \cdot A \cdot (R \cdot \Omega_{rotor})^2} \xleftrightarrow{\text{units}} \left[\frac{N}{kg \cdot m^{-3} \cdot m^2 \cdot (m \cdot s^{-1})^2}\right] = \left[\frac{kg \cdot m \cdot s^{-2}}{kg \cdot m \cdot s^{-2}}\right] = [-] \tag{24}$$

$$C_{P,id} = \frac{P_{id}}{\rho_{fluid} \cdot A \cdot (R \cdot \Omega_{rotor})^3} \xleftrightarrow{\text{units}} \left[\frac{W}{kg \cdot m^{-3} \cdot m^2 \cdot (m \cdot s^{-1})^3}\right] = \left[\frac{N \cdot m \cdot s^{-1}}{kg \cdot m^2 \cdot s^{-3}}\right] = [-] \tag{25}$$

3.2.1.1 Application to experimental data

Knowing the radius of the rotor $R = 12.7 \cdot 10^{-2}$, the rotating disk area A can be calculated:

$$A = \pi \cdot R^2 = \pi \cdot (12.7 \cdot 10^{-2})^2 \approx 5.067 \cdot 10^{-2} \tag{26}$$

The rotational speed of the rotor can be obtained for the RPM values measured earlier by the following transformation:

$$\Omega_{rotor} = \frac{\text{RPM} \cdot 2\pi}{60} \tag{27}$$

The Tables 13 and 14 have been created, using Equation (27), from RPM values contained in 10 and 11 respectively. Note that, since blade speed has not been measured for case 1, there will be no angular speeds for that particular case.

CASE 2: Normal battery (11.1V)										
Duty cycle [%]	55	60	65	70	75	80	85	90	95	100
Angular speed [s^{-1}]	397.93	425.16	450.29	476.47	506.32	523.91	551.87	572.82	594.07	611.56

Table 13: Rotational speed of the rotor, computed from experimental data, for case 2

Since the fluid in which the test bench is immersed is air, the fluid density $\rho_{fluid} = \rho_{air} \approx 1.225$.
Now that all the parameters qualifying the motor-rotor pair are known, it is possible to derive the thrust and power coefficients from Equations (24) and (25). The results are displayed in Tables 15 and 16 for cases 2 and 3 respectively.

For a better visual representation, these values are also represented in the Figure 27.

CASE 3: High battery (12V)											
Duty cycle [%]	40	45	50	55	60	65	70	75	80	85	90
Angular speed [s^{-1}]	360.76	396.57	418.88	460.24	494.38	525.69	555.01	578.37	601.09	641.93	649.26

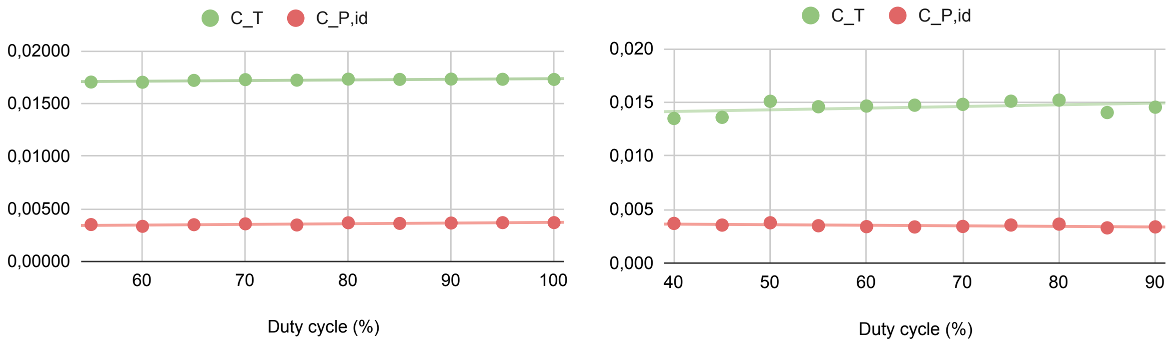
Table 14: Rotational speed of the rotor, computed from experimental data, for case 3.

CASE 2: High battery (11.1V)										
Duty cycle [%]	55	60	65	70	75	80	85	90	95	100
C_T [-]	0.0134	0.0133	0.0147	0.0152	0.0148	0.0156	0.0154	0.0156	0.0155	0.0154
$C_{P,id}$ [-]	0.00349	0.00333	0.00348	0.00356	0.00345	0.00366	0.00360	0.00364	0.00368	0.00368

Table 15: Determination of the thrust and ideal power coefficients, from experimental data, for case 2.

CASE 3: High battery (12V)											
Duty cycle [%]	40	45	50	55	60	65	70	75	80	85	90
C_T [-]	0.0135	0.0136	0.0151	0.0146	0.0146	0.0147	0.0148	0.0151	0.0152	0.0140	0.0145
$C_{P,id}$ [-]	0.00369	0.00353	0.00375	0.00347	0.00338	0.00336	0.00340	0.00354	0.00362	0.00327	0.00336

Table 16: Determination of the thrust and ideal power coefficients, from experimental data, for case 3.



(a) Case 2: Normal battery charge

(b) Case 3: High battery charge

Figure 27: Thrust and ideal power coefficients, from experimental data, for case 2 and 3

By looking at the means, it gives $C_T = 0.0149$ (resp. 0.0145) and $C_{P,id}$ =0.00356 (resp. 0.00349) for case 2 (resp. 3). This similarity in numerical values could be explained by the fact that C_T and $C_{P,id}$ primarily depend on the advance ratio λ (ratio between the voltage and the angular speed), the blade Reynolds number Re, and on the prop geometry (i.e. $C_T = C_T(\lambda, Re, geometry); C_P = C_P(\lambda, Re, geometry)$) as explained in those lecture notes from MIT [51]. Only the advance ratio differs from the two cases: It has a mean value of 0.0167 for case 2, while it is equal to 0.0149 for case 3. Since these two measurements are very similar, this could explain why the values of C_T and $C_{P,id}$ are close in the two scenarios but the real reason for this difference could have another origin. This point will not be studied further.

3.2.2 Drag force

In this section, the rotor tilt will be considered in order to refine the Equations (22) and (23).

Considering the power coefficient, the Equation (23) does not take into account the drag force on the blade element. The main effect of the drag is to generate a torque on the blade element that needs to be compensated by requiring more power from the rotor side. Therefore, Equation (23) can be rewritten as follows:

$$C_{P,drag} = \frac{C_T^{3/2}}{\sqrt{2}} + \frac{1}{8} \cdot \sigma_{rotor} \cdot C_d \tag{28}$$

where:

---

[21] This is the power that the rotor must provide in order to get a thrust $T_{rotor}^{(N)}$ at the end.

- $C_{P,drag}$: power coefficient, considering drag effect $[-]$,
- σ_{rotor}: rotor solidity $[-]$,
- C_d: drag coefficient $[-]$.

The rotor solidity, σ_{rotor}, mentioned here, is the ratio between the blade area and the disk area created by its rotation. In terms of equations, it gives:

$$\sigma = \frac{\text{Blade area}}{\text{Disk area}} = \frac{N_b \cdot c_b \cdot R}{\pi \cdot R^Z} = \frac{N_b \cdot c_b}{\pi \cdot R} \tag{29}$$

where:

- N_b: number of blades $[-]$,
- c_b: chord length of the blade [m].

The thrust coefficient can also be improved. Indeed, the latter is modeled from an integral taking into account the rotor solidity, the induced influx ratio and the angle of attack. This improved version of the thrust coefficient will not be considered in this work as it is outside of the scope.

3.2.2.1 Application to experimental data

Knowing that the rotor is composed of two blades (i.e.$N_b = 2$) and that the chord length c_b of the blades is precisely equal to its radius R, the formula expressing the solidity σ of the rotor (29) can be re expressed such as:

$$\sigma = \frac{N_b}{\pi} = \frac{2}{\pi} \approx 0.636 \tag{30}$$

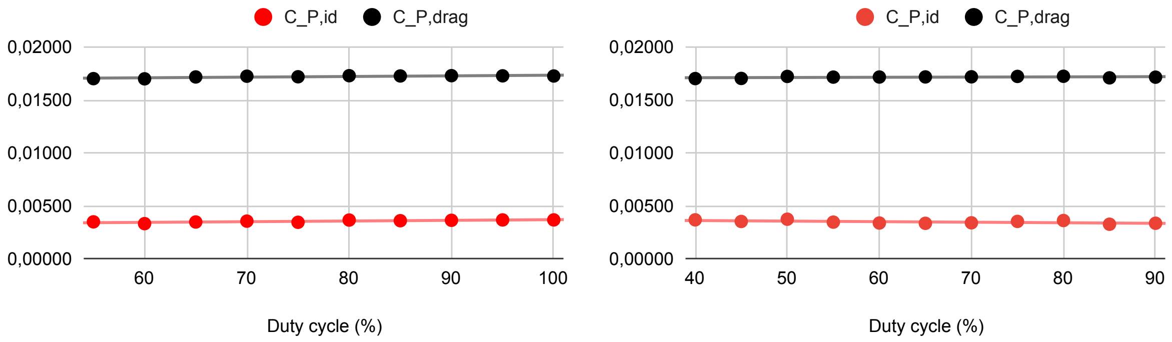
The drag coefficient C_d has not been measured, so it will be assumed that $C_d \approx 0.2$. This value has been taken from a scientific paper [52] where researchers measured C_d for a rotor having the same radius ($R = 0.127$) as the ones used in the test bench.

The Table 17 contains the updated values for the power coefficient for cases 2 and 3.

Duty cycle [%]	40	45	50	55	60	65	70	75	80	85	90	95	100
CASE 2: 11.1V $C_{P,drag}$ [-]	-	-	-	0.01702	0.01700	0.01718	0.01725	0.01720	0.01730	0.01727	0.01730	0.01728	0.01727
CASE 3: 12V $C_{P,drag}$ [-]	0.01703	0.01704	0.01723	0.01716	0.01717	0.01718	0.01719	0.01723	0.01724	0.01709	0.01716	-	-

Table 17: Updated version of the power coefficient, considering drag effect

For a better visual representation, these values are also represented (and compared to previous version) in the Figure 28.



(a) Case 2: Normal battery charge

(b) Case 3: High battery charge

Figure 28: Power coefficients considering drag effect, from experimental data, for case 2 and 3

By looking at the means, it gives $C_{P,drag} = 0.01721$ (resp. 0.01716) for case 2 (resp. 3). Those values are again quite close, which is linked to the explanation given in section 3.2.1.1. On the other hand, it can be seen that the magnitude of the power coefficients, considering the effect of drag, has increased by a factor ≈ 3 compared to the ideal case. This was expected since the drag acts as a loss, which forces the rotor to provide more power to overcome this additional force. Thus, we have the following relationship:

$$P_{id} < P_{drag} \tag{31}$$

where P_{drag} denotes the power provided by the rotor to the air to get a thrust $T_{rotor}^{(N)}$, considering drag losses.

3.2.3 Non-ideal effects

When it comes to modelling reality, non-ideal effects are part of the game. In this case, the creation of a vortex at the end of the blade is qualified as a non-ideal effect. This vortex reduces the effective lift of the blade. The concrete modeling of this vortex is quite difficult in practice and can be modeled in first approximation by a tip loss factor which can be taken into account in the expressions of C_T and C_P.

The previous expression for the power coefficient is larger in real life than the value given by Equation (28). To model the non-ideal effects, a induced power correction factor, $\kappa \approx 1.15$ can be added to Equation (28), such as:

$$C_{P,non-id} = \kappa \cdot \frac{C_T^{3/2}}{\sqrt{2}} + \frac{1}{8} \cdot \sigma_{rotor} \cdot C_d \tag{32}$$

where:

- $C_{P,non-id}$: power coefficient, considering non-ideal effects $[-]$,
- $\kappa \approx 1.15$: induced power correction factor $[-]$[22].

The model could be even closer to reality if the drag coefficient is considered to vary with the angle of attack but this approach will not be detailed here. Note that the value attributed to κ is a typical average value that will be used in further calculations.

Considering the thrust coefficient, its model could again be improved by taking into account the losses at the root of the rotor (a piece of the rotor does not provide lift, it only adds weight). This final expression will not be part of this work.

3.2.3.1 Application to experimental data

Taking $\kappa = 1.15$ for the induced power correction factor, the upgraded version of the power coefficient can be easily computed from Equation (28), whose results can be found in Table 18.

Duty cycle [%]	40	45	50	55	60	65	70	75	80	85	90	95	100
CASE 2: Normal battery voltage (11.1V) $C_{P,non-id}$ [-]	-	-	-	0.01718	0.01716	0.01737	0.01745	0.01739	0.01750	0.01748	0.01751	0.01749	0.01747
CASE 3: Normal battery voltage (12V) $C_{P,non-id}$ [-]	0.01719	0.01721	0.01743	0.01735	0.01736	0.01737	0.01738	0.01743	0.01744	0.01727	0.01734	-	-

Table 18: Updated version of the power coefficient, considering non-ideal effects

For a better visual representation, these values are also represented (and compared to previous version) in the Figure 29.

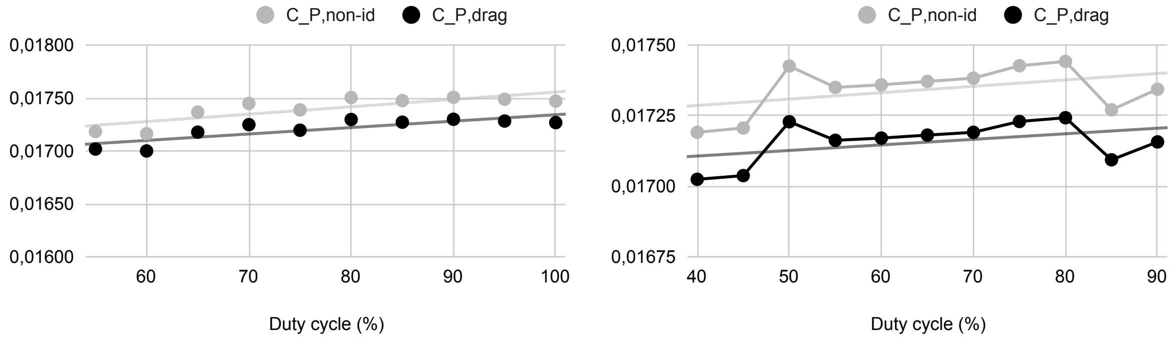
As it can be seen from Figure 29, the real power coefficients are larger since the rotor must provide more power to overcome the losses coming from non-ideal effect. Thus, we have the following relationship:

$$P_{drag} < P_{non-id} \tag{33}$$

where P_{non-id} denotes the power provided by the rotor to the air to get a thrust $T_{rotor}^{(N)}$, considering drag and non-ideal effects.

---

[22] This is a typical average value that will be used in further calculations.



(a) Case 2: Normal battery charge

(b) Case 3: High battery charge

Figure 29: Power coefficients considering drag and non-ideal effects, computed from experimental data, for cases 2 and 3

4 Theoretical model

In this section, a theoretical model of the motor-ESC interaction will be established. The goal of this model is to mathematically described the interaction between the ESC and the BLDC motor. The theoretical model will be presented as well as the system of equations composing it.

4.1 Presentation of the model

First of all, it is important to have a good visualization of the global system before entering the equations. Figure 30 represents the theoretical model [53] that will be studied in the following section.

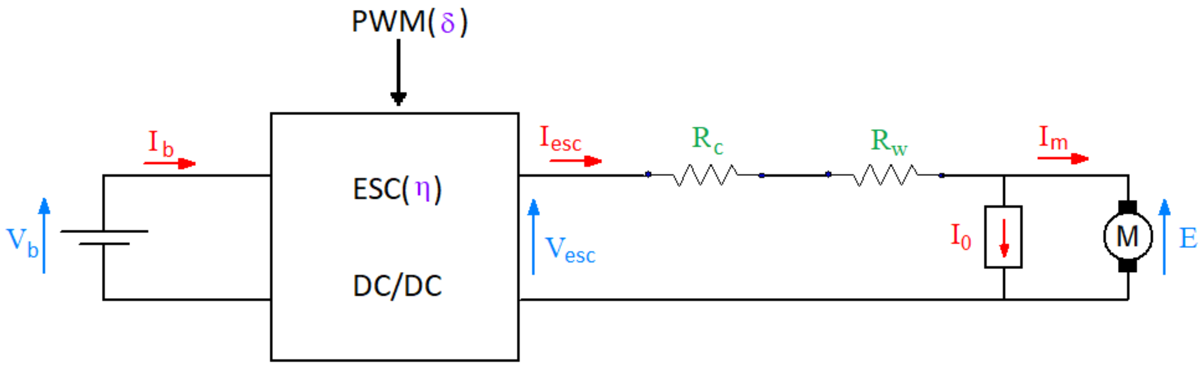


Figure 30: Theoretical model representing interaction between ESC and motor. The voltages are represented in blue, the currents in red, the resistances in green and the ESC parameters in purple.[23]

The different elements composing the model pictured in Figure 30 will be detailed in the following sections.

4.1.1 Battery

The battery is the power supply of the model. Its goal is to provide power to the next block (i.e. the ESC). The battery is modelled by a voltage source V_b and a current I_b.

4.1.2 ESC

The ESC acts like a DC/DC converter: its role is to transmit a fraction of the input voltage, coming from the battery, V_b to its output V_{esc}, with a certain efficiency η_{esc}. This fraction will be determined by the PWM signal (see section 4.1.3). I_{esc} denotes the current at the output terminals of the ESC.

4.1.3 PWM

The PWM signal determines which proportion of the battery voltage V_b will be transmitted to the ESC output V_{esc}. This ratio is linked to the duty cycle δ of the PWM command.

4.1.4 Motor

The motor is modelled by two resistors: R_c represents the connections resistance, while R_w represents the winding resistance. There is the existence of an idle current (or no-load current), I_0, which produces the magnetic field in the motor when there is no mechanical load. The current passing through the motor is represented by I_m, while E is the voltage at the motor's terminals

4.2 Equation system

Now that the different elements composing the model have been explained, the equation system will be detailed:

The voltage V_{esc} at the output terminals of the ESC can be computed from the battery voltage V_b and the duty cycle δ of the PWM command, thanks to the power balance of the ESC:

$$V_{esc} = \delta \cdot V_b \tag{34}$$

The ESC outgoing current I_{esc} can be derived from Kirchhoff's voltage law (on the left part of the model) and the relation established at Equation (34):

$$\begin{aligned}
& V_{esc} \cdot I_{esc} = \eta_{esc} \cdot (V_b \cdot I_b) \\
\iff \quad & I_{esc} = \frac{\eta_{esc} \cdot V_b \cdot I_b}{V_{esc}} \\
\iff \quad & I_{esc} = \frac{\eta_{esc} \cdot V_b \cdot I_b}{\delta \cdot V_b} \\
\iff \quad & I_{esc} = \frac{\eta_{esc} \cdot P_b}{\delta \cdot V_b} = \frac{\eta_{esc} \cdot I_b}{\delta} \qquad (35)
\end{aligned}$$

where P_b denotes the battery power. Note that the ESC efficiency η_{esc} comes into play at this stage.

The motor current I_m can be expressed, using Kirchhoff's current law:

$$I_m = I_{esc} - I_0 \tag{36}$$

The motor voltage E is expressed thanks to Kirchhoff's voltage law (on the right part of the model):

$$\begin{aligned}
& E + (R_c + R_w) \cdot I_{esc} = V_{esc} \\
\iff \quad & E = V_{esc} - (R_c + R_w) \cdot I_{esc} \\
\iff \quad & E = \delta \cdot V_b - (R_c + R_w) \cdot I_{esc} = \delta \cdot V_b - R_m \cdot I_{esc} \qquad (37)
\end{aligned}$$

where R_m denotes the motor resistances (i.e. $R_m = R_c + R_w$).

The system of equations is therefore composed as follows:

$$\begin{cases}
I_{esc} = \dfrac{\eta_{esc} \cdot P_b}{\delta \cdot V_b} \\
I_m = I_{esc} - I_0 \\
E = \delta \cdot V_b - R_m \cdot I_{esc}
\end{cases} \tag{38}$$

This system of equations (38) will be used to express the power of the motor P_m, according to the parameters of the model, via a power balance:

$$\begin{aligned}
& P_m = E \cdot I_m \\
\Longleftrightarrow \quad & P_m = E \cdot (I_{esc} - I_0) \\
\Longleftrightarrow \quad & P_m = (\delta \cdot V_b - R_m \cdot I_{esc}) \cdot (I_{esc} - I_0) \\
\Longleftrightarrow \quad & P_m = \delta \cdot V_b \cdot I_{esc} - \delta \cdot V_b \cdot I_0 - R_m \cdot I_{esc}^2 + R_m \cdot I_{esc} \cdot I_0 \\
\Longleftrightarrow \quad & P_m = \delta \cdot V_b \cdot \frac{\eta_{esc} \cdot P_b}{\delta \cdot V_b} - I_0 \cdot (\delta \cdot V_b - R_m \cdot I_{esc}) - R_m \cdot I_{esc}^2 \\
\Longleftrightarrow \quad & P_m = \eta_{esc} \cdot P_{elec} - I_0 \cdot E - R_m \cdot \left(\frac{\eta_{esc} \cdot P_{elec}}{\delta \cdot V_b} \right)^2 \qquad (39)
\end{aligned}$$

where P_{elec} denotes the electrical power (i.e. $P_{elec} = P_b$).

The Equation (39) shows that the power P_m developed by the motor is proportional to the initial electrical power P_{elec} minus the losses coming both from the motor resistances and the no-load current. Note that the motor speed ω_{motor} can be obtained from the voltage E at the terminals of the motor thanks to the motor velocity constant K_v:

$$\omega_{motor} = E \cdot K_v \qquad (40)$$

Note also that the torque τ can be obtained from the motor current I_m, using the motor torque constant K_t:

$$\tau = I_m \cdot K_t = \frac{I_m}{K_v} \qquad (41)$$

5 Datasheets

In this section, the motor parameters will be extracted from the data sheets in order to compare them with the theoretical model and the experimental measurements later on.

From the motor's data sheet (see Appendix A.3.1, the following information can be found:

- The velocity constant: $K_v = 1000$ $[\frac{\text{RPM}}{\text{V}}]$,
- The no-load current: $I_0 = 0.5$ [A],
- The motor resistance (winding): $R_m = 0.090$ $[\Omega]$,
- The number of cells allowed for the LiPo battery pack: $N_{cells} = 2 - 3$ $[-]$,
- The efficiency of the motor: $\eta_{motor} = 0.8$ $[-]$.

Regarding the efficiency η_{motor}, there is no clear indication in the data sheet of what this efficiency represents or how it is calculated. It can be assumed that it is the efficiency of transforming the electrical energy supplied to the motor into mechanical energy to make it turn. Therefore, the losses would be about 20%.

6 Comparison

In this section, the data acquired during the experimental step, in section 3.1.3 will be compared with the data provided by the manufacturer (via data sheets, see section 5) as well as the values that can be extracting by the means of the theoretical model established in section 4. The ultimate goal is to establish a first design of the drone, on the basis of theoretical formulas, possibly slightly modified, according to the confidence that we will have acquired towards the data of the supplier.

6.1 Motor no-load current I_0

The no-load current is easy to measure. Indeed, it requires to simply run the test bench with the rotor removed from the top of the motor. Then, the duty cycle is gradually increased until the motor has enough power to start turning freely. Once this stage is reached, the no-load current I_0 corresponds to the value of the current consumed. The no-load current has not been measured. It will be assumed that $I_0 = 0.5A$ in the following.

6.2 Motor velocity constant K_v

The motor velocity constant can be computed from the experimental data:

$$K_v = \frac{\omega_{motor,no-load}}{\delta \cdot V_b} \tag{42}$$

where $\omega_{motor,no-load}$ represents the speed of the no-loaded motor (i.e. without a rotor on top of it), expressed in RPM.

The results obtained by applying Equation (42) on the experimental measures (see Tables 10 and 11) are displayed in Table 19.

Duty cycle [%]	40	45	50	55	60	65	70	75	80	85	90	95	100
CASE 2: Normal battery voltage (11.1V) $K_v[-]$	-	-	-	622.44	609.61	595.98	585.59	580.78	563.40	558.56	547.55	537.98	526.13
CASE 3: Normal battery voltage (12V) $K_v[*]$	717.71	701.30	666.67	665.91	655.69	643.59	630.95	613.67	597.92	600.98	574.07	-	-

Table 19: Motor velocity constant, K_v, computed from experimental data

As it can be seen from the values contained in Table 19, the velocity constant is not constant across the different measurements. The reason behind that phenomenon can be explained: the rotational speed of the motor, $\omega_{motor,no-load}$ has been measured while the motor was loaded, as shown in Figure 21. The effect of the loading on $\omega_{motor,no-load}$ is that it will decrease the rotational speed $\omega_{motor,no-load}$ due to drag losses occurring. The loading generates a torque that represents an additional force that must be overcome before making the motor spin. The bigger the rotational speed, the larger the losses.

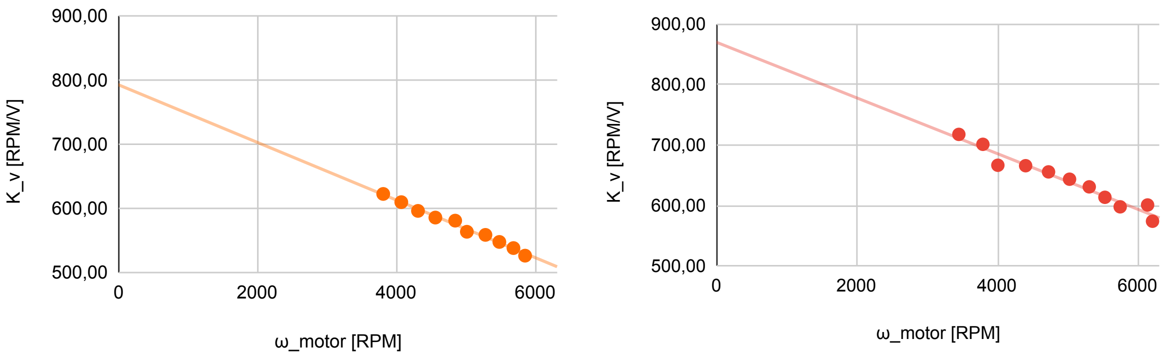
Hopefully, the real value $\omega_{motor,no-load}$ can be estimated. By looking at Equation (42), K_v is defined as a constant and the applied voltage, $\delta \cdot V_b$, is increased linearly during the experiments. Therefore, it can be deduced that $\omega_{motor,no-load}$ evolve linearly with respect to $\delta \cdot V_b$ in order to keep K_v constant. Therefore, the losses occurring due to loading are also linear as a first approximation[24]. Thus, losses → 0 as $\omega_{motor} \to 0$. The Figure 31 displays the evolution of K_v in function of ω_{motor} for the two experimental cases studied in this work.

6.2.1 Case 2: normal battery level (11.1V)

By analyzing the Figure 31, for case 2, we can see that the values of speeds measured experimentally follow indeed a linear behavior. An interpolation gives a value $K_v \to 800$, as $\omega_{rotor} \to 0$. Note that this point is not achievable in practice because the voltage and velocity are zero. However, this is the point for which losses have the least impact on the measure since they evolve linearly with velocity, as explained above. This behavior can be seen on the graph: the higher the RPM speed, the lower the K_v value. Therefore, $\omega_{motor,no-load} \approx 800$ as a first approximation.

---

[24] In practice, drag increases much faster than the velocity: it is proportional to the square of the velocity but it is not the only loss to consider[54].



(a) Case 2: Normal battery charge

(b) Case 3: High battery charge

Figure 31: Evolution of K_v in function of ω_{motor}, from experimental data, for case 2 (left) and 3 (right)

By looking at the values written in the data sheet (see section 5), the supplier indicates that $K_v = 1000$. However, there is also an indication about efficiency $\eta_{motor} = 0.8$. If it is assumed that this efficiency η_{motor} represents the amount of P_{elec} converted into P_{mech}. Therefore, Equation 42 is rewritten in the following form:

$$K_v^* = \eta_{motor} \cdot K_v = \frac{\omega_{motor,no-load}}{\delta \cdot V_b} \tag{43}$$

where K_v^* denotes the K_v value accounting for the motor efficiency. In that case, $K_v^* = 0.8 \cdot 1000 = 800$, which corresponds to the K_v value estimated from the experimental measures.

6.2.2 Case 3: high battery level (12V)

The same description and analyse could be done for the case 3. In that case, $K_v \to 875$, as $\omega_{rotor} \to 0$ which differs from case 2. However, K_v is a constant by definition and such a difference should not be observed in practice. A potential explanation could be illustrated in the Figure 32.

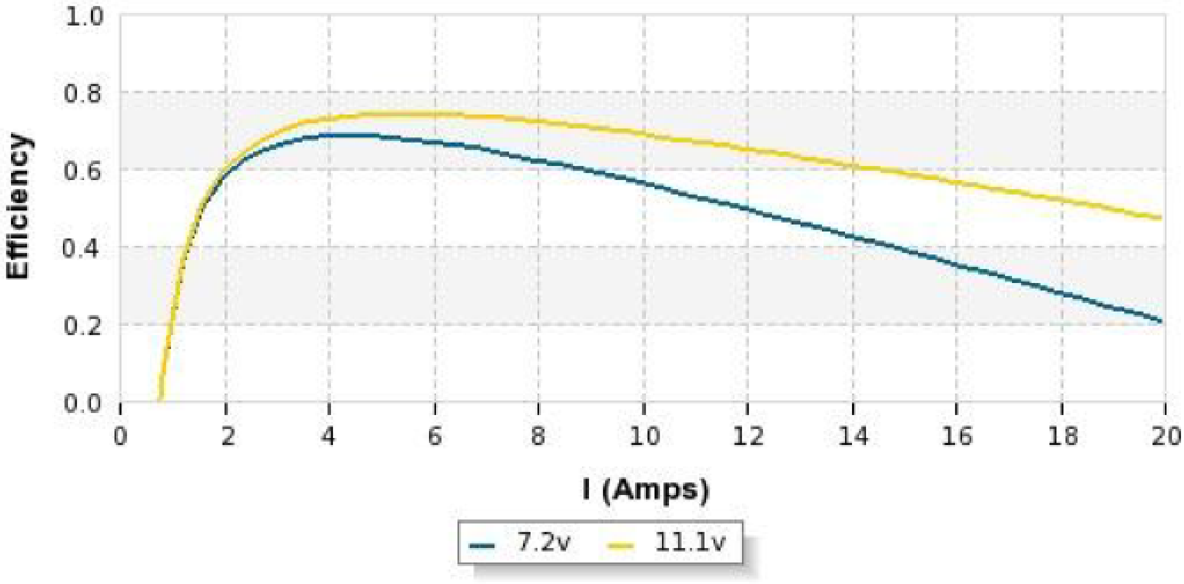


Figure 32: Evolution of the efficiency of the motor, in function of the voltage

This Figure 32 is taken from the original data sheet (see Appendix A.3.1). By looking at it, it can be inferred that the efficiency curve depends on the input voltage. In particular, the bigger the input voltage is, the better is the efficiency. Therefore, since the input voltage is 12V in case 3, it can be expected that:

$$\eta_{motor}(@12\text{V}) > \eta_{motor}(@11.1\text{V}) \tag{44}$$

From Equation (44), it follows that:

$$K_v^*(@12\text{V}) = \eta_{motor}(@12\text{V}) \cdot K_v > K_v^*(@11.1\text{V}) = \eta_{motor}(@11.1\text{V}) \tag{45}$$

This result could be a first element to explain the difference between the two calculated K_v values. Moreover, the experimental data obtained for case 3 seems to contain more noise than those from case 2. Indeed, some points in the Figure 31 deviate slightly from the general trend line. These differences could be linked to measurement errors (user and/or measuring device). For the following, it can be concluded that the experimental results validate the K_v value given by the supplier.

6.3 Motor resistance R_m

The motor resistance can be computed from experimental data.
The first step consists in computing the voltage E at the terminals of the motor, using Equation (40). Then, injecting the values obtained for E in Equation (38) yields to the motor resistance R_m.

Since the voltage E at the terminals of the motor has not been measured, an approximation will again be proposed. In the data sheet, a value of efficiency, $\eta_{motor} = 0.8$ is given. This efficiency certainly relates the ability of the motor to transform the input electrical power, P_{elec}, into output mechanical power P_{mech}. Using the same notations as the ones given in section 4, the efficiency, η_{motor}, can be expressed as follows:

$$\begin{aligned} & P_{mech} = \eta_{motor} \cdot P_{elec} \\ \iff \quad & E \cdot I_m = \eta_{motor} \cdot V_b \cdot I_b \\ \iff \quad & E = \eta_{motor} \cdot \frac{V_b \cdot I_b}{I_m} \end{aligned} \tag{46}$$

Then, injecting Equations (35) and (36) in (46):

$$\begin{aligned} & E = \eta_{motor} \cdot \frac{V_b \cdot I_b}{I_{esc} - I_0} \\ \iff \quad & E = \eta_{motor} \cdot \frac{V_b \cdot I_b}{\frac{\eta_{esc} \cdot I_b}{\delta} - I_0} \end{aligned} \tag{47}$$

The Equation (47) allows to compute the voltage E from the battery and ESC parameters as well as the no-load current,I_0, previously determined (see section 6.1). In the following, the value $I_0 = 0.5$ will be used[25]. The values obtained for the voltage E using the Equation (47) are displayed in Table 20.

Duty cycle	40	45	50	55	60	65	70	75	80	85	90	95	100
CASE 2: Normal battery voltage (11.1V)	-	-	-	5.61	6.07	6.48	6.90	7.34	7.77	8.22	8.66	9.11	9.57
CASE 3: High battery voltage (12V)	4.41	4.89	5.37	5.85	6.32	6.80	7.28	7.76	8.24	8.74	9.25	-	-

Table 20: Voltage E at motor's terminals, computed from experimental data

Finally, applying Kirchhoff's voltage law on the right part of the theoretical model displayed in Figure 30, R_m can be computed:

$$\begin{aligned} & V_{esc} = I_{esc} \cdot R_m + E \\ \iff \quad & R_m = \frac{V_{esc} - E}{I_{esc}} \end{aligned} \tag{48}$$

---

[25] Note that Io which usually appears in data sheets as a constant is not a constant, its value depends in particular on the rotation speed of the motor, as indicated in this article [55].

The values obtained for R_m are illustrated in Table 21.

Duty cycle	40	45	50	55	60	65	70	75	80	85	90	95	100
CASE 2: Normal battery voltage (11.1V)	-	-	-	0.083	0.095	0.108	0.115	0.121	0.125	0.129	0.132	0.135	0.139
CASE 3: High battery voltage (12V)	0.066	0.079	0.090	0.098	0.103	0.108	0.110	0.112	0.114	0.119	0.125	-	-

Table 21: Resistance of the motor R_m, computed from experimental data.

First of all, by looking at Table 21, we can see that the resistance R_m increases in amplitude as a function of the voltage. In reality, as explained in this article [55], the internal resistance of the motor is not constant but varies with the temperature of the winding. The winding temperature is directly related to the current flowing in the motor. The variations in the value of the motor resistance that are visible in the Table 21 are therefore related to the increase of the current as the duty cycle increases. To get an idea of the value of the resistance under all circumstances, the average value, for case 2 (resp. 3), is: $R_m = 0.118$ (resp. 0.102).

Moreover, the values obtained corresponds between the two cases studied, while considering the same applied voltage (i.e. $\delta \cdot V_b$).For instance, considering case 3, at 55% duty cycle, the applied voltage is equals to: $0.55 \cdot 12 = 6.6V$ while, for case 2, at 60% duty, the applied voltage is equals to: $0.6 \cdot 11.1 = 6.66V$. These applied voltage are quite close, which is also true for the value of the calculated resistance R_m (0.103 vs 0.108 Ω). The small differences between the corresponding values are probably due to the accuracy of the speed measurements (rounding + measurement error).

The value announced by the supplier for the motor resistance is $R_m = 0.090\Omega$. By looking at Table 21, this value $R_m \approx 0.090\Omega$ can be found for a duty cycle [50%-55%]. As the value of this resistance varies with the temperature, as indicated above, the reference value written in the data sheet must certainly have been taken for a motor operating in this duty cycle range, but this is only a guess at this stage. To confirm this hypothesis, it would be necessary to contact the supplier and ask him about the conditions under which the performance tests were carried out.

6.4 Conclusion

To conclude this section, it can be said that the values provided by the data sheets are not totally off the mark. However, the conditions under which these parameters were measured are never indicated. Therefore, it is much better to make some experimental measurements in order to validate the motor characteristics in person.

Part IV

Learning kit

This part is focused on the experience gained on the learning kit. Indeed, it is still too early to jump headlong into the construction of a drone. At this stage, only a motor can be controlled with an ESC thanks to the test bench detailed in section 2. However, at the moment, no experience with a real flight controller is available. This part will therefore apply all the theory of Section 2 on a drone built from a learning kit. It is really a "learning-by-doing" part. First of all, details will be given about the kit. Then, its construction will be illustrated. After that, the description of the software used to parameterise the drone will be made. After that, the remote coding will be discussed to conclude on the control system and the evaluation of the flight performances.

1 Kit description

In this section, the learning kit, which will be called simply "kit" hereafter, will be detailed. The selection criteria and the different elements of the kit will be discussed.

1.1 Selection criteria of the kit

As said in the introduction, the idea was to get familiar with a drone before getting into the practical realization. From then on, some selection criteria were developed in order to work with the most adequate kit to validate these objectives:

- The first selection criterion was to choose a kit to assemble by hand, i.e. "DIY" (Do it yourself) instead of a RTF ("Ready to fly"),
- The second selection criterion concerns the FC: it was desired to find a FC that was easy to handle on the software level. Indeed, this would allow to reuse the FC during the construction of the hexadrone and to apply what will have been learned on the kit in order to lose as little time as possible and to increase the chances that the hexadrone ends up flying.
- The third and final criterion, less important than the first two, was the price and delivery time. Indeed, it was necessary to be able to make the hands on the kit as soon as possible in order not to delay the construction of the hexacopter in a second time!

1.2 QAV 250 kit

Based on the criteria developed in Section 1.1, the winning kit of the selection is the QAV 250 kit, developed by Holybro [56]. The vendor's webpage can be found in Appendix A.3.10. The Figure 33 shows the different parts of the kit and the drone once assembled.

As the kit contains a lot of elements, the next sections will describe them while referencing the data sheets when relevant. Be aware that the numbering used on Figure 33 will be used to refer to the components.

1.2.1 Kit frame

The frame of the kit is composed of three plates and four arms labeled as "1" and "2" respectively in Figure 33. Note that the rightmost plate will not be used for the assembly because it is a general kit that contains bulk parts used to build other types of drones, with other motors for instance. The arms contain a slot at the end to attach the motor with screws. The arms will be attached to the top and bottom plates with screws as well. The material used to make the frame is carbon fiber: the drone will be rigid and resistant to shocks (which is good for a beginner).

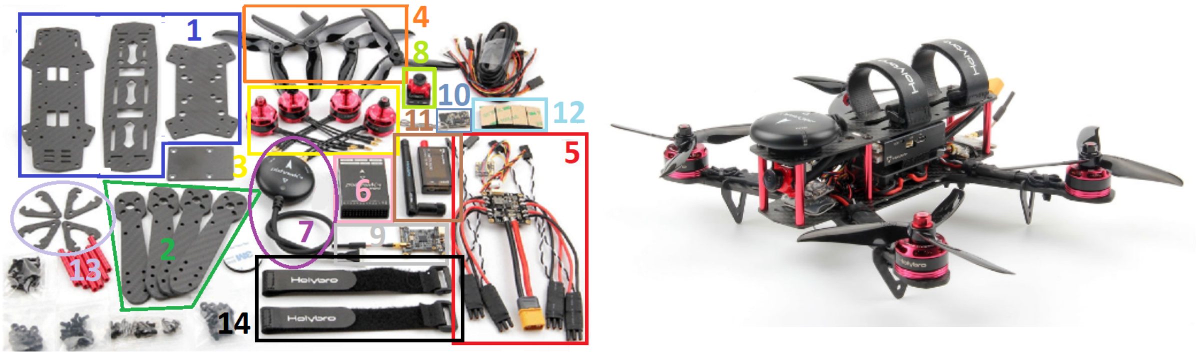


Figure 33: The drone once assembled as well as the components it contains, as displayed on the seller's website.

Sources: https://www.studiosport.fr/complete-kit-diy-qav250-avec-pixhawk-4-mini-holybro-a20876.html

1.2.2 Kit motors

The motors of the kit are labeled as "3" in Figure 33. They are referenced as "motor DR2205 KV2300"by the reseller. However, there is not much information about them, the only data sheet found so far, Appendix A.3.11, contains some experimental values that do not correspond to the kit specs. It will therefore be necessary to go through an experimental phase to characterize these motors (see Section 6.1).

1.2.3 Kit rotors

The rotors of the kit are labeled as "4" in Figure 33. The rotors are made of plastic and have a 5-inch diameter.

1.2.4 Kit PDB and ESCs

The power distribution board and the ESCs of the kit are labeled as "5" in Figure 33. The design of the power board is very simplified: its purpose is only to transmit the power from the battery to the ESCs. It does not contain any additional features as explained in the data sheet (see Appendix A.3.12). This PDB can accept battery voltage from 2S to 10S and a maximum current of 30A. Concerning the ESCs (see Appendix A.3.13), they can support a current of 20A max and run the BLHeli S firmware, but they do not have an embedded voltage regulator. By combining the limitations on the current of the 2 components, the limit is 20A maximum. Note that they communicate with the FC by the means of PWM signal. Note that the ESCs and the XT60 connector of the future battery arrive already soldered with the PDB, which saves time during assembly.

1.2.5 Kit FC

The flight controller of the kit is labeled as "6" in Figure 33. The FC in question here is the Pixhawk 4 mini (PX4 mini in short), whose user guide is given in Appendix A.3.14. The Pixhawk 4 mini is powered by 5V through the PDB. It has 8 PWM output channels to send its control signals (up to 8 motors can be controlled simultaneously). The pixhawk 4 mini has been optimized to run PX4 flight control software. Note that PX4 is open-source computer code: it is therefore easy to integrate your own model within existing models, as it will be detailed in section 4.1. It also has a USB connection so that it can be connected to a computer and configure the parameters of the drone using the recommended software, namely QGroundControl, QGC in short.

1.2.6 Kit GPS

The GPS, associated to the flight controller of the kit, is labeled as "7" in Figure 33. The GPS, whose data sheets can be found in Appendix A.3.15, must also be powered in 5V, like the PX4 mini. It is equipped with a LED to visually represent the state of the FC and a safety switch (see Definition 9 [57]) to add a hardware security to the drone. Note that the GPS must be correctly aligned with the FC.

Definition 9. Safety switch

Safety switches can be used to immediately stop motors or return the vehicle in the event of a problem.

1.2.7 Kit FPV camera, video transmitter and OSD

The FPV camera, the video transmitter and the OSD of the kit are labeled as "8", "9" and "10" respectively in Figure 33. The purpose of the FPV camera is to send a continuous video signal during the flight through the video antenna to the pilot. Thus, the pilot sees "through the eyes" of the drone. Concerning connections, the camera is connected to an OSD (on-screen display) which allows to make the link between the battery (to power the camera), the radio antenna (to send the outgoing video signal) and the PX4 mini (to detect the commands sent by the FC like starting the video). Note that the OSD allows to add to the video stream real time information sent by the FC. Data sheets relative to the FPV camera, the video transmitter and the OSD are available in Appendices A.3.16, A.4 and A.5 respectively.

1.2.8 Kit telemetry

The telemetry radio of the kit are labeled as "11" in Figure 33. Note that this option was not available at the time of purchase. The purpose of the telemetry kit [58] is to allow communication between the drone in flight and an external device (i.e. a computer). The telemetry kit therefore establishes a wireless communication to send information from the FC, GPS or PDB. This allows the pilot to have a direct feedback from the drone and to adjust his piloting accordingly. As the learning kit was devoid of this component, it will not be discussed more than this.

1.2.9 Kit mounting foam

The mounting foams of the kit are labeled as "12" in Figure 33. Since the FC is composed of an accelerometer and a gyroscope, it is necessary to prevent it from external vibrations. Indeed, vibrations could introduce errors in the measurements taken by the FC and lead to a possible problem. It is precisely the mounting foam that isolates the FC from external medium and high frequency vibrations [59].

1.2.10 Kit legs

The drone's legs of the kit are labeled as "13" in Figure 33. They are made of carbon fiber and allow the drone to stand straight at rest.

1.2.11 Kit battery strap

The battery's straps of the kit are labeled as "14" in Figure 33. The role of these straps is to fix the battery on the drone. This one must be very well attached because if it came to be unhooked in full flight, it would be the drama.

1.2.12 Kit battery added

The kit did not include a battery. Based on the current and voltage limitations of the components described above, a 4S battery, shown in Figure 34, was selected to complete this kit.

The main characteristics of the battery's data sheet (see Appendix A.7) are as follows:

- Number of cells: 4 (i.e. 14.8V),
- Capacity: 1550 mAh,
- Weight: 176g,
- Length: 7.6cm,
- Width: 3.8cm,
- Height: 3.3cm.

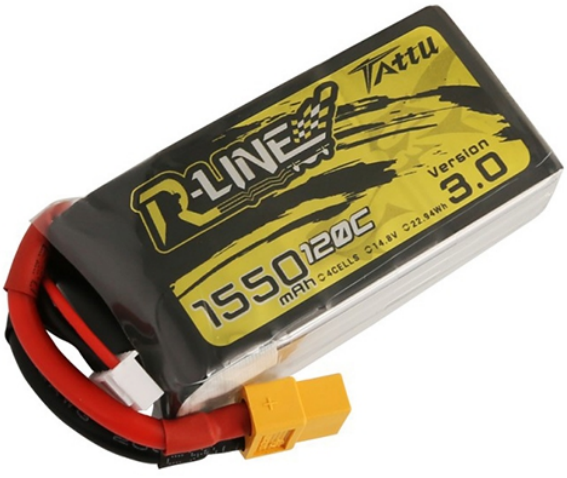


Figure 34: A 4S battery of capacity 1550mAh, completing the learning kit.
Sources: https://www.studiosport.fr/batterie-lipo-4s-1550-mah-120c-r-line-v3-tattu-a17666.html

1.2.13 Remote control and receiver

The kit also did not include a remote control and receiver. Therefore, it was decided to use the Tanaris x9 remote control and the FrSky X8R receiver that were provided by the promoter. They are both illustrated in Figure 35.

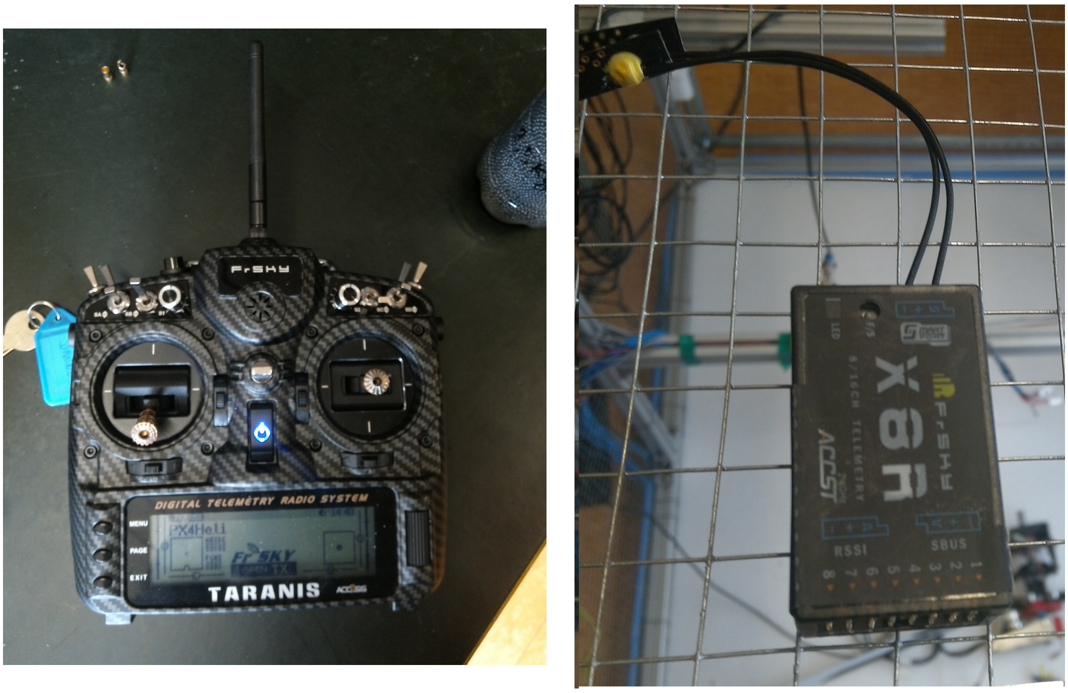


Figure 35: The Tanaris x9 remote control on the left, the FrSky X8R receiver on the right, used in combination of the kit.

Concerning the receiver, it is equipped with a simple 3-wire connector (positive voltage, negative voltage and control) that allows it, using a digital protocol, to communicate with multiple channels simultaneously (this model supports up to 16 channels). Therefore, since the receiver uses the S-BUS protocol, it must be connected to the "RC_IN" input of the PX4 mini. Its data sheet can be found in Appendix A.8. It communicates over the 2.4 GHz frequency band with the remote control.

Regarding the remote control, it is a very good quality programmable remote control. It has a multitude of switches and buttons in order to communicate using up to 24 channels simultaneously, as indicated in its data sheet (see Appendix A.9). The programming of the remote control will be addressed instead to the section 4.

2 Kit construction

In this section, the different steps encountered during the construction of the kit will be illustrated. The construction of the whole drone was largely based on the document provided at the time of purchase (whose link is available in Appendix). Some steps being not very clear or outdated, this website was also consulted [60].

The first step, after unpacking the kit and laying out all components on a table, is to assemble the frame consisting of 2 plates and 4 arms. Note that the top plate will be added once the interior of the drone is filled.

The second step is to install the 4 motors in the places provided on the motors. Then, the PDB can be fixed in the center of the frame and the ESCs can be connected to the motors. Note that there is no color code to connect the ESCs to the motors. After testing, it is enough to simply reverse 2 connections in order to make the motor turn in the opposite direction.

The third step consists in fixing the PX4 mini in the middle of the drone, on the small plastic plate covering the PDB. The plate is made of plastic in order to electrically isolate the PDB. Note that you also need to glue the mounting foam between the plastic plate and the PX4 mini to isolate it from mechanical vibrations. Note that the ESCs have been fixed to the frame with Colson collars to prevent them from wandering once the drone is in the air.

The last step is to install the top plate, which was left out in step 1. The frame being finished, the GPS and the battery can now be placed on the top of the drone. Once the rotors are installed on the motors, the assembly will be finished: the quadcopter is ready to be configured using QGroundControl, which will be explained in the next section.

The Figure 36 illustrates these 4 assembly steps.

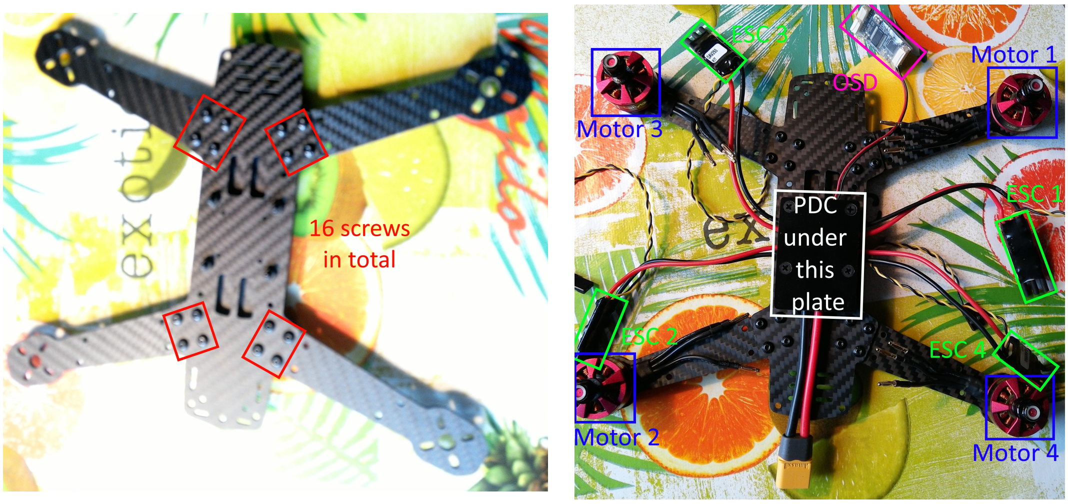
3 QGroundControl

This section will discuss the QGroundControl software (QGC in short). This software allows you to configure the parameters of the PX4 mini quite easily, using a very visual and interactive user interface. First of all, the interaction between the PX4 mini and QGC will be discussed. Then, the different tabs of QGC will be detailed one by one, applying in parallel on the learning kit. Finally, some parameters allowing to set up the drone differently and to unlock some functions will be described. Of course, it is impossible to talk about all the parameters that the FC has. Only the parameters that have been found interesting during the year will be presented. The reader is invited to consult this web page if he wants to have a complete list of the parameters with their function [61].

3.1 Interaction PX4 mini-QGC

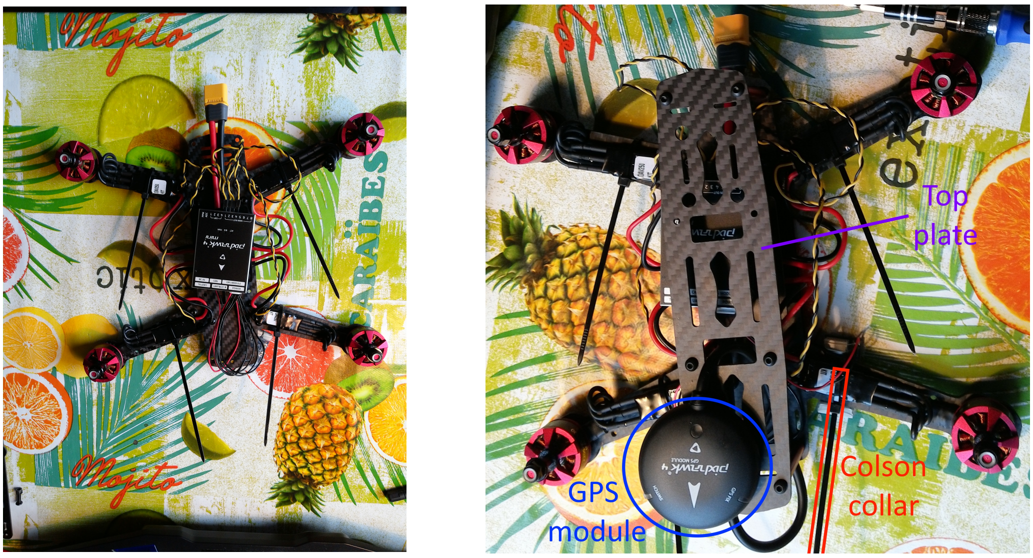
QGC allows to install the PX4 firmware on Pixhawk flight controllers. In fact, the PX4 mini has a 32GB SD card to store FC configuration files as well as logs obtained during flights, for example. Since it is an open source code, anyone can modify the code, add features and then flash this new version of the firmware in the PX4 mini. This approach will be used in the Section 4.1.

In order to flash the firmware on the SD card of the PX4 mini, it is necessary to connect the PX4 mini to a computer with a USB cable for data exchange. It is also possible to use a telemetry kit to perform this exchange, each side must have a transceiver at its disposal. The big advantage of the telemetry



(a) Step 1: fixing together the parts composing the frame.

(b) Step 2: fixing motors, PDB and ESCs on the frame.



(c) Step 3: adding PX4 mini on bottom plate.

(d) Step 4: adding GPS on top plate.

Figure 36: The different steps of the kit assembly.

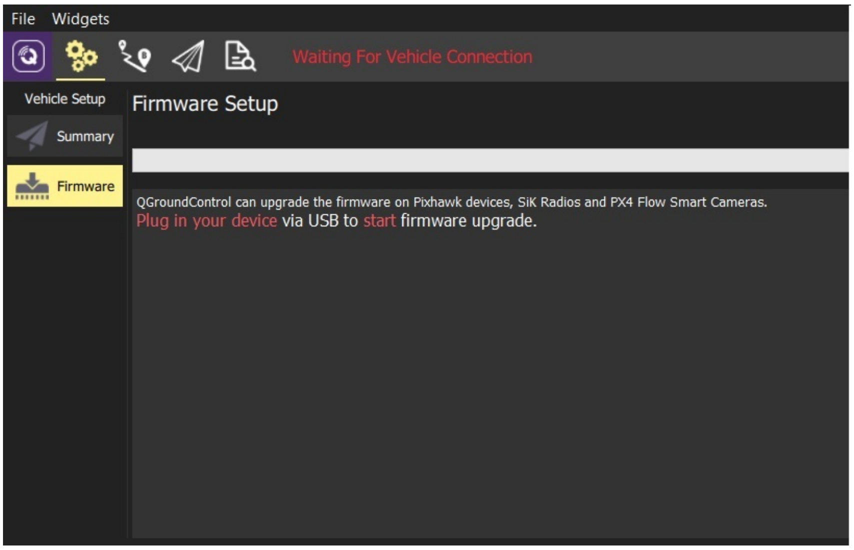
kit over the USB cable is that it establishes a wireless connection between the PX4 mini and QGC. So it is possible to fly the drone while receiving real-time data. It is, of course, possible to receive real-time data using the USB cable but the drone will not be able to take off very high, limited by the length of the cable. In this work, the use of a telemetry kit was not considered. A future work could study the possibility to collect and use the data in real time in order to improve the performances of the drone for instance.

3.2 Software configuration using QGC

In this section, the different tabs used to configure the PX4 mini will be described. During the entire configuration, the PX4 mini must be connected to the computer using the USB cable. Note that the USB cable allows the exchange of data between QGC and the PX4 mini, while powering the latter. All information is taken from the official QGC documentation [62].

3.2.1 Firmware

This tab allows you to update the FC firmware or to inject your own version, using a simple button as illustrated in Figure 37.



QGroundControl can upgrade the firmware on Pixhawk devices, SiK Radios and PX4 Flow Smart Cameras.
Plug in your device via USB to start firmware upgrade.

Figure 37: QGC firmware tab.
Sources: https://docs.qgroundcontrol.com/master/en/SetupView/Firmware.html

3.2.2 Airframe

Once the firmware is updated, it is necessary to select the airframe that corresponds to our drone, as illustrated in Figure 38. The selection is made on the type of drone (shape and number of motors) and the model. The drone type allows to determine for instance how many PWM outputs should be activated, based on the number of motors, while the model allows to load the control system parameters with the tuned values provided by the vendor. It is also possible to select generic airframes if the model is not present in the list, but it is rather recommended to create your own airframe to be sure of your shot (see Section 4.1.2).

In our case, it is a quadrotor x, model QAV250[26]. This airframe defines PWM outputs 1 to 4 as main (i.e. main motor control). PWM outputs 5 to 6 are defined as auxiliary (i.e. do not allow control of the main motors as opposed to servo-motors for example) while outputs 7 to 8 are disabled. Note that all parameters will be reset once the airframe is changed.

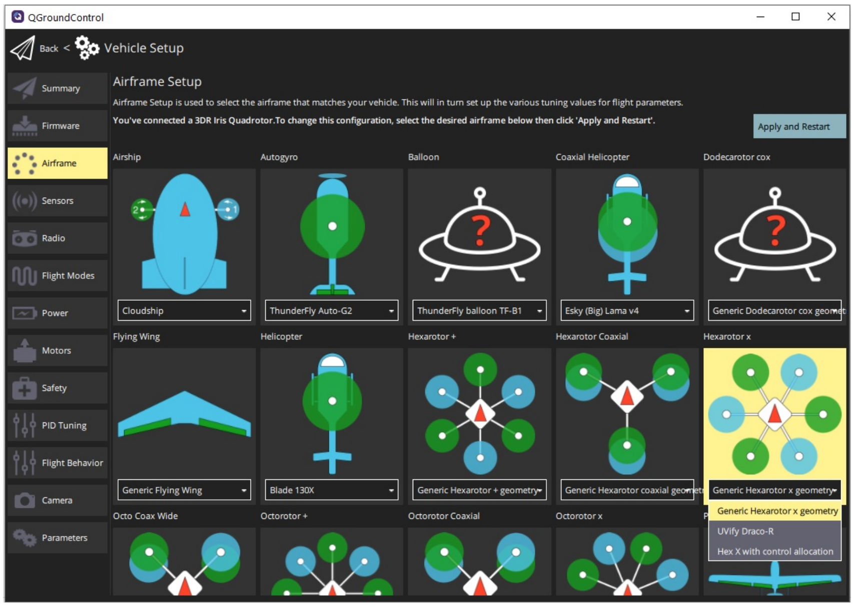
3.2.3 Radio

Once the airframe is selected, the radio must be set up, as illustrated in Figure 39.

First, the remote control must be calibrated. The calibration allows the PX4 mini to make a model of the remote control by taking note of the different inputs and their range. To do this, it is necessary to move the joysticks, as well as any switches/buttons that could be used, from their min to max position, in a precise order. In that way, it will encode the physical positions of the remote hardware.
Secondly, once the PX4 mini is aware of the different inputs, it will automatically attribute channels to main actions of the drone (i.e. throttle, yaw, pitch and roll), depending on the remote mode selected as illustrated in Figure 40. In our case, the remote will be used in mode 2, which is the most used mode in practice.

---

[26] The complete list of all airframes is available on the following web page: https://docs.px4.io/v1.12/en/airframes/airframe_reference.html

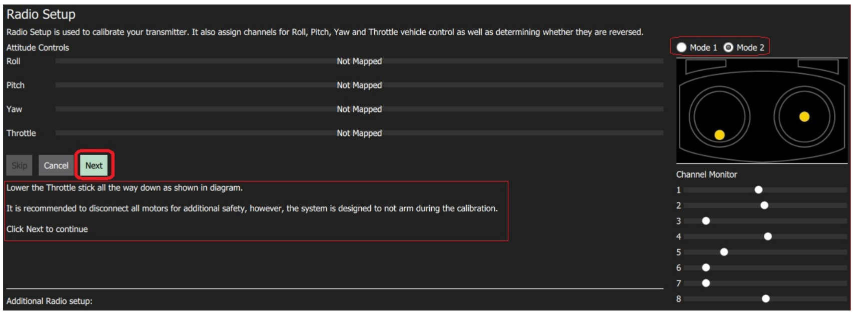


Airframe Setup

Airframe Setup is used to select the airframe that matches your vehicle. This will in turn set up the various tuning values for flight parameters.

You've connected a 3DR Iris Quadrotor.To change this configuration, select the desired airframe below then click 'Apply and Restart'.

Figure 38: QGC airframe tab.
Sources: https://docs.qgroundcontrol.com/master/en/SetupView/airframe_px4.html



Radio Setup

Radio Setup is used to calibrate your transmitter. It also assign channels for Roll, Pitch, Yaw and Throttle vehicle control as well as determining whether they are reversed.

Attitude Controls

Roll Not Mapped

Pitch Not Mapped

Yaw Not Mapped

Throttle Not Mapped

Lower the Throttle stick all the way down as shown in diagram.

It is recommended to disconnect all motors for additional safety, however, the system is designed to not arm during the calibration.

Click Next to continue

Additional Radio setup:

Figure 39: QGC radio tab.
Sources: https://docs.qgroundcontrol.com/master/en/SetupView/Radio.html

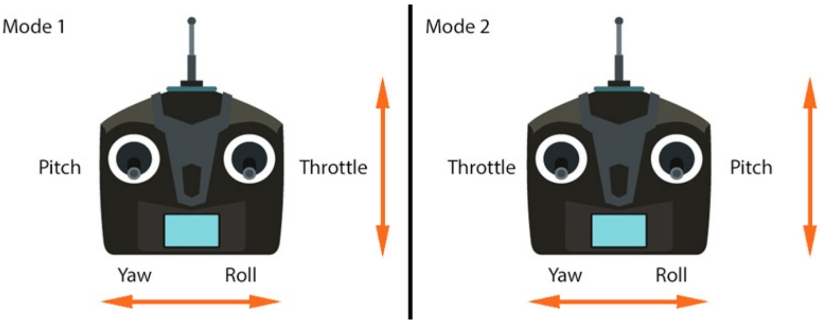


Figure 40: The only difference between modes 1 and 2 lies in the action attributed to the joysticks when they are moved vertically: In mode 1 (resp. 2), the left joystick controls vertically pitch (resp. throttle) while the right joystick controls vertically throttle (resp. pitch).
Sources:
https://blog.studiosport.fr/comment-changer-le-mode-de-votre-radiocommande-frsky-mode-1-et-mode-2/1

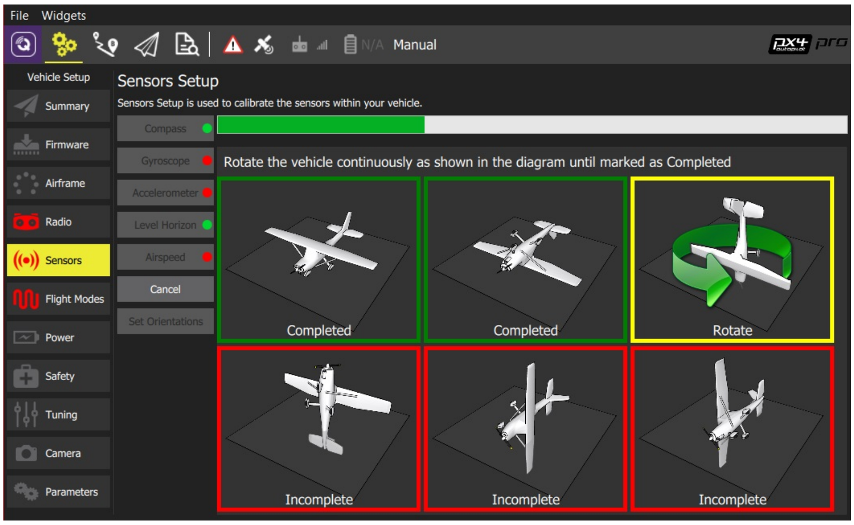
Once the calibration is done, it is possible to map additional channels (differing from the ones attributed to the main control defined at the previous step) to optional actions such as Kill Switch, Flight mode selection or even optional hardware control (servomotor), as described in the Section2.9.5. Note that some mappings are only available in "Flight mode" tab (see Figure 42). In our case, the following additional channels are mapped to actions:

- channel 5: Arming switch,
- channel 6: Flight mode selection,
- channel 7: Kill Switch,
- channel 8: AUX 1,
- channel 9: AUX 2.

where "AUX" are linked to auxiliary PWM outputs 5 and 6 from PX4 mini, as explained in Section 3.2.2.

3.2.4 Sensor

It is time to calibrate the different sensors of the drone, as illustrated in Figure 41.



Sensors Setup is used to calibrate the sensors within your vehicle.

Rotate the vehicle continuously as shown in the diagram until marked as Completed

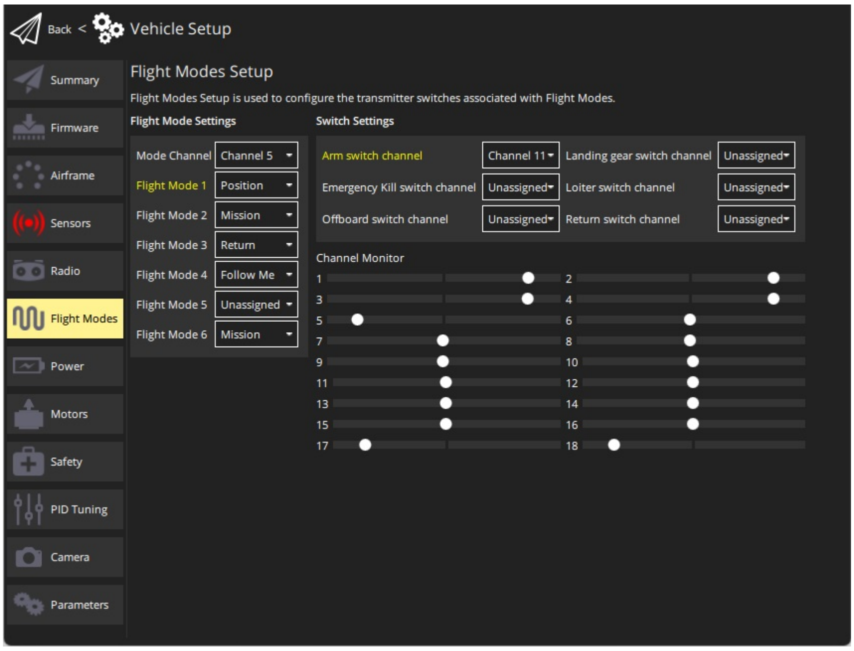
Figure 41: QGC sensor tab.
Sources: https://docs.qgroundcontrol.com/master/en/SetupView/sensors_px4.html

In this tab it is possible to calibrate the compass, the gyroscope and the accelerometer. It is also possible to define the level horizon as well as to define the relative orientation between the PX4 mini and the GPS. Normally, the calibration does not need to be redone if the drone is used in/with similar locations/conditions. The calibration process is different depending on the sensor:

- compass: the drone must turn on itself, in 6 predefined positions. This allows the drone to align itself with the north magnetic field of the Earth, thus calibrating its magnetometer (electronic compass measuring magnetic fields) [63].
- gyroscope: the drone must be placed on a flat surface. This allows the drone to detect relative movement from the level horizon, in order to ensure stability.
- accelerometer: the drone must be placed in 6 predefined positions (same as compass) without moving. This will allow the drone to detect the rate of change of movements, in order to ensure stability.
- level horizon: the drone must be placed in its hover position. This allows the drone to know its inclination with respect to the ground and thus remain straight when it glides

3.2.5 Flight modes

Once the radio channels of the remote are mapped to the drone's actions, this is time to set up the different flight modes of the drones, as illustrated in Figure 42.



Flight Modes Setup

Flight Modes Setup is used to configure the transmitter switches associated with Flight Modes.

Flight Mode Settings

Mode Channel: Channel 5
Flight Mode 1: Position
Flight Mode 2: Mission
Flight Mode 3: Return
Flight Mode 4: Follow Me
Flight Mode 5: Unassigned
Flight Mode 6: Mission

Switch Settings

Arm switch channel: Channel 11
Landing gear switch channel: Unassigned
Emergency Kill switch channel: Unassigned
Loiter switch channel: Unassigned
Offboard switch channel: Unassigned
Return switch channel: Unassigned

Figure 42: QGC flights mode tab.
Sources: https://docs.qgroundcontrol.com/master/en/SetupView/FlightModes.html

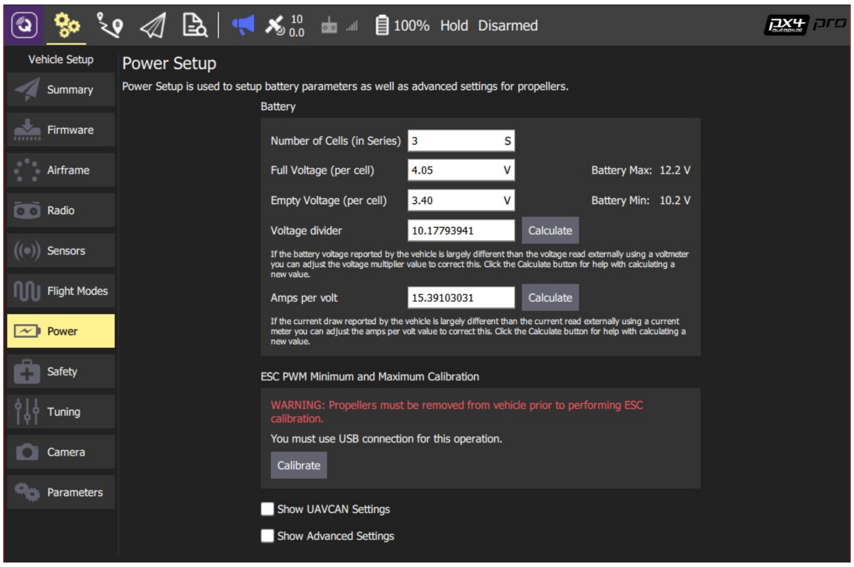
Depending on the type of hardware used to select the flight modes, a different number of flight modes can be encoded. For example, in our case, a 3-position switch is used, so a maximum of 3 different flight modes can be set up. In the following, a brief description of the flight modes used during the year will be given below. The complete description of the available flight modes can be found on this web page[64].Paid attention that some modes require certain criteria in order to be used (GPS position or altitude or remote control for instance). If the criteria are not validated, then the drone will automatically switch to a less restrictive flight mode.

- Position mode: This manual flight mode is very pleasant for beginners since the drone fixes its position once the controls are released. To do so, this mode requires the presence of a GPS to locate the drone in space and to be able to refocus if the wind makes it deviate from the target position. This mode allows to evaluate the robustness of the control system and the accuracy of the GPS.
- Altitude mode: This is the manual flight mode that the PX4 mini automatically switches to if the Position mode is selected beforehand but the GPS is not connected or has failed to locate the drone in space. This time, when the controls are released, the drone maintains its altitude , but not its position. It will therefore move at constant altitude until it has dissipated all the energy from its acceleration or if the wind blows.
- Takeoff Mode: This automatic flight mode allows the drone to automatically climb to a predefined takeoff altitude, once the motors are started.
- Landing Mode: This automatic flight mode allows the drone to automatically land at a predefined landing speed.

In our case, automatic takeoff/landing and position mode are mainly use.

3.2.6 Power

This tab allows to enter the battery information as well as performing ESC calibration if needed, as illustrated in Figure 43.



Power Setup

Power Setup is used to setup battery parameters as well as advanced settings for propellers.

Battery

Number of Cells (in Series): 3 S

Full Voltage (per cell): 4.05 V — Battery Max: 12.2 V

Empty Voltage (per cell): 3.40 V — Battery Min: 10.2 V

Voltage divider: 10.17793941

If the battery voltage reported by the vehicle is largely different than the voltage read externally using a voltmeter you can adjust the voltage multiplier value to correct this. Click the Calculate button for help with calculating a new value.

Amps per volt: 15.39103031

If the current draw reported by the vehicle is largely different than the current read externally using a current meter you can adjust the amps per volt value to correct this. Click the Calculate button for help with calculating a new value.

ESC PWM Minimum and Maximum Calibration

WARNING: Propellers must be removed from vehicle prior to performing ESC calibration.

You must use USB connection for this operation.

☐ Show UAVCAN Settings

☐ Show Advanced Settings

Figure 43: QGC power tab.
Sources: https://docs.qgroundcontrol.com/master/en/SetupView/Power.html

The voltage (S) of the battery used must be specified as well as the voltage values considering that a cell is full or empty. On the basis of these values, the PX4 mini will be able to evaluate the state of the battery when the PDB returns current and voltage measurements and thus trigger safety measures if necessary. The calibration of the ESCs can be activated by pressing the corresponding button. This allows to see if the FC and the ESCs understand each other. Please note that you should always remove the rotors from the motors when starting the calibration because the motors could run at maximum speed if they do not support the calibration.

3.2.7 Motor

Once the information regarding the battery have been recorded, the motors can be individually tested in the motor tab, as illustrated in Figure 44

This tab allows you to test the motors history to check if the connections between the battery, the possible PDB, the ESCs and the motors are correct. It also allows you to see if the motors are arranged correctly and that they rotate in the expected direction in relation to the airframe selected in Section 3.2.2[27]. Note that it is recommended to remove the rotors from the motors. A video illustrating this step is available in Appendix A.1.2.

3.2.8 Safety

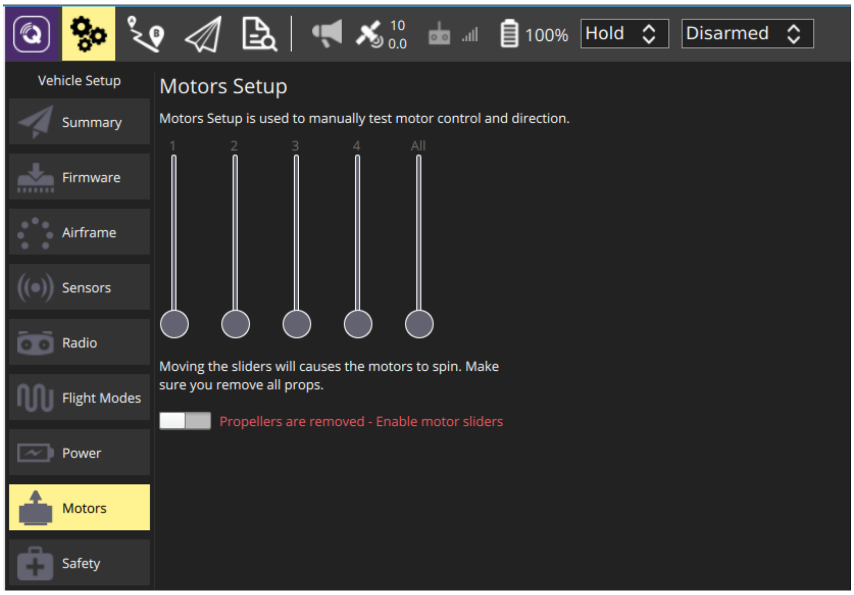
Before flying the drone, it is important to anticipate the behaviors that the drone should exhibit if a problem occurs while in flight, as shown in Figure 45.

Among the problems that can be encountered, there are (non-exhaustive list):

- low battery,
- loss of remote control signal,

---

[27] If a motor does not rotate in the appropriate direction, it is sufficient to swap 2 motor-ESC connections. If a motor is not arranged correctly (In our case, motor 1 should be top right), then this means that the ESC connector is connected to the wrong PWM output of the PX4 mini.



Motors Setup

Motors Setup is used to manually test motor control and direction.

Moving the sliders will causes the motors to spin. Make sure you remove all props.

Propellers are removed - Enable motor sliders

Figure 44: QGC motor tab.
Sources: https://docs.qgroundcontrol.com/master/en/SetupView/Motors.html

- obstacle encountered,
- limit of the flight area reached.

Among the options, named failsafe action, available if a problem occurs, it is possible to choose to land the drone directly or to trigger the RTH (return to home)[28] for instance. In our case, the automatic landing option has been selected for each failsafe action.

3.2.9 Tuning

The last thing to do before taking off the drone is to configure the control system, as shown in Figure 46.

The control system, which will be described in more detail in Section 5, is composed of 4 controllers whose gains can be manually modified. There is even an option to perform automatic tuning but this requires the use of a telemetry kit. The control system will determine the overall behavior of the drone in flight, so this step should not be neglected.

3.2.10 Camera

In order to take nice pictures/videos during the flight, it is possible to configure the camera, as shown in Figure 47.

This tab allows you to configure the triggering of the camera using the remote control. However, to date, it has not yet been possible to configure the FPV camera of the kit. A later research work could look into this problem.

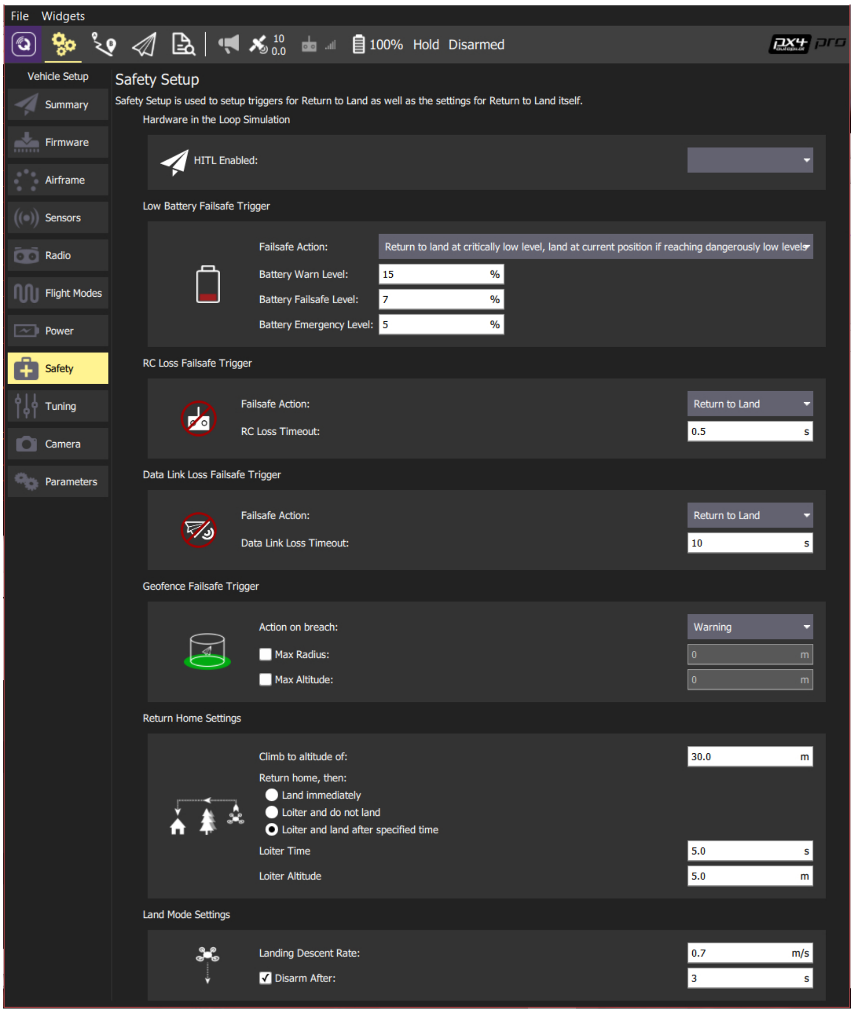
3.2.11 Parameters

Each of the PX4 mini parameters saved in the previous steps is visible in the parameters tab, as shown in Figure 48.

the list of parameters is very useful to check the changes made (indicated in red). It is also very easy to save the parameters to return the drone to a certain known state. It is impossible to describe all the parameters. Below is a short description of the parameters found useful during the year:

---

[28] If RTL is triggered, the drone will fly back automatically to its take-off area



Safety Setup

Safety Setup is used to setup triggers for Return to Land as well as the settings for Return to Land itself.

Hardware in the Loop Simulation

HITL Enabled:

Low Battery Failsafe Trigger

Failsafe Action: Return to land at critically low level, land at current position if reaching dangerously low levels
Battery Warn Level: 15 %
Battery Failsafe Level: 7 %
Battery Emergency Level: 5 %

RC Loss Failsafe Trigger

Failsafe Action: Return to Land
RC Loss Timeout: 0.5 s

Data Link Loss Failsafe Trigger

Failsafe Action: Return to Land
Data Link Loss Timeout: 10 s

Geofence Failsafe Trigger

Action on breach: Warning
☐ Max Radius: 0 m
☐ Max Altitude: 0 m

Return Home Settings

Climb to altitude of: 30.0 m
Return home, then:
○ Land immediately
○ Loiter and do not land
● Loiter and land after specified time
Loiter Time 5.0 s
Loiter Altitude 5.0 m

Land Mode Settings

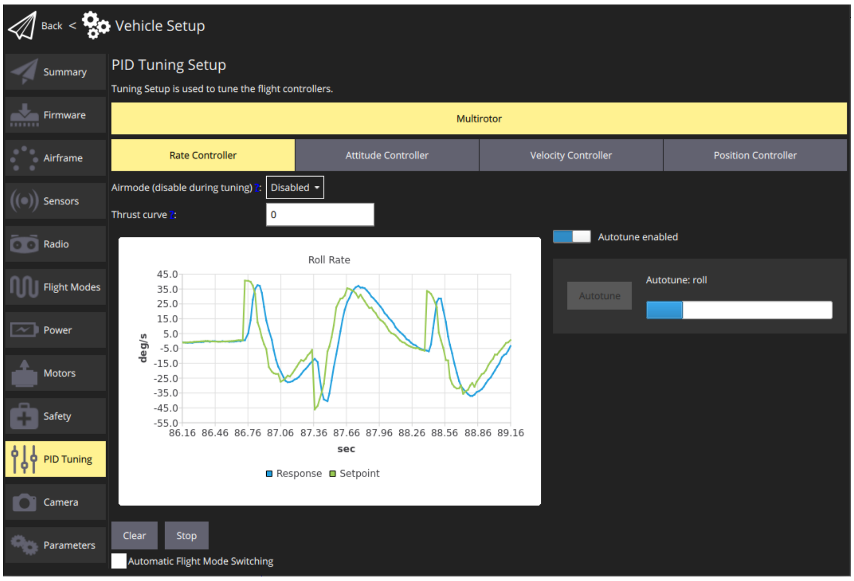
Landing Descent Rate: 0.7 m/s
☑ Disarm After: 3 s

Figure 45: QGC safety tab.
Sources: https://docs.qgroundcontrol.com/master/en/SetupView/Safety.html

- *COM_ARM_WO_GPS*: allows to arm the drone only with GPS. It was a very useful software security since the main flight mode used for the drone was position, which requires GPS position.
- *COM_RC_ARM_HYST*: represents the time that the left joystick must be positioned at the bottom right in order to trigger the arming or disarming of the vehicle. Very used at the beginning, it has been replaced by an arm/disarm switch (*COM_ARM_SWISBTN* = 0), for more convenience.
- *COM_DISARM_LAND* and *COM_DISARM_PRFLT*: set the time-out for auto-disarm of the motor after landing or if the drone cannot take-off. It allows to stop the motors if the drone does not detect the remote during take-off for instance (and that RC loss failsafe has not been set up).
- *COM_PREARM_MODE* and *CBRK_IO_SAFETY*: The combination of these two parameters indicates the conditions under which the current flow through the circuit. By setting *COM_PREARM_MODE* = 2 and *CBRK_IO_SAFETY* = 0, it allows to move optional hardware and not motors when the drone is power-up. The current will flow in the motors only when the arm switch will be triggered.

4 Remote coding

In this section, some information regarding the programming of the Tanaris X8 remote control will be given.



Tuning Setup is used to tune the flight controllers.

Airmode (disable during tuning) : Disabled

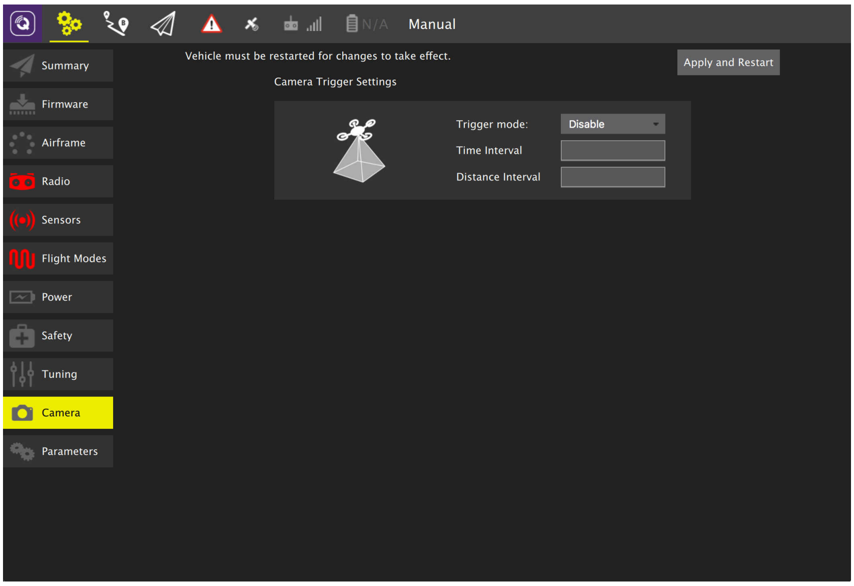
Thrust curve : 0

Autotune enabled

Autotune: roll

☐ Automatic Flight Mode Switching

Figure 46: QGC tuning tab.
Sources: https://docs.qgroundcontrol.com/master/en/SetupView/tuning_px4.html



Vehicle must be restarted for changes to take effect.

Camera Trigger Settings

Trigger mode: Disable

Time Interval

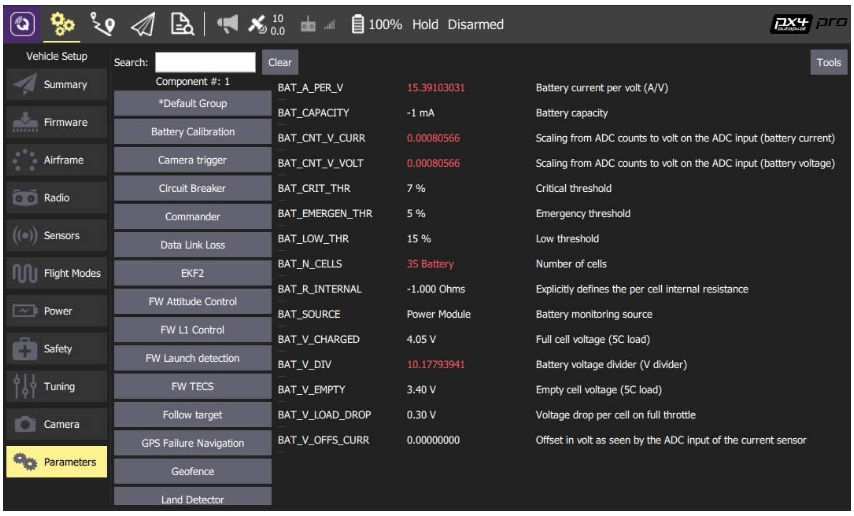
Distance Interval

Figure 47: QGC camera tab.
Sources: https://docs.qgroundcontrol.com/master/en/SetupView/Camera.html

As a reminder, the first step before configuring the transmitter is to bind it with an adequate receiver (FrSky X8R in this case). Once this it done, the remote control can be programmed by going through all the menus.

4.1 Creation new model

The first step is to create a new model, by giving it a name, in the remote control which will take care of recording all the parameters that will be configured later. Thus, it will be possible to use the remote control with several drones: it will be enough to simply change the model to go from one configuration to another. It is also possible to specify which receiver will be able to link or not with the remote control. This is illustrated in Figure 49



BAT_A_PER_V	15.39103031	Battery current per volt (A/V)
BAT_CAPACITY	-1 mA	Battery capacity
BAT_CNT_V_CURR	0.00080566	Scaling from ADC counts to volt on the ADC input (battery current)
BAT_CNT_V_VOLT	0.00080566	Scaling from ADC counts to volt on the ADC input (battery voltage)
BAT_CRIT_THR	7 %	Critical threshold
BAT_EMERGEN_THR	5 %	Emergency threshold
BAT_LOW_THR	15 %	Low threshold
BAT_N_CELLS	3S Battery	Number of cells
BAT_R_INTERNAL	-1.000 Ohms	Explicitly defines the per cell internal resistance
BAT_SOURCE	Power Module	Battery monitoring source
BAT_V_CHARGED	4.05 V	Full cell voltage (5C load)
BAT_V_DIV	10.17793941	Battery voltage divider (V divider)
BAT_V_EMPTY	3.40 V	Empty cell voltage (5C load)
BAT_V_LOAD_DROP	0.30 V	Voltage drop per cell on full throttle
BAT_V_OFFS_CURR	0.00000000	Offset in volt as seen by the ADC input of the current sensor

Figure 48: QGC parameters tab.
Sources: https://docs.qgroundcontrol.com/master/en/SetupView/Parameters.html

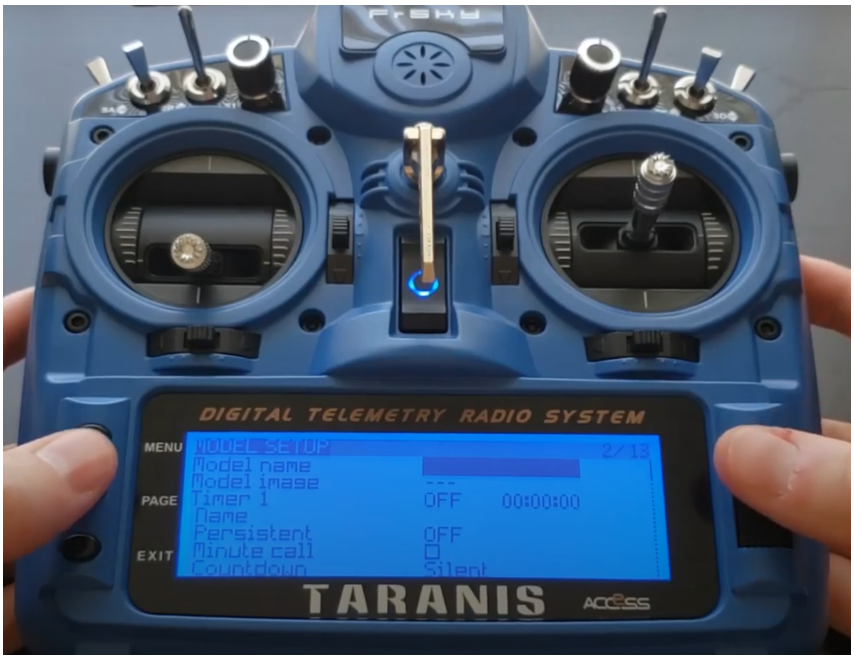


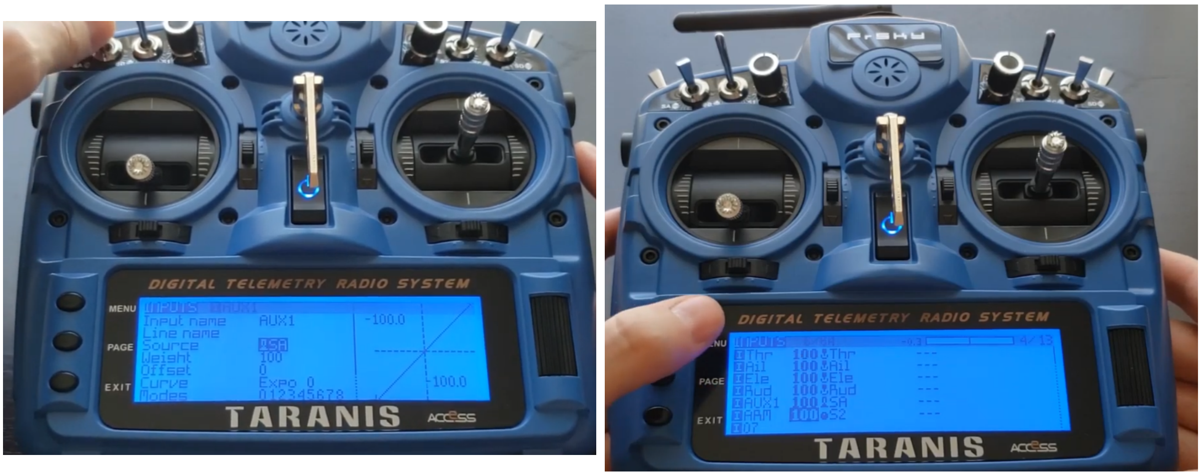
Figure 49: Creation of a new model in the remote, to store future parameters.
Sources: https://www.youtube.com/watch?v=w40P1CQQ9Ls

4.2 Adding inputs

Now that the model is created, it is necessary to indicate which inputs of the remote control are to be taken into consideration (for instance, a limited number of switches can be used). To do this, it is necessary to assign a name to the input and then select the source by physically moving the hardware input (this way the remote control records the min and max values of the signal corresponding to the input). This is illustrated in Figure 50

4.3 Mapping inputs to channels

Once all the remote control inputs have been assigned to the model, each of them needs to be assigned to a different radio channel in the mixes tab, see Figure 51. Each mix is assigned a source (i.e. input)

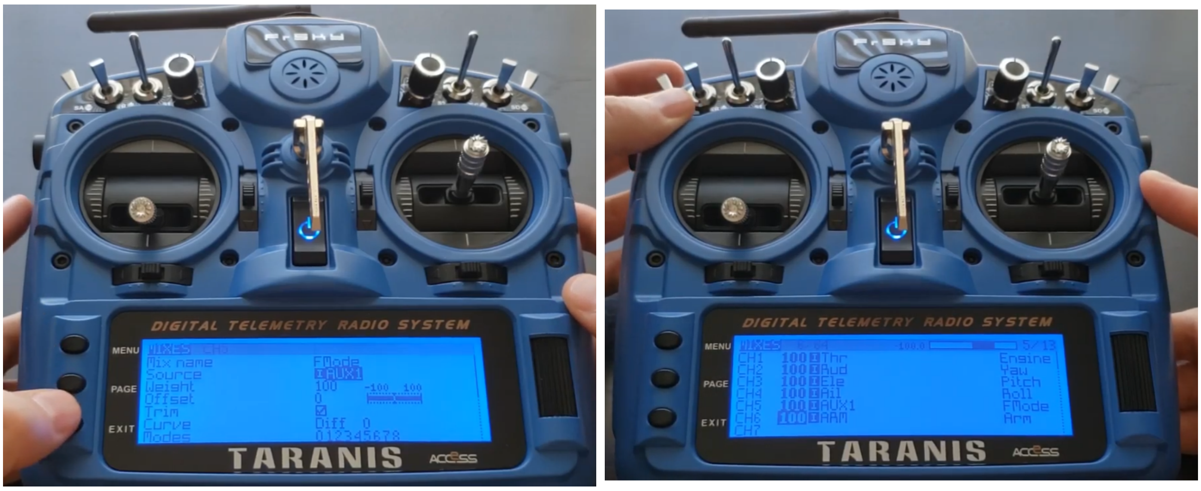


```
INPUTS
Input name      AUX1        -100.0
Line name
Source          SA
Weight          100
Offset          0
Curve           Expo 0        100.0
Modes           012345678
```

```
INPUTS                -0.3          4/13
I Thr     100 Thr        ---
I Ail     100 Ail        ---
I Ele     100 Ele        ---
I Rud     100 Rud        ---
I AUX1    100 SA         ---
I ARM     100 S2         ---
I 07
```

Figure 50: Adding physical inputs of the remote in the model.
Sources: https://www.youtube.com/watch?v=w40P1CQQ9Ls

and a name (which may be different from the name of the input).



```
MIXES
Mix name        FMode
Source          I AUX1
Weight          100      -100   100
Offset          0
Trim            ☑
Curve           Diff  0
Modes           012345678
```

```
MIXES                 -100.0          5/13
CH1   100 I Thr                 Engine
CH2   100 I Rud                 Yaw
CH3   100 I Ele                 Pitch
CH4   100 I Ail                 Roll
CH5   100 I AUX1                FMode
CH6   100 I ARM                 Arm
CH7
```

Figure 51: mapping each physical input of the remote to a different radio channel.
Sources: https://www.youtube.com/watch?v=w40P1CQQ9Ls

This section has covered the basics of programming the Tanaris X8 remote control. Of course, not all the options of the remote control have been covered, this is just an overview of the most useful functions.

5 Control system

This section will cover the control system of the PX4 mini. The information given in this section will be largely taken from the official web page [65]. The control system is composed of 4 controllers (P and PID). These controllers are arranged in levels: the most important level controller passes its results to the controller of the next level down. The 4 controllers, by level importance, are: rate controller, attitude controller, velocity controller and position controller. Therefore, it is very important to tune the rate controller because it will impact all other controllers as illustrated in Figure 52

All the controllers present in Figure 52 use Extended Kalman Filter algorithm to process sensor measurements and estimate the states of the drone.

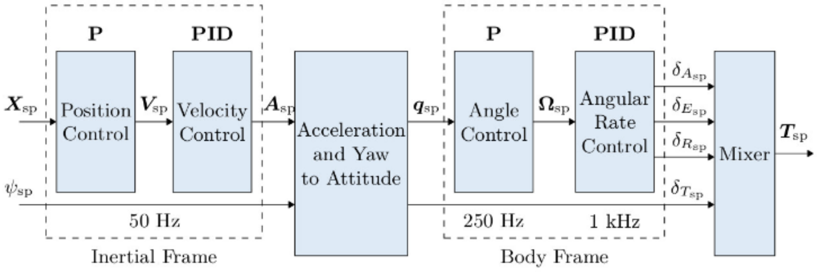


Figure 52: Cascade of controllers, where the highest-level controller is situated on the right.
Sources: https://docs.px4.io/v1.12/en/flight_stack/controller_diagrams.html

5.1 Rate controller

The rate controller regulates the yaw, pitch and throttle variations of the drone, so there is an independent PID for each movement. If the variations are too big, the drone will be quickly destabilized while it has an extremely long reaction time if the rates are too small. Note that the term derivative here is used to avoid having an instantaneous response (i.e. a spike) when the setpoint varies. This is to smooth the response of the controller.

5.2 Attitude controller

The attitude controller is responsible for controlling the orientation of the drone. It is much easier to tune than the rate controller since there is only one proportional term. It is enough to make the drone fly in stabilized mode and to increase the proportional term until oscillations or overshoots are observed which indicate that the response is too strong.

5.3 Velocity controller

The velocity controller is responsible for controlling the velocity of the drone as the name suggests it. It is possible to push the system not to respect the velocity controller by limiting for example the maximum horizontal speed that the drone can take at any time. This is a process that can be dangerous in a sense that it could increase by a lot the response time of the whole system, which would be unable to react quickly to an external disturbance stronger than usual.

5.4 Position controller

The position controller is responsible for controlling the position states of the drone as the name suggests it. This is the lowest-level controller. Its tuning does not very much affect the overall behavior of the system.

6 Evaluation

In this section, the flight performances of the drone kit will be analysed. In fact, the goal is to check if the behavior of the kit can be determined mathematically before it flies. Since the motors are not well known, it will be necessary to go through an experimental phase in order to get some characteristics of the engines. Then, the construction and the configuration of the drone will be verified by flying the drone. Finally, the drone will undergo some endurance tests and the results will be checked to see if they correspond to the mathematical model.

6.1 Characterisation of the kit motors

As announced in Section 1.2.2, there is nearly no data regarding the 2205-2300KV motors. Therefore, in order to be able to predict the behavior of the drone, an experimental phase, similar to the one described in the Section 3.1.3, will be conducted. Therefore, the test bench had to be slightly modified as illustrated in Figure 53. Note that due to the tilt of the rotor blade, the tachometer also had to be tilted in order to align the sensor of the latter with the aluminium below the blade.

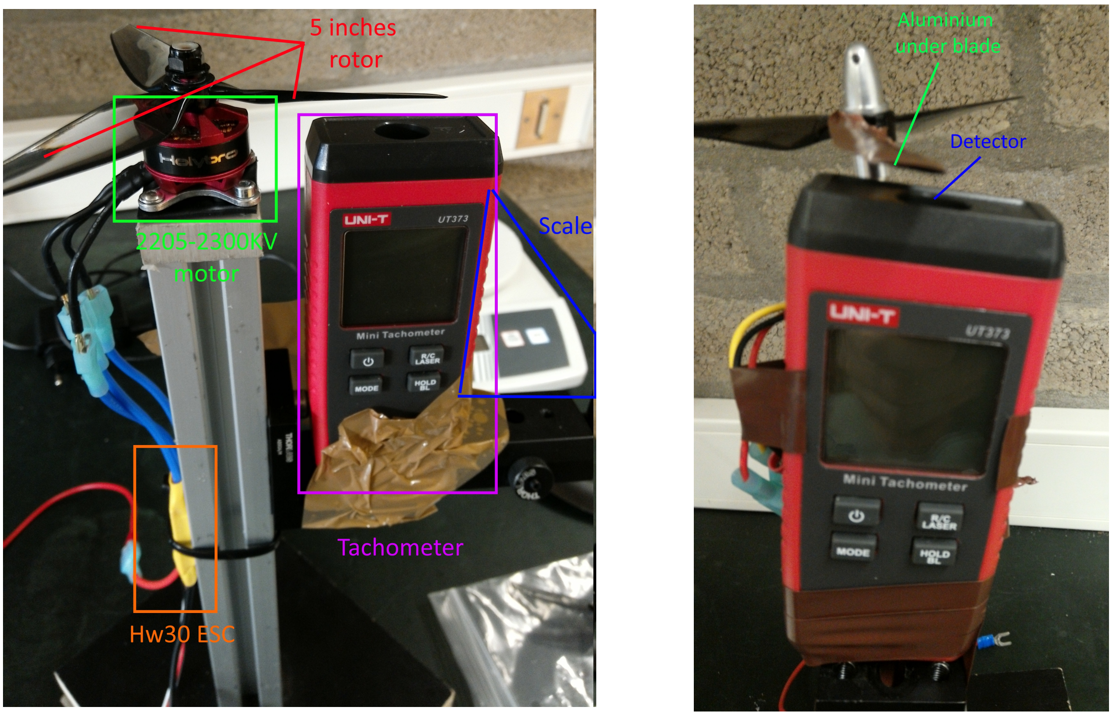


Figure 53: Adaptation of the test bench, from Section 3.1.3, to accommodate the kit motors.

The original results obtained with the adapted test bench are available on the Google spreadsheet "2205-2300KV + 5" rotor", in Appendix A.2.1. They will be summarized in Table 22 for convenience.

Experimental measurements on kit motors													
Duty cycle [%]	40	45	50	55	60	65	70	75	80	85	90	95	100
Case 1: 10V													
Current [A]	2.50	2.90	3.00	3.45	3.90	4.40	5.10	5.60	6.47	7.00	7.90	9.00	10.00
Power [W]	25	29	30.6	34.6	39.56	44.20	51	56	65.60	71.40	80	90.20	100
Thrust [g]	119	136	142	157	176	196	215	233	261	279	300	332	350
Case 2: 11.1V													
Current [A]	2.2	3.3	3.2	3.8	4.45	5.15	5.9	6.7	7.65	8.6	9.8	-	-
Power [W]	31	32.6	34	43	49.8	55.6	66	75	87	96.8	108	-	-
Thrust [g]	141	133	137	184.5	213	239	267	295	320	350	383	-	-
Case 3: 12V													
Current [A]	2.78	3.25	3.5	3.9	4.46	4.9	5.7	6.4	7.2	8.2	9.3	10	-
Power [W]	33.5	38.9	41.6	47.6	54	60.6	68	77.5	87.6	98.6	114	116.8	-
Thrust [g]	146	167	174	194	214	236	260	286	315	347	382	390	-

Table 22: Experimental data acquired for the kit motors, using the adapted test bench.

Looking at the values in Table 22, it can be seen once again that the voltage has no influence on the overall behavior of the motor. Therefore, the following will only refer to the case closest to the battery used with the kit, i.e. case 3 = 12V).

Using the experimental data, the thrust versus thrust over power ratio curve can be plotted, which is shown in Figure 54 for case 3.

This curve will be very useful in section 6.3, when predicting the flight time of the UAV

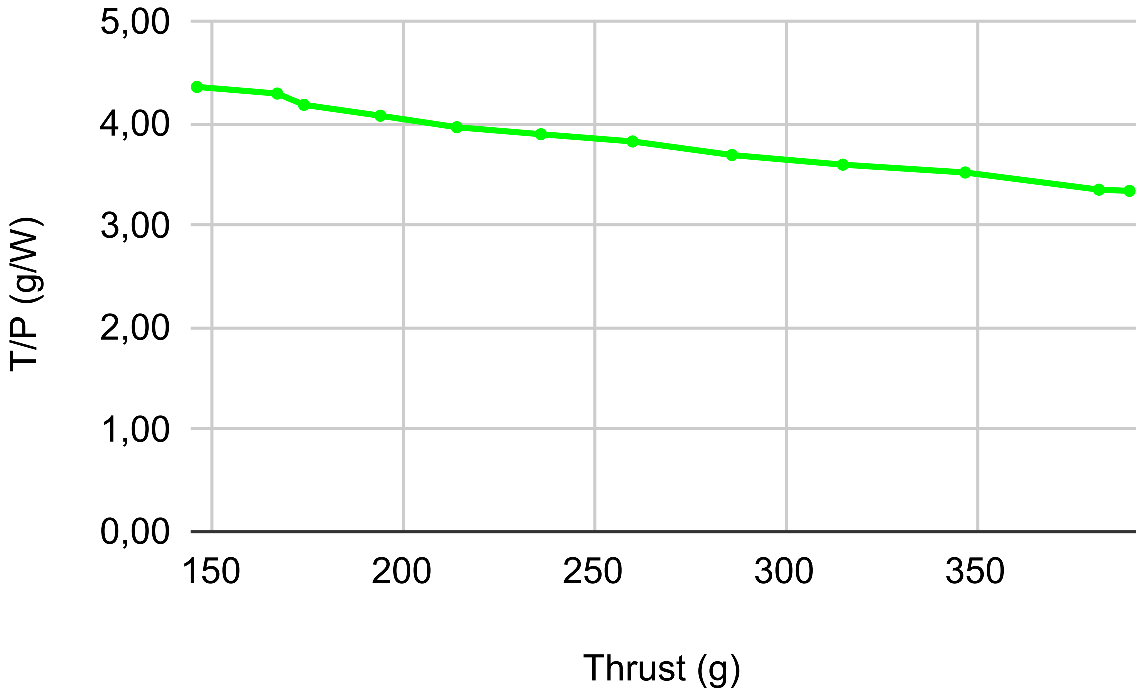


Figure 54: Evolution of the kit motor efficiency, $\frac{T}{P}$, in function of the thrust T.

6.2 Flights

It is now time to check if the drone can fly. Knowing that the drone weighs about 650g when fully assembled, and that a motor can generate up to about 400g of thrust when it is at 100%, a quick calculation $(4 \cdot 400 > 650)$ shows that there is enough thrust to make the drone take off.

Before launching it into the wild, a final precaution is to check that the control system is working properly. To do this, a paradrone will be used to artificially move the drone and observe the reactions of the control system, while preventing the drone from flying away. Note that in order for the drone to be maintained on the paradrone, a clamping piece had to be machined. The paradrone and the clamping piece can be seen in Figure 55.

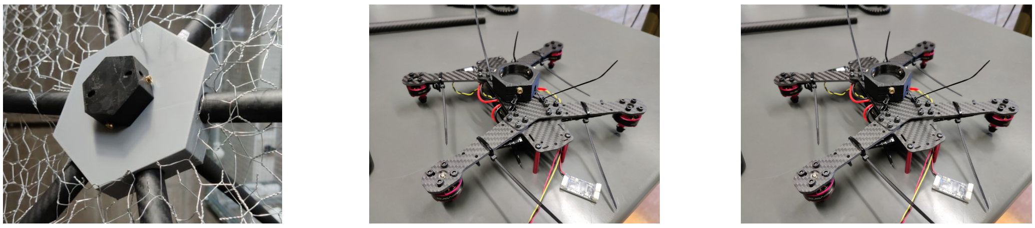


Figure 55: The paradrone, on the left, which will hold the drone in place with the help of the clamp on the right.

A video illustrating the paradrone in action is available in Appendix A.1.3.

Now that all possible checks have been made, it's time to give it a shot and see what happens. A video of the live demo that was performed on 04/15/2022 is available in the AppendixA.1.5. Note that the explanations given are in French. [29]

---

[29] Peut etre ajouter sous titres dans vidéo

6.3 Flight time prediction

From a physical point of view, the flight time is the time during which the drone is able to stay in the air. Since the drone consumes more power when it is moving, the drone must hover at a fixed location. In a perfect world with no air friction and no wind, this is the flight mode that would allow the drone to stay in the air the longest because the motors produce just enough thrust to compensate the weight and nothing more. In practice, the flight mode position will be used to keep the drone fixed in the 3D space. The control system should recenter the drone if the wind wants to take it away, at the cost of a little more power consumption. From then on, several measurements will be made, in the closest climatic conditions, in order to have coherent measurements.

Therefore, 3 endurance tests were performed and the results are presented in the Table 23. Note that for these tests, the battery was fully charged at the beginning and discharged during the test until it reached 2% battery level, from which it triggered an emergency landing failsafe. A video illustrating the test is available in Appendix A.1.4.

Kit flight time tests				
Test number	GPS lock time [min]	Flight time [min]	battery remaining at the end [%]	Weather
1	00:53	09:41	2	Cloudy Light wind
2	00:52	10:03	2	Blue sky No wind
3	00:55	10:01	2	Blue sky No wind

Table 23: Flight times obtained for the learning kit drone, in hovering.

From a mathematical point of view, the flight time is expressed as the ratio between the battery power and the power consumed by the motors, as expressed by the following equation:

$$t_{fly} = \frac{C_{bat} \cdot V_b}{N_{motor} \cdot P_{motor}} \cdot t_{h\_to\_min} \tag{49}$$

where C_{bat} represents the battery capacity, V_b the battery voltage, N_{motor} the number of motors composing the drone and $t_{h\_to\_min}$ a factor to convert hour to minutes.

In the case of the kit, the rotors must generate a global thrust to compensate the weight of the whole kit, which corresponds to 650g. This means that a single rotor must generate a thrust of 650/4 = 162.5g.

Looking at the Figure 54, it can be seen that a T/P ratio of approximately 4.3 is obtained, which means that the consumption of the motor, to generate the required thrust of 162.5g, is approximately 162.5/4.3 = 29.4.

In order to be more precise on the evaluation of the power consumed by the motor, a linear interpolation of the experimental data will be carried out:

$$\begin{aligned} &\bar{P}_{motor}(x) = 33.5 + (x - 146) \cdot \frac{38.9 - 33.5}{167 - 146} \\ \Longleftrightarrow \quad &\Leftrightarrow \bar{P}_{motor}(162.5) = 33.5 + (162.5 - 146) \cdot \frac{38.9 - 33.5}{167 - 146} \approx 37.43 \text{ W} \end{aligned} \tag{50}$$

Since the drone is composed of 4 motors and that the battery capacity is 1.55Ah while its voltage is 14.8V, the value obtained in Equation (50) can be injected into Equation (49):

$$t_{fly} = \frac{1.55 \cdot 14.8}{4 \cdot 37.43} \cdot 60 = 9.19\,\text{min} = 9\,\text{min}\,12\,\text{s} \tag{51}$$

Looking at the values obtained for the flight time (see Table 23), the mathematical predicted time (=9 min 12s) is lower than the average of the 3 measurements (= 9 min 55s). This means that the motors consumed less power than predicted by the linear interpolation (see Equation (50)). Note also that the weight of the kit has been rounded, meaning that the actual thrust required to hover the drone is certainly less than 162.5g. In order to refine the calculated flight time, it would be necessary to measure precisely the weight of the kit as well as the consumption of the motor.

Part V

Design

This part is focused on the construction of the drone, from its theoretical design to its practical realization. A section will also be dedicated to the evaluation of its performances (regarding the specifications). Therefore, it will refer to several pieces of information scattered throughout the other sections. This part is special in a sense that it summarizes the work of the collaboration over one quadrimester (February 2022 → May 2022) of François, a civil engineer student in electricity (ULiège), with Luca, an electromechanical student (Henallux). The plan of this part is as follows: first, a theoretical design, based on experimental measurements will be presented. Then, the specifications regarding this design will be established based on calculations. After that, the parts machined from an improved version of the theoretical design will be illustrated to conclude on the performance evaluation of the drone thus built.

1 Theoretical design

This section will present the theoretical drone design that has been established partially from experimental measurements (Section 3) and components coming from the UAV kit (section 1.2). Note that this design will evolve iteratively over time towards the final design: some parts will have to be refined based on tests performed on samples while others will have to be revised following problems encountered as it will be explained in Section 3. In fact, the goal of this first theoretical design was to have some numbers in mind in order to establish plausible specifications, see section 2. It is therefore a rough approximation that will serve to launch the discussion on what can be done next. Each part of the drone will be designed before combining the whole into the long-awaited theoretical design.

1.1 General shape

A drone can take a bunch of different shapes, as already mentioned in section 2.1.3. For this first theoretical design, the goal was to be based on the advantages that a "Hybrid X" type design could bring: space in the center to store the electrical components among other things as well as good flight performance, hence the "Hybrid X" inspiration. Since at this stage, the thrust required by the drone to take off and fly smoothly is not yet known, a Hybrid X4/X6 design, was chosen. The main constraint was to offer the possibility to modify the number of motors easily without having to machine new parts. Note that according to the name "Hybrid X4/X6", the idea was to keep a symmetrical design, with a even number of motor. In the following, the notation HX4 (resp. HX6) will be used to designate the Hybrid X4 (resp. Hybrid X6) design.

1.2 Motor and rotor

In order to have some figures in mind to fix the first design, it was decided to reuse the motors and rotors studied experimentally at the section 3. To refresh the reader's memory, an illustration of these is available in Figure 56

Based on the large size of the rotors and the small K_v value of the motor[30], it can already be said that it will be a stable rather than an agile drone. The interesting figures (see appendices A.3.1 and A.3.5) to be retained for the dimensioning are the following:

- motor weight: = 50 [g],
- motor diameter: = 28 [mm],
- motor height: = 41 [mm],
- rotor weight: = 9 [g],
- rotor diameter: = 25.5 [cm],

---

[30] Remember that A2212 motor has a velocity constant $K_v = 1000$.

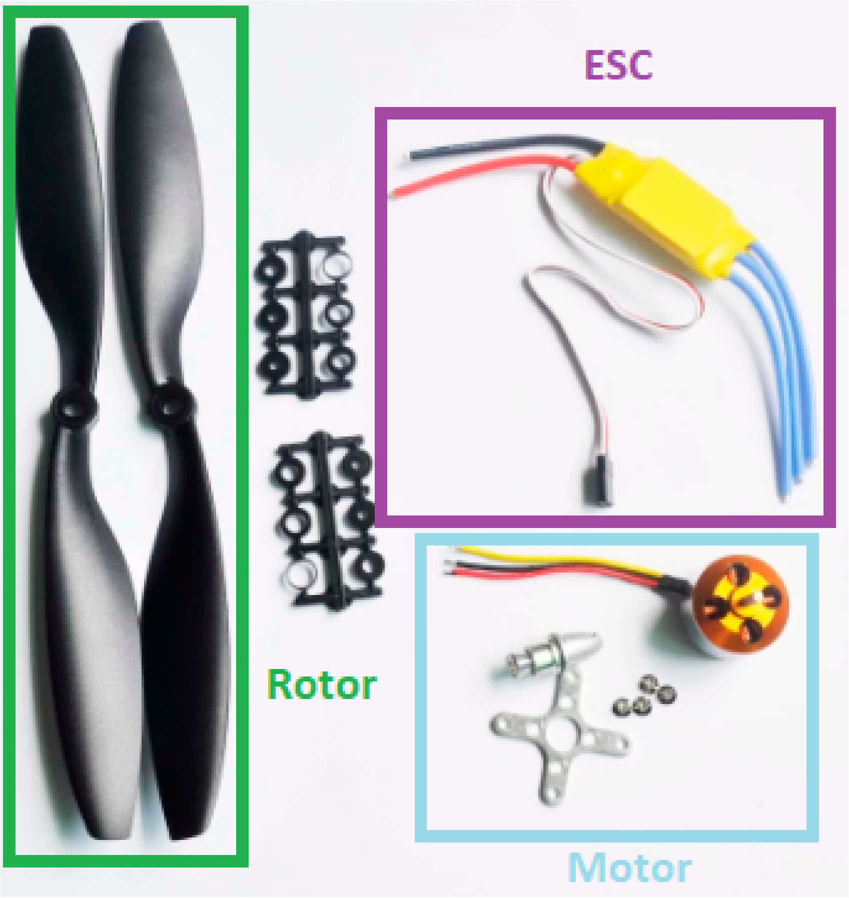


Figure 56: Motor and rotor used for the theoretical design.
Sources: https://fr.aliexpress.com/item/32818103779.html

- rotor thickness: = 2.5 [cm].

1.3 ESC

Same as for the motor and rotor, it has been decided to reuse what is already well known in terms of ESC, i.e. HW30 ESC, also used during the experimental measurements. To refresh the reader's memory, a representation of this ESC is available in previous Figure 56.

The choice of the ESC does not directly impact the modeling of the drone, except for the maximum value of the currents that can pass through the motors (here, 30A).The interesting figures (see appendix A.3.2) to be retained for the dimensioning are the following:

- esc weight: = 32 [g],
- esc length: = 26 [mm].
- esc width: = 55 [mm].
- esc height: = 13 [mm].

1.4 PDB

In order to properly distribute the power from the battery to our motors, this design will incorporate a power distribution board. Since the design of this drone must be able to switch intelligently between either 4 or 6 motors, it requires a PDB with at least 6 motor connections. A PDB meeting these criteria is shown in figure 57. It will be named PM07 hereafter.

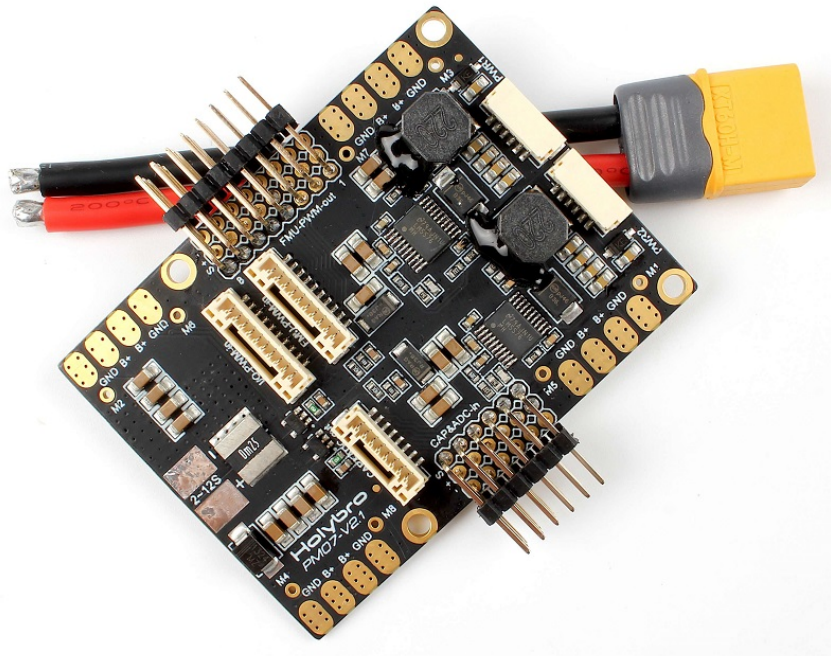


Figure 57: Power distribution board used for the theoretical design.
Sources: https://docs.px4.io/v1.12/en/power_module/holybro_pm07_pixhawk4_power_module.html

The interesting figures (see appendix A.3.6) to be retained for the dimensioning are the following:

- pdb weight: = 36 [g],
- pdb length: = 68 [mm],
- pdb width: = 50 [mm],
- pdb height: = 8 [mm],

1.5 Battery

The battery powering the drone must be compatible with the PDB, the ESCs and the motors selected in the previous step. Putting together all the specifications in terms of power supply, a choice must be made between 2-3S batteries. This is one of the most critical choices for the design: indeed, the weight of the battery often accounts for a good third of the overall weight of the drone and obviously, the heavier the drone, the more thrust it requires to fly. But the more cells the battery contains (and therefore the heavier it is), the more autonomy it will have (for the same current)[31]. It is therefore a trade-off between weight and flight time. For the purpose of this theoretical design, the battery illustrated in Figure 58 has been chosen for its autonomy.

The interesting figures (see appendix A.3.7) to be retained for the dimensioning are the following:

- battery capacity: = 5000 [mAh],
- battery voltage: = 11.1 [V],
- battery C-rate: = 25 [−],
- battery length: = 135 [mm],
- battery width: = 44 [mm],
- battery height: = 28.5 [mm],

---

[31] The detailed calculation about flight time will be performed in section 2

Figure 58: Traxxas battery 3S, 5000 mAh used for the theoretical design.
Sources: https://www.mcmracing.com/fr/home/134831-traxxas-trx2832x-power-cell-lipo-5000mah-111v-3s-25c-short-135mm-020334283290

- battery weight: = 340 [g].

1.6 Electronics

The last step is to choose the brain of the drone, i.e. the flight controller. There are many models on the market whose price often varies according to the power and capabilities of the FC. One of the constraints imposed for the construction of the final drone was to reserve, if possible, the FC of the kit (see ...). Therefore, in this theoretical design, the same FC will be used, namely the Pixhawk 4 mini "PX4 mini", shown in figure 59. In addition, the GPS, from the same kit, will also be reused.

The interesting figures (see appendices A.3.8 and A.3.9) to be retained for the dimensioning are the following:

- PX4 mini weight: = 36 [g],
- PX4 mini volume: = $55 \cdot 38 \cdot 15.5$ [mm],
- GPS weight: = 21 [g],
- GPS diameter: = 5 [cm],
- GPS height: = 1.5 [cm],

1.7 Frame weight estimation

Now that the weight of all the elements composing the drone is known, it is still necessary to estimate the weight of the frame, which strongly depends on its size (and thus of the various elements which we chose to integrate into the drone) in order to have an idea of the global weight of the whole drone.

Figure 59: Flight controller "PX4 mini" Pixhawk 4 mini (left) and corresponding GPS module (right).
Sources: https://docs.px4.io/v1.12/en/flight_controller/pixhawk4_mini.html
https://www.reichelt.com/fr/fr/module-gps-pixhawk-px4-2-gen-connecteur-6-broches-ph-px4-2-gps-p291790.html?CCOUNTRY=443&LANGUAGE=fr&&r=1

This estimation will be done on the basis of a rough drawing of the frame, made with Google SketchUp[32].

The frame will be composed as follows:

- A top plate containing GPS and eventual payload (see section 2),
- A bottom plate containing the battery, the PX4 mini and the PDB
- Either 4 or 6 arms, each of them containing both ESC and motor.

1.7.1 Bottom and top plate

The 3 elements, of length "$length_i$" and width "$width_i$" composing the bottom plate will be represented by a square, whose size "$size\_plate$" is determined by:

$$size\_plate = max(max_i(length_i); \sum_{i=1}^{3}(width_i)) = max(135; 132) = 135 \text{ [mm]}. \tag{52}$$

From (52), it can be concluded that all the elements can fit inside a square of size 17cm (some margins have been taken into account for accordance). Since the top plate contains only the GPS module at this stage, it will also be modelled by the same square. The 2 plates are illustrated in Figure 60

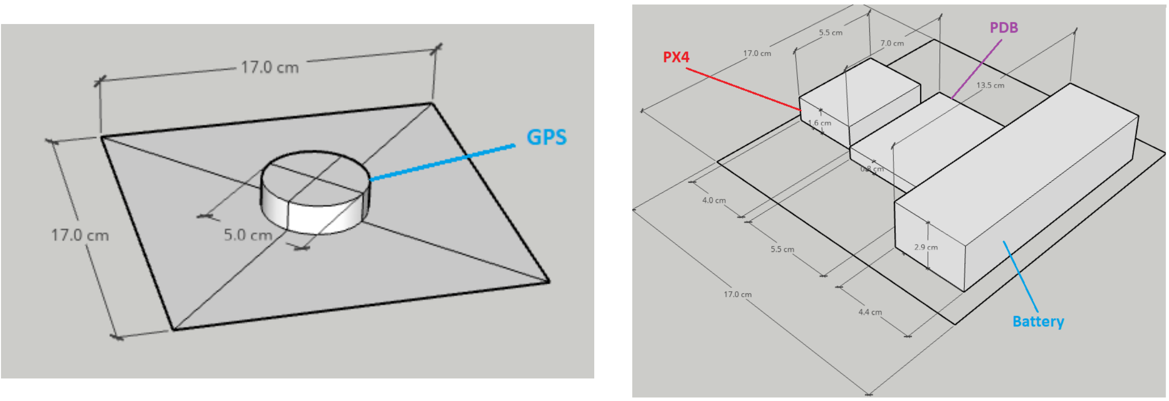


Figure 60: Theoretical design for the plates: Top plate on the left, bottom plate on the right.

---

[32] Available through this link: https://www.sketchup.com/fr/plans-and-pricing/sketchup-free

1.7.2 Arm

The arm will be modelled by a rectangle containing the motor at one extremity and the ESC at the other, as shown in Figure 61

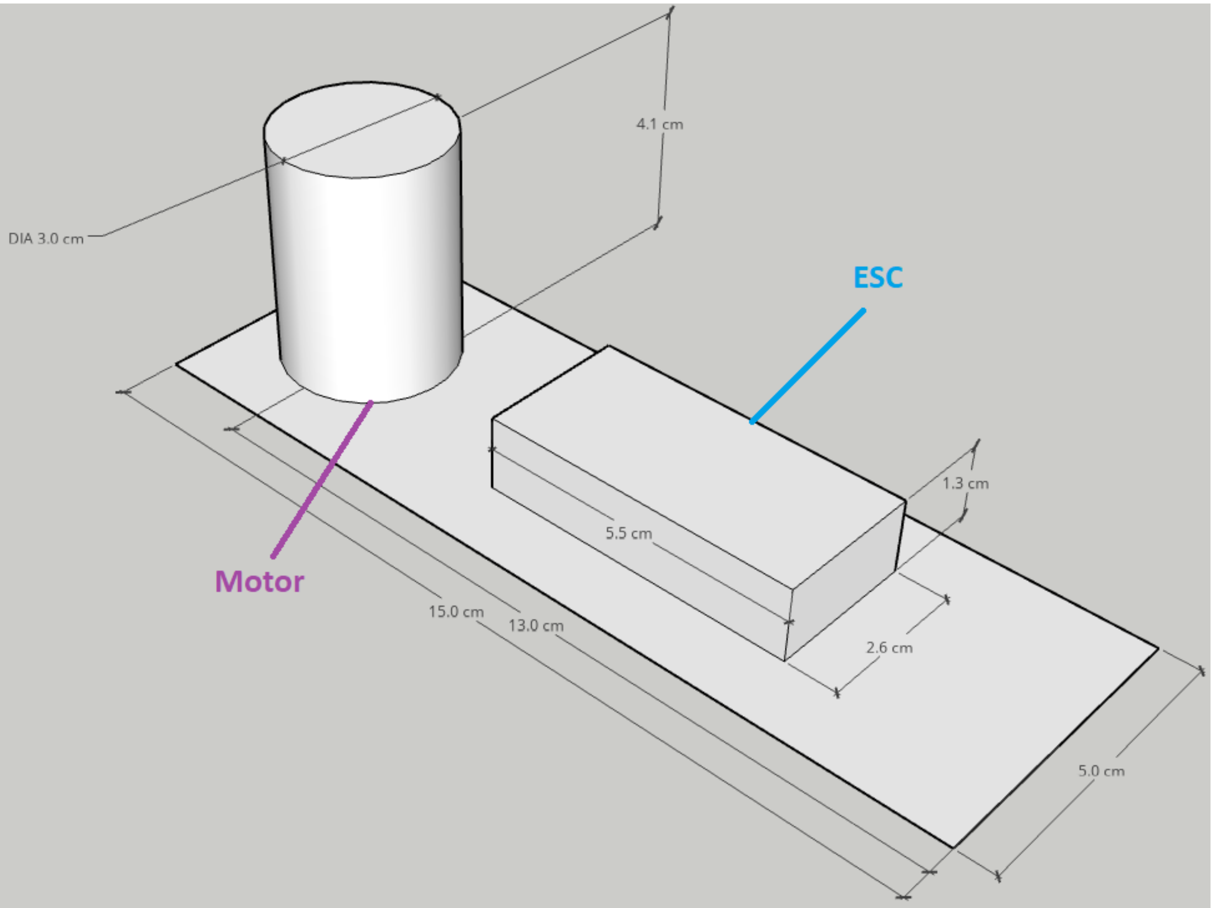


Figure 61: Theoretical design for the arms: it is composed of the motor on the left, the ESC on the right

1.7.3 Computation of the estimated weight

Now that the schematic of each part of the frame is available, it is time to estimate its weight. To do this, a thickness of 2mm (resp. 16mm) will be considered for the plates (resp. arms), which is typical of drones found in the market, to estimate the full volume of the frame. Its weight will then be estimated, for a given material, using its density.

The volume of the frame is composed of the volume of the 2 plates added to the volume of the corresponding number of arms, due to hybrid mode.
The volume of the plates is given by:

$$V_{plates\_design} = 2 \cdot 17 \cdot 17 \cdot 0.2 = 115.6 \text{ cm}^3 \tag{53}$$

The volume of one arm is given by:

$$V_{arm\_design} = 15 \cdot 5 \cdot 1.6 = 120 \text{ cm}^3 \tag{54}$$

Combining results from Equations (53) and (54) allows to compute the full volume of the frame, regarding the number of arms:

$$V_{frame\_hx4} = V_{plates\_design} + 4 \cdot V_{arm\_design} = 595.6 \text{ cm}^3 \tag{55}$$

$$V_{frame\_hx6} = V_{plates\_design} + 6 \cdot V_{arm\_design} = 835.6 \text{ cm}^3 \tag{56}$$

To compute the frame weight, 2 typical materials will be considered: carbon fiber ($\rho = 1.8$) and PLA ($\rho = 1.25$) as discussed in section 2.1.2. The results obtained for the 2 types of frame are summarised in Table 24.

	Weight of the frame [g]	
	carbon fiber ($\rho = 1.8$)	PLA ($\rho = 1.25$)
HX4	1072.08	744.5
HX6	1504.08	1044.5

Table 24: Estimation of the frame weight for the theoretical design

Now, the total weight of the drone can be computed from data provided in section 1.2 to 1.6 and frame weight estimation (see Table 24).
The weight of the different components, $W_{drone\_component}$, that will be used in the drone is given by:

$$\begin{aligned} W_{drone\_component} &= nb_{arms} \cdot (\underbrace{50}_{motor} + \underbrace{9}_{rotor} + \underbrace{32}_{ESC}) + \underbrace{36}_{PDB} + \underbrace{36}_{PX4mini} + \underbrace{21}_{GPS} + \underbrace{340}_{battery} + W_{payload} \\ &= 433 + (nb_{arms} \cdot 91) + W_{payload} \end{aligned} \tag{57}$$

where nb_{arms} represents the number of arms considered (either 4 or 6 here) and $W_{payload}$ denotes the weight of the eventual payload.

Therefore, considering a payload of 300g, the weight of the whole design can be computed from Equation (57). The results obtained are displayed in Table 24.

	Weight of the whole drone [g]	
	carbon fiber ($\rho_{cf} = 1.8$)	PLA ($\rho_{pla} = 1.25$)
HX4	1072.08 + 1097 = 2169.08	744.5 + 1097 = 1841.5
HX6	1504.08 + 1279 = 2783.08	1044.5 + 1279 = 2323.5

Table 25: Estimation of the drone weight for the theoretical design

2 Specifications

This section will present the specifications that has been established from the theoretical design. The role of this specification is to establish the objectives that the drone must be able to validate once built. First, the material for the frame will be chosen, then the flight time will be computed.

2.1 Material selection

The choice of material for the frame is crucial since it plays a role in the rigidity of the drone while contributing more or less to the overall weight.

Considering the results obtained in Table 25, using PLA[33] seems to be more interesting in view of the weight reduction that could be obtained, compared to a realization in carbon fiber. Moreover, PLA is much less expensive than carbon fiber. However, a drone made of PLA loses rigidity compared to carbon

---

[33] PLA is a thermoplastic polyester that serves as the raw material in 3-D printing or additive manufacturing processes and applications [66]

fiber. Nevertheless, a good optimization of the parts and their thickness could compensate this weakness. Finally, the Microsys laboratory[34], affiliated to the ULiège, can provide a 3D printer to machine the different parts. All these points make the balance towards PLA but carbon fiber is not excluded (see Section 3.1). The Figure 62 shows the 3D printer at our disposal to machine the parts in PLA.

2.2 Flight time

The flight time of the drone can be computed but it requires to go through a number of steps:

- First of all, it is necessary to know the total weight of the drone..
- Then, it is necessary that the motors are able to give to the drone a total thrust superior, or at least equal to its weight otherwise the drone can't take off or crashes if it is already in flight. As explained in Section 2.3.2.1, a general thumb's rule is to consider a safety factor S_f of value 2 when looking at the thrust-to-weight ratio.
- Using the experimental data the thrust that a motor has to deliver can be related to its power consumption.
- Once the power dissipated by the motor is known, it is possible to express the flight time as the ratio between the power that the battery can deliver and the motors consumption.

2.2.1 Total weight

Since the material selected is PLA (see Section 2.1), the total weight of the drone is 1841.5 g (resp. 2323.5 g) for HX4 (resp. HX6) as displayed in Table 25).

2.2.2 Thrust required by the rotors

The total thrust that the rotors must be able to provide is computed as follows:

$$T^{(g)}_{all\_rotors} = S_f \cdot W^g_{uav} = 2 \cdot 1841.5 = 3683 \text{ g for HX4} \tag{58}$$

$$T^{(g)}_{all\_rotors} = S_f \cdot W^g_{uav} = 2 \cdot 2323.5 = 4647 \text{ g for HX6} \tag{59}$$

Therefore, the thrust that a single motor must generate can be inferred from Equations (58) and (59):

$$T^{(g)}_{rotor} = \frac{T^{(g)}_{all\_rotors}}{4} = \frac{3683}{4} = 920.75 \text{ g for HX4} \tag{60}$$

$$T^{(g)}_{rotor} = \frac{T^{(g)}_{all\_rotors}}{6} = \frac{4647}{6} = 775.5 \text{ g for HX6} \tag{61}$$

2.2.3 Power consumption: questioning

By looking at the Table 10 containing data collected during the experimental phase for input voltage = 11.1V[35] (see section 3.1.3), It can be seen that the maximum thrust that a motor can provide is 587g at 100% duty cycle. This value is lower than the values required by the 2 designs HX4 and HX6. This is mainly due to the overestimation of the frame weight: when evaluating the weight of the latter, full volumes were considered. However, when printing in 3D, the so-called honeycomb technique is used. As explained in this article [67], the honeycomb structure is widely used in 3D printing, especially for filling. Most of the objects printed in this way are 20-30% full.
Moreover, during the calculation, a security factor S_f has been taking into account. However, the real use of this factor is to be able to guarantee the control of the drone in flight in difficult climatic conditions (strong wind): the drone must be able to produce more thrust to correct its trajectory. In this case, by rounding the maximum thrust provided by one rotor to 600g and the estimated weights for the 2

---

[34] Mycrosys' web page can be found on the following link: http://www.microsys.uliege.be/

[35] This value comes from the voltage of the battery chosen in the theoretical design (see section 1.5), which was 3S

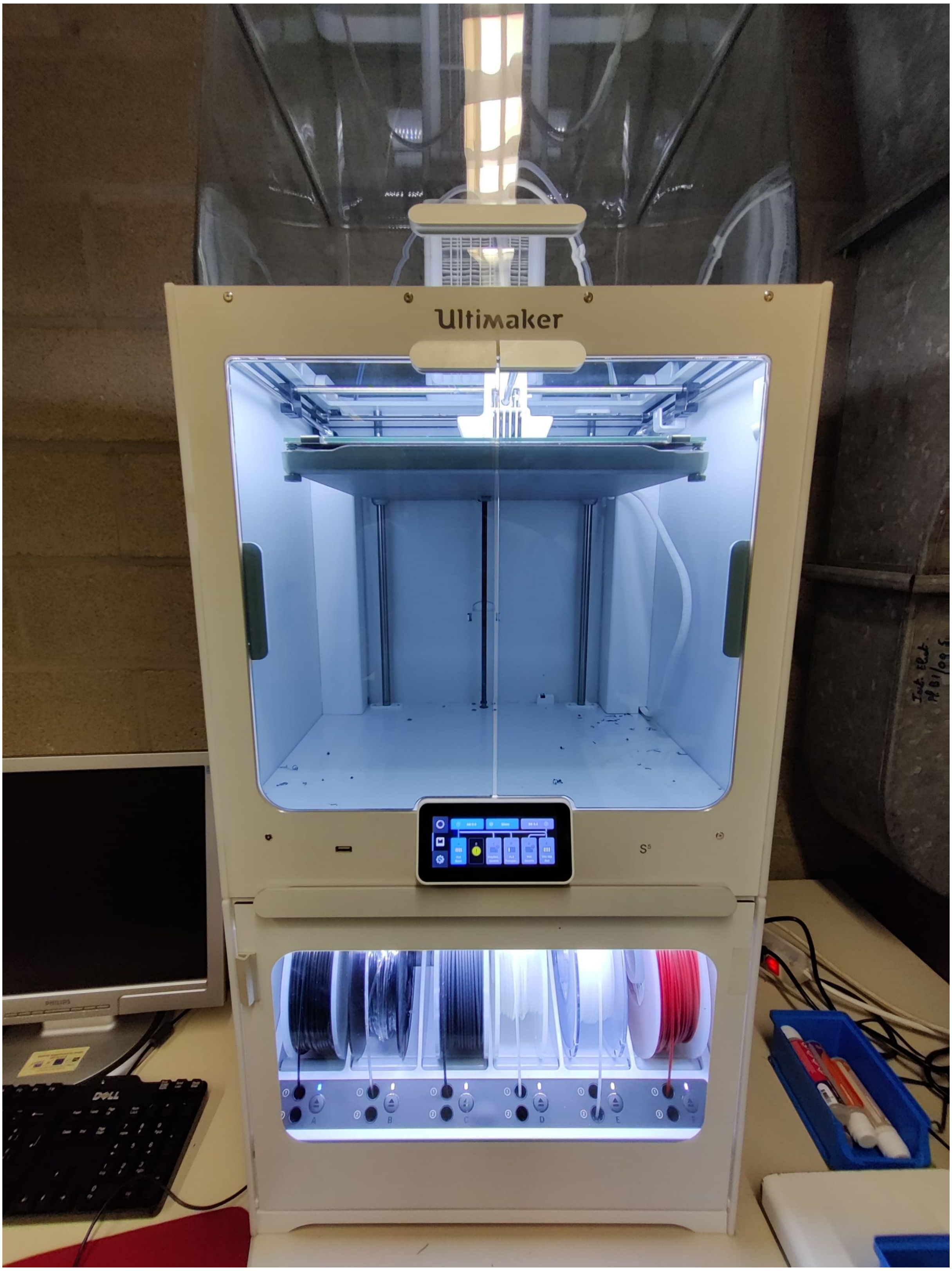


Figure 62: 3D printer, belonging to Microsys laboratory

models, there is a thrust to weight ratio $\frac{T^g_{rotor}}{W^g_{uav}} \approx \frac{600 \cdot 4}{1800} \approx 1.3$ for HX4; while for HX6, it is rather

$\frac{T^g_{rotor}}{W^g_{uav}} \approx \frac{600 \cdot 6}{2400} \approx 1.5$. Although the security factor $S_f = 2$ is not found, 1.5 is still a good value, especially if the honeycomb technique is used for the printing of the parts, because they will be lighter (so the total weight of the drone decreases, which increase the ratio T^g_{rotor}/W^g_{uav}).
Finally, by comparing the values calculated by Equations X(60) and (61), it can be seen that the conditions imposed by HX4 on $T^{(g)}_{rotor}$ are much more difficult to accomplish in practice than those of the HX6 model. In summary, the HX4 mode will be abandoned from the HX4/6 theoretical design. Therefore, the design of the quadcopter will not be mentioned in the following. In building the specification, the objective has therefore changed in favor of building a solitary hexacopter, without quadcopter mode. Note that it could be interesting, within the framework of another research work, to see if it is not possible to incorporate the quadcopter mode by modifying the theoretical design.

2.2.4 Power consumption: validation

As mentioned in the previous point, the theoretical design has evolved towards the building of an hexacopter (i.e. HX6 alone). In the following, in order to compute the flight time of the hexacopter, the power consumption of the motors during hoover (i.e. $T^g_{rotor} = W^g_{uav}$) is needed. Therefore, the security factor will not be considered anymore and Equations (59) and (61) becomes:

$$T^{(g)}_{all\_rotors} = W^g_{uav} = 2323.5 \text{ g for HX6} \tag{62}$$

$$T^{(g)}_{rotor} = \frac{T^{(g)}_{all\_rotors}}{6} = \frac{2323.5}{6} = 387.25 \text{ g for HX6} \tag{63}$$

This time, the value of the thrust required by one rotor, $T^{(g)}_{rotor}$, is achievable in practice. By looking again in Table 10, it corresponds to a duty cycle of 75% and a power consumption of 57W. It means that in practice, the hexadrone will not take off as long as the throttle joystick does not reach 75% of its amplitude. It means that it remains only 25% of the joystick freedom to control the drone's altitude. It can certainly be used in practice but it is not the most enviable situation. Building a frame lighter than the one modelled in the theoretical design is then critical, since it will allow to use less thrust to make the drone glider, yielding to a smaller duty cycle and therefore a better control for the altitude.

2.2.5 Flight time: computation

Knowing the capacity of the battery, C_{bat}, the battery voltage, V_b, and the consumed power, the flight time, t_{fly}, can finally be computed using Equation (49):

$$t_{fly} = \frac{C_{bat} \cdot V_b}{N_{motor} \cdot P_{motor}} \cdot t_{h\_to\_min} = \frac{5 \cdot 11.1}{6 \cdot 57} \cdot 60 = 9.73\ min = 9\ min\ 44\ s. \tag{64}$$

2.3 Conclusion

In conclusion, the aim is to build a hexadrone, whose frame will be made out of PLA (using a 3D printer). The goal is to reach a value close to 700g for the whole frame, resulting in a hexadrone weighing 2kg in its entirety. This would be a nice target and would require some optimisation of the parts, regarding the mechanical design. Thus built, the hexacopter should be able to fly during 9min44, while lifting a payload of 300g.

3 Upgraded design

In this part, the different upgrades brought to the theoretical design until its final form will be briefly summarized. In addition, the 3D machined parts will also be illustrated. The technical drawing and the machining of the mechanical parts being mainly the part of the work for which Luca was responsible, please refer to his thesis in order to have more explanations on some points [68].

3.1 Arms evolution

The arms were challenging in terms of weight and fixation because the size assigned to them in the theoretical design was too small. In order to avoid adjacent rotors colliding or influencing each other [69], it was decided to fix the size of an arm at 25cm (previously, 15cm in theoretical design), which increases the weight of the arms by 2/3. In order to respect the 700g target for the frame, it was agreed to use carbon fiber tubes, as illustrated in Figure 63, to make the arms in order to gain weight and rigidity. However, since they are tubes, it will be necessary to foresee fixing parts to fix the tube to the motor (see Section 3.3) and to the main body of the drone. Moreover, the ESC will not be able to be attached on the arm and will have to be fixed on one of the two plates.

Figure 63: Carbon fiber tube used to make the arms of the drone. They will have to fit into the other parts.

Sources: https://hobbyking.com/fr_fr/carbon-tube-50cm.html?___store=fr_fr

3.2 Top and bottom plates evolution

It was first decided to combine the 2 plates into a large closed box that would contain all the electronic components. In order for the box to correctly hold the tubes representing the arms (see Section 3.1), a clamping ring system was imagined, as illustrated in Figure 64.

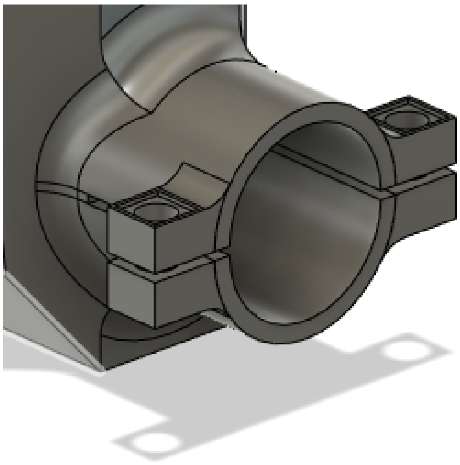


Figure 64: Clamping ring system to tighten the arm into the main body of the drone.

In order to be able to tighten this ring, the box had to be cut in two parts for its 3D printing, as illustrated in Figure 65. In addition, a groove had to be made all along the part so that the 2 parts could fit together.

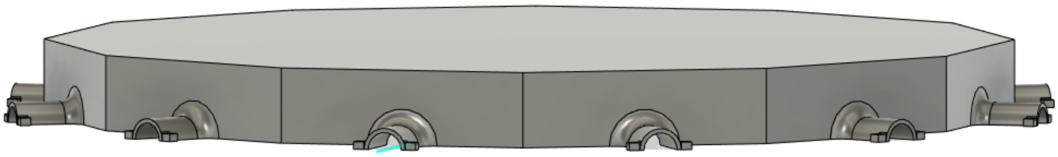


Figure 65: Box cut into 2 parts for its 3D printing.

However, these 2 parts were challenging in terms of realization because the 3D printer is limited when it comes to the size of the parts it can print: they had to be cut into 2 half-plates each. The final half-plate 3D printed is illustrated in Figure 66.

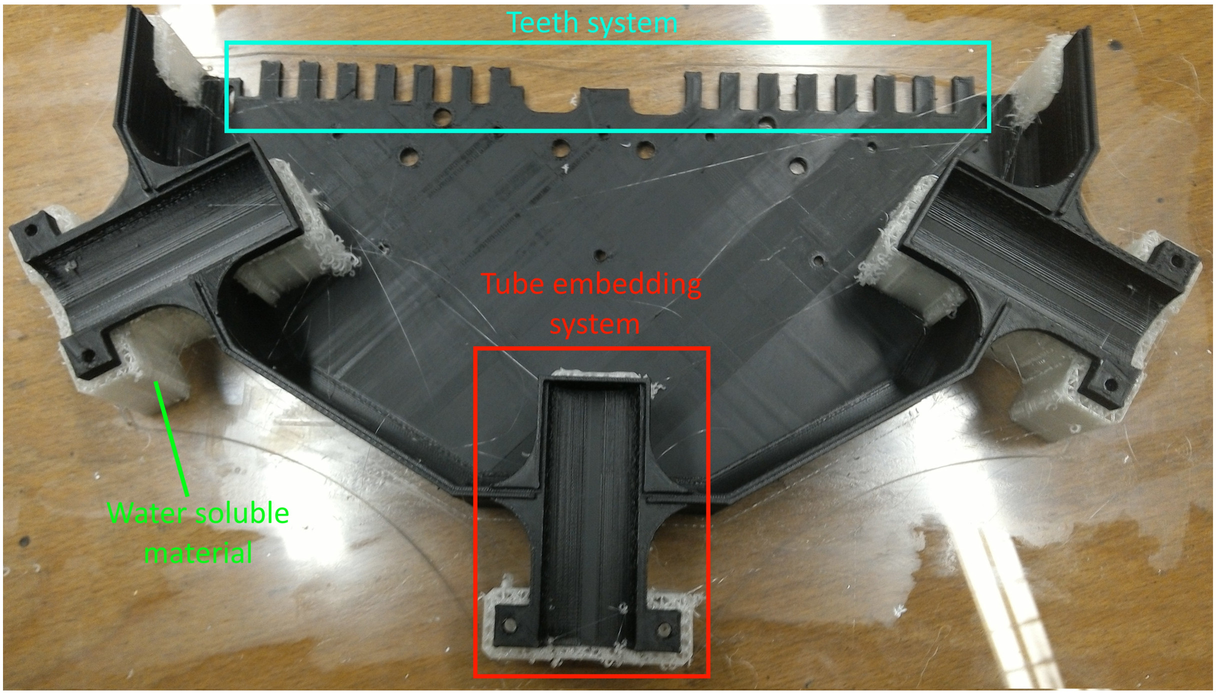


Figure 66: Final half top plate 3D printed. For a complete drone, 4 parts like this one were needed.

A system of teeth was set up to interlock the 2 half plates, as illustrated in Figure 67.

Once the 2 half-plates are nested, the teeth are melted in order to get a rigidity comparable to a one-piece realization. The result can be observed in Figure 68, which represents the bottom plate, mounted on legs.

3.3 Motor mounting

The motor supports were not foreseen in the theoretical design since they result from the use of tubes for the arms (see Section 3.1). Since there is one for each motor, the part must be well optimized in order to save weight on the whole design. The Figure 69 displays the difference between the first and last design, as well as the practical realization of the motor mounting.

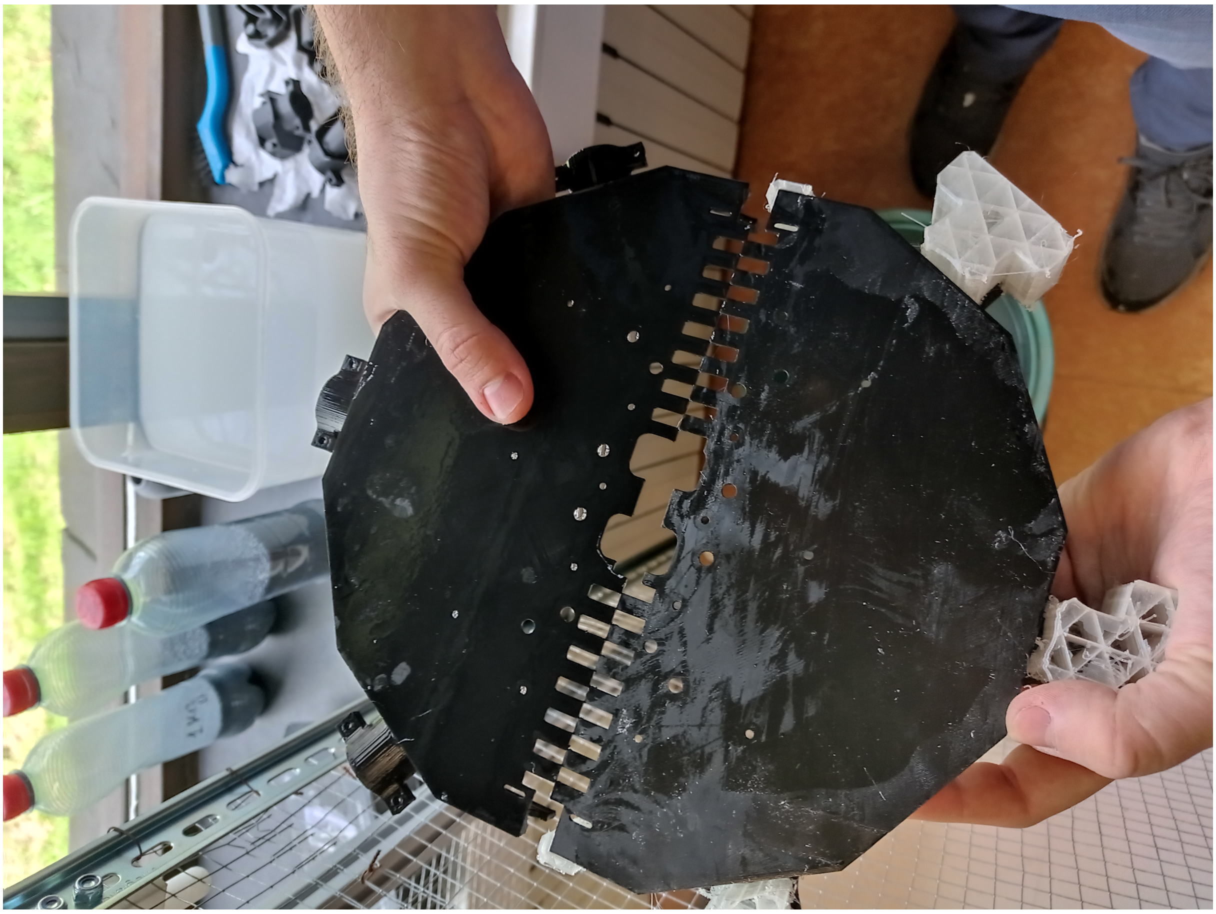


Figure 67: The system of teeth set up to interlock two half plates.

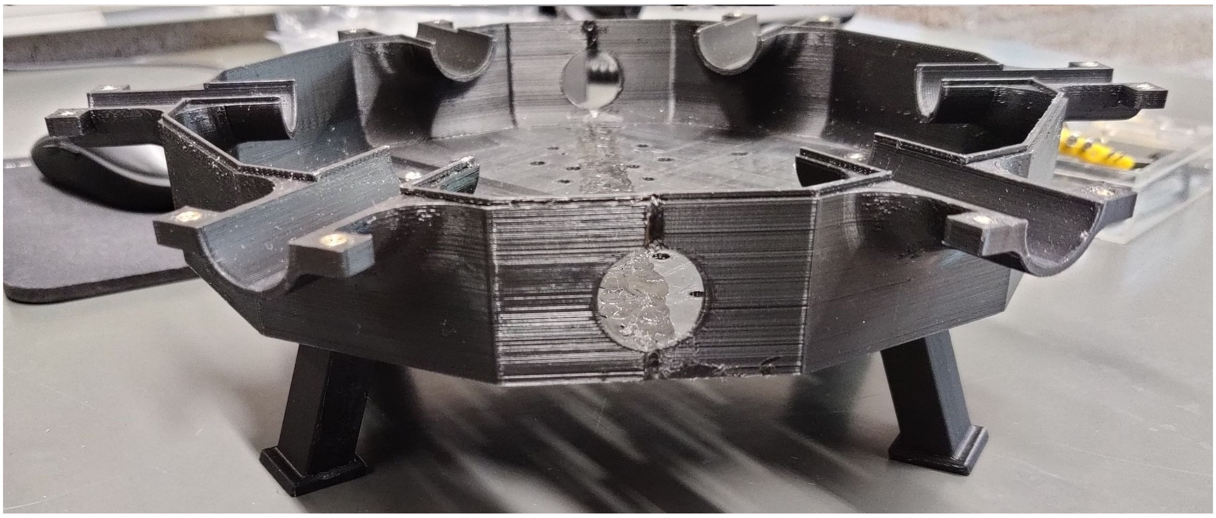
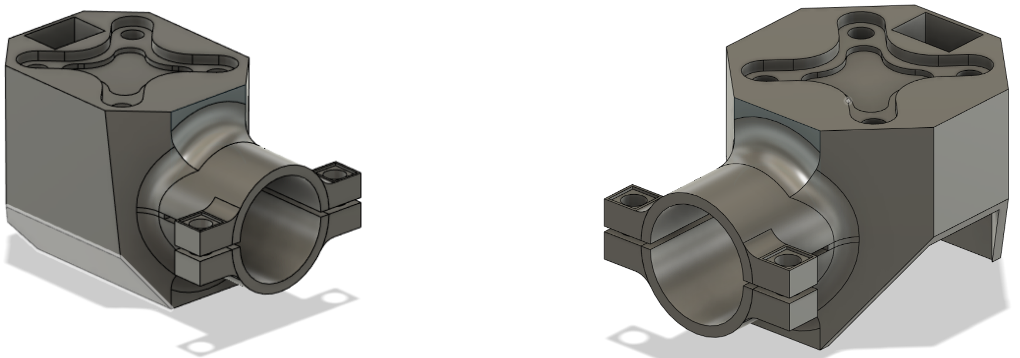


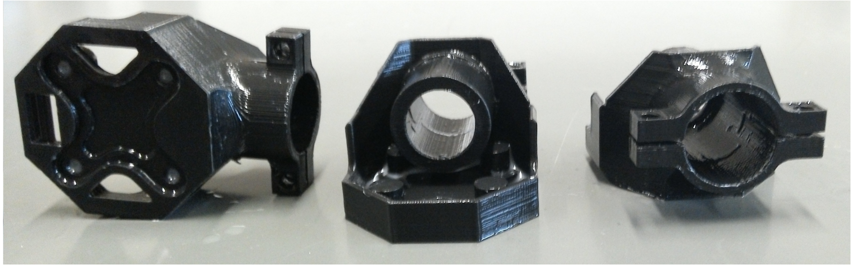
Figure 68: Final shape of a plate, once the teeth are melted. Note that feet were added to the design to improve landing.

The main challenge was the cleaning of the part once it was printed: the 3D printer added support in the hollow tube hosting the arm, in order to be able to print the upper part of the. It was therefore necessary to use a support material that could be dissolved in water because it was difficult to access this area to remove the support added by the 3D printer. It is important to know that the support material



(b) Last design

(a) Initial design



(c) Practical realisation

Figure 69: Evolution of the design of the motor mounting towards its practical realisation.

dissolves very well in hot water, but a too hot water also deforms the plastic. Figure 70 shows the motor mounting having a good bath.

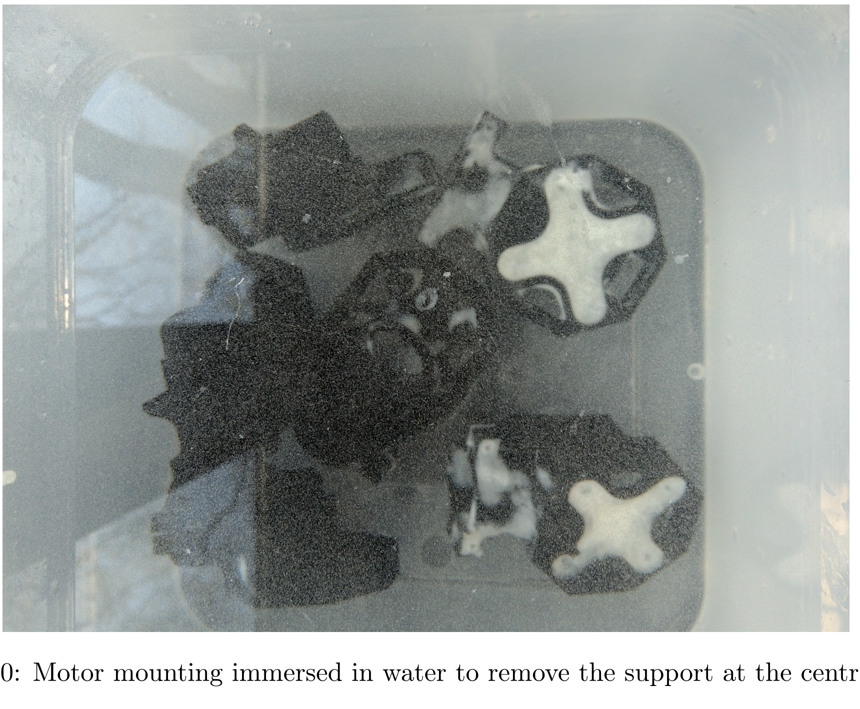


Figure 70: Motor mounting immersed in water to remove the support at the central tube.

3.4 Final form

In this section the different assembly steps will be illustrated until arriving at the fully assembled drone.

The first step is to assemble the motors on the motor mounts, as illustrated in Figure 71.

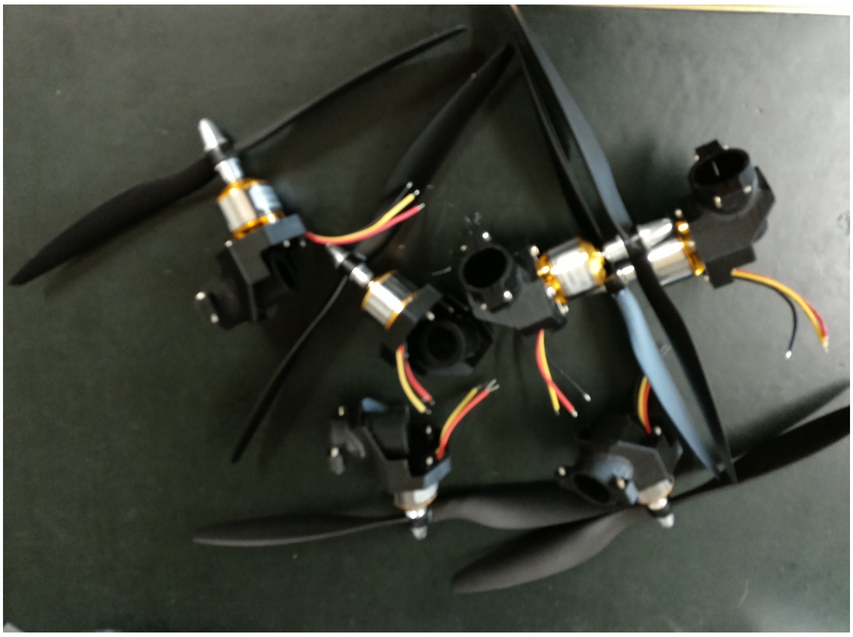


Figure 71: The first step of the assembly: fixing motors on motors mounting.

The second step of the assembly consists in equipping the upper and lower plates with the electronic material, before closing the box. In our case, the lower plate is equipped with the PX4 mini, the radio receiver and the servo motors (see Section 3.5), while the PDB and the ESCs are suspended from the upper plate with velcro strip. The battery and the GPS will be put on top of the drone, external to the box. The equipment attached to upper and lower plate are illustrated in Figure 72.

The third and last step of the assembly is to tighten the arm into the motor mount and the main base, thanks to the clamping ring. Note that you have to be careful to align the motor correctly upwards (thanks to an oriented slot). This step is illustrated in Figure 73.

The Figure 74 represents the hexacopter, once it is fully built. Note that it is quite impressive by its size.

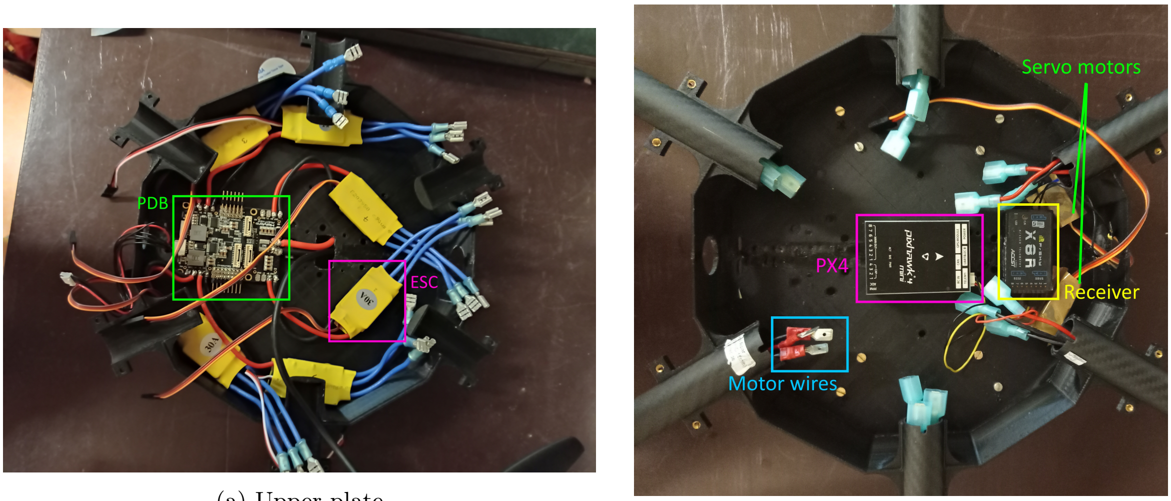
3.5 Add-on feature

In order to add some electronics, it was decided to add a ball dropping system to the drone. The idea is the following: a turret, fixed under the drone, stores the balls while the drone is in flight. This turret has a flap that can be locked with a screw. The balls are blocked at the end of the turret by the servo-motors. These motors that can be activated at any time to free the passage and thus release the balls. This feature is illustrated in Figure 75.

3.5.1 Button vs switch

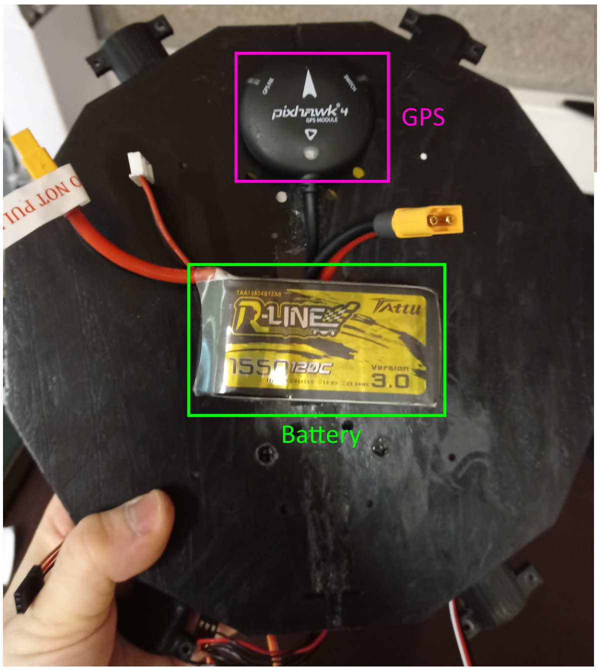
The servo motor will be controlled by an auxiliary PWM output of the PX4 mini(see Section 4.1). It will be the remote control that will send messages to the PX4 mini to modify this output accordingly. To send these control messages, there is a choice between using a switch or a button on the remote. However, as shown in the video in Appendix A.1.6, it is better to use a switch because:

- When flying a drone, your eyes are riveted in the air, so it's easier to activate a switch than a button,



(a) Upper plate

(b) Lower plate



(c) External to the box

Figure 72: The second step of the assembly: equipping plates with electronics.

- The switch allows you to switch from the closed to the open position in 1 flip. The release operation is therefore faster than with a button.

Note that the 2 servo motors will be powered by the battery eliminator circuit (BEC) of the ESCs. The BECs are voltage regulators that convert main LiPo battery voltage to lower voltage (14.8 to 5V here).

3.6 Schematic representation of the hexadrone

In order to better represent the hexadrone with all its improvements, a schematic representation is shown in Figure 76.

Each element is represented by a different color, as indicated by the legend. By looking closely at the Figure 76, it is possible to notice numbers associated to each part. In fact, every component was ordered

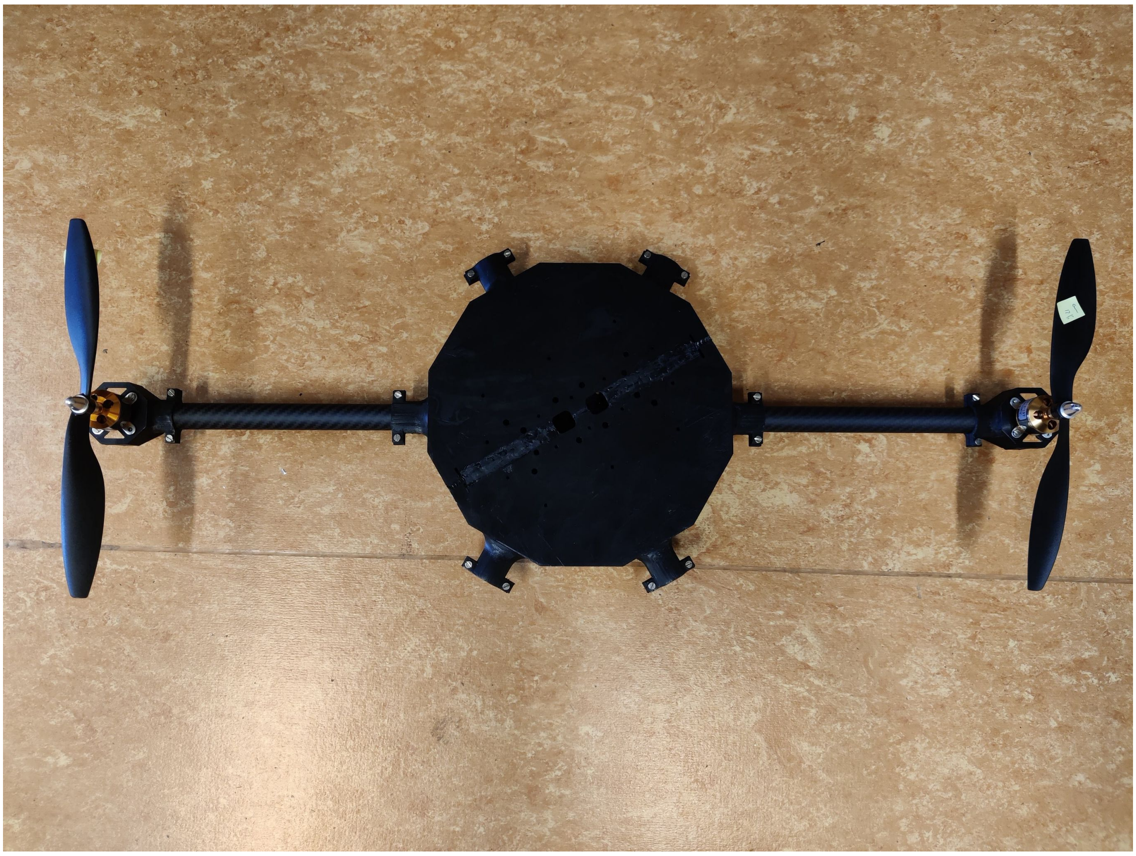


Figure 73: The third step of the assembly: fixing motors mounting on main base, by the means of arms.

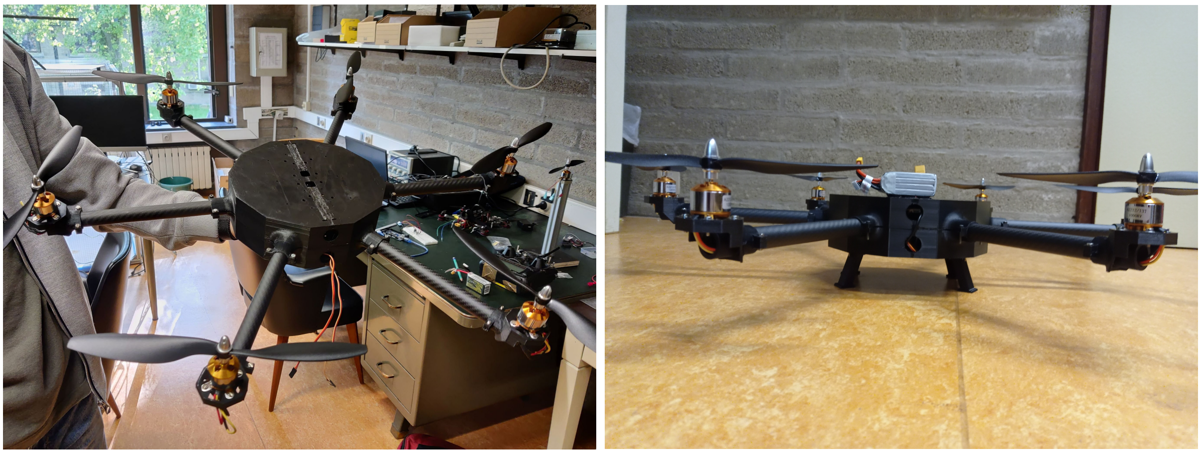


Figure 74: The hexacopter totally built.

with a small safety factor, to be sure to have replacement material if a component should fail[36]. Therefore, the numbers allow to identify the component of the batch that was selected in the final design.

3.6.1 Selection of the component within the batch

The easiest way to select a component from the batch is to look at the weight. Indeed, the lighter the drone will be, the better its performance will be. Every gram counts so it is worth optimizing the weight of the electronic components when selecting. The weights of all the components received are indicated on

---

[36] Fortunately, there were reserves because an ESC and a rotor had to be replaced after an accident.

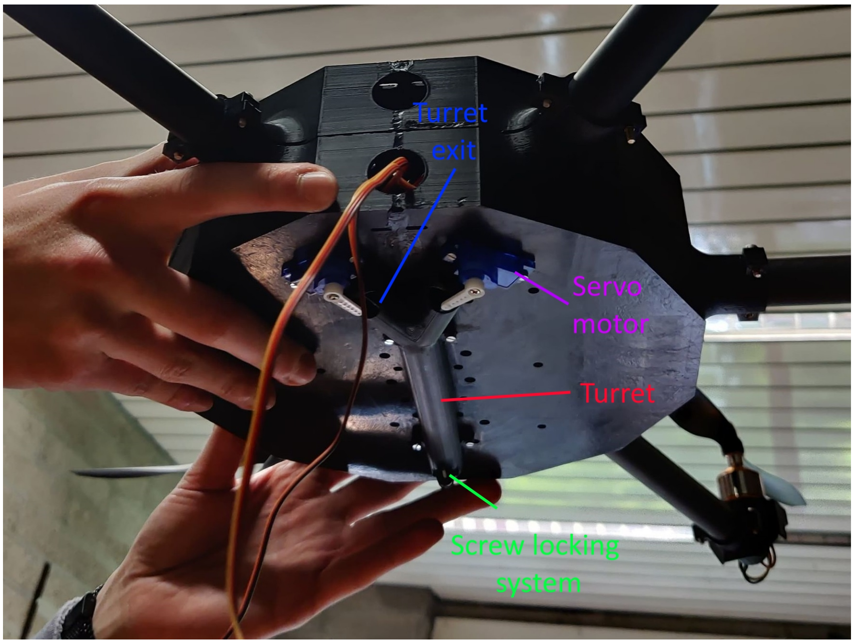


Figure 75: Add-on feature added to the hexacopter: turret that can release balls during flight.

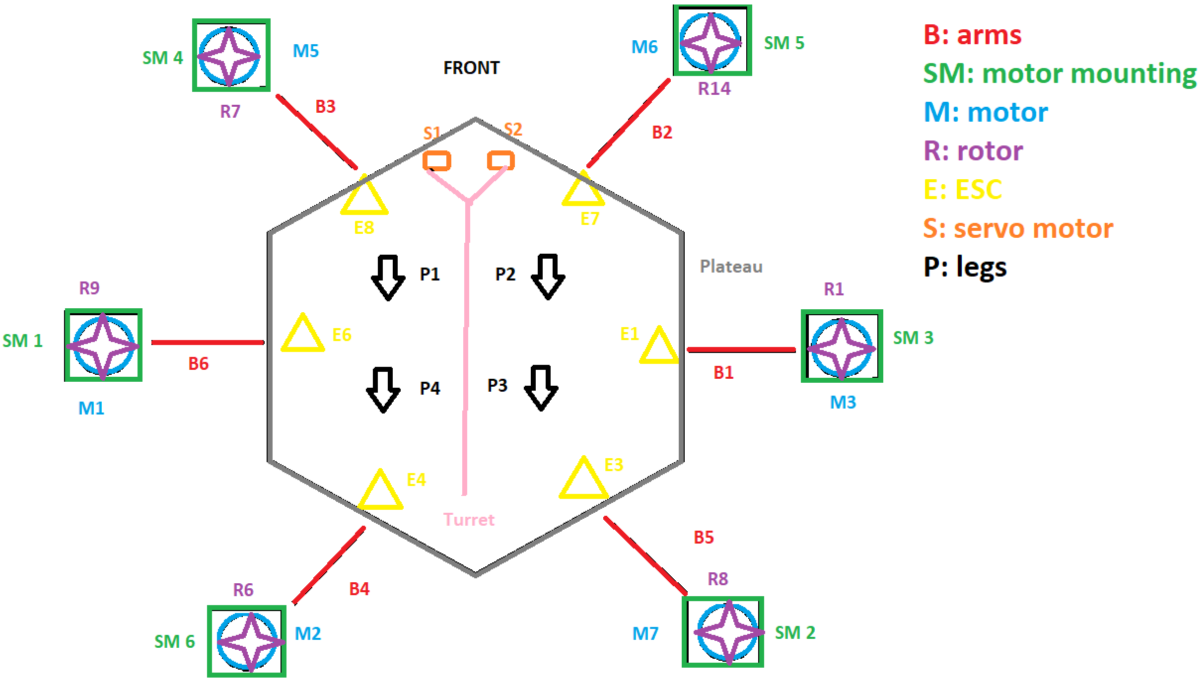


Figure 76: Schematic representing the hexadrone in its integrity. Colors indicate similar elements.

the Google spreadsheet found in the Appendix A.2.2. Each component is sorted according to a column, so it is easy to make the link between the schema and the weights associated with the components.[37]. Once the weight of all the components was measured, it was then easy to calculate the weight associated with the frame and the electronics and to check if the target was reached.

---

[37] Note that the components used in the final design are framed.

3.6.2 Weight of the frame

The frame is composed of the following elements:

- Plates: a top plate + a bottom plate + 12 inserts + 12 screws accounting for 412.91g,
- Arms: 6 arms accounting for 137.96g,
- Motor mounting: 6 motor mountings + 36 screws + 36 inserts accounting for 132.7g,
- Legs: 4 legs + 8bolts + 8 screws accounting for 33.06g.

The frame has a total weight of 732.49g, which is very close to the target from Section 2.3.

3.6.3 Weight of the welding wires

As noted in Section 3.1, the fact of using tubular arms did not allow to fix the ESCs on it. Thus, since it was necessary to put them inside the box, long wires running inside the tubular arm had to be soldered to make the connection between the motor located at the end of the arm and the esc located at the other end. As a reminder, the arms are approximately 25 cm each. Since each ESC-motor combination requires 3 wires, the length of the welding wires is: $3cdot6cdot25 = 4.5$ m, which adds a significant weight to the drone. Weighing the arms with the welding wires and removing the weight of all the elements except the unknown wires, the total weight is approximately 133.77g.

3.6.4 Weight of the electronics and other

The electronics composed of the following elements:

- Motor: 6 motors + 6 crosses + 24 screws accounting for 309.86g,
- Rotor: 6 rotors + 6 supports accounting for 79.93g,
- ESC: 6 ESCs + welding wires + connectors accounting for 268.28g,
- Servo motor: 2 servo motors + 4 screws + 4 bolts accounting for 22.32g.

The components categorized as "other" are the following:

- Battery: 177.12g,
- GPS fixed with glue: 33.29g,
- PX4 mini fixed with velcro: 38.33g,
- Receiver and cable: 16.5g,
- PDB and power cable: 40.5
- Turret with balls: 14.69g.

The electronics has a total weight of 680.39g, while the "other" category represents a weight of 320.43g. Combining both, one ends up with 1000.82g. It is well below the weight that was planned in the theoretical design at Section 1.7.3. This is due to the fact that a lighter battery is used. In addition, the values used to predict the weight of the electronics came from the data sheets and were constant (however, variations were observed within the batch).

3.6.5 Total weight of the hexadrone

Adding results coming from Sections 3.6.2 and 3.6.4[38], the following value is obtained:

$$W^{(g)}_{Hexa} = 732.49 + 1000.82 = 1733.31 \tag{65}$$

---

[38] Welding wires are already included in ESCs weight

By weighing the drone with a scale, the value 1753g is obtained, which is close to the value obtained at Equation (65). Thus, the weights of the welding wires are well estimated.

4 PX4 programming

This section will describe the software changes that had to be made in the PX4 in order to pilot the drone. First, the creation of the frame will be detailed from a software point of view while the control system will be explained in a second time

4.1 PX4 firmware

Since the drone is "unique" in the world, its dynamics are also unique: it is our duty to set up the drone as we wish. Going through the list of available frames for the hexacopter (see Section 3.2.2), there was only a general airframe. Moreover, in the definition of this frame, only the first 6 PWM outputs of the PX4 mini were active (for the control of the main motors). However, two auxiliaries PWM outputs were also required to control the servo motors.

4.1.1 Mixer definition

Since some problems were encountered with the frame selection, the solution was to create a new hexacopter frame in PX4. To do this, the entire PX4 firmware code had to be downloaded. Once the open source code was downloaded, it was necessary to write a mixer file (see Appendix B.3) defining the status of the PWM outputs of the PX4 mini controller[39]. Knowing that PX4 mini outputs are assigned in the order of writing within the mixer file, the first line of the file specifies, with the help of an "R" tag, that 6 MAIN PWM outputs are desired. From then on, outputs 1 to 6 of the PX4 mini will be defined as PWM MAIN and can be used to control our 6 motors. The following lines of code define outputs 7 and 8 of the PX4 mini as auxiliary. They can therefore be used to control our servo motors.

4.1.2 Airframe definition

Now that the mixer file is ready, it must be integrated into an airframe file (see Appendix B.2)[40]. The mixer file created in the previous step is integrated into the airframe using the keyword "set MIXER "followed by the mixer name. Next, the number of PWM channels to be activated must be specified, using the keyword "PWM_OUT".

4.1.3 Compilation

Once the 2 previous steps are done, it is necessary to indicate that the mixer and airframe files must be considered during the compilation. This is done by adding their name into "cmake" file. Back in QGC, in the firmware tab, there is the possibility to load its own firmware by going to the advanced options, as illustrated in Figure 77. Be aware that all the saved parameters concerning a vehicle will be lost. So remember to save them first.

4.2 PID control

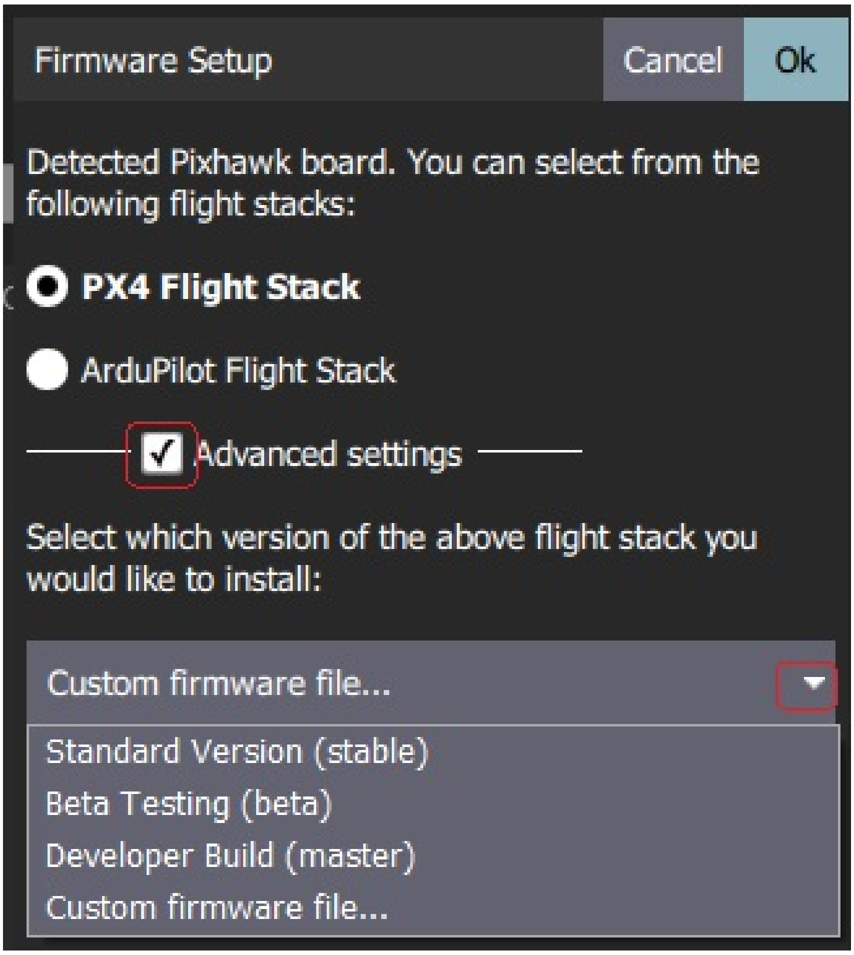
5 Performances evaluation

In this section, the flights performances of the drone will be analysed, as for the kit. This time, since our motors are well known, there is no need to go through an experimental phase again. We will thus, initially, check the flight performances of the hexadrone in hovering flight before making it undergo endurance tests.

---

[39] The writing of this file was largely inspired by the mixer file of the kit which contained 2 AUX PWM but only 4 MAIN PWM.

[40] The writing of this file was largely inspired by the airframe file of the general hexacopter which had only 6 PWM output channels.



Detected Pixhawk board. You can select from the following flight stacks:

● **PX4 Flight Stack**

○ ArduPilot Flight Stack

☑ Advanced settings

Select which version of the above flight stack you would like to install:

Custom firmware file...

Figure 77: Compilation of the custom PX4 firmware, containing the new hexacopter airframe.

5.1 Flights

The drone has made several test flights. Indeed, it was very difficult to tune the control system. Indeed, since no telemetry kit was available, the autotuning function, which allows to see in real time the curves related to the different PID, could not be used. The work was thus carried out in "trial and error" mode. The parameters are changed, a flight is made, instabilities are noticed, the parameters are modified to try to stabilize the hexadrone.

The first conclusive test that appeared corresponds to the video in Appendix A.1.7. In this video, the drone reacts correctly to the instructions given to it, via the remote control. A stabilization phase is also observed (00:53 to 1:16) during which the drone remains quite stable. However, it flies quite close to the ground so it suffers the full force of the wind column that its rotors generate.

The second notable success is illustrated by the video in the Appendix A.1.8. Between the 2 videos some changes have been applied to improve the overall behavior of the drone:

- Decrease of max horizontal speed (10 to 1 m/s)
- Decrease of max vertical speed (3 to 1 m/s)
- Increased drone response time
- Changing the foam around the pixhawk to reinforce its insulation against external vibrations

The first two changes were intended to make the drone easier to drive. Indeed, since it is much bigger than the kit, it is also less lively. In order to avoid that it gets carried away in all directions and to have time to react if it behaves badly, the overall speed has been decreased. This allows, among other things, to make its movement phases more stable (less oscillations on the rolling axis). The third change is there to counteract the effect of the first 2 changes. Indeed, by decreasing the overall speed of the drone, it makes it much more difficult to recover from a rapid external disturbance. By slightly increasing the response time, overshoot is avoided while reaching the target value of the control system faster. As for the fourth change, it is there to try to reduce as much as possible the errors introduced by the vibrations of the drone on its rolling axis.

However, there was not always success when trying to fly the hexadrone as illustrated by the video in Appendix A.1.9. In this video, the drone turns on itself due to an tilted motor. There is a thrust component in the horizontal plane, while ideally the thrust should only be vertical.

The strangest behavior that could be observed is illustrated by the video located in Appendix A.1.10. After a certain time, the drone describes circles on itself, whose amplitude increases with time. This problem still persists today. Some hypotheses have been discarded, but there are still tracks to explore:

- Defective sensor: it is likely that the gyroscope is defective, making the drone unable to locate itself in space. However, the drone remains fixed in a direction while turning, as guided by the compass,
- Comparison between the data returned by the compass magnetometer and that of the PX4 mini.
- Bad regulation: a torque is created when all thrust should be equal in hovering flight, which induces rotation.

This could be a very interesting research topic whose goal is to understand what is wrong and to correct it in order to make the hexadrone stable again.

5.2 Flight time prediction

The endurance tests were somewhat disturbed due to the behaviors described above. Only one measurement, judged to be correct, was obtained in practice. This is illustrated in Table 26

Kit flight time tests				
Test number	GPS lock time [min]	Flight time [min]	battery remaining at the end [%]	Weather
1	-	04:51	2	Blue sky Small wind

Table 26: Flight times obtained for the hexadrone, in hovering.

From Equation (65), the hexadrone weights at rest 1733.31g. Since it has 6 motors, a single motor should generate a thrust of $1733.31/6 = 288.885g$ to compensate exactly the weight of the hexadrone. Looking at the experimental values obtained for the motor (see Table 11), it corresponds to a power consumption $\approx 38.2W$, using linear interpolation again.

From a mathematical point of view, using Equation (49), the expected flight time is equals to:

$$t_{fly} = \frac{C_b}{V_b} N_{motor} P_{motor} \cdot t_{h\_to\_min} = \frac{1.55 \cdot 14.8}{6 \cdot 38.2} \cdot 60 = 6min \tag{66}$$

Comparing this result with the one obtained in practice, we see that there is 1 minute difference. This could be explained by the fact that the motors consumed more than expected because they had to supply the servo motors continuously. To be sure, it would be necessary to start again the experiment without the servo motors.

Part VI

Conclusion

To conclude, I would like to express my personal feelings about the past year. It was a great year, full of problems but also solutions. It was really nice to work in collaboration with Luca. I could realize that our association allowed us to go further than if we were each on our own. I was able to bring the scientific and theoretical side to the work, while Luca was able to bring with him all his practical expertise. Luca has become more than a colleague, he is a true friend. Concerning the results obtained, I am obviously happy to have seen the drone take off but I remain a little disappointed because I think we were really close to make it completely stable. To finish, I would like to thank once again all the people who helped me in this life experience, it was crazy! If anyone wants to continue my work, I would feel really proud.

Part VII

Acknowledgements

Finally, I would like to warmly thank all the people who helped build this project, and made it a success.

Mr. BOURGUIGNON Denis,
Mr. CALDERON JIMENEZ Angel,
Mr. DELFINO Luca,
Mr. DIGREGORIO Gabriel,
Mr. GREFFE Christophe,
Mr. HARMEL Bernard,
Mrs. HARMEL Marie,
Mrs. HENEAUX Camille,
Mrs. HENEAUX Emilie,
Mr. HEUCHAMPS Alexandre,
Mr. GREFFE Christophe,
Mr. LORENZETTI Maxime,
Mrs. LALLEMAND Pascale,
Mr. PIROTTIN Thomas,
Pr. REDOUTÉ Jean-Michel,
Mr. SHEHABI Ammar,

Thank you very much !

Part VIII

Appendices

A Links

This Appendix contains links relative to videos, Google spreadsheets or datasheets.

A.1 Video links

The following links will redirect you to Youtube, to the videos shot during the year.

A.1.1 Presentation of the test bench

https://youtu.be/I0TV4kN3Hko

A.1.2 Test the motors from learning kit

https://youtube.com/shorts/yt4Gon-vLmM

A.1.3 Paradrone

https://youtu.be/POomz-50J3w

A.1.4 Flight time test

https://youtu.be/e6_7C3uhiuo

A.1.5 Live demo: kit

https://youtu.be/MOotC2y3h48

A.1.6 Servo motor control

https://youtu.be/ykBTLEoaLko

A.1.7 Hexadrone flight test

https://youtu.be/Q7kESmxUzo0

A.1.8 Hexadrone flight test: limited vibrations and speed

https://youtu.be/ffv1fNyTdUA

A.1.9 Hexadrone spinning

https://youtu.be/RnKr72v1IoU

A.1.10 Hexadrone circling

https://youtu.be/PrTNp8TMJW8

A.2 Spreadsheet links

The following links will redirect you to Google Spreadsheet, to the data tables completed during the year.

A.2.1 Experimental measurements on motors

https://docs.google.com/spreadsheets/d/11o8I9wjfHaVnF6gPZx-bqxkwZHyNzrxf54wE90KuQZY/edit?usp=sharing

A.2.2 Hexadrone: weight of the different parts

https://docs.google.com/spreadsheets/d/1xbBXTn65oyi8DlET7SINfh6IgsYKP9GxBvHvKyQKa2k/edit?usp=sharing

A.3 Datasheet links

The following links will redirect you either to downloadable pdf's or to web pages containing technical information about the compounds handled during the year.

A.3.1 A2212 motor

https://www.rhydolabz.com/documents/26/BLDC_A2212_13T.pdf

A.3.2 HW30A ESC

https://www.optimusdigital.ro/index.php?controller=attachment&id_attachment=451

A.3.3 KERN High precision scale

https://dok.kern-sohn.com/downloads/de/EMB%202000-2/file/EMB-BA-e-1636.pdf

A.3.4 UT373 digital tachometer

https://supereyes.ru/img/instructions/UT373%20English%20Manual.pdf

A.3.5 1045 Rotor

https://www.daraz.com.bd/products/1-pair-multi-rotor-fan-glass-fiber-nylon-1045-1045r-1045-inch-motor-fan-bla-de-i178911050.html

A.3.6 PM07 Power distribution board

https://docs.px4.io/v1.12/en/power_module/holybro_pm07_pixhawk4_power_module.html

A.3.7 Traxxas battery

https://www.mcmracing.com/fr/home/134831-traxxas-trx2832x-power-cell-lipo-5000mah-111v-3s-25c-short-135mm-020334283290

A.3.8 PX4 Flight controller

https://docs.px4.io/v1.12/en/flight_controller/pixhawk4_mini.html

A.3.9 PX4 GPS module

https://www.reichelt.com/index.html?ACTION=7&LA=3&OPEN=0&INDEX=0&FILENAME=X200%2FPIXHAWK4-GPS-QUICK-START-GUIDE.pdf

A.3.10 QAV 250 kit

https://www.studiosport.fr/complete-kit-diy-qav250-avec-pixhawk-4-mini-holybro-a20876.html

A.3.11 2205-2300KV motor

https://www.hobbywing.com/products/enpdf/2205-2300KV.pdf

A.3.12 PM06-V2 PDB

http://www.holybro.com/product/micro-power-module-pm06/

A.3.13 BLHeli S 20A ESC

https://www.flyingtech.co.uk/electronics/holybro-blheli-s-esc-20a-2-4s-speed-controller

A.3.14 Pixhawk 4 mini

https://docs.px4.io/v1.12/en/flight_controller/pixhawk4_mini.html

A.3.15 GPS module PX4

https://www.reichelt.com/index.html?ACTION=7&LA=3&OPEN=0&INDEX=0&FILENAME=X200%2FPIXHAWK4
GPS-QUICK-START-GUIDE.pdf

A.3.16 Foxeer micro FPV camera

https://download.foxeer.com/Razer_Cam_Manual.pdf

A.4 Atlatl hv video transmitter

http://www.holybro.com/manual/Holybro-Atlatl-HV-v2-Manual.pdf

A.5 Micro OSD

http://www.holybro.com/manual/Holybro_Micro_OSD_V2_Manual.pdf

A.6 Qav 250 build

http://www.holybro.com/manual/Pixhawk4Mini_QAV250_Kit_QuickStartGuide.pdf

A.7 Tattu 4S LiPo battery

https://www.studiosport.fr/batterie-lipo-4s-1550-mah-120c-r-line-v3-tattu-a17666.html

A.8 FrSky X8R receiver

https://www.frsky-rc.com/wp-content/uploads/2017/07/Manual/X8R.pdf

A.9 Tanaris X9 remote control

https://www.frsky-rc.com/wp-content/uploads/Downloads/Manual/X9%20Lite/X9%20Lite-%20manual.pdf

B Computer codes

The following points will include the original source codes written and used during the year.

B.1 Controlling a motor with an ESC

```
#Import useful lib
import time
import RPi.GPIO as GPIO

#Params
FREQUENCY = 50  #Control frequency: 50 Hz == 20 ms
DUTY_MIN = 5 # 1000us = 1ms
DUTY_MAX = 10 # 2000us = 2ms
DUTY_INC = 0.05  #10us = 0.01ms => range = (DUTY_MAX-DUTY_MIN)/DUTY_INC = 100

def calibration():
    print("You selected calibration option")
    pwm.start(0)
    print("Disconnect ESC power then press enter")
    inp = input()
    if inp == ’’:


        print("PWM signal, duty = {}, frequency = {}".format(DUTY_MAX, FREQUENCY))
        print("Connect battery, wait 1 beeps and press Enter")
        pwm.start(DUTY_MAX)
        inp = input()
        if inp == ’’:

            print("PWM signal, duty = {}, frequency = {}".format(DUTY_MIN, FREQUENCY))
            print("Press Enter when all beeps finished")
            pwm.start(DUTY_MIN)
            inp = input()
            if inp == ’’:
                print("Successfully calibrated")

def manual():
    print("You selected manual option")
    pwm.start(0)
    print("Select duty cycle between 0.0 and 100.0")
    inp = input()
    duty_user = float(inp)
    if(0<= duty_user and duty_user <= 100):
        DUTY_ESC = DUTY_MIN + duty_user*DUTY_INC
        print("Starting PWM signal, duty = {}, frequency = {}\nType Ctrl+C to stop".format(DUTY_ESC, FREQUENCY))
        try:
            while(True):
                pwm.start(DUTY_ESC)

        except KeyboardInterrupt:
            print("PWM signal stopped by user")
            pwm.start(0)
            print("Type c to continue, q to quit")
            inp = input()
            if(inp == ’c’):
                print("Continuing manual mode")
                manual()

            elif(inp == ’q’):
                print("Quiting manual mode...back to main menu")
                pwm.stop()


            else:
                print("Unknown command .. quiting")
                pwm.stop()
                stop()
```

```python
 else:
        print("Error: duty must be float number between 0 and 100. You typed {}".format(duty_user))

def stop():
    print("You selected stop option")
    pwm.stop()
    #Safely exit program
    GPIO.cleanup()
    print("Quitting and cleanup")

##### MAIN #####
print("Initialisation of the pins and creation of pwm object")
#Defining numbering system used by PI
GPIO.setmode(GPIO.BOARD)  #We use physical pin num
configuration = GPIO.getmode()
if(configuration == 10):
    print("PI is using the numerotation: BOARD")
    control_pin = 12
    print("Control signal will be generated on pin {}".format(control_pin))

elif(configuration == 11):
    print("PI is using the numerotation: BCM")
    control_pin = 18
    print("Control signal will be generated on pin {}".format(control_pin))

else:
    print("Error: unknown numerotation used")
    exit(1)

#I/O initialization
GPIO.setup(control_pin, GPIO.OUT, initial=GPIO.LOW)  #output pin, starting with low
control_initial_value = GPIO.input(control_pin)
print("Control signal initialised at value {}".format(control_initial_value))

#Create PWM object
pwm = GPIO.PWM(control_pin, FREQUENCY)

while(True):
    print("Select one of the following options by typing it correctly in the terminal: ")
    print("calibration or manual or stop:")
    inp = input()

    if(inp == "calibration"):
        calibration()

    elif(inp == "manual"):
        manual()

    elif(inp == "stop"):
        stop()
        break

    else:
        print("Error: unknown command ... program will quit")
        stop()
        break
```

Listing 1: Python code for motor control (see 2)

B.2 Definition of an airframe for the hexacopter in PX4 firmware

`data.txt`

```sh
#!/bin/sh
#
# @name MY HEXA
#
# @type Hexarotor x
# @class Copter
```

```
#
# @output MAIN1 motor 1
# @output MAIN2 motor 2
# @output MAIN3 motor 3
# @output MAIN4 motor 4
# @output MAIN5 motor 5
# @output MAIN6 motor 6
#
# @output AUX1 feed-through of RC AUX1 channel
# @output AUX2 feed-through of RC AUX2 channel
# @output AUX3 feed-through of RC AUX3 channel
#
# @maintainer Lorenz Meier <lorenz@px4.io>
#
# @board bitcraze_crazyflie exclude
#

. ${R}etc/init.d/rc.mc_defaults

# MAV_TYPE_HEXAROTOR 13
param set-default MAV_TYPE 13

param set-default CA_ROTOR_COUNT 6
param set-default CA_ROTOR0_PX 0.0
param set-default CA_ROTOR0_PY 0.5
param set-default CA_ROTOR0_KM -0.05
param set-default CA_ROTOR1_PX 0.0
param set-default CA_ROTOR1_PY -0.5
param set-default CA_ROTOR2_PX 0.43
param set-default CA_ROTOR2_PY -0.25
param set-default CA_ROTOR2_KM -0.05
param set-default CA_ROTOR3_PX -0.43
param set-default CA_ROTOR3_PY 0.25
param set-default CA_ROTOR4_PX 0.43
param set-default CA_ROTOR4_PY 0.25
param set-default CA_ROTOR5_PX -0.43
param set-default CA_ROTOR5_PY -0.25
param set-default CA_ROTOR5_KM -0.05

set MIXER my_X6

# Need to set all 8 channels
set PWM_OUT 12345678
```

B.3 Definition of a mixer for the hexacopter in PX4 firmware

data.txt

```
# Hexa X

R: 6x

AUX1 Passthrough
M: 1
S: 3 5  10000  10000      0 -10000  10000

AUX2 Passthrough
M: 1
S: 3 6  10000  10000      0 -10000  10000
```

References

[1] Brussels Airport. Brussels airport et skeyes testent un drone de sécurité ainsi qu'un système de détection des drones. `https://www.brusselsairport.be/fr/pressroom/news/drone-test`. [Online; accessed 14-May-2022].

[2] Dronenerds. The evolution of drones, then and now. `https://enterprise.dronenerds.com/the-evolution-of-drones/`. [Online; accessed 14-May-2022].

[3] Wallonie recherche CRA-W. Uavsoil : Des drones pour optimiser la gestion des parcelles agricoles. `https://www.cra.wallonie.be/fr/uavsoil-des-drones-pour-optimiser-la-gestion-des-parcelles-agricoles`. [Online; accessed 14-May-2022].

[4] Wikipedia contributors. Unmanned aerial vehicle — Wikipedia, the free encyclopedia. `https://en.wikipedia.org/w/index.php?title=Unmanned_aerial_vehicle&oldid=1087218757`, 2022. [Online; accessed 14-May-2022].

[5] Eurocarbon B.V. Chopped carbon fibre. `https://www.eurocarbon.com/composite-reinforcement/standard-programs/chopped-carbon/?gclid=Cj0KCQjwyYKUBhDJARIsAMj9lkH1t9_5kgMxSpPAqdtgc1muMZEP1gtg8x4qoe9ujQdjsn3Rlcr_rxkaAtH6EALw_wcB`. [Online; accessed 15-May-2022].

[6] BCN3D Technologies. Pla filament: Pla strength, pla temperature, and benefits. `https://www.bcn3d.com/pla-filament-stands-for-strength-temp/`. [Online; accessed 15-May-2022].

[7] The Wood Database. Balsa. `https://www.wood-database.com/balsa/`. [Online; accessed 15-May-2022].

[8] The Wood Database. Cherrybark oak. `https://www.wood-database.com/cherrybark-oak/`. [Online; accessed 15-May-2022].

[9] ifconfig. How does the center of gravity affect a quadcopter? `https://drones.stackexchange.com/questions/1094/how-does-the-center-of-gravity-affect-a-quadcopter`. [Online; accessed 20-May-2022].

[10] MrReid. The difference between centre of mass and centre of gravity. `http://wordpress.mrreid.org/2014/09/12/the-difference-between-centre-of-mass-and-centre-of-gravity/`. [Online; accessed 20-May-2022].

[11] Kerbal space program contributors. Center of thrust. `https://wiki.kerbalspaceprogram.com/wiki/Center_of_thrust`. [Online; accessed 20-May-2022].

[12] Drone24Hours. Definitive guide on how to choose a frame for a drone racing fpv. `https://www.drone24hours.com/blog/how-to-choose-mini-quad-frame/?lang=en`. [Online; accessed 21-May-2022].

[13] GetFPV. All about multirotor drone fpv frames. `https://www.getfpv.com/learn/new-to-fpv/all-about-multirotor-fpv-drone-frame/`. [Online; accessed 21-May-2022].

[14] JESS VILVESTRE. This quarter-sized, self-powered drone is the smallest in the world. `https://futurism.com/this-quarter-sized-self-powered-drone-is-the-smallest-in-the-world`. [Online; accessed 21-May-2022].

[15] Admirator. Top 5 biggest drones you can buy currently on the market. `https://dronepedia.xyz/largest-quadcopters/`. [Online; accessed 21-May-2022].

[16] Ahmed N. Khan. Drone wind resistance levels [explained]. `https://flythatdrone.com/blog/drone-wind-resistance-levels-explained/`. [Online; accessed 21-May-2022].

[17] Wikipedia contributors. Aircraft principal axes — Wikipedia, the free encyclopedia. `https://en.wikipedia.org/w/index.php?title=Aircraft_principal_axes&oldid=1083531617`, 2022. [Online; accessed 21-May-2022].

[18] Emissary drones. What is pitch, roll and yaw ? `https://emissarydrones.com/what-is-roll-pitch-and-yaw`. [Online; accessed 21-May-2022].

[19] Unmanned systems technology. Uav & drone propellers overview. `https://www.unmannedsystemstechnology.com/category/supplier-directory/propulsion-power/propellers/`. [Online; accessed 21-May-2022].

[20] EPI Inc. Propeller performance factors - basic information to help select the correct propeller. `http://www.epi-eng.com/propeller_technology/selecting_a_propeller.htm`. [Online; accessed 21-May-2022].

[21] Oscar Liang. Brushed motors vs brushless motors for quadcopter. `https://oscarliang.com/brushed-vs-brushless-motor/`. [Online; accessed 21-May-2022].

[22] Wikipedia contributors. Thrust-to-weight ratio — Wikipedia, the free encyclopedia. `https://en.wikipedia.org/w/index.php?title=Thrust-to-weight_ratio&oldid=1070319065`, 2022. [Online; accessed 22-May-2022].

[23] Drone nodes. Drone motor fundamentals – how brushless motor works. `https://dronenodes.com/drone-motors-brushless-guide/`. [Online; accessed 22-May-2022].

[24] Charles Blouin - Tyto Robotics. How to measure brushless motor and propeller efficiency. `https://www.tytorobotics.com/blogs/articles/how-to-measure-brushless-motor-and-propeller-efficiency`. [Online; accessed 22-May-2022].

[25] John Reid - Rotor drone pro. Understanding kv ratings. `https://www.rotordronepro.com/understanding-kv-ratings/#outer-popup`. [Online; accessed 22-May-2022].

[26] Eric - Learnign RC. Brushless motor kv constant explained. `http://learningrc.com/motor-kv/`, 2015. [Online; accessed 22-May-2022].

[27] Wikipedia contributors. Motor constants — Wikipedia, the free encyclopedia. `https://en.wikipedia.org/w/index.php?title=Motor_constants&oldid=1084875237`, 2022. [Online; accessed 22-May-2022].

[28] Dejan - How to mechatronics. How brushless motor and esc work. `https://howtomechatronics.com/how-it-works/how-brushless-motor-and-esc-work/`. [Online; accessed 23-May-2022].

[29] Tarun Agarwal- ElProCus. What is electronic speed control (esc) & its working. `https://www.elprocus.com/electronic-speed-control-esc-working-applications/`. [Online; accessed 22-May-2022].

[30] Oscar Liang. Lihv (4.35v) vs lipo (4.20v) | battery for multirotor. `https://oscarliang.com/lihv-lipo-drone-battery-hvli/`. [Online; accessed 23-May-2022].

[31] Jacob B - StackExchange. Why do lihv batteries degrade faster than an equivalent lipo? `https://drones.stackexchange.com/questions/520/why-do-lihv-batteries-degrade-faster-than-an-equivalent-lipo`. [Online; accessed 23-May-2022].

[32] Korishan - Second Life Storage. What does s and p stand for? what is xs, yp or xsyp with x and y being numeric digits? `https://secondlifestorage.com/index.php?threads/what-does-s-and-p-stand-for-what-is-xs-yp-or-xsyp-with-x-and-y-being-numeric-digits.6379/`. [Online; accessed 24-May-2022].

[33] Get FPV. All about multirotor drone batteries. `https://www.getfpv.com/learn/new-to-fpv/all-about-multirotor-fpv-drone-battery/`. [Online; accessed 24-May-2022].

[34] Power Sonic. What is a batteryc rating. `https://www.power-sonic.com/blog/what-is-a-battery-c-rating/#:~:text=The%20battery%20C%20Rating%20is,10%20Amps%20for%20one%20hour`. [Online; accessed 24-May-2022].

[35] Oscar Liang. Fpv drone flight controller firmware overview. https://oscarliang.com/mini-quad-fc-firmware/. [Online; accessed 24-May-2022].

[36] Oscar Liang. Fpv drone flight controller explained. https://oscarliang.com/flight-controller-explained/, 2021. [Online; accessed 24-May-2022].

[37] Wikipedia contributors. Atmospheric pressure — Wikipedia, the free encyclopedia. https://en.wikipedia.org/w/index.php?title=Atmospheric_pressure&oldid=1087938829, 2022. [Online; accessed 26-May-2022].

[38] Dictionary. Fail-safe definition. https://www.dictionary.com/browse/failsafe#:~:text=for%20fail%2Dsafe-,fail%2Dsafe,unlikely%20to%20fail%3B%20foolproof. [Online; accessed 26-May-2022].

[39] Drone nodes. Drone transmitter and receiver – radio control system guide. https://dronenodes.com/drone-transmitter-receiver-fpv/. [Online; accessed 26-May-2022].

[40] Mario TipsForDrones. Can drones fly without wi-fi? https://tipsfordrones.com/can-drones-fly-without-wi-fi/. [Online; accessed 26-May-2022].

[41] solenerotech EN. Tutorial: How to control a brushless motor with raspberry pi. https://solenerotech1.wordpress.com/2013/09/09/tutorialhow-to-control-a-brushless-motor-with-raspberry-pi/. [Online; accessed 02-June-2022].

[42] Electrical Engineering Stack Exchange - ShoutOutAndCalculateShoutOutAndCalculate. ESC Signal Standard and PWM frequency. https://electronics.stackexchange.com/questions/474764/esc-signal-standard-and-pwm-frequency. [Online; accessed 02-June-2022].

[43] Arduino Forum - System. Controlling an ESC. https://forum.arduino.cc/t/controlling-an-esc/261062, Oct 2014. [Online; accessed 02-June-2022].

[44] Wiki/PWM. Home. https://sourceforge.net/p/raspberry-gpio-python/wiki/PWM/. [Online; accessed 02-June-2022].

[45] ElectronicWings. Raspberry pi PWM generation using python and C: Raspberry pi. https://www.electronicwings.com/raspberry-pi/raspberry-pi-pwm-generation-using-python-and-c#:~:text=The%20default%20PWM%20mode%20on%20Raspberry%20Pi%20is%20Balanced%20mode.&amp;text=This%20function%20is%20used%20to%20set%20the%20range%20for%20PWM,The%20default%20range%20is%201024.&amp;text=We%20can%20generate%20PWM%20on%20every%20GPIO%20pin%20of%20Raspberry,Software%20PWM%20library%20of%20wiringPi. [Online; accessed 02-June-2022].

[46] Instructables AGTx and Instructables. Driving an esc/brushless-motor using raspberry pi. https://www.instructables.com/Driving-an-ESCBrushless-Motor-Using-Raspberry-Pi/, Sep 2017. [Online; accessed 02-June-2022].

[47] Developpez.com - GALODE Alexandre. Gestion du port gpio du raspberry pi avec python. https://deusyss.developpez.com/tutoriels/RaspberryPi/PythonEtLeGpio/. [Online; accessed 02-June-2022].

[48] Engineers Garage. RPi Python Programming 16: Analog output and software PWM. https://www.engineersgarage.com/articles-raspberry-pi-python-software-pwm-led-fading/. [Online; accessed 02-June-2022].

[49] Raspberry Pi Stack Exchange. What's the difference between soft PWM and PWM. https://raspberrypi.stackexchange.com/questions/100641/whats-the-difference-between-soft-pwm-and-pwm, 2021. [Online; accessed 02-June-2022].

[50] Maxime Lorenzetti. 'design, build, test and optimisation of a contrarotating propeller. Master's thesis, ULB, 2020-2021. (Chapter 1).

[51] Dc motor / propeller matching - lab 5 lecture notes. https://web.mit.edu/drela/Public/web/qprop/motorprop.pdf, 2005. [Online; accessed 04-June-2022].

[52] Siew Yeong and Sharul Sham Dol. Aerodynamic optimization of micro aerial vehicle. *Journal of Applied Fluid Mechanics*, 9:2111–2121, 07 2016.

[53] Nicolas Michel, Anish Sinha, Zhaodan Kong, and Xinfan Lin. Multiphysical modeling of energy dynamics for multirotor unmanned aerial vehicles. pages 738–747, 06 2019.

[54] NASA. Velocity effects on aerodynamic forces. `https://www.grc.nasa.gov/www/k-12/airplane/vel.html`. [Online; accessed 06-June-2022].

[55] Hal le blog modelisme rc de hal. Théorie du moteur électrique brushless. `https://le-blog-modelisme-rc-de-hal.over-blog.com/article-theorie-du-moteur-electrique-brushless-44939559.html`, 2022. [Online; accessed 06-June-2022].

[56] Holybro. Qav250 kit. `http://www.holybro.com/product/pixhawk-4-mini-qav250-kit/`. [Online; accessed 08-June-2022].

[57] Dronecode. Safety configuration (failsafes). `https://docs.px4.io/v1.9.0/en/config/safety.html`. [Online; accessed 08-June-2022].

[58] Unmanned Systems Technology. Wireless telemetry systems. `https://www.unmannedsystemstechnology.com/expo/wireless-telemetry/#:~:text=Drone%20telemetry%20is%20data%20gathered,as%20the%20aircraft's%20power%20source.` [Online; accessed 08-June-2022].

[59] Ardupilot. Vibration damping. `https://ardupilot.org/copter/docs/common-vibration-damping.html`. [Online; accessed 08-June-2022].

[60] PX4 User guide. Holybro qav250 + pixhawk4-mini build. `https://docs.px4.io/master/en/frames_multicopter/holybro_qav250_pixhawk4_mini.html`. [Online; accessed 08-June-2022].

[61] PX4 v1.9.0 User Guide. Full parameter reference. `https://docs.px4.io/v1.9.0/en/advanced_config/parameter_reference.html`. [Online; accessed 08-June-2022].

[62] QGroundControl User Guide. Vehicle setup. `https://docs.qgroundcontrol.com/master/en/SetupView/SetupView.html`. [Online; accessed 08-June-2022].

[63] Drone Sense. Compass calibration: Tips to get in the air much faster in mission-critical operations. `https://dronesense.com/compass-calibration-tips-to-get-in-the-air-much-faster-in-mission-critical-operations/#:~:text=Compass%20calibration%20on%20a%20drone,to%20that%20of%20true%20north).` [Online; accessed 08-June-2022].

[64] PX4 User Guide. Px4 flight modes overview. `https://docs.px4.io/v1.12/en/getting_started/flight_modes.html`. [Online; accessed 08-June-2022].

[65] PX4 User Guide. Controller diagrams. `https://docs.px4.io/v1.12/en/flight_stack/controller_diagrams.html`. [Online; accessed 08-June-2022].

[66] Techopedia. What is polylactic acid (pla)? - definition from techopedia. `https://www.techopedia.com/definition/29379/polylactic-acid-pla`, Jun 2013. [Online; accessed 07-June-2022].

[67] Honeycomb structures: From infill to bicycle tires. `https://3dprint.com/221695/honeycomb-structures-from-infill-to-bicycle-tires/`, Oct 2021. [Online; accessed 07-June-2022].

[68] Luca Delfino. étude et conception mécanique d'un hexadrone. Master's thesis, Henallux, 2021-2022.

[69] user2813274. How much distance do i need between quadcopter propellers to avoid issues? `https://aviation.stackexchange.com/questions/22269/how-much-distance-do-i-need-between-quadcopter-propellers-to-avoid-issues`, October 2015. [Online; accessed 08-June-2022].