AIgle

Final Report

Group 19

TU Delft

Page Intentionally Left Blank

Aigle

Final Report

by

Group 19

Dries Borstlap	4648099
Friso Dam	4297148
Nadine Duursma	4665236
Weronika Dziarnowska	4551117
Victor Guillet	4488636
Yestin van Haaren	4667581
Danny Huang	4669045
Max van Huffelen	4678214
Jeroen Riessbacher	4438051
Georg Strunck	4680421
Reinier Vos	4663160

Tutor	Christophe De Wagter
Coach	Bertrand Mercier
Coach	Yi Zhang

Version	**Change**	**Date**
0.1	Draft	June 22, 2020
1.0	Final	June 30, 2020

Preface

This report is written during the Design Synthesis Exercise (DSE), undertaken as part of the TU Delft Aerospace Engineering Bachelor program. It builds up on the work presented in the Midterm Report [**MidtermReport**], which builds up on the foundations set in the Project Plan [1] and Baseline Report [**BaselineReport**]. The design team consists of eleven bachelor students working full time for 10 weeks.

This report is written with the intention of paving the way for future drone racing endeavor and facilitating research and development of artificial intelligence.

The group would like to extend special thanks to Ir. C. De Wagter for his tremendous insight into drone racing, and for sharing his knowledge and experience in autonomous drone racing. The group would also like to express their gratitude to B. Mercier and Y. Zhang for their valuable advice and help during the design phase, and D. Martini for his input throughout the project. They have been present throughout the entire development of the project, and their contribution has proven essential to its success.

Group 19
Delft, June 2020

Contents

Nomenclature

List of Symbols

δ	Displacement	[m]
η	Efficiency	[%]
γ	Learning factor	[-]
λ	Signal wave length	[m]
ω	Angular rate	[rad/s]
ω_n	Natural frequency	[rad/s]
ϕ	Roll angle	[°]
ψ	Yaw angle	[°]
ρ	Density	[kg/m^3]
σ	Axial stress	[Pa]
τ	Shear stress	[Pa]
θ	Pitch angle	[°]
ζ	Damping ratio	[-]
A	Area	[m^2]
a_i	Acceleration in i-direction	[m/s^2]
B	Fibre orientation fraction	[-]
c	Damping coefficient	[Ns/m]
C_D	Drag coefficient	[-]
D	Drag	[N]
d	Distance	[m]
DoD	Depth of Discharge	[%]
E	Young's modulus	[Pa]
e	error	[-]
F	Fragility	[-]
f	Fibre volume fraction	[-]
F_c	Centrifugal force	[N]
G	Shear modulus	[Pa]
g	Gravitational acceleration	[m/s^2]
H	Drop height	[m]
I	Current	[A]
I_{ij}	Moment of Inertia	[m^4]
K	Rotor drag constant	[-]
k	Spring stiffness	[N/m]
K_i	Gain	[dB]
L	Length	[m]
M	Moment	[Nm]
m	Mass	[kg]
P	Power	[W]
q	Distributed load	[N/m]
R	Resistance	[Ω]
r	Radius	[m]
R_j^i	Rotation Matrix	[-]
R_i	Reaction force in i-direction	[N]
S	Surface Area	[m^2]
S_{ij}	Distance between points i and j	[m]
T	Thrust	[N]
t	Time	[s]
T_{min}	Minimum thickness	[m]
V	Voltage	[V]
v	Velocity	[m/s]
W	Weight	[N]

List of Abbreviations

AI	Artificial Intelligence
AIO	All-In-One
AIRR	Artificial Intelligence Robot Racing
ANP	Autonomous Navigation Performance
APC	Advanced Precision Composites
API	Application Programming Interface
AWG	American Wire Gauge
C&S	Control and Stability
CFD	Computational Fluid Dynamics
CFRC	Carbon Fibre Reinforced Composite
CPU	Central Processing Unit
CSI	Camera Serial Interface
DDPG	Deep Deterministic Policy Gradient
DDQL	Double Deep Q-learning
DQL	Deep Q-learning
DRL	Drone Racing League
DSE	Design Synthesis Exercise
ESC	Electronic Speed Controller
FC	Flight Controller
FFC	Flat Flexible Cable
FOV	Field Of View
FP	Flight Performance
fps	Frames Per Second
FPV	First Person View

GPIO	General-Purpose Input/Output
GPU	Graphics Processing Unit
IMU	Inertial Measurement Unit
IROS	International Conference on Intelligent Robots and Systems
LCA	Life Cycle Assessment
LiDAR	Light Detection and Ranging
LiPo	Lithium Polymer
LSTM	Long Short Term Memory
MCU	Microcontroller Unit
MQTB	Miniquad Test Bench
MV/SE	Machine Vision and State Estimation
NeurIPS	Conference on Neural Information Processing Systems
P&P	Power and Propulsion
PCB	Printed Circuit Board
PEI	Polyetherimide
PID	Proportional Integral Derivative
PSU	Power Supply Unit
PWM	Pulse Width Modulation
QL	Q-learning
RAM	Random Access Memory
Relu	Rectified Linear Unit
RPM	Revolutions Per Minute
SLAM	Simultaneous Localisation And Mapping
SODIMM	Small Outline Dual In-line Memory Module
Sust	Sustainability
SWOT	Strengths, Weaknesses, Opportunities and Threats
TAIR	Transonic AIRfoil
TD3	Twin Delayed Deep Deterministic Policy Gradient
UART	Universal Asynchronous Receiver-Transmitter
UE	Unreal Engine

Executive Overview

Mission

In recent years, interest in autonomous drone racing has increased. In order to keep these competitions fair, each team has to work with the same drone. In the past, these drones were provided by the organisers. This lead to both logistical issues for tournament organisers and teams lacking the ability to test their algorithms, since they had no access to the drone. To overcome these disadvantages, and to accommodate the advancement of Artificial Intelligence (AI), Group 19 of the 2020 Spring Design Synthesis Exercise (DSE) designed an open-source racing drone for future autonomous drone races.

In order to ensure that all teams have access to the drone and that no unnecessary strain is put on the supply of competition organisers, the cost of the drone shall be below 2500€ and be manufacturable using only standard methods. Furthermore, to ensure high quality, the drone shall be able to fly and manoeuvre at high speeds while being impact resistant and sustainable.

The AIgle racing drone that is developed and meets above requirements, will perform well compared to its competitors. The only drone for autonomous racing that has been widely produced for multiple teams is the Drone Racing League's (DRL) RacerAI, which is heavier, more expensive (5000 US$ compared to AIgle's 1410€), while also being outperformed by AIgle in terms of flight performance.

To achieve these qualities, a number of concepts have been investigated. Five concepts have been looked into; One focusing on computational power, a sensor-heavy concept, a highly durable concept and two balanced concepts including a tricopter and a quadcopter. These concepts were traded off based on autonomous navigation performance, flight performance, sustainability, risk, mass and cost. The winning concept, which is the durable concept, is developed further. The durability concept balances the ability to survive accidents during testing while maintaining good autonomous navigation performance, flight performance and sustainability compared to other concepts, thus giving a clear advantage to its users.

To ensure minimal risk during development, a technical risk assessment is performed. This assessment particularly showed that the drone is susceptible to crashes which can damage the expensive electronics on board of the drone. As such, emphasis is put on ensuring their safety and mitigation of the risks as a whole. In fact, of the 26 action points of the risk mitigation strategy, 20 have been successfully implemented in full.

Sustainability is an important factor to consider when designing any product. It concerns balancing the social, environmental and economic factors of a design. A sustainable development strategy is implemented by first conducting a life cycle assessment (LCA), which analyses the racing drone over all the stages of its life. Using the LCA, the possible ways the sustainability of the drone can be affected are investigated. A set of sustainability criteria are derived, which aim to identify what parts of the design affect the sustainability of the drone. Some of the criteria are used in the trade-off of concepts, and therefore contribute to the choice of design to be worked on further. After the design has been finalized, a sustainability evaluation is conducted. The final design is analysed with respect to each sustainability criteria. Additionally, certain recommendations are made to improve the sustainability even further and their effect on the design's performance are explained.

Development

The propulsion and power subsystem provides both the thrust and power for the drone. A database of motor and propeller data is accessed to find an optimal match. A battery was ultimately added to this duo based on the power requirement. All of these components are off-the-shelf. This matching process is automated with code, which allows for frequent iteration and consistently achieve the desired performance characteristics. The final iteration consists of the following components: an APC 6x4E propeller, a SunnySky Edge R2306 motor and a Tiger Power 3600 mAh battery. This configuration yields a thrust to weight ratio of at least 4 in dynamic conditions. Furthermore, it is predicted the drone

is able to achieve a maximum speed of 32.1 m/s.

The electronics are the brain and the nerves of the drone. It has to provide environmental sensing, computational power and power distribution. First of all, a trade-off concerning mass and repairability is performed for the system architecture, which mainly concerns power distribution. The chosen architecture has the lowest mass of all options and consists of a 4-in-1 Electronic Speed Controller (ESC), a flight controller and companion computer. Secondly, the electronic hardware components are chosen. Since processing should only be performed on-board of the drone, a ”companion computer” is needed besides the flight controller, which takes of care of the heavy processing. The chosen companion computer is the Nvidia Jetson Xavier NX for its high computational capabilities, with a low mass and a small form factor. This companion computer needs a separate carrier board which enables connections to peripherals such as the camera, flight controller and the WiFi module. For this, it was decided to design a custom carrier board as it saves mass and volume. In total, it is estimated that the custom carrier board saves at least 26% in mass compared to an off-the-shelf solution. Finally, cables and connectors are sized to allow them to withstand the heat associated with flying at full throttle for the full flight time of the drone.

The structure of the drone is designed to protect the sensitive electronics when being subjected to a drop of 3 meters and the drone should not break when crashing in a gate at 30 m/s. As the electronic components can only withstand small loads, a dampening system is designed to reduce impact loads. The frame layout has already been chosen earlier on as a ’deadcat’ frame. This is a frame in X-shape, with the forward facing arms bent at a wider angle. It allows the camera to have a sufficient field of view, and also protects it and the other electronics from head on crashes. The protection is designed based on the scenario of a 3m drop and the dampening system is divided into multiple stages. The points of contact with the object the drone crashes into are naturally the arms or the belly and top of the drone. Therefore propeller guards are attached at the motor mount in order to protect the propellers from injuring its surroundings, but also to provide cushioning of the electronics. Furthermore a ’landing gear’ is added to cushion the electronics from hard impacts. These are placed below the mainframe and are 3D printed, just like the propeller guards. The second form of low level cushioning is provided by the main frame and the arms, but they are rather stiff and only provide minimal cushioning. The last layer of protection are the standoffs for the electronics, that are acting as a spring and damper. This means that the processor and other electronic parts are able to move up to 8mm inside the main frame housing.

To demonstrate the capability of the drone, software has been developed that can run on the drone. The software is split in three parts; Machine Vision & State Estimation (MV/SE), Navigation and finally Control & Stability. MV/SE uses the data from the camera to determine the location of the drone with respect to gates that are in view. Navigation uses this data to determine the path the drone should take. The waypoints the drone has to go through are used by Control & Stability to give actual commands to the propulsion to execute.

MV/SE manages to identify the pixels of the gate from a single image by following the pixels of the gate through the image until all four corners are found. This works even under a roll angle of up to 26°. Using the found corners of the gate and the gate’s known size, the location and orientation of the gate can be found with errors typically no greater than 5%. The states are then stored in a database so they can be accessed later. This is used to find the speed of the drone which can be used by other programs.

Navigation has investigated and developed a variety of different methods from various fields to determine the path the drone should take. The varying complexity and methodology demonstrate the variety of approaches and implementations teams using the AIgle racing drone can use. Furthermore, by demonstrating the applicability of deep reinforcement learning, a baseline has been provided to future teams that wish to build their own autonomous navigation protocol.

Control & Stability developed the software responsible for translating desired waypoints to actual actions for the drone to do. This is demonstrated using an algorithm based on one developed by Daniel Mellinger, that generates the required inputs for Betaflight, which runs on the flight controller.

Using this method, high levels of accuracy can be reached; waypoints are reached with disparities no greater than 8 cm, and velocity and acceleration errors in the program are no greater than just 0.10m/s and 0.01m/s, respectively.

All in all, after completion of the design, the following key parameters are known. The mass of the final design is 1.3 kg, with an estimated power of 2.1 kW. Furthermore, the estimated cost of the drone including manufacturing is 1410€. Finally, in Figure 1 the render of the final design can be seen.

Figure 1: CATIA render of the final design.

Product Verification and Validation

Most of the requirements that have been imposed by the customer have been met. However, some requirements could not be met, most notably the mass requirement of 1 kg. The current mass of the drone is 1.3 kg, after performing mass optimisation for the electronics. Even if structure mass is further optimised, the mass requirement can not be met while complying with all other constraints. However, the drone is equipped with a state-of-the-art computer, which has been released in April 2020, and is specialised in AI computing. Furthermore, a dedicated dampening system is implemented to protect this computer. In addition, the mass of 1.3 kg is less than half of the RacerAI. Therefore, exceeding the mass requirement of 1 kg by 300 grams is considered justifiable.
Furthermore, the requirements that are only related to the sensor-heavy concept have not been considered, as they are not applicable anymore. There are also some requirements that are unverifiable at this stage, and post-DSE tests have been established to verify these.
Finally, a sensitivity analysis of the drone showed that the cost requirement can confidently be met, while the mass requirement can still not be met. Furthermore, sensitivity analysis showed that all requirements which were initially complied with are still complied with if the mass would increase or other parameters would change for the worse.

Post-DSE Activities

After finishing the DSE, tasks still remain to fully complete the design. First of all, the validation, component placement and routing of the custom carrier board design has to be outsourced to an external company. Secondly, the necessary components are ordered, produced and assembled into a prototype. This prototype will be tested for validation purposes and the design will be updated accordingly. These tests have been worked out in detail, with the test type, objective, and affected parameters. Once the design is finished, it will be published in various places to ensure visibility and accessibility. Furthermore, partnerships will be sought for large scale adoption in autonomous drone races.
Finally, the design is maintained through feedback from the open-source community and to keep up with the development of new technologies. The estimated timeline is that the design is finalised before 15 August 2020, the publishing is finalised before 29 August 2020, the finding of partnerships before 13 September 2020, and the design maintenance before 3 November 2020. It is estimated that the finalisation of the drone design costs 7720€, the publication and market entry 600€, and the maintenance and usage 270€ over one year. Finally, the total cost of the drone including production is estimated to cost 1410€.

1

Introduction

Drone racing has gathered increasing attention and interest in the past few years, establishing itself as a sport on par with more traditional ones, such as Formula 1. A subset, autonomous drone racing, has been growing rapidly in popularity, recently capturing the interest of the industry, the general public, and the scientific community. Since its beginning in 2016 with the IROS Autonomous Drone Race [2], the number of competitions have expanded. Industry players, such as the Drone Racing League (DRL) hosted its own competition in 2019, and fully virtual drone races were held by NeurIPS, also in 2019. However, the essence of these competitions has remained the same. Teams compete to develop the most effective software to pilot a standardised drone as quickly as possible through gates placed in around a circuit. A standardised drone is a critical aspect of the competition, as it ensures that the software is the sole deciding factor. During the DRL races, the competition host provided the drones, which the competitors would be flying, to ensure that no drone had a mechanical advantage over its opponents. This, however, induces a number of issues. The cost of hosting a competition goes up drastically as the cost of producing, maintaining, and repairing the drones becomes the responsibility of the host. It also means that competitors would not be able to effectively test, adjust, and optimise their algorithms prior to the race, reducing the teams' ability to compete. In an effort to alleviate these issues, group 19 of the Delft University of Technology 2020 Spring DSE aims to provide an open-source hardware solution for race organizers and teams to be used in future autonomous drone racing competitions, with as goal to facilitate a worldwide collaboration on the development of artificial intelligence in agile autonomous drone flight.

This report aims to provide a clear and detailed description of the process that was followed in the final design. This also includes an overview of the trade-off which resulted from the concept design, an evaluation of the risks present, and an analysis of the sustainability of the solution. The production will also be investigated, along with a market analysis. Guidelines and demonstrations are also provided to showcase the drone's potential and provide a basis for further development.

The report is structured as follows. Chapter 2 provides a mission overview. Chapter 3 present a market analysis of the drone market, and chapter 4 details the trade-off, which resulted from the conceptual design phase. Chapter 7 explains the power and propulsion sizing and selection, chapter 8 describes electronics and carrier board design, and chapter 9 describes the design process which was followed for designing the structure. Chapter 10 describes the software and software guidelines established, addressing in particular three major components, machine vision & state estimation, navigation, and control and stability. A design summary is then presented in chapter 11, and the production and operation of the design is described in chapter 12. A reliability, availability, maintainability, and safety analysis is found in chapter 13, and lastly, verification & validation and post-DSE activities are depicted in chapter 14 and 15 respectively. A conclusion is then found in chapter 16.

2

Mission Overview

This chapter provides a mission overview of the project. The project objective can be found in section 2.1, the functional analysis in section 2.2, and the requirements are elaborated in section 2.3.

2.1. Project Objectives

The mission need statement of the project is to

"Facilitate a worldwide collaboration on the development of artificial intelligence in agile autonomous drone flight, accomplishing faster decision making than human pilots".

The biggest platform for Artificial Intelligence (AI) drone developers are drone racing competitions such as the DRL and the IROS races. The DRL competitions require the teams to use the provided hardware, which is not available prior to the race, making it impossible to test the software on the drone. On the other hand, the IROS competitions require teams to not only design the software, which should be the main focus, but also the hardware. In order to facilitate and encourage more development in the field of autonomous drones, one solution to both problems could be the following: creating an AI racing drone hardware design that is available open-source such that developers can easily manufacture the drone and test their software extensively. Therefore, the project objective statement is to

"Provide an open-source hardware racing drone design and demonstration software for the artificial intelligence robotic drone race competition by eleven students in ten weeks".

In order for the design to be useful to developers, it should consist of off-the-shelf products that can be bought worldwide. It should also utilise an impact resistant structure that can be manufactured using simple machining techniques, and it should have a flight performance that is comparable to or better than existing AI racing drones. To allow for efficient testing and racing, the system should also be reliable and durable to limit the time and money spent on repairs. And most importantly, the drone should be designed in a way that maximizes the system capabilities such that novel algorithms can be used by developers.

2.2. Functional Analysis

The product that is being designed and the way to achieve that design can be described as a system. In order to reach the end goal efficiently, it is necessary to know what is expected of that system. A functional analysis is meant to give an overview of all system functions. For the purpose of this project, the system does not only consist of the operation of the product, but also the way it is designed, distributed, produced and disposed of.

The logical order in which these functions are performed is represented in the functional flow diagram as can be seen in Figure 2.1. The diagram is structured chronologically and each function has a unique label, so it can be referred to.

Apart from the functional flow diagram, a functional breakdown structure is also constructed and can be seen in Figure 2.2. Opposed to having a chronological structure, it shows all functions structured hierarchically. This reveals the relations between different functions. Some functions are the same as in the functional flow down diagram, which is indicated by using the same numbering of the function. If it is a function not used in the functional flow diagram, it is numbered alphabetically.

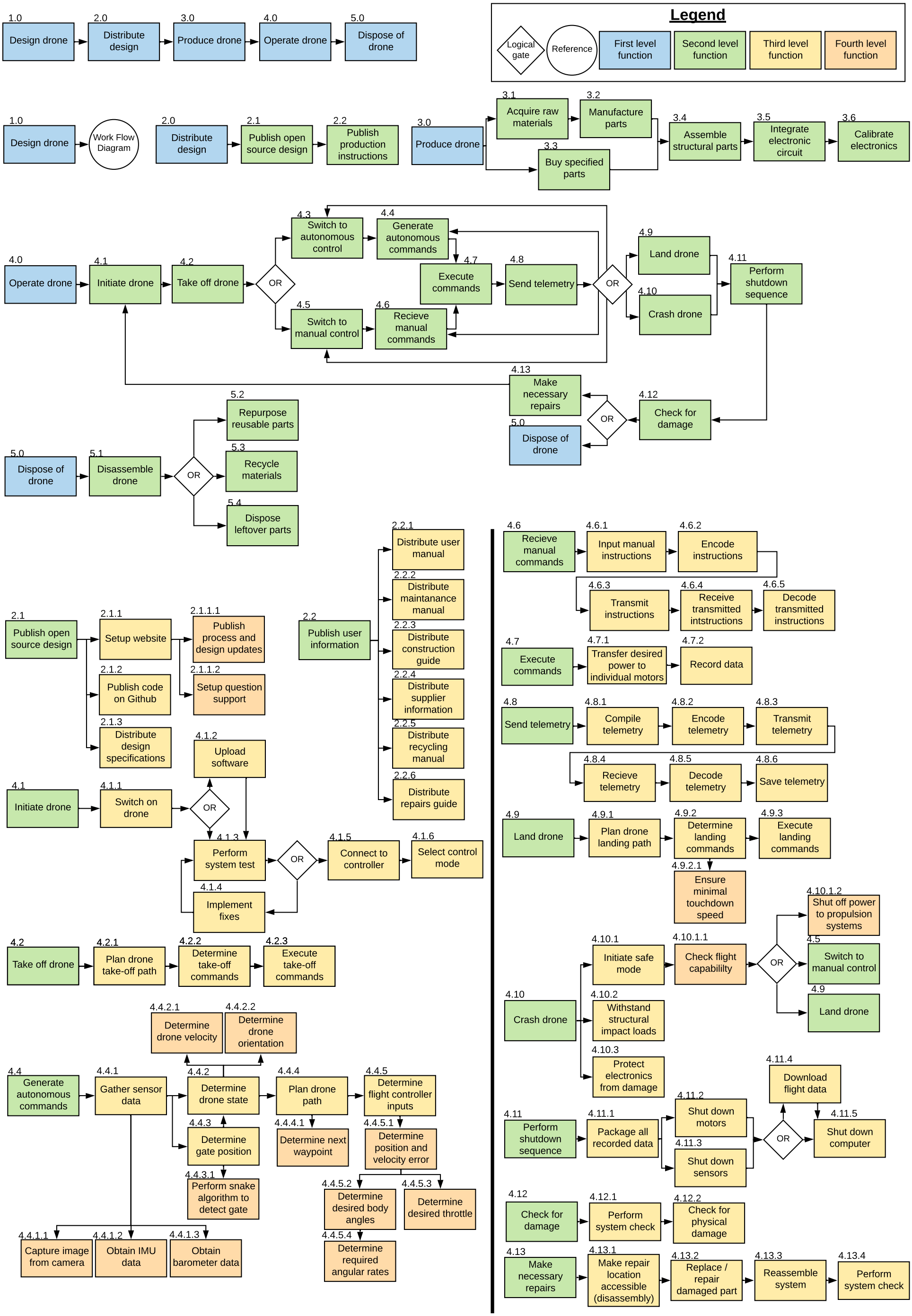


Figure 2.1: Functional flow diagram of the AI racing drone project

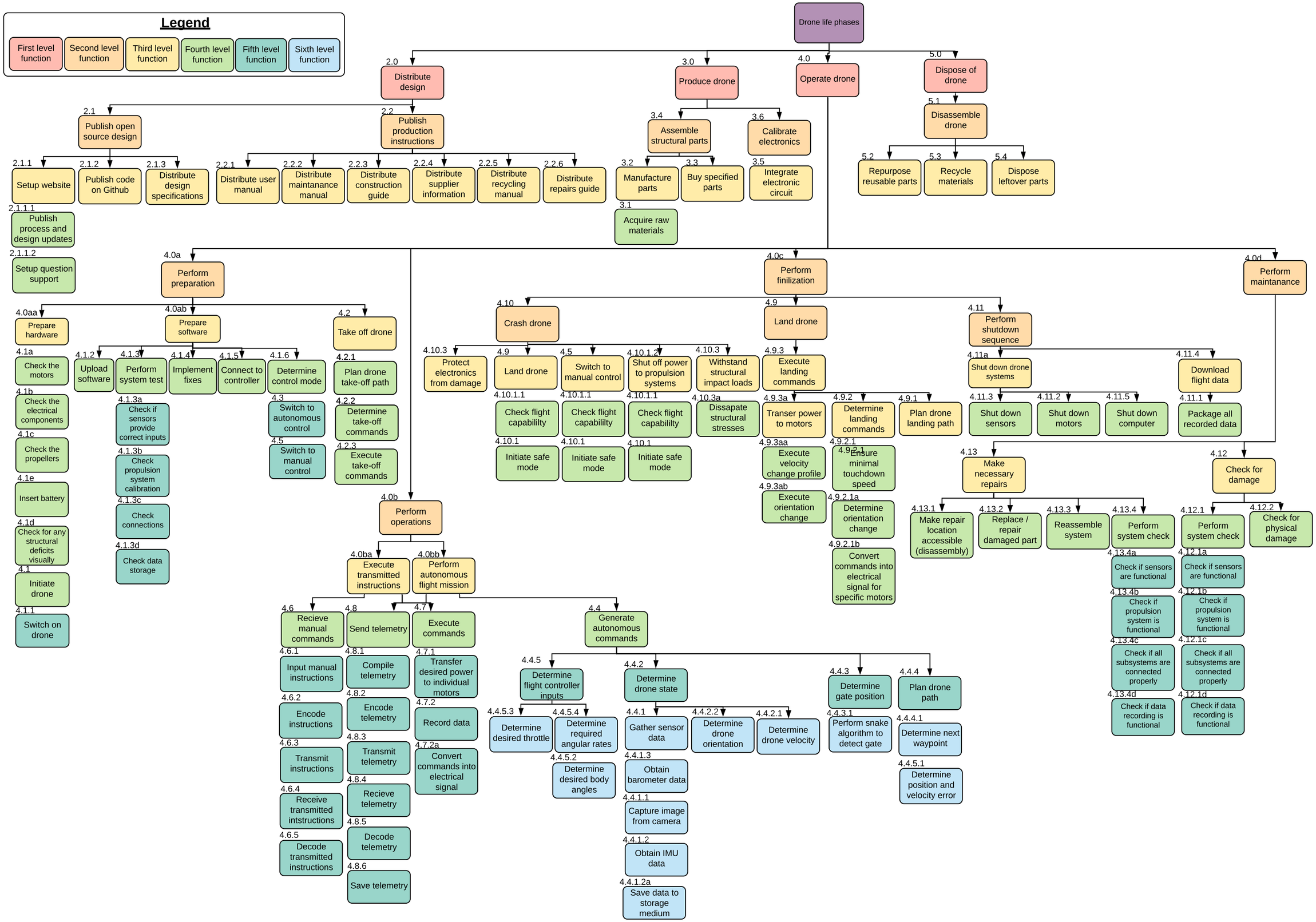


Figure 2.2: Functional breakdown structure of the AI racing drone project

2.3. Requirements

In this section, the stakeholder requirements, on which the design of the drone is based are presented. Multiple requirements have been generated based on those, in order to steer and facilitate the design process. These requirements are discussed in detail in the Midterm Report [**MidtermReport**]. Moreover, at the beginning of every subsystem chapter, the relevant requirements are stated. Moreover, both the stakeholder requirements and the requirements generated by the team are summarised in section 14.1, where it is also discussed if the design meets them or not.

The stakeholder requirements are summarised in the following list:

Performance

AIG-TE-HW-PF1: The drone shall have a maximum speed of at least 30 m/s.

AIG-TE-HW-PF2: The drone shall be able to make a 3 meter radius arc while flying at 10 m/s.

AIG-TE-HW-PF3: The drone shall be able to linearly accelerate at 19.61 m/s^2 from zero velocity.

AIG-CR-04: ~~The drone shall be able to fly for at least 5 minutes.~~
Renegotiated to AIG-TE-HW-P-E1

AIG-TE-HW-P-E1: The battery shall have a begin-of-life capacity of at least equal to (100% throttle power plus 100% computer power plus 100% power of all other systems) times 60 seconds plus 100% computer power times 120 seconds.

AIG-TE-EB-M4: The drone shall have enough computational power to process images at 60 Hz and calculate a flight path.

Safety and reliability

AIG-TE-HW-CO5: The drone shall have manual take-over capability.

AIG-TE-HW-MP-IR1: The drone shall suffer no damage to on-board systems or structure if the drone is dropped from 3 meters onto a concrete floor.

AIG-TE-HW-MP-IR2: The drone shall suffer no fracture, plastic deformation or loss of electronic function if the drone flies into a gate at maximum flight speed.

Sustainability

AIG-CR-10: ~~The drone shall only consist of recyclable materials.~~
Renegotiated to AIG-NT-S4

AIG-NT-S4: The sustainability of the drone shall be maximized.

AIG-TE-P-R2: Each component of the drone shall be able to be replaced individually.

Engineering budgets

AIG-TE-EB-M3: The maximum drone dimension in all length, width and height shall be smaller than 50 cm.

AIG-TE-EB-M1: The total mass of the drone shall be no greater than 1 kg.

AIG-TE-HW-S-V1: At least one front facing camera shall be present.

AIG-TE-HW-S-V1.3: A front facing camera on the drone shall have a frame rate of at least 60 Hz.

Other

AIG-NT-F-D1: The hardware costs of one drone shall not exceed 2500 euros.

AIG-TE-P-M-M1: All parts of the drone shall be manufacturable using 2D milling, 3D printing and printed circuit board machines.

AIG-TE-HW-C4: The drone shall use on-board computation only.

AIG-TE-HW-C5: The drone shall use the 'Betaflight' program as attitude controller.

AIG-TE-HW-CO6: The drone shall have a telemetry option.

AIG-TE-P-R1: All drone modules shall be able to be repaired/replaced in at most 2 minutes.

3

Market Analysis

This chapter includes the market environment that the AIgle drone will enter. After describing the purpose of the drone concept in section 3.1, the current market is analysed in section 3.2 by looking at possible competitors and the volume of the market. This translates into the demands from the market in section 3.3 and the stakeholders of the project design in section 3.4. Then, the future market is discussed in section 3.5, followed by the internal strengths and weaknesses as well as the external opportunities and threats, which are explained in a Strengths, Weaknesses, Opportunities and Threats (SWOT) table in section 3.6. Lastly, the return on investment is briefly given in section 3.7.

3.1. Value Proposition

AIgle aims to provide the scientific society with a racing drone design that is steered by AI and would be able to fly through a race track. The AI drone is designed to be ruggedized for high speed gate crashes and ground drops, while still being as lightweight as possible and providing high computing power. It is the purpose of AIgle to create a base design for testing AI programs and therefore, to facilitate the development of autonomous navigation.

3.2. Current Market Analysis

First, a look into the current possible competitors is given in order to estimate the impact of the AIgle drone on the market. After that, the volume and different segments of the market are investigated.

3.2.1. Product Competitors

Currently, there is only one AI racing drone that has been produced for multiple teams. This is the RacerAI drone designed and built by the DRL. So far, it has only been used for AI racing competitions and it is not available to buy. Nevertheless, it is estimated to cost around $5000 to build it [1] as it uses injection moulded components, which are expensive at low production numbers [3]. Furthermore, due to its heavy weight of 3kg and large size of 70cm x 50cm x 20cm, the drone is difficult to control [2].

There are also other drones that rely on artificial intelligence for object avoidance but they do not decide themselves on the path to take. One example is the Skydio2 autonomous drone. It can either be directly piloted or it can follow the path of an emitter, while also avoiding objects. It costs currently $999 and weighs 775g with its small size of 22.3 cm x 27.3 cm x 7.4 cm. It can reach up to 15.6 m/s and has a maximum flight time of 23 minutes [3]. Even though the drone is not capable of fully autonomous flight, its hardware layout has the potential to suit a completely AI piloted drone.

Another type of drones are first person view (FPV) racing drones. These are piloted directly by a human and only have to relay a video back to the pilot. One of the high-end products available is the Mr Steele Apex 5″ Quadcopter RTF, which is priced at 899 US$ for its minimum version. Its largest dimension is 25cm and it weighs 399g without the battery, which results in an approximate total weight of 600-800g [4]. Despite the small size and good FPV performance, these drones are not suitable to join the market of AI drones, as their hardware differs too much from what autonomous drones require.

The AIgle drone design is estimated to cost around 1410€ and to have a high maximum speed of up to 32.1 m/s. It has a diagonal wheelbase of approximately 30 cm and weighs around 1.3 kg. Thus, it can be seen that the AIgle drone comes a little closer to the Skydio2 than the RacerAI, outperforming both in agility, speed, and ruggedness.

[1] Unpublished conversation with Christophe de Wagter
[2] Unpublished email conversation with Fede Paredes
[3] https://www.skydio.com/, (Retrieved June 10, 2020)
[4] https://www.getfpv.com/mr-steele-apex-5-quadcopter-rtf-bundle-6s.html, (Retrieved June 10, 2020)

3.2.2. Volume of Market

Currently, there is no large market for fully autonomous racing drones yet. First FPV drone racing competitions started in early 2014 growing into a much larger sport over the years, with drones reaching up to 100 mph. Some of them are non-profit competitions, such as MultiGP. Others are for profit, such as the DRL, which is sponsored mostly by television companies across the world, such as SKY, with the goal to bring the idea of drone racing to the wide public [5].

The first autonomous drone race started in 2016 as a technology competition held by IROS and has lead to the DRL organizing their first AI drone race in 2019 [6]. This last race has attracted 424 teams, representing 2300 people, from which 9 teams entered the race after the entrance tests [6].

The market for unmanned aerial vehicles is comparably much larger and has its roots much further back in history with unmanned hot air balloons already appearing as early as 1849 and reaching a technologically sophisticated level and availability to the public around the year 2000 [7]. Besides the many military applications, an autonomous aircraft can be used in other civilian purposes doing work that would be difficult to do by hand. Those are, for example, applications in ground surveillance for safety or mapping purposes, as well as photography of specified objects, or agricultural tasks such as crop mapping [8]. In general, the degree of autonomy always depends on the purpose. The application for a drone like the AIgle concept can therefore lie in any of these purposes and therefore, it has a large market volume, provided the programming of the drone is designed for the purpose. The AIgle drone is not designed to have an additional payload and therefore can only be used for visual information gathering.

The design concept of the AIgle drone is intended to be published openly such that it can be used by any person as a basis to develop their own drone for their specific purpose. The intended purpose of this design is to be comparable to drones used in races such as the DRL Artificial Intelligence Robot Racing (AIRR) competitions.

3.3. Demands from the Market

The AIgle drone is specifically designed for autonomous AI races. This includes not just the race itself, but also the training before the race, which imposes certain requirements on the design.

First and foremost, the drone has to be able to decide on its own on which path to take. This requires good visualization of the surroundings by a camera and high computing performance to interpret this camera information. Developing such AI software requires extensive testing, meaning that crashes are very likely in this phase. Therefore, the drone needs to be very well protected and has to be designed to sustain a 3m drop, as well as gate crashes at high speeds. Due to the complexity of AI software, malfunctions are likely and thus, the drone should also have manual takeover capability in order to ensure the safety of the surroundings and of the drone itself.

As the drone is meant to be used for the drone races, high maximum speed and agility are highly important, much more so than flight time. This translates into a high thrust to weight ratio, requiring powerful motors and a lightweight design. The lightweight design also positively contributes to the crash resistance of the drone.

Furthermore, it can be understood that the current DRL and AIRR races require the pilots to design their software beforehand. The teams therefore can only practise with the drone that is used during the competitions just minutes before its start. It would be much better if the teams had the drone earlier, fitting their program perfectly onto the specific drone. This is possible with an easy to build and openly available drone design, which the AIgle team aims to create.

Finally, besides the technical requirements, the drone is meant to enter the racing market, which gains profit by broadcasting the races on television. This in turn requires a good looking design, including

[5] https://thedroneracingleague.com/sponsors/ (Retrieved June 29, 2020)
[6] https://thedroneracingleague.com/ (Retrieved June 10, 2020)
[7] https://consortiq.com/short-history-unmanned-aerial-vehicles-uavs/ (Retrieved June 29, 2020)
[8] https://mydeardrone.com/uses/, (Retrieved June 10, 2020)

shape and light effects.

3.4. Stakeholders

There are two main stakeholders that are to be considered in the AIgle drone design. These are the drone race organisers and the participants of these competitions.

The drone race organisers will decide for themselves if the proposed AIgle drone fits their purposes. As there are different races with different goals, it might be reasonable that the AIgle design will be used as a basic layout, which can be slightly modified for the purpose of each race. The DRL also relies on the public watching the races in person and on TV [6]. Therefore, these spectators and the TV providers are indirect stakeholders to the application of the AIgle drone design to these races. Furthermore, the organisers are to decide whether to allow their participants to train with the competition drone beforehand or not. It would allow them to steer the teams towards writing a more widely applicable software instead of one that suits only one specific drone perfectly.

The participants are another group of important stakeholder of the races and the drone design. The teams are usually created within technological companies and universities, which both have the resources and knowledge to build the drone, program it correctly and train it. Each of these have their own preferences on whether to use the AIgle drone concept or to build their own. The process of developing a flying drone in this layout also requires significant financial resources of the company or university. Naturally, it is also possible that individuals would try to design their own drone by themselves, meaning they would do it in their own free time as hobbyists.

Furthermore the availability of production resources is also very important to the building of the AIgle drone. This means that the AIgle design should be designed such that it can be built without highly sophisticated technology. It is necessary that the participants and the organisers have access to the production techniques that are to be used for building the drone.

Besides those three, it should also be mentioned that the AIgle concept is being designed by TU Delft students within the bachelor curriculum. It is the current understanding of the design team that the design is to be made publicly available, allowing the racing organizers to choose if they would like to use it. Moreover, each country has a different legislation on unmanned aerial vehicles. This might restrict the use of the AI drone for specific purposes or limit its characteristics such as the maximum speed.

Lastly, the success of the project depends on the skills and resources of the design team. The team consists of eleven students in their third year of the aerospace bachelor curriculum of the Delft University of Technology, who are doing this project as their graduation project.

3.5. Future Market Prediction

This section discusses the predicted future developments and the possibility of new markets. After that, the possible market share that the AIgle concept could achieve is looked at.

3.5.1. New Markets & Future Development

As mentioned in subsection 3.2.2, 424 teams have applied to the AIRR race in 2019. From those only 9 teams have been selected to actually race with the RacerAI drone of the DRL. With an open source design such as the AIgle, many more teams would be able to start programming their own code and it is expected that all the teams would produce the drone beforehand by themselves. This would give them higher chances of developing a good software for the upcoming AIRR races or others. It should be noted that some companies and educational institutions might be building more than one drone, allowing them to train more often and proceed with the programming faster.

Furthermore, it is expected that the threshold to design and build an AI drone would be lowered significantly. This would allow programmers and hobby drone pilots to enter this niche market without the need for specific knowledge, which has been required so far for these projects.

Considering the AIRR race as described before, it can be estimated that around 500 of the AIgle

drones will be built within the first year. The production of this specific design will likely fade out over the next four to five years due to newer technologies arising. However, the design has the potential to set a good baseline in the first years after being published, such that altered versions can be used for many years, and modified as new technologies and applications arise.

Besides the application of the Aigle drone for racing competitions, it could also be used for different purposes in a slightly modified version. This is mainly left up to the programmer. The design can be used in different markets, as mentioned in subsection 3.2.2. Thus, it could, for example, be used for facial recognition used to facilitate police investigations, such as searching for criminals in cities. If a project like this would be tried, it would likely first be tested within one city using around 10 drones. Also, it might be feasible to program the drone for agricultural purposes, such as crop investigation. For this project to start, it is also expected that around 10 drones would enter the market as a feasibility study, one for each farmer. Naturally, there are more markets than the ones mentioned. For the project, it is expected that its feasibility will first be tested and after that, the AIgle drone design could be produced in larger numbers.

3.5.2. Share of Future Markets

Currently, there are no direct competitors to the AIgle concept except for the RacerAI made by DRL. Therefore, the AIgle drone should be able to outperform existing solutions and will have the advantage of being one of the first such designs on the market. This means that the model should be very well designed in order to be used as 'new standard' of AI drones. This can only be achieved by further improving and testing of the AIgle drone. This means that in the first years, race organisers and the participants might build the AIgle concept with their own small modifications. During this phase, all models could be easily traced back to be a derivative of the AIgle design. Later on, a few concepts will likely lead the way for certain applications. Still, it should be mentioned that by far not all AI drone builders will be using the AIgle concept. It can be expected that part of the market does not even know about the AIgle concept. This has been estimated to be around 20-40% of the market. From the ones that will know about the design, not all will build it but rather develop or continue their own design. Those are expected by the AIgle team to cut down the entire percentage of market share to around 50%. But as can be seen, it is difficult to argue what the exact market is and how to trace the original concept down the line for more than two years. For the case that it is only looked at the drone race participants and assuming that the competition adopt our design, it is likely to have a close to 100% market share.

In order to reach a large share of the market, the AIgle design will have to be released efficiently. This could be done by setting up a website with all the necessary information to build the drone and to get it flying. Once everything is ready, it would be time to contact drone racing organisations, such as DRL and propose the AIgle design to them. This proposal should fit the plans of the corresponding organisations. For the DRL, it might be important that the exact design that they will use during the competition is not published, depending on the layout they want to give the competition. Furthermore, the AIgle design should be communicated to newspapers with a technical audience. Again, it is very important to have the concept tested properly and running perfectly because otherwise it would give a negative impression.

3.6. SWOT Analysis

For a good insight into the potential of the AIgle project, a SWOT analysis is performed and presented in Table 3.1. To the right, the external threats and opportunities are shown, while on the other side, the internal strengths and weaknesses are presented. This analysis helps the AIgle team to know and work on their strengths and weaknesses, while preparing for external threats and opportunities.

Table 3.1: SWOT analysis for project launch organisation and execution based on the market analysis

Strengths	Weaknesses
Very high speed and good agility of AIgle drone design	Short flight time
Ruggedized design helps in training and prevents from expensive crashes	Design needs to be tested and validated
Quick repairs due to modularity of AIgle drone	Requires the participating teams to build the drone themselves
High autonomous navigation performance allows for different purposes	Lack of marketing experience
AIgle concept will be made accessible at no cost	Lack of aesthetic design experience
Opportunities	**Threats**
New markets opening with the need for aerial footage	Unpredictability of cost of materials and hardware for production
Easy accessibility of drone technology to the public	Success of AIgle acceptance by racing organisations is critical
New materials or manufacturing techniques increases the performance of the AIgle drone	Future of AI drone races and their broadcasting is not certain
TU Delft reputation or student status helps improve traction	Restricting legislation
Improving the development of AI piloted drones	Rising of competing open-source designs

3.7. Return on Investment

The production cost of one drone has been estimated to lie at 1410€ and a maximum share of 400-500 drones built in the year of release has been found to be possible as described in the sections before. The purpose of the project is to provide a design of an AI racing drone at no cost. It is thus designed such that it can be built by the customers themselves and therefore, the design uses only simple manufacturing techniques that are also applicable to building just one product.

With this concept, no profit in terms of money is generated. Instead, it is the goal to build a basis, on which the knowledge of autonomous navigation and control can be enhanced. With this concept, the design has already been largely finished, which allows to lower the threshold to programming artificial intelligence significantly as explained in subsection 3.5.1. Using this approach, a return of investment will become positive as soon as the first concepts are built and researchers start developing software for them. At this stage, only the student team has invested their time in the design at no cost to the customer.

If one is only interested in the monetary return of investment, it can be looked at the individual 'customer' himself. The customer decides to build the drone for their AI training purposes, which costs around 1419€. This cost is negligible compared to the profit the customer could make, given that the first prize in the last AIRR race in 2019 was 1,000,000 US$ [9]. This brings the money into perspective and acts as additional motivation.

It is also possible that the competition organisers would build the drone themselves. This has been done already with the RacerAI for the DRL AIRR races. This drone has been estimated to cost around 5000 US$ to build [1]. Still, it seems unlikely that the DRL made a negative return on investment afterwards. They gain their positive return on investment from selling the racing 'show' to the television companies and by attracting sponsors. This positive return of investment would only increase by using the AIgle drone. This is not just because it is significantly cheaper to build, but also it is designed to withstand hard crashes, which lowers repair costs.

[9] https://thedroneracingleague.com/airr/, (Retrieved June 25)

4

Trade-Off Summary

This chapter serves as a summary of the conceptual design phase presented in detail in the Midterm Report [**MidtermReport**]. The developed concepts and their general characteristics are first presented in section 4.1. Then, section 4.2 presents the trade-off process of choosing the winning concept. Finally, section 4.3 presents the technical budget breakdown of the chosen concept.

4.1. Design Concepts

The goal of the project is to design an autonomous racing drone to facilitate a collaboration on the development of AI racing drones. The drone should allow its users to apply novel software solutions, as well as allow for frequent testing and easy repairing, while keeping the vehicle small, fast, and agile. Naturally, not all these aspects can be maximised and some compromises should be made. In order to determine what drone characteristic to focus on for best results, five conceptual designs have been created, each focusing on optimizing a different design aspect. The concepts, their philosophies and most important design characteristics are presented in Table 4.1. Full descriptions of the concepts and the design methodologies can be found in the Midterm Report [**MidtermReport**].

Table 4.1: Summary of the characteristics of the drone conceptual designs.

	COMP_C1	**DUR_C2**	**SENS_C3**	**LWQ_C4**	**LWT_C5**
Description	Focus on computational power by using a state of the art SoC.	Focus on durability by adding extra payload protection and strengthening the base frame.	Focus on data gathering capabilities by using various types of extra sensors.	Focus on low mass and agility by using a balanced payload and structural protection.	Focus on low mass and agility by using a balanced payload and the benefits of a tri-copter configuration.
Propulsion configuration	Quad-copter	Quad-copter	Quad-copter	Quad-copter	Tri-copter
Frame configuration	HX - frame	Deadcat frame	X - frame	Deadcat frame	Y - frame
Size (largest diagonal) [mm]	360	250	250	250	240
Sensors	low-latency camera with FoV of 130^o and FHD at 60 fps, IMU on the FC	low-latency camera with FoV of 130^o and FHD at 60 fps, IMU on the FC	low-latency camera with FoV of 130^o and FHD at 60 fps, IMU on the FC, stereo camera with 8m range, LiDAR with 10m range	low-latency camera with FoV of 130^o and FHD at 60 fps, IMU on the FC	low-latency camera with FoV of 130^o and FHD at 60 fps, IMU on the FC
Companion computer	NVIDIA Xavier AGX	NVIDIA Xavier NX	NVIDIA Xavier NX	NVIDIA Xavier NX	NVIDIA Xavier NX
Computer speed [TOPs]	32	21	21	21	21
Computer mass [g]	650	180	180	180	180
Estimated protective cover mass [g]	60	140	60	60	60
Estimated base frame mass [g]	180	170	130	140	120
Estimated total dry mass [g]	1100	610	670	510	480
Estimated total wet mass [g]	1680	980	1100	840	810

4.2. Trade-Off and Sensitivity Analysis

After defining and sizing the five conceptual designs, one concept had to be chosen such that it could be designed in detail in the final design phase. In order to choose the concept that suited the project objectives best, a trade-off was performed using various criteria. Concepts were scored for each criteria. The results were collected in a trade-off table and the overall weighted average scores were calculated.

4.2.1. Trade-off Method

Autonomous Navigation Performance (ANP) - Weight of 35%
The autonomous navigation performance of the drone directly influences how advanced and efficient the AI drone software can be. This makes this criterion the most driving with the highest weight. The ANP is related to the speed of the AI computer, as well as the quality and variety of data gathering devices. The criterion is assessed quantitatively for the computer clock speed, and qualitatively for the estimated data gathering capabilities.

Flight Performance (FP) - Weight of 20%
The flight performance characteristics are a crucial aspect of a racing drone, which is why this criterion has the second highest weight. A drone's flight performance is highly influenced by the frame shape, overall drone size, and the mass moments of inertia. All these have been estimated using sizing based on statistics and off-the-shelf products, and then creating CATIA models of the drone configurations. The positive influence of these values on flight performance is then qualitatively assessed and a weighted average calculated for each concept.

Risk - Weight of 20%
In order to be able to efficiently test and race the drone, it should have a sufficient impact resistance and be easily repairable. The more frequently the drone gets damaged upon impact, the longer the down time required to fix and/or replace parts. For this reason, the risk of structural and electrical damage should be minimised but also, the design method should be reliable to avoid unforeseen malfunctions. These three types of risks are estimated qualitatively, resulting in a weighted average score for each concept.

Sustainability (Sust) - Weight of 10%
Keeping the design process and the product itself sustainable is important, however, it is not a driving factor in this particular project, which is why this criterion received a medium importance weight. Since the design process for each concept can be kept equally sustainable, it is the product sustainability that this criterion assesses. It takes into account the amount of power consumed by the drone, the amount of material required to make the drone, and the qualitatively estimated drone durability.

Mass - Weight of 10%
The concept's mass is considered in the trade-off as it affects the drone's controllability and usually, larger mass means larger size, which increases the probability of crashing into a gate. It is also a stakeholder requirement to keep the drone mass under 1 kg. The concepts' masses are estimated based on off-the-shelf product characteristics, statistics and preliminary designs in CATIA.

Cost - Weight of 5%
The cost of each concept should be taken into account as it is a stakeholder requirement to stay within a budget of €2500. However, it is desirable to use an smaller budget for building the drone such that the remaining resources can be spent on any necessary fixes and part replacement. This would give drone software developers more opportunity to test, as crashing would not result in repair costs that the teams cannot afford.

4.2.2. Trade-off

Using the aforementioned criteria, a trade-off is performed and summarised in Table 4.3. Concepts are awarded points on a scale of 0-5 in each criterion and using the respective weights, a weighted average is calculated and shown in the last column. The color coding of the cells, as well as the logic behind rewarding scores, are explained in Table 4.2.

Table 4.2: Legend for the Trade-off Table [**MidtermReport**].

Score Range	Quality	Color
≥ 4.5	Excellent	Green
$2.5 \leq x < 4.5$	Good	Blue
$1 \leq x < 2.5$	Marginal	Yellow
0	Unacceptable	Red

Table 4.3: Trade-off table [**MidtermReport**].

	ANP	FP	Risk	Sust	Mass	Cost	Score
COMP_C1	4.2	1.5	2.2	1.0	0.0	3.3	2.38
DUR_C2	3.2	3.8	3.8	4.0	4.2	4.9	3.81
SENS_C3	4.0	4.2	2.0	2.8	2.0	4.3	3.30
LWQ_C4	3.2	4.4	2.8	3.6	4.9	4.9	3.66
LWT_C5	3.2	4.9	0.0	3.2	5.0	5.0	3.17

In the ANP criterion, most concepts have a good score with the exception of COMP_C1, which has excellent computational power and SENS_C3, which has excellent data gathering capabilities.

In the FP criterion, most concepts score good, due to their medium size and low moments of inertia. The worst score is awarded to COMP_C1 as it is the largest and heaviest concept, and the best score is awarded to LWT_C5 as it has a very low weight and uses a tri-copter configuration, which results in excellent flight performance.

Moving on to the Risk criterion, the best score is awarded to DUR_C2 due to its high impact resistance and the second best score is awarded to LWQ_C4 due to its balanced structure. Then, two marginal scores are awarded to COMP_C1 and SENS_C3 as they both include heavy and expensive electronic devices that are not well protected, and finally, an unacceptable score is awarded to LWT_C5 as the gimble necessary in a tri-copter configuration breaks very easily and is difficult to repair.

Then for the Sustainability criterion, most concepts receive a good score as they use a similar amount of material and energy, with SENS_C3 and LWT_C5 receiving lower scores sue to their low durability. Concept COMP_C1 is awarded a marginal score due to its large size and power consumption and low durability.

Moving on to the Mass criterion, most concepts receive a good or excellent score, with SENS_C3 awarded a marginal score as it exceeds 1 kg by 10%, and COMP_C1 awarded an unacceptable score as it exceeds 1kg by far more than 10%. And finally, in the Cost criterion, all concepts score good or excellent as they all stay well within the budget of €2500.

As concepts COMP_C1 and LWT_C5 receive one unacceptable score each, they are removed from further consideration. The three remaining concepts have similar overall scores so the winning concept can be chosen only after a sensitivity analysis.

4.2.3. Sensitivity Analysis

The sensitivity analysis is performed in order to study how much the criteria weights affect the trade-off results. Weights have been assigned in a very much qualitative way, for example, an important criterion could weigh 20% but also even 50% and only by sensitivity analysis can this weight sensitivity be determined. The analysis is summarised in Table 4.4. Taking the ANP criterion as an example, the analysis is done as follows. First, an equal weight of 3% is taken off five concepts and the total of 15% is added to the initial ANP weight. Then, a weight of 15% is taken off the initial ANP weight and distributed evenly among the other criteria. This procedure is repeated for all other criteria, where the weight added or subtracted is proportional to the initial criterion weight.

Table 4.4: Sensitivity analysis for the trade-off [**MidtermReport**]. Green cells signify the winning concept per weight distribution.

	Old score	ANP +15%	ANP -15%	FP +15%	FP -15%	Risk +15%	Risk -15%	Sust +10%	Sust -10%	Mass +10%	Mass -10%	Cost +5%	Cost -5%
DUR_C2	3.81	3.65	3.96	3.76	3.85	3.85	3.76	3.80	3.81	3.82	3.79	3.86	3.76
SENS_C3	3.30	3.44	3.15	3.48	3.11	3.05	3.54	3.25	3.34	3.15	3.44	3.36	3.23
LWQ_C4	3.66	3.52	3.79	3.73	3.58	3.45	3.87	3.61	3.70	3.77	3.54	3.71	3.60

As it can be clearly seen in Table 4.4, the DUR_C2 concept is winning in all but one iteration. It is thus the winner of the trade-off and is therefore designed in detail, as described in the following chapters.

A preliminary render of the durability focused concept from the Midterm Report [**MidtermReport**] is presented in Figure 4.1.

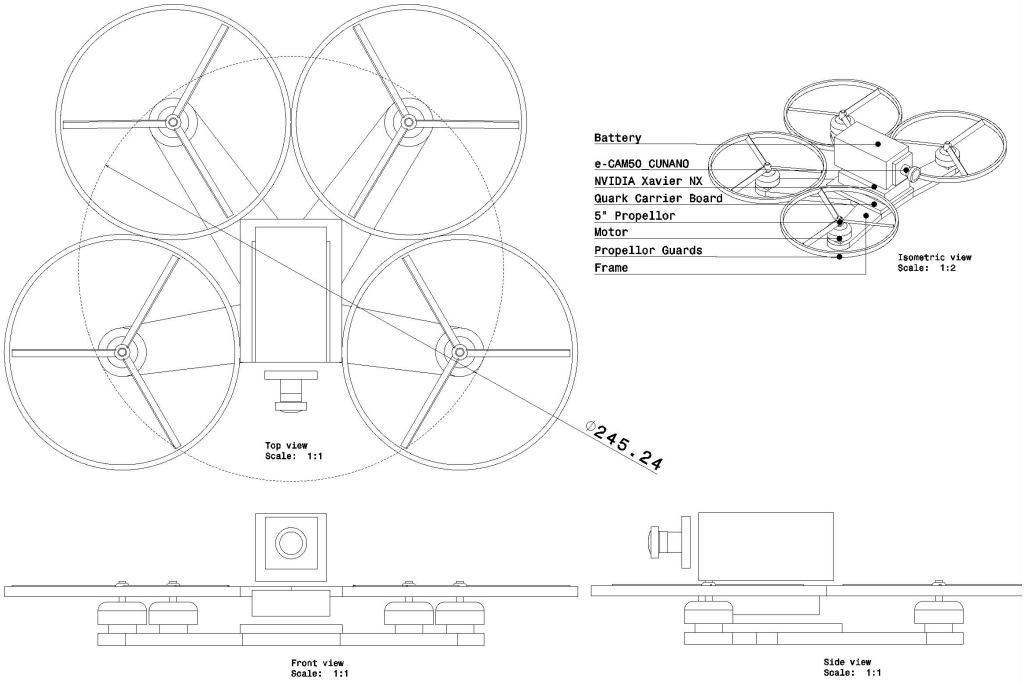


Figure 4.1: Render of the conceptual design of the DUR_C2 concept.

4.3. Technical Budget

To aid the further design process, a technical budget breakdown was made. This can be seen in Table 4.5. The system engineer has the role of checking whether or not subsystems stay within their budgets and to manage the budgets when subsystems are not able to comply with them. This is important since otherwise there is a significant chance of the design increasing excessively in mass or cost and thus not meeting the requirements anymore.

Table 4.5: Technical budget breakdown

Subsystem	Budgeted Mass (%)	Possible error range (+-%)	Budgeted Mass (g)	Budgeted Power (%)	Possible error range (+-%)	Budgeted Power (W)	Budgeted Cost (%)	Possible error range (%)	Budgeted Cost (€)
Structure	30	-	300	-	-	-	17	-	425
Frame	25	10	250	-	-	-	15	15	375
Cover	5	10	50	-	-	-	2	10	50
Aerodynamics	-	-	-	-	-	-	-	-	-
Electronics	17	-	170	16	10	48	42	-	1,050
Communications	2	10	20	1	10	3	2	10	50
Computer	15	10	150	15	20	45	40	20	1,000
Power and Propulsion	36.5	-	365	72	-	216	14	-	350
Battery	25	10	250	2	10	6	6	10	150
Propellor	2.5	10	25	-	-	-	2	10	50
Motor	9	20	90	70	10	50	6	10	150
Machine Vision	15	-	150	6	-	18	20	-	500
Sensors	15	10	150	6	10	18	20	10	500
Navigation	-	-	-	-	-	-	-	-	-
Control	-	-	-	-	-	-	-	-	-
Other	1	-	-	1	-	3	6.5	10	163
Production	-	-	-	-	-	-	6	10	150
LEDs	1	1	10	1	1	1.00	0.5	1	13
Total	100		1000	95		300	100		2,500

5

Technical Risk Assessment

The technical risk assessment is done in order to identify risks, which will be done in section 5.1. Furthermore, ways of mitigating the risks are shown in section 5.2. Keeping the risk mitigation in mind is a good driver for the design phase, as it gives a higher probability that the system is be able to perform its mission. Risk depends on two factors, the probability of it occurring and the impact it has when occurring. For the risk assessment, a ranking system is set up in which the probability will be given a rank based on the amount of flights it takes for a risk to happen. The rank of the impact will be based on the influence a risk has on the performance of the drone, if it is still able to perform its mission, but also the impact on resources, such time and money.

	Probability		Impact	
1	Very low	Over 10^6 flights	Negligible	Small inconvenience, no direct influence on performance.
2	Low	Between 1000-10^6 flights	Marginal	Little impact, small impact on performance and/or resources.
3	Moderate	Between 100-1000 flights	Significant	Mission success is harder, significant impact of performance and/or resources.
4	High	Between 10-100 flights	Critical	Mission success questionable, big impact on performance and/or resources.
5	Very high	Every 10 flights	Catastrophic	Mission failure, performance and/or resources fully compromised.

5.1. Risk Identification

In this section the most probable risks for each subsystem are identified and given a risk score. This is done mostly based on estimations of the probability and impact (indicated P and I in Table 5.1, respectively), but also based on experience from past drone races provided by the customer. As mentioned before, the impact is based on the influence a risk has on the mission, performance and resources. For example, a computer module costs around 200 times more than a single propeller, as seen in chapter 12. Therefore it will have a much higher impact.

Table 5.1: Identified risk with their drivers, probability, effect and impact.

ID	Risk	Drivers	P	Effect	I
		Risks regarding the Structures subsystem			
ST1	Breaking of structural components	Gate crash at maximum speed of 30 m/s	5	Structural component needs to be replaced, possible damage to on-board systems	3
ST2	Breaking of protective structural components	Drop from 3m height	3	Protective structural component has to be replaced	2
ST3	Breaking of load bearing structural components	Drop from 3m height	2	Structure cannot carry the full load, increased damage with next crash or drop, structure needs to be replaced	4
ST4	Failure of structural integrity over time	Loosening of part connections due to vibrations	2	Can badly influence the structural and flight performance of the drone	2
		Risks regarding the Electronics subsystem			
ES1	Breaking of the computer	Gate crash at maximum speed of 30 m/s Drop from 3m height Overheating	5	Broken computer needs to be replaced	5
ES2	Breaking of the camera	Gate crash at maximum speed of 30 m/s Drop from 3m height	5	Broken camera needs to be replaced	4
ES3	Breaking of the flight controller	Gate crash at maximum speed of 30 m/s Drop from 3m height	3	Broken flight controller needs to be replaced	3
ES4	Breaking of the ESC	Overheating Gate crash at maximum speed of 30 m/s Drop from 3m height	4	Broken ESC needs to be replaced	2

ES5	Cables overheat	Cables cannot withstand current profile over time	2	Can start a fire in the system	5
ES6	Connection to motors, ESC, FC, or carrier board are lost	Loose connector fit or improper solder connection	3	Loss of the specific system in flight resulting in a crash	4
ES7	Connection to WiFi module is lost	Loose connector fit	2	Telemetry to ground station is cut off	1
ES8	Connection to receivers is lost	Loose connector fit or improper solder connection	2	Manual takeover is not possible	4
ES9	Connection to camera is lost	Loose connector fit	2	The drone loses control or crashes	4
ES10	Electronics are damaged by static electricity	Build-up and discharge of static electricity	4	Electronic components on PCB's are damaged	4
	Risks regarding the Propulsion and Power subsystem				
PP1	Breaking of the motor	Overheating of the motor, due to overdesigning the propellers Gate crash	4	Drone cannot perform mission properly New motor is needed	3
PP2	Breaking of the propeller	Gate crash Spinning at too high RPM	5	Propeller needs to be replaced	2
PP3	Battery catching fire	Overheating of the battery Puncturing of the battery Short circuit	4	Possible damage to the whole system and surroundings. Battery and damaged systems need to be replaced	5
PP4	Lack of power	Too high DoD of the battery Efficiency of the connection between battery and output power too low Degradation of the battery	4	Reduction in performance Battery, replacement of motor or electrical components might be needed	3
	Risks regarding the Control and Stability				
CS1	Malfunctions due to singularity occurrence in Euler angle based algorithms	Using Euler angles	3	Control code malfunctions, drone is uncontrollable and is likely to crash	4
CS2	Malfunctions due to unforeseen control system errors	Faulty control code	3	Wrong instructions are outputted to Betaflight, drone likely crashes	4
CS3	Not receiving the required input at the nominal rate	Data transmission malfunction	2	Instructions are executed later then should be, drone might crash	3
CS4	Losing communication with Betaflight	Data transmission malfunction	2	Instructions are not executed, drone likely crashes	4
	Risks regarding the Machine vision, State estimation and Navigation				
MN1	Misidentified Gate	Multiple gates visible	5	Going in the wrong direction, course will not be completed	2
MN2	Unclear image	Drone rotating quickly	4	Next coordinate and velocity will be inaccurate, stacking up the inaccuracies might lead to a crash	3
MN3	Wrong gate location calculated	Gate is viewed from a very high angle	4	Going in the wrong direction, course will not be completed	3
MN4	Wrong path determination	Failure in Machine vision state estimation	5	Path chosen is not optimal Possible crash into obstacles	3

5.2. Risk Mitigation Strategy

A risk mitigation strategy is made in order to lower the risks that are mentioned in section 5.1. A good design approach can be made based on the mitigation strategy as it will point out necessary measures that have to be taken. The mitigated probability and impact are indicated MP and MI in

Table 5.2, respectively.

Table 5.2: Mitigation plan for the design phase, with an updated probability and impact if mitigation is successful.

ID	Risk	Mitigation strategy	MP	MI
		Risks regarding the Structure		
ST1	Breaking of structural components due to high speed gate crash	Design with protective parts, that either deform or are allowed to break, requires increased crushing zone with increased speed Design in a way the protective components will break and make them easy replaceable	4	2
ST2	Breaking of protective structural components during 3m drop	Design protective components flexible or strong enough to not break	2	2
ST3	Breaking of load bearing structural components due to 3m drop	Design structure strong enough to not break with the protective structure, lessening the impact	1	4
ST4	Failure of structural integrity of the drone over time	Regular maintenance can fully eliminated this risk	1	1
		Risks regarding the Electrical Subsystem		
ES1	Breaking of the computer	Design impact protection mechanism and landing gear in order to lower impact acceleration peak Implemented active cooling solution	3	5
ES2	Breaking of the camera	Design impact protection mechanism landing gears to lower impact acceleration peak Place camera inside the structure instead of having it stick out	4	4
ES3	Breaking of the flight controller	Design landing gears to lower impact acceleration peak	1	3
ES4	Breaking of the ESC	Design landing gears to lower impact acceleration peak Pick ESC with a margin on current rating	2	2
ES5	Cables overheat	Thicker cables are used	1	5
ES6	Connection to motors, ESC, FC, or carrier board are lost	Connectors on the carrier board are used with a locking mechanism and tight fit In the user manual it will be made clear to the user to ensure strong solder connections	2	4
ES7	Connection to WiFi module is lost	WiFi module is tightly connected on carrier board	1	1
ES8	Connection to the receiver is lost	Add redundant receiver that can shut down power to the propulsion system	1	2
ES9	Connection to camera is lost	Connector on carrier board and camera have locking mechanism	1	4
ES10	Electronics are damaged by static electricity	User is required to wear anti static wrist bands Carrier board is equipped with TVS and zener diodes	2	4
		Risks regarding the Propulsion and Power subsystem		
PP1	Breaking of the motor	Motor should be designed to deliver enough torque for the propeller. Propeller guards to reduce impact loads. Easy access to motor for fast replacement.	3	2
PP2	Breaking of the propeller	Easy access to propeller for fast replacement	5	1
PP3	Battery catching fire	Discharge rate of the battery has to be sufficient at maximum operational power Protection for and fastening of the battery so it stays in place does not get punctured	3	5
PP4	Lack of power	Set a minimum operational voltage to avoid a voltage drop.	2	3
		Risks regarding the Control and Stability		
CS1	Malfunctions due to singularity occurrence in Euler angle based algorithms.	No use of Euler angles which will fully eliminate the risk	1	1
CS2	Malfunctions due to unforeseen control system errors.	Control needs to be thoroughly tested on simulations	2	4
CS3	Not receiving the required input at the nominal rate.	Allocate a minimum amount of processing power to continuously run the control code	1	2
CS4	Losing communication with Betaflight.	Ensure a good wired connection	1	4

Risks regarding the Machine vision and Navigation				
MN1	Misidentified Gate	Use a RANSAC algorithm to lower chance of gate misidentification	4	2
MN2	Unclear image	Connect to IMU and use IMU for state estimation	4	1
MN3	Wrong gate location calculated	A better algorithm will result in a lower probability of this risk, eventually eliminating it. However, that is something that will be done by drone racing teams, for now this reduction seems feasible.	2	3
MN4	Wrong path determination	A better trained algorithm will always lower the probability, with the time available only a certain amount of training can be done. Therefore this reduction in probability seems feasible.	3	2

5.3. Risk Maps

The ranking of the risk identification Table 5.1 and the risk mitigation Table 5.2 can be mapped in a risk map, as is shown in Table 5.3 and Table 5.4. In the risk map the ”Red” section is where the risk is the highest, these risks should be avoided if possible. The ”Orange” section is still a high risk zone, these risks have to be mitigated as much as possible. The medium risks are found in the ”Yellow” section, these risks do not directly have a big impact on the system, therefore it is not necessary to take immediate action, but if the risk can be easily mitigated and lowered this is of course preferred. Finally, the risks in the ”Green” section are in an acceptable place, they will have almost no influence on the system, therefore no mitigation is needed. However, the same holds as for the medium risks, if it can be easily fixed and the risk will be lower, that is always preferred.

As can be seen in the risk maps if the mitigation strategy is fully executed a lot of the risks will get to the acceptable region. However, three of the risks stand out as they are still a high risk even after mitigation:

- ES1 - Breaking of the computer
- ES2 - Breaking of the camera
- PP3 - Battery catching fire

These are mainly based on the experiences of past drone races, in which it became clear how vulnerable the electronics are. This is especially the case when crashing into a gate at 30 m/s. The focus for the final design will therefore be on protecting the electronics. However, as will be shown in chapter 9 the forces the drone will experience during such a crash are actually too high for the system itself to handle, even with a lot of extra protection. Therefore, it might be necessary to consider other options in order to reduce these risks, such as lowering the speed to a value that can be more easily handled by the drone or using gates as crumble zones in order to reduce the loads on the drone.

Table 5.3: Risk map before the mitigation strategy.

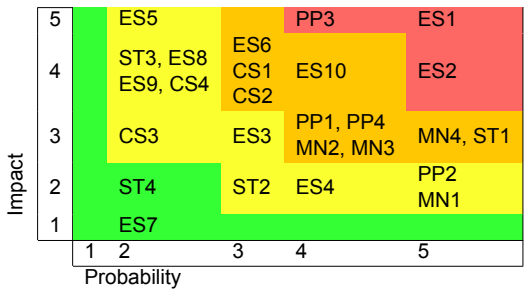
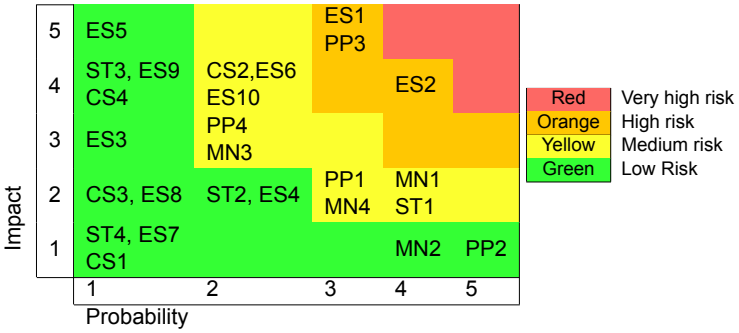


Table 5.4: Risk map after mitigation and expected risk distribution for the final design.



6

Sustainable Development Strategy

This chapter explains the implementation of sustainable design for the autonomous drone. Sustainability is about balancing the social, environmental and economic aspects of the design. First of all, section 6.1 describes the approach of how sustainability is implemented. Then, section 6.2 describes the sustainability criteria, and section 6.3 explains how they are implemented.

6.1. Approach to Sustainable Development

The way in which sustainability is implemented into the design is by creating a set of criteria to be used in the design process as a guide towards a sustainable design. Some of these criteria are used during the trade-off of concepts, and influence which design is chosen. The sustainability criteria are considered throughout the rest of the design process. Then, an evaluation is done for the final design in section 11.5, explaining how much these criteria influence the design and whether the sustainability can be improved. These sustainability criteria are created by conducting a life cycle assessment (LCA) of the drone. This LCA is done by assessing the sustainability impact of the drone through all phases of its life. The following phases are considered:

1. Raw material acquisition
2. Material manufacture
3. Product component manufacture
4. Final product manufacture and assembly (done by the consumer)
5. Product consumption
6. End-of-life phase

From the LCA, a set of sustainability aims are identified, which are then made into a more concise and measurable set of sustainability criteria. Each criterion is stated and its relation to the LCA is explained, as done in the next section.

6.2. Sustainability Criteria Description

The sustainability criteria are stated below, and it is explained how each criterion aims to improve the sustainability of the drone design. The criteria all come from the LCA, therefore, the phases of the drone's life which are influenced by the criterion is also stated.

Criterion 1: Estimated power consumption
Criterion 1 aims to improve the sustainability of the product consumption phase. The power consumption of the drone should be limited as much as possible to reduce the overall energy usage during the drone's operating life. The batteries are most likely charged using unsustainable electrical energy sources, such as coal or gas power plants.

Criterion 2: Volume of solid material
Criterion 2 aims to benefit the end-of-life phase. The amount of material of the waste product influences how much material will end up in landfills, or even how much energy is required to recycle and reuse the materials in the waste product. This criterion affects the first three phases: raw material acquisition, material manufacture, and product component manufacture. The reason is that the amount of material influences the energy consumed during these phases.

Criterion 3: Durability of design
Criterion 3 also aims to improve sustainability in the end-of-life phase, since a longer operational life of the drone leads to the postponement of waste generation. This ultimately leads to less waste being generated over a long time, as less drones are being used over this time period. This also benefits the other phases of the drone's life, since less drones need to be manufactured.

Criterion 4: Energy usage of processes during manufacturing phases
Criterion 4 concerns the material manufacturing, component manufacturing and final manufacturing phases. The energy usage of creating the materials and components, and assembling it all together depends on the processes used. These processes can often be determined by considering the materials used.

Criterion 5: Material waste during manufacture phases
Criterion 5 aims to improve the sustainability of the material manufacture, component manufacture and final manufacturing phases. The amount of waste material from certain processes such as machining should be minimized.

Criterion 6: Types of joints used for assembly
Criterion 6 aims to improve the end-of-life phase, where the drone should be able to be disassembled as easily as possible for reuse and recycling. Hence, the joints should be made temporary using, for example, bolts, as opposed to permanent joints such as adhesive bonding. This also makes it easier for single components to be replaced if broken, instead of replacing larger groups of components.

Criterion 7: Volume of biodegradable and recyclable material
Criterion 7 aims to improve the end-of-life phase. The more the drone consists of biodegradable material, the less material gets incinerated, or ends up in landfills. Biodegradable materials can end up being composted, while recyclable materials can be processed to be used again in another product. However, some materials are more difficult to recycle and it should be thoroughly considered how they can be used in the most sustainable way possible.

Criterion 8: Sustainability of component suppliers
Criterion 8 aims to improve the raw material acquisition, material manufacture, and product component manufacture phases. Suppliers should acquire their materials from sustainable sources, as well as manufacture their products in sustainable ways. This includes environmental aspects, as well as social aspects, such as fair treatment of workers.

Criterion 9: Location of component suppliers
Criterion 9 aims to benefit the final manufacturing phase. Although it is not part of manufacturing, it is about how the consumer obtains the components from suppliers, before starting the final manufacturing of the drone. The farther away the components are manufactured, the more energy is consumed for transportation costs. Therefore, it is beneficial if certain components are manufactured all over the world, such that consumers can obtain components with the least amount of transportation.

Criterion 10: Lifespan of components
Criterion 10 benefits the end-of-life phase, as fewer components break down, and hence, less waste is produced. This criterion is closely related to criterion 3: Durability of design, as the lifespan of components likely depends on the durability of the design to withstand impacts. However, even without impacts the components still have a limited lifespan, which would be beneficial to be as long as possible. This criterion also benefits the first three phases, as less components would need to be made.

6.3. Usage of Sustainability Criteria

The sustainability criteria are first used in the trade-off of concepts, as described in chapter 4. The criteria used in the trade-off are:

- **Criterion 1:** Estimated power consumption
- **Criterion 2:** Volume of solid material
- **Criterion 3:** Durability of design

Only three criteria are used, as not enough information is known about the concepts at the early stages. Each concept is scored on these three criteria. The criteria are given a weight based on how much they influence the sustainability of the design. All scores for each concept are summed, with their respective weight factors applied, resulting in an overall sustainability score of for each concept. The sustainability of criteria are also considered throughout the design, and then evaluated in section 11.5.

7

Power and Propulsion

The design of the power and propulsion system is based on the performance of the drone. When looking at racing drones specifically, high performance is crucial. The performance parameters are based on the requirements, which are discussed in section 7.1. The requirements alone do not say anything about the sizing of the power and propulsion system. Therefore, the requirements are evaluated and design parameters, for which the power and propulsion can be sized are determined for the most limiting case, which is explained in section 7.3. Then, section 7.4 discusses the methodology to solve for these design factors. The method is verified based on an initial sizing of the drone in section 7.5. Then, the results for the final iteration are given in section 7.6. Finally, these results are validated in section 7.7

7.1. Requirements

The design of the power and propulsion subsystem is based on requirements, which have been set up in the Baseline Report [**BaselineReport**], and which have been refined and updated in the Midterm Report [**MidtermReport**]. The related requirements are as follows:

- **AIG-TE-HW-MP-MO6**: During normal operations, the RPM of the propellers shall not exceed 270,000/diameter [inch].
- **AIG-TE-HW-PF1**: The drone shall have a maximum speed of at least 30 m/s.
- **AIG-TE-HW-PF2**: The drone shall be able to make a 3 meter radius arc while flying at 10 m/s.
- **AIG-TE-HW-PF3**: The drone shall be able to linearly accelerate at 19.61 m/s^2 from zero velocity.
- **AIG-TE-HW-P-E1**: The battery shall have a begin-of-life capacity of at least equal to (100% throttle power plus 100% computer power plus 100% power of all other systems) times 60 seconds plus 100% computer power times 120 seconds.
- **AIG-TE-HW-P-E2**: The maximum battery discharge rate shall be at least equal to the maximum use, meaning full throttle, full computational power and full power on all other systems in Watt.
- **AIG-TE-EB-M1.1**: The total mass of the propulsion unit subsystem shall be no greater than 115 g.
- **AIG-TE-EB-M2**: The drone shall be able to supply enough power to all subsystems for them to all function simultaneously.
- **AIG-NT-U3**: All materials used in the drone shall comply with the RoHS 1 and RoHS 2 restrictions on hazardous substances.
- **AIG-NT-R1**: The maximum speed of the drone shall be lower than 44.7 m/s.

It is important to note that requirement AIG-TE-HW-PF1 was originally removed in the Midterm Report [**MidtermReport**], since it was considered to be unfeasible. During the final design phase, it became apparent that this requirement was removed by mistake (based on flawed calculations). Throughout the following sections, this requirement shall be investigated in a more detailed manner, to give a more definitive answer on the predicted maximum speed.

7.2. Aerodynamic Characteristics

Many irregular surfaces in the proximity of four fast spinning propellers cause aerodynamics to be incredibly hard to predict precisely. Nevertheless, it can be useful to predict drag for making calculations on maximum speed, turning radius, and improving the accuracy of the control code. This chapter presents all reasoning used to estimate the drone's drag.

Calculating drag is done by splitting up the total drag into body drag and rotor drag, and determining both forms of drag separately.

$$D = D_{rotor} + D_{body} \tag{7.1}$$

Body drag is defined as the parasitic drag induced by the physical shape of the drone and is given by Equation 7.2. It is a function of the drag coefficient (C_D) and surface area (S) in the free stream velocity (V) with air density (ρ). Body drag holds a quadratic relation with velocity. Rotor drag, given by Equation 7.3, is caused by the propellers. There, m represents the mass of the drone and K is a constant. When the propellers are rotating in the same plane as the drone is moving, one half of the propeller always rotates towards the free stream velocity and the other half away [4]. This induces drag, which is linear with respect to velocity.

$$D_{body} = C_D \frac{1}{2} \rho v^2 S \tag{7.2}$$

$$D_{rotor} = Kvm \tag{7.3}$$

Body Drag

First, the body drag is calculated and then, the rotor drag in the next paragraph. Estimating the body drag accurately is extremely difficult, if not impossible without the use of computational fluid dynamics (CFD), which is outside of the scope of this project. An approximated solution is determined by splitting up the drone in multiple components and determining the drag characteristics of each one. It is important to note that these components are assumed to have no effect on the aerodynamics of other components. This assumption decreases accuracy but is necessary to make calculations feasible. In this case, the drone is split up into the body, four arms and four motors. The body is assumed to be a cylinder facing forward, the arms are cylinders attached to the body, and the motors are cylinders on top of the arm ends. This can be visualized as in Figure 7.1. Cylinders were chosen instead of three dimensional ellipses or other all-round continuous shapes because the corresponding drag coefficient represents reality more. The sharp edges on a cylinder's sides induce flow separation at higher velocities, just like the camera cut-out or other imperfections in the drone frame cause. A smooth 3D elliptical shape does not have this effect and therefore, yields a non-realistic drag coefficient.

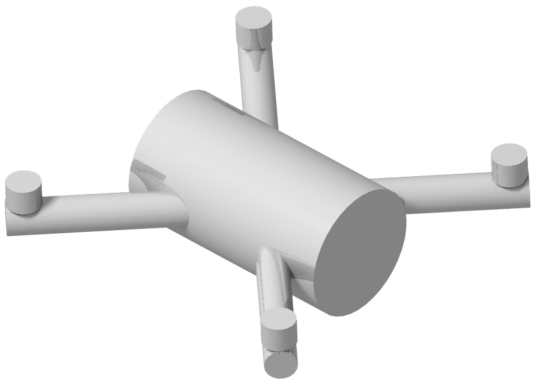


Figure 7.1: The simplified drone shape made out of 9 cylinders

The dimensions of these respective parts have been iterated several times based on the most recent values of the outer drone dimensions determined in chapter 9. An overview is shown in Table 7.1.

Table 7.1: The dimensions of the simplified drone parts used for determining drag

	Length [cm]	Diameter [cm]
Frame	18.0	11.6
Arms	10.0	2.9
Motors	2.1	2.8

The total body drag is assumed to be the sum of all component drags as seen in Equation 7.4.

$$D_{body} = C_{D_{frame}} \frac{1}{2} \rho v^2 S_{frame} + C_{D_{arms}} \frac{1}{2} \rho v^2 S_{arms} + C_{D_{motors}} \frac{1}{2} \rho v^2 S_{motors} \tag{7.4}$$

Here, S is the surface area normal to the airflow of the corresponding cylinder. This can be hard to calculate for an arbitrary air flow direction so a final simplification is made. Instead of the velocity being one vector with a certain orientation with respect to the drone, the velocity vector is split up into it's x, y and z component with respect to the drone coordinate system. Drag is thus determined separately for the x-axis, y-axis and z-axis of the drone. This simplification allows to easily estimate the drag coefficient of each cylinder and corresponding surface area.

Analysis is done on the drag coefficient for flow going parallel and normal to cylinders. The results are given in Table 7.2. For both cases, the reference surface area is the cross section area normal to the air flow. It is also indicated for which Reynolds numbers these are valid. For the AIgle drone, the operating Reynolds number is determined to be in the order of 10^4. Therefore, the results in the table yield a decent approximation.

Table 7.2: Drag coefficient for cylinders

	C_D	**Reynolds number**	**Source**
Flow along cylinder length	0.9	$> 10^3$	[5]
Flow along cylinder width	1.0	$10^2 - 10^5$	[6]

Having reviewed drag coefficients from a lot of different sources and for many different shapes, it is clear that these results are not universally agreed upon. One thing that is clear is the fact that all the determined drag coefficients, including the ones presented, are close to one. Therefore, from this point on, a drag coefficient of one ($C_D = 1.0$) is assumed for further calculations.

Having estimated the drag coefficient, equation Equation 7.4 can be used to determine the total body drag with respect to velocity for x, y and z directions. The surface area's needed to complete this formula can be determined from Table 7.1.

Rotor Drag

Rotor drag is given by Equation 7.3. The mass (m) of the drone is known and can be found in chapter 11. This value is iterated upon when changes occur. The only remaining unknown is the constant K. This value is specific to the drone and can only be accurately determined using real-life tests. For now, a value of $K = 0.5$ is chosen based on literature [7] [4].

With all the unknowns determined, the total drag can now be plotted against velocity, as can be seen in Figure 7.2. Additionally, the total thrust required to fly at this velocity is also plotted. The thrust is calculated by taking the force vector required to counteract drag and counteract gravity. The code used to achieve this result can be found in the public git folder of the project [1].

Figure 7.2 shows how the thrust force in a state of hover is equal to the weight of the drone and how it converges towards the drag force when flying at a high velocity. The required thrust can later be plotted against the maximum thrust to find the maximum velocity, as shown later in this chapter. Furthermore, the relation of drag and velocity is also used for predicting the turn radius and when writing the control code later on.

7.3. Performance Parameters

The approach for a design for optimal performance is that the most limiting requirement has to be found and should be the main driver of the design phase. As explained in subsection 7.3.1, the most limiting parameter that can be used for sizing the propulsion system is the thrust-to-weight ratio that comes from the turn requirement AIG-TE-HW-PF2. The design of the propulsion system is therefore

[1] https://vguillet.github.io/AIgle_Racer/, (Retrieved June 22, 2020)

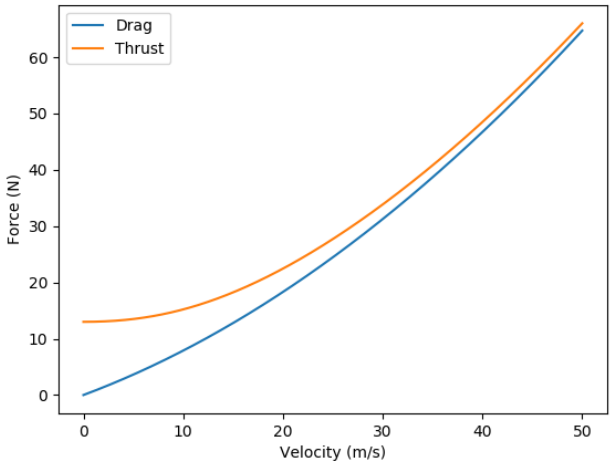


Figure 7.2: Drag and required thrust plotted agianst velocity

driven by the thrust-to-weight ratio. When the thrust-to-weight ratio is known, also the acceleration and speed can be calculated as shown in subsection 7.3.2 and subsection 7.3.3.

7.3.1. Turn Requirement

One of the most important parameters is the minimum thrust-to-weight. This is determined by evaluating all requirements related to flight performance. Through analysis, the most limiting manoeuvre is identified and occurs when turning in a 3 meter radius at 10 m/s in the horizontal plane (see requirement AIG-TE-HW-PF2).

The thrust-to-weight ratio is calculated using a point mass model, of which the free body diagram can be seen in Figure 7.3. In this diagram, the drone would be flying in x-direction and turning around a point in the y-direction.

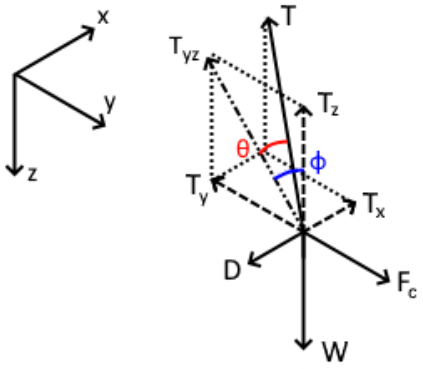


Figure 7.3: Free body diagram of point mass model for the turning performance requirement.

Since the turn is performed at a constant 10 m/s, there is no tangential acceleration. Equating the thrust component in x-direction and drag, Equation 7.5 is obtained. Here, the drag is assumed to consist only of rotor drag, since only this results in an explicit solution for the thrust-to-weight ratio, pitch angle and roll angle. The assumption leads to an underestimation of the final thrust-to-weight ratio, which is quantified later in this section.

$$T_x = D = 0.5m \cdot v \tag{7.5}$$

Then, since the manoeuvre happens in the horizontal plane, there is no acceleration in z-direction. Therefore, the thrust component in vertical direction equals the weight, as described in Equation 7.6.

$$T_z = W = m \cdot g \tag{7.6}$$

Moreover, the centrifugal force equals the thrust component in y-direction, as described by Equation 7.7.

$$T_y = F_c = \frac{m \cdot v^2}{r} \tag{7.7}$$

Finally, the thrust components can be combined to obtain the total thrust.

$$T = \sqrt{T_x^2 + T_y^2 + T_z^2} = m\sqrt{25 + (\frac{100}{3})^2 + g^2} \tag{7.8}$$

Dividing both sides of Equation 7.8 by $m \cdot g$, an explicit solution is obtained for the thrust-to-weight ratio. Furthermore, the pitch angle θ can be calculated using the cosine-rule as stated in Equation 7.9. By rewriting, θ is the only unknown and can be solved for, since the mass in each term can be divided out.

$$T_x^2 = T_{yz}^2 + T^2 - 2T_{yz} \cdot T \cdot \cos\theta \tag{7.9}$$

Lastly, the roll angle can be calculated with simple trigonometry with Equation 7.10.

$$\phi = \arctan\frac{T_y}{T_z} = \arctan\frac{\frac{m \cdot v^2}{r}}{m \cdot g} \tag{7.10}$$

Using r = 3 m, v = 10 m/s and g = 9.81 m/s^2, a thrust-to-weight ratio of 3.6, a pitch angle of 8.2° and a roll angle of 74° are obtained.

This model assumes a perfectly controlled drone. Since this is not likely in reality, it is important to have a margin on the thrust-to-weight ratio for imperfect control. It is assumed that the drone has a velocity of 10 m/s when entering the turn, and that the turning of the drone is delayed by 2 frames. Furthermore, the framerate of the camera is at least 60 frames per second (see requirement AIG-TE-HW-SV1.3), which is also the most limiting framerate for the delay. Due to this delay, the drone has to fly a turn with a radius of 2.7 m instead of 3 m. Using the equations described before, a thrust-to-weight ratio of 4.0, a pitch angle 7.4° and a roll angle of 75° are found.

With the final mass of the drone known, the earlier made assumption that drag consists only of rotor drag can be evaluated. Now, also taking into account the body drag, these equations become implicit and dependant on the mass, since the drag is now given by Equation 7.1 with K = 0.5.

Taking the final mass of the drone of 1.3 kg and including control margin, the obtained thrust-to-weight is still 4.0, the pitch angle is 8.8°, and the roll angle is 75.3°. This shows that the body drag has a negligible effect on the calculated T/W requirement.

Maximum tilt angle is another variable, which comes into play during this design process. It is related to the maximum speed that can be achieved. The maximum tilt angle is easily computed by determining at what angle the thrust vector is oriented in such a manner that its vertical thrust component (T_z according to Figure 7.3) is equal to the weight of the drone and T_y is equal to zero (since the drone is moving straight forward). In that case, the drone shall not ascend or descend, and the thrust component in the flight direction (T_x) is at its maximum. This maximum tilt angle is equal to 75°, computed using Equation 7.11. In this equation, T is the total thrust (not a component) and W is the weight of the drone.

$$\text{Maximum tilt angle} = arccos(\frac{T_z}{T}) = arccos(\frac{W}{T}) \tag{7.11}$$

It is important to note that originally it was misinterpreted that the thrust-to-weight requirement had to be met in worst dynamic condition. This dynamic condition is at 30 m/s (based on AIG-TE-HW-PF1) combined with a maximum tilt angle. It is the worst condition possible, since the thrust produced by a propeller decreases as the axial airspeed (perpendicular to the propeller) increases; this is at its maximum at the assumed top speed of 30 m/s and at a 75° tilt . These conditions are applied in subsection 7.4.4 up to subsection 7.4.7. However, this logic is the reason why originally (first iteration) the propulsion and power system was designed for a thrust-to-weight of 4 at a flight speed of 30 m/s

and tilt angle of 75°. Later on, it was realised that the thrust-to-weight requirement (determined by AIG-TE-HW-PF2) equals 4 and has to be met in 10 m/s at a pitch angle of 7.4° (these parameters are used for the final iteration). Using 30 m/s and a 75° tilt angle would lead to an over-design.

7.3.2. Acceleration Requirement

Requirement AIG-TE-HW-PF3 states that the drone must be able to linearly accelerate with 2g or 19.61 m/s^2. When drawing the free body diagram and kinetic diagram, which is done in Figure 7.4, two interesting directions of acceleration can be found - fully vertical acceleration and fully horizontal acceleration. The equations of motion are set up for those cases in Equation 7.12 and Equation 7.13, and in both cases the maximum acceleration is achieved when the drag is zero. Therefore, it is assumed that the conditions are ideal and the drone accelerates with no initial velocity so the acceleration is at maximum for a very brief moment.

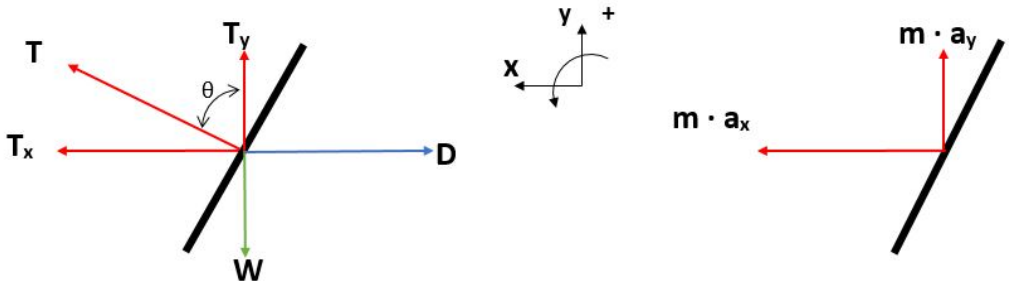


Figure 7.4: Free body diagram and Kinetic diagram of the acceleration

$$T_x - D = m \cdot a_x \tag{7.12}$$

$$T_y - W = m \cdot a_y \tag{7.13}$$

The thrust-to-weight ratio for the fully vertical acceleration is easily calculated because, when there is no acceleration in the horizontal direction, the thrust in that direction is also zero. Only the weight has to be countered with an acceleration of 1g. The thrust-to-weight ratio for vertical acceleration is then 3. For the horizontal direction however, it is a little more complicated since the drone has to stay in the air. Therefore, a part of the thrust goes into the vertical direction: $T_y = 1 \cdot m \cdot g$. With a thrust component equal to $T_x = 2 \cdot m \cdot g$, the maximum acceleration is achieved at a total thrust-to-weight ratio of 2.24 with a pitch angle $\theta = 63.4°$. So this is definitely not the most limiting case.

7.3.3. Maximum Forward Speed

For the calculation of the maximum forward speed, Equation 7.12 and Equation 7.13 can be used once again, together with the free body diagram and the kinetic diagram in Figure 7.4. The maximum forward speed is reached when the drone cannot accelerate anymore in the horizontal direction: $a_x = 0$ and $a_y = 0$. If that is the case, then $T_x = D$, in which $T_x = T_{req} \cdot sin(\theta)$ since the drone still has to stay in the air.

The curve for the required thrust as shown in Figure 7.5, is obtained from the drag calculations mentioned in section 7.2. It can be seen that for a speed of 30 m/s, a thrust of 39 N is required for these drag conditions. With a first estimated drone weight of 1.1 kg [**MidtermReport**], this leads to a required thrust-to-weight ratio of around 3.6. This is still lower than the thrust-to-weight required for the turn. Therefore, the speed is not the limiting requirement either.

7.4. Component Determination Methodology

The propulsion and power subsystem consists of 4 main components, namely the motors, propellers, battery and the electronic speed controller (ESC). The ESC choice is discussed in chapter 8. In order to ease the assembly/manufacturing process, the entire subsystem consists of off-the-shelf

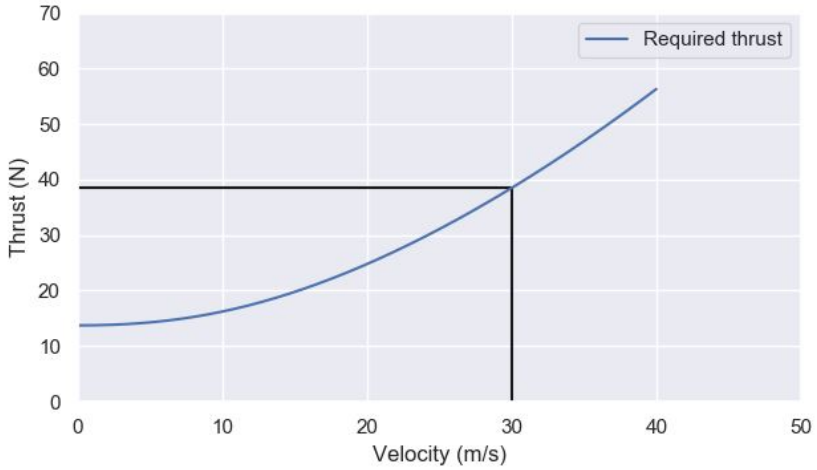


Figure 7.5: Drag over speed curve to determine the required thrust at 30 m/s

components. Consequently, this means that no component was designed by the team themselves but an extensive statistical matching process was done in order to obtain the optimal components. The goal is to design a robust propulsion system, which meets all of the requirements in dynamic settings. This process starts off by stating the assumptions made in order to make this approach work in subsection 7.4.1. Then, the system inputs are determined in subsection 7.4.2. Next, the databases are acquired, on which more information can be found in subsection 7.4.3. After this, the propellers are analysed, which is discussed in subsection 7.4.4. These propellers are then matched to suitable motors, as discussed in subsection 7.4.5 and subsection 7.4.6. Finally, the battery is added to this combination, as discussed in subsection 7.4.7.

7.4.1. Assumptions for Power and Propulsion

- Motor efficiency of 80% (including ESC). There is a difference in the given power values acquired from mini quad test bench and the propeller database from Advanced Precision Composites (APC) propellers. The motor database gives the electrical power used, and the propeller database provides the shaft power. To convert the shaft power to electrical power, an efficiency for the motor is used, which is estimated at 80% based on literature [8, 9, 10].
- The torque needed to spin the propeller is mostly influenced by the pitch and diameter of the propeller. Since the propellers used for testing of the motor database do not match with the propellers given by the propeller database, the only way to compare these databases is by making this assumption. This means that the drag caused by the airfoil and the shape of the tip are neglected. By doing that, a comparison can be made between propellers with the same diameter and pitch. This assumption creates uncertainty about the performance capabilities of the matched propeller and motor duo's. The use of a verification tool and validation attempts to tackle this uncertainty.
- It is assumed that the drone is flying at an altitude of 3 m, the temperature in the hall is 25 °C, and the pressure is at standard sea level pressure of 1013 hPa. These values are needed for the correct settings for eCalc. Other values for these parameters would give different results, mainly regarding the thrust and speed. However, it should be noted that changing of these parameters (e.g altitude of 30 m instead of 3 meters) does not have a significant effect on the results. The effects have a much bigger impact at very high altitudes (> 100 m) but this is not considered to be the operating range of the drone.
- Battery depth of discharge (DoD) is assumed to be at 80%. As the battery is nearly depleted, the voltage drops [11]. To ensure an adequate voltage is transferred to the requiring systems, this DoD is assumed.
- The C-rate provided by a battery's manufacturer is assumed to be not as reliable and therefore, is down-scaled slightly. This process, the down-scaling margin and logic is discussed further in subsection 7.4.7.

7.4.2. System Inputs and Important Parameters

The inputs required to determine the propulsion and power subsystem components are given in Table 7.3. Note that all inputs in this table were used for the first iteration and thus are all subjected to changes (as the design is iterated) with the exception of the thrust-to-weight, the flight and idle time, and the battery DoD.

Table 7.3: Propulsion & power system inputs required for component determination (first iteration)

Property	**Value**	**Unit**
Preliminary mass estimate	1.1 [**MidtermReport**]	kg
Thrust/weight	4	-
Flight speed	30	m/s
Maximum tilt angle	75	deg
Flight time (motors at 100%)	60	s
Idle time (motors at 0%)	120	s
I_{dry}	2	A
η_{motor}	80	%
$DoD_{battery}$	80	%

As mentioned in subsection 7.3.1, the flight speed and thrust that are inputted into the matching tool (for the first iteration) follow the logic mentioned at the end of subsection 7.3.1.

It should be noted that the flight time at 100% throttle was initially set at 5 minutes, but is renegotiated with the customer to 1 minute. This is because a flight time of 5 minutes increased the mass estimate too drastically. Furthermore, it should be considered that the flight time necessary to complete the previous AIRR races 2 and 3 (2019 AIRR season [2]), were 35 s and 12 s, respectively. In addition, it is logical to assume that during these races the drone's propulsion system does not operate at full throttle constantly due to the different maneuvers it performs. Thus, it can be said that the flight time of Table 7.3 should be sufficient to complete an entire race.

In order to obtain the optimal match of propellers and motors, comparison is done with the focus on Revolutions Per Minute (RPM) and specific thrust in static condition. Specific thrust is the ratio between thrust produced and power inputted into the motor, given in g/W. Thus, to some extent, it is related to motor efficiency. Therefore, it is an important motor parameter to take into account. This specific thrust is used in static conditions instead of dynamic conditions because motor behaviour in dynamic conditions is not readily available. Furthermore, it is a common convention for motor manufacturers to give this specific thrust instead of actual motor efficiency, and it is therefore used throughout this analysis. It shall become apparent why comparison of these two parameters (RPM and specific thrust) allows to provide strong matches between propellers and motors.

7.4.3. Database Acquisition

The method used for the selection of the power and propulsion system is based on three databases - one for propellers, one for motors, and the last one for batteries. In this section, a small description of the databases is given regarding where they are coming from, together with the methodology behind each database and why it was chosen.

Propeller Database: The full propeller database was acquired from APC propellers [3]. The propellers manufactured by this company are made from a long fiber composite designed by APC, which is stronger than the standard glass-filled nylon propellers. Therefore, the propellers can be relatively thin, which is beneficial for the mass and aerodynamic performance. The airfoil used for their Thin-electric and Multi-rotor propellers is an Eppler E63 airfoil on the inside of the propeller, and an airfoil similar to Clark-Y closer to the tip. The calculations that are done by APC are based on the vortex theory as described by [12], giving the thrust and shaft power, including their coefficients, together

[2] https://thedroneracingleague.com/airr/,(RetrievedJune29,2020)
[3] https://www.apcprop.com/, (Retrieved June 22, 2020)

with propeller efficiency for different RPM and free stream speeds. For the lift-drag data, NASA's TAIR (Transsonic AIRfoil [13]) code was used by APC, which is best used for high speed conditions. Therefore, there might be some inaccuracies for low speed flying conditions. This database and therefore, this propeller type is chosen over other propeller types, because the top 4 fastest drones in the world right now also use APC propellers [4]. Finally, only the 5 and 6 inch propellers are evaluated, again since these are the most used propeller sizes for high speed drones. The database contains 25 types of 5 and 6 inch propellers, where every propeller was tested in approximately 35 different RPM settings and for every RPM setting, 30 different flight speeds were analysed. This gives a total of $25 \cdot 35 = 875$ data points when analyzing a particular flight speed.

Motor Database: The motor database was acquired from test bench results provided by Miniquad Test Bench (MQTB) [5]. It consists of approximately 300 different motors, each motor was tested at least 5 times with different 5 and/or 6 inch propellers. Summed up, this accounts for over $300 \cdot 5 = 1500$ data points. It is important to note that for these motor tests, other propellers were used than the APC propellers. MQTB do their testing on a standard Turnigy thrust test set-up, with some additional sensors to measure voltage, current and RPM, driven by a code optimised by themselves [6]. This is generally known as a trustworthy method to test the static thrust a motor can deliver since it is also supported by big motor manufacturers like T-Motor. Therefore, it is deemed to be good enough for this design phase. The values in this database with regard to power, current and voltage are subject to efficiency losses due to wiring and the ESC, which is used during testing.

Battery database: The battery database is based on seller data [7]. This is not always the most trustworthy data, especially with regard to the C-rating of the battery. A constant C-rating above 100C is highly unlikely. For example, when a battery with a capacity of 2200 mAh has a C-rate of 100C, it can constantly provide a current of I = 220 A. When the internal resistance of a battery is then around R = 6 mΩ, it would convert $P = I^2R =$ 290 W of power into heat, which cannot be dissipated fully, potentially resulting in overheating and damaging the battery. A method to handle this problem is described in subsection 7.4.7. This database consists of a total of 146 batteries of 4S, 5S and 6S volt class. Every 'S' stands for a nominal voltage of 3.7 V, thus e.g. a 4S battery provides a nominal voltage of $4 \cdot 3.7 = 14.8$ V. Only these volt classes (4,5 & 6S) are examined since they match the voltage required by the motors in the motor database.

All in all, the databases used are considered big enough in size and consist of enough recent data to select the best components for the propulsion and power subsystem.

7.4.4. Propeller Selection

After setting up the propeller database, the first step is to filter out the propellers, which do not meet the thrust requirement in dynamic conditions. In order to limit over-designing, a top limit is also imposed; the bottom limit remains equal to the limits mentioned in Table 7.3. Analysis of the propeller database shows a decrease in thrust and increase in shaft power as flight speed increases. Furthermore, as the tilt angle increases, so does the axial speed hitting the propeller perpendicularly. Thus, at a tilt angle of 75° and a maximum flight speed of 30 m/s, propellers producing thrust of at least 1089.0 grams up to 1237.5 grams at an axial flight speed of ≈29 m/s are chosen from the propeller database. This analysis is conducted twice, one time for 5 inch APC propellers and one time for 6 inch APC propellers.

The next step is to obtain the static specific thrust of the adequate propellers by using Equation 7.14. In this equation, the static thrust is divided by the dynamic shaft power opposed to the static shaft power because this incorporates a design margin. Furthermore, the motor efficiency is assumed to be 80%, as mentioned in subsection 7.4.1.

$$\text{Specific thrust} = \frac{\text{Thrust}_{\text{static}}}{\text{Shaft power}_{\text{dynamic}}} \cdot \eta_{motor} \tag{7.14}$$

[4] https://fpvdronereviews.com/guides/fastest-racing-drones/, (Retrieved June 22, 2020)
[5] https://www.miniquadtestbench.com/, (Retrieved June 22, 2020)
[6] https://github.com/VirtualEnder/TestStandv2, (Retrieved June 22, 2020)
[7] https://www.banggood.com/, (Retrieved June 22, 2020)

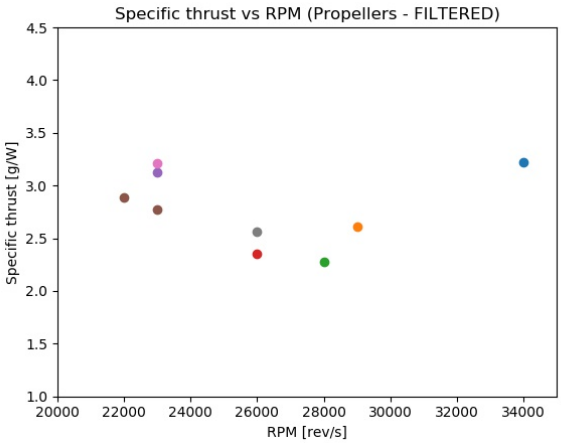


Figure 7.6: Filtered 6 inch APC propeller database, for parameters of Table 7.3

Plotting the filtered database with regard to specific thrust and RPM for a certain minimum thrust (1100 grams) and flight speed (30 m/s), yields Figure 7.6, which is used later on in matching exercise.

7.4.5. Motor Selection

This data set is also filtered using a bottom and top thrust limit. However, since the motors are only tested in static conditions, the dynamic thrust limits (used in subsection 7.4.4) have to be converted to static requirements. The dynamic thrust limits are multiplied by the average of 'Factor A', which is based on the static and dynamic thrust given in the propeller database, as can be seen in Equation 7.15. This is done for all the propellers passing through the filter discussed in subsection 7.4.4 and the resulting value is ≈1.55.

$$\text{Factor A} = \frac{Thrust_{static}}{Thrust_{dynamic}} \tag{7.15}$$

Ultimately, this means that those motors are kept that should be able to provide enough thrust to meet the dynamic thrust requirement in dynamic conditions. After applying this filter, the remaining motors often have multiple data points (due to motors being tested with multiple propellers). In order to take into account the full potential of every motor, the motor and tested propeller combination providing the highest specific thrust is kept. However, one should keep in mind that the values for the same motor tested with different propellers never differ substantially. Once again, this process is done twice, once for all motors tested with 5 inch propellers, and once for motors tested with 6 inch propellers. Plotting all remaining motors with regard to RPM and specific thrust, for a certain minimum thrust and flight speed, yields Figure 7.7.

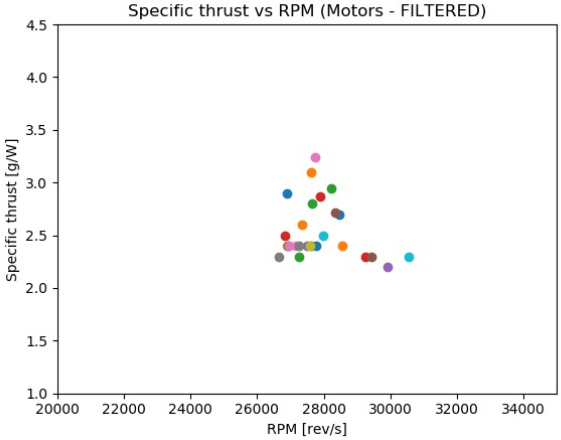


Figure 7.7: Filtered 6 inch motor database, for parameters of Table 7.3

7.4.6. Motor and Propeller Matching

After both motor and propeller databases have been filtered, the next step is to search for optimal duo's (= motor + propeller). The goal of this step is to fit an APC propeller (with known dynamic performance) to a suitable motor. Ultimately, this should obtain a duo which meets the propulsive requirements in dynamic conditions. This process begins by combining Figure 7.6 and Figure 7.7, and examining if potential duo's are even present at all for these dynamic conditions. The result for a certain minimum thrust and flight speed is visible in Figure 7.8.

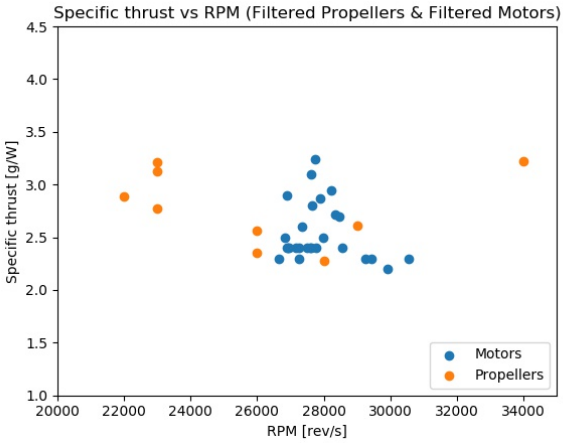


Figure 7.8: Filtered 6 inch motor & propeller databases, for parameters of Table 7.3

The most important player in the duo's is the propeller. This is because if it is operating in the right conditions (shaft power, RPM and torque), it should be able to produce exactly the desired thrust in dynamic conditions. In order to ensure that these conditions are possible, a motor should be chosen with higher RPM and specific thrust compared to the propeller. However, the motor's RPM should be as close to that of the propeller as possible (<1000 RPM difference), to ensure the motor operates at a high efficiency. This is a logical step to take since motors are designed for a certain purpose and setting. Thus, if a motor is designed for, for example, an RPM of 30000, forcing it to operate at 20000 RPM would never yield the highest motor efficiency. On the other hand, the motor's specific thrust should be as high as possible since this is an indication of its efficiency. Another important reason for this is related to Equation 7.16. How duo's are matched with regard to torque in this equation is mentioned in the following paragraph.

$$\text{Power}_{\text{Shaft}} = \text{Torque} \cdot \text{RPM} \tag{7.16}$$

During flight, the propeller requires a certain amount of torque to achieve the desired performance, the motor should be able to supply this. The RPM's should match closely, yet the motor's RPM should be higher than that of the propeller. In addition, the power delivered to the shaft by the motor should be enough and should be higher. The ratio (between motor and propeller) of the difference in input power should be larger than the ratio of difference of the RPM. This ensures the torque delivered by the motor is in fact higher than required by the propeller instead of lower, according to Equation 7.16. Once again, the input power should not exceed the required amount significantly, to avoid a loss of motor efficiency when operating at a lower power setting. In order to ensure this, the APC propeller dynamic input power and motor input power (using the dynamic transformed static requirements) are checked for compliance manually. In this case, compliance means motor input power is higher than propeller dynamic input power required. As a result, it can be confidently said that the motor is able to provide the torque required in dynamic condition (i.d. the most limiting power condition). All in all, this should provide the optimal duo's for our drone. In general, the matching tool provides multiple duo's for the given inputs. In order to identify the best duo from these matches, their properties are examined manually with regard to closeness of their (propeller and motor) respective RPM's and the motor's specific thrust relative to that of the propeller. The match with the highest specific thrust is

chosen as the 'best' duo. Through this process, the best performing motor for this case is chosen as a match for the propeller.

7.4.7. Battery Selection

After setting up the battery database, suitable batteries are kept that have sufficient capacity and available current. The thresholds to filter out the non-suitable batteries are given in Equation 7.17 and Equation 7.18.

$$I_{required} = \#\text{ of motors} \cdot I_{motor} + I_{dry}, \tag{7.17}$$

where I_{dry} is the total current used by all components other than the propulsion system e.g. the processor, given in Table 7.3. Then, I_{motor} is the current of the motor (during testing and at 100% power) for the given duo, as mentioned in subsection 7.4.6.

$$\text{Battery capacity required} = \frac{(I_{required} \cdot t_{flight}) + (I_{dry} \cdot t_{idle})}{3600 \cdot \text{DoD}_{battery}} \tag{7.18}$$

Here, $I_{required}$ and I_{dry} are given in Equation 7.17, with t_{flight} as the flight time at 100% motor power and t_{Idle} is the idle time at 0% motor power, both given in Table 7.3. The battery DoD is set at 80%, as mentioned in subsection 7.4.1.

Furthermore, the motor in every duo is evaluated with regard to its volt class (4, 5 or 6S). An important thing to note is that the efficiency losses with regard to wiring and the ESC are incorporated in this determination of the thresholds, since the observed voltage during testing of the motors in the motor database is consistently lower than the voltage supplied by battery of similar volt class. In addition, the efficiency losses of the ESC and wiring are minimal. This has to be the case to avoid overheating of said components. Thus, by identifying the correct volt class of every motor in a duo and using this to find suitable batteries, these efficiency losses are taken into account.

For the batteries in the battery database, the available current is often not given and therefore, has to be calculated using Equation 7.20. However, considering that the C-rate provided by the battery manufacturer is less reliable, the given C-rate is down-scaled using

$$\text{Down-scaled C-rate} = \text{Given C-rate} - [(\text{Given C-rate} - \text{Average C-rate}) \cdot 0.20], \tag{7.19}$$

where the 'average C-rate' is calculated for every volt class (4, 5 and 6S) separately. The value 0.20 was chosen through trial-and-error, to ensure only the best performing batteries would be kept in the database.

$$I_{available} = \text{Down-scaled C-rate} \cdot \text{Battery capacity} \tag{7.20}$$

All in all, the batteries meeting the required current, required capacity, and match in volt class are kept for every duo. Eventually, out of these suitable batteries, the most light weight option is chosen to be added to the duo, in order to minimise the weight of the configuration.

7.4.8. Ducted Fans Consideration

Ducted fans are considered as a means to increase the maximum thrust, or reduce the power consumption for a given thrust. This section elaborates on potential benefits of ducted fans but more importantly, it also explains why they are not used for the final design.

The main thought behind ducted fans is to reduce tip vortices. This is achieved if the ducted fan fits sufficiently snug around the tips of the propellers so there is no room for air to go around the tip of the propeller. Reduced tip vortices cause the pressure underneath the propeller to increase, and above the propeller to decrease, yielding more thrust. Another side effect that is important to consider is the coanda effect. This effect describes how fluids have the tendency to stick to curved surfaces when moving parallel to them. A main advantage of ducted fans is achieved when a semicircular shape is put on top of the cylindrical duct, as can be observed in Figure 7.9. Using the coanda effect, this guides air to the propeller. Bernoulli's law says that moving air causes a lower pressure and thus, the air moving over this semicircular section induces a low pressure region, resulting in extra lift. A

problem occurs however when flying at high velocities. In this case, a large velocity (and thus very low pressure) region is observed on the semicircular section closest to the airflow, as represented in Figure 7.9. At lower velocities, higher pressure is seen on the other side of the duct. This results in the drone going back to horizontal position. In other words, the drone is likely over-stable and might have a big problem executing very agile manoeuvres.

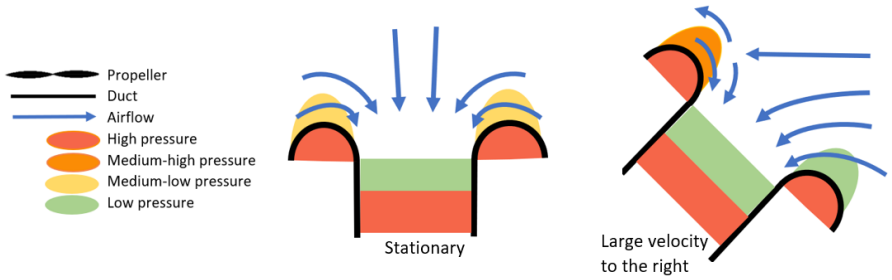


Figure 7.9: Air flow over ducted fans, stationary and at large velocities.

One can also consider a ducted fan without semi-circular additions. There are reports claiming that this would still yield an efficiency increase of 5-10%. However, this does not appear to be a universal conclusion. The number of results accessible online is not sufficient to justify this choice for the final design. Detailed CFD analysis could be used to make conclusions but falls outside of the scope of this project. Therefore, no ducted fans are considered. Propeller guards are however still being used with the purpose of structural protection, as discussed in chapter 9.

7.5. Matching Tool Verification

In order to allow for rapid and frequent iteration, the matching tool based on the motor, propeller and battery databases has to be verified. If it can be proven to work correctly, the outputted triplets (motor, propeller and battery) will consistently adhere to inputs given to the matching tool mentioned in section 7.4. Changing these inputs (iteration) shall allow for optimisation of the final design. To verify the matching tool, the online simulator eCalc [8] is used. This online simulator has a tool specifically designed for multi-copter drones to check their flight performance.

7.5.1. Verification Triplets

Multiple triplets/ configurations are obtained by running the tool at dynamic maximum speed conditions and thrust required in this condition: 30 m/s at 75° tilt angle and T/W of at least 4.0 . In order to illustrate the robustness of the matching tool, these triplets, with both 5 and 6 inch propeller types, are inputted into eCalc and compared to the inputs of the matching tool. Other than the triplets changing themselves, all other inputs are kept constant for a certain propeller type. These inputs are given in the list below:

- Model weight is estimated at 250 grams and model size estimated at 250 mm diagonal for 5 inch triplets. For 6 inch triplets, this is 280 grams and 280 mm diagonal. [**MidtermReport**]
- Tilt limit is 75° and field elevation is 3 meters (sea level), which complies with Table 7.3.
- The battery has a DoD of 80% and is fully charged, which complies with Table 7.3.
- Payload weight and current (= I_{dry}) is 185 grams and 2 Amperes respectively, which complies with Table 7.3 [**MidtermReport**].
- The ESC is a 4in1 variant of ≈18 grams and can handle 45 Amperes per motor.

All eCalc inputs related to weight (model and payload) are based on the total and component estimated weight obtained during the Midterm report [**MidtermReport**]. The specific motors, propellers and batteries obtained from the matching tool are inputted directly (without adjustments) into eCalc. The exact triplets are given in Table 7.4.

[8] https://eCalc.ch/, (Retrieved June 22, 2020)

Table 7.4: Top 5 triplets (5 & 6 inch) inputted into eCalc for verification purposes

	Propeller	Motor	Battery
5 inch Triplet #1	APC MultiRotor MR 5.25” x 4.75”	Cobra CP-2207 (2300 kv)	LiPo 3000mAh - 65/100C - 4S
5 inch Triplet #2	APC Electric E 5” x 5”	T-Motor F60 PRO III (2500 kv)	LiPo 3000mAh - 65/100C - 4S
5 inch Triplet #3	APC Electric E 5.5” x 4.7”	iFlight XING-E X2306 (2450 kv)	LiPo 4000mAh - 55/80C - 4S
6 inch Duo #1	APC MultiRotor MR 6” x 3”	Cobra CP-2207 (2450 kv)	LiPo 3000mAh - 65/100C - 4S
6 inch Duo #2	APC MultiRotor MR 6” x 4”	iFlight XING-E X2306 (2450kv)	LiPo 4200mAh - 55/80C - 4S

7.5.2. Verification Results and Discussion

The results from comparing the eCalc results to the matching tool inputs mentioned in subsection 7.5.1 and Table 7.3 are given in Table 7.5. Errors are either relative or absolute, given in the second column. All relative results are compared to the outputs of eCalc using Equation 7.21.

$$\text{Relative error} = \frac{\text{own data} - \text{eCalc data}}{\text{eCalc data}} \cdot 100\% \tag{7.21}$$

Table 7.5: Top 5 triplets (5 and 6 inch) inputted into eCalc: comparison results

Property	Error	5 inch Triplet #1	5 inch Triplet #2	5 inch Triplet #3	6 inch Triplet #1	6 inch Triplet #2
Total weight	Relative [%]	11.9	12.5	2.3	8.6	-2.4
Thrust/Weight	Absolute [-]	0.10	-0.20	-0.50	-0.70	-0.84
Max tilt angle	Absolute [°]	2.0	1.0	0.0	0.0	0.0
Flight time	Absolute [s]	-12.0	-0.2	-8.4	2.0	-9.1
Motor efficiency	Absolute [%]	-1.5	-4.3	-0.1	-2.4	1.0

In this table, the flight time takes into account both the flight time at 100% power and in idle setting. The reason to give some errors in absolute sense is that this provides a clearer overview for the differences between the matching tool and eCalc than it would give for the property’s relative error. When examining Table 7.5, in general, the following can be concluded.

- The mass estimate inputted into the matching tool is overestimated compared to eCalc (total weight).
- The thrust-to-weight requirement is met with a margin (T/W > 4).
- The maximum tilt angle can not always be fully realised but is very close.
- The flight time requirement is met with a margin.
- Motor efficiency inputted into the matching tool is underestimated.

Although the matching tool is not a complete match with the eCalc tool, the errors seen between the two are not too significant. Furthermore, in general, the values of eCalc exceed the inputs of the matching tool (based on the requirements). Thus, even if differences are more significant in magnitude, the propulsion and power requirements shall be met regardless. Furthermore, this type or verification research was conducted multiple times before and after this set, and never showed significantly higher errors. The remaining under- and over-estimations are also not significant enough to be tweaked in the matching tool, with the exception of the mass estimate. The total mass estimate (Table 7.3) is changed during iteration to find to closest match for that drone mass. Therefore, it can be said that the matching tool is verified to an extent, with generally no consequences for requirement compliance.

7.6. Power & Propulsion System Results (Final Iteration)

After the matching tool is verified, it proves iteration is possible and can consistently provide an adequate power and propulsion system for the drone. Through iteration, Table 7.3 is kept the same since it is based on the requirements, yet the flight speed and mass estimate are changed. The mass estimate is changed from 1100 grams to 1286 grams. This in turn means that the eCalc inputs, model weight and payload weight (including ESC) are changed to 536 grams and 250 grams, respectively. The flight condition which was originally set at 30 m/s and 75° tilt angle is set to only 10 m/s and 7.4° tilt angle because a thrust-to-weight of 4 at 30 m/s is not required and leads to over-design (this logic

is also mentioned in subsection 7.3.1). The configuration/ triplet outputted from the matching tool for these conditions is given below:

- **Propeller**: APC 6x4E [9]
- **Motor**: SunnySky Edge R2305 (2300 kv) [10]
- **Battery**: Tiger Power: 3600 mAh battery (65 C-rate) - 4S class [11]
- **ESC**: T-Motor F45A V2.0 [12], elaborated on further in subsection 8.3.4

The results with regard to flight performance are given in Table 7.6. This table also contains comparison between eCalc and the matching tool's parameters. Errors are all relative and have been obtained using Equation 7.21. All in all, it is clear that this configuration complies with the requirements mentioned in section 7.1, in both static and dynamic conditions.

Table 7.6: Final configuration results

Property	Matching tool's parameters	eCalc parameters	Error [%]
Total weight [grams]	1286	1294	-0.6
Thrust/Weight [-]	4.3	4.1	4.9
Flight time [s]	63	66	-4.5
Motor efficiency [%]	80	86.6	-7.6

In this table, the flight time takes into account both the flight time at 100% power and in idle setting, combining both duration into one flight time at 100% power. Furthermore, the eCalc indicated hovering time is equal to 9.6 minutes. The motor efficiency is the motor efficiency at maximum setting. The difference of 8 grams in total weight between the matching tool and eCalc was investigated and is most likely due to an overestimation of the wiring by eCalc. In reality, it is logical to assume the mass estimate is closer to 1286 than 1294 grams when examining Table 11.1 since the subsystem total masses of this table are inputted exactly into eCalc. In addition, it can be said that if eCalc used a closer lower mass estimate of 1286 grams, the thrust-to-weight indicated would increase as a consequence (from 4.1 to ≈4.2). Thus, this does not affect this flight performance parameter. In all further sections, a thrust-to-weight estimate equal to 4.1 shall be used since it is the lower estimated value.

Furthermore, the RPM and current of this motor and propeller duo indicated by eCalc is equal to 27509 revolutions per second and 40.03 Amperes, respectively. The voltage and the mechanical shaft power at maximum setting indicated are equal to 13.18 V and 456.6 W, respectively. These values cannot be verified due to the lack of comparative data since in the matching exercise only predictions of optimal motor and propeller duos are made; their combined properties in dynamic conditions with regard to e.g. power or RPM cannot be estimated properly. Although the eCalc values are only an indication, selection of e.g. the ESC shall occur by using the eCalc value for the current plus a margin. The same is done with the RPM to design for sufficient structural integrity to handle vibration loads. In the future, the design's propulsion and power system could be reiterated further, after more in-depth analyses have been conducted with regard to the mass estimate, using the code and methodology described in the previous sections.

When applying the calculation for the acceleration from subsection 7.3.2, with a total thrust-to-weight of 4.1 , a maximum linear acceleration of 3.98g is achieved in horizontal direction. As mentioned, this is the initial acceleration, after the drone gains some speed it starts experiencing drag, which lowers the acceleration.

Using the method mentioned in subsection 7.3.3 to calculate the thrust-to-weight based on the speed the other way around, the maximum speed can be calculated. When using a constant thrust-to-weight

[9] https://www.apcprop.com/product/6x4e/, (Retrieved June 21, 2020)
[10] https://sunnyskyusa.com/products/r2305, (Retrieved on June 21, 2020)
[11] https://www.banggood.com/nl/Tiger-Power-14_8V-3600mAh-65C-4S-Lipo-Battery-XT60-Plug-for-RC-Model-p-1583272.html?rmmds=detail-left-hotproducts__2&cur_warehouse=CN, (Retrieved June 21, 2020)
[12] http://store-en.tmotor.com/goods.php?id=899, (Retrieved June 20, 2020)

ratio of 4.1 with a weight of 1286 grams, the total thrust is around 52 N. Equating this value to the drag, a speed of around 38.9 m/s is found. However, when the speed gets higher, the thrust the propeller can deliver becomes lower, which is not taken into account with this method. Luckily, the propeller database provides the thrust at different speeds. When plotting this thrust in the same graph as the drag, as shown in Figure 7.10, a speed of 32.1 m/s is found, which seems more reasonable.

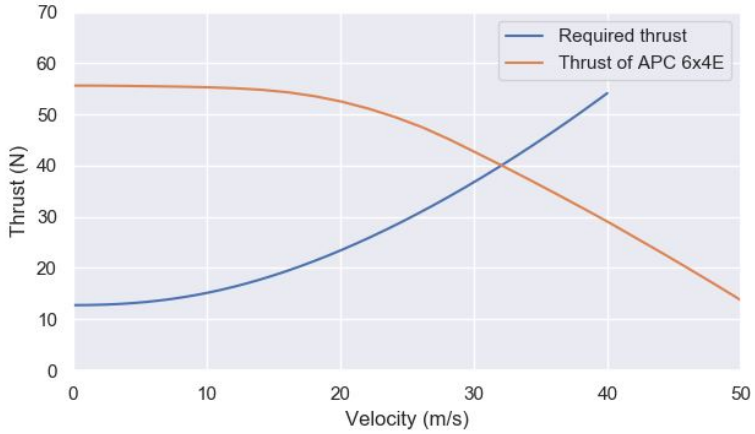


Figure 7.10: The thrust provided by the APC 6x4E propeller versus the required thrust over speed

The camera is pitched 33.5°upwards (described in section 9.10). The cruise speed of the drone is encountered when the camera's center of focus is oriented exactly into the flight direction (relative orientation of the camera's pitch angle = 0°). In this setting, a speed of 9.1 m/s can be achieved in the flight direction; this value is computed using a similar method to subsection 7.3.3. However, in this case, the magnitude of the T_x component (in flight direction) is computed for a pitch of 33.5°, after which the intersection with the drag curve is obtained.

7.7. Validation

The most important parameter driving the design of the power and propulsion subsystem is the thrust-to-weight ratio. The limiting thrust-to-weight ratio given by the turn requirement AIG-TE-HW-PF2 has to be met in dynamic conditions. To check whether eCalc gives the dynamic or static thrust-to-weight ratio, over a hundred motor-propeller combinations out of the motor database from MQTB are plugged into eCalc. Only motors and propellers that can be found in eCalc are chosen, in order to get the most reliable results. Then, the trust-to-weight ratio found by eCalc is compared to the static thrust-to-weight ratio given by motor database. This gives a relative difference between those two T/W ratios and for every combination, this difference is shown in Figure 7.11 in the form of a histogram. Then, a normal distribution is fitted to this histogram and, as can be seen, the mean of those differences is $\mu = 0.66$, with a standard deviation of $\sigma = 0.044$. This value of μ for the difference between the static and dynamic thrust-to-weight ratio is almost the same as the inverse of 'Factor A', $\frac{1}{1.55} \approx 0.65$, which was earlier assumed to be the difference between static and dynamic thrust, as described in subsection 7.4.5. Thus, it is safe to assume eCalc gives a thrust-to-weight ratio in dynamic condition, as well as the matching tool. Therefore, the required thrust-to-weight to make the turn is indeed met in dynamic conditions.

Another aspect that is not yet validated is the speed. When using the method as described in subsection 7.3.3 with the APC 6x4E propeller, a speed of 32.1 m/s is found. To validate this speed, the thrust-to-weight ratios of existing drones are plotted in Figure 7.12, together with the speeds those drones were able to reach [13] [14]. For all these points, an exponential regression line is fitted, in which the thrust-to-weight ratio can be filled in, in order to get the speed. The T/W ratio that is needed for this calculation is the static T/W, which can be obtained by using the difference mean μ. This leads to $T/W_{static} = \frac{T/W_{dynamic}}{\mu} = \frac{4.1}{0.66} = 6.2$. When filling in this T/W into the exponential function, a speed

[13] https://fpvdronereviews.com/guides/fastest-racing-drones/, (Retrieved June 20, 2020)
[14] https://klsin.bpmsg.com/how-fast-can-a-quadcopter-fly/, (Retrieved June 20, 2020)

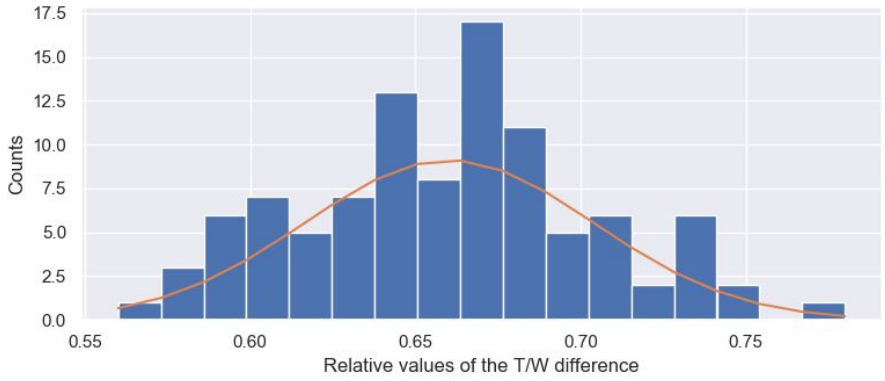


Figure 7.11: Normal deviation of the difference between T/W ratios from eCalc and MQTB

of 29.8 m/s is found, which is around 7.2% lower than the calculated speed of 32.1 m/s.

There are three main factors that can explain this difference. First of all, the drag estimation can be off since this is a very rough estimation that is not verified. Secondly, as mentioned in subsection 7.4.3, the TAIR code used in setting up the propeller database is best used for high speeds, and since this is a relatively low speed, there might be some inaccuracies that are not taken into account. And thirdly, the difference in static and dynamic thrust, which is now given as $\mu = 0.66$, is not constant for every motor-propeller combination. There is a deviation in the difference given by $\sigma = 0.044$. When a confidence interval of 95% is taken (i.e. $\pm 2\sigma$) for the T/W, a range of the static T/W ratio is obtained of 5.5 < T/W < 7.2. When using these values to calculate the speed based on the exponential function from Figure 7.12, a speed range of 27.7 - 32.6 m/s is obtained. The calculated speed of 32.1 m/s lies within this range and is therefore deemed feasible and also valid as maximum speed of the drone. However, to fully validate the final design, the drone has to be actually built and tested, which is outside the scope of the DSE.

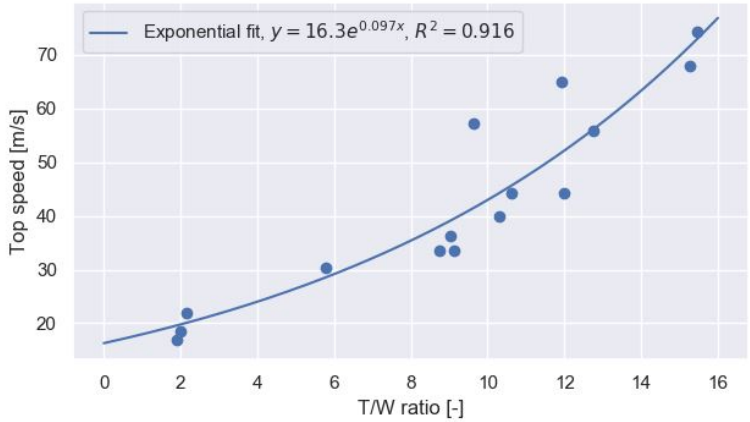


Figure 7.12: T/W-Speed relation of existing drones.

8

Electronics

The electronics are the brain of the drone. It has to provide environmental sensing, computational power, and power distribution to the drone. In this chapter, the design process of the electronics subsystem is described. First of all, the requirements that need to be fulfilled are listed in section 8.1. Secondly, a trade-off for the system architecture is performed in section 8.2. After this, hardware components are discussed and selected in section 8.3. The carrier board is discussed separately in section 8.4. Finally, the cables of the drone are sized in section 8.5.

8.1. Requirements

The design of the electronics subsystem is based on requirements, which have been setup in the baseline report [**BaselineReport**] and which have been refined and updated in the midterm report [**MidtermReport**]. The relevant requirements for the electronics subsystem will now be given. The requirements are as follows:

- **AIG-TE-HW-E1**: The drone shall be outfitted with a cooling system that prevents any component from being heated to temperatures higher than their respective operational limit.
- **AIG-MP-DR1**: All data stored on the drone before a malfunction occurs shall be recoverable.
- **AIG-MP-SM-SDS2**: The system shall be able to be turned off manually.
- **AIG-S-A1**: The acceleration of the drone shall be measured 60 times per second.
- **AIG-S-O1**: The angular rate of the drone shall be measured 60 times per second.
- **AIG-S-V1**: At least one front-facing camera shall be present.
- **AIG-S-V1.1**: The resolution of the front-facing camera shall be at least equal to the minimum required for gate detection at maximum gate distance with the used field of view measured in megapixel.
- **AIG-S-V1.2**: The field-of-view of the front-facing camera shall be at least 90 degrees.
- **AIG-S-V1.3**: The front-facing camera shall capture images at 60 Hz or greater.
- **AIG-TE-HW-CO2**: The range of remote communications to the drone shall be at least 300 m.
- **AIG-TE-HW-CO4**: The rate of data sent from the drone to remote computers shall be at least equal to the data rate of the essential systems, namely the IMU, cameras and other sensors in kB/s during normal flight conditions.
- **AIG-TE-HW-CO5**: The drone shall have manual take-over capability.
- **AIG-TE-HW-C06**: The drone shall have a telemetry option.
- **AIG-TE-HW-C1**: The RAM of the drone system shall be at least 500 MB.
- **AIG-TE-HW-C2**: The memory storage of the drone system shall be at least equal to the average flight time provided by the battery times the data rate of all system elements times 10 flights.
- **AIG-TE-HW-C4**: The drone shall use on-board computation only.
- **AIG-TE-HW-C5**: The drone shall use the 'Betaflight' program as attitude controller.
- **AIG-TE-EB-M1.2**: The total mass of the electronics subsystem shall be no greater than 420 g.
- **AIG-TE-EB-M4**: The drone shall have enough computational power to process images at 60 Hz and calculate a flight path.
- **AIG-TE-A1**: The drone shall have lights visible from every angle.
- **AIG-TE-A2**: The drone lights shall emit at least 2.5 W.
- **AIG-TE-P-M-M1**: All parts of the drone shall be manufacturable using 2D milling, 3D printing and printed circuit board machines.

- **AIG-TE-P-M-A1**: It shall be possible to assemble the drone using only standard drone assembly techniques.
- **AIG-TE-P-R2**: Each component of the drone shall be able to be replaced individually.
- **AIG-NT-S2**: The components of the drone shall be interconnected using temporary joints.
- **AIG-NT-U3**: All materials used in the drone shall comply with the RoHS 1 and RoHS 2 restrictions on hazardous sub-stances.
- **AIG-NT-R2**: The drone shall have an 'arming switch', which can be used to allow the drone to move under its own power.

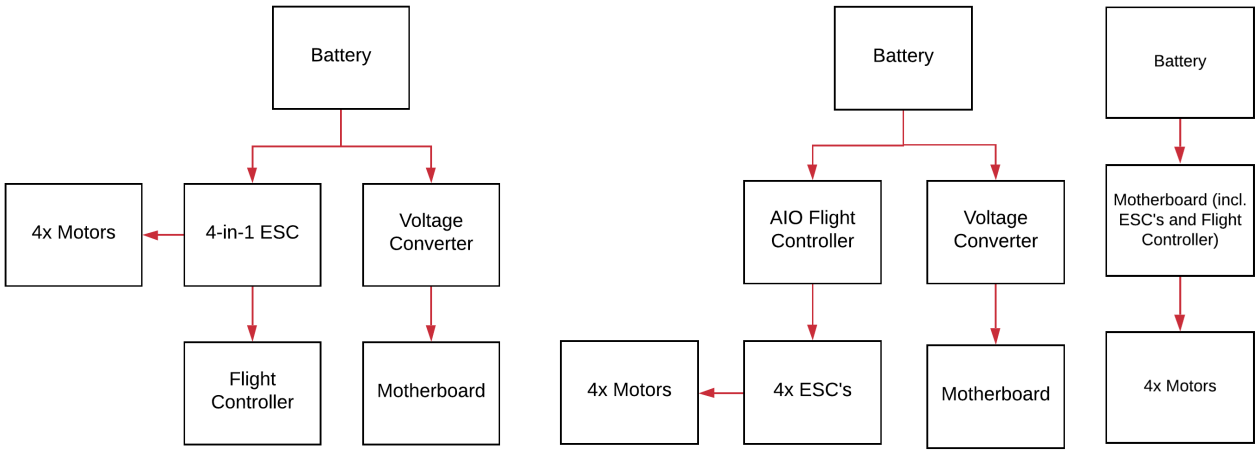
8.2. System Architecture

From the requirements and functional analysis, it is found what components are needed. These components are divided into two categories, namely primary components that strongly affect the system architecture with respect to power distribution, and secondary components that have no impact on the architecture.

The identified primary components are the electronic speed controller (ESC), flight controller (FC), and the motherboard of the "companion computer". This companion computer, besides the flight controller, is needed to supply enough computational power to run the AI, since the drone shall use on-board computation only, as described in the top-level requirements. Secondary components are the camera, communication modules, and motors. For the trade-off between different architectures, only the primary components are taken into account, since the secondary components are hardly affected by the choice of the architecture.

The feasible options are listed below and shown in Figure 8.1.

1. Separate boards: 4-in-1 ESC, a non-AIO (All-In-One) flight controller, a motherboard, and a 5V/5A voltage regulator. The battery connects to the 4-in-1 ESC and to the voltage regulator, which is connected to the motherboard. The flight controller connects to the 4-in-1 ESC.
2. Separate boards: AIO flight controller, 4 separate ESCs, a motherboard and a 5V/5A voltage regulator. The battery connects to the AIO flight controller and the voltage regulator, which is connected to the motherboard. The 4 ESCs are connected to the AIO flight controller.
3. A to-be-designed motherboard that includes a flight controller, 4 ESCs and a voltage regulator. The motherboard is connected to the battery.



(a) Option 1 (b) Option 2 (c) Option 3

Figure 8.1: System architecture options

In this list, the specified connections only refer to power connections as shown in Figure 8.1. The 4-in-1 ESC and the AIO flight controller have the ability to connect to the battery directly, while the normal flight controller and the carrier board do not. The motherboard in option 3 is a to-be-designed

motherboard that can directly connect to the battery and has the flight controller and ESCs on the same board. The option that is excluded from this list is the AIO flight controllers including ESCs on the same board since they are only available with F4 processors.

In Table 8.2 the mass is given for options 1 and 2.

Table 8.1: Mass of options 1 and 2, including ESC, flight controller, voltage regulator and carrier board.

Option	ESC [g]	Flight Controller [g]	Motherboard [g]	Voltage Regulator [g]	Total [g]
1	11	8	33	3	55
2	4x6	13	33	3	73

To calculate the average mass of the flight controller and ESC, databases of flight controllers [1] and ESCs [2] are used from Oscar Liang. For a fair comparison, ESCs with 30A rating are used and F7 flight controllers are used with Betaflight compatibility. Furthermore, for options 1 and 2 the voltage regulator is assumed to be the Pololu D24V50F5 [3] standalone voltage regulator. Finally, it is assumed that all options have the same unoptimised Quark Carrier [4] motherboard and the motherboard in option 3 has equivalent functionality. For all three options, the possibility exists to optimise the motherboard, which is further discussed after the trade-off. The mass for option 3 is harder to estimate. By combining every component into one board, mass is saved by centralizing power distribution and regulation. However, other printed circuit board components from the ESCs and flight controller are still needed. Furthermore, running a high current through the PCB requires copper traces with a large cross-sectional area to reduce heat, which increases the mass. Therefore, it is assumed that option 3 is comparable with option 1 in mass.

For the trade-off, mass is given a higher weight factor, since it is considered a more important criterion than repairability. A reduction in mass induces the snowball effect in which more mass is reduced. Also, lower mass results in better impact performance and manoeuvrability. Although better repairability results in lower repair times and lower repair costs, mass is a much more critical constraint than cost with respect to the requirements, as seen in the Midterm Report [**MidtermReport**].

Option 1 is found to have a 25% lower mass than option 2. Concerning repairability, option 1 has a disadvantage, since if an ESC breaks, the 4-in-1 ESC needs to be replaced fully. Lastly, option 3 has a large disadvantage in repairability. If one component breaks, the entire board has to be replaced.

In Table 8.2, the trade-off scores can be found. Scores are given from 1-3, where the lowest mass scores a 3, and the scores are determined linearly, based on Table 8.1. The option with the best repairability scores a 3. A small scoring range and integer scoring are used to mitigate the effect of uncertainties. From this trade-off, it is seen that option 1 is the clear winner, and therefore this is the system architecture chosen for the electronics subsystem. Lastly, a sensitivity analysis is performed by changing the weight factors. Since mass is considered more important than repairability, and therefore always has a higher weight factor, option 1 always wins.

Table 8.2: Trade-off table of system architectures

Option	Mass [70%]	Repairability [30%]	Total
1	3	2	2.7
2	2	3	2.3
3	3	1	2.4

[1] https://docs.google.com/spreadsheets/d/1VuBpQVZflz5zVNUG43qKTq4Mkwt-cTssWvb1CGqskQk/edit, (Retrieved June 20, 2020)
[2] https://docs.google.com/spreadsheets/d/1SYzJWKTu2vFVL99mt5tQxc2sTiHCXNAEZyGodfy3uE0/edit, (Retrieved June 20, 2020)
[3] https://www.pololu.com/product/2851, (Retrieved June 20, 2020)
[4] http://connecttech.com/product/quark-carrier-NVIDIA-jetson-xavier-nx-nano/, (Retrieved June 20, 2020)

8.3. Hardware

In this section, the primary and secondary components that are part of the chosen system architecture are discussed, except for the motor, which has been discussed in chapter 7, and the carrier board, which is discussed in the next section. In this section, all components are first discussed. After this, the hardware block diagram is discussed that shows the connections between the components.

8.3.1. Companion Computer

This section gives the reasoning behind the preferred companion computer, as was chosen in the midterm report. The chosen companion computer is the NVIDIA Jetson Xavier NX, mainly due to its low mass, small form factor, and computational performance. NVIDIA is the market leader in GPU (Graphic Processing Unit) computing, which is necessary for fast AI learning and inference. For this reason, the computers considered for the drone are all single-board computers from the NVIDIA Jetson series.

Table 8.3 shows the performance comparisons [5] for the Jetson series of computers, tested with various vision-based deep learning inference models.

Table 8.3: Benchmark comparisons in frames per second for the Jetson series of computers tested with seven vision-based deep learning models at various resolutions

	Classification			**Object Detection**		**Segmentation**	**Image Processing**
	Inception -V4 (299x299)	**ResNet -50 (224x224)**	**VGG -19 (224x224)**	**SSD Mobilenet -V1 (300x300)**	**Tiny YOLO -V3 (416x416)**	**U-Net (256x256)**	**Super Resolution (481x321)**
AGX Xavier	525	1941	275	1587	1098	240	281
Xavier NX	317	824	65	888	562	146	151
TX2	24	83	23	92	108	39	33

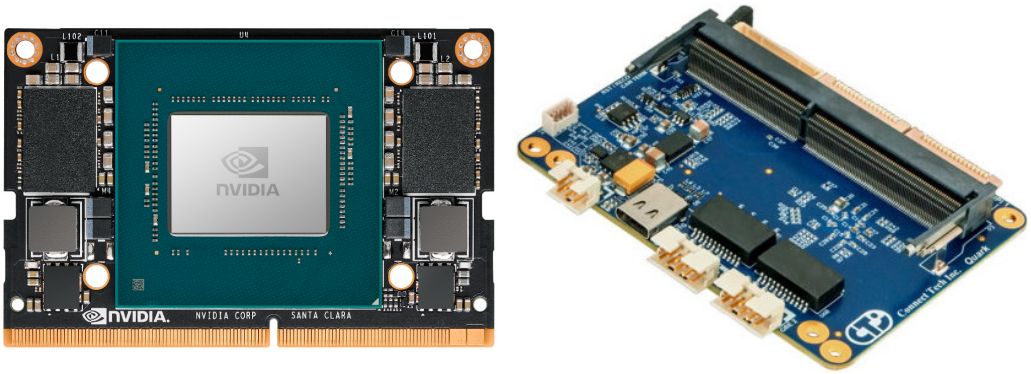
Inference is the process of interpreting new data based on what the model has previously learned, which is important during a drone race. Table 8.3 shows the frames per second (fps) that can be processed for various tasks such as classification, object detection, segmentation, and image processing. Combinations of these four tasks may be required for the drone to be able to effectively detect gates. It can be seen that the Jetson AGX Xavier vastly outperforms all other computers. This enables it to process more images per unit distance of drone flight, which can be crucial at high speeds.

The major drawback of the AGX Xavier is its mass, which results in a drone design that greatly surpasses the 1 kg drone mass requirement, as was seen from sizing the concept that focused on large computational power in the Midterm Report [**MidtermReport**]. On the other hand, the TX2 is slightly outdated and offers relatively low computational power. It can be seen in Table 8.3 that the Xavier NX is significantly more computationally powerful than the TX2, which does not meet the customer requirement AIG-TE-EB-M4 of processing 60 fps for each deep learning model. The more effective deep learning performance of the Xavier NX can likely be explained by the GPU, which consists of more cores, improving it's parallel processing capabilities. The Xavier NX has additional advantages over the TX2, including its smaller form factor, while having the same power usage and impact resistance, which is 50g for both computers [14]. All these aspects clearly show that the more recent Xavier NX outperforms the TX2 in nearly every way.[6]

The Xavier NX is a computer module, which consists of a 6-core NVIDIA Carmel ARM®v8.2 64-bit Central Processing Unit (CPU), a 384-core NVIDIA Volta™ GPU with 48 Tensor Cores GPU, 8 GB

[5] https://devblogs.NVIDIA.com/jetson-xavier-nx-the-worlds-smallest-ai-supercomputer/, (Retrieved June 21, 2020)
[6] https://developer.NVIDIA.com/embedded/develop/hardware, (Retrieved June 20, 2020)

128-bit LPDDR4x 51.2GB/s Random Access Memory (RAM), and 14 MIPI Camera Serial Interface 2 (CSI-2) camera data lanes that can handle speeds up to 1.5 Gb/s per lane. This is an important feature as it speeds up the image processing. Normally, a computer consists only of one motherboard in which the CPU, GPU, and RAM are inserted. This same board also provides power regulation as well as input and output connections to peripherals. Important to note is that the Xavier NX computer module needs a separate ”carrier board” in which it can be inserted, see Figure 8.2. This carrier board provides the computer with input and output connections to peripherals, such as the camera, flight controller, and WiFi module. Moreover, the carrier board provides power regulation and distribution to the computer and peripherals.



(a) The Xavier NX computer module

(b) The Quark Carrier board

Figure 8.2: Figure of the Xavier NX module which has to be inserted into a carrier board. The Quark Carrier board is shown as an example with the black connector in which the Xavier NX module has to be inserted.

The user is expected to buy the Jetson Xavier NX developer kit as a test bench during the development phase of the AI software, with a screen, keyboard, and mouse. The developer kit consists of the Xavier NX module as well as a carrier board. This carrier board has many connections, such as HDMI, Ethernet, and USB, which are not needed during a drone flight. Furthermore, the carrier board is large in size and would require a large drone body frame. Therefore, besides the carrier board that comes with the developer kit, a separate carrier board is needed, which is only used on the drone. This carrier board is discussed further in section 8.4.

It would also be possible to connect to the Xavier NX wirelessly via the secure shell (SSH) protocol, while it is on the carrier board in the drone. However, this requires the Xavier NX to have a wireless internet connection, which would first need to be set up using the carrier board from the developer kit. Hence the development kit, screen, keyboard, and mouse are necessary for the initial wireless connection setup of the WiFi module. After setup, the Xavier NX module is moved from the development kit carrier board to the carrier board in the drone.

Finally, the development kit also comes with a cooler for the Xavier NX module, which is designed by NVIDIA to comply with operational temperature limits. This cooler is taken together with the Xavier NX from the development kit’s carrier board to the carrier board inside the drone. Another advantage of the active cooling solution is that it helps in preventing thermal throttling, possibly increasing computational performance. Furthermore, it opens up the possibility to slightly overclock the CPU and/or GPU to further increase performance. All in all, requirement AIG-TE-HW-E1 which states that the drone shall be outfitted with a cooling system to prevent any component from overheating is complied with.

8.3.2. Camera

The camera is an important sensor that enables the drone to perform machine vision tasks. In order to choose the best possible camera available, a range of cameras is investigated, which meet the requirements. The camera shall meet the following three requirements: AIG-TE-HW-SV1.1, AIG-TE-HW-SV1.2, and AIG-TE-HW-SV1.3. To summarize, they state that the camera shall have a high enough resolution to detect a gate, it shall have a field of view of at least 90 degrees, and its frame

rate should be at least 60 fps.

Due to the recent release of the Xavier NX computer, there is only one known camera on the market that supports the Xavier NX. Other cameras do not have supporting software drivers. Fortunately, the supported camera is the e-CAM50_CUNANO, which is the main camera chosen for all design concepts in the midterm phase, and meets all the requirements. In Table 8.4, the specifications of the camera are found. The field of view (FOV) is stated in the diagonal, horizontal, and vertical directions (D/H/V). The camera supports multiple resolutions at different frame rates, but only the one closest to the required 60 fps is shown.

Table 8.4: Specifications of the chosen e-CAM50_CUNANO camera.

Size [mm]	Mass [g]	FOV (D/H/V) [deg]	Resolution [px]	power @ 1920x1080 [W]	Max power [W]	Supply voltage [V]
30x30x26	17.5	139/114/87	1920x1080 @ 65fps	1.26	1.65	3.3

The camera meets the frame rate requirement, as it is able to deliver 1920 x 1080 resolution at 65 fps. The camera can easily differentiate gates at this resolution, as a pixel width corresponds to about 5 cm when filming at a 30 m distance, while gates have dimensions of over 1 m. Furthermore, this is a higher resolution camera than cameras that have been used in past races [15] [16] as computational power is still the limiting factor.

One of the main advantages of the camera is its field of view, which is 139 °diagonally. This allows the drone view ahead of itself, even at high roll and pitch angles. Another advantage of the camera is that it has colour-sensing, as opposed to being monochromatic. Colour images can enhance object recognition due to their improved edge detection capabilities [17], which will be useful for gate detection during the drone race. A disadvantage of the camera is that it uses a standard rolling shutter, as opposed to a more complex global shutter. Since not all pixels are exposed during the same time-frame, the images become more distorted as the drone's speed or rotation rate increases. This image distortion may affect machine vision.

8.3.3. Flight Controller

The chosen flight controller is the Lumenier F7 LUX[7]. It is chosen from a database with nearly 200 flight controllers [8]. The main requirements of the FC are:

- Supports the Betaflight firmware.
- Contain at least 2 outputs of 5V for the two receivers.
- Contain two UART connections for two receivers.
- Contain an UART connection for the companion computer (Xavier NX).
- Contain a 4-in-1 ESC header.

The type of processor is also an important consideration. The Lumenier F7 Lux has a F7 processor that runs at 216 MHz. Although the H7 processor has a higher clock speed than the F7, there is only one flight controller available that has it, which is an AIO flight controller and is not compatible with the chosen system architecture. Furthermore, the higher clock speed cannot be taken advantage of, since Betaflight 4 as of now only supports a maximum loop time of 8 kHz. Therefore, flight controllers with F7 processors are chosen over the H7 processors.

Some other features of the FC that make it better than others are as follows. Firstly, it contains two separate inertial measurement units (IMU), which allows for different settings depending on the user's needs. The default IMU (MPU6000) has a sampling rate of 8kHz and is more resistant to noise than the second IMU (ICM20602), which runs at 32kHz. Secondly, the FC contains a barometer, which only

[7] https://www.lumenier.com/products/electronics/flight-controllers, (Retrieved June 20, 2020)

[8] https://docs.google.com/spreadsheets/d/1VuBpQVZflz5zVNUG43qKTq4Mkwt-cTssWvb1CGqskQk/edit, (Retrieved June 20, 2020)

has an accuracy of 1 meter, making it hard to use indoors. However, it gives the option for the user to still use it, and the barometer does not compromise the overall mass of the flight controller since it is still the most lightweight flight controller found that has all the other mentioned functions. Thirdly, the FC supports SmartPort, giving the user another choice of telemetry communication between the ground station.

8.3.4. ESC

The chosen 4-in-1 ESC is the T-Motor F45A V2 [9]. Similarly to the flight controller, it is chosen from a database of ESCs [10].

The main requirements of the ESC are:

- Has to be a 4-in-1 ESC.
- Has a higher current rating than needed by the motors of 40 A
- Has support for Dshot1200.
- Has a current sensor and ESC telemetry.
- Has a power output for LED.

The current needed by the motors is 40 A, the ESC needs a higher current rating than this value. Since the current rating of ESCs often goes up in steps of 5 A, it is important to choose an ESC with a current rating of 45 A to have enough margin to meet the motor current requirement. Dshot is a digital protocol as opposed to PWM, which is an analog protocol. The main advantage of Dshot is its higher update rate than PWM protocols, resulting in lower latency, which leads to an advantage for racing drones. Moreover, it is decided to have an ESC with a current sensor and ESC telemetry to provide better debugging options for the user, as this accommodates autonomous drone development. Finally, to comply with requirement AIG-TE-A2, which states that the drone lights shall emit at least 2.5W, an ESC is desired that has an output to connect with the LED. Using these requirements the ESCs in the database are selected, after which they are sorted based on their mass. Finally, the ESC that has the lowest mass is chosen, which is the T-Motor F45A V2.

8.3.5. Battery Charger

The lithium-polymer (LiPo) battery charger chosen is the ISDT D2. A big advantage is that the charger has an AC input as opposed to a DC input, which would require a power supply unit (PSU). This PSU is additional hardware that would need to be bought, increasing costs and volume to the entire system. Users will likely always have access to an AC power supply, considering that the drone race is indoors. The charger supports 1-6 cell batteries of many different types. This makes the charger flexible, as the user may want to use a different type of battery. It is a dual-channel charger, meaning that it can charge two batteries completely separately. Since the charger can supply 200 W of power, and each LiPo only requires under 50 W, up to four batteries can be charged if an additional parallel charging board is used. This can be useful during the testing phase of the drone if the user has multiple batteries and wants to reduce the waiting time between flight tests.

A good feature of the charger is its user interface and automation. It will automatically detect the type of battery and cell count, which can reduce the risk of damaging the battery. The charging current can be set with 0.1A precision, allowing for optimum charging of the battery. During charging, a number of parameters are displayed such as current, capacity, power, and temperature, as well as voltage per cell. This information can help indicate how charged up the battery is, and also the health of the battery.

The charger has three modes: charge, discharge, and storage. Discharging is used to safely dispose of the battery. The storage mode charges or discharges the battery to a specific voltage to best preserve its health for long term storage.

[9] http://store-en.tmotor.com/goods.php?id=899, (Retrieved June 20, 2020)

[10] https://docs.google.com/spreadsheets/d/1SYzJWKTu2vFVL99mt5tQxc2sTiHCXNAEZyGodfy3uE0/edit, (Retrieved June 20, 2020)

8.3.6. Communication and Data Handling

Communication

One of the top-level requirements states that the drone shall have a telemetry option, for which the chosen solution is a WiFi module. An overview of the communication flow and data handling is shown in Figure 8.3. Data rates have only been calculated for connections that would influence and accommodate the design process, which are given in Table 8.5.

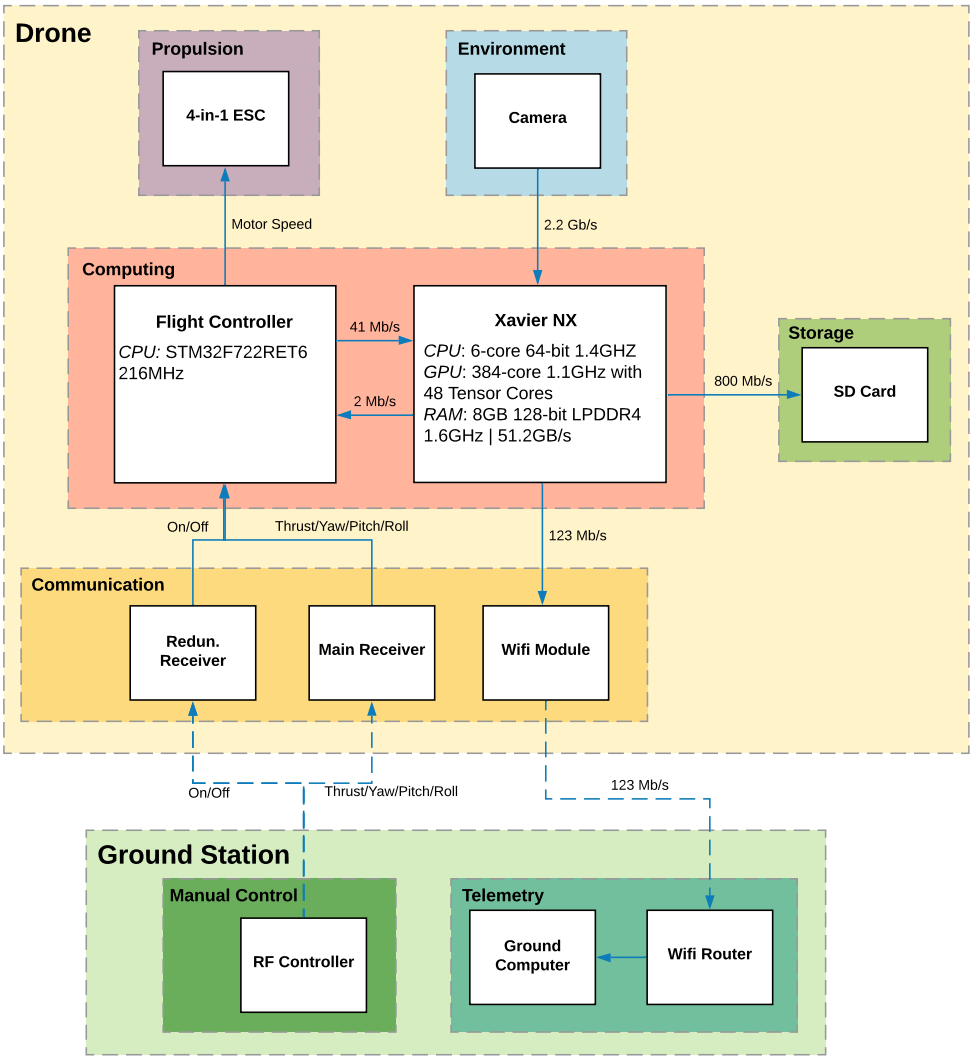


Figure 8.3: Combined communication flow and data handling diagram

Table 8.5: The data rates between the computer and other components.

Connection	Frequency	Data size	Number of Parameters	Data Rate
FC to Xavier NX	32kHz	64bit	20	41Mb/s
Xavier NX to FC	32kHz	16bit	4	2.0Mb/s
Camera to Xavier NX	75 Hz	24bit/px RBG frame (1280x960)	1	2.2Gb/s
Xavier NX to ground computer (via WiFi)	32kHz	64bit	60	123Mb/s

Table 8.5 shows all the data transferred to and from the Xavier NX computer, which is also depicted in Figure 8.3.

The data rate from the camera to the NX at a resolution of 1280x960 at 75 fps exceeds the maximum write speed of 800 Mb/s that the microSD has. However, at a lower resolution of 640x480, the data rate from the camera is 553Mb/s, allowing for all images to be stored. This also allows plenty of space for other data to be stored. Therefore, the camera can be set to this resolution, or the images can be compressed for storage after machine vision tasks, this should not be a problem for neural networks since they are still only limited to low-resolution images [15]. However, in the future, the carrier board can be equipped with a M.2 Key-M connector for a M.2 SSD. This would have a write speed that can match the total data rate of 2.24 Gb/s [11].

The FC is assumed to send up to 20 parameters, each 64-bit floating-point numbers, at a rate of 32kHz to the Xavier NX. Similarly, the NX sends 60 parameters of data to the ground station via WiFi. These parameters include inertial measurement data, motor data, as well as any other data from the computer that the user wants to store for future debugging.

Regarding communications, the chosen WiFi module is the Intel Wireless-AC 9560. With Intel being a market leader in microprocessors, reliability and quality can be expected. The module has a mass of 2.8 g, and a flat form factor with a M.2 Key-E connection. The specific model chosen is the latest one that provides the highest data transfer rate of up to 1.73 Gbps, while not compromising mass. It supports two antennas, with the ability to transmit and receive signals on two separate radio channels. As the WiFi module is inserted into the carrier board, which is inside the drone frame, external antennas are needed. The chosen antenna is the Laird Connectivity 0600-00057 [12], which can be connected to the WiFi module using an IPEX MHF4 connector and has a gain of 2 dB.

The chosen main receiver is the FrSky R-XSR. Compared to other receivers, it has a low mass of 1.5 g. Other factors that influenced the choice are additional functions, of which SmartPort telemetry is one. It was decided together with the flight controller to have this option open for the user if this would not compromise the weight or volume. Furthermore, with the redundancy function, this receiver supports the attachment of a redundant receiver. Considering requirement AIG-NT-R2, which states that the drone shall have an 'arming switch', it was decided to have a redundant receiver as a safety measure to disconnect power to the drone if it loses control.

The redundant receiver does not require the functionality that the R-XSR has. Therefore, the most lightweight option is chosen. The chosen redundant receiver is the FrSky XM, which has a mass of 1 g. It is connected to the SBUS-IN port on the R-XSR. Both the main and redundant receivers already include antennas and have an advertised range of >2 km and 600 m respectively, which should be more than sufficient, since autonomous drone racing has taken place in indoor environments to this date [15].

Finally, the range of the WiFi module is not specified and has to be further investigated. Assuming the transmission power and sensitivity of the WiFi router is better than the WiFi module, the transmission power and sensitivity of the WiFi module are the most limiting, and they are therefore used to estimate the range. Assuming that free-space loss is the only form of attenuation, Equation 8.1 can be used to calculate the range.

$$P_r = P_t(\frac{\lambda}{4\pi d})^2 \tag{8.1}$$

First of all, the 2.4 GHz frequency is considered, for which the most limiting transmission power is 15.5 dBm and sensitivity is -72.1 dBm [18]. Including antenna gain, this results in a range of 300 m. For 5 GHz, the most limiting transmission power is 11 dBm and sensitivity is -59.8 dBm, which results in a range of 21 m. From this can be concluded that the range is significantly smaller for the 5 GHz frequency.

The data rate of the WiFi Module is 258 Mb/s on 2.4 GHz frequency [18], while the data rate needed to the ground station is only 123 Mb/s as seen in Table 8.5. Therefore, it can be concluded that the combination of the range and the data rate at 2.4 GHz is sufficient. The assumption that free-space

[11] https://ssd.userbenchmark.com/, (Retrieved June 22, 2020)

[12] https://nl.mouser.com/ProductDetail/Laird-Connectivity/0600-00057?qs=KEDsXO96ri%252BfQ6ydx8Rynw%3D%3D, (Retrieved June 20, 2020)

loss is the only form of attenuation, results in the range of 300 m for 2.4 GHz are being overestimated. However, even taking into account a margin, the range is sufficient since drone racing takes place indoors. The drone race organisers shall place the WiFi routers such that there is WiFi coverage on the entire track. Furthermore, the spacing of the WiFi routers shall be decreased further if the 5 GHz band is used, in the case in the future a higher data rate is needed to allow for more telemetry data.

Data Storage

The data storage device chosen is the microSD card, mainly because it is extremely lightweight and since the Xavier NX has a microSD card slot, it can be implemented in a compact way. MicroSD cards can also store large amounts of data. The amount of data required will be determined in this section. Firstly, the two requirements to be complied with are discussed.

Requirement AIG-TE-HW-MP-DR1 states that all data stored on the drone before a malfunction occurs shall be recoverable. This is met, since the carrier board, as described in section 8.4, has a built-in feature where if the power shuts off, the carrier board signals the Xavier NX module to shut off safely. Therefore, the data stored in the SD card should be recoverable.

Furthermore, the storage requirement AIG-TE-HW-C2 states that the memory storage of the drone system shall be at least equal to the average flight time provided by the battery times the data rate of all system elements times 10 flights. The maximum write speed of the microSD card is 800 Mb/s. For a duration of ten flights, each one minute long, the required storage space is 480 Gb or 60 GB. Hence, the required SD card chosen has a storage of 64 GB, which is the lowest standard storage size that meets the requirement. The chosen microSD card is the Kingston Technology Canvas Select Plus 64 GB microSD card [13], which meets the aforementioned requirements. Lastly, the operating system and other software can be stored on the onboard 16 GB eMMC 5.1 storage in the Xavier NX module.

8.3.7. Hardware Block Diagram

All hardware components have been discussed except for the carrier board, which is discussed in the next section. In Figure 8.4, the connections between the pins of all components are shown. Colors are used to be able to distinguish cables better. Furthermore, the diagram is used to establish requirements on the connections of the carrier board.

[13] https://www.bol.com/nl/p/kingston-technology-canvas-select-plus-flashgeheugen-64-gb-microsdxc-klasse-10-uhs-i/9200000122641889/?bltgh=i1xYt0AqnsG0hb9m66VCvQ.1_44.48.ProductTitle (Retrieved June 27, 2020)

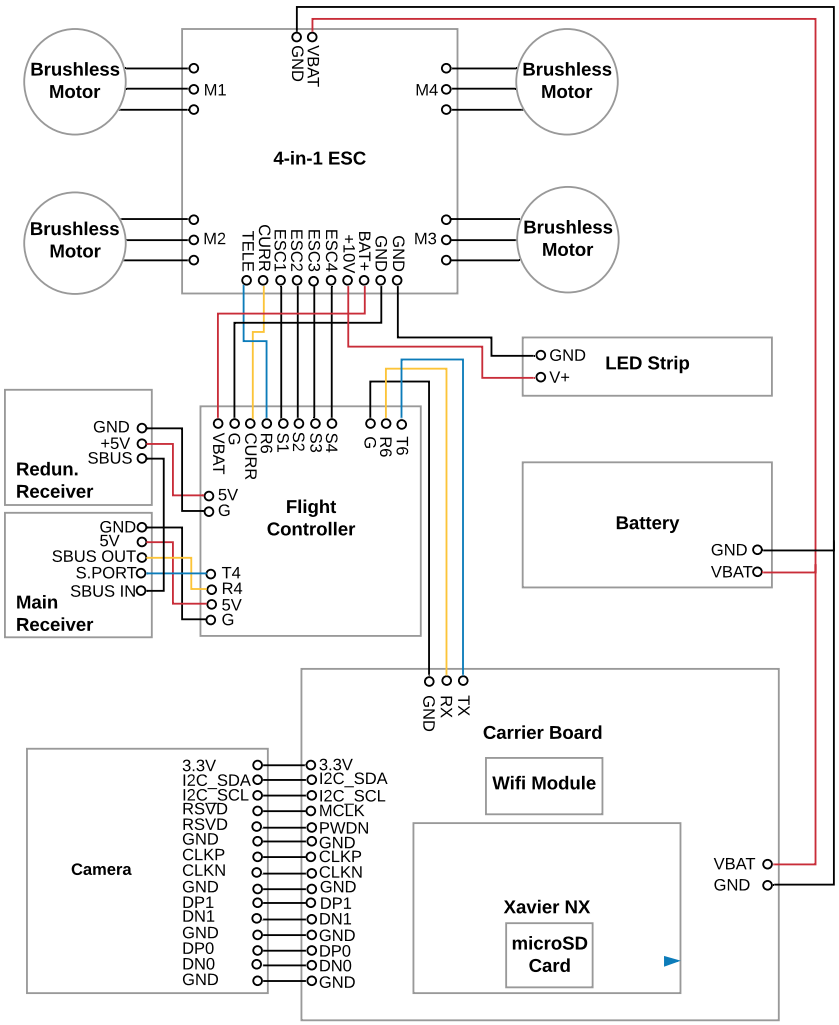


Figure 8.4: Hardware block diagram including connections between electrical components.

8.4. Carrier Board

The carrier board provides the Xavier NX module with input and output connections to peripherals, such as the camera, flight controller, and WiFi module. Moreover, the carrier board provides power regulation and distribution to the computer and peripherals. In this section, the carrier board that is mounted inside the drone is described in more detail.

After choosing option number 1 from the system architectures, more options can also be explored for the carrier board. The first option considered for the carrier board is an off-the-shelf solution. The Quark Carrier is the most lightweight and compact off-the-shelf option that is found. However, it is not optimised for the case of being used on an AI drone with the given other electrical components. It either provides more functionality than needed or it lacks functionality, which can be seen in Table 8.6. Moreover, the shape of the board is fixed and cannot be changed accordingly to optimise component placement and body frame mass. Also, it has connectors sticking out to the sides of the board, which increases body volume and therefore, body frame mass.

Furthermore, the camera connector is not favorably positioned, since the connector is not on the same side of the board where the camera is located. Finally, connecting a separate WiFi module and voltage regulator to the carrier board requires more space in the drone body frame, which leads to a heavier structure.

Table 8.6: Functions that are provided by the Quark Carrier, their necessity, and missing functionality

Function	Ethernet (2x)	USB	Camera (2x)	40-Pin GPIO	MicroSD	4-Pin Fan	5V Input	Battery Voltage Input	WiFi
Present	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No	No
Needed	No	No	Only 1x	Only 2x UART	No	Yes	No	Yes	Yes

The second option is to design a custom carrier board, resolving all the aforementioned problems. Before allocating many resources to this process, it is estimated how much weight is saved by designing a custom carrier board, not considering the body frame mass. The mass estimates are found in Table 8.7. For the off-the-shelf solution the Quark Carrier, ESP8266-12F [14] and Pololu D24V50F5 [15] are used. For the custom design solution, the voltage regulator is included in the board and the WiFi module is the Intel Wireless-AC 9560 [16]. The custom carrier board mass is estimated using the Quark Carrier board as reference. First, the mass of all connectors is subtracted, and it is graphically estimated that there is a possible reduction in printed circuit board (PCB) surface area of 10%, assuming equal PCB density as the Quark Carrier. Finally, the mass of the necessary connectors is added to obtain the estimate. As seen in Table 8.7, a mass reduction of approximately 20% can be accomplished with the custom carrier board design. This is excluding the mass and volume reduction of the drone body frame, which would result in an even lower mass. Therefore, it is decided to make a custom design for the carrier board.

Table 8.7: Initial mass estimate for off-the-shelf and custom design option

Mass	**Off-The-Shelf**	**Custom**
Carrier Board [g]	33	28
WiFi Module [g]	1.7	2.8
Voltage Regulator [g]	3	-
Total [g]	37.7	30.8

8.4.1. Requirements

First of all, the more detailed functions of the PCB are determined from self-imposed requirements on connections and from design guidelines imposed by NVIDIA [17]. Self-imposed requirements are:

- The carrier board shall have a 260-pin SODIMM connector for the NVIDIA Xavier NX computing module.
- The carrier board shall have a UART connection in order to connect with the flight controller
- The carrier board shall have a 15-pin MIPI CSI-2 camera connector.
- The carrier board shall take in battery voltage and convert it to appropriate voltages.
- The carrier board shall have a M.2 Key E connector for the WiFi module.
- The carrier board shall have mounting holes for the NVIDIA Xavier NX, the WiFi module, and attachment to the body frame.
- The connectors shall be in a position that minimises cable distance as much as possible.

The most driving requirements imposed by NVIDIA are related to power regulation and logic, which are listed below:

- The carrier board shall have a power-up sequence in which the module is enabled after it has been ensured that the power supply is stable.
- The carrier board shall enable the power supply for other components on the carrier board after the module is enabled and fully powered.
- The carrier board shall disable the module if a shutdown is requested from the module, due to overheating, for example.

[14] https://www.addicore.com/ESP8266-ESP-12F-p/ad483.htm, (Retrieved June 17, 2020)
[15] https://www.pololu.com/product/2851/specs, (Retrieved June 17, 2020)
[16] https://www.intel.com/content/www/us/en/products/wireless/wireless-products/dual-band-wireless-ac-9560.html, (Retrieved June 17, 2020)
[17] https://developer.NVIDIA.com/jetson-xavier-nx-product-design-guide-v10, (Retrieved June 17, 2020)

- The carrier board shall disable the module if there is a sudden power loss.

8.4.2. Circuit Design

The custom carrier board is designed with the open-source schematics of the NVIDIA Xavier NX Developer Kit carrier board design P3509 A01 [18] as a reference. Furthermore, the design of the carrier board PCB is performed using Altium Designer version 20.1.11. All symbols, footprints, and 3D models are imported from Altium's manufacturer part library as well as the Component Search Engine [19] by SamacSys. Moreover, the schematics of the carrier board can be found in Figure 8.9, Figure 8.10 and Figure 8.11. In the next sections, parts of the schematics are discussed.

Voltage Dividers

The first aspect noted is the input voltage difference to the reference design. The reference design uses a DC voltage input of 19V, while the drone has a nominal battery voltage input of 4S or 14.8V. First of all, voltage dividers, which are pairs of resistors, need to be adapted to the new voltage. The resistances of the voltage divider (R53 and R54 in D2 of Figure 8.11) are adapted such that the output voltage remains the same as with the 19V DC input using Equation 8.2.

$$V_{out} = \frac{R_2}{R_1 + R_2} \cdot V_{in} \tag{8.2}$$

Furthermore, the voltage detector pulls the SHUTDOWN_REQ pin of the Xavier NX (Pin 233 in D6 of Figure 8.11) low, when the detected input voltage drops below the threshold voltage of 2.0V. For this voltage detector, the resistances of the divider (R23 and R26 in A4 of Figure 8.11) are calculated assuming an input voltage of 90% of the nominal voltage, or 13.3V would indicate that the battery is almost depleted [11], which is based on a battery depth of discharge of 90%. Again, Equation 8.2 is used, and the total resistance is kept close to the original value.

Pull-up and Pull-down Resistors

Pull-up and pull-down resistors are used to ensure what the state is of a binary signal [19]. Without such a resistor, the pin is left floating and the state would be fluctuating between an on and off state. A pull-up resistor connects the signal with a certain input voltage, while a pull-down resistor connects the signal to ground. These resistors typically have a value between 1 $k\Omega$ and 100 $k\Omega$. For the custom PCB design, the resistances from the reference design are used.

Voltage Converters

Voltage converters are needed to convert battery voltage to 5V, 3.3V and 1.8V. The voltage conversion occurs in two stages. First of all, two buck converters are used to convert battery voltage to 5V and 3.3V. These buck converters are highly power-efficient with the chosen buck converter TPS54527DDAR having an efficiency between 85% and 90%, depending on current. The buck converter circuits (U5 and U10 in A7 and C7 of Figure 8.11) are the recommended layout for 5V and 3.3V output, which are specified in the datasheet [20]. Since buck regulators use a switching mechanism, this will cause a voltage ripple, which is a problem for sensitive components such as the MCU (microcontroller unit). Here, the second stage comes into play, which consists of two linear regulators (U12 and U13 in C5 and D5 of Figure 8.11) converting 5V to 3.3V and 3.3V to 1.8V. Linear regulators have a stable voltage output since it uses an adjustable resistance mechanism instead of a switching mechanism. However, due to this resistance, linear regulators are inefficient since there is a power loss in heat, and they are only used in the circuit when necessary.

[18] https://developer.NVIDIA.com/jetson-xavier-nx-developer-kit-carrier-board-design-package-p3509-a01, (Retrieved June 17, 2020)
[19] https://componentsearchengine.com/, (Retrieved June 18, 2020)
[20] https://www.ti.com/lit/gpn/TPS54527, (Retrieved June 20, 2020)

Power Logic

Furthermore, a microcontroller unit (MCU) is present on the carrier board. This MCU enables the power logic to be executed as described from the requirements in subsection 8.4.1. The MCU used in both the reference design and the custom design is the EFM8SB10F2G [21] from Silicon Labs. For this MCU to work firmware needs to be programmed into its flash memory. The open-source firmware [22] and instructions are provided by NVIDIA, which can be programmed over a UART interface. The receiver and transmitter pins on the microcontroller unit (P0.5 and P0.4 respectively in C2 of Figure 8.11) lead to an external connector on the carrier board in order to allow programming via another computer.

Level Shifters

Level shifters are used to transfer from voltage domains [20]. For example, the digital signals from the Xavier NX use 1.8V, while the fan works on 5V signals. Level shifters are also implemented for the UART connectors from 1.8V to 3.3V.

8.4.3. Results, Verification and Validation

After finishing the schematics, they are compiled and verified using the built-in ”Schematic Validation” function in Altium. This function checks for logical and electrical errors, such as unconnected pins, and having inputs wrongly connected to other inputs. Furthermore, the schematic itself is visually verified by comparison with the reference schematic and design guide.

Since validation, component layout, and routing of the traces cannot be performed by the team due to a lack of resources, this is outsourced to an external company. Based on baseline hour estimates of 49 hours [23] and a cost of 90-145 USD per hour, it is estimated that this process costs between 3950 and 7100 Euros.

Size

Furthermore, through iterations with the structures department, it was decided to place the carrier board above the flight controller and 4-in-1 ESC as shown in Figure 8.5. The reasoning behind the layout is that it minimises the total drone body volume and frontal surface area, improving the mass and aerodynamic performance. Meanwhile, other constraints are met, such as the airflow at the top of the cooler not being obstructed and the minimisation of cable length between all components. Consequently, the camera connector is at the front of the carrier board, the UART connector for the flight controller on the side, and the power connector at the back, close to the connection from the battery to the 4-in-1 ESC.

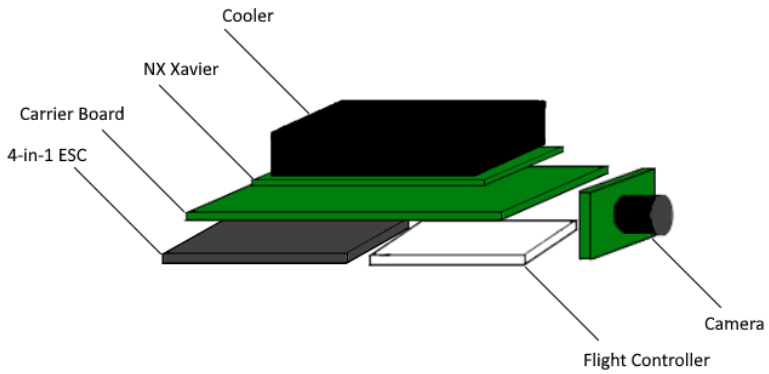


Figure 8.5: Chosen layout in 3D of the 4-in-1 ESC, flight controller, carrier board and camera.

The overall size of the carrier board is mainly constrained by the size of the NVIDIA Xavier NX module

[21] https://www.silabs.com/mcu/8-bit/efm8-sleepy-bee/device.efm8sb10f2g-qfn20, (Retrieved June 17, 2020)
[22] https://developer.NVIDIA.com/embedded/NVIDIA-jetson-agx-xavier-and-jetson-xavier-nx-power-button-supervisor-firmware, (Retrieved June 17, 2020)
[23] https://www.911eda.com/solutions/pcb-design-cost, (Retrieved June 17, 2020)

and the 260-pin connector, as well as the placement of the camera connector. After finishing the schematics, all connectors and components are placed on the PCB. The component layout on the front and back of the PCB can be seen in Figure 8.6 and Figure 8.7 respectively. It can be visually verified from these figures that all components can fit on the carrier board, with enough surface area left for rearrangements.

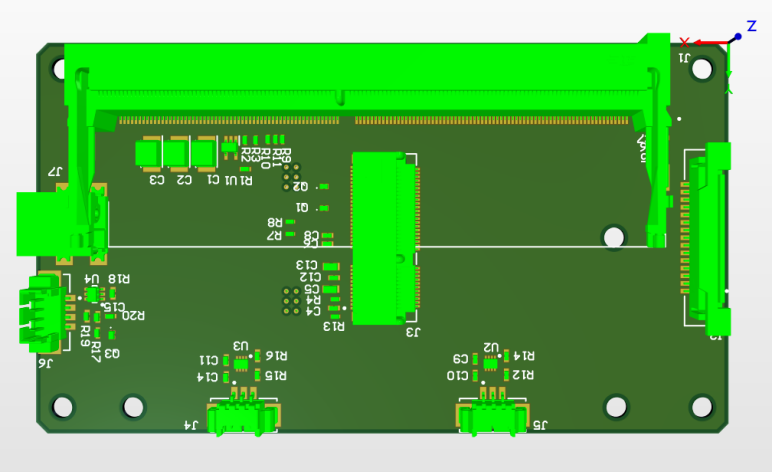


Figure 8.6: Front side of the carrier board design.

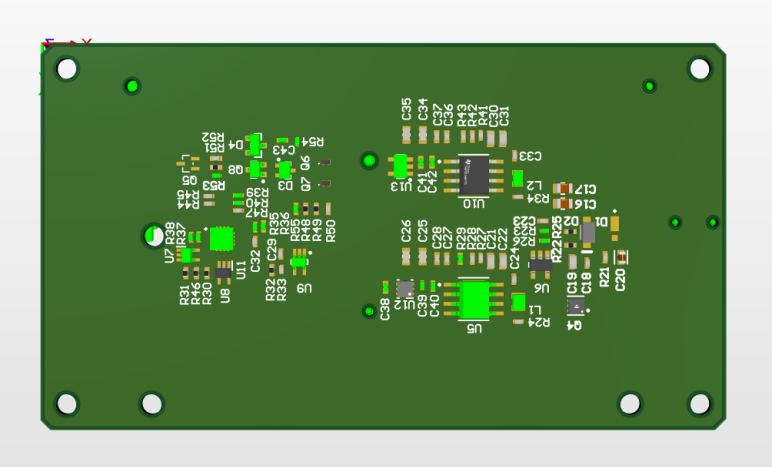


Figure 8.7: Back side of the carrier board design.

Furthermore, it is assumed that the carrier board uses the same diameter for the mounting holes and the same distance between the edge of the PCB and the mounting holes as the Quark Carrier, which are 2.7 mm and 2 mm respectively. These values are taken from a 3D model [24]. Also, the thickness of the carrier board is assumed to be the same as the Quark Carrier, which is 1.6 mm. Initially, the size of the carrier board was also constrained by the flight controller and 4-in-1 ESC, as the mounting holes of the carrier board need enough clearance from these components. However, later it was decided that the carrier board is not attached to the same baseplate as the 4-in-1 ESC and flight controller, lifting the constraint. Finally, this resulted in a carrier board length of 91 mm, a width of 51.3 mm, and a thickness of 1.6 mm.

Mass

The mass of this carrier board is estimated assuming equal density to the Quark Carrier. First of all, the mass of the connectors is taken away from the Quark Carrier, after which the PCB mass per square millimeter is known. After multiplication with the custom carrier board surface area, and adding

[24] http://connecttech.com/product/quark-carrier-NVIDIA-jetson-xavier-nx-nano/, (Retrieved June 22, 2020)

the mass of the connectors on the custom carrier board, a total mass is found of 29.5 g. This is higher than the initially estimated mass of 28 g in Table 8.7.

However, due to the reduction in carrier board size and removal of connectors sticking outside of the board, the body frame can become smaller. Assuming the body frame is a rectangular box with constant thickness, a reduction in one of the dimensions of the carrier board results into a slice being removed from this rectangular box, while an increased dimension results into a slice being added, as the carrier board is the largest component in the body frame. Using this method, it can be estimated how much mass is saved in the body frame.

Moreover, it is assumed that there are 5 mm gaps between components in the vertical direction to ensure enough clearance between components. This results in a box height of 45.7 mm. The dimensions of the Quark Carrier including the connectors is 82.6 mm by 63.6 mm, and the dimensions of the custom carrier board are 91 mm by 51.3 mm. Furthermore, the thickness of the body frame is 3 mm and the material used (PEI Ultem) has a density of 1270 kg/m^3 (see chapter 9). With this, it is calculated that a slice is added of 5.5 cm^3, while a slice is removed of 10.1 cm^3, resulting in a body frame mass reduction of 5.8 g.

The off-the-shelf solution has a total mass of 37.7 g (see Table 8.7), and the custom carrier board solution a mass of 29.5 + 2.8 = 32.3 g (carrier board + WiFi module). This means that a total mass saved is (37.7+5.8) - 32.3 = 11.2 g, which is a mass reduction of 26%. Moreover, the body frame mass saved from not needing extra space for a separate WiFi module, and voltage regulator is not taken into account, which would result in a higher reduction.

Cost

For the cost, two options are identified. It is assumed that the PCB is a 4 layer PCB with a copper thickness of 1 oz and a total thickness of 1.6 mm. Furthermore, the PCB has a number of 53 unique components and 138 total components. Using EuroCircuits, the minimum order quantity is one, resulting in a price of 80 Euros for PCB manufacturing and a yet unknown price for assembly. The second option is PCBWay, where the minimum order quantity is five, resulting in a price of 63 Euros per PCB including assembly. Thus, it would be the most cost-efficient if the drone race organisation order the PCB's in bulk and distribute them to the participants. These participants can reimburse the race organisation if required. It is also possible for the participants to order a single PCB by themselves, although this is guaranteed to result in a higher cost.

8.5. Cables

Most of the hardware are interconnected by cables with sizes that mainly depend on how much current it must carry. Table 8.8 [25] shows the main parameters of the cables used. Table 8.9 shows all the cables required between each connection. These connections as well as the voltage and current are visually shown in the electrical block diagram in Figure 8.8.

Table 8.8: AWG cables used in the design with their parameters.

Cable type	Permissible cont. current (A)	Cable resistivity (Ω/km)	Cable weight (g/m)
10 AWG	140	6.3	61
16 AWG	35	24.4	20
26 AWG	3.5	123	4
28 AWG	1.25	227.2	4

[25] https://www.banggood.com/nl/1M-8101214161820222426-AWG-Silicone-Wire-SR-Wire-p-921159.html?rmmds=search&ID=512906&cur_warehouse=CN, (Retrieved June 17, 2020)

Table 8.9: Cables required for each connection, with their approximate weight and voltage drops. The connections are visualised in Figure 8.8.

Connection	Current (A)	Voltage (V)	Cable Type	Cable Length (cm)	Cable Count	Cable Mass (g)	Voltage Drop (V)	Power loss (W)
Battery to ESC	160	14.8	10 AWG	20	2	24.4	0.19 (1.3%)	64.5
Battery to Carrier Board	1.6	14.8	26 AWG	20	2	1.6	0.04 (0.27%)	0.13
ESC to Motors	40	14.8	16 AWG	10	12	24	0.09 (0.61%)	40.1
ESC to FC	0.2	14.8	28 AWG	10	8	3.2	0.00 (0%)	0.00
FC to Main Receiver	0.07	5	28 AWG	5	3	0.6	0.00 (0%)	0.00
FC to Redun. Receiver	0.03	5	28 AWG	5	3	0.6	0.00 (0%)	0.00
ESC to LED Strip	0.12	10	28 AWG	10	2	0.8	0.00 (0%)	0.00
Carrier board to Camera	0.5	3.3	15-pin FFC cable	10	1	3	-	-

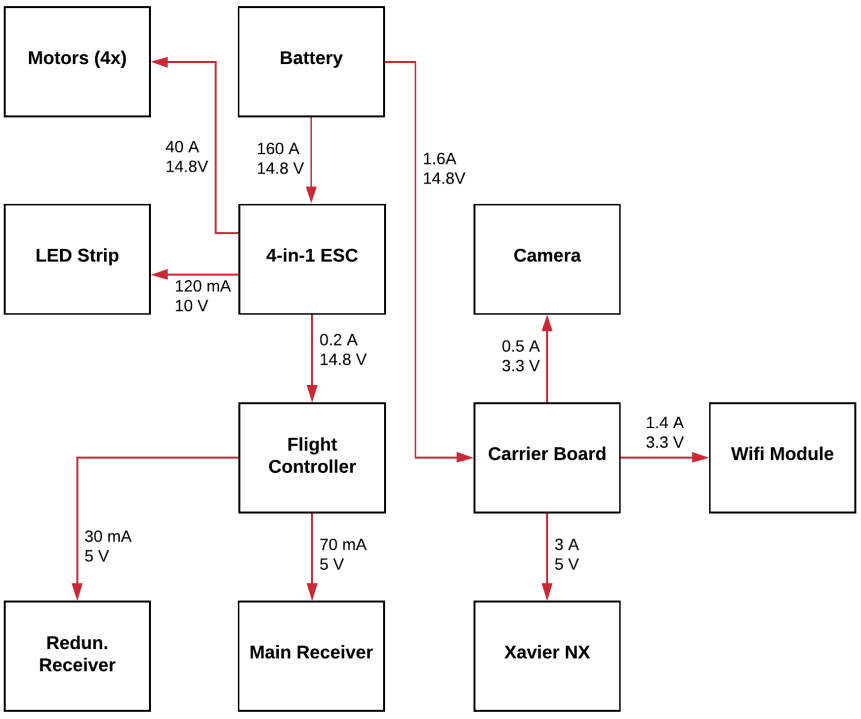


Figure 8.8: Electrical block diagram showing voltage and current for each component.

The connections from the carrier board to the WiFi module and the Xavier NX are not listed in the table as they slot directly into the carrier board and hence require no cables. Besides the camera cable, all other cables have a copper conductive core, covered by a silicon insulator. These types of silicon wires are commonly used in drones due to their high heat resistance and flexibility, which allows for small turn radii [26]. The cable masses have been calculated based on their estimated lengths,

[26] https://oscarliang.com/wire-awg-chart-quadcopter-rc/, (Retrieve June 17, 2020)

and the number of cables required for each connection. The voltage drops have been calculated by considering the current, resistivity of the cable, and length of the cable, as shown in Table 8.9. It can be seen that voltage drops are extremely small, with a maximum of 1.3% from the battery to the ESC. These voltage drops are therefore considered negligible.

The power loss in the cables is also calculated, mainly to show how much heat is produced, however, it is not used in the cable sizing. It can be seen that the power loss for both wires from the battery to ESC is 64.5W, over 4 times higher than the maximum power of the Xavier NX. Similarly, the combined power loss of cables connecting the motors is also relatively high at 40.1W. These wires are placed a reasonable distance away from most electronics, hence it will likely not cause any overheating. In the future, these power losses can be used to design more accurate cooling solutions for the drone.

It should be noted that the maximum current from the battery to ESC and ESC to motors slightly exceeds the permissible continuous current. The main problem of exceeding the current limit is that the wire will heat up, which can melt the insulator and cause a fire. There are two reasons why the probability of this happening is quite low. Firstly, the wires themselves can withstand a much higher current depending on the time duration. Secondly, the cables may experience some airflow, which would increase convective heat transfer out of the cable.

The camera cable uses a 15-pin flat flexible cable (FFC), whose mass is estimated as the data is not readily available. Also, the voltage drop has not been calculated but can be assumed to be negligible, just like the other low current cables.

The wiring between the battery and the ESC will contain an XT150 connector, which can handle currents up to 150A. Other more lightweight connectors such as XT60, XT90, and EC5 connectors are only suitable for continuous currents of 60, 90, and 120 A respectively, and hence are not used. Additionally, a 3.5x1.35mm coaxial power connector is used to connect the battery to the carrier board. The XT150 and coaxial connectors weigh 18g [27] and 2g [28] respectively. The total cable weight from Table 8.9, with a 5% margin to take inaccuracies into account, is 60g. All the cables, combined with the connectors leads to a total of 80g.

[27] https://www.amazon.com/Amass-Female-Connector-Battery-Charge/dp/B01EUFHAIK, (Retrieved June 17, 2020)

[28] https://www.amazon.co.uk/sourcingmap%C2%AE-5-5mm-2-5mm-Connectors-Barrel/dp/B00H55L4S0, (Retrieved June 17, 2020)

Figure 8.9: Electronic schematic of I/O connectors.

M.2 KEY E

J3A — 2199230-4

GND	1	1	
USB_DP	3	3	USB2_AP_P
USB_DM	5	5	USB2_AP_N
GND	7	7	
SDIO_CLK	9	9	SDIO_CLK
SDIO_CMD	11	11	SDIO_CMD
SDIO_DATA0	13	13	SDIO_DATA0
SDIO_DATA1	15	15	SDIO_DATA1
SDIO_DATA2	17	17	SDIO_DATA2
SDIO_DATA3	19	19	SDIO_DATA3
SDIO_WAKE#	21	21	
SDIO_RESET#	23	23	
GND	33	33	
AP_PCIE0_TXP0	35	35	C6 0.1uF → PCIE1_TX0_P
AP_PCIE0_TXN0	37	37	C8 0.1uF → PCIE1_TX0_N
GND	39	39	
AP_PCIE0_RXP0	41	41	PCIE1_RX0_P
AP_PCIE0_RXN0	43	43	PCIE1_RX0_N
GND	45	45	
REFCLK0P0	47	47	PCIE1_CLK_P
REFCLK0N0	49	49	PCIE1_CLK_N
GND	51	51	
CLKREQ0#	53	53	PCIE1_CLKREQ
PEWAKE0#	55	55	PCIE_WAKE
GND	57	57	
AP_RESERVED/PETXP1	59	59	
AP_RESERVED/PETXN1	61	61	
GND	63	63	
AP_RESERVED/PERXP1	65	65	
AP_RESERVED/PERXN1	67	67	
GND	69	69	
RESERVED/REFCLKP1	71	71	
RESERVED/REFCLKN1	73	73	
GND	75	75	

R10 47kOhm, R11 47kOhm — VDD_3V3_SYS

J3B — 2199230-4

3.3V	2	2	C4 0.1uF, C5 10uF → VDD_3V3_SYS
3.3V	4	4	
LED1#	6	6	
PCM_CLK/I2S_SCK	8	8	I2S1_SCLK
PCM_SYNC/I2S_WS	10	10	I2S1_LRCK
AP_PCM_IN/I2S_SDIN	12	12	I2S1_SDIN
AP_PCM_OUT/I2S_SDOUT	14	14	I2S1_SDOUT
LED2#	16	16	
GND	18	18	
UART_WAKE#	20	20	R4 47kOhm → VDD_3V3_SYS; BT_M2_WAKE_AP
AP_UART_RXD	22	22	UART0_RXD
AP_UART_TXD	32	32	UART0_TXD
AP_UART_CTS	34	34	UART0_CTS
AP_UART_RTS	36	36	UART0_RTS
VENDOR_DEFINED	38	38	
VENDOR_DEFINED	40	40	
VENDOR_DEFINED	42	42	
COEX3	44	44	
COEX2	46	46	
COEX1	48	48	
SUSCLK_32KHZ	50	50	M2E_SUSCLK_32KHZ
PERST0#	52	52	R9 47kOhm → VDD_3V3_SYS; PCIE1_RST
W_DISABLE2#	54	54	W_DISABLE2
W_DISABLE1#	56	56	W_DISABLE1
I2C_DATA	58	58	I2C2_SDA
I2C_CLK	60	60	I2C2_SCL
ALERT#	62	62	R13 47kOhm → VDD_1V8; M2E_ALERT*
RESERVED	64	64	
UIM_SWP/PERST1#	66	66	
UIM_POWER_SNK/CLKREQ1#	68	68	
UIM_POWER_SRC/GPIO1/PEWAKE1#	70	70	
3.3V	72	72	C12 0.1uF, C13 10uF → VDD_3V3_SYS
3.3V	74	74	
MNT 1	76	76	
MNT 2	77	77	

UART

J4 — 53261-0371

UART2_RX	1	1	UART2_RX
UART2_TX	2	2	UART2_TX
GND	3	3	
MNT 1	4	4	
MNT 2	5	5	

GND

UART

J5 — 53261-0371

UART1_RX	1	1	UART1_RX
UART1_TX	2	2	UART1_TX
GND	3	3	
MNT 1	4	4	
MNT 2	5	5	

GND

CSI CAMERA

J2 — 1-1734248-5

1	1	
2	2	CSI_D0_N
3	3	CSI_D0_P
4	4	
5	5	CSI0_D1_N
6	6	CSI0_D1_P
7	7	
8	8	CSI0_CLK_N
9	9	CSI0_CLK_P
10	10	
11	11	CAM0_PWDN
12	12	CAM0_MCLK
13	13	CAM0_I2C_SCL
14	14	CAM0_I2C_SDA
15	15	

R5 47kOhm, R6 47kOhm → VDD_3V3_SYS
C7 0.1uF → VDD_3V3_SYS
GND

W_DISABLE2

VDD_3V3_SYS — R7 47kOhm — W_DISABLE2
BT_M2_EN — Q1 DMN26D0UFB4-7 — GND

W_DISABLE1

VDD_3V3_SYS — R8 47kOhm — W_DISABLE1
W_DISABLE1_CTRL — Q2 DMN26D0UFB4-7 — GND

LEVEL SHIFTER

U2 — TXB0302DQMR

C10 0.1uF → VDD_1V8
C9 0.1uF → VDD_3V3_SYS
R12 100kOhm
R14 100kOhm → VDD_1V8

UART1_RXD	1	8	VCCA / VCCB
UART1_TXD	2	7	A1 / B1 → UART1_RX
	3	6	A2 / B2 → UART1_TX
GND	4	5	OE

LEVEL SHIFTER

U3 — TXB0302DQMR

C14 0.1uF → VDD_1V8
C11 0.1uF → VDD_3V3_SYS
R15 100kOhm
R16 100kOhm → VDD_1V8

UART2_RXD	1	8	VCCA / VCCB
UART2_TXD	2	7	A1 / B1 → UART2_RX
	3	6	A2 / B2 → UART2_TX
GND	4	5	OE

LEVEL SHIFTER

U4 — SN74LVC1T45DCKR

FAN_PWM — R18 100kOhm — DIR, A, B, VCCA (VDD_5V_SYS), VCCB (VDD_1V8), GND — R19 4.7kOhm — FAN_PWM_Q

FAN HEADER

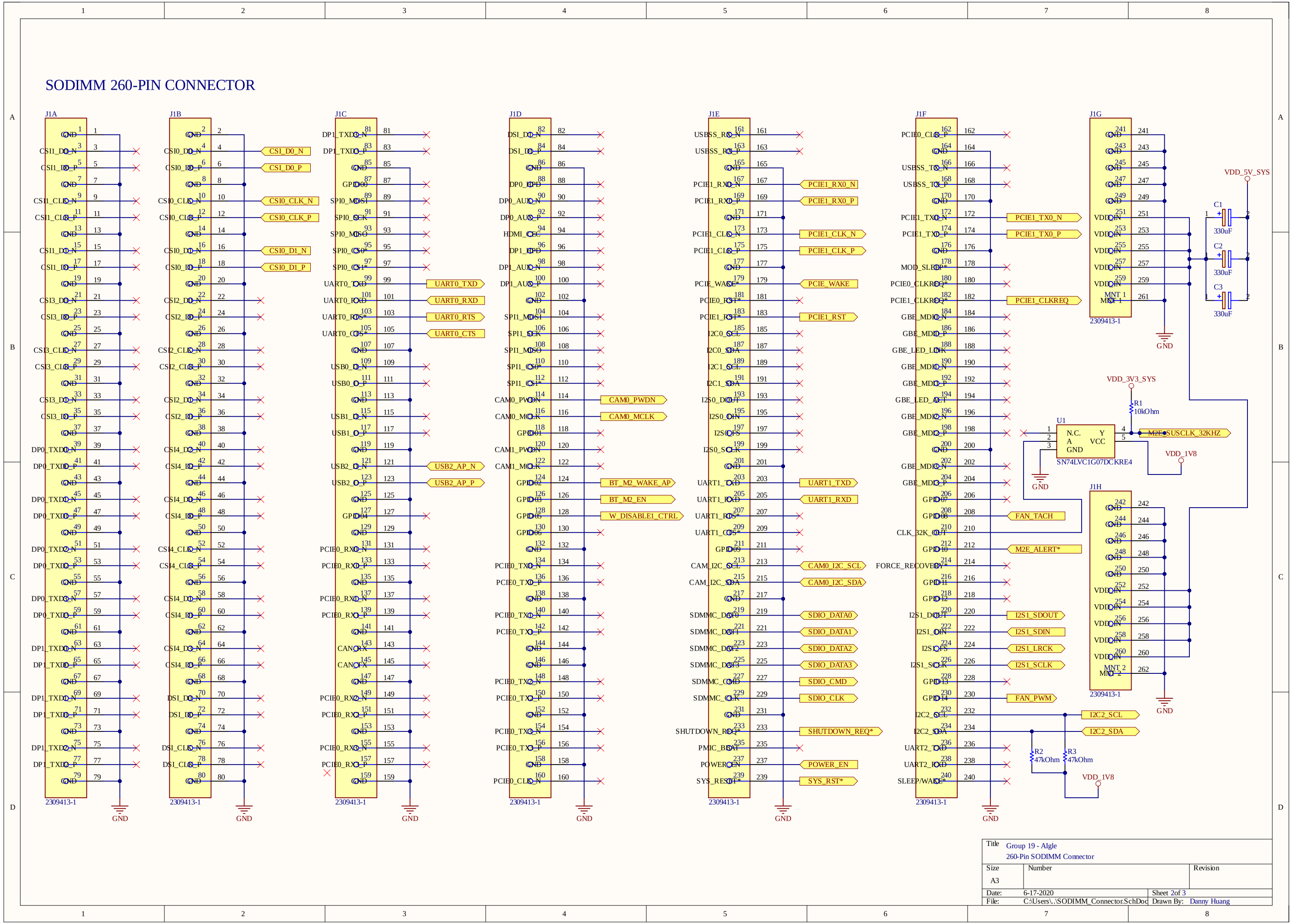
J6 — 533980471

GND	1	
PWR	2	VDD_5V_SYS
TACH	3	R17 10kOhm → FAN_PWM_Q
PWM	4	C15 0.1uF
MNT 1	5	
MNT 2	6	

Q3 DMN26D0UFB4-7 — R20 10kOhm → VDD_1V8 — FAN_TACH

Title	Group 19 - AIgle I/O Connectors
Size	A3
Number	
Revision	
Date:	6-17-2020
Sheet	1 of 3
File:	C:\Users\..\IO_Connector.SchDoc
Drawn By:	Danny Huang

Figure 8.10: Electronic schematic of the 260-pin SODIMM connector.



Title	Group 19 - AIgle 260-Pin SODIMM Connector		
Size A3	Number		Revision
Date:	6-17-2020	Sheet 2of 3	
File:	C:\Users\..\SODIMM_Connector.SchDoc	Drawn By:	Danny Huang

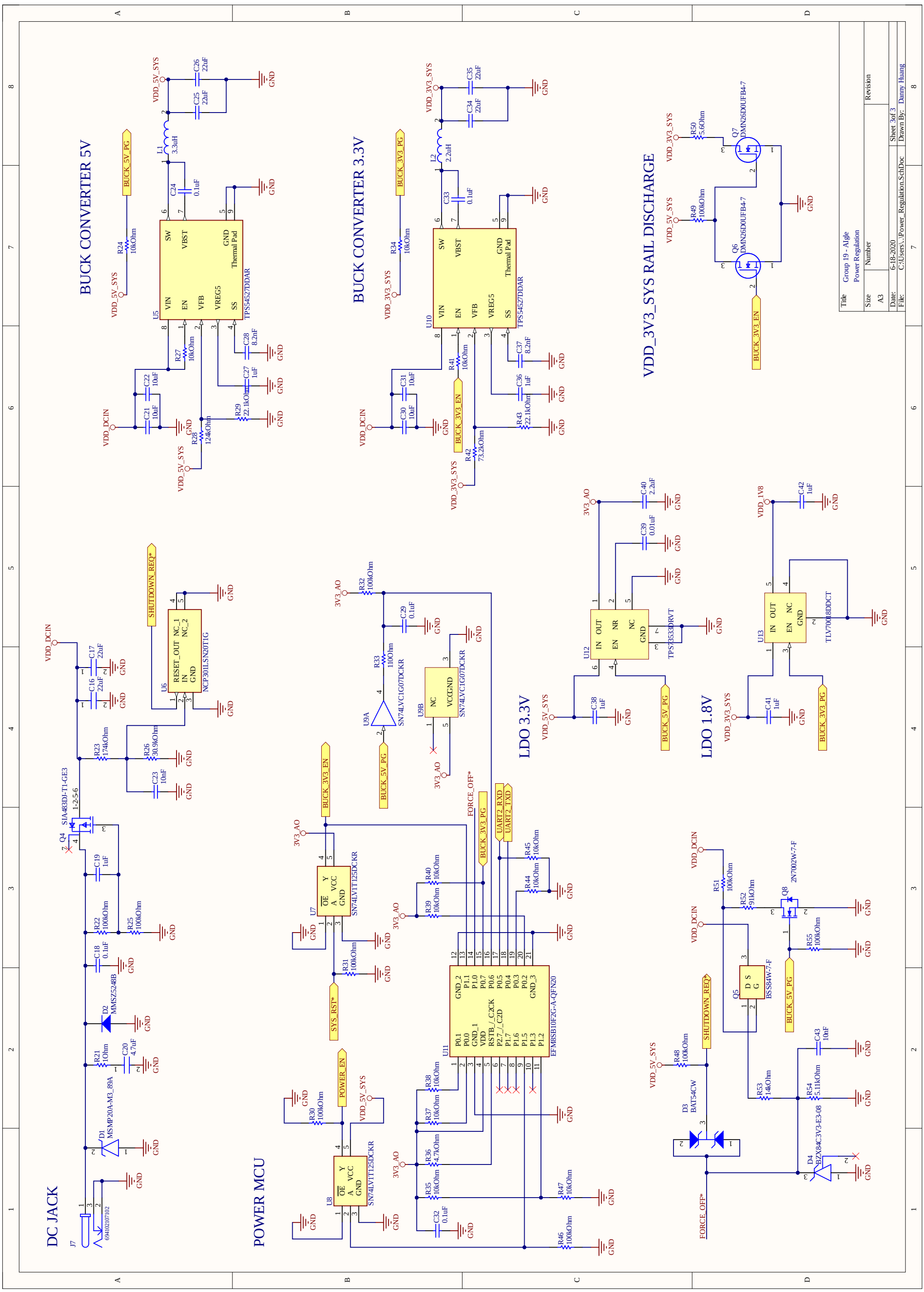


Figure 8.11: Electronic schematic of the power regulation components.

9

Structures

In the structural sizing of the drone, many design decision have to be made regarding, for example, the drone frame, arm dimensions, payload support and aerodynamic shell shape. The requirements that need to be fulfilled are shown in section 9.1. The design approach used is explained in section 9.2. First, the material options are investigated in section 9.3. Because the combination of shapes and materials that can be used for every component are constrained by manufacturing techniques, section 9.4 describes the manufacturing limitations that need to be taken into account in the structural design. With that in mind, the frame layout is discussed in section 9.5. Then, the arms are sized for operational, impact and vibrational loading in section 9.6. A similar approach was used for the payload support in section 9.7. Lastly, the motor mount design is discussed in section 9.8 and finally, the models' verification and validation is presented in section 9.9.

9.1. Requirements

The design of the structures subsystem is based on requirements, which have been setup in the Baseline Report [**BaselineReport**] and which have been refined and updated in the Midterm Report [**MidtermReport**]. The relevant requirements for the structures subsystem are given below. The requirements are as follows:

- **AIG-TE-HW-MP-IRI1**: The drone shall suffer no damage to on-board systems or structure if the drone is dropped from 3 meters onto a concrete floor.
- **AIG-TE-HW-MP-IRI2**: The drone shall suffer no fracture, plastic deformation or loss of electronic function if the drone flies into a gate at maximum normal flight speed.
- **AIG-TE-HW-LR1**: The elastic deformation strain during normal flight conditions shall not exceed the amount where any of the propulsion units or sensors are offset by more than 10 °.
- **AIG-TE-HW-LR2**: No plastic deformation of the drone shall occur during normal flight conditions.
- **AIG-TE-HW-LR3**: No failure of the structural components of the drone shall occur during normal flight conditions.
- **AIG-TE-EB-M1.3**: The total mass of the structures subsystem shall be no greater than 300 g.
- **AIG-TE-EB-M3**: The maximum drone dimension in all length, width and height shall be smaller than 50 cm.
- **AIG-TE-P-M-M1**: All parts of the drone shall be manufacturable using 2D milling, 3D printing and Printed Circuit Board printing machines.
- **AIG-TE-P-M-M2**: All parts of the drone shall be able to be manufactured independently.
- **AIG-TE-P-M-A1**: It shall be possible to assemble the drone using only standard drone assembly techniques.
- **AIG-TE-P-R1**: Each structural component of the drone shall be replaceable within 2 minutes.
- **AIG-TE-P-R2**: Each component of the drone shall be able to be replaced individually.
- **AIG-NT-S2**: The components of the drone shall be interconnected using temporary joints.
- **AIG-NT-S4**: The sustainability of the drone shall be maximized.
- **AIG-NT-U3**: All materials used in the drone shall comply with the RoHS 1 and RoHS 2 restrictions on hazardous substances.
- **AIG-NT-U4**: The drone can be hand-held safely while the rotors are spinning.

9.2. Design Approach

The structural design is an iterative process. It requires inputs from other subsystems, such as the dimensions of the electronic components, the mass of the motors and the propeller size from the propulsion group. In addition, structural parameters that need to be calculated are heavily dependent on each other. Furthermore, the shape and material options are limited by manufacturing constraints. Therefore, iteration is required. An overview of this iterative structural design approach can be found in the N2 chart in Figure 9.1.

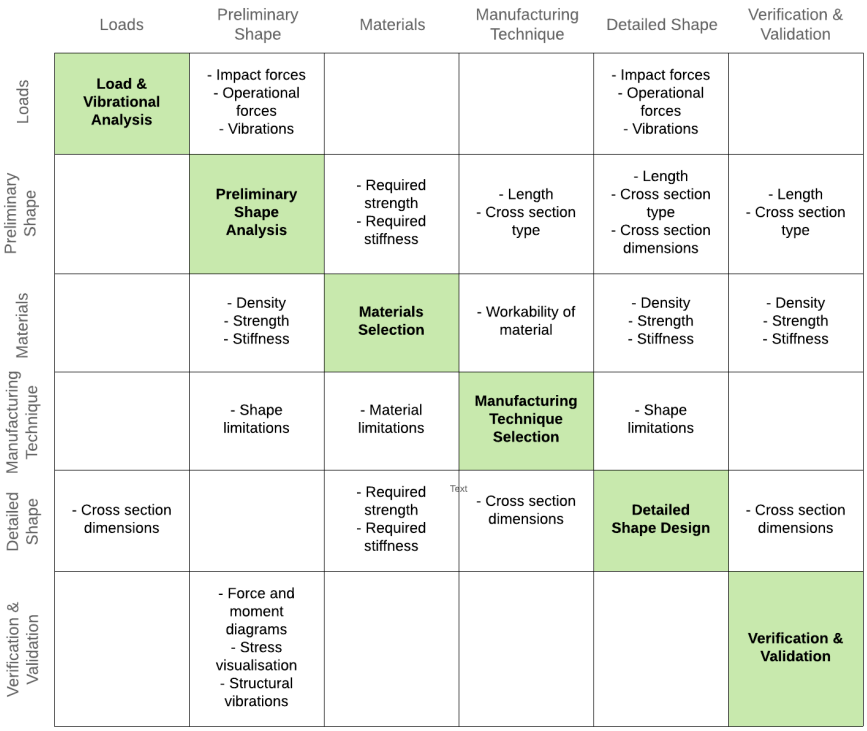


Figure 9.1: The N2 chart for the structural design approach, showing the interdependencies of different design steps

9.3. Material Options

It is important to have a good overview of all the materials and manufacturing techniques that can be used before starting the design of the shape. An overview of the considered materials can be found in Table 9.1 [1] [2] [3] [4] [5] [6] [7] [8] [9] [10] [11] [12]. Material properties like the specific strength, specific stiffness, and

[1] https://smile.amazon.de/KKmoon-Carbonfaserplatten-Carbonplatte-Kohlefaserplatte-gl%C3%A4nzend/dp/B07TD99S3V/ref=pd_sbs_201_2/259-7528502-4145615?_encoding=UTF8&pd_rd_i=B07TD99S3V&pd_rd_r=31ceea3e-ba32-4421-bc6c-183ed295f413&pd_rd_w=l4qCx&pd_rd_wg=L4uOS&pf_rd_p=42bf0ad8-ce6f-4127-a2f0-106727020a41&pf_rd_r=55QX1WNQAFSSV9AFNS9X&refRID=55QX1WNQAFSSV9AFNS9X&th=1, (Retrieved May 15, 2020)

[2] https://amzn.to/2Vvd2t2, (Retrieved May 15, 2020)

[3] https://www.alibaba.com/product-detail/Economy-Glass-Fiber-Reinforced-Plastic-font_62262304823.html?spm=a2700.supplier-normal.35.8.5eb77419VYOKO6, (Retrieved June 12, 2020)

[4] https://www.alibaba.com/product-detail/High-quality-7068-7075-t6-aluminium_62423321551.html?spm=a2700.7724857.normalList.23.7e0b251bk8wJdW&s=p, (Retrieved May 15, 2020)

[5] http://nguyen.hong.hai.free.fr/EBOOKS/SCIENCE%20AND%20ENGINEERING/MECANIQUE/MATERIAUX/COMPOSITES/Carbon%20Fiber%20Composites/91697_04.pdf, (Retrieved June 12, 2020)

[6] http://www.matweb.com/search/DataSheet.aspx?MatGUID=a020d1cceadc4a4ab2216b2f6b5dd458, (Retrieved June 12, 2020)

[7] https://www.3d4makers.com/products/petg-carbon, (Retrieved June 12, 2020)

[8] https://www.3d4makers.com/products/pei-filament,(Retrieved June 12, 2020)

[9] https://ninjatek.com/wp-content/uploads/2018/10/Cheetah-TDS.pdf, (Retrieved June 12, 2020)

[10] https://www.rockwestcomposites.com/media/downloads/Epon-862_(03-2005).pdf, (Retrieved June 20, 2020)

[11] https://www.rockwestcomposites.com/media/downloads/34-700.pdf, (Retrieved June 20, 2020)

[12] https://www.rockwestcomposites.com/media/downloads/Epon-862_(03-2005).pdf, (Retrieved June 20, 2020)

specific cost were accounted for. Specific properties are properties per unit mass, establishing these allows to compare the materials based on their performance over weight, This is crucial to consider to minimise the weight as much as possible. Materials should be strong and stiff but a rubber-like material is also included, which can be used for the dampening of the structural components.

Table 9.1: Comparison of materials considered for structures [21][22][23]

Material	**Yield Strength** $[MPa]$	**Ultimate Strength** $[MPa]$	**Youngs Modulus** $[GPa]$	**Density** $[kg/m^3]$	**Specific cost** $[€/kg]$	**Specific strength** [Y.Strength/rho]	**Specific stiffness** [YoungsMod/rho]
Carbon Fibre	-	2572 (tension) 1365 (compression)	137 GPa (tension) 127 GPa (compression)	1800	277	1.43 (tension) 0.758 (compression)	0.076 (tension) 0.070 (compression)
Resin Epon 862	-	65.63	3.18	1174	-	0.0559	0.003
60 % Glass Fibre Reinforced Polymer	-	130		1550	146	0.083	0.005
Aluminium	655	683	73.1	2850	2.5	0.23	0.026
PLA - 3D printing	45	50	3	1240	20	0.036	0.002
Polyether Imide (PEI) ULTEM 1010 - 3D printing	–	105	3.2	1270	280	0.083	0.003
PETG CARBON FILAMENT - 3D printing	–	53	4	1317	102	0.04	0.003
Rubber - 3D printing	9	39	0.026	1220	85	0.007	$2 * 10^{-5}$

9.4. Manufacturing Techniques

While designing the structural components, not only the materials need to be accounted for, but also the feasibility of manufacturing techniques, given the shape, materials and customer constraints. Due to the requirements, either 2D milling and lathing can be used, where stock material would be bought and machined to the desired shape, or 3D printing, where material is added layer by layer. Parts that cannot be manufactured due to equipment limitations, cost, and time constraints need to be ordered online. A special case to be discussed is whether the carbon fibre lay-up with a 3D printed mould might be worth the effort and cost, especially for the frame cover. Note that the part design is a concurrent process. First, information is gathered to compare the manufacturing techniques and materials and after that, the material and shape can be selected for each part. After that, it is checked if and how it is possible to manufacture every component.

Another point that influences this choice is the availability and effort required to build the parts. This includes the availability of outsourcing possibilities or the equipment it would require to make the part yourself. It also accounts for the time it takes to make the part with the manufacturing technique, which is important for repairs. A minor consideration is the cost that the respective process requires. It speaks for itself that cheap processes are preferred. However, considering that the drone cost is significantly lower than the required maximum amount, this criterion can be evaluated with a low priority.

A trade-off between the possible manufacturing techniques is performed by giving qualitative scores in the form of pluses and minuses. The highest score is indicated with two pluses, and the lowest score with two minuses. The results are presented in Table 9.2.

9.5. Frame Layout Selection

The following section explains how the frame layout is chosen, by determining the configuration of the arms, the modularity and the main frame's characteristics. Aspects like performance, manufacturability and repairability are taken into account.

9.5.1. Frame Configuration

A deadcat frame was selected in the preliminary design phase. It is a variation of the X-frame where the forward arms are spread wider apart, such that the camera has an unobstructed field of view. Other characteristics of this frame are a high torsional rigidity, a centralised mass distribution, thus low mass moment of inertia allowing for a good agility performance, and a sufficient mounting space

Table 9.2: Comparison of allowed part manufacturing techniques [3][24][25]

Manufacturing Technique	Shape complexity	Materials	Availability	Accuracy	Effort	Cost
2D Milling & Lathing	Basic 2D shapes	Any material	Requires experience, minimum equipment, DIY or outsource	Medium accuracy (dependent on tooling and experience)	Minutes - hours (dependent on size and shape)	Medium equipment cost, low outsourcing cost
Score	-	++	+	+	+	+
3D Printing	Highly complex shapes	PLA, powder and filaments. Carbon fibre possible with material property degradation	No experience required, 3D printer, DIY or outsource	Medium to high accuracy (dependent on 3D printer)	Minutes - hours (dependent on size and shape)	High equipment cost, low outsourcing cost
Score	++	-	++	+	+	+
Carbon-fibre layup	Customized 3D shapes possible, limited by complexity	Only Carbon fibre	Requires experience, special tooling, outsource only	high accuracy	Hours	High cost
Score	+	+	–	++	-	-

for the electronics.

9.5.2. Arms Attachment

The two durability objectives to be met are rigidity and modularity. The former indicates that the frame shall protect the electronics in crash as set by the customer requirements, the latter implies that components have to be easily reparable at low cost. For that reason it was chosen to separate the arms from the body, to make them replaceable rather than discarding the whole drone body in case of damage. A drawback is that this adds a little more weight to the structure, however this does not overcome the advantage of having a modular design.

9.5.3. Body Frame Selection

For the AIgle drone design a monocoque main frame has been chosen. This means, that the main frame is one closed box that carries all the loads. There are openings in this box allowing access to the on-board systems, and for air to cool the processor.

In contrary, a flat base plate body on which all electronics are mounted was considered. Often this also includes a top plate to cover up the electronics.

The advantage of the monocoque frame is, that many functions can be incorporated into one part. This means that besides being the load bearing connecting point between the arms and the on-board systems, it can also act as protective shell. Furthermore this allows the drone to be shaped in a more aerodynamic way, reducing the drag and allowing air to be channelled efficiently to the cooler.
In addition, there is more freedom in placing the arms at the desired height, such that the center of thrust is on the same plane as the center of gravity. The choice for a monocoque frame also leads to the necessity of 3D printing because of its closed shape. On one hand this limits the choice of materials significantly, but on the other hand this decision also allows for very specialized shapes which would not be possible with 2D milling. Additionally, 3D printing is much easier than 2D milling, because minimum practice is required to 3D print the existing CAD model.

9.6. Drone Arm Design

9.6.1. Material Selection

The arms of the drone have two main functions. First, the arms shall transfer the thrust loads from the motors to the main body, secondly they need to take up the loads during crash. For this, both a high strength and stiffness are required. In addition, to minimise the weight, a low density is favoured. The material with the highest specific strength and stiffness is carbon fibre. Note that Table 9.1 specifies carbon fibre properties for a filament wound carbon fibre tube, where the tensile and compressive properties are already normalised to a 60 % fibre volume content according to the data sheet [13].

[13]https://www.rockwestcomposites.com/media/downloads/34-700.pdf, (Retrieved June 18, 2020)

However, the Young's modulus follows from the rule of mixtures as in Equation 9.1. The fibre volume content f, was set to a value of 0.6. B is a fraction that accounts for the fibre orientation, which is not unidirectional in this case. By accounting for a random distribution of the fibres loaded in three directions rather than unidirectional, B takes a value of 0.2 in the worst case [26]. The values of the Youngs modulus of carbon fibre $\mathrm{E_f}$ and the resin $\mathrm{E_r}$ were taken from Table 9.1. This resulted in a tensile Young's modulus of 82.45 GPa and a compressive Young's modulus of 76.45 GPa. For the combined density, a similar approach is used as in Equation 9.2. This yields a density of 1549.6 $\mathrm{kg/m^3}$ for the composite. The specific stiffness in compression is thus found to be 0.049, which is still significantly higher than the specific stiffness of the other materials in this comparison. One of the drawbacks is the high specific cost of the composite, but as the cost is not a main constraint over performance, a carbon fibre composite was selected as the material for the arms.

$$E_c = E_f f + B E_r (1 - f) \tag{9.1}$$

$$\rho_c = \rho_f f + \rho_r (1 - f) \tag{9.2}$$

9.6.2. Cross Sectional Shape Selection

The cross sectional shape of the arms is decided to be a hollow tube. In the next part, the design approach is explained in more detail.

Several cross sectional shapes were considered for the drone arms. For example a solid square, hollow square, solid rectangle, hollow rectangle, a solid cylinder and a hollow cylinder.
The drone arm is loaded both in axial compression and bending. Thus both the axial stiffness, which is proportional to the cross sectional area, and the bending stiffness, which is proportional to the area moment of inertia were evaluated. In order to compare the different options, the cross sectional area has been fixed to 1 $\mathrm{cm^2}$. This means that the different options have the same mass per unit length, and also the same axial stiffness when the same material and beam length are considered. This way, only the bending performance needs to be evaluated. This bending performance is influenced by the shape and therefore the area moment of inertia could be used to compare the different options. In Table 9.3 the different shapes are listed together with their properties and how they were derived.

Table 9.3: Comparison of part manufacturing techniques that can be used

Shape	Formula for Area	Dimensions for A=1cm^2	Formula for Area Moment of Inertia	Area Moment of Inertia [cm^4]
Solid Square	a^2	$a = 1$	x,y: $\frac{a^4}{12}$	x,y: 0.083
Hollow Square	$a^2 - (a-2t)^2$	$a = 2$ $t = 0.13$	x,y: $\frac{a^4-(a-2t)^4}{12}$	x,y: 0.058
Solid Rectangle	$a \cdot b$	$a = 0.33$ $b = 3$	x: $\frac{ba^3}{12}$ y: $\frac{ab^3}{12}$	x: 0.009 y: 0.74
Hollow Rectangle	$a \cdot b - (a-2t) \cdot (b-2t)$	$a = 1$ $b = 3$ $t = 0.135$	x: $\frac{ba^3-(b-2t)\cdot(a-2t)^3}{12}$ y: $\frac{ab^3-(a-2t)\cdot(b-2t)^3}{12}$	x: 0.16 y: 1.01
Solid Circle	πr^2	$r = 0.56$	x,y: $\frac{\pi r^4}{4}$	x,y: 0.08
Hollow Circle	$\pi r^2 - \pi(r-t)^2$	$r = 1.5$ $t = 0.11$	x,y: $\frac{\pi(r^4-(r-t)^4)}{4}$	x,y: 1.04
I-Beam	$at + 2bt$	$a = 3$ $b = 1$ $t = 0.2$	x: $\frac{ta^3}{12}+2(\frac{bt^3}{12}+bt\cdot\frac{A+t}{2})$ y: $\frac{at^3+2tb^3}{12}$	x: 1.09 y: 0.04

From this comparison it can be seen that the hollow cross sections and the I-beam have the highest area moment of inertia, thus the highest bending stiffness when comparing the same material and

beam length. The I-beam has a high moment of inertia in the x-direction, while this value is lower in y-direction. The moment of inertia of the hollow circle is the same in both directions. Because it is unpredictable in which direction the impact loads will act, a unidirectional bending stiffness is preferred. Therefore, the hollow cylinder was selected as the cross section of the arms.

9.6.3. Loading Conditions

To evaluate all loading conditions, first a reference frame was established. Here the X-axis points in forward flight direction, the Z-axis points downwards and the Y-axis completes the right handed coordinate system. The origin is fixed in the center of gravity as shown in Figure 9.2.

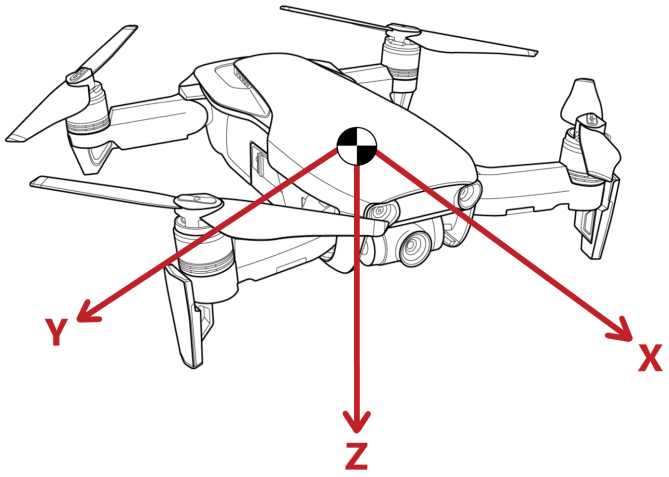


Figure 9.2: The coordinate system used for structural analysis [14]

Operational Loads: Take-Off

The first loading condition that was evaluated for sizing the arms is take-off. Here, the propellers generate maximum thrust leading to a thrust over weight ratio of 4.1. The arms were modelled as a fixed cantilever beam, and the force generated by the propellers as well as the weight of the motor was modelled as a point load on the tip. The weight of the arms itself was considered as a distributed load per unit arm length: $q_a = \rho g A$. However its value is so small that it turned out to be negligible in the stress analysis. The free body diagram is shown in Figure 9.3.

[14] https://dribbble.com/shots/7084995-Mavic-Air-Line-Art-Drawing, (Retrieved June 14, 2020)

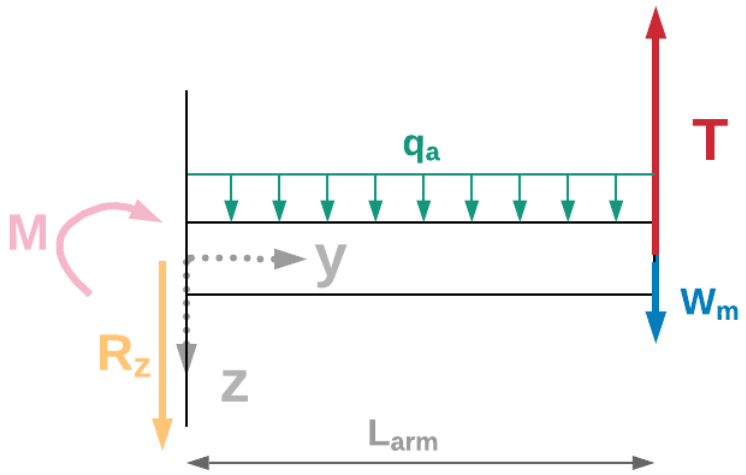


Figure 9.3: Free Body Diagram and sign convention of a beam in bending with distributed load q_a, and forces at the tip of the beam (t and W_m)

The reaction force R_y and reaction moment M were evaluated with Equation 9.3 and Equation 9.4 once the drone has reached a constant velocity, thus equilibrium conditions hold.

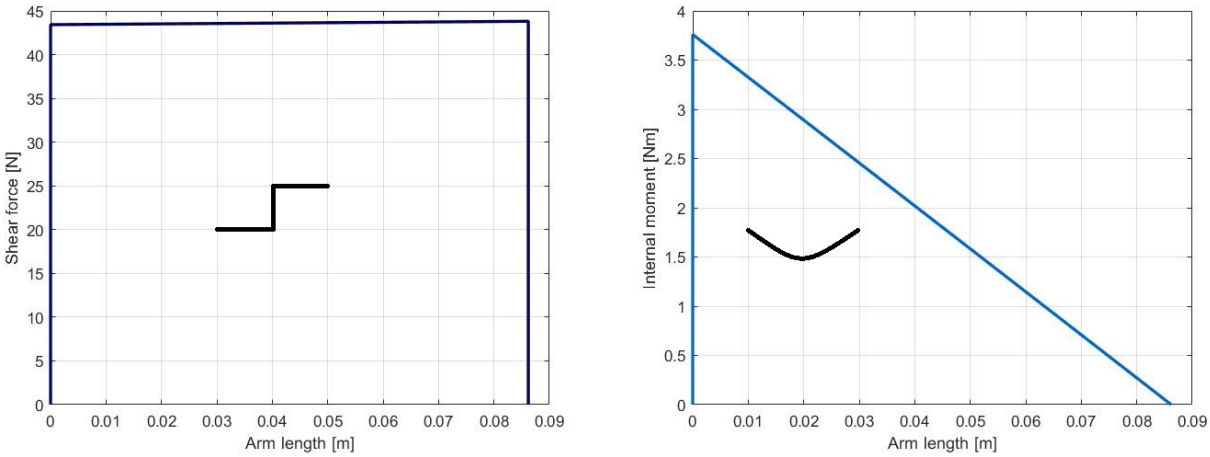
$$R_z = T - W_m - q_a L_{arm} \tag{9.3}$$

$$M_r = TL_{arm} - W_m L_{arm} - \frac{1}{2} q_a L_{arm}^2 \tag{9.4}$$

The loading diagrams were generated with Equation 9.5 and Equation 9.6, where $[y]^0 = 0$ if y < 0, and $[y]^0 = 1$ if y ≳ 0. The loading diagrams are shown in Figure 9.4.

$$V = -R_z[y]^0 - q_a[y]^1 + T[y - L]^0 - W_m[y - L]^0 \tag{9.5}$$

$$M = M_r[y]^0 - R_z[y]^1 - \frac{1}{2} q_a[y]^2 + T[y - L]^1 - W_m[y - L]^1 \tag{9.6}$$



(a) Shear force (b) Moment

Figure 9.4: The loading diagrams of the drone arms during take-off

Then, the axial and shear stresses are evaluated with Equation 9.7, Equation 9.8 and Equation 9.9 respectively, and the equivalent von Mises stress was obtained with Equation 9.10. The von Mises stress turned out to have an order of magnitude of 2 -5 MPa for various thickness radius conditions that seemed achievable in the design, which is much lower than the ultimate strength of 1345 MPa of the composite. For that reason, this loading condition was not considered as critical for the drone arm design.

$$\sigma_a = \frac{R_y}{A} = 0 \tag{9.7}$$

$$\tau_{sh} = \frac{V}{A} \tag{9.8}$$

$$\sigma_b = \frac{-Mz}{I_{xx}} \tag{9.9}$$

$$\sigma_{vm} = \sqrt{(\sigma_a + \sigma_b)^2 + 3\tau_{sh}^2} \tag{9.10}$$

Impact: 3-Meter Drop

Secondly the arms are required to survive a 3 meter drop on concrete. To do so, the drone was modelled as a point mass m_2, and the arm that hits the ground as a spring with stiffness k_2, as shown in Figure 9.5. Furthermore m_1 is the mass of the concrete ground, and k_1 its stiffness.

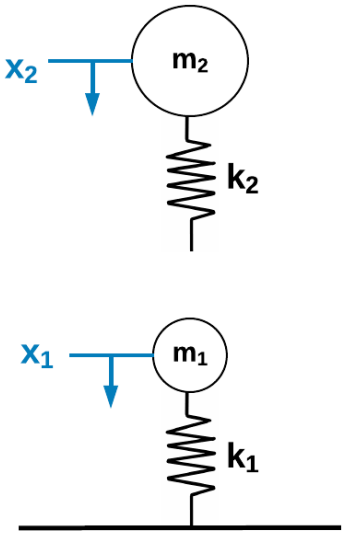


Figure 9.5: The free body diagram of the drone, presented as m_2 with one arm having stiffness k_2 dropping on the concrete ground presented with m_1 and k_1. Note that when hitting the ground, k_2 shall be in contact with m_1

In Equation 9.11, the equations of motion belonging to the lumped mass-spring system are presented.

$$m_1\ddot{x_1} = -k_1x_1 + k_1x_2 \tag{9.11}$$

$$m_2\ddot{x_2} = k_1x_1 + (-k_1 - k_{2_a})x_2 \tag{9.12}$$

The most critical design scenario for the arms is when the drone arm drops vertically on the ground, rather than at an angle. This is because carbon fibre was selected as the material, which has a significant lower compressive strength than tensile strength. In bending, one side of the beam is in

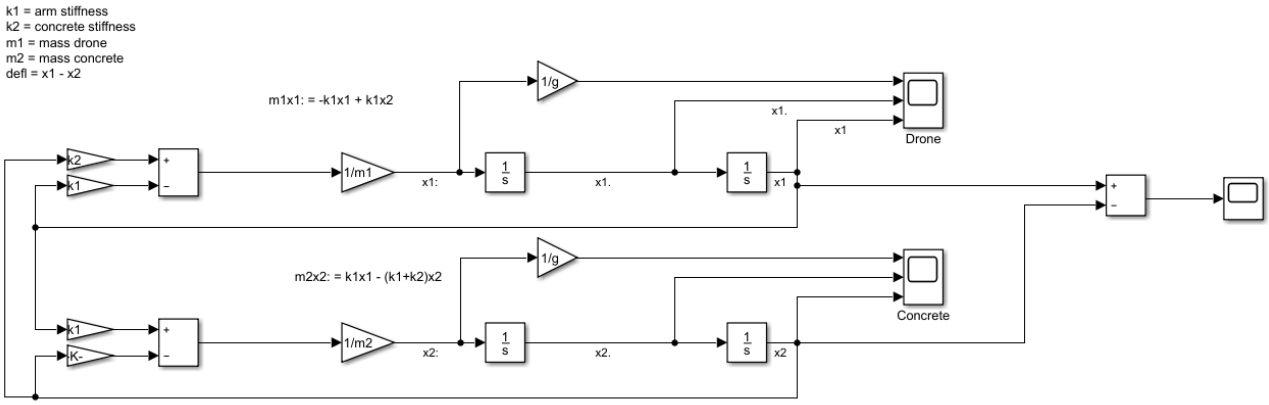


Figure 9.6: The Simulink model for a two degrees of freedom lumped mass spring system.

compression, whereas the other half is in tension, thus pure compression is expected to be the most critical design scenario.

The velocity at which the arms hit the ground was evaluated with Equation 9.13 that followed from the conservation of energy where potential energy is purely converted into kinetic energy, neglecting drag. This was used as the initial condition for m_2. After hitting the ground, the initial velocity of m_1 followed from the conservation of momentum in Equation 9.14. Since the mass m_1 of concrete was set at a very high value, its initial velocity approaches zero.

$$v_2 = \sqrt{2gh} \tag{9.13}$$

$$v_1 = \frac{m_2 v_2}{m_1 + m_2} \tag{9.14}$$

Then, the displacement of the spring was found with equation Equation 9.15.

$$\delta_a = x_2 - x_1 \tag{9.15}$$

Those equations of motion were modelled in Simulink to evaluate the displacement, velocity and acceleration of the point masses in the system, as shown in Figure 9.6. The inputs values for m_2 was taken to be the total mass of the drone, for which a value of 1.29 kg was found after iteration. With regards to the concrete ground, its mass m_1 and its stiffness k_1 are set as a very high value of 10^{10} in the simulation. Because this value is very high, further increasing or slightly decreasing their magnitude did not affect the results of the simulation.

The value for k_2 was found from iteration. This iterative process is described in Equation 9.6.3 to make it compliant with all loading cases.

Impact: Gate Crash

A similar simulation was conducted for the gate crash. To start with, the drone was again modelled as a point mass m_2 with a forward velocity $\mathrm{v}2$. The arms, for which a deadcat frame was selected, make an angle θ with the symmetry axis of the drone as shown in Figure 9.7. Then, a local coordinate system was defined, where the x-axis is aligned with the longitudinal axis of the drone arms, and the y-axis is perpendicular to that seen from a top-view.

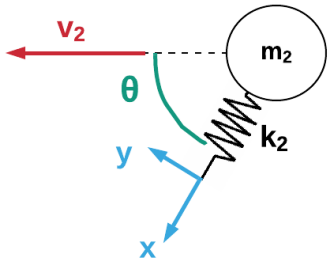
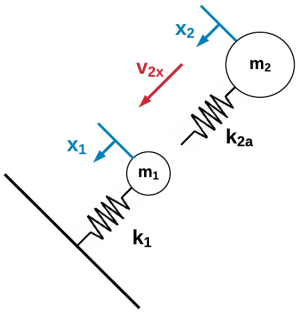


Figure 9.7: The Free Body Diagram for the gate crash scenario under an angle θ

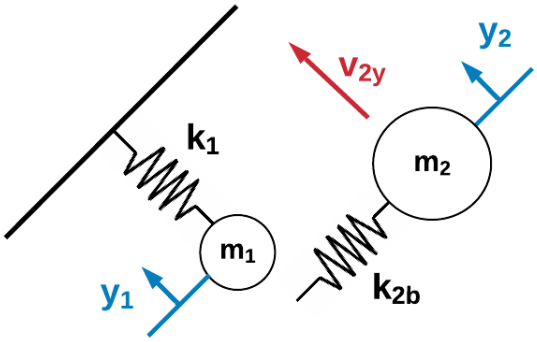
The velocity components were split in axial and perpendicular velocities with respect to the frame, computed with Equation 9.16. This is a valid assumption since the conservation of energy as in Equation 9.17 still holds. The lumped mass-spring systems are presented in Figure 9.8a and Figure 9.8b.

$$v_{2_x} = v_2 \cos\theta \qquad v_{2_y} = v_2 \sin\theta \tag{9.16}$$

$$E = \frac{1}{2}m_2 v_2^2 + \frac{1}{2}k\delta^2 + \frac{1}{2}I\omega^2 = \frac{1}{2}m_2 v_{2_x}^2 + \frac{1}{2}m_2 v_{2_y}^2 + \frac{1}{2}k_a\delta_x^2 + \frac{1}{2}k_b\delta_y^2 + \frac{1}{2}I\omega^2 \tag{9.17}$$



(a) Axial



(b) Bending

Figure 9.8: Lumped mass spring systems

The equations of motion corresponding to both systems are given in Equation 9.18 and Equation 9.20. m_1 and k_1 are here the mass and stiffness of the gate respectively.

$$m_1\ddot{x_1} = -k_1x_1 + k_1x_2 \tag{9.18}$$
$$m_2\ddot{x_2} = k_1x_1 + (-k_1 - k_{2_a})x_2 \tag{9.19}$$

$$m_1\ddot{y_1} = -k_1y_1 + k_1y_2 \tag{9.20}$$
$$m_2\ddot{y_2} = k_1y_1 + (-k_1 - k_{2_b})y_2 \tag{9.21}$$

Equation 9.22 holds for the initial velocity of the gate m_1, where again the conservation of momentum was used. The displacement of the arms for both loading cases was found with Equation 9.24.

$$v_{1_x} = \frac{m_2 v_{2_x}}{m_1 + m_2} \tag{9.22}$$

$$v_{1_y} = \frac{m_2 v_{2_y}}{m_1 + m_2} \tag{9.23}$$

$$\delta_a = x_2 - x_1 \tag{9.24}$$

$$\delta_b = y_2 - y_1 \tag{9.25}$$

Again, this lumped mass-spring system was modelled in Simulink. The drone mass m_2 was taken to be 1.29 kg. The gate characteristics were specified for two different gates types. In the IROS competition, a lightweight gate with a mass m_1 of 3 kg was taken. It is modelled as a hollow carbon fibre beam impacted at its half length, having a total length of 1.2 meter, a diameter of 5 cm and a thickness of 3 mm. The material of this gate is carbon fibre, for which the same composite E modulus of 76.45 GPa was assumed. Its moment of inertia (I_{gate}) was then found with Equation 9.26, and its stiffness with Equation 9.27. This yielded an IROS gate stiffness k_1 of $3.43 \cdot 10^4$ N/m.

$$I_{gate} = \frac{\pi}{64}\left[D_{gate}^4 - (D_{gate} - t_{gate})^4\right] \tag{9.26}$$

$$k_1 = k_{gate} = \frac{48 E_{gate} I_{gate}}{5 L_{gate}^3} \tag{9.27}$$

Where D_{gate} is the diameter of the gate. The same equations were used to compute the characteristics of a Drone League Race (DRL) gate, which is also an autonomous drone racing competition. Here, a gate mass of 100 kg was assumed for a steel gate, covered with a foam layer. With gate length of 2 meters, a Youngs modulus of 0.745 MPa for a foam, and a solid cylinder with a diameter of 15 cm, a DRL gate stiffness k_1 of $4.44 \cdot 10^4$ was found.

It was noticed that the results are mostly affected by the gate mass, whereas the gate stiffness does not affect the arm deflection much, even if the magnitude of k_1 changes significantly. For that reason, the stiffness of the gate was not worked out with a higher accuracy.
It immediately turned out that a higher gate mass results in higher arm deflections. This makes sense as the gate is harder to move, thus the arms take up that displacement. In the simulation this was reflected in the equations of motion that account for a lower m_1, as well as in the conservation of momentum giving m_1 a lower initial velocity. Because the DRL gates have the highest gate mass, those represent the most critical condition to design for. Therefore, the required arm stiffness was designed to sustain a gate crash in a DRL gate.

Results

As mentioned before, the drone arms were sized using an iterative process. The maximum displacement for bending and axial loading was computed from material properties and dimensions with Equation 9.28 [27].

$$\delta_a = \frac{\sigma_a L}{E} \qquad \delta_b = \frac{\sigma_b L^2}{3ER} \tag{9.28}$$

First the arms were seized to withstand axial loading. For an arm length of 10 cm, that was found after iteration to ensure a sufficient propeller clearance with the body, a Youngs modulus of 76.45 GPa and a compressive strength of 1345 MPa, the maximum deflection for axial displacement of the drone arms was found to be 1.69 mm. With a safety factor of 1.5 on the stress, the maximum deflection may not exceed 1.13 mm.

With a deadcat frame, the front arms are spread wider apart than the aft arms to have a wide camera field of view. For the initial seizing, the angle theta was taken to be 57 degrees, which is half the field of view angle of the camera. This was done to ensure that the propellers are outside the field of view of the camera. The magnitude of k_2 was iterated in the Simulink model, until the displacement output reaches a value lower than 1.13 mm. It was found that the gate crash at 30 m/s was the most critical case, therefore this model was used for seizing the arms. The first value of the axial arm stiffness that showed an arm displacement lower than approximately 1.1 mm, was $k_{a_{req}}$ = 2.0 $\cdot 10^8$.

By knowing this axial stiffness, the required dimensions of the drone arm were derived. In the axial stiffness formula, Equation 9.29, the area was isolated. This resulted in Equation 9.30. As an example, a carbon fire tube with a radius of 1.5 cm combined with a thickness of 3mm would be sufficient to meet the axial loading requirements.

$$k_a = \frac{AE}{L} \tag{9.29}$$

$$A = \pi \left[R^2 - (R-t)^2\right] = \frac{k_{a_{req}} L}{E} \tag{9.30}$$

Secondly, it was verified how to meet the bending requirements as well. The maximum displacement for bending is found with Equation 9.31.

$$\delta_b = \frac{\sigma_b L^2}{3RE} \tag{9.31}$$

Since this displacement also depends on the value for R, a value for the radius was required as an input. The higher the value for R, the higher the moment of inertia as in Equation 9.32, thus the better bending performance of the beam. When considering the maximum radius that is feasible for the structure of the arms, a maximum value for R of 1.5 cm was selected. That yields an outer diameter of 3cm, as which does barely exceed the dimensions of the motor. This value was used to perform the first iteration that gives a maximum displacement evaluated with Equation 9.31 of 3.58 mm with a safety factor of again 1.5 on the stress.

$$I = \frac{\pi}{4}\left[R^4 - (R-t)^4\right] = \frac{k_{b_{req}} L^3}{3E} \tag{9.32}$$

Again an iteration was performed, until the displacement output from the Simulink model was smaller than 3.58 mm. This requires a bending stiffness $k_{b_{req}}$ of $7 \cdot 10^7$ N/m or higher. However, when computing the thickness of the arms that meet those stiffness requirements with Equation 9.32 and Equation 9.33, even a solid cylinder with a thickness of 1.5 cm was not sufficient to achieve the desired bending stiffness.

$$k_b = \frac{3EI}{L^3} \tag{9.33}$$

This unreasonably high thickness required is caused by assumptions made in the model. As an effect of not accounting for the rotation, the model assumes that all kinetic energy of the drone is converted into displacements. However in reality, a high proportion of the bending stresses is converted into rotational motion. The equations of motion in the model are given in Equation 9.34. Here, I is the rotational moment of inertia about the contact point of the arm with the wall, equal to $I_0 + m_{drone} L_{arm}^2$. θ can be linked to the displacement with $\theta = \frac{x_2}{L_a rm}$.

$$I\ddot{\theta} + k_b \theta = 0 \tag{9.34}$$

In addition, the arms were modelled as a clamped beam. However, in reality the root of the arms are only dynamically constrained by the mass of the drone, in contrast to a fixed constraint from which the thickness was computed.

If it was desired to design the drone arms to comply with the stiffness and deflection requirements of the model, three design options could have been considered. First, the radius of the beam can be increased to meet the bending requirements. Whereas it slightly decreases the maximum allowed deflection which is proportional to $1/R$, it highly increases the bending stiffness being proportional to R^4 that could be used as an input that is iterated over. As an example, a carbon fibre tube with a diameter of 6 cm and a thickness of 3.8 mm would just meet the bending requirements.
Secondly, the angle θ that the arms make with the symmetry plane can be reduced. When doing so, the axial over perpendicular velocity ratio: v_{2_x}/v_{2_y} increases, than a higher axial displacement needs to be accounted for, and a lower perpendicular displacement, lowering the required bending stiffness. However, this is not feasible, as it would obstruct the camera field of view in addition to increasing the required axial stiffness, what should be iterated over than again.
Also the material properties can be changed. However, no material with a higher specific strength and stiffness than carbon fibre was considered, and it is doubtful whether other materials can meet those stiffness requirements without increasing the dimensions. Therefore, this option was also eliminated.
Lastly, the bending stiffness increases when lowering the length of the arms. This is also no option, as the propeller blades will than touch the frame.
For those reasons and by using common sense, it was chosen to only use the model for the axial seizing of the arms. The model was considered to be valid for axial loading, as in pure axial loading, no rotational motion occurs. However, the axial stiffness of the beam was still evaluated with the fixed beam theory, whereas this constraint is in reality, again, only is a dynamic constraint of the mass of the drone itself. How this exactly affects the stiffness Equation 9.29 is recommended for to evaluate in further research. Also the bending performance needs to be verified with testing, or by another model in future work.

By looking what carbon fibre tubes are commercially available when designing for meeting the axial loading requirements, a tube with an outer diameter of 2.858 cm, and a thickness of 3.18 mm was selected. The axial and bending arm stiffness would be $2.042 \cdot 10^8$ N/m and $5.56 \cdot 10^6$ N/m respectively, which meets the axial loading requirements.

Vibrations

Besides the impact loads which act on the drone over a short amount of time, the drone also experiences dynamic loading i.e. vibrations induced by the motors. When the forced frequency of those induced vibrations lies too close to the natural frequency of a part of the drone, the part starts oscillating at increasing amplitudes. Especially the on-board systems like the camera, IMU and ESC shall not vibrate at high amplitudes, because that lowers the accuracy of their data outputs. Therefore it is crucial to avoid natural frequencies that lie too close to the forced frequency in the design, as well as designing a damping system lowering the amplitude of the vibrations when the former cannot be avoided. This will be the case if the natural frequency lies within 20% above or below the forced frequency [15].
The motors that have been selected are the SunnySky Edge R2306 2300kv motors. With the APC 6x4E propeller they run at 27000 rpm which can be translated to the forced angular frequency of 450 Hz by dividing the angular rate in rpm by 60 seconds. From this the angular velocity can be calculated to be 2827 rad/s by multiplying with 2 π.
The part that lies the closest to the motor is the motor mount and especially the arm. The arm is here the most critical component, because it is one of the longest yet thinnest parts of the drone. Than the natural frequency of the arm is computed by modelling the arm as a massless spring with a sinusoidal point-load, the vibration, at the tip of it. Furthermore a single degree of freedom with undamped free vibration was assumed to evaluate the vibrational modes independently. The spring stiffness for the vibrational modes in axial loading, bending and torsion can be found in Equation 9.35.

[15] Unpublished conversation with Bertrand Mercier

$$k_a = \frac{AE}{L} \qquad k_b = \frac{3EI}{L^3} \qquad k_t = \frac{GJ}{L} \tag{9.35}$$

For all three loading cases, the natural frequency of the arms is found to be much higher than the forced frequency. The most critical value has been found for the bending and is therefore explained in more detail. For the calculations a Youngs Modulus of 76.45 GPa has been used, just as in the crash models. The diameter and thickness have been taken from the arm sizing due to the crash to be 2.86 cm and 3.1mm respectively. The length of the arm was found to be 9.5cm after iteration. From this the spring stiffness of the arm is calculated to be $5.56 \cdot 10^6$ N/m in bending. Then the natural frequency was found with Equation 9.36.

$$\omega_n = \sqrt{\frac{k}{m}} \tag{9.36}$$

The mass that has been taken here is the mass of the motor and propeller at the tip of the arm, m = 34 grams. From that the natural angular frequency of $1.3 \cdot 10^4$ rad/s is calculated. It can be understood from this that the natural frequency of the arm is by more than a factor of 4 larger than the induced frequency from the motor.
Considering the main body, it is much larger than the arms, thus has a higher area moment of inertia, while not being noticeably longer than the arms in length. Only the Youngs Modulus will be a bit lower as it is made out of 3D printed filament. Still it can safely be assumed that under this circumstances the natural frequency of the main body is still significantly larger than the reduced forced angular frequency.
The electronics are protected by vibration dampers. The electronic flight controller as well as the ESC board are connected via dampening standoffs to the base plate of the mainframe. The processor itself is mounted via a spring system, which is also dampening. Both those systems reduce the vibrations induced by the motors sufficiently. The camera is mounted directly on the same baseplate. As can be seen from current FPV racing drones [16] the ESC's, flight controller and camera do not require more elaborate damping of the motors induced vibration.

9.7. Payload Dampening System Design

During impact loading, the electronics experience a high acceleration as the forces only act for a short amount of time on the drone. In this section is it explained how the electronics shock dampers were seized for impact loading.

First the processor was mounted with extra clearance to the flight controller and ESC boards, that can withstand a higher g-load. This clearance allows to use spring dampers for extra cushioning and dampening of the processor during the crash. As this turned out not be be sufficient to protect from the g-forces during the 3m drop scenarios, landing gears on the bottom of the drone were added.

Similarly to the design of the arms, lumped mass spring models have been used in the approach to design a protection system for the electronics. The most critical electronic component is the NX processor, as it can only withstand a maximum g-load of 50g under operational conditions, and 140g with power off. To avoid failure of this expensive component under operational conditions, a spring dampening system is required.

The design for the 3m drop requirement has been approached by modelling the carrier board, the processor and its cooler as a point mass on top of a spring and a damper which in turn are connected to the main body. The main body is also modelled as a point mass and includes the entire drone mass minus the mass of the electronics mentioned above. The mainframe will be connected to the floor through a single spring, which represents the landing gear as a mass-less flexible beam. The floor itself is modelled as solid concrete with an appropriate large mass and stiffness of 10^{15} kg and 10^{10} N/m respectively. The free body diagram of this model can be seen in Figure 9.9.

[16] https://www.getfpv.com/catalogsearch/result/?q=fpv%5C%20race%5C%20drone, (Retrieved June 16, 2020)

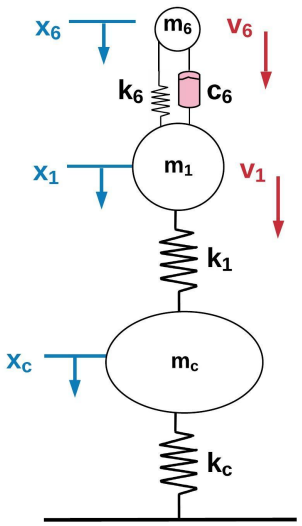


Figure 9.9: Free body diagram of mass-spring-damper system for electronics

The numbering has been chosen to be '6' for the electronics, '1' for the main body and 'c' for concrete. The initial velocity slightly after impact of the concrete was evaluated with the conservation of momentum in Equation 9.37, that approaches zero which makes sense. The initial velocity of both the electronics and the main body are 7.67 m/s after the drop out of 3m height as explained earlier in Equation 9.13.

$$v_c = \frac{m_{drone} v_{drone}}{m_{drone} + m_{concrete}} \tag{9.37}$$

The mass of the electronics sums up to 106 grams (carrier board, processor and cooler), while the mass of the drone has been taken to be the total mass of 1.29 kg minus the electronics mass. Because the space within the drone is very limited, the spring system on which the electronics are mounted is also limited in size. Because the reduction of g-forces is directly dependent on the displacement that is possible, clearance between the electronics on top of the spring system and the other electronics or structures is required. The spring system inside the mainframe should be kept small, preferably occupying less than 1cm of space. For this a solution of Isolation Dynamics Corp has been found on which the carrier board can be mounted with a small connecting part, that will be 3D printed. From them the SM2-044-A damper has been chosen because of its dimensions and its properties [17] [18]. It has a maximum deflection of 8mm under a load of around 28N under a 45°tilted angle, resulting in a spring constant of 4027N/m (23lbs/in). Under compression (0°tilted) the spring constant increases to 9106N/m (52lbs/in). Assuming a linear relationship between those two values, a spring constant of 6847N/m can be achieved under an angle of 20°. The manufacturer mentions a damping ratio of 0.2. This can be translated into a damping coefficient of 10.8 Ns/m using the calculation of Equation 9.38 as shown in [28].

$$\zeta = \frac{c}{c_{cr}} \qquad c_{cr} = \sqrt{mk} \qquad c = \zeta\sqrt{km} \tag{9.38}$$

Here ζ is the damping ratio of 0.2, k the spring stiffness of the damper and m the mass of the electronics. A representation of this damper can be seen below in Figure 9.10.

Because four of these spring-dampers will be used to attach the carrier board to the mainframe, the properties of the overall electronics dampening system can be computed as follows. The four springs

[17] http://www.isolator.com/NewDocs/SMSeries/SM2CurveData.pdf, (Retrieved June 20, 2020)
[18] http://www.isolator.com/NewDocs/SMSeries/SM2CatalogData-Curve.pdf, (Retrieved June 20, 2020)

Figure 9.10: Representation of the damper to be used for mounting the carrier board. [19]

are mounted in parallel meaning that both the spring constants are multiplied by four and the damping constant as well [28]. Those values then can be used as characteristics for k_6 and c_6 in the model depicted in Figure 9.9.
This model has then the following equations of motion in Equation 9.41:

$$m_6\ddot{x}_6 = -k_6x_6 + k_6x_1 - c_6\dot{x}_6 + c_6\dot{x}_1 \tag{9.39}$$

$$m_1\ddot{x}_1 = -(k_6 + k_1)x_1 + k_1x_c - c_6\dot{x}_1 + c_6\dot{x}_6 \tag{9.40}$$

$$m_c\ddot{x}_c = -(k_c + k_1)x_c + k_1x_1 \tag{9.41}$$

From this a Simulink model has been made and the accelerations, velocity and displacement of each component could be plotted versus the time. The only variable that has been left unknown is the spring stiffness of the landing gear, represented by k_1. Now the model could be iterated such that the electronics on top of the spring system would experience no loads higher than 50g. Furthermore it is not allowed for the electronics to displace more than 8mm with respect to the drone frame, because it would otherwise hit the maximum extension of the spring system and increase the g-forces again. From this a spring stiffness of 10.000N/m is found to be suitable, while keeping a safety margin of around 10g to the 50g limit. This spring stiffness will be used for sizing the protective landing gear in CATIA. If desired, this value could be increased a bit in order to come closer to the 50g. It should be noted, that if this is done the stiffness of the electronics spring system, k_6, needs to be increased as well. This can be done by adjusting the mounting angle, which could increase the spring stiffness to maximal 6847N/m. Below the graphs depicting the relative displacement of the electronics to the mainframe and its acceleration are shown in Figure 9.11 and Figure 9.12.

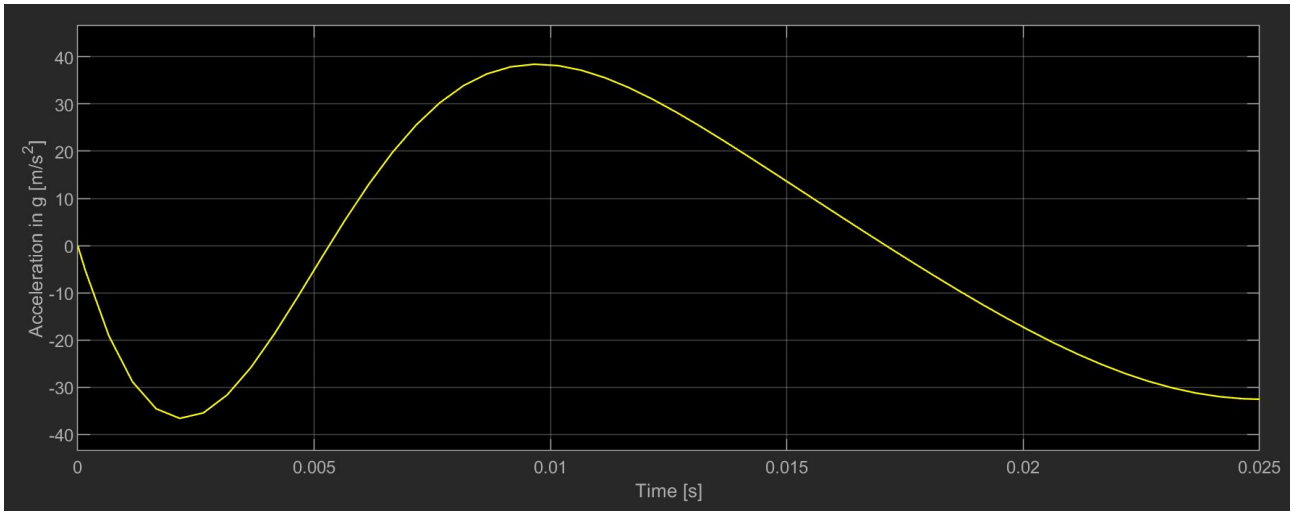


Figure 9.11: Acceleration response of electronics upon impact at 7.67m/s (drop height 3m)

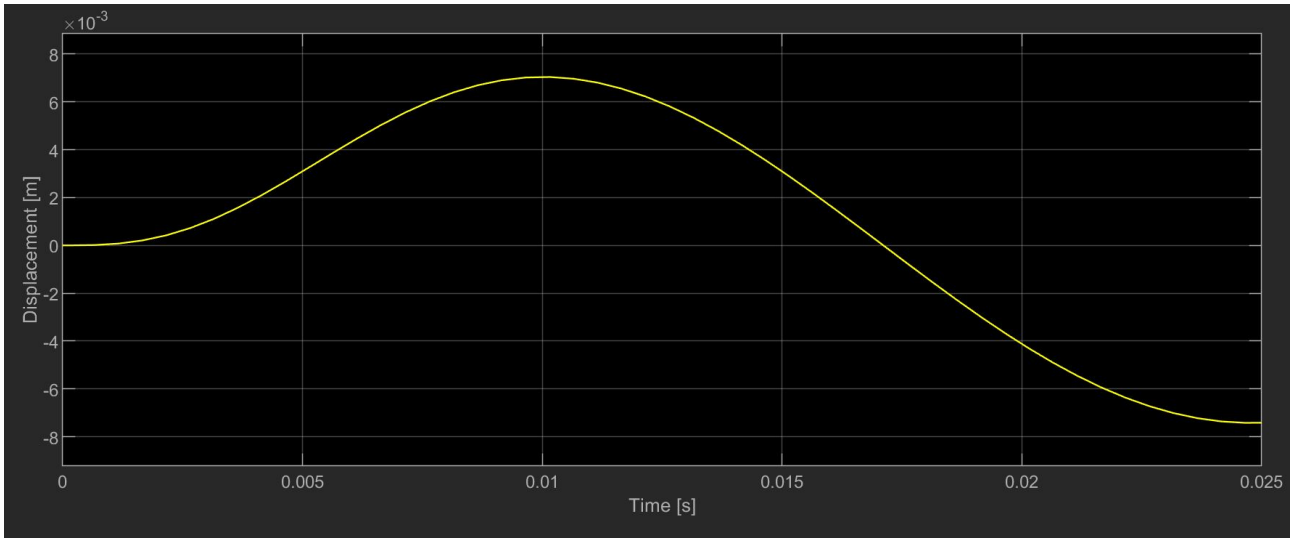


Figure 9.12: relative displacement response of electronics to mainframe upon impact at 7.67m/s (drop height 3m)

It should be noted that in these oscillations only the time frame before the displacement comes back to zero should be looked at [29]. This is, because it is a crash and the drone will behave differently after it hits. As can be seen in Figure 9.12 the crash 'ends' for the electronics at 0.017s after first contact. Within this time frame the second acceleration peak is the critical point as can be seen in Figure 9.11. It can be found that the maximum acceleration lies at 39 grams. The maximum deflection also lies below at around 7mm, which allows for a little larger acceleration even.

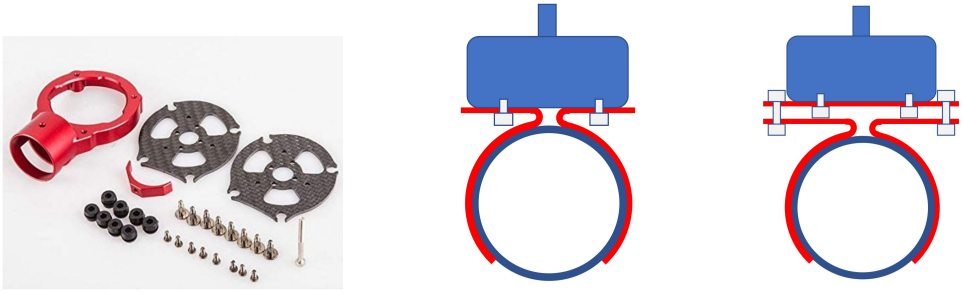
The same approach was used to seize the dampening system for a 30 m/s gate crash, however it quickly turned out that with realistic input values, this requirement cannot be met. Therefore the electronics dampening has not been designed for gate crashes at its maximum speed of 30m/s. To prevent the electronics from failure in a gate crash, it would be suggested to have either a lightweight gate, a highly cushioned gate, a flexible gate or a gate placed on wheels, which prevents the drone from decelerating too quickly when the gate moves along. For example this could be achieved by using gates as in piloted FPV races [20]. They are rather lightweight and resemble more the stiffness of a cloth flag instead of a solid. Otherwise it would also be possible to have cushioning foam on the gates, similarly as they are used on ski slopes.
It should also be noted that the electronic components, such as the processor and the camera, are prevented from direct collision by the cover and the forward facing arms on which the impact force acts directly.

To conclude the design for impact protection for the electronics, the dampening system is only designed to cope with a 3m drop. A crash at 30m/s would require a much larger crushing zone of more than 50cm. Based on this result, it is also recommended to use gates which are flexible during races, to reduce the acceleration loads acting on the drone.

9.8. Motor Mount Design

For mounting the motor, different possibilities were investigated as explained in this section. For arms with a tube diameter of up to 16mm, there are lightweight motor mounts available to be bought off-the-shelf with masses lower than 13 grams per piece, including mounting screws [21].
Unfortunately this is not the case for motor mounts having tube diameters of 20mm upwards. For a motor mount with a 3cm tube diameter, the most lightweight off-the-shelf motor mount that can be found weighs 45 grams [22]. Therefore, different concepts have been investigated as shown in Figure 9.13.

[20] https://www.youtube.com/watch?v=HonTgEm6PhM, (Retrieved June 25, 2020)
[21] https://ebay.to/3ieXXp0, (Retrieved June 17, 2020)
[22] https://www.amazon.nl/HONG-YI-HAT-AntiVibe-CNC30AM-Accessoires/dp/B087RP9426?th=1, (Retrieved June 17, 2020)



(a) Off-the-shelf motor mount [23]

(b) Lightweight critical design solution

(c) Heavy design solution

Figure 9.13: Different motor mount types for visualisation

The motor mount in Figure 9.13a is the most lightweight option that was found for a 3 cm tube, having a total mass of 45 grams. The connection between the circle and the tube connection looks thin and prone to failure during crash. On the other hand, it is made out of aluminium, which allows for yielding before breakage.

Figure 9.14: Mount of plate to tube arms with carbon fibre [30]

The second option in Figure 9.13b is using tube-plate connectors made out of carbon fibre as shown in Figure 9.14. This system has a large contact area between the tube and the connector, which allows for a higher adhesive bonding strength. This design requires a tube of the arm that is 23.4 mm (9.5 grams) longer than for the off-the-shelf option, because the distance between the mounting holes of the motor is 13.4 mm ($2sin(45°) \cdot 19/2$; with 19mm the outer diameter around the mounting holes) plus 5 mm of clearance on each side. From [30] it can be found that two connectors of 23.4mm each, weigh approximately 5.5 grams together. Including 4 screws weighing each 2 grams and assuming an adhesive mass of 8 grams, the total mass for one motor mount sums up to approximately 31 grams.

There is an additional challenge to overcome, which is the space for mounting the screws. As can be seen in Figure 9.13b, the screws heads will touch the connector. This can be prevented by drilling off a part of the connector that will be in the way of the screw head. This does weaken the connector locally.

Therefore the third 'heavy design' option has been considered. As shown in Figure 9.13c the space problem can be avoided by adding another thin carbon plate on top, which will be used for the motor to be mounted on top and the connector mounted below. The screws hold the plates together at the outer part of the connector, such that it does not interfere with the curved section. Still it should be noted, that this includes drilling eight holes into the plate, which also weakens the structure a little, though not as badly as the second option. The mass of this design will be the mass of option 2, plus another set of screws weighing 8 grams and a thin carbon fibre plate, which is assumed to have the mass of one connector, equal to 2.75 grams. As such, the mass of one motor mount for design option 3 would equal 41.75 grams.

The fourth option would be to 3D print a custom designed motor mount. This has the advantage of having only one part and one set of screws. But this option will not be further considered, because the strength of the strongest 3D printing filament is around ten times weaker than the strength of carbon fibre and would therefore require a larger and heavier design, at least heavier than option 1.

From those four options, the second option has been decided upon, which saves more than 10 grams per motor mount in comparison to the off-the-shelf items and the heavy design solution. With four motor mounts in total, the weight savings of 40 grams in total has outweighed the slightly less strong structure.
It should be noted that the mass of the screws are already included in the mass budget of the motor and propeller. The arm length of 9.5 mm was designed for after iteration, including the extra length to be added for the motor mounts. Therefore, the mass accounted for one motor mount as a subpart is 13.5 grams per mount in the mass budget.

9.9. Verification and Validation

9.9.1. Verification

For verifying the calculations that came to the final design choice, different tests and approaches are used to do so.
When building the Simulink models, it has always been started with a single degree lumped mass-spring system. Following this, more detailed and complicated models were written and the results of those were compared to the initial results. Thus, for example, for a drop out of 3m, a single mass-spring system predicted a deflection of around 6cm necessary to reduce the loads on the electronics to 50g. Then in a more detailed model the arms itself were modelled, to further split up the mass for higher accuracy. In this more detailed model it is then checked if the displacements, velocities or accelerations match up with the previously gained results.

In addition, all displacements were evaluated by hand using matrix equations and La Place transformation. Indeed, the output showed approximately the same amplitude for the sinusoidal solution of the displacement x.

Furthermore the electronics cushioning distance was also calculated using a different approach. This approach is used in the packaging industry for shipping electronics, for calculating the minimum required foam thickness for transportation when dropping from a certain height. The method is described in [31] and [32]. The following equation allows to compute the minimum thickness, T_{min}, by dividing the drop height, H, by the fragility, F, which is the amount of 'g' that the electronics can sustain:

$$T_{min} = \frac{H}{F} \tag{9.42}$$

Using the maximum g-rating of the most critical component, the on-board computer with 50 g, and a drop height of 3 m, a minimum thickness of 6 cm foam is required to protect the electronics sufficiently. The foam packaging is of course not suitable for the processor, as it will overheat when in operation. However it gives another way to estimate the cushioning distance that will be required. For the 3m drop calculations, a required displacement of just below the 6cm has also been found in section 9.7. This verifies the approach of using mass-spring-damper models to adequately describe the behaviour of the drone during impact crashes.

9.9.2. Validation

For validation, the design result and the tools used need to be looked at.
For the responses of the drone components due to crash scenarios, Simulink has been used to solve and plot the coupled differential equations. This can be validated by comparing the outcome of the model to an existing published model.
First a very simple mass-spring-damper model has been derived and implemented in Simulink. Then this model has been compared to an existing solution from [29]. In this comparison the same input values, the initial velocity, mass and spring constant, are used. Plotting the graphs of acceleration, velocity and displacement resulted in exactly the same graphs. They are all displayed in Figure 9.15.

Furthermore the approach can be validated with a published approach of simulating car crashes. As described in [29], it is possible to model the crash as a spring-damper problem. For this, the initial velocity needs to be the initial input condition of the first integrator, which is integrating from the

acceleration to velocity. The initial displacement is set to be zero. Also it is only of interest to look at the time frame until the displacement comes back to zero. After that, the 'end' of the crash that can be simulated with the mass-spring-damper models is reached [29].

Furthermore, the final design is compared to similar solutions for which information is available on the internet. In general it is found that most racing drones use a flat carbon fibre frame[24]. However, there are also a few racing drones that use the concept of tubular arms [25] [26] [27]. They are all rated as extraordinarily crash resistant and are about the same size as the AIgle drone, using 5, 6 or 7 inch propellers. Still it can be seen that in all cases the arm tube is about half or 2/3 of the motor diameter. The motors used on these drones have a maximum diameter of less than 3cm, which would bring the diameter of one arm down to 15-20mm. Also their mainframes are made by carbon fibre manufacturing or injection moulding, which allow the frame to have a high strength. Still none of them seem to have any further protection like propeller guards or protective landing gear. This is probably, because the electronic components used in the FPV racing drones are much more robust than the processor that the AIgle drone requires for autonomous navigation and control. In addition the electronic components are not state-of-the-art processors and camera, having a lower cost to replace, thus the design durability is not of primary importance.

Lastly it should be noted, that the structure of the drone can be built and tested in a controlled environment. This is the last step of validation of the structural components. For example, a way of doing is is by building the structure, adding dummy weights for the other components and including an accelerometer at the critical locations. Then drop tests and gate crashes can be executed. The dummy weights and accelerometers are used, not to damage the expensive electronic components when testing.

9.10. Structural Design Results

The final structural design, as explained earlier, can be found in the final design summary in Figure 11.2.

This frame is designed such that the entire drone structure, including its electronics as well as all other components, shall not break in a drop out of 3m, This is be accomplished by having propeller guards and cushioning landing gear, that reduce the impact loads, such that the camera, flight controller and ESC board survive the impact. The propeller guards fit 6" propellers and weigh each 13.8 g. The landing gear is currently weighing 29 grams in total and assumed to have a spring constant of 10.000N/m. The processor, the carrier board as well as the cooler are mounted on spring-damper standoffs. This further reduces the impact accelerations to below 50g by allowing 8mm of cushioning distance.

At this stage the arms, that are carbon fibre tubes, are the only part that is designed to be capable of crashing in a gate at 30 m/s. The current design includes 95mm long tubes as arms with an outer diameter of 2.858cm and a wall thickness of 3.18mm, which is the closest fitting commercially available tube dimension above the calculated requirements. This results in a mass of 37.6 grams per arm, thus 157 grams for four arms in total.

Besides the crash scenarios, an effort is made to investigate natural frequencies of the drone to verify that they do not lie too close to the vibration frequency induced by the motors. From that it could be concluded that the vibration frequency of the motors lies below the natural frequencies of the components, thus amplifying vibrations will not occur.

The motor mounts are designed to be low weight, made out of carbon fibre. This is necessary for the

[24] https://www.banggood.com/nl/search/racing-drone-fpv/8771-0-0-1-1-48-0-price-0-0_p-1.html, (Retrieved June 18, 2020)
[25] https://fpvfrenzy.com/fastest-drones-best-racing-drones/, (Retrieved June 18, 2020)
[26] https://www.nitrorcx.com/87h-kdw280-rtf-24g-black.html, (Retrieved June 18, 2020)
[27] https://www.kickstarter.com/projects/148955951/havocmini-racing-quadcopter-freestyle-drone, (Retrieved June 18, 2020)

large arm diameter, as off-the -shelf components add a high weight to the structure. As the arm mass for the motor mount is already included in the mass of the arms, the mass of one motor mount results in 13.5 grams.

The mainframe and propeller guards are 3D printed out of PEI Ultem filament [28] , which is one of the strongest yet lightest 3D printing filaments available. The cover of the mainframe is shown in the model as transparent, which can be achieved by using transparent filament, such as the HDglass filament of FormFutura [29] . For a stronger design with approximately the same mass, the PEI Ultem filament could also be used for the cover. It is also suggested to leave the color of the top cover filament to be decided upon by each team themselves, such that they can personalise their drone in the race.

The mass of the bottom mainframe and cover sums up to 175 grams and 57 grams respectively. In the design of those components, the aesthetics were taken highly into account. The abstract design looks appealing, which is necessary for its application in international sponsored races. In addition it should to motivate people to build the drone and participate in drone racing, which meets the project objective statement to advance the development in artificial intelligence.
In the cover of the mainframe, slits are included for forward to aft air circulation during forward flight. In addition there are also slits right above the cooler of the processor, allowing to diffuse the processor its heat to the outer environment. The camera is behind the protective top cover, lowering its risk to damage in crash. The battery can be removed via the back of the lower mainframe through another cover.

Lastly, the landing gear is made out of rubber filament, like the Cheetah filament from NinjaTek [30] , in order to be flexible enough to cushion impacts.

The exact materials to be used are specified in detail in section 12.1 in Table 12.1. The camera can be mounted on the dampening system and is placed at an upward angle of 33.5°. This allows the camera to have a horizontal facing when flying at around 10m/s, when the drone is pitching down 33.5°. It was ensured that the drone has a 10°downward field of view, even when the drone is hovering perfectly horizontal. The angles can be taken from Figure 9.16 below. At maximum speed, a forward pitch of 75°is achieved, resulting in an upward viewing angle from the horizontal of 2°.

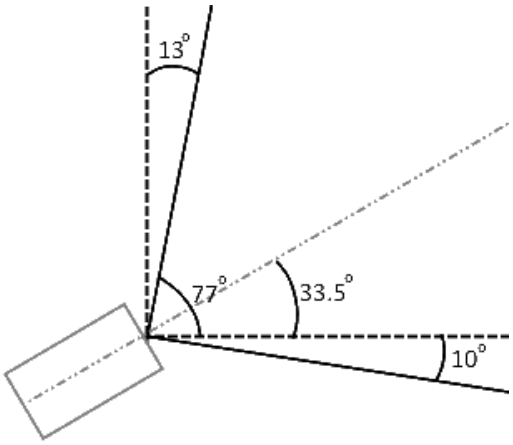


Figure 9.16: Camera installation under an angle of 33.5°and resulting field of view in horizontal hovering

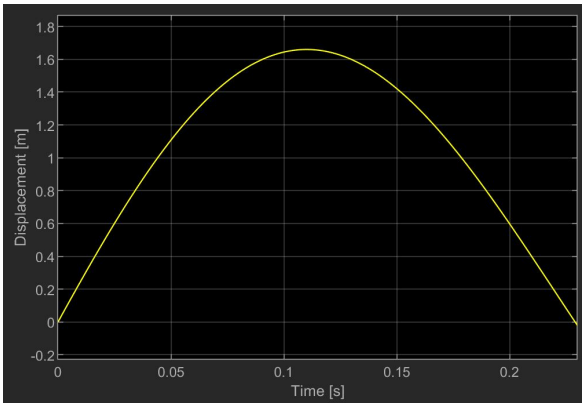
9.11. Future Work

Due to the limited project time and resources, the structural design is not fully optimised yet. This section includes suggestions for future work and improvements.

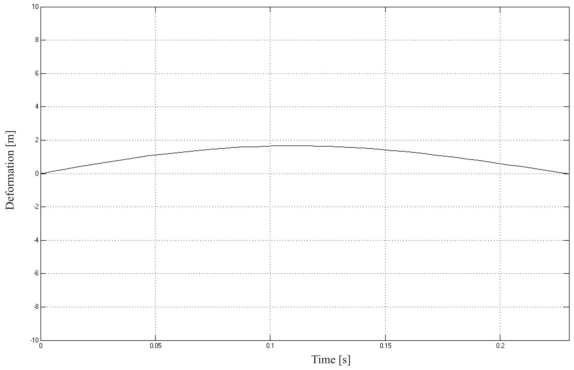
[28]https://www.3d4makers.com/products/pei-filament (Retrieved June 20, 2020)
[29]https://3dprint.com/57124/formfutura-hdglass-filament/ (Retrieved June 20, 2020)
[30]https://ninjatek.com/wp-content/uploads/2018/10/Cheetah-TDS.pdf, (Retrieved June 20, 2020)

First, the drone components that have been designed in CATIA should be tested with finite element software, for example with the build-in function of CATIA itself or in ANSYS. This shall also be used as another tool to verify the lumped mass-spring models. Real life structural tests also need to be performed to validate these results. This will require physically constructing the drone.
A lot of focus has been put on the sizing of the arms for crashes. Arms have uniform shapes and can be analyzed using simplifications like the beam theory. Analysing more complex parts such as the mainframe and its landing gear is challenging. An effort has been made to identify characteristics of the components, such as the spring stiffness or material selection. However, the general structure needs to be analysed in more detail. This analysis might also include physical tests as it is very difficult to accurately predict the behaviour of a structures in a collision.
When doing so, it might turn out that arms with a smaller diameter are sufficient for the 30m/s impact. This would also mean that a reduction in mass would be possible.
Changing the requirement from a 30m/s crash to a 15m/s crash is also a suggestion. This would be in line with designing the electronics and structures to break all at roughly the same loads. This has been analysis in order to estimate by how much it is possible to reduce the arm mass. Using the same approach as used for the arm sizing for 30m/s, it is investigated what arm size would be necessary for a crash with 15m/s. From this follows a required tube thickness of 2.1mm when having an outer tube diameter of 16mm, which is the largest still commonly used diameter for drone arms. This allows for the use of lightweight off-the-shelf motor mounts [31] , which weigh about 18 grams a piece. This motor mount also does not require a longer arm, because it is mounted at the tip and not on top of it, a picture can be seen in Figure 9.13a.This saves at least 3cm of tubing per arm resulting in a required arm length of only 7cm. Different tubes that are available off-the-shelf have been found with the required dimensions [32] [33] that weigh around 0.87 grams/cm and 1.14 grams/cm. Due to slight differences in dimensions and mass descriptions, a safety factor of 2 is applied and thus a mass of 2 grams/cm been used. For four arms of each 7cm length this reduces to a arm mass of 56 grams for all four arms. Together with the slightly more lightweight motor mounts, this design change could allow for a 100 grams reduction in total mass of the structures. This is a high reduction in total mass of the drone. Therefore it can be worth to trade off between a lower mass, or meeting the requirement of sustaining a 30 m/s gate crash.
Furthermore, research on the materials and manufacturing techniques can be done by looking at different manufacturing techniques that would be suitable and easy to implement for most customers. Possible changes can be to use a different, stronger or more lightweight material for the mainframe, requiring carbon-fibre layup that significantly increases the strength. Also the manufacturing can be outsourced to a company, allowing for better manufacturing techniques at higher cost. Therefore the requirement that only 3D printing, 2D milling and standard PCB assembly is allowed, might need to be reconsidered and refined in order to increase the drone its performance, dependent on the customer.
Lastly, 3D printing, which is the current manufacturing technique selected, also enables other optimisation options. For example, the 'fill percentage' of the 3D printed structural components can be increased or decreased, after which the thickness can be varied upon. Lastly, topology optimisation computes what surfaces from the structure do not carry high loads, after which those surfaces can be removed. This is also suggested to investigate in the future design to further reduce the weight of the structure.

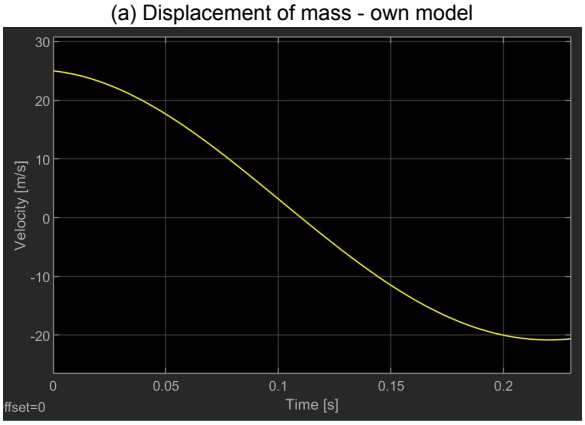
[31] https://www.ebay.com/itm/Motor-Mount-Fixed-Base-Seat-Holder-Bracket-Carbon-Tube-Quadcopter-Multicopter/183770383761?_trkparms=aid%3D1110006%26algo%3DHOMESPLICE.SIM%26ao%3D1%26asc%3D20200520130048%26meid%3Db5d7f2504f19429bac11414a2004bb03%26pid%3D100005%26rk%3D4%26rkt%3D12%26mehot%3Dco%26sd%3D222287715447%26itm%3D183770383761%26pmt%3D1%26noa%3D0%26pg%3D2047675%26algv%3DSimplAMLv5PairwiseWebWithBBEV2bDemotion%26brand%3DUnbranded&_trksid=p2047675.c100005.m1851&autorefresh=true, (Retrieved June 25, 2020)
[32] https://usa.banggood.com/14x16mm-Carbon-Fiber-Tube-For-RC-Drone-FPV-Racing-Multi-Rotor-p-1240981.html?rmmds=search&ID=532657&cur_warehouse=CN, (Retrieved June 25, 2020)
[33] https://www.rockwestcomposites.com/45553-hm, (Retrieved June 25, 2020)



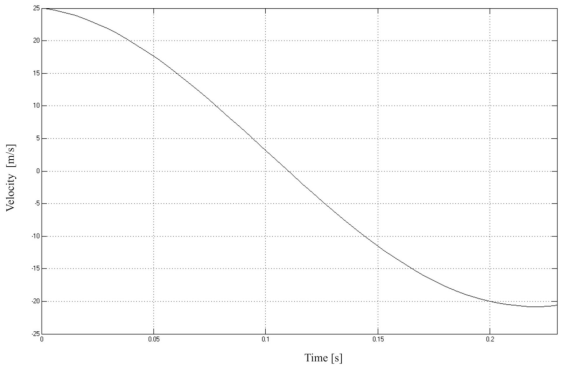
(a) Displacement of mass - own model



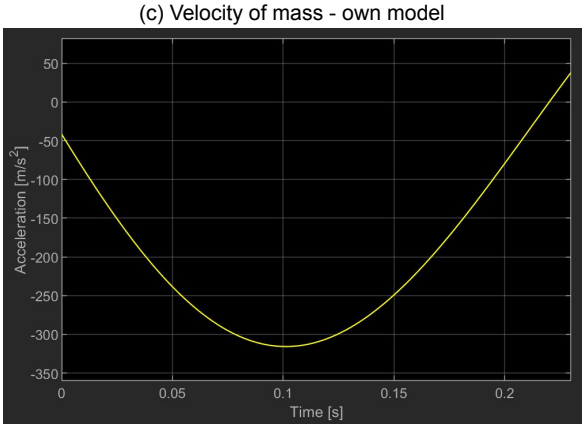
(b) Displacement of mass - validation model



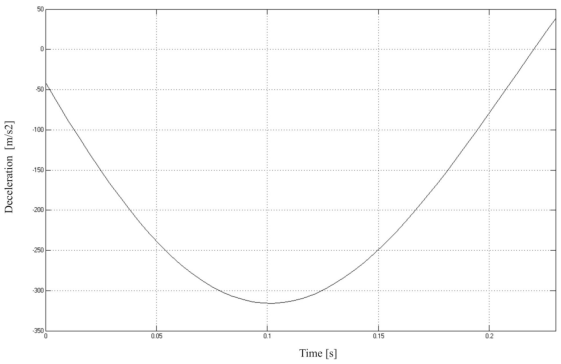
(c) Velocity of mass - own model



(d) Velocity of mass - own model



(e) Acceleration of mass - own model



(f) Acceleration of mass - validation model

Figure 9.15: Comparison of results of own model on the left versus the validation model to the right for acceleration, velocity and displacement of a mass-spring-damper system [29]

10

Software

This chapter describes the software used for the AIgle racer and its development. The main purpose of the software divisions is to demonstrate the capability of autonomous flight and to give a basis which future teams can use to jump-start their own development of autonomous flight software. Section 10.1 first analyse the different requirements directly related to the software. A general overview and pipeline is then presented in section 10.2, which can be used as a foundation for the development of future racing drone autopilot software. A description of the reference frame used is included in section 10.4, and finally demonstrations of the different possible approaches and methodologies are presented in the subsequent sections. This includes Machine Vision & State Estimation in section 10.5, Navigation in section 10.6, and Control and Stability in section 10.7. All the software documented bellow can be found on GitHub [1].

10.1. Requirements

The design of the software subsystem, consisting of machine vision, navigation, and control is based on requirements, which have been setup in the baseline report [**BaselineReport**] and which have been refined and updated in the midterm report [**MidtermReport**]. The requirements related to software are as follows:

- **AIG-TE-HW-MP-MO1**: The drone shall be able to continue flying in the case one motor loses operational functionality during flight.
- **AIG-TE-HW-MP-MO7**: If the drone loses functionality, it shall automatically switch to safety mode within 0.5 seconds.
- **AIG-TE-HW-MP-SM-SDS1**: The system shall be able to shut down within 0.1 seconds of being ordered to shut down.
- **AIG-TE-HW-MP-SM-SDS4**: The system shall turn itself off within <0.5> seconds of detecting a malfunction.
- **AIG-TE-HW-MP-SM-SDS4.1**: If the system detects the batteries contain less than 15% of battery charge, the drone shall land within <10> seconds.
- **AIG-TE-HW-MP-SM-CP2**: The drone shall be switched from autopilot mode to manual control within 0.5 seconds from the order to switch being given.
- **AIG-TE-HW-MP-SM-S-A1**: The acceleration of the drone shall be measured 60 times per second
- **AIG-TE-HW-MP-SM-S-O1**: The angular rate of the drone shall be measured 60 times per second.
- **AIG-TE-HW-C4**: The drone shall use on-board computation only.
- **AIG-TE-HW-C5**: The drone shall use the 'Betaflight' program as attitude controller.
- **AIG-NT-U1**: The drone shall have a 'safety mode', which, when activated, attempts to prevent hazards and incidents.

10.2. Software Overview

With the requirements determined, the first thing to look into before making a detailed design of the software is its structure. It should be known which elements the software has, what they will do exactly and how they will interface. This will be described in section 10.3. Here, the Software Block Diagram will be elaborated upon, showing what a normal implementation is expected to look like. The implementation used to demonstrate the capabilities of the drone follow this design, but does not check data quality/raise flags due to the constraints in time to implement the algorithms.

[1] https://vguillet.github.io/AIgle_Racer/, (Retrieved June 27, 2020)

10.3. Software Block Diagram

The Software Block Diagram, gives an overview of the different parts of the software and how they are related. Here, the blocks show processes, diamonds decisions, and cylinders data storage. Arrows indicate the order of processes (for example, once 'Gather Data' has been performed, 'Reformat Sensor Data' can be done). The software has been split into three parts, Machine Vision & State Estimation (MV/SE), Navigation and Control, each of which has their own parts. Some parts of software fall outside these three groups and are indicated as such. As such, Figure 10.1 provides a template for constructing software intended for drone racing, ensuring safety and limited risk during operation.

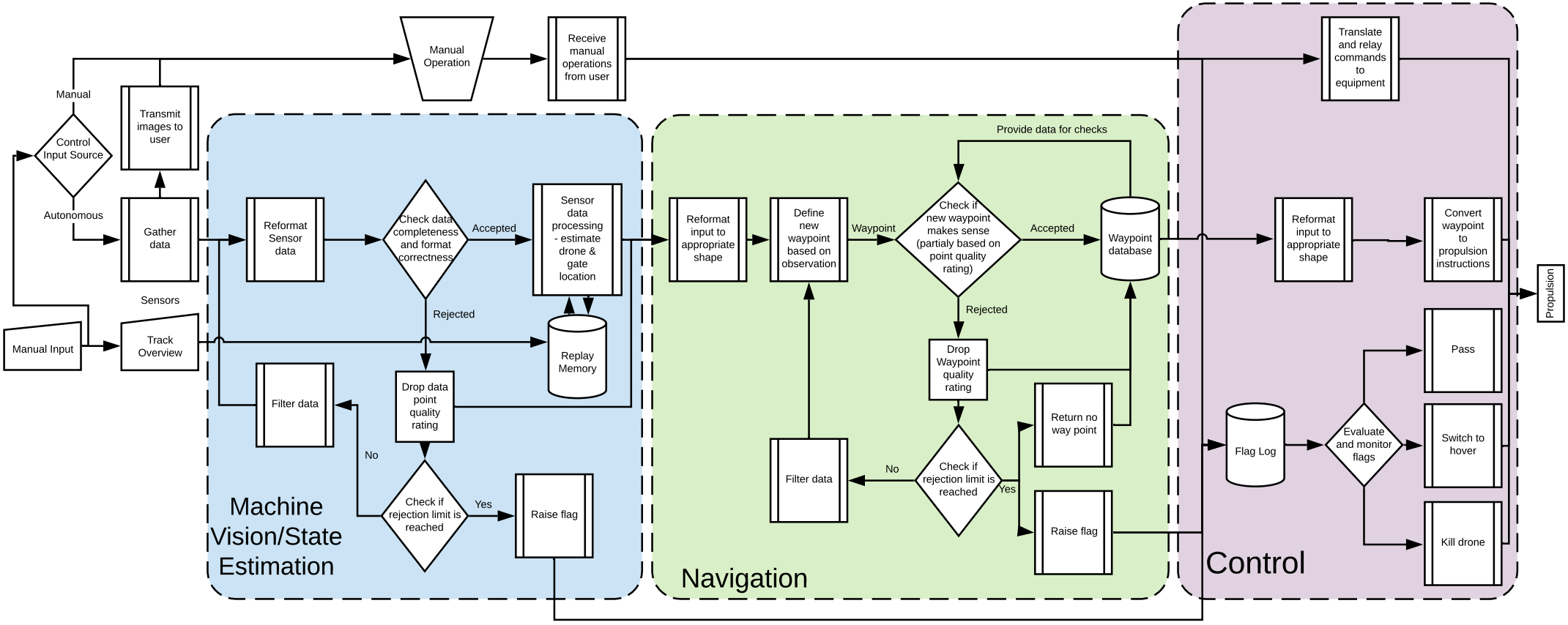


Figure 10.1: The software block diagram depicting the major software components and their relations

10.4. Drone Coordinate Frames

There are several coordinate frames that are necessary to describe the dynamics of quad-copter flight. To begin with, the inertial reference frame has its x-axis point North, y-axis point East and z-axis point towards the ground. The origin can be any point in the vicinity of the drone. This reference frame is used to keep track of the drone's location in the racing space.

The vehicle reference frame v has its axes point in the same directions as the inertial reference frame, with the only exception being the origin that is shifted to the c.g. of the drone. Rotating the vehicle reference frame by a positive drone yaw angle ψ results in the vehicle-1 frame v_1. Subsequent rotation of the vehicle-1 frame by the positive drone pitch angle θ results in the vehicle-2 frame v_2.

Finally, rotating the vehicle-2 frame by the positive drone roll angle ϕ results in the body reference frame with axes x_b, y_b and z_b. These axes are aligned with the drone geometry and they rotate together with the drone. This reference frame is used to measure drone rotational rates and the drone velocity. The rotation matrices necessary to convert between the body and the vehicle frames are given by Equation 10.1 and Equation 10.2.

$$\begin{aligned} R_v^b(\phi,\theta,\psi) &= R_{v2}^b(\phi)R_{v1}^{v2}(\theta)R_v^{v1}(\psi) \\ &= \begin{pmatrix} c\theta c\psi & c\theta s\psi & -s\theta \\ s\phi s\theta c\psi - c\phi s\psi & s\phi s\theta s\psi + c\phi c\psi & s\phi c\theta \\ c\phi s\theta c\psi + s\phi s\psi & c\phi s\theta s\psi - s\phi c\psi & c\phi c\theta \end{pmatrix} \end{aligned} \tag{10.1}$$

$$R_b^v(\phi,\theta,\psi) = R_v^{v1}(-\psi)R_{v1}^{v2}(-\theta)R_{v2}^b(-\phi) \tag{10.2}$$

10.5. Machine Vision & State Estimation

In this chapter, a description is given of the methodology, development and results of the Machine Vision & State Estimation subsystem. subsection 10.5.1 presents a general overview of the various options for machine vision and discusses the selected variants. subsection 10.5.2 continues by describing how the selected method has been implemented, and which results it has yielded.

10.5.1. Approaches

The different approaches to Machine Vision and State Estimation can generally be divided into three categories depending on which sensor(s) are used. Monocular Vision uses a single camera, Stereo Vision uses two or more cameras, and the Other options use sensors other than cameras, such as LiDAR. When considering different options, two important parameters are the weight and power consumption (as a higher power consumption leads to heavier batteries, which equates to more weight). As both single and stereo cameras are capable of state estimation [33][34], a single camera has the obvious advantage due to fewer cameras equating to less weight. Cameras are preferred over other options as they are generally lighter[35] and more efficient[36].

The general approach to determining the location of the gate using a single image is divided into a number of steps. Firstly, the pixels of the gate in the image must be determined. Secondly, this information is used to determine the lines on which points of the gate must lie. Lastly, using known dimensions of the gate, the exact location of the gate can be found.

10.5.2. Implementation

With the general approach known from the previous subsection, an exact implementation must be determined. The first part, detecting the gate, is done by first defining a range of colors, which contains the color of the gate. This allows room for some variation in color due to, for example, lighting. Random pixels of the image are checked until one is found that lies within this range. The pixels above and below it are checked as well, repeating until the upper- and lowermost pixel have been found. Then, pixels to the left and right are checked, repeating until all four corners of the gate are found.

Since the size of the field of view is known, the location of a pixel in the image can be algebraically related to the line that passes through the pixel and the actual corner. The first approach to finding the location of the gate based on this information was using the known dimensions of the gate. Since the line through the camera and the corner is known, the x-, y- and z-coordinates of the corner can be calculated if the distance of the point along this line is known. These relations are known for all four corners. Secondly, since the exact dimensions of the gate are known, the distance between any two points of the gate is also known. As such, a system of equations can be set up for the distance between any two points based on their locations. This system of equations can be solved numerically to find the location. However, a major downside of this implementation emerged quickly; if a system of four equations is used (for example by setting up equations for the four sides of the gate) the solution is not unique. In fact, there is an infinite amount of solutions possible. To solve this, more equations were added using the known diagonal distance, thus yielding a system of six equations. While unique, the solution of this system of equation was found to commonly contain errors, likely due to the equations being overconstrained. Since no solution for this was found, a similar but better documentation was attempted instead.

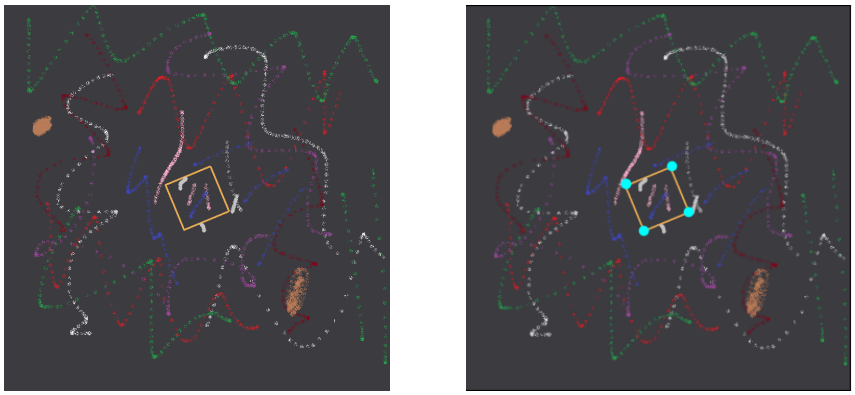
In order to find the gate location with less errors, a Perspective-n-Point algorithm [37] was used. This algorithm aims to find the exact locations of the four corners of the gate. Similar to the previous algorithm, the Perspective-n-Point algorithm also calculates the distances to each point by setting up a system of six equations, as follows:

$$S_{oi}^2 + S_{oj}^2 - 2 * cos(\alpha_{ij}) * S_{oi} * S_{oj} - S_{ij}^2 = 0 \qquad i, j \in 1..4, i \neq j \tag{10.3}$$

Here, S_{ab} is the distance between any two points a and b. The subscript o is used to indicate the location of the camera, i and j are used to indicate any two different corners of the gate. As such, S_{ij} is simply the distance between two corners of the gate, which equals to either the height, width or diagonal of the gate, all of which are known. By solving the system of equations, the distance from the camera to each of the corners can be found. The direction of the distance is also known, as the line along which the point lies has been determined previously. Thus, the relative location of the gate to the camera can be calculated easily.

10.5.3. Results & Future work

The results of this implementation are generally accurate. When testing on a generated image with the real coordinates known exactly, the error is typically within 5%. An example of this is presented in Figure 10.2a. This image shows the input of the MV/SE algorithm. Noise is added to test the robustness of the algorithm. Here, the orange square is used as the gate. To demonstrate that the algorithm correctly identifies the gate, even with noise present, the detected corners of the gate are shown in Figure 10.2b, overlaid on the original image.



(a) The input image for the MV/SE algorithm with noise.

(b) The gate as detected from the image by MV/SE.

Figure 10.2: MV/SE input image and detected gate.

Using this, an estimate of the gate location and orientation is made. This estimate is plotted below, in Figure 10.3. Here, the z-axis indicates distance away from the camera along the direction of view. the y-axis indicates distance above the drone, and the x-axis distance to the right of the drone.

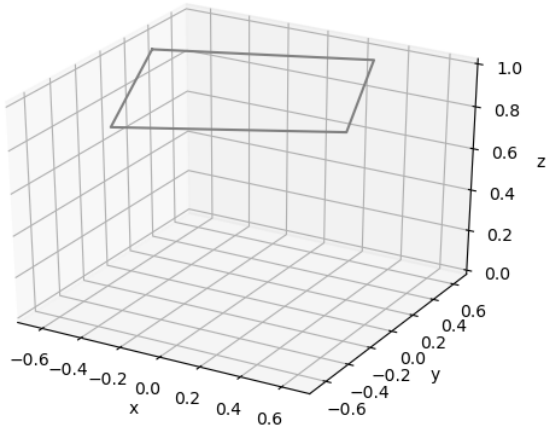


Figure 10.3: The estimate of the gate location

The only case where the typically small error is exceeded is when the camera views the gate from a high side angle, in which case the error becomes significant. The cause for this phenomenon is unclear at this point, and further research into its cause in the future is recommended. Furthermore, the MV/SE software was not designed with its runtime as a primary concern. As such, this is an area that can be improved upon - something that drone racing teams are recommended to do to increase their racing performance.

10.6. Navigation

This section describes the methodology followed in the the development of the navigation demonstration software used by the AIgle racer.

10.6.1. Background

Autonomous drone navigation has been a long time challenge for engineers, and the applications of a robust solution have the potential to revolutionise modern transportation systems. With large firms such as Lockheed Martin sponsoring major events such as the AlphaPilot Innovation Challenge, progress has accelerated in the last few years. The solutions offered were notably impacted with the development of modern AI solutions, which brought about new approaches for solving the problems associated with autonomous navigation. The navigation methods available greatly depend on the data available, and the intended applications. As such, a number of solutions are available, from which a selection will be evaluated in subsection 10.6.2.

10.6.2. Approaches

Various approaches to navigation have been considered, ranging from simple vector based solutions to much more advanced machine learning techniques. Of the different options available, four methods are analysed, and are described in more details in the next subsections. These include:

1. Fly-to-goal navigation
2. QL navigation
3. Double DQL navigation (with prioritised experience replay)
4. DDPG navigation (with prioritised experience replay)

These methods, listed in order of ascending complexity, each present a set of improvement over the previous which is described and investigated in the following subsections.

10.6.3. Fly-To-Goal Navigation

Fly-to-goal navigation is by far the simplest method evaluated here.

Upon receiving the states from the drone, the navigation generates a number of way points on the vector connecting the current drone position and the center of the gate. The trajectory is effectively broken into segments, connecting each gate to the other.

This method, due to its extreme simplicity, presents a number of advantages and disadvantages:

- **Advantages:** The algorithm is extremely fast, acting as a mere ”translator” for the results of the MV/SE. This also suggest that the accuracy of the path created will be mostly dependent on the MV/SE.

  The method is robust. A slight off-course shift would not impact the ability of the drone to reach the gate as long as the MV/SE is able to generate the states correctly
- **Disadvantages:** Due to extreme simplicity, the path and trajectories generated will always remain sub-optimal at best (when compared to trajectories created by a human pilot). The drone is unable to adjust its velocity, turn radius, and angle of approach of the gates. These are all necessary for achieving trajectories comparable to the ones of a human pilot

The advantages and disadvantages listed above suggest that there is a need for a much finer control of the drone, and while being robust enough for simpler pipelines, fly-to-goal navigation lacks the key features which would allow for achieving human level navigation.

10.6.4. QL Navigation

Q-learning (QL) navigation is the first approach involving machine learning. QL is the simplest algorithm of the QL algorithms family. QL is a model-free reinforcement learning algorithm, meaning no model is necessary to be provided to it. It aims at learning a Q-function, which in turns allows the agent (in this case, the AIgle racer drone) to determine what action is best to take given a state.

In Q-learning, the agent aims at solving a temporal difference problem, and learns to maximise expected return when followed in a given environment.

The environment in this case is the race course, in which the drone is able to take steps (our actions) of a given range of magnitude in all directions. This is done following drone flight dynamics.

The states of the drone are then its current position with respect to the gate, its velocity, the gate orientation, and any other information which is useful to determine an optimised trajectory. A number of states are considered here as ”final states”. These are states at which a run is considered as over. This can be a state which implies the goal has been reached (in our case, the drone being inside the gate), or failure states (such as the drone registering a collision with the ground/gate/etc).

The agent is then able to develop a coherent and robust Q-function to navigate an environment through training. In this case, it is the process of optimising behavior through performing a large number of attempts at navigating in the given environment. The agent gathers observations, from the different resulting states and learns from them. Each attempt is called an epoque, and an epoque is made of a defined number of steps.

The agent makes use of a Q-tables (as shown in Figure 10.4), containing every combination of every state and actions possible. For every given state, an evaluation (referred to as Q-value) of the expected return for each action is determined. The Q-table is initialised randomly, and the weights are then adjusted through a balance of (stochastic) exploration and optimisation during training.

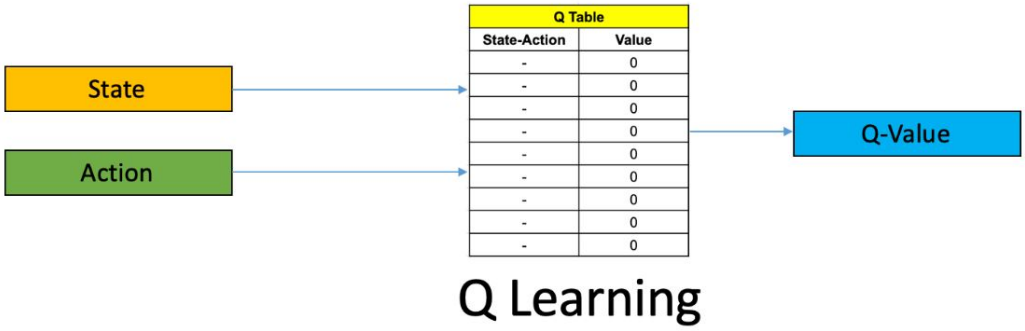


Figure 10.4: A Q-table, containing every combination of state-action possible is gradually adjusted to present the "potential" reward for every possible combination[2].

The actual values of the different possible actions is determined using a Bellman Equation and a reward function. The basic principle behind the Bellman equations is:

"The value of your starting point is the reward you expect to get from being there, plus the value of wherever you land next" (Open AI, Spinning Up, Introduction to RL, Part 2[3])

$$Q^{\text{new}}(s_t, a_t) \leftarrow Q(s_t, a_t) \quad + \quad \alpha \quad \cdot \left(r_t + \gamma \cdot \max_a \; Q(s_{t+1}, a) - Q(s_t, a_t) \right) \tag{10.4}$$

In the above equation, r_t is the reward received upon moving to state s_{t+1} from state s_t, and γ is the learning rate ($0 < \gamma < 1$).

The reward is then used within Equation 10.4 to determine the the Q-values of each action in the Q-table. At each step, a reward is returned based on a weighted average of the old value and the new reward resulting from the action taken.

Two types of rewards can be used. Increasing rewards promote accumulation, as the agent attempts to determine the action which will lead the largest increase in reward over time. Negative rewards on the other hand encourage optimisation, as the agent aims at reaching a final state in the least amount of steps possible to achieve the least negative reward possible. In our case, the reward function varied slightly depending on whether the agent was being trained to reach a gate, or race the entire track, and will be described in more detail in subsection 10.6.7.

The success of the policy is then directly related to amount of training an agent undergoes in a given environment, as having visited every state of the state space is required to take an "intelligent" action at every state.

This approach again presents a number of advantages and disadvantages:

- **Advantages:** The algorithm is able to learn more complex behaviors, allowing for avoiding recurring obstacles, and possibly determining trajectories taking better advantages of the drone flight dynamics and characteristics than a simple straight line.
- **Disadvantages:** The size of the Q-table and the amount of training necessary is directly proportional to the number of states, meaning the algorithm is only viable for limited complex state/actions spaces. An observation containing a large amount of discreet datapoints would lead to an unmanageable table size. The entire table needs to be stored on board, resulting in huge memory requirements for complex environments. The agent is also not able to improvise if it finds itself in a state it has never encountered before.

While already being an improvement over the fly-to-goal method, this approach still presents severe limitations, preventing it from achieving human-level navigation.

[2]https://www.analyticsvidhya.com/blog/2019/04/introduction-deep-q-learning-python/, (Retrieved June 3, 2020)
[3]https://spinningup.openai.com/en/latest/index.html, (Retrieved June 1, 2020)

10.6.5. Double DQL Navigation (with Prioritised Experience Replay)

Double DQL [38] (double deep Q learning) navigation, while still based on the same founding principles as the QL navigation, is the first method here utilising deep learning techniques. Deep learning refers to the use of neural networks in the problem-solving process.

Double DQL navigation is an improved version of the simpler DQL (deep Q learning) algorithm. DQL algorithms follows the exact same structure as QL algorithms, with one major exception. The Q-table is replaced by a neural network, which is trained to estimate the correct Q-value of the different actions given a state, as shown in Figure 10.5.

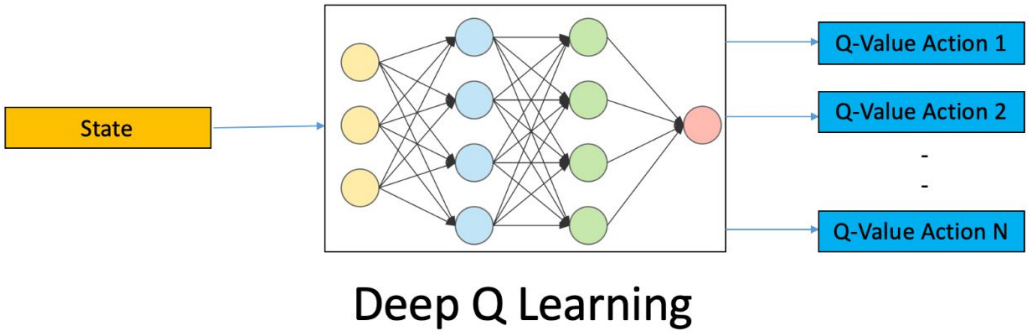


Figure 10.5: A neural network is used to estimate the Q-value of every action based on the input state[4]

This modification, although seemingly small when put in the context of the full algorithm sequence, has massive implications and impacts on the algorithm performance.

- **Advantages:** The neural network replacing the Q-table solves a large number of issues of QL navigation. It first allows for much more complex policies to be learned, taking into account much more state information (such as drone velocities, gate angle etc) and allowing for a much larger action space (the drone would be able to adjust its velocity on top of its displacement).

  The use of a neural network also unlock the ability to improvise. There is no need to visit the entire state space, as the network when trained well is able to develop a general strategy, allowing for a reasonable action to be taken for every state regardless of whether it has been previously seen/visited.

  Neural networks allow for relatively inexpensive processing to be done on board, as once trained, the neural network contains all the information necessary for inference and is the only component which needs to be included on-board. The cost of inference is then solely linked to the complexity of the neural network, and not that of the policy learned.

- **Disadvantages:** Training such algorithms is more difficult, and require more training time. Development is also much more complex, as neural networks do not have a systematic approach in terms of network structure determination and meta-parameters definitions.

DQL also present two extra difficulties:

- **The bias problem:** DQL tends to overestimate rewards in noisy environments, where observations might present inaccuracies. This can result in non-optimal training outcomes[5].
- **The moving target problem:** The high rate at which is updated, combined with the fact that a single network is responsible for both choosing actions and evaluating them leads to further training instability[5].

Those issues are however not without solutions. Double deep Q learning (Double DQL) attempts at addressing these by introducing a second neural network, which is effectively a slower updating version of the original network. In this configuration, the main network is in charge of selecting the next

[4]https://www.analyticsvidhya.com/blog/2019/04/introduction-deep-q-learning-python/, (Retrieved June 3, 2020)
[5]https://pylessons.com/CartPole-DDQN/, (Retrieved June 2, 2020)

best action, and is updated and fitted at every iteration. The target network (the second network) is responsible for calculating the action Q-value to update when training the agent on its replay memory. A target network is necessary as otherwise the main network would be trying to fit to a target value define by itself, which can lead to abrupt, and potentially counterproductive changes. It is initiated with weights equal to that of the main network, and it is updated using either (or both) of these methods:

- **Soft update:** The weights of the target network are gradually updated to match that of the main network using a moving average.
- **Hard update:** Every set number of iteration, the weights of the target networks are set to be equal to that of the main network.

Using soft updates allows for a more gradual convergence towards the desired outcome, while hard updates allow for the target network to catch up to the main network quickly, which has different impacts on the training results.

This double network configuration (visible in Figure 10.6) allows for decoupling the action selection from the Q value prediction, and generally tend to lead to more stable training.

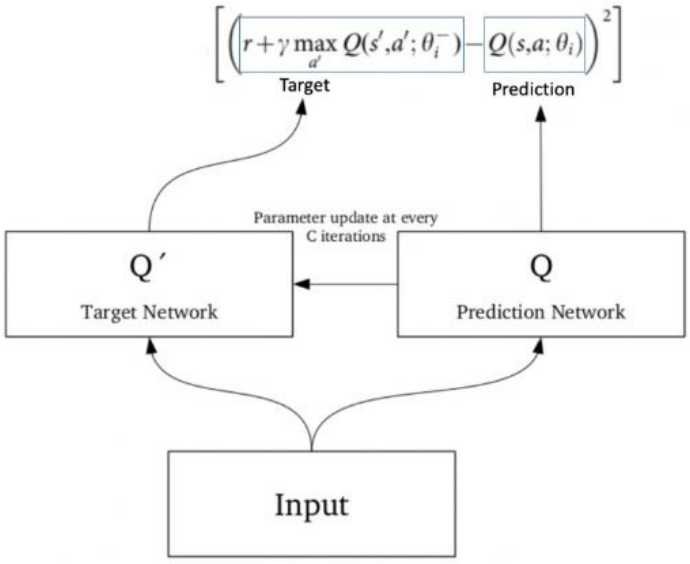


Figure 10.6: A neural network is used for predicting, and one for setting the target[6]

To further enhance training performance, an experience replay memory is introduced for the agent. The memory (of a given maximum size) stores epoques the agents undertake, and includes the observation, the action taken, the resulting state, and the reward. The agent is then able to sample at every step a mini-batch (a given number of experiences) from its memory, and learn from it. The simplest version of this memory samples mini-batches randomly. It should be noted however that not all experiences yield the same value and learning potential. To better employ the replay memory, the concept of prioritized memory replay is introduced. This type of memory puts an emphasis on the more valuable experiences, and aims at further optimising training through targeting efforts on the experiences contained in the memory which the network has the most to learn from. A bias is then introduced in the sampling of the memory, which is proportional to the value of each respective experience. This concept is applicable to every algorithm making use of the experience replay, and as such is used in Double DQL and DDPG (described in the next subsection).

The final resulting navigation protocol, Double DQL navigation with prioritised experience replay, provides a much more powerful and optimised solution for drone navigation. The use of deep learning opens the gate to vastly more complex strategies, and is able to navigate much more efficiently. This also enables for significantly more information to be provided to the drone, which would allow for more

[6]https://www.analyticsvidhya.com/blog/2019/04/introduction-deep-q-learning-python/, (Retrieved June 3, 2020)

complex, effective, and optimised trajectories to be developed.

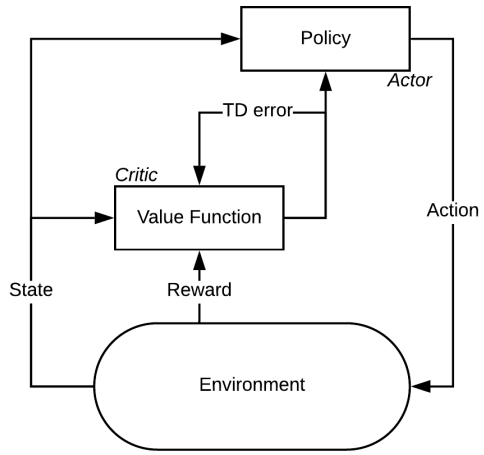
This method however still presents a major flaw. Double DQL works in discrete environments, and while it is able to learn behaviors in a discretised representation of a continuous environment, its is still significantly constrained and limited by it.

10.6.6. DDPG Navigation (with Prioritised Experience Replay)

DDPG (double deterministic policy gradient) is the last navigation method that is investigated here. This method is an interpolation between Q-learning and policy optimisation methods, presenting two major advantages. It is able to retain Q-learning based methods' substantial efficiency, while also taking advantage of policy-based optimisations techniques (they are principled, as optimisation is directly performed with a specific goal in mind). This careful trade-off between the two allows for a more stable and targeted training. Secondly, it enables developing and learning policies in a continuous environment, which is preferred in the drone racing case. DDPG is based on the actor-critic architecture.

"Deep Deterministic Policy Gradient (DDPG) is an algorithm which concurrently learns a Q-function and a policy. It uses off-policy data and the Bellman equation to learn the Q-function, and uses the Q-function to learn the policy." (Open AI, Spinning Up, Algorithms Doc, Deep Deterministic Policy Gradient[7])

Note: Off-policy means that data collected at any point during training can be used at each update, regardless of the agent's exploration choices



As seen in figure Figure 10.7, one of the networks, the actor, is tasked to select the next action to be taken, and the critic network is tasked with evaluating the quality of the action selected given the observation. The rest of the algorithm follows the same sequence as Double DQL and QL, and also involves a training phase with an agent balancing exploration and optimisation in a given environment.

Figure 10.7: Two networks handling different portions of the problem

This algorithm was demonstrated as being particularly apt at solving complex continuous problems such as physics based challenges and car driving, which happens to be very similar tasks in principle to the racing drone case.[39].

DDPG's hybrid properties makes it well suited for such applications, and can be combined with most of the techniques described earlier in the Double DQL description. The inherent problems resulting from the use of neural networks described earlier (bias and moving target) are also present and can be addressed similarly using the main-target networks system for both the actor and the critic (enabling a more stable training). The prioritised experience replay method is also applicable here for the same reasons. Overall, DDPG demonstrates significant potential for drone racing applications:

- **Advantages:** DDPG further improves over Double DQL by allowing for training and learning continuous policies, unlocking truly optimised policies and making it particularly well suited for the task at hand.

[7] https://spinningup.openai.com/en/latest/algorithms/ddpg.html, (Retrieved June 1, 2020)

The hybrid nature of the algorithm allows it to reap the benefit of both Q-learning and policy optimisations methods

The method is relatively computationally inexpensive to run once training has been completed, and requires limited on-board processing power.

- **Disadvantages:** The DDPG algorithm is even more complex than Double DQL, and its implementation can prove to be a challenge.

  DDPG training involves updating four neural networks simultaneously, which requires significant processing power to be trained fast, making learning very complex policies a resources intensive process.

Overall, DDPG presents the most potential of all the methods cited so far. A well balanced network architecture combined with fine tuned meta-parameters (the parameters which dictates various properties of the training process in machine learning, such as learning rate etc.) and the right observations could enable very complex and effective trajectory planning and execution.

10.6.7. Implementation

The environment used for testing is Airsim [40], a testing environment developed by Microsoft. Airsim allows for controlling a virtual Drone in real and accelerated time in a realistic setting with realistic flight mechanics. Airsim consists of a plugin and an API which allows for programmatically controlling a model in the Unreal game Engine (UE). The drone is as such flown in a realistic environment and according to realistic flights mechanics. One of the other major advantages of using a game engine such as Unreal Engine is that it allows for example to include animated character models running around the track, ensuring algorithm resilience to non-static backgrounds in a race setting. This allows for custom environments to be created rapidly to effectively represent situations the drone will find itself in. For this demonstration, a simple circular race track consisting of 6 different gates at various angles was created to be used as the training ground for the agent, and can be seen in Figure 10.8.

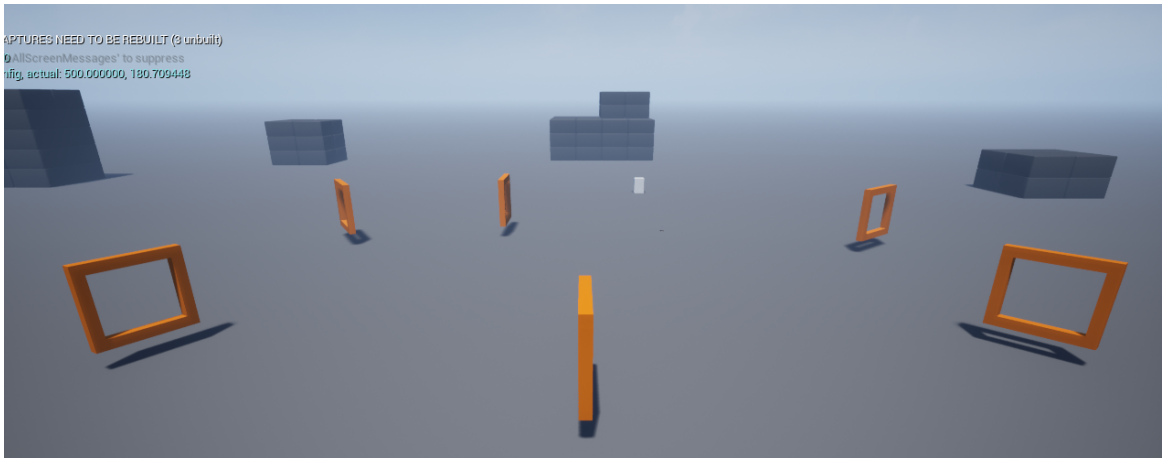


Figure 10.8: The racing track used in this demonstration, created in Unreal Engine

To achieve reasonable training times, the clock speed of the simulation was set to x250. This maintained a good simulation accuracy, despite occasionally failing to detect collisions.

Of the previously-cited navigation approaches, two are investigated in details and developed as proofs of concepts; Double DQL and DDPG. These where chosen as upon evaluation of their advantages and disadvantages, they where the most promising candidates. As explained earlier, a large portion of the algorithm remains the same, and as such is kept consistent in the two cases.

The observations work as follow. At each step, the agent is provided with a vector of its position relative to the center of the next gate. The observation is then a vector [x, y, z]. The decision was made to not provide more data in the observation such as velocity vectors [u, v, w], as more data

would make the training more difficult, long, and complex. The lack of velocity information would however result in it not being able to optimise its trajectory accordingly, and not take full advantage of drone flight mechanics. Providing position information is nonetheless enough to demonstrate the ability of the method to effectively learn to navigate the environment.

The actions are set as follow. At every steps, the agent can perform a move composed of 0 to 3 units of displacement in x, y, and z directions respectively (the smallest step would be of (0, 0, 0) and the largest (3, 3, 3)). Although it is also possible for the agent to be provided with a range of velocity options at every step, the velocity here was kept constant. This was done so as the lack of velocity observation meant that the agent's best option would be to always select the highest speed, as it wouldn't be able to take advantage of varying it.

The reward is set as follow. The reward on any non-final states is the negative of the vector magnitude between the drone and the gate center. Reaching a failure final state (registering a collision in our case) results in a penalty of -200. Reaching the center of the gate leads to a bonus of +300. To limit the working range and further speed up training, a collision is considered as detected if the agent goes above or bellow a certain altitude. It should be noted that the gates here are created in Unreal Engine, and as such do not generate signals when crossed. As such, a gate is considered as crossed when the agent got within a set distance from the coordinates corresponding to its center.

The agent is allowed to take a set number of steps, which is referred to as its age. Reaching the maximum step count is considered as a failure final state in itself.

A number of meta-parameters and settings are available to control training progress and evolution. These include:

For the **Double DQL:**

- **Epoques count:** The length/number of attempts made during training
- **Memory size:** The maximum number of experiences recorded in the replay memory of the agent
- **Minimum replay memory size:** The minimum number of experience necessary to use replay memory for training
- **Minibatch size:** The size of the sampled minibatch from the replay memory when training
- **Training rate:** The rate at which the network "learns"
- γ**:** Discount, the parameter defining importance given to short-term/long-term reward
- τ**:** The rate at which the target network is updated
- **Hard update every:** The epoque rate at which the target network is hard updated
- ϵ**:** The probability of the agent taking a random step as opposed to an optimal one
- τ **decay:** The τ decay mode
- γ **decay:** The γ decay mode
- ϵ **decay:** The ϵ decay mode

  Four decay modes are available: "Fixed value", "Linear decay", "Exponential decay", "Logarithmic decay". The learning rate could have also been throttled, but this was not attempted here.

For the **DDPG**, the available hyper-parameters and settings are the same as for the **Double DQL**, with a few exceptions:

- **Actor learning rate:** The rate at which the actor network "learns"
- **Critic learning rate:** The rate at which the critic network "learns"

Finally, both algorithms can make use of different structures of neural network, which was also investigated.

10.6.8. Results

Different tasks of gradually increasing complexity were trained in order to demonstrate the effectiveness of the methods. These include the following tasks (the algorithm used is mentioned first):

- Double DQL: Fixed starting point, fixed gate goal
- Double DQL: Randomised starting point, fixed gate goal
- Double DQL: Randomised starting point, randomised track goal
- DDPG: General navigation

Double DQL: Fixed starting point, fixed gate goal

The first challenge is simply attempting to teach the drone to fly to a fixed gate from a fixed starting point. Reaching the gate is considered as a final state in this case. The goal of this is to investigate whether the agent can derive a simple policy and learn to fly in the direction of a fixed goal, while avoiding going too high or too low (which are considered final states as doing so results in a collision with the invisible bounds included in the reward function). Each individual purple line represents an epoque of training (the trajectory followed by the drone per attempt). The network structure used here was as follow:

Input (Observation shape) →Dense (512), relu →Dense (256), relu →Dense (64), relu →Output (Action space shape), linear

Note: To understand the above notations, it reads an input in the shape of the observation, followed by a hidden dense layer of 512 neurons with a relu activation function. a hidden dense layer of 256 neurons (with relu), a hidden layer of 64 nodes (with relu) and finally an output layer the size of the action space with a linear activation function. "Dense" refers to how the neurons are connected in the network, a dense layer is a layer fully connected, so every neuron of the layer is connected to every neuron of the previous and next layers. The activation functions are what is used to obtain the output of a neuron. Relu stands for Rectified Linear Unit.

Figure 10.13 demonstrate the results of the training.

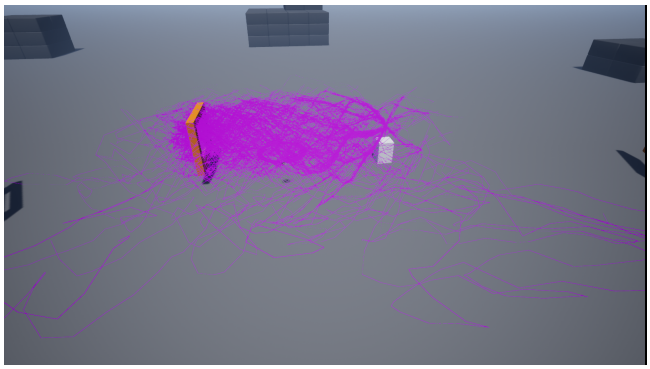


Figure 10.9: Double DQL: Fixed starting point, fixed gate goal. Render of drone path per epoque

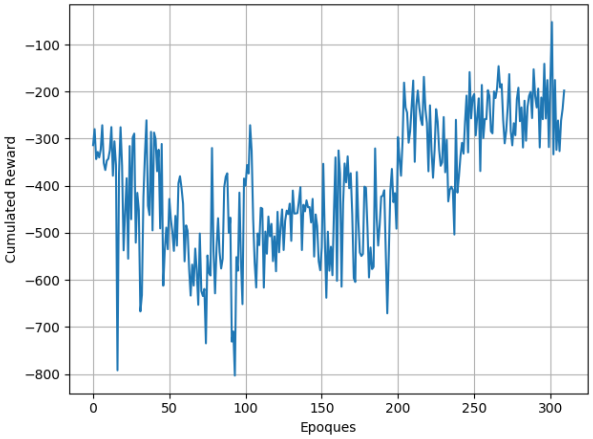


Figure 10.10: Reward per epoque

In the above figure, the agent was trained for 300 epoques. During training, its was observed that the agent converges quickly after an initial dip. This dip is however necessary to properly internalise the environment and reward rules. Once internalised the reward climbs quickly, to reach a maximum of less than -100. This experiment demonstrate that this goal is very achievable by the agent, and quickly converges to a solution.

Double DQL: Randomised starting point, fixed gate goal

The next task is similar to the one described above, with the exception that the starting point is now randomised. The goal gate remains constant while the starting point is placed randomly at the beginning of every epoque. The neural network structure was kept the same as the one previously described. The epoques are bundled in batches in the diagram to improve clarity, and trend lines are added for both the average batch reward, and the best individual reward per batch to better monitor learning progress.

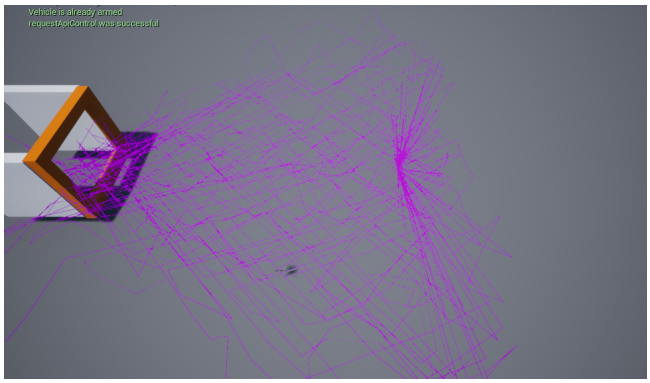


Figure 10.11: Double DQL: Randomised starting point, fixed gate goal. Render of drone path per epoque for the final batch

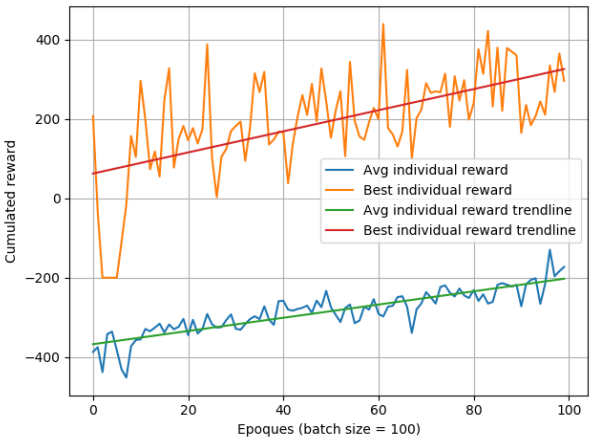


Figure 10.12: Reward per epoque

Yet again, the agent is able to effectively solve the environment and reward function, converging to a solution over time. It should be noted that because of the way the ”crossed the gate” signal is generated (being at a certain distance from the center coordinate), when starting at very steep angles, the agent would opt for going around the gate instead of through. The white side-plates were thus added to prevent learning such behavior. This solution is acceptable as the gathered observations is provided by the simulation, and is not impacted by the content of the environment in this configuration.

Double DQL: Randomised starting point, randomised track goal

The final task is getting the agent to go around the whole track. The starting point is randomised, and the goal and age is reset every time the agent made it to its goal gate. To further encourage generalisation, the direction of the track is switched randomly, which combined with the random starting point presents the agent with more varied scenarios. To obtain good results, the network structure had to be heavily modified. This is done so because signs of overfitting (the network learning to solve this specific problem as opposed to this ”type” of problem) became apparent during testing. The final network structure was as follows:

Input (Observation shape) →Dense (8), relu →Dense (16), relu →Dense (24), relu →Dense (32), relu →Output (Action space shape), linear

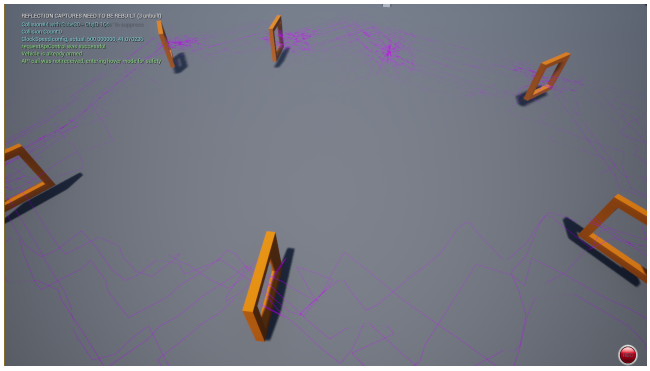


Figure 10.13: Double DQL: Randomised starting point, randomised track goal. Render of drone path per epoque for the final batch

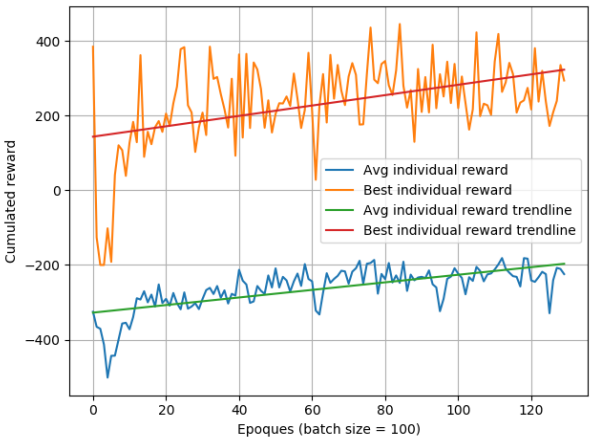


Figure 10.14: Reward per epoque

This task proved to be the most difficult to train for. This is mainly due to two factors. First, the training time significantly went up. Resetting the age of the agent every time it crosses a gate means the run time per epoque is multiplied by the number of gates crossed every time. The variety of situations and observations the agent find itself in also posses its own challenge in itself. Deriving a general policy proves particularly challenging using this setup as the agent is much more exposed to certain

situations than others. It should be noted that although the starting location with respect to the first goal was set randomly, the starting gate interval was kept constant, with the agent always starting in the 1/6 gate interval. As a result, gates 1 and 6 in Figure 10.15, are seen at the start of every epoque, while later situations, such as gate 5 and 6, require the agent first making it through gates 1/2/3/4, which is much less frequent.

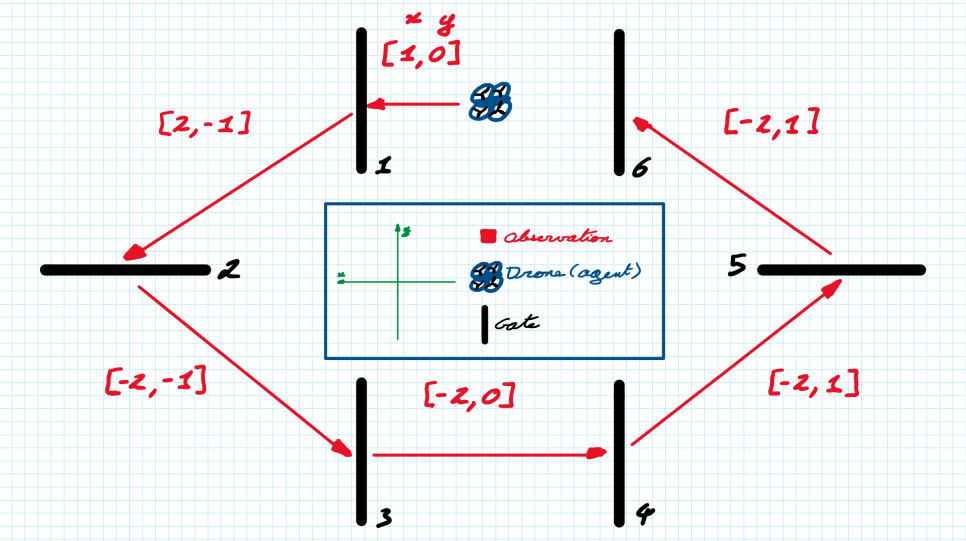


Figure 10.15: Schematic representation of a track (does not match the track used here exactly)

As such, behavior in specific situations where reinforced disproportionately with respect to others, resulting in an imbalanced training. Consequently, deriving a general policy could benefit greatly from a different training setup. This could be achieved easily by randomly picking the starting gate interval, and training the drone on randomised circuit layouts to further encourage generalisation of learned policies.

Despite all the above mentioned issues, the agent was still able to determine effective policies, gradually improving its performance to the point of making it through a large number of gates consistently, and occasionally making it around the track completely.

DDPG

Upon completion, the DDPG algorithm made a main issue apparent for this project. The processing power required to train four neural networks simultaneously made training for large epoques counts unrealistic. The limited processing power available meant that it was not possible to test the approach on the tasks described above in the timeframe of the DSE.

10.6.9. Future Work

A large number of improvements are now possible to further the performance and accuracy of the system.

Further training and optimisation of the current algorithms is a first and important step. Using and/or finding more optimised meta-parameters and/or network structures (through systematic search or using optimisation techniques such as genetic algorithms) could have a significant impact on performance.

Providing the agent with more data in its observations would also enable to better understand it. SLAM techniques, which are methods used to map out a 3 dimensional environment on-the-go, could provide the drone with a full observation of it's surrounding for example. This however needs to be investigated as more data does not always lead to performance improvement.

Other algorithms variations could also be considered. DDPG for example tends to be very dependent on hyper-parameters, and also presents a tendency to overestimate Q values at times. Twin Delayed

DDPG (TD3) is one of the possible solutions, building up on DDPG to create a likely more robust and effective solution for navigation.

To further enhance these skills, imitation or apprenticeship learning could be considered to provide example navigation for the agent to learn and build up from. This could in turn allow for shorter training times and for human-based navigation policies to be developed.

To further increase performance and decrease the software pipeline size, images could be used directly as input to the networks. The network would effectively take over the role of MV/SE and learn to fly through gates on the basis of it camera view and images collected. This would allow for even less software components on-board. Airsim allows for sampling first person images directly from the simulation. This allows for a realistic data sampling and observation to be collected, which would allow for this type of training to be put in place easily.

Flying based on the input images could in turn allow the agent to learn to fly a race with competition, and know how to react to avoid crashing while attempting to outmaneuver an opponent.

Another possible improvement would be to include Long Short Term Memory Layers (LSTM) layers to the network structure, to enable the agent to develop temporal based policies. LSTM layers are a type of recurrent neural layer, allowing for temporal representations to be developed. This would effectively enable the agent to take into account its previous states in its evaluation of the state it finds itself in and its decision on what action to take next.

Transfer learning could also be used to apply the learned policies to other purposes. Transfer learning involves taking part of a trained network to enhance and transfer learning to a different application.

This and all the above combined could allow for a very wide range of autonomous drone applications, such as commercial package delivery, high speed emergency response, autonomous search and rescue, and autonomous military fighters (the navigation system could in theory be taught dog-fighting with the same methods just as effectively as it could learn to race).

10.7. Control and Stability

The control and stability software generates commands that are sent to the flight controller board, which then sends commands to the ESC to control the motors and through that, the motion characteristics of the drone. The commands are based on the trajectory determined by the Navigation software and on the state information measured by the on-board instruments. These commands are fed to the firmware that does the low level control. However, as per requirement, the drone can also be controlled manually using the receiver that is connected to the flight controller. This chapter describes how the control software is structured and what results it generates. First, the general quad-copter dynamics and principles of motion are explained in subsection 10.7.1. Then, the control software inputs, outputs, and the general program structure are described in subsection 10.7.2. This is followed by a detailed explanation of the algorithm used to generate the required commands, as described in subsection 10.7.3. In order to test the program and tune the algorithm, a simulation tool is developed, as described in subsection 10.7.4. Then, the gains are tuned by analysing the system response to a step input in subsection 10.7.5. The resulting control system performance demonstrated with the tool is presented in subsection 10.7.7, where various verification tests and stability analyses are discussed. The section is concluded with a reflection on the system results and recommended future work in subsection 10.7.8.

10.7.1. Drone Flight Dynamics

A quad-copter is an underactuated system, which means that there are more degrees of freedom (3 translations and 3 rotations) than actuators (4 rotors). This means that movements in certain degrees of freedom are coupled. The body forces of a drone are shown in Figure 10.16.

Opposite rotors rotate in the same direction, such that there is a moment equilibrium in all directions. As thrust is always perpendicular to the x-y plane of the drone, flight direction is changed by changing the attitude of the body. In general, higher speeds are achieved at higher (negative) pitch angles,

which can be reached by increasing T_3 and T_4, and decreasing T_1 and T_2, for negative $\Delta\theta$. A rolling motion, especially useful to counteract the centrifugal force during turning, is achieved by increasing T_1 and T_3, and decreasing T_2 and T_4, for positive $\Delta\phi$. To adjust heading, a yawing motion can be used, which is achieved by increasing T_1 and T_4, and decreasing T_2 and T_3, for positive $\Delta\psi$. This is possible thanks to rotating propellers exerting a moment that is counteracted by the body, which means a propeller rotating in the positive direction contributes to the body rotation in the negative direction and vice versa. If the net value of this moment is non-zero, the drone yaws.

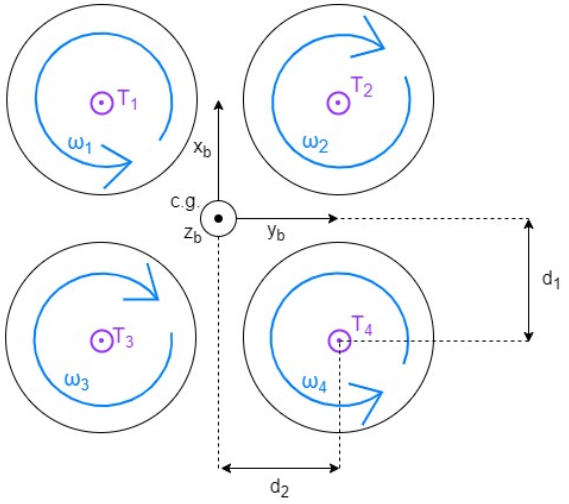


Figure 10.16: Top view of a quad-copter with the thrust forces and the coordinate system.

It is important to understand the principles of drone dynamics, however, the algorithm used in the control software is not explicitly based on them, which is why the quad-copter equations of motion are not given here. The algorithm is based on computing the total (not per motor) thrust force vector, which is sufficient to generate the commands necessary for the flight controller firmware.

10.7.2. System Architecture

The Control software is built to be compatible with Betaflight, which is a commonly used open-source drone firmware. It is a state of the art flight controller firmware for quad-copters, which is typically used for FPV drones controlled by a human pilot. It can also be used on autonomous drones if combined with a control algorithm that generates commands that the human pilot would otherwise make.

The Betaflight firmware runs on the flight controller board, where it computes the required thrust per motor and sends appropriate commands to the ESC. In order to do so, it requires input in the form of a binary signal that contains the magnitude of the desired total thrust and the desired angular rates about the three body axes:

$$\mathbf{u}_{\text{des}} = [\|\mathbf{F}_{\text{des}}\|,\ \omega_{\text{B,x,des}},\ \omega_{\text{B,y,des}},\ \omega_{\text{B,z,des}}]^T. \tag{10.5}$$

The control software is thus made to compute this information at a high frequency. The state information necessary for this is received from a range of sources. First of all, the Navigation system provides the desired trajectory information in the form of a vector that contains the coordinates of the next waypoint $\mathbf{r}_{\text{des}}$ and the desired velocity magnitude v_{des} along the flight direction:

$$\mathbf{x}_{\text{nav}} = [\mathbf{r}_{\text{des}},\ v_{\text{des}}]^T, \quad \text{where} \quad \mathbf{r} = [x\ , y\ , z]^T. \tag{10.6}$$

Secondly, the Machine Vision system provides information about the current position $\mathbf{r}$, attitude angles ϕ, θ and ψ, and velocity $\dot{\mathbf{r}}$. Finally, the IMU measures the current angular rates in body coordinate system $\omega_{\text{B,x,des}}$, $\omega_{\text{B,y,des}}$ and $\omega_{\text{B,z,des}}$, and body acceleration $\ddot{\mathbf{r}}$. With these inputs, the control system performs calculations based on the algorithm by D. Mellinger [41], as described in subsection 10.7.3, and outputs the signal to Betaflight.

The control system structureis visualised in Figure 10.17. The control software is built as a system of nested loops, those being a position, a velocity and an attitude loop. The position loop takes the desired position and the velocity magnitude as inputs from the navigation system and combines these with the current position to output a desired velocity vector. The velocity loop uses this desired velocity vector, in combination with the current velocity, to output a desired acceleration vector. It also calculates the desired throttle, which is outputted directly to Betaflight. The attitude loop takes the desired acceleration vector, as well as the current body rotation, current body accelerations and the current body rotation rate. It combines these to calculate the desired angular rates, which it outputs to Betaflight. Betaflight uses the desired angular rates and desired throttle to determine the desired RPM per motor, which is then outputted to the ESC, which sends it to the motors. This will affect drone kinematics, which will be measured by the IMU and the Machine Vision system, which will send their data back to all three aforementioned loops, thus completing the system.

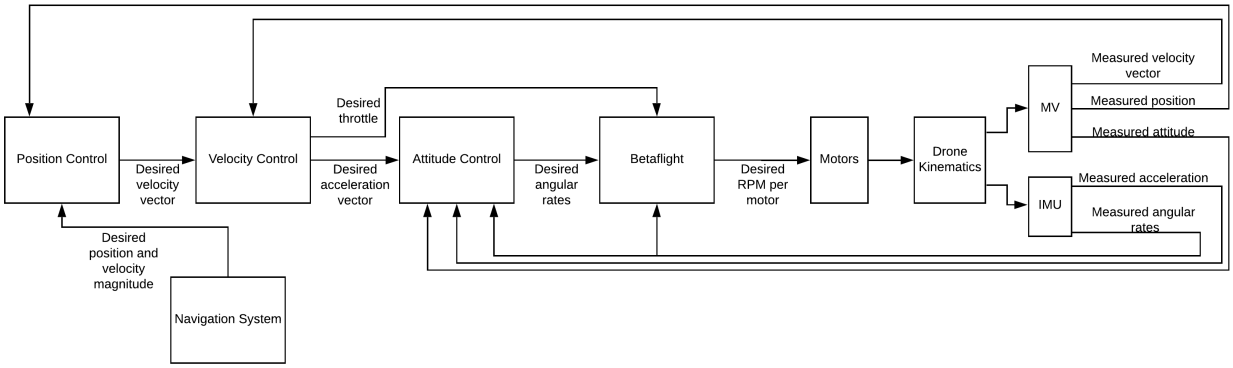


Figure 10.17: Control system structure

At each level of the Control system, PID controllers are used to compute desired linear and angular speeds. PID stands for Proportional-Integral-Derivative, and it is a commonly used feedback mechanism. The working principle of a PID controller is visualised in Figure 10.18.

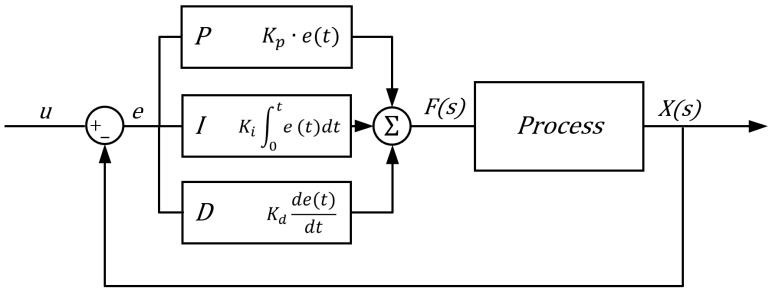


Figure 10.18: Block diagram of a PID controller [8].

In the picture, u is the desired state, from which the measured state is subtracted to obtain the error e. This value is then processed in parallel using P, D and I gains, and then all outcomes are summed up to compute $F(s)$, which can be further processed before outputting the signal $X(s)$. The gains have to be carefully tuned, which can be done only after understanding the principles behind using the P, I and D controllers, as they all have different influence on the system behaviour and stability.

The P - controller provides output that is proportional to the error. Without the I - and D - controllers, it can provide a stable operation but it always maintains a steady state error. The response speed increases with increasing the P gain. In order to remove the steady state error, the I - controller is used, which integrates the error over time. The lower the I gain, the lower the steady state error

[8] https://university.listenlights.com/wp-content/uploads/2017/11/cover-page-3.png, (Retrieved June 20, 2020)

but the response time increases. This can be solved by using a D - controller, which anticipates the system behavior by computing the rate of change of the error. This allows to increase the response time by increasing the D gain.

10.7.3. Control Algorithm

The calculations in the control software are based on the algorithm proposed by D. Mellinger [41]. This algorithm provides most of the necessary equations, with some required equations missing, which were thus established by the team. The method presented in this section is thus a combination of the Mellinger algorithm and assumptions made by the team.

To describe the drone attitude in the world reference frame, the Mellinger algorithm uses rotation matrices that are based on the body coordinate system:

$$^{W}R_{B} = [\mathbf{x_B}\ \mathbf{y_B}\ \mathbf{z_B}]. \tag{10.7}$$

The vectors of the body axes are computed such that the z-axis is aligned with the force exerted by the motors. The current thrust force is calculated using

$$\mathbf{F} = m\ddot{\mathbf{r}} - mg\mathbf{z_W} - \mathbf{D}, \tag{10.8}$$

where $\ddot{r}$ is the acceleration measured by the IMU, and D is the drag force calculated as described in section 7.2. In order to determine the desired thrust force, a PID controller is used. The controller uses multiple gains, where each is a diagonal matrix, which allows to tune the motion independently about each axis. First, the position and velocity errors are computed according to

$$\mathbf{e_P} = \mathbf{r} - \mathbf{r}_{\text{des}},\ \ \mathbf{e_V} = \dot{\mathbf{r}} - v \cdot \frac{\mathbf{e_P}}{\|\mathbf{e_P}\|}, \tag{10.9}$$

where r and $\dot{r}$ are current position and velocity, while r_{des} is the desired waypoint and v is the desired velocity magnitude, as indicated by the Navigation system. Using these errors, the desired acceleration is calculated with

$$\ddot{\mathbf{r}}_{\text{des}} = K_a \mathbf{e_V}, \tag{10.10}$$

where K_a is the derivative gain. Finally, using the errors, together with the gravity and drag force, the desired thrust force is obtained with

$$\mathbf{F}_{\text{des}} = -K_P \mathbf{e_P} - K_V \mathbf{e_V} - mg\mathbf{z_W} + m\ddot{\mathbf{r}}_{\text{des}} - \mathbf{D}, \tag{10.11}$$

where K_V is the proportional gain and K_P is the integral gain. With this, the z-axis of the body coordinate system is calculated using

$$\mathbf{z_B} = \frac{\mathbf{F}_{\text{des}}}{\|\mathbf{F}_{\text{des}}\|}. \tag{10.12}$$

Then, based on the desired yaw angle ψ_{des}, calculated using simple geometry, the heading vector is computed with

$$\mathbf{x_C} = [cos\psi_{\text{des}},\ sin\psi_{\text{des}},\ 0]^T. \tag{10.13}$$

Now, the remaining body axes can be calculated using

$$\mathbf{y_B} = \frac{\mathbf{z_B} \times \mathbf{x_C}}{\|\mathbf{z_B} \times \mathbf{x_C}\|},\ \ \mathbf{x_B} = \mathbf{y_B} \times \mathbf{z_B}. \tag{10.14}$$

The rotation matrices based on the current and the desired thrust force are now used to compute the attitude error according to

$$\mathbf{e_R} = \frac{1}{2}\left(R_{\text{des}}^T R - R^T R_{\text{des}}\right)^{\vee}, \tag{10.15}$$

where $^{\vee}$ represents the mapping of $so(3)$ elements to $\mathbb{R}^3$. Subsequently, the angular rate error is calculated with

$$\mathbf{e}_{\boldsymbol{\omega}} = \boldsymbol{\omega}_{\mathbf{B}} - \boldsymbol{\omega}_{\mathbf{B},\text{des}}, \quad \boldsymbol{\omega}_{\mathbf{B},\text{des}} = K_{RR}\mathbf{e_R}, \tag{10.16}$$

where K_{RR} is the rotational rate gain, which is a proportional gain that allows to determine the required rotational rate based on the attitude error. Finally, all the necessary values are computed and the elements of the output vector u_{des} are obtained with

$$u_{1,\text{des}} = \mathbf{F}_{\text{des}} \cdot \mathbf{z_B}, \quad \begin{bmatrix} u_{2,\text{des, moment}} \\ u_{3,\text{des, moment}} \\ u_{4,\text{des, moment}} \end{bmatrix} = -K_R\mathbf{e_R} - K_\omega\mathbf{e}_{\boldsymbol{\omega}}, \tag{10.17}$$

where K_R is the rotation matrix gain and K_ω is the angular rate gain. The first output element is $u_{1,\text{des}}$, which is the total thrust force in N. The remaining output elements are currently the moments about three body axes in Nm, and not angular rates in $\frac{rad}{s}$, like required for Betaflight. This can be however easily achieved by integrating the vector over time and dividing its elements by the appropriate mass moments of inertia of the drone:

$$\begin{bmatrix} u_{2,\text{des, rate}} \\ u_{3,\text{des, rate}} \\ u_{4,\text{des, rate}} \end{bmatrix} = \begin{bmatrix} I_{xx}^{-1} & 0 & 0 \\ 0 & I_{yy}^{-1} & 0 \\ 0 & 0 & I_{zz}^{-1} \end{bmatrix} \int \begin{bmatrix} u_{2,\text{des, moment}} \\ u_{3,\text{des, moment}} \\ u_{4,\text{des, moment}} \end{bmatrix} dt \tag{10.18}$$

However, the moments and throttle as calculated with Equation 10.17 might not be achievable, especially in aggressive manoeuvres or in case of bad gain tuning. Therefore, limits are imposed on these moments and throttle to assure the drone is capable of executing the given commands. These limits can be calculated by considering the drone performance. The maximum throttle is taken from the propulsion group, as described in section 7.6, as equal to 52 N. The maximum pitch and roll moment can be calculated by using the maximum force generated by each motor in combination with the arm length. The arm length and angles with respect to body can be found in section 9.6. From this and the CATIA drawings, the moment arms for each motor can be determined. For pitching the drone, they are found to be 8.2 and 12.3 cm for front and back motors, respectively. The moment arms for roll are 14.2 and 11.6 cm for the front and back motors, respectively. The maximum motor force of 13 N can be found in section 7.6. This gives a maximum moment of 3.5 Nm for roll and between 2.1 and 3.2 Nm for pitch. For the pitch moment, the upper values are used for calculations, as pitching backwards is considered less critical than pitching forward. These moments however, this can only be achieved if the opposite motors are completely turned off. This will however not always be the case and therefore a more conservative calculation is performed in which the opposite motors still produce thrust equal to half the drone mass, as would be the case in hover. The maximum roll and pitch moments then become 2.4 Nm for pitch and 2.5 Nm for roll, which is thus used as a limit for $u_{2,\text{des}}$ and $u_{3,\text{des}}$. The maximum yaw moment is determined by the torque generated by the motors. This is defined by Equation 7.16. Since the motors are known the power and RPM can be calculated. The ratio of these will differ in different flight conditions. Here hover flight is considered, which gives a power of 456.6 W with a RPM of 27509. Converting this to rad/s and dividing the power by that number gives a torque of one motor of 0.16 Nm. This means a maximum yaw moment of 0.32 Nm can be achieved in optimal flight conditions. A safety margin is not considered here, since yaw is mostly used in non aggressive flight profiles, where opposing motors could be turned off completely.

10.7.4. Flight Simulation

A flight simulation program is developed, in order to give visual confirmation that the drone behaviour is logical. The drone can not be physically built during this project, which makes simulation useful. The software on the physical drone would receive inputs of processed data from the camera and the IMU. For now, these values are simulated and the program does not include the interfaces necessary for reading sensor input. Furthermore, in real life, the control software would output a signal to Betaflight, which would then perform the low level control. However, in a simulated environment, communication with the firmware is not possible, which is why it is assumed that the outputted commands are performed without errors and delays. Considering the available resources, this assumption is justified as Betaflight is the state of the art drone firmware.

The goal of the simulation is to give a visual view on how the drone is performing. To achieve this goal, three factors are important. The first is that time should be real life. The simulation should not be sped up or be in slow motion. The second is that drone position can be easily visualized on screen. This is done using a reference coordinate system which can be rotated using mouse inputs. Also, scaling of the drone depending on how far it is from the viewer is implemented. The third element is the angles of the drone, which make the drone view rotated as would be the case in real life.

To accomplish this task, a 3D rendering engine is made in Python using the Pygame library, as can be seen in Figure 10.19. This is done in several steps. First, a 3D coordinate system is defined. Just as the standard drone coordinate system, the z-axis is pointing to the ground, the x-axis forward into the screen and the y-axis to the right side. This coordinate system is visualized on a 2D screen using a projection matrix. Depth-scaling is also accounted for in this projection matrix.

$$\text{projection matrix} = \begin{bmatrix} 0 & 1 \cdot (\text{depth scaling}) & 0 \\ 0 & 0 & 1 \cdot (\text{depth scaling}) \end{bmatrix} \tag{10.19}$$

Next, using a standard rotation matrix, as given in Equation 10.2, the coordinate system can be rotated with respect to the viewer. Mouse inputs are assigned to this rotation operation. Afterwards, the drone is given a certain position in the coordinate system using a simple x, y and z coordinate. Next, the drone is plotted with the correct orientation by using the rotation matrix from Equation 10.2 with ϕ, θ and ψ being equal to the yaw, pitch and roll angles, respectively. Finally, this plot is updated in specified time steps to show a fluent simulation. Each time step, the position of the drone is added to a list, which is also plotted to see the drone's path.

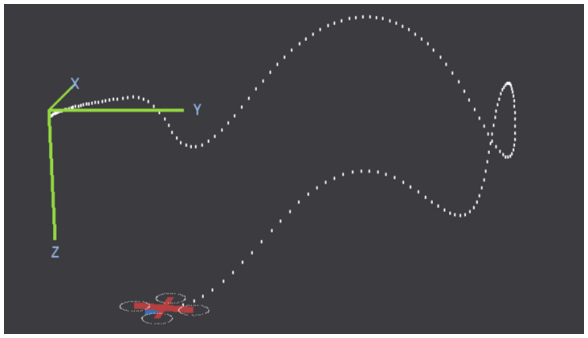


Figure 10.19: Simulation with drone path (white dots) determined by manual control inputs

10.7.5. Gain Tuning

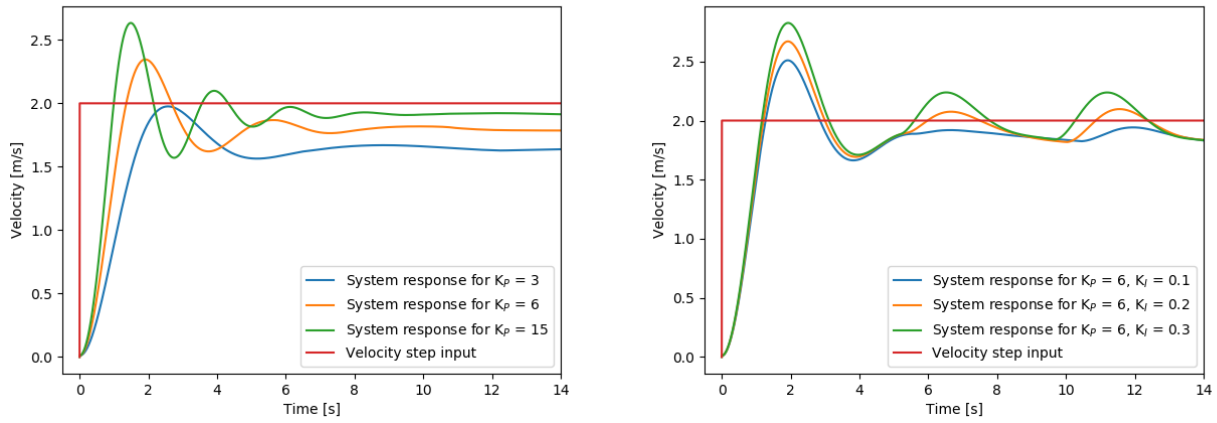
As discussed in subsection 10.7.2, PID controllers involve gains, which have to be properly tuned. The P gain makes sure the response is proportional to the error, the I gain reduces the steady state error, and the D gain decreases the response time. These values can be chosen correctly by giving the system a step input, such as commanding it to instantaneously go from a zero velocity to $2m/s$, and seeing in what manner and how fast the system performs this command.

The gains were first tuned using trial and error to find ranges of values that produce reasonable results. After this initial tuning, the aforementioned analysis using a step input was performed to tune the gains

used to control the translational motion. The rotational motion gains were left at the values determined initially through trial and error, as they gave reasonable results and further tuning them did not have much influence on the system. Finding more optimal values is recommended for future work.

Moving on to the procedure of tuning the translational motion gains, the I and D gains were first set to zero and only the P gain used. When varying the value between 1 and 20, which was the initially chosen range, the response is always characterised by a big overshoot that is quickly damped to a constant value with a steady state error, which can be seen in Figure 10.20a. Using standard tuning techniques like the Ziegler-Nichols or the Tyreus-Luyben methods requires setting the system such that it responds to a step input in the form of a sustained periodic oscillation. However, in this case, the system is a damped oscillation, no matter the gain. For this reason, the program gains could not be tuned using standard methods and instead, it was tuned by observing the system response to different gain values when using a P, a PI or a PID controller. Naturally, a quick response with a small overshoot that is quickly dampened to a small steady state error is preferred.

Looking at Figure 10.20a, the best value for the P gain when using it alone is 6. This is because it allows for a significant reduction of the steady state error - compared to the gain of 3, at which the error decreases from $-0.4m/s$ to $-0.2m/s$. At the same time, the overshoot decreases from $0.7m/s$ to $0.4m/s$, compared to the response when using a gain of 15. And on top of that, the rise time is significantly higher than for a gain of 3, while not being much lower than for a gain of 15, which means that for higher gains, the rise time drops slower and it is not worth increasing the gain further, considering the significant increase in overshoot. A proportional gain of 6 was thus set as a base for further gain tuning. Then, the I gain was added and varied to find one that reduces the steady state error. The results are presented in Figure 10.20b.



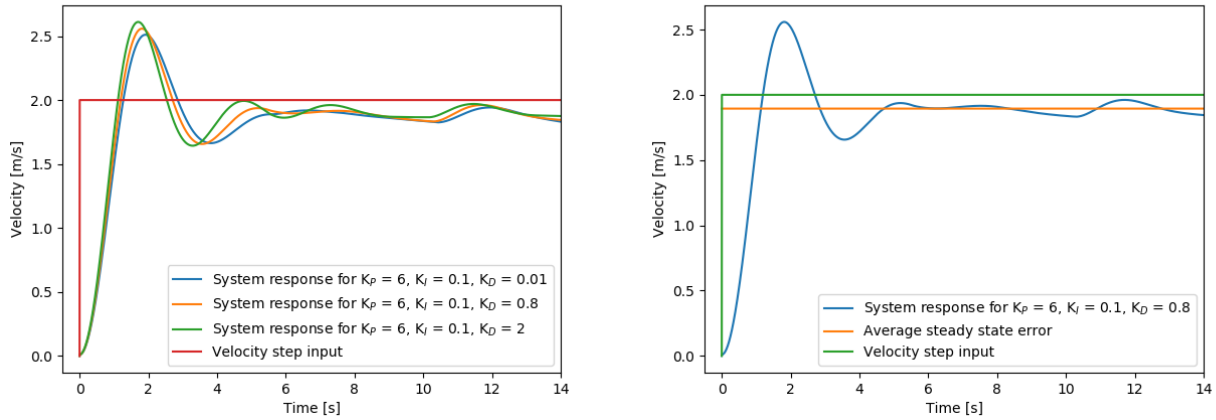
(a) P controller, varying the P gain.

(b) PI controller, varying the I gain.

Figure 10.20: System response to a velocity step input.

As can be observed, compared to the controller with only P gain, with K_P of 6, the I gain of 0.1 reduces the average steady state error from $-0.2m/s$ to $-0.1m/s$ and increases the overshoot from $0.4m/s$ to $0.5m/s$. The response time does not change compared to the controller with only P gain. Furthermore, larger I gains induce larger oscillations in the steady state response. These are centered around the step input velocity so on average the error would be close to zero. However, the oscillations reach large amplitudes of $0.2m/s$, and the overshoot is significantly increased. It was decided to opt for a small overshoot rather than a small steady state error, as long as both values are within reasonable margins. This is because small deviations from the desired speed are considered acceptable by the team, as these are only indications from Navigation. On the other hand, large overshoots could significantly affect the system stability if a wind gust or a delayed input are encountered. For this reason, the I gain of 0.1 was chosen. Finally, using a P gain of 6 and an I gain of 0.1, the D gain was added and varied as can be seen in Figure 10.21a. Adding the D gain clearly does not make a big difference with regards to the steady state error - it is still around $-0.1m/s$. Larger D gains do however

increase the response speed and reduce phase lag and for this reason, a D gain of 0.8 was chosen. Using the chosen gain values, the step input response has an overshoot of 28% and a steady state error of 5%, with a settling time of 4.5 s, as visualised in Figure 10.21b.



(a) PID controller, varying the D gain.

(b) Final values of PID controller gains.

Figure 10.21: System response to a velocity step input.

10.7.6. Verification and Validation

For the control and stability subsystem, only verification can be performed. Validation would involve testing the performance of the algorithm on an actual drone, which is outside of the scope of this project. Therefore, only verification of the code is performed. This is done in two steps. The stability of the code is assessed by a number of different tests. The outcome of these tests are often 2D or 3D graphs showing the location, velocity or control outputs of the algorithm. For the former two, the outputs need to be implemented, updated and plotted. Another way of assessing the outcome of the test is by using the simulation as described in subsection 10.7.4. These plotting and implementation tools also need to be verified to assure they can be trusted. The verification of these tools is first discussed, after which the verification of the algorithm is discussed.

Three important tests are performed to verify the implementation, plotting and the simulation. First of all, step inputs are applied to the throttle and angular rates. After this, the angles and drone paths are plotted to verify if the results are as expected. The exact numbers have been checked by hand calculations. Next to this, other important parameters such as the pitch, roll and yaw angles are set to constant values and again, the drone response is checked and verified by visual inspections and hand calculations. Secondly, pulse inputs are given to throttle and angular rate variables to assess the stability of the simulation and implementation code. The drone path, angles, velocity, and acceleration are plotted to verify the results. This is checked by hand calculations. Thirdly, values of zero are given to throttle and angular rate variables, as well as to the body angles and moments. This is done to check the system for singularities, which could result in wrong implementation or simulation. All these tests were done and passed successfully.

The algorithm and its stability is verified using five tests. Firstly, by giving a step input for a desired angle. Secondly, by simulating a wind gust pulse disturbance. Thirdly, by providing input data with noise. Then, by adding a time delay to input values. And finally, by providing different types and shapes of arcs. The setup and result of all of these tests is explained in the following section.

Desired angle step input

The goal of this test is to assess the stability, convergence and rate of convergence of the algorithm to a desired angle input. This test is implemented by hard coding a constant, desired angle instead of letting the algorithm calculate and update it itself. The result for a yaw angle step input of 40 degrees can be seen in Figure 10.22. As can be seen, the angle is approached at a constant moment (in fact, the maximum allowed value as explained in subsection 10.7.3), however, does not oscillate and

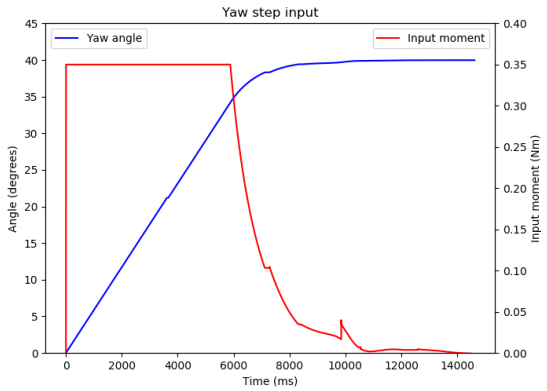


Figure 10.22: System response to yaw step input.

quickly settles. Therefore, this test is passed.

Wind gust pulse disturbance

The goal of this test is assessing the sensitivity of the system to disturbance force. This can be a wind gust, a small object hitting the drone or the drone bumping into an object. This test is implemented by giving the drone an acceleration of 10 000 m/s^2 in negative y direction for the duration of 0.001 seconds. This corresponds to a short wind gust of 10 m/s. This is done while the drone has a speed of 5 m/s and is moving in the positive x direction. The path of the drone is compared to its path without the disturbance input. This can be seen in Figure 10.23. As can been seen, the wind gust significantly alters the path of the drone, however the drone still manages to reach the desired point and returns to its previous course with minimal oscillations. Therefore, this test is passed.

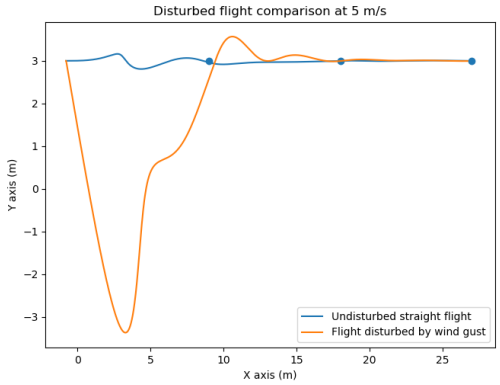


Figure 10.23: 5 m/s straight flight compared with 5 m/s disturbed flight

Noisy input data

The goal of this test is assessing the ability of the drone to follow a trajectory in case of noisy input data. This test is implemented by providing a semi-circular drone trajectory. The inputs of the system, namely velocity and waypoint position are combined with 20 cm and 20 cm/s random Gaussian noise. The performance of the drone is then compared to a drone without noisy input. Based on the ability of the drone to still follow the trajectory, the test can be assessed as passed or failed. The results of this can be seen in Figure 10.24 and Figure 10.25.

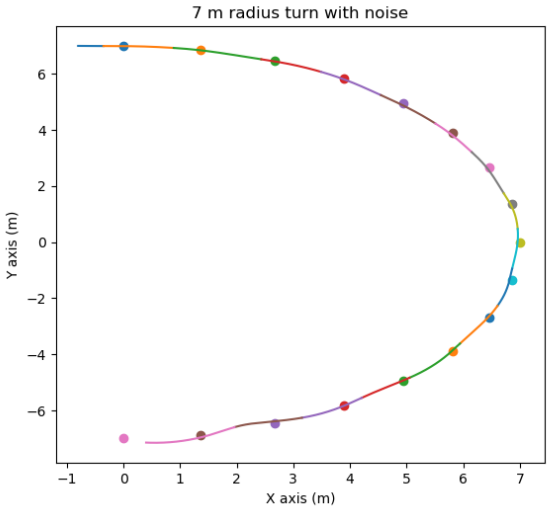


Figure 10.24: Drone trajectory during a turn with input noise

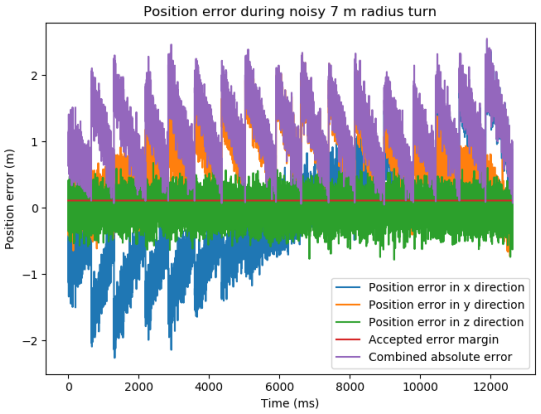


Figure 10.25: Position error input during a turn with input noise

These figures can be compared with Figure 10.26 and Figure 10.27, where the same manoeuvre is executed without noise. As can be seen, though the input is much more noisy, the drone is still able to perform the turn with high accuracy.

Delayed input data

The goal of this test is assessing the ability of the system to handle delays in the system, which can be caused by slow computation or speed loss in cables and conversions. This test is not implemented by the team, but future teams are encouraged to perform it. This test would be implemented by providing the drone with a simple trajectory at flight speed of 25 m/s. The current position input should be delayed by at least 1 ms. The performance of the drone is then compared to a drone flying without delay. Based on the deviations of the drone from the path, the test can be assessed as passed or failed.

Different turn shapes

The goal of this test is assessing the ability of the system to fly through turns other than the one it was specifically designed for (i.e. a 3 m radius arc at 10 m/s). This test is not implemented by the team, but future teams are encouraged to perform it. This would be implemented by flying the drone through the 3 m radius turn at higher and lower speeds and by flying the drone through smaller and bigger radius turns at 10 m/s. Based on the observed sensitivity of the drone to other types of turns this test can be assessed as passed or failed.

10.7.7. Results

Based on the theory of D. Mellinger, adapted by the team, a basic control code has been developed. The gains are tuned by observing the system response to step inputs. In addition to this, extensive verification has been performed and a simulation has been developed. It must however be stressed that this code is meant as a starting points for teams using the drone in a race and is therefore, not fully optimal yet. An example of this is the 7 m radius semicircular trajectory as can be seen in Figure 10.26. The errors during this turn can be seen in Figure 10.27. These show that the drone is very effective in moving from waypoint to waypoint. However, the speed is still low at just 2 m/s. Also it is not the 3 m radius arc which should be aimed for. The reason for this are the limitations applied to maximum thrust and angular rates. Before imposing those limits, the code could perform the required 3 m turn at 10 m/s but imposing limits means the drone cannot rotate as fast and it falls out of the turn. This is likely due to imperfect implementation of the Mellinger algorithm and due to sub-optimal gain tuning, which can both be improved in future work.

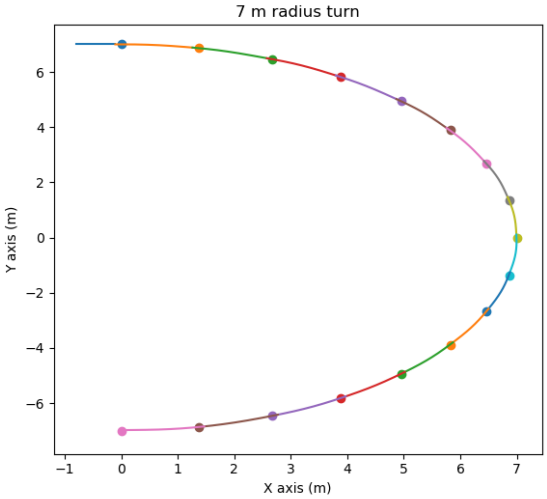


Figure 10.26: 7 m radius semicircular trajectory.

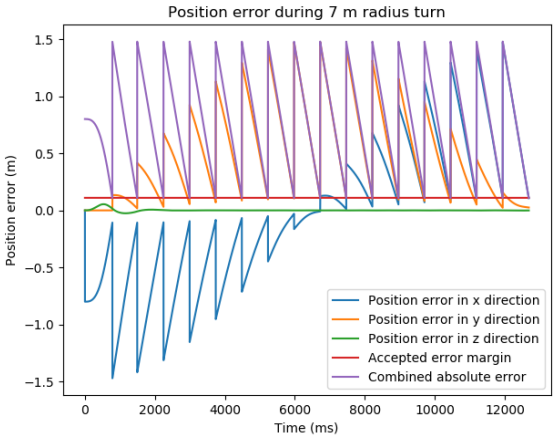


Figure 10.27: Position error during a 7 m radius turn.

10.7.8. Future Work

The next step in the development of this control code is validation. The code should be used on actual hardware to validate it works as expected. Based on this improvement to the code should be made, as well as gains adjusted to reach the desired result. Once this is done teams should try to optimize the code as to assure the drone can reach all their manoeuvrability requirements, which are possibly even more aggressive than the ones mentioned in this report. Next to this the performance of the drone should be compared with the results mentioned in [41]. Based on this further improvements might be possible to make the drone even more agile and capable of executing complicated manoeuvres as shown in the paper. Another important feature to be added and validated in flight is a safety mode, which safely lands or shuts off the drone in case of a malfunction.

11

Design Summary

In conclusion to the design steps that are in the chapters up until now, the final design of the AIgle drone is presented in this chapter. That is done first in the section 11.1, which gives visuals of the AIgle drone model. Succeeding that the technical budgets are reflected upon in section 11.2, from which it can be seen that the mass budget was exceeded significantly while the financial budget is not used entirely. At the end of this chapter the sustainability of the final design is reevaluated, looking into the different aspects of sustainability regarding the lifetime of the AIgle drone.

11.1. Drone Layout

Below in Figure 11.1 the model of the assembled drone can be seen. In this render a transparent cover is chosen, showing the electronics housed inside. In Figure 11.2 a more detailed drawing can be found giving the overall dimensions of the AIgle drone concept. All the drawings, renders as well as the actual CAD files and the software programs can be accessed on GitHub[1].

Figure 11.1: CATIA rendering of final AIgle drone design concept

[1] https://vguillet.github.io/AIgle_Racer/, (Retrieved June 27, 2020)

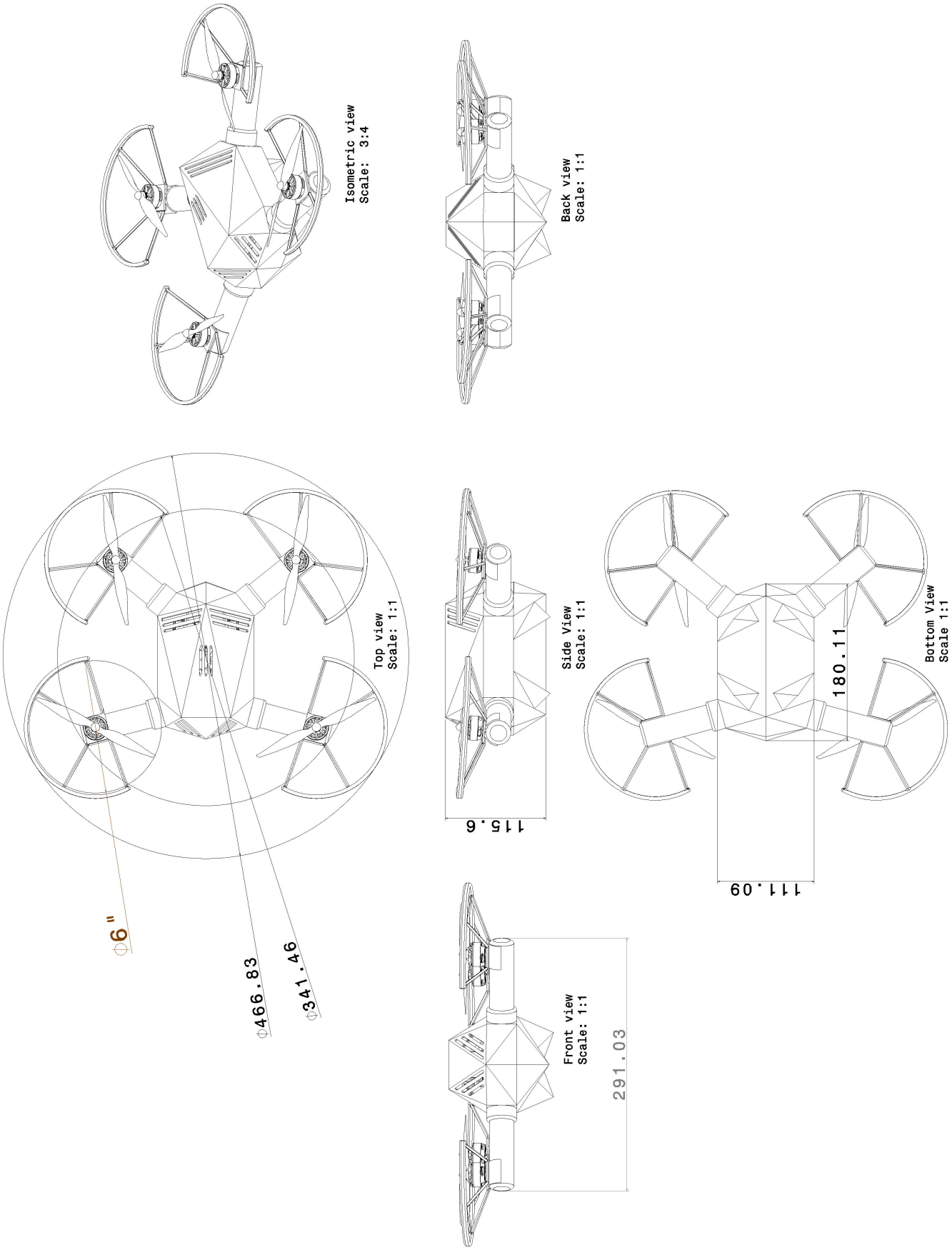


Figure 11.2: CATIA drawings of final AIgle drone design concept in mm except for the 6” propeller

11.2. Technical Budgets

Now that all exact dimensions of the drone are known, the technical budgets can be reevaluated. Though the budget was continuously monitored throughout the entire design process, now a final version can be shown. This can be found in Table 11.1. Here both the preliminary budget, as described in section 4.3, and the final budget breakdown are added side to side so they can be compared.

Table 11.1: Technical budget breakdown

Subsystem	Budgeted Mass (g)	Final Estimated Mass (g)	Error margin(+-%)	Budgeted Power (W)	Final Estimated Power (W)	Error margin(+-%)	Budgeted Cost(€)	Final Estimated Cost(€)	Error margin(+-%)
Structure	300	534.9	-	-	-	-	425	273.80	-
Frame	250	100	10	-	-	-	375	30.00	5
Arms including mounts	-	150.3	10	-	-	-		96.00	5
Cover	50	132.2	10	-	-	-	50	41.52	5
Motor mounts		54.0	10	-	-	-		49.25	5
Propeller guards		55.3	10	-	-	-		17.03	5
Damping systems		43.1	10	-	-	-		40.00	5
Aerodynamics	-	-	-	-	-	-	-	-	-
Electronics	180	233.47	-	48	23.2	-	1,050	690.24	-
Flight controller	-	6	2	3	2	2	-	40.20	5
Communications modules	20	5.3	2	3	5	2	50	54.50	5
Antennas	-	14.4	2	-	-	-	-	5.32	5
Computer	150	76	2	45	15	2	1,000	357.44	5
4 in 1 ESC	-	17.5	2	-	-	-	-	60.00	5
Printed Circuit Board	-	30	10	-	-	-	-	150.00	50
Wiring	-	80.27	-7,+2	-	-	-	-	10.00	5
MicroSD Card	-	0.5	2	-	-	-	-	10.29	5
LED Strip	10	3.5	2	1	1.2	-	13	2.49	5
Power and Propulsion	365	500.42	-	216	2,120	-	350	107.33	-
Battery	250	352	2	6	-	-	150	27.91	5
Propellor	25	20.42	2	-	-	-	50	8.18	5
Motor	90	128	2	50	2,120	5	150	71.24	5
Machine Vision	150	17.5	-	18	1.7	-	500	79.53	-
Camera	150	17.5	2	18	1.7	2	500	79.53	5
Navigation	-	-	-	-	-	-	-	-	-
Control	-	-	-	-	-	-	-	-	-
Other	-	3.5	-	3	-	-	163	27.49	-
Total	1000	1,285.79		300	2,144.9		2,500	1,175.90	

As can be seen, the final budget breakdown is differing significantly from the first estimate. This has a number of reasons that can be approached on a subsystem basis. First of all the structures subsystem: Here one can observe that the cost budget is met, however the mass budget is greatly exceeded. This is mainly due to the fact that the size necessary to survive the crash scenarios was heavily underestimated and damping systems as well as a stronger structure needed to be designed. Furthermore a number of components, like the motor mounts, were not yet clearly identified and therefore not estimated in the preliminary phase. However, these components are crucial to a functioning design and therefore, their mass is necessary to be included. Second of all the electronics subsystem: Here again the cost budget is met, however the mass budget is exceeded. This is partly due to more components being identified than in the preliminary phase, however most of the mass gain is caused by the wiring mass. In the preliminary phase this was estimated as low to negligible, however, after investigation it turns out to be the heaviest component of the subsystem. This is due to the fact that heavy wires are need for the high power demand of the motors, as described in section 8.5. Then the power and propulsion subsystem: Also here the cost budget is met and the mass budget is exceeded. Here further investigation and improved tools show that more mass is required to meet the flight requirements than initially estimated. Lastly the machine vision subsystem: Here both the cost and mass budget are met. The great difference between the preliminary estimation and the final value can be accounted to the fact that for the current design only one sensor (apart from those integrated in the electronics) is used, instead of the multiple which was accounted for in the initial budget. What can also be seen in the table is that the error margins are greatly reduced. This is partly due to improved and better verified tools, but mainly due to the fact that for electronics, power and propulsion and machine vision actual off the shelf components were selected with known masses and cost. The error margin for wiring has a more negative than positive bound since the mass already includes a 5% safety factor.

As already indicated in chapter 9 a 100 gram more lightweight design would be possible if the arms are to be designed for just 15m/s crashes. This might also allow to use the next more lightweight battery, which is 50 gram less than the currently used one [2]. Still, this battery choice would need still to be verified with an iteration of the tools as explained in chapter 7.
If this would be the case a total reduction of 150 gram would result in a total mass of the drone of 1139 gram.

The decision was made to leave out the processing cost from the final budget. This was because the decision of how and where to allocate processing resources to is mostly dependent on the pipeline and software design elaborated by the competing teams themselves. As a results it is subject to major variations and including one would not be representative of the various possible designs.

11.3. Performance Characteristics

Next is the performance characteristics of the drone. This applies mostly for the hardware related subsystems, since those are the only subsystems whose performance results are not set by the user. First a summary of the Power and Propulsion subsystem performance is given in subsection 11.3.1. Then the performance of the Electronics subsystem is given in subsection 11.3.2. Finally the Structural performance is mentioned in subsection 11.3.3.

11.3.1. Power and Propulsion

Starting with the propulsion system, the AIgle is equipped with four SunnySky Edge R2305 (2300kv) motors, which are, in combination with the APC 6x4E propeller, able to deliver a maximum total thrust of 51.6N. The total mass of 1286g leads to a thrust-to-weight ratio of 4.1. This enables the drone to comply with the turn requirement AIG-TE-HW-PF2, which was the most limiting requirement for the P&P subsystem. With this thrust-to-weight, a maximum linear acceleration of 4g can be met in vertical direction and 3.98g in horizontal direction. The thrust it can deliver at maximum speed is 42.6 N, resulting in a total maximum speed of 32.1 m/s. AIgle has a total flight time of 63 s with every subsystem at full power, or a hover time of 9.5 minutes.

11.3.2. Electronics

One of the main systems of the electronics is the Xavier NX companion computer. It contains a 384-core GPU with 48 tensor cores, as well as 8GB RAM. It designed for AI applications, allowing it to perform various machine vision tasks while processing at least the required 60 frames per second using various deep learning models. The Xavier NX module fits inside a carrier-board, which is custom designed to contain the required connections to the flight controller, battery, WiFi module, camera, and micro SD card. The flight controller, which runs the Betaflight firmware, has an F7 processor which runs at 216MHz, as well as two different IMUs with sampling rates of 8kHz and 32kHz. It also has a barometer, and supports SmartPort telemetry. The ESC supports the dshot1200 communication protocol, which has lower latency than other communication protocols between the FC and ESC. It also has a current sensor and supports ESC telemetry, which allows for debugging. The camera has frame rate of 65 fps at a resolution of 1920x1080, and a field of view of 139 degrees diagonally. The communication modules include two radio receivers and one WiFi module, which are all able to transmit and/or receive the necessary information to and from the ground stations at (at least) 300m distance.

11.3.3. Structures

The purpose of the structure of the drone lies primarily in providing structural integrity, thus providing a basis on which the other on-board systems can be mounted. A stiff frame is required for good control characteristics and high efficiency in accelerating and maneuvering. The stiffness is achieved by having carbon fibre arms connected to a 3D printed mainframe made out of Polyether Imide (PEI),

[2] https://www.banggood.com/ZOP-Power-14_8V-3200mAh-75C-4S-Lipo-Battery-XT60-Plug-for-RC-Airplane-p-1548932.html?rmmds=search&cur_warehouse=CN, (Retrieved June 25, 2020)

one of the stronger and stiffer 3D printing filaments commercially available.

Besides that the structure provides protection of the sensible on-board systems. The design of the drone includes propeller guards and landing gear as first points of contact during crashes. They are providing the first crumble zone. This allows the most electronic components to survive a 3m drop onto concrete, resulting in an impact velocity of 7.7m/s, or about 28km/h. But the processor is much more fragile and can handle only impact loads of up to 50g during operational conditions. That is why both the carrier board along with the processor and the cooler on top are mounted on spring-damper standoffs, which allow for another 8mm of cushioning zone.
Furthermore the drone is designed such that it is possible to safely grab it from beneath. Also the propeller guards prevent injuries in case of contact with humans.

11.4. Risk Evaluation of Final Design

For the evaluation of the risk mitigation strategy, the risk map in Table 5.4 is compared to the risk map that is expected from the actual final design in table Table 11.3. A strong focus was placed on the protection of the electronics, especially effecting the following risks:

- ES1 - Breaking of the computer
- ES2 - Breaking of the camera
- PP3 - Battery catching fire

A large number of measures to reduce those risks have been implemented. These include shock dampers for the Xavier NX, a protective shell around the electronics, the camera is placed inside the shell, there are extra crumble zones added. Despite all of this, as can be read in section 9.10, the drone will still not be protected from a crash of 30 m/s. Especially the effect on the computer (ES1) was not as big anticipated, therefore there is a small change there from the mitigation plan. It can, however, withstand a drop from 3m which equals an impact velocity of 7.7 m/s.

There are a few options that can be taken into account in order to further protect the drone. First, lowering the speed requirement significantly would result in a lower risk for the drone, which is something that can be considered by the customer. Having competitions use a highly cushioned gate reducing the deceleration of the drone on impact would also lower the risk greatly. The exact design of the gates and an evaluation of how it would actually perform in such a situation is something that can be investigated in future work.

Furthermore, ST1 - breaking of structural components due to a crash at 30 m/s - will be more likely to happen than expected. Because this also includes the protective structural components, which are designed to break in a crash in order to provide a crumble zone for the electronics.

Lastly, the mitigation strategy for MN1 - misidentified gate - was the use of a RANSAC algorithm. However, a RANSAC algorithm is not used during the design and the actual risk will therefore be higher than initially thought. This is something that can be easily implemented by the racing teams designing their own algorithm, so it will have minor consequences for now.

Table 11.2: Risk map after the mitigation strategy.

Impact \ Probability	1	2	3	4	5
5	ES5		ES1 PP3		
4	ST3, ES9 CS4	CS2, ES10 ES6 PP4 MN3		ES2	
3	ES3				
2	CS3, ES8	ST2, ES4	PP1 MN4	ST1 MN1	
1	ST4, ES7 CS1			MN2	PP2

Table 11.3: Expected risk map after the final design.

Impact \ Probability	1	2	3	4	5
5	ES5		PP3	ES1	
4	ST3, ES9 CS4	CS2 ES6 ES10 PP4 MN3		ES2	
3	ES3				
2	CS3, ES8	ST2, ES4	PP1 MN4		ST1 MN1
1	ST4, ES7 CS1			MN2	PP2

11.5. Sustainability Evaluation of Final Design

This section analyzes the final design based on the sustainability criteria described in chapter 6. Since the final design is not compared to any other design, it makes no sense to score it. Therefore, a qualitative analysis is done which discusses to what extent each criterion has been implemented towards a sustainable solution. It also discusses whether the sustainability can be improved, and how this would affect the performance of the drone.

Criterion 1: Estimated power consumption
Reducing the power consumption of the drone has been a major driver throughout the design, not only for improved sustainability, but also for improved performance such that battery life during flight can be maximized. The power consumption at maximum throttle is about 2200W, and is able to fly at this throttle setting for about a minute. This can be compared to the RacerAI used in the AIRR drone race, which has almost double the weight at about 3kg [3]. For the same performance, it can be deduced that the RacerAI would require a lot more power as the thrust requirement is over double. Therefore, it can be said that the drone design is more sustainable in terms of power consumption than it's current competitor. The power consumption could be aimed to be reduced even further by reducing the weight further. The weight of the drone is made up of the LiPo battery, the structure, and the electronics, all of which cannot be changed much due to their crucial functions. More dramatic design adjustments could be made, which would influence many systems in the design and possibly require many additional design iterations.

Criterion 2: Volume of solid material
The volume of material has been aimed to be reduced, mainly because it brings functionality to the drone by making it more compact. Only necessary components that bring functionality to the drone were sought for. For example, a custom carrier board was designed, as all carrier boards on the market contain components that are not a necessity to the drone. Another way in which volume was reduced was by using a carbon fibre composite, which is lightweight, and hence, requires less volume of material. However, the structure also contains the polyetherimide (PEI) thermoplastic for the main frame, as well as stainless steel for the dampers. These materials are not as structurally efficient and therefore, volume could be saved by finding more lightweight solutions by, for example, using only carbon fibre instead.

Criterion 3: Durability of design
Durability of the design was one of the main goals, which was achieved by designing a structure that can isolate the main hardware from impact loads and collisions. The drone is protected from drops up to a height of 3m on concrete. This greatly increases the life of the drone, but harder crashes could damage components and would be wasteful. The most fragile parts of the drone are the computer and the camera, which cannot be replaced by more rugged hardware. The best way to improve durability would be to improve the structure even further to make it more resistant to impact. For example, this could be done by making the structure larger, allowing for larger displacements during collision to reduce the deceleration of the hardware. This however, causes other sustainability issues resulting from the increased volume and mass.

Criterion 4: Energy usage of processes during manufacturing phases
The main materials of the drone structure are carbon fibre reinforced composite (CFRC), polyetherimide thermoplastic, and stainless steel. CFRC is known to require more energy than other materials due to the relatively newer technologies being used. For example, one study found that CFRC requires 800MJ/kg, compared to 50MJ/kg for producing steel [42]. Additionally, polymers were found to have a similar carbon footprint to stainless steel [43]. Therefore, the sustainability of the material manufacturing phase could be improved by eliminating the use of CFRC. However, this would result in other phases of the life cycle to become less sustainable, and affect performance of the drone, due to weight increase. There is not much that needs to be done in the final manufacturing phase. Firstly, there is soldering, which is necessary and cannot really be reduced. Secondly, there is 3D printing of the base of the structure. Both of these processes do not take so much energy, especially considering

[3] https://www.wired.com/story/coders-versus-human-pilots-drone-race/, (Retrieved June 20, 2020)

that they are done only once. The base could have its material changed to remove the need for 3D printing, but this would not necessarily improve the overall sustainability.

Criterion 5: Material waste during manufacture phases
It is difficult to determine the material waste during the component manufacturing phase. However,there is not much material waste in the final manufacturing phase, as it consists only of soldering, 3D printing and bolting components together. These processes all result in nearly zero waste. However, there can be some over-buying of components due to suppliers only selling at larger quantities. This can lead to waste, so it is recommended that the consumer buys only what is required to build and maintain the drone.

Criterion 6: Types of joints used for assembly
The major systems in the drone are mainly connected by bolts, which are temporary joints. This improves the maintainability, reusability and recyclability of the drone. On the other hand, the electrical components consist of PCBs with multiple sub-components of various materials. These PCBs are more difficult to disassemble. The only permanent joints used are to connect the arms with the motor mount using a thermoset adhesive. Since both parts are carbon fibre composites containing a thermoset matrix, the adhesive does not make the part less recyclable.

Criterion 7: Volume of biodegradable and recyclable material
No part of the drone can be considered biodegradable, however most of the drone can be recycled in some way. Most of the drone consists of electronic components, which can be recycled, but it is more complex due to the many materials they consist of. The structure of the material is made up of a carbon fibre composite material, polyetherimide, and stainless steel. The carbon fibres within the composite can be recycled, usually by burning away the thermoset matrix, and therefore it is not fully recyclable [4]. Also, the resulting recycled fibres are a lower grade. Polyetherimide is a thermoplastic and can therefore also be fully recycled by melting. Stainless steel in also fully recyclable. It would be possible to use biodegradable materials, however, this would likely impact the durability of the design, and therefore not necessarily improve the sustainability. A way to more easily recycle the drone would be to use less electronics, however, this is not easily done, as the electronics are a critical part of the drone. Also, recyclability can be improved by using only 100% recyclable materials. Therefore, carbon fibre could be replaced by thermoplastics and metals. However, this wouldn't necessarily improve sustainability as weight and volume would increase.

Criterion 8: Sustainability of component suppliers
Not too much information is given about each supplier, however, some things are known. Firstly, all electronic components are RoHS certified, meaning they do not contain very hazardous substances, which could be damaging to the environment and dangerous to workers during manufacturing and recycling [5]. Secondly, Nvidia, the company that supplies the Xavier NX computer, shows that they consider sustainability by describing their efforts in their annual 'Corporate Social Responsibility Report' [6] . For example, NVIDIA has set goals on reducing greenhouse gases, waste, and energy. They also aim to improve social sustainability such as employee welfare and diversity.

The sustainability of suppliers can be improved by making sure products are manufactured in countries that promote welfare of workers. For example, factories in China are known to sometimes have bad working conditions. The problem is that China is the largest exporter of products, so avoiding components from China is difficult and not always traceable.

Criterion 9: Location of component suppliers
It is not always possible to track the location of the suppliers. Consumers should always try to obtain the necessary components from local suppliers to avoid excessive transportation. However, this is not always in the best interest of the consumer, as countries such as China often sell the same products at much lower prices. Additionally, most branded components, such as the computer and flight controller,

[4] https://www.compositesworld.com/articles/sustainable-inline-recycling-of-carbon-fiber, (Retrieved June 20, 2020)
[5] https://www.rohsguide.com/rohs-faq.htm, (Retrieved June 20, 2020)
[6] https://www.nvidia.com/content/dam/en-zz/Solutions/documents/FY2019-NVIDIA-CSR-Social-Responsibility.pdf, (Retrieved June 20, 2020)

would be manufactured at only one location in the world. So distributing the components to consumers around the world is unavoidable.

Criterion 10: Lifespan of components
The lifespan of components is not accurately known as suppliers generally do not state this. However, it can be said with high certainty that the major components would last long since they come from reputable companies. Not much can be done to increase the lifespan of components, except to maintain them well.

Conclusion
Some suggestions for how the sustainability of each of the phases could be improved were discussed. It is however difficult to determine precisely how the overall sustainability of the design can be improved, as most design changes would improve one phase of the product's life, while worsen another phase. For example, the use of carbon fibre reinforced composites is favoured for criteria 1 and 2, reducing energy usage of the drone and it's material volume. However, it is not favoured for criteria 4 and 7, due to it's poor manufacturing and recycling characteristics. To have a clear measure of how of how certain design changes can affect the overall sustainability requires more complex analyses, requiring more details about manufacturing processes used in the creation of components, which is not available.

Overall, the final design is quite sustainable considering, the strict requirements that needed to be met. Sustainability was not set at the highest priority, as specific performance parameters needed to be met. Implementing sustainable solutions would often affect performance too much, especially in terms of weight and energy usage. Therefore, the sustainability criteria that were implemented very well were mainly to also improve the overall performance of the drone.

12

Production and Operations

The following chapter explains the process of producing the drone in detail. This is done in section 12.1 with the help of a flow chart and a table with all the parts that need to be acquired. After that the operations and logistics are explained in section 12.2, describing the necessary environment for the AIgle drone to work in.

12.1. Manufacturing, Assembly, Integration Plan

In the Figure 12.1 the production plan of the AIgle drone design can be seen. It starts with ordering the raw components. Preparing them is next, followed by assembly. Finally the assembly needs to be checked, code can be uploaded and the drone can be tested. In order to get good overview of the materials required to build the drone a bill of materials is provided below in Table 12.1 including a list of suppliers.

Table 12.1: Bill of materials and list of suppliers for AIgle drone parts

Part description	**Name**	**Cost** [€]	**Supplier**
Power & Propulsion			
4 x Propellers	APC 6x4E	8.18	See weblink
4 x Motors	SunnySky R23x05 (2300 kv)	71.24	See weblink
Battery	Tiger Power: 3600 mAh battery 65C 4S	27.91	See weblink
Structures			
Motor mount connector	Carbon Fiber 90 Degree Tangent Tube Mount - 12” Long	49.25	See weblink
Tube for arms	Tube, Filament Wound, Sanded, 0.875x1.125x12 Inch	96	See weblink
Dampers for electronics	IDC S2 Dampers - SM2-044-A	≈40	See weblink
3D printing filament	PEI ULTEM 1010 FILAMENT 500g	139	See weblink
Electronics			
Receiver 1	FrSky R-XSR	23	See weblink
Receiver 2	FrSky XM	13	See weblink
WIFI Module	Intel Wireless-AC 9560	18.50	See weblink
Carrier Board	Custom Carrier Board	≈150	See weblink See weblink See weblink
4 in 1 ESC	T Motor F45A V2.0 6S 4IN1 32 bit	60	See weblink
Flight controller	Lumenier Lux F7	40.20	See weblink
LED’s	DC12V 5050 LED Strip Licht	2.49	See weblink
Computer	NVIDIA Xavier NX + Development kit	357.44	See weblink
Camera	e-CAM50_CUNANO	79.53	See weblink
Wiring	0.2m 10AWG and 0.2m 26AWG	10	See weblink
XT150 Connector	XT150 Connector/Plug	5.32	See weblink
DC Power Connectors	Sourcingmap 5.5mm x 2.5mm DC Power Connectors Male	12	See weblink
Antennas WIFI	Laird Connectivity 0600-00057	6	See weblink
MicroSD 64 GB	Kingston Canvas Select Plus 64 GB	10.29	See weblink

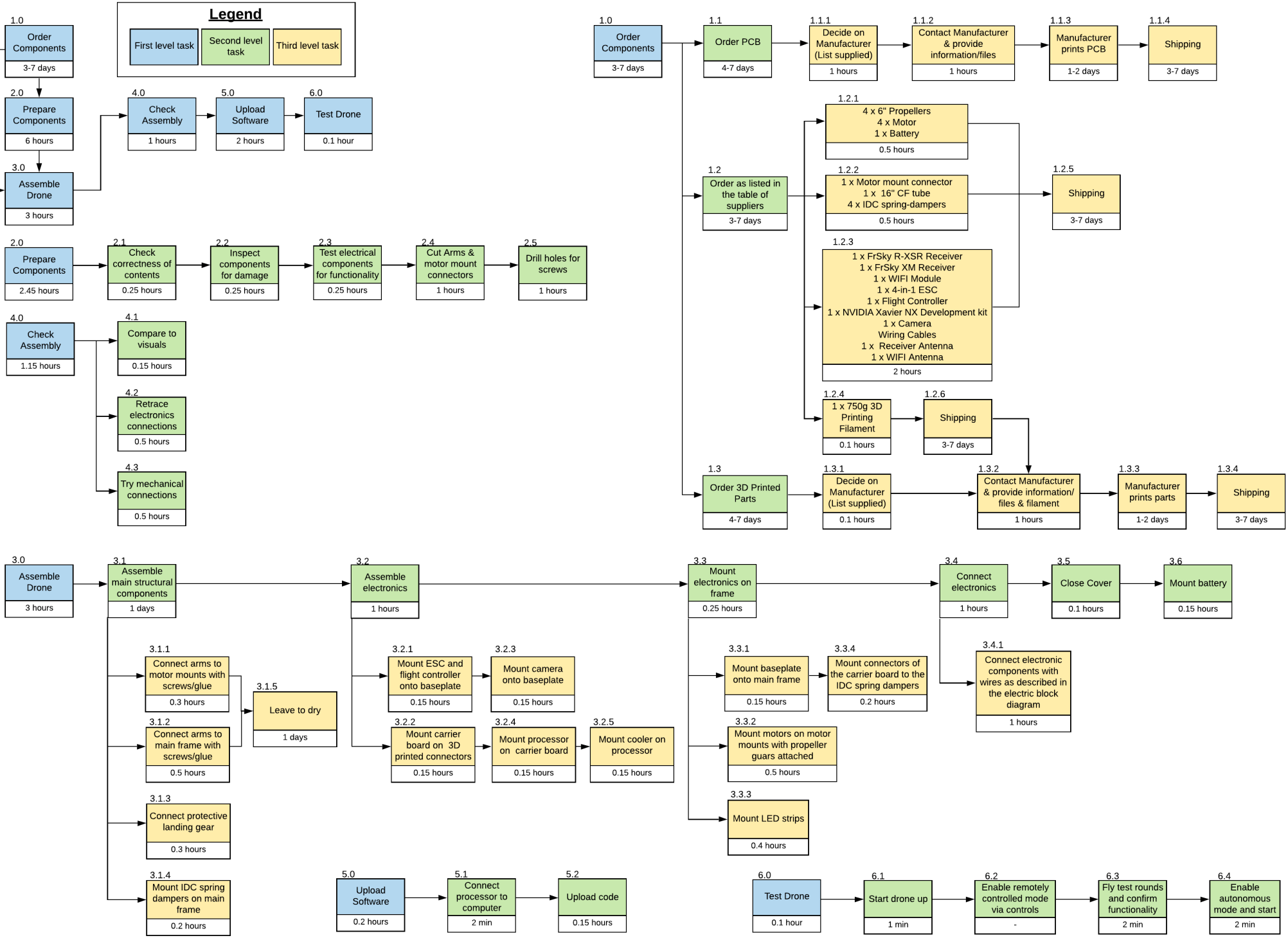


Figure 12.1: Production Plan of Algle drone

12.2. Operations and Logistics

Although the drone design and its characteristics are important, the system cannot operate in a vacuum. The process of using the drone is paired with operations and logistics. These are all actions required to build, use, maintain and dispose of the drone. This can be shown in a flowchart as found in Figure 12.2. It shows all steps from begin of life to end of life, as well as all equipment and documentation required. The diagram is split into six categories, namely operations, repairs, support, maintenance, logistics and disposal. Operations include all main functions and can be described as the normal use. Repairs include all actions and information necessary for repairing the drone when a part breaks, so that the lifetime of the drone can be maximized. Support shows all the information and communication that the developer of the system has to have with its users. It assures that the users can optimally use the system and ensures that feedback can be given and implemented. This step can be applied at any step in Figure 12.2. Maintenance describes all steps needed for maintaining the drone, which can prevent crashes and ensures better performances overall. Logistics has to do with the moving of the drone and the setting up for operations. It thus includes (dis)assembly and storage, but also charging and ground equipment. Disposal has to do with the end of life phase of the drone. The steps described ensure that the drone is properly disposed of, in an as much sustainable way as possible.

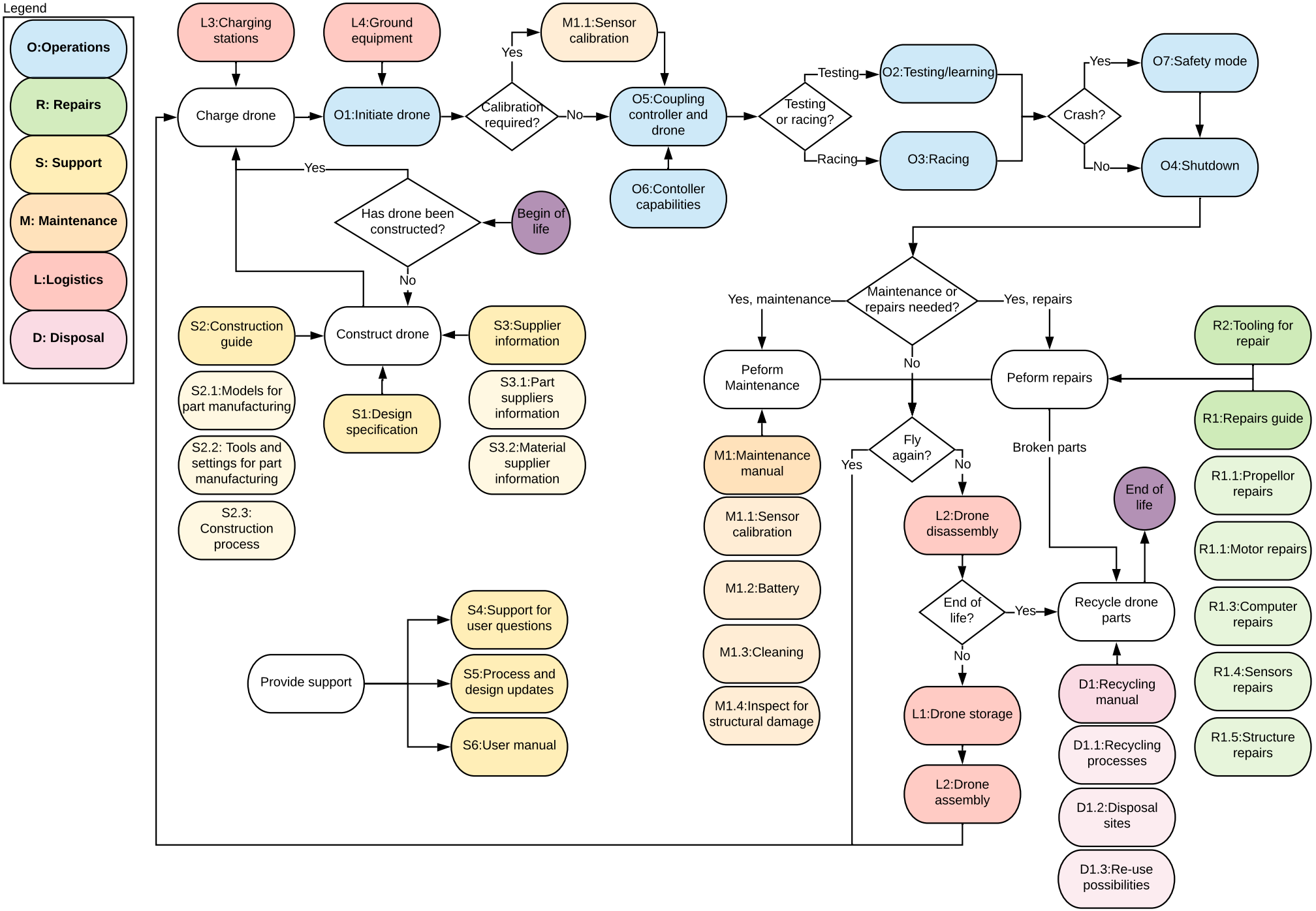


Figure 12.2: Operations and logistics flow Diagram

13

Reliability, Availability, Maintainability, Safety

In the following chapter, the reliability, availability, maintainability and safety characteristics of the AIgle drone are discussed. The reliability describes the range of the conditions, in which the drone can operate without malfunctioning, as discussed in section 13.1. In section 13.2, it is investigated whether the different components that are used for building the drone are available to the manufacturer and under which circumstances. After that the guidelines on how to maintain the AIgle drone correctly for ensuring the reliability are explained in section 13.3. Finally, in section 13.4, the safety characteristics of the drone and its environment are listed.

13.1. Reliability Characteristics

The characteristics regarding the drone's reliability are given in the list below:

1. The Xavier NX is rated for mechanical half sine shocks of 50g during operational condition and 140g during non-operational condition [14].
2. The Xavier NX is rated to be powered on for 168 hours with a temperature of 85 deg and 85% relative humidity [14].
3. The structure is designed to provide the structural integrity of the drone. If maintained correctly, no parts should loosen and endanger the drones functionality.
4. Safety factors of 2 have been used for the design of structural components.
5. The materials used for the structural components are able to resist high humidity and full submersion in water. The functionality should however always be checked after such events (see maintenance).
6. Motor power required at maximum settings (100% throttle) is below the power that the motor can handle with a margin: 528 Watts [1] indicated of 1000 Watt maximum [2].
7. Motor temperature at maximum setting (100% throttle) is below the temperature that the motor can handle with a margin of 30 degrees Celsius: 42 degrees Celsius (indicated by eCalc, see footnote) (at 25 degrees Celsius ambient temperature) of 90 degrees Celsius maximum (see SunnySky footnote).
8. Propulsion system maximum electrical load is below the load that can be provided by the battery with a minimal margin of 20 C. 45 C used (indicated by eCalc, see footnote) of the 65 C available [3].

For the previous list the following should be mentioned about numbers 6, 7 and 8. During the design process of the propulsion system discussed in chapter 7, the online tool eCalc is used as a verification tool. Besides information on multi-copter flight performance, eCalc also provides an indication on the maximum load (of the entire drone), maximum motor temperature and maximum motor electrical power that the propulsion system might encounter during maximum performance settings (100% throttle). Although these numbers are hard to validate, they do give an indication of the expected conditions. Therefore, attention was paid to keeping these parameters under their limits (given by the respective manufacturers) with a considerable margin. All in all, this approach should provide a reliable system, although it does lead to a possibly over-designed system as well.

[1] https://ecalc.ch/xcoptercalc.php, (Retrieved June 22, 2020)
[2] https://sunnyskyusa.com/products/r2305, (Retrieved June 22, 2020)
[3] https://www.banggood.com/nl/Tiger-Power-14_8V-3600mAh-65C-4S-Lipo-Battery-XT60-Plug-for-RC-Model-p-1583272.html?rmmds=detail-left-hotproducts__2&cur_warehouse=CN, (Retrieved June 22, 2020)

13.2. Availability Characteristics

Availability of the components is determined by the ease, at which they can be acquired. The drone's components have been divided into different groups.

1. Electrical components, with the exception of the PCB board, consist of off-the-shelf components, which can easily be ordered online worldwide.
2. The carrier board PCB can be ordered from a PCB manufacturer, who can also assemble the PCB. The carrier board PCB has characteristics that are within the capabilities of PCB manufacturers. [4] [5] [6] These large-scale PCB manufacturers have a turnaround time of 24 hours or less. They provide boards in quantities as low as 1-5 units.
3. Frame, propeller guards and the landing gear can be 3d printed. There exist 3D print shops at many technical universities, but it is also possible to order a 3D printed part according to a CAD file online. Those are then be shipped to the location of assembly.
4. 2D machining is not required. Only access to a sharp and tough hole drilling tools for carbon fibre is required.
5. 3D printing filaments can be ordered online.
6. Arms and motor mounts consist of components, which can be produced with minimum effort after buying the required products; e.g. requires cutting of a tube or drilling.

13.3. Maintainability Characteristics

It is important to clearly outline what actions have to be done with regard to maintainability of the drone. Following these procedures will prevent failures. These procedures will be discussed in this section

1. The arms, motors, and the battery can all be disconnected without electrical tools and replaced whenever damaged.
2. The shell can easily be removed to access inner components, which can in turn be replaced.
3. The structure is designed to be modular, thus, single components can be replaced without the need to replace the whole drone.
4. The structure of the drone (carbon fibre and 3D printed parts) should be stored in a dark environment and contact with UV radiation minimized as much as possible during operations. This radiation can degrade the materials over time.
5. The 3D printed parts should not be subject to temperatures above 170° Celsius.
6. The propellers can be replaced without electronic tools.
7. The user is required to check and clean dust accumulation from the cooling fan of the Xavier NX once every 100 hours of flight.
8. The user is required to clean the lens of the camera before every flight.
9. In non-operational conditions, the batteries are required to be disconnected from the drone and stored in a fireproof container.
10. When stored, the leads of the batteries have to be isolated from each other and the environment using insulation material, such as tape or a plastic cap.
11. The batteries have to be stored at room temperature, and in a dry environment.
12. The batteries have to be stored at a specific voltage, which can be done by setting the charger to 'storage' mode.

In order to ensure that the drone can perform up to its full potential, maintenance and check ups should occur frequently. The first check should occur post-production, after which all components should be checked separately, to find flaws as early on as possible. A similar check should occur post-assembly but in this case, for the entire system. Multiple checks (with regard to flight performance) should occur to prove the drone is able to perform to intentions. Examples of this can be running a system test which performs load tests on both the computer and the propulsion components, data of which is evaluated afterwards. Before every race, a check-up should occur, similar in nature to the post-assembly check,

[4] https://jlcpcb.com/capabilities/Capabilities, (Retrieved June 19, 2020)
[5] https://www.pcbway.com/capabilities.html, (Retrieved June 19, 2020)
[6] https://www.eurocircuits.com/pcb-assembly-proto/, (Retrieved June 19, 2020)

in order to prove the drone is ready and nothing occurred between assembly and race setup stage. After the race another check for damage should occur. First the drone shall be inspected for external damage. After this check is passed, the internal components should be inspected both visually and by using volt and current meters. This should give insight into the electrical performance of the drone's circuit and could potentially spot flaws before the next race. Any software related system malfunctions should be detected during the race or when reviewing the drone's memory post-race. This can also be achieved by doing system tests in which all systems and their connections are checked. Whenever the drone is not racing for a couple of months, it would be advisable to do a quarterly inspection, similar in method to the post-assembly check mentioned earlier. Results of all checks, tests and inspections should be documented and kept with the drone.

13.4. Safety Characteristics

The following list describes the safety features and procedures necessary to ensure safe utilisation of the drone, during its lifetime.

1. To protect the electronics on board, power is not to shut off immediately, but through the means of capacitors slowly decreasing the current flow.
2. A redundant receiver is included in the drone. Its purpose is to act as a 'kill switch' to shut off power to the propulsion system if the drone were to become a danger.
3. The users are required to ground themselves using an anti static wrist band before touching any PCB to prevent damage from static electricity.
4. The user is required to route the cables and antennas such that there is no danger of being struck by the cooler fan.
5. The user is required to prevent cables from lying on top of any PCB components as the PCBs can reach high temperatures.
6. The carrier board is equipped with power logic that performs a shutdown sequence if the voltage becomes too low, or if the power is suddenly disconnected. This is to prevent a hard shutdown, which may cause data to be lost or corrupted.
7. The carrier board is equipped with Zener diodes, which prevent overvoltage.
8. The carrier board is equipped with decoupling capacitors to prevent jumps in voltage that may cause damage to its components.
9. The structure uses propeller guards to prevent damage of the environment by the propellers spinning at high speeds.
10. The structure uses propeller guards, landing gears and extra springs to protect the on-board electronic systems from g-loads above 50g in a 7.7m/s crash.
11. The drone arms can withstand a crash of 30m/s in axial compression loading.
12. The drone can be caught safely with a hand from beneath with rotors spinning.
13. The 3D printed parts (housing the battery as well) are flame retardant according to UL94 V0 and 5VA ratings [7].
14. For further protection of the environment, a barrier in form of a net in between is recommended.

In the design process, the redundancy philosophy has been taken into account. This means that the design approach used the 'safe-life' or 'fail-safe' ideology. The 'safe-life' method is based on designing the part such that it does not fail within a certain spectrum, for example, within a specified load range and time, after which it needs to be replaced. The 'fail-safe' method uses redundancy to cope with possible failures. This means that, for example, two lightweight solutions are implemented and if one of them breaks it can easily be spotted and replaced. Before it can be replaced, the second solution can provide the necessary function.
The AIgle drone is very sturdy and is designed to protect the on-board systems in different crash scenarios. Besides that, it is also designed such that the injuries to the environment are minimized.

[7] https://www.3d4makers.com/products/pei-filament, (Retrieved June 12, 2020)

14

Product Verification and Validation

It is important to investigate what requirements are complied with after the design has been finalized. The requirements originally set up during the Baseline Report [**BaselineReport**] are compared to the drone's properties and capabilities in this chapter. First of all, in section 14.1 the compliance matrix is given with the requirements and whether they are complied with. Secondly, a sensitivity analysis is performed in section 14.2 to evaluate compliance with the requirements in case some parameters change.

14.1. Compliance Matrix

In order to summarise which requirements the design meets, a compliance matrix is presented in Table 14.1. In the list of requirements, the identifying label and requirement description is given. Furthermore, it is indicated whether the requirement is a customer requirement and whether it is complied with. Lastly, in the note column, the related section or subsection is referred to. Some requirements have been noted as unverifiable requirements. These requirements have been established during the earlier design phase and not enough information about the system is currently known to perform verification. However, post-DSE tests are established, in which system tests are performed to still verify these requirements, as described in section 15.4. Furthermore, earlier established requirements that are related to a specific design choice and which are not applicable anymore have been noted with N/A.

Table 14.1: Compliance matrix

Label	Requirement	Customer	Compliance	Note
AIG-TE-HW				
-E1	The drone shall be outfitted with a cooling system that prevents any component from being heated to temperatures higher than their respective operational limit.		✓	subsection 8.3.1
-L1	The drone shall be able to be used for 70 flight hours total before a component needs to be replaced, excluding replacement of components damaged due to accidents.		✗	Unverifiable
-MP-DR1	All data stored on the drone before a malfunction occurs shall be recoverable.		✓	subsection 8.3.6
-MP-IR1	The drone shall suffer no damage to on-board systems or structure if the drone is dropped from 3 meters onto a concrete floor.	✓	✓	section 9.10
-MP-IR2	The drone shall suffer no fracture, plastic deformation or loss of electronic function if the drone flies into a gate at maximum flight speed.	✓	✗	section 9.10
-MP-MO6	During normal operations,the RPM of the propellers shall not exceed 270,000/diameter [inch].		✓	section 7.6
-MP-MO7	If the drone loses functionality, it shall automatically switch to safety mode within 0.5 seconds.		✗	Unverifiable

-MP-SM-SDS1	The system shall be able to shut down with in 0.1 seconds of being ordered to shut down.		✗	Unverifiable
-MP-SM-SDS2	The system shall be able to be turned off manually.		✓	section 8.4
-MP-SM-SDS4	The system shall turn itself off within <0.5> seconds of detecting a malfunction.		✗	Unverifiable
-MP-SM-SDS4.1	If the system detects the batteries contain less than 15% of battery charge, the drone shall land within <10> seconds.		✗	Unverifiable
-MP-SM-CP2	The drone shall be switched from autopilot mode to manual control within 0.5 seconds from the order to switch being given.		✗	Unverifiable
-PF1	The drone shall have a maximum speed of at least 30 m/s.	✓	✓	section 7.7
-PF2	The drone shall be able to make a 3 meter radius arc while flying at 10 m/s.	✓	✓	section 7.6
-PF3	The drone shall be able to linearly accelerate at 19.61 m/s^2 from zero velocity.	✓	✓	section 7.6
-LR1	The elastic deformation strain during normal flight conditions shall not exceed <the amount where any of the propulsion units or sensors are offset by more than 10°>.		✓	Equation 9.6.3
-LR2	No plastic deformation of the drone shall occur during normal flight conditions.		✓	section 9.3
-LR3	No failure of the structural components of the drone shall occur during normal flight conditions.		✓	section 9.3
-S-D1	The distance to the environment shall be measured at least 60 times per second.			N/A
-S-D2	The resolution of the distance measurements shall be at most 0.5 meter minus the biggest dimension of the drone in meters.			N/A
-S-D3	The distance from the drone to its environment shall be measured for distances up to 50 meters.			N/A
-S-VE1	The velocity of the drone shall be measured 60 times per second.			N/A
-S-A1	The acceleration of the drone shall be measured 60 times per second.		✓	subsection 8.3.3
-S-O1	The angular rate of the drone shall be measured 60 times per second.		✓	subsection 8.3.3
-S-V1	At least one front facing camera shall be present.	✓	✓	subsection 8.3.2
-S-V1.1	The resolution of the front-facing camera shall be at least equal to the minimum required for gate detection at maximum gate distance with the used field of view measured in megapixel.		✓	subsection 8.3.2

-S-V1.2	The field-of-view of the front-facing camera shall be at least 90 degrees.		✓	subsection 8.3.2
-S-V1.3	The front-facing camera shall capture images at 60 Hz or greater.	✓	✓	subsection 8.3.2
-P-E1	The battery shall have a begin-of-life capacity of at least equal to (100% throttle power plus 100% computer power plus 100% power of all other systems) times 60 seconds plus 100% computer power times 120 seconds.	✓	✓	section 7.6
-P-E2	The maximum battery discharge rate shall be at least equal to the maximum use, meaning full throttle, full computational power and full power on all other systems in Watt.		✓	section 13.1
-CO1	The latency of communications between the drone system and remote computers shall be no greater than 80 milliseconds.		✗	Unverifiable
-CO2	The range of remote communications to the drone shall be at least 300 meters.		✗	subsection 8.3.6
-CO3	The packet loss rate of remote communications shall be no greater than 5 % during normal flight conditions.		✗	Unverifiable
-CO4	The rate of data sent from the drone to remote computers shall be at least equal to the data rate of the essential systems, namely the IMU, cameras, and other sensors during normal flight conditions.		✗	subsection 8.3.6
-CO5	The drone shall have manual take-over capability.	✓	✓	subsection 8.3.6
-CO6	The drone shall have a telemetry option.	✓	✓	subsection 8.3.6
-C1	The RAM of the drone system shall be at least 500 MB.		✓	subsection 8.3.1
-C2	The memory storage of the drone system shall be at least equal to the average flight time provided by the battery times the data rate of all system elements times 10 flights in GB.		✗	subsection 8.3.6
-C4	The drone shall use on-board computation only.	✓	✓	subsection 8.3.1
-C5	The drone shall use the 'Betaflight' program as attitude controller.	✓	✓	section 8.2
AIG-TE-EB				
-M1	The total mass of the drone shall be no greater than 1 kg.	✓	✗	section 11.2
-M1.1	The total mass of the propulsion unit subsystem shall be no greater than 115 g.		✗	section 11.2
-M1.2	The total mass of the electronics subsystem shall be no greater than 420 g.		✗	section 11.2
-M1.3	The total mass of the structures subsystem shall be no greater than 300 g.		✗	section 11.2

-M2	The drone shall be able to supply enough power to all subsystems for them to all function simultaneously.		✓	section 11.2
-M3	The maximum drone dimension in all length, width and height, shall be smaller than 50 cm.	✓	✓	section 9.10
-M4	The drone shall have enough computational power to process images at 60 Hz and calculate a flight path.		✓	subsection 8.3.1
AIG-TE				
-A1	The drone shall have lights visible from every angle.		✓	subsection 8.3.7
-A2	The drone lights shall emit at least 2.5 W.		✗	subsection 8.3.7
AIG-TE-P				
-M-M1	All parts of the drone shall be manufacturable using 2D milling, 3D printing and Printed Circuit Board printing machines.	✓	✓	section 9.4
-M-M2	All parts of the drone shall be able to be manufactured independently.		✓	section 12.1
-M-A1	It shall be possible to assemble the drone using only standard drone assembly techniques.		✓	section 12.1
-R1	Each structural component of the drone shall be replaceable within 2 minutes.	✓	✗	Unverifiable
-R2	Each component of the drone shall be able to be replaced individually.	✓	✓	section 9.10
AIG-NT-R				
-TE1	The drone shall be designed by a team of 11 people.		✓	-
-T1	The drone shall be design in detail within 10 weeks.		✓	-
-T2	The team shall work on designing the drone full time.		✓	-
AIG-NT-F				
-D1	The total cost of the drone, including manufacturing, shall be no greater than 2500 euros.	✓	✓	section 15.3
-P1	The total cost of printing during the design process shall be no greater than 62.50 euros [44].		✓	-
AIG-NT				
-S1	The rated lifespan of components shall be at least 2 years.		✗	Unverifiable
-S2	The components of the drone shall be interconnected using temporary joints.		✗	section 11.5
-S3	The amount of waste material during the final production phase shall not exceed 3 kg.		✗	Unverifiable
-S4	The sustainability of the drone shall be maximized.	✓	✓	section 11.5
-U1	The drone shall have a 'safety mode', which, when activated, attempts to prevent hazards and incidents.		✗	Unverifiable

-U2	The drone should adhere to European fire safety regulations.	✗	Unverifiable
-U3	All materials used in the drone shall comply with the RoHS 1 and RoHS 2 restrictions on hazardous substances.	✓	section 11.5
-U4	The drone can be hand-held safely while the rotors are spinning.	✓	section 9.10
-R1	The maximum speed of the drone shall be lower than 44.7 m/s [45].	✓	section 7.7
-R2	The drone shall have an 'arming switch', which can be used to allow the drone to move under its own power.	✓	subsection 8.3.6

Requirements AIG-TE-HW-S-D1 until -S-D3 and AIG-TE-HW-S-VE1 are marked N/A since these are related to depth cameras and optical flow sensors respectively, which have not been incorporated in the design after the midterm trade-off.

AIG-TE-HW-MP-IR2 is not met since only the structure itself can handle the gate crash at maximum speed. The electronics can only survive a lower impact speed of 7.7 m/s, which stems from requirement AIG-TE-HW-MP-IR1. Furthermore,

Moreover, AIG-TE-HW-CO2 can only be met for the receivers to control the drone, as the range is not achievable for WiFi telemetry. AIG-TE-HW-CO4 can not be met as the data rate from the camera is larger than the WiFi module can handle. Only lower resolution images can be streamed to the ground station. Also, AIG-TE-HW-C2 can not be met, since the microSD card is limited in write speed. Due to this, the images from the camera can only be stored on the microSD card at a lower resolution. In future work, the carrier board can be upgraded to have an M.2 SSD, which would allow for compliance with this requirement.

Finally, the mass requirements AIG-TE-EB-M1 until AIG-TE-EB-M1.3 are not met. The final mass of the drone is 1.3 kg, and each of the subsystems is also heavier than is required. As mass optimisation for electronics already has been performed, and compliance with other requirements is still necessary, the mass requirement of 1 kg cannot be met. Even after performing mass optimisation for the structure of the drone in the future, it is unlikely that the drone will be lighter than 1 kg. However, the drone is equipped with a state-of-the-art computer, which is specialised in AI computing and is capable of processing images at 60 Hz while having a small form factor. Furthermore, a dedicated dampening system is implemented to protect this computer. In addition, the mass of 1.3 kg is a large improvement over the RacerAI, which weighs 3 kg. On top of that, the structure of the drone is made very durable and modular, which improves its sustainability greatly. Therefore, exceeding the mass requirement of 1 kg by 300 grams is considered justifiable.

14.2. Sensitivity Analysis

In this section, the sensitivity of the compliance with all requirements is investigated. It should be noted that most requirements mentioned in section 14.1 which are complied with are not in danger of non-compliance due to changes in design. This is because requirements such as AIG-TE-HW-S-V1.3 are properties of selected components (in this case the camera) and since there is currently no reason to change the camera this requirement will be met regardless.
Requirements related to product price and flight performance are affected by changes in mass and cost estimates. The used mass and cost estimate for certain subsystems together with higher end estimates are given in Table 14.2.

Table 14.2: Actual used mass & cost estimates and their respective higher end estimates

Subsystem	Mass [g]	Higher-End Mass Estimate [g]	Cost [€]	Higher-End Cost Estimate [€]
Propulsion & Power	500	551	107	122
Structures	536	565	299	369
Electronics	250	263	770	876
Total	1286	1379	1176	1367

The higher-end mass estimates are obtained through the following procedures:

- **Propulsion & Power**: The propeller and motor mass do not change, since the component choice is unchanged. However, a heavier battery with more capacity could be used if the required current by the motor and other components is higher in reality, while still taking into account that the flight time requirement has to be met. In addition, for the same flight performance related requirements, an increase in the mass of other components also causes a battery with higher capacity to be selected. By running the matching tool mentioned in section 7.4 for the same motor and propeller, but for the mentioned changed inputs a different battery is selected. In this case, the battery capacity would increase from 3600 mAh to 4000 mAh (4S & same C-rate). This would account for an increase in the mass of ≈50 grams (352 changed to 403 grams). [1]

- **Structures**: Most likely the components remain at the same mass, or even reduce as explained in section 9.11. But for a worst-case scenario, the masses of each component are increased with a small amount of 5% on top. There are two exceptions to that. It is expected that the arms of the drone do not increase in mass because they are already designed with big safety factors in mind. Therefore no additional margin is added. On the other hand, a larger margin of 30% is added for the dampening system, because no exact masses could be found for the standoffs.

- **Electronics**: All electronic components could have a difference in mass than is provided by the manufacturer. For off-the-shelf components, a 1 % margin is taken into account for the mass. For the cables and the carrier board, a margin is used of 10%, since there is relatively more uncertainty in these components. This sums up to a 5% margin on the mass of the electronics subsystem.

On the other hand, the higher-end cost estimate is obtained by applying an additional design factor. The reason for this increase in cost could be additional production costs for specific components such as the PCB. Specifically, a 50% margin was used on the cost of the carrier board, since the costs are very uncertain. Furthermore, an additional €5 to €10 is added to every component to account for increased shipping costs (depending on the size of the component). The result of all these increases, both in mass and cost can be found in Table 14.2.

Evaluating Table 14.2 with regard to the requirements mentioned in section 14.1 it is clear that cost-related requirements are not in danger; the higher end cost estimate is still lower than the maximum price of €2500 with by large margin. Furthermore, looking at the mass, it is also clear that mass requirement AIG-TE-EB-M1 is not met in any case.

Requirements related to structural performance are not in danger of non-compliance. The higher mass estimates do not lead to a decrease in structural performance, in fact, they could potentially strengthen certain parts.

[1] https://nl.aliexpress.com/item/4000247884952.html?spm=a2g0o.productlist.0.0.4d7a160d4fXxOz&s=p&ad_pvid=202005140043488972127260323200022513085_1&algo_pvid=61c87fcc-2202-452a-9a21-3d05b5d91eec&algo_expid=61c87fcc-2202-452a-9a21-3d05b5d91eec-4&btsid=0ab6f82c15894422284988001e6c0d&ws_ab_test=searchweb0_0,searchweb201602_,searchweb201603_, (Retrieved June 22, 2020)

Moving on, the next requirement which is investigated is AIG-TE-HW-P-E1, regarding the flight time. Due to the battery with higher capacity (same C-rate, still 4S and same DoD) requirements AIG-TE-HW-P-E1 and AIG-TE-HW-P-E2 are still met. If it assumed that the motors require the maximum current the ESC would allow (= 45 Amperes) and the I_{dry} is increased from 2 Amperes to 5 Amperes, the effective required flight time required (flight time at max & idle setting combined into one variable) is 62 seconds (for a 4000 mAh battery) instead of the original 63 seconds (for a 3600 mAh battery). The eCalc indicated load is equal to 43 C (of 65 C maximum available) and the flight time in the max setting is at least 66 seconds (of 62 seconds required). Once again it should be stated that these are only an indication, yet since due to their relatively large margin it can be concluded that the requirements are still complied with.

The flight performance-related requirements (AIG-TE-HW-PF1, -PF2 & -PF3) are once again evaluated using eCalc similar to section 7.5 and section 7.6. Using the same motor, propeller, and ESC, but different values for battery mass, payload mass, and structure mass. The eCalc indicated total mass estimates (final iteration estimate and higher-end estimate respectively) went up from 1294 to 1393 gram when using the same inputs for subsystem total masses of Table 14.2. This difference in total mass estimates (between eCalc and Table 14.2) was discussed earlier in section 7.6 and the same argument can be applied here. All in all, the eCalc indicated thrust-to-weight drops from 4.1 to 4.0. Calculating the thrust-to-weight using the own matching tool using the higher-end mass estimate causes a drop from 4.3 to 4.1. This means that it can confidently be said that the requirement AIG-TE-HW-PF2 (translated into a required thrust-to-weight of 4.0 in subsection 7.3.1) is still complied with. For the remainder of this sensitivity analysis, the thrust-to-weight shall equal 4.0 since it is the lower estimate when choosing between eCalc and the own matching tool.

The AIG-TE-HW-PF1 requirement compliance is not affected due to this drop in thrust-to-weight. For a thrust-to-weight ratio of 4.0 the maximum speed is estimated at 31.5 m/s (same method is used as discussed in section 7.7), this value ranges between 27.4-32.0 m/s (for a 95% confidence interval). Thus, although the maximum speed decreases when compared to the final iteration mass estimate, the speed requirement is still complied with.

Finally, AIG-TE-HW-PF3 is still met. A thrust-to-weight of 4.0 exceeds the 2.24 required for compliance (subsection 7.3.2). In the future, it should be investigated how the requirements which are not complied with can be incorporated into the design. This can be achieved through more iterations or by adapting requirements in order to make them acceptable for the clients. It is recommended that if more iterations are to take place the main focus should be on decreasing the mass. This can possibly be done at the expense of the price since there is still a large margin there.

It should be noted that other parameters were also investigated; these include the battery depth-of-discharge and the drag acting on the drone. It is important to note that these values where already over-estimated compared to reality as discussed in chapter 7. The depth of discharge is currently set at 80 % meaning that 20 % of the battery's capacity will not be used. Decreasing the depth-of-discharge to 75 % would indeed negatively affect the requirements, but it means a full quarter of the battery is not used. This is an illogical assumption to make. Furthermore, the drag coefficient is currently equal to 1.0. If this increases by 10 %, to a coefficient of 1.1, the maximum speed for the final iteration mass estimate would drop from 32.1 to 31.6 m/s. Considering the higher end mass estimate in combination a higher drag coefficient (of 1.1), yields a speed of 31.1 m/s (same method as described in subsection 7.3.3). Thus, even in this worst-case scenario compliance is still upheld.

15
Post-DSE Activities

The goal of this chapter is to describe the steps that shall be followed Post-DSE. The project Design and Development logic is presented in section 15.1. A timeline is established and a Gantt chart is shown in section 15.2.

15.1. Project Design and Development Logic

Upon completion of the symposium, a number of steps still need to be completed to ensure the success of the project. Four major steps remain, finalisation of the design, publishing it, developing an ecosystem around it and maintaining the design.

After the initial publication of the results, a final design will then be developed for testing purposes. An initial prototype will be built on the basis of the research published. The necessary components will be ordered, produced, and assembled. The resulting prototype will then be tested rigorously to ensure that all requirements are met through in-depth verification and validation. If needed, a number of iterations will be performed to reach a well optimised design that complies better with the requirements. Results will be recorded and analysed, and the final design will then be published along with the report and testing results.

The publishing of the results and design will be done through a number of channels. A Git repository will be maintained containing the report, the code, 3D models, and blueprints of the project. Forums posts, a website, and articles will also be published to ensure visibility.

Developing an ecosystem around the project will be important. Finding investors and partnerships can be beneficial to ensure wide adoption and success of the project. These partners can help with a number of activities, such as keeping the design up-to-date, and streamlining manufacturing, production, and distribution for larger scale adoption and visibility.

Work will also be necessary in maintaining the design. Updating supplier information and making design update will also be performed, with the possibility of letting the open source community take charge of the process.

All of the above mentioned processes are summarised in Figure 15.1.

15.2. Gantt Chart

To get a better overview of the timeline of above discussed post-DSE steps, a project gantt chart is made.. All the tasks described in Figure 15.1 are included in it, with an estimation of when and how long it will take. The gantt chart can be found in Figure 15.2.

15.3. Cost Breakdown

The cost breakdown structure is describing the costs that occur per activity and sums these up into an overhead cost. This analysis includes the production of one drone, the finalization of the development, the publication and market entry as well as the maintenance and usage of the AIgle drone. The diagram can be seen in Figure 15.3. The costs have been derived from the production cost of one drone with the parts specified in section 12.1. The cost of the PCB final design is estimated based on [1]. The manufacturing cost has been based on estimates provided by [2] .

[1] https://www.911eda.com/solutions/pcb-design-cost, (Retrieved June 20, 2020)
[2] https://3delft.nl/, (Retrieved June 22, 2020)

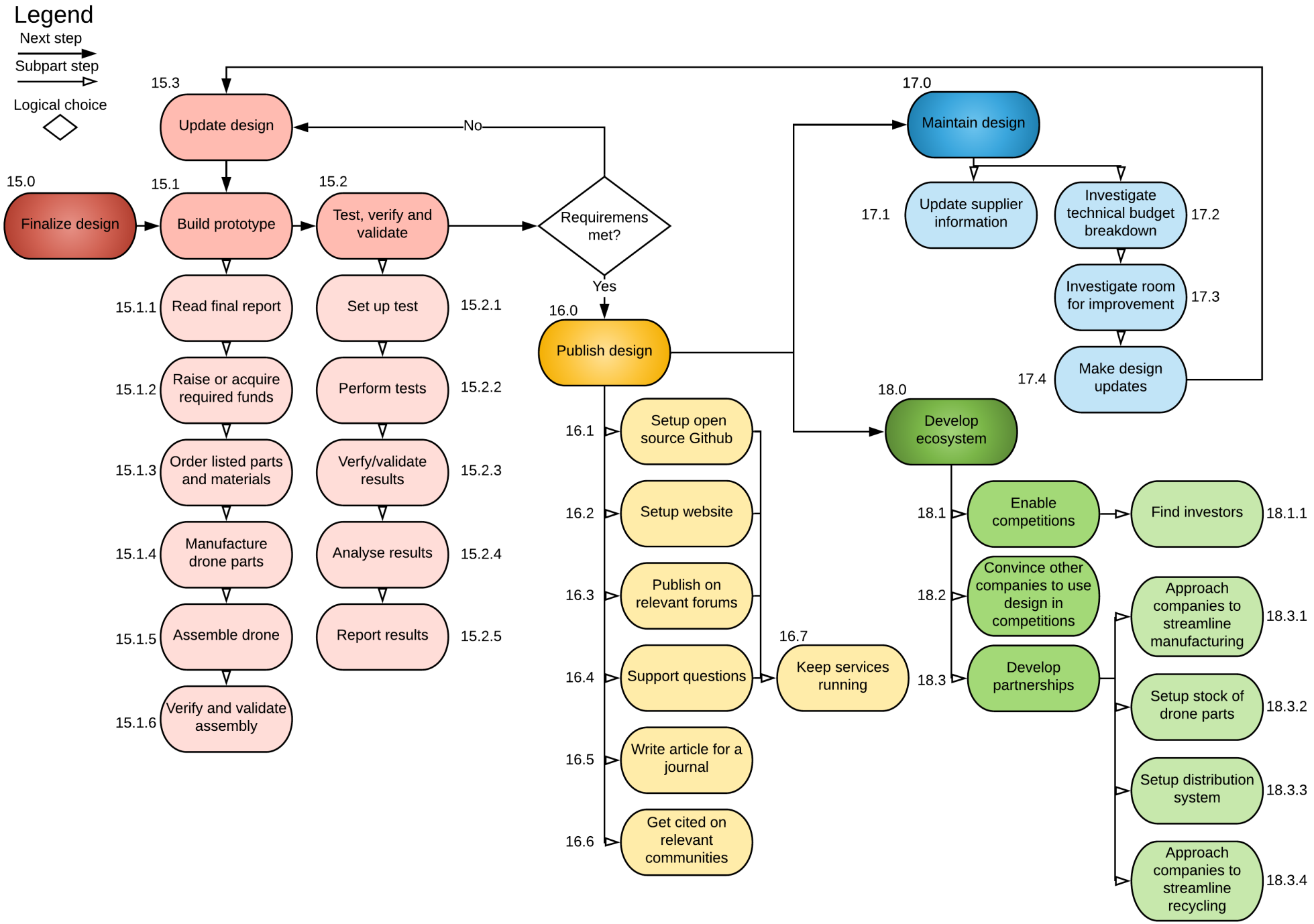


Figure 15.1: The development logic is depicted on this diagram

ID	Task Name	Duration	Start
0	**Project Algle**	**105 days?**	**Fri 3-7-20**
1	**Post -DSE phase**		
2	15.1.1 Read final report	1 day	**Fri 3-7-20**
3	15.1.2 Raise or acquire required funds	12 days	**Mon 6-7-20**
4	15.1.3 Order listed parts and materials	5 days	Tue 21-7-20
5	15.1.4 Manifacture drone parts	5 days	Mon 27-7-20
6	15.1.5 Assemble drone	3 days	Thu 30-7-20
7	15.1.6 Verify and validate assembly	1 day	Mon 3-8-20
8	15.2.1 Set up test	1 day	Tue 4-8-20
9	15.2.2 Perform tests	2 days	Wed 5-8-20
10	15.2.3 Verify/validate results	2 days	Fri 7-8-20
11	15.2.4 Analyse results	3 days	Mon 10-8-20
12	15.2.5 Report results	2 days	Wed 12-8-20
13	16.1 Setup open source Gihub	15 days	Fri 14-8-20
14	16.2 Setup website	15 days	Fri 14-8-20
15	16.3 Publish on relevant forums	5 days	Fri 14-8-20
16	16.4 Support questions	60 days	Fri 4-9-20
17	16.5 Write article for a journal	3 days	Fri 21-8-20
30	16.6 Get cited on relevant communities	3 days	Fri 14-8-20
31	16.7 Keep services running	60 days	Fri 4-9-20
18	17.1 Update supplier information	60 days	Fri 4-9-20
19	17.2 Investigate technical budget breakdown	60 days	Fri 4-9-20
20	17.3 Investigate room for improvement	60 days	Fri 4-9-20
21	17.4 Make design updates	60 days	Fri 4-9-20
22	18.1 Enable competitions	30 days	Fri 14-8-20
23	18.1.1 Find investors	30 days	Fri 14-8-20
24	18.2 Convince other companies to use design in competitions	30 days	Fri 14-8-20
25	18.3 Develop partnerships	30 days	Fri 14-8-20
26	18.3.1 Approach companies to streamline manifacturing	15 days	Fri 14-8-20
27	18.3.2 Setup stock of drone parts	30 days	Fri 14-8-20
28	18.3.3 Setup distribution system	15 days	Fri 14-8-20
29	18.3.4 Approach companies to streamline recycling	30 days	Fri 14-8-20

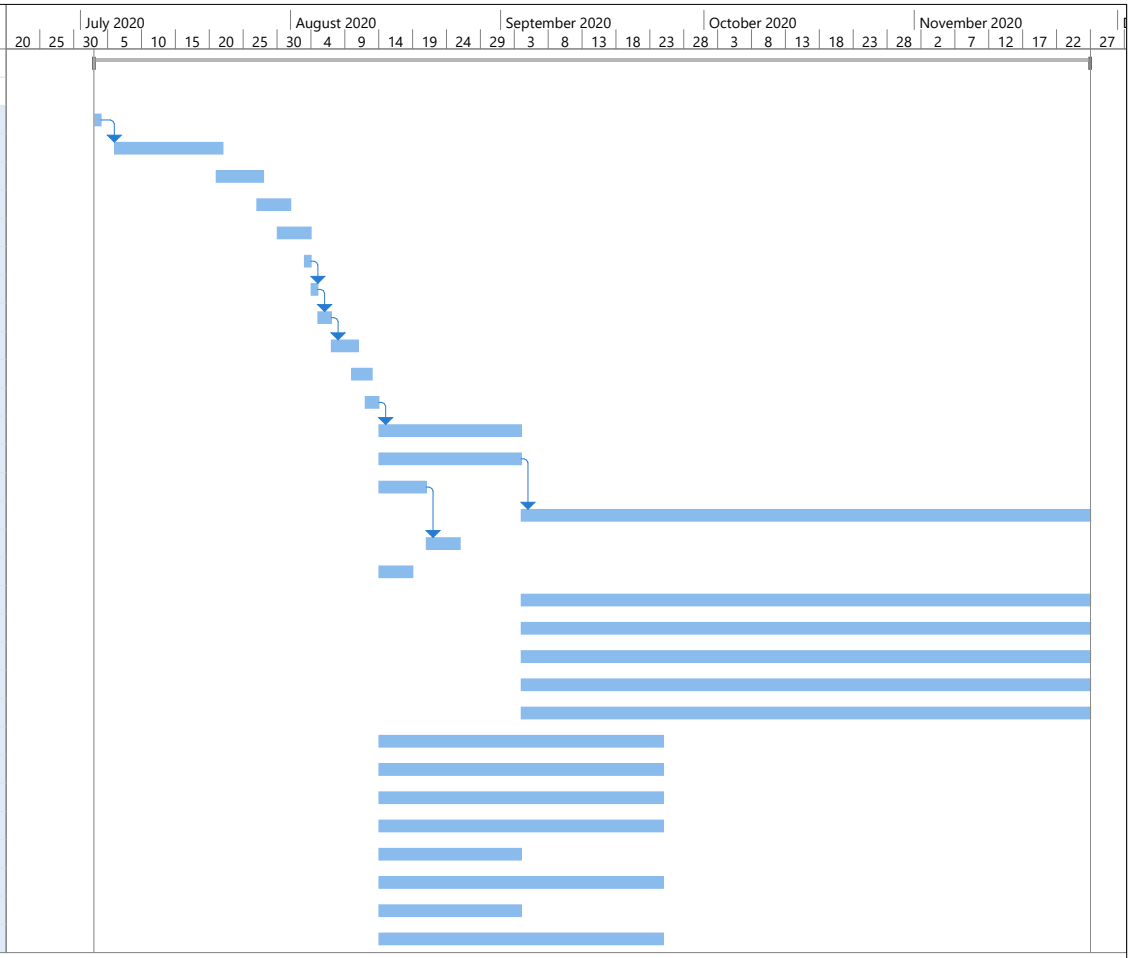


Figure 15.2: The Gantt chart describing the timeline post-DSE

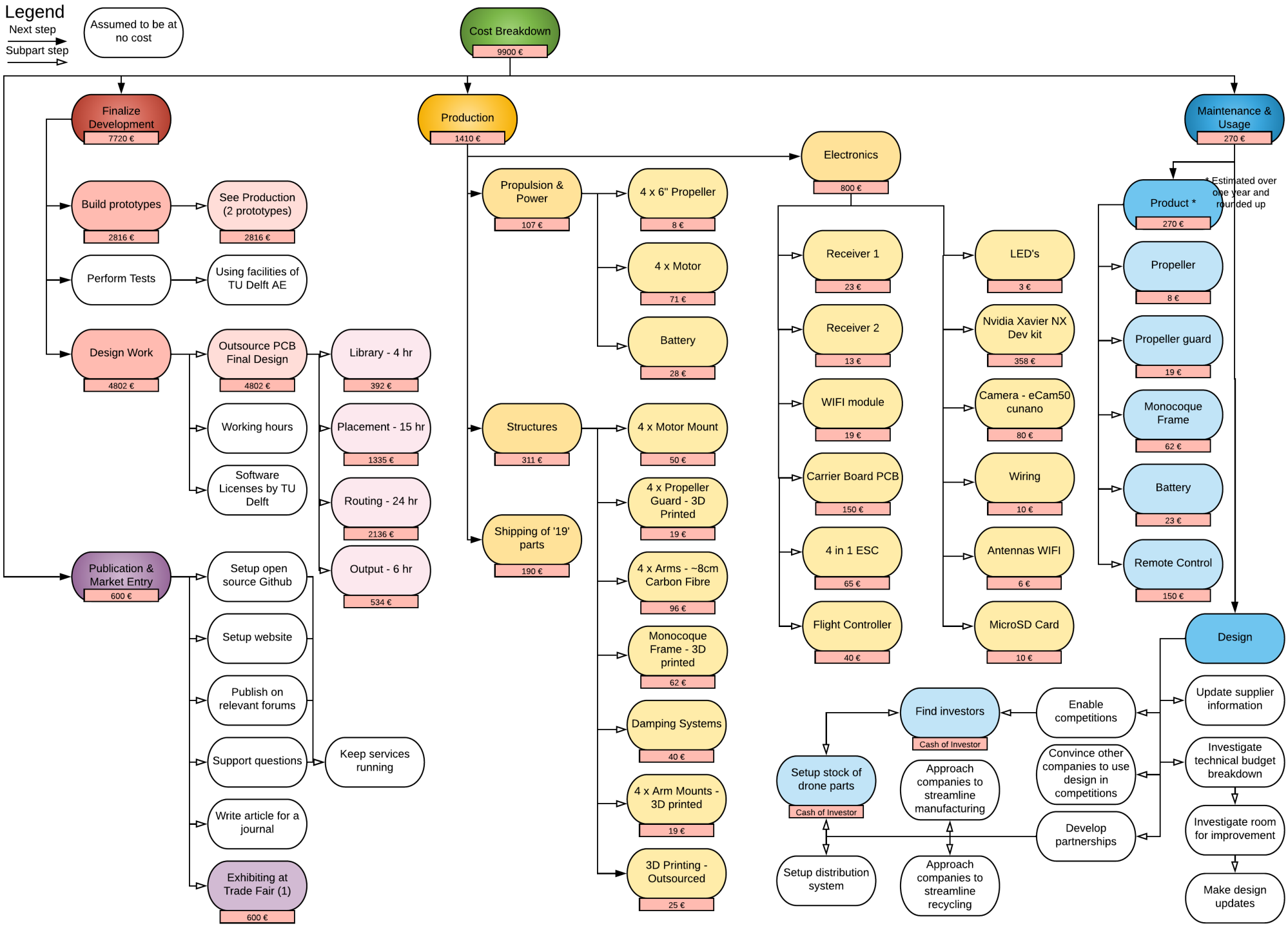


Figure 15.3: Cost breakdown structure of the AIgle project

15.4. Post-DSE Tests

In this section Post-DSE tests are proposed to further check compliance with product requirements (section 14.1), which could effectively validate the design. The objective of the tests is given as well as what parameters/components they could affect.

Table 15.1: Test and demonstration to be conducted on produced drone.

#	Test name	Type	Test objective	Related subsystem(s)	Affected parameter/ component(s)
1	Dynamic thrust test	Demo	Prove the estimated dynamic thrust matches real dynamic thrust	Propulsion	Motor & propeller combination
2	Track completion test	Demo	Prove an entire (real) track can be completed autonomously (proof of concept)	Navigation, Machine Vision & Control	Navigation & Control algorithm Machine vision algorithm
3	Data-rate stress test	Demo	Prove that data-handling architecture can handle maximum incoming data	Software & Electronics	Processing architecture
4	Electric load stress test	Demo	Prove that electronic architecture can handle all loads at maximum power use-age	Propulsion & Electronics	Electrical architecture
5	Cooling stress test	Demo	Prove cooling works properly and check max temperature achieved in stressful settings	Structures, Propulsion, Electronics	Motor, certain structural components & all electronics
6	Telemetry test	Demo	Prove communications are working properly with regard to signal strength, latency period & distance covered	Electronics	Telemetry related components
7	Control test	Test	Prove desired location matches up with given instructions by control (test done in real life environment)	Control	Control algorithm
8	State determination test	Demo	Prove that state can be determined (velocity & orientation) when given at least two time sequenced images (taken with moments from each other in reality)	Machine Vision	State determination of Machine Vision algorithm
9	Gate tracking test	Demo	Prove that a gate is tracked/identified whenever pictures at different angles are provided (using actual on board components)	Machine Vision	Gate tracking of Machine Vision algorithm
10	Gate crash test	Test	Prove structural components can protect internal components during a competition gate crash at high speed (test shall not be conducted with actual internal components on board, but guided through telemetry)	Structures	All relevant structural components
11	Drop test	Test	Prove structural components can protect internal components during a drop from 3 meters (again actual internal components not required for this test)	Structures	All relevant structural components
12	End-to-end system test	Test	Prove that complete data from sensors end up at relevant locations and signals sent out (e.g. to motors) are received and carried out as desired	Electronics	All electronic components: specifically their connections
13	Replacebility test	Demo	Prove components can be replaced within the required time frame	Structures & Electronics	Structural and electronic connections
14	Shut/Swith test	Demo	Prove that the drone can shut off and switch control modes (e.g. safety mode/ manual control) within the required time frame	Electronics & Software	Telemetry units, capacitors, other switch related components
15	Hand-held test	Demo	Prove that the drone can be hand-held safely while all components are fully operational	Electronics & Propulsion	Cooling units distance between body and motors/propellers
16	Endurance test	Demo	Prove that the drone can be operated for 70 hours before replacing parts (crashes are excluded)	All hardware related systems	All components
17	Endurance test	Demo	Prove that the drone can be operated for 70 hours before replacing parts (crashes are excluded)	All hardware related systems	All components
18	Battery forced landing test	Test	Prove that the drone lands within 10 seconds after a battery charge of <15% has been detected	Electronics, Navigation & Control	Voltage meters, navigation/control software
19	Maximum speed	Test	Prove that the drone can achieve the computed maximum speed	Propulsion	All propulsion & power related components

16

Conclusion

The objective of the final report is to present a finalised and detailed design, building up on the work performed in the mid-term and base-line reports. All of the main subsystems making up the drone were investigated in depth, and design choices were made to end up with a coherent and effective final solution. This report also evaluates the logistic, sustainability, and risk related to the concept and plans out the steps which will follow upon the end of the DSE.

Upon selecting the Durability concept, work distribution was reviewed to ensure the most effective approach was adopted. The group was divided in six sub-systems, each addressing a critical element of the design. Each subsystem was reassembled under two main branches, hardware and software. Hardware includes Structures, Electronics, and Power & Propulsion, while software includes Machine Vision & State estimation, Navigation, and Control & stability. This two branch system is supervised by two system engineers, each in charge of maintaining consistency in their branch section, and working together to ensure a coherent and unified final design.

To further streamline the development process, and ensure that the design goals are met, the previously determined requirement list was reviewed, and updated in light of the new information made available as the details design progressed.

Hardware first of all ensured every subsystem were up to date with the requirements relevant to their subsystem. Next to this a communication document was setup so that all subsystems could effectively communicate, as all of the hardware systems were very interdependent.

Power & propulsion focused on the selection of motor, propeller and a battery to ensure certain flight maneuvers are possible. This selection procedure was done through an iterative matching exercise which in turn provided the drone with the right components to preform in both static and dynamic conditions. Ultimately, it was chosen to use an APC 6x4E propeller, a SunnySky R2306 (2300 kv) motor and a Tiger Power 4S battery.

The electronics mainly concerned the hardware selection, and how components are interconnected. It is decided to connect the LiPo battery to both a 4-in-1 ESC and the carrier-board, which contains the Nvidia Xavier NX computer. This carrier board was custom-designed, saving at least 26% in mass compared to an off-the-shelf solution. Furthermore, to provide telemetry, there is a WiFi module on the carrier board and there are two receivers present for manual control, where the second one is for redundancy and to act as a 'kill switch'. Moreover, the camera has a sufficient resolution for gate detection and at least 60 Hz. In addition, the flight controller has an F7 processor, allowing for faster processing. Finally, the 4-in-1 ESC supports a digital protocol, allowing for lower latency and thus providing an advantage in racing.

The structures of the drone incorporate the mainframe, the arms as well as protective components and the mounting of the on-board systems. This results in a X-frame, with the forward facing arms spread wider allowing a good camera field of view. The arms of the drone are made out of carbon fibre tubes as well as the motor mounts. The mainframe and protection components are 3D printed. The protection mechanism relies on the deformation of the landing gear and propeller guards, as well as the stand-offs for the processor, carrier board and cooler, which are spring-dampers made out of metal wires.

Software first provided an overview and guidelines on the development strategy future teams using the AIgle racing platform could follow. A general pipeline was developed, and guidelines on the software structure were laid out. The three main subsystems making up software then primarily focused on demonstrating the possible approaches which could be followed for their respective task. Machine Vision & State Estimation focused primarily on retrieving and processing sensor data to get an es-

timate of the position of the drone in 3D space during the race. MV/SE made implementations for various methods. These all share a general layout, where first the images were processed to find the gate corners in the image; information which was then used to find the location of the corners in 3D by using the known size of the gate. It is found that there are very achievable methods of Machine Vision and State Estimation, though different methods vary in accuracy. It is still to be investigated which method yields the best results.

Navigation focused on determining the path and trajectories to be followed by the drone during the race. A large number of methods were considered. An in depth analysis of a selection of more complex Deep Learning reinforcement learning techniques were reviewed, including DDQL and DDPG, and demonstrated. It was concluded that the applications of such methods is not only achievable, but also presents a significant potential for drone racing.

Following the instructions emitted by navigation and translating the desired drone movement into commands is done by control and stability. Commands are then sent to the FC, which sends it to the ESC, which relays them to the motors. The control system is a series of nested loops, which control position, velocities, attitude respectively. The latter determines the angular rates that should be outputted to betaflight. The PID mechanism is used for performing control. Gains are tuned by observing the system response to step inputs and implementing changes accordingly.

The final design with all the design choices was then summarised, ensuring that the work of the different subsystems across software and hardware summed up to a coherent design. The production and operation logistics was then detailed to clearly show how to construct the drone and all subsequent steps involved in the drone lifetime. The information required for optimal use during all phases of its life was summed up. After that the Reliability, Availability, Maintainability and Safety characteristics of the drone were identified, to ensure a robust and safe design. In addition, a maintenance and evaluation procedure for the drone was set up.

Verification and validation of the overall project and design were then performed. Through the compliance matrix it is seen that the final iteration drone design complies with almost all requirements. However, most notably, the mass requirement of 1 kg can not be met with, as the current mass of the drone is 1.3 kg. However, the drone is equipped with a state-of-the-art computer, released in April 2020, which is specialised in AI computing. Furthermore, a dedicated dampening system is implemented to protect this computer as well as the camera, as these are costly components. In addition, the mass of 1.3 kg is a large improvement over the RacerAI, which weighs 3 kg. Therefore, exceeding the mass requirement of 1 kg by 300 grams can be considered justifiable. Furthermore, sensitivity analysis showed that all requirements which were initially complied with are still complied with if the mass would increase or other parameters would change for the worse.

Finally, the post-DSE part of the project was addressed. The steps for prototyping and validating the design were laid out, and the tasks necessary for the completion, production, and adoption of the project where established and scheduled in a Gantt chart.

Bibliography

[1] D. Borstlap, F. Dam, N. Duursma, W. Dziarnowska, V. Guillet, Y. van Haaren, D. Huang, M. van Huffelen, J. Riessbacher, G. Strunck, and R. Vos. *Design Synthesis Exercise: Project Plan*. April 2020.

[2] H. Moon et al. *Challenges and implemented technologies used in autonomous drone racing*. *Intelligent Service Robotics* (2019).

[3] R. Hedayati and J. Sinke. *Production of Aerospace Systems (AE3211-II): 8. Additive manufacturing*. March 2020.

[4] R. Gill and R. D'Andrea. *Propeller Thrust and Drag in Forward Flight*.

[5] *Cylinder in Cross Flow Comparing CFD Simulations w/ Experiments*. Retrieved June 13, 2020. http://www.users.miamioh.edu/sommerad/NSF%5C%20Files/comparison_with_experiments.pdf.

[6] J. Scott. *Drag of Cylinders & Cones*. Retrieved June 13, 2020. http://www.aerospaceweb.org/question/aerodynamics/q0231.shtml.

[7] S. Li et al. *Visual model-predictive localization for computationally efficient autonomous racing of a 72-g drone*. *Journal of Field Robotics* (2020).

[8] M. Miyamasu and K. Akatsu. *Efficiency Comparison between Brushless DC Motor and Brushless ACMotor Considering Driving Method and Machine Design*. *IEEJ Journal of Industry Applications* 2.1 (2012), pp. 79–86.

[9] A. Gong, R. MacNeill, and D. Verstraete. *Performance Testing and Modeling of a Brushless DC Motor, Electronic Speed Controller and Propeller for a Small UAV Application*. *Propulsion and Power of Unmanned Aerial Systems I*. 2018.

[10] D. Laskay. *BLDC Motor Controllers...for Maximum Performance & Efficiency*. Tech. rep. Data Device Corporation, 2015.

[11] C. Zhang et al. *Evaluating the Degradation Mechanism and State of Health of LiFePO4 Lithium-Ion Batteries in Real-World Plug-in Hybrid Electric Vehicles Application for Different Ageing Paths*. *Energies* 10 (January 2017), p. 110.

[12] S. Goldstein. *On the vortex theory of screw propellers*. *Proceedings of the Royal Society of London* 123.792 (1929), pp. 440–465.

[13] F. C. Dougherty et al. *TAIR - a Transonic Airfoil Analysis Computer Code*. NASA. May 1981. https://ntrs.nasa.gov/archive/nasa/casi.ntrs.nasa.gov/19810017538.pdf.

[14] *Data Sheet Nvidia Jetson Xavier NX System-on-Module. External Product Specifications (EPS)*. Version Version 1.3. January 2018.

[15] S. Jung et al. *A direct visual servoing-based framework for the 2016 IROS Autonomous Drone Racing Challenge*. *Journal of Field Robotics* 35.1 (2018), pp. 146–166.

[16] E. Kaufmann et al. *Beauty and the Beast: Optimal Methods Meet Learning for Drone Racing*. *CoRR* abs/1810.06224 (2018). http://arxiv.org/abs/1810.06224.

[17] K. N. Plataniotis and A. N. Venetsanopoulos. *Color Image Processing and Applications*. Springer, 2000.

[18] *Intel Wireless-AC 9560 (Jefferson Peak 2). External Product Specifications (EPS)*. Version Revision 1.7. January 2018.

[19] T. L. Floyd. *Digital Fundamentals*. 11th ed. Pearson, 2014. ISBN: 9780132737968.

[20] B. Horan. *Practical Raspberry Pi*. Apress, 2013. ISBN: 1430249714.

[21] Smiths. *7068 Aluminium Alloy Technical Datasheet* (2018).

[22] J. Floor. *GETTING A GRIP ON THE ULTIMAKER 2 TENSILE STRENGTH OF 3D PRINTED PLA: A SYSTEMATIC INVESTIGATION* (2015).

[23] D. Shinde, L. Emmanwori, and A. Kelkar. *Comparison of mechanical properties of EPON 862/W with and without TEOS electrospun nanofibers in nanocomposite*. June 2014.

[24] R. Hedayati and J. Sinke. *Production of Aerospace Systems (AE3211-II): 2. Part manufacturing - 1*. March 2020.

[25] R. Hedayati and J. Sinke. *Production of Aerospace Systems (AE3211-II): 4. Part manufacturing - 3*. March 2020.
[26] J. Martin. *6 - Composite materials*. *Materials for Engineering (Third Edition)*. Third Edition. Woodhead Publishing, 2006, pp. 185–215. ISBN: 978-1-84569-157-8.
[27] R. C. Hibbeler. *Mechanics of materials*. Pearson, 2018.
[28] S. Turteltaub. *Vibrations (AE2135-II)*. 2017.
[29] S. C. Deac et al. *Modeling and simulation of cars in frontal collision* (2017).
[30] DragonPlate. *1" Carbon Fiber Tangent Tube Mount™ - 12" Long*. Retrieved June 17, 2020. https://app-dragonplate-dev.azurewebsites.net/1-carbon-fiber-tangent-tube-mount-12-long.
[31] T. Serksnis. *Designing Electronic Product Enclosures*. Springer, 2018. ISBN: 3319693956.
[32] J. M. Hoffman. *Shock, awe, and vibration*. Retrieved June 5, 2020. https://www.machinedesign.com/news/article/21817007/shock-awe-and-vibration.
[33] L. Quan and Z.-D. Lan. *Linear N-Point Camera Pose Determination*. *Transactions on Pattern Analysis and Machine Intelligence* 21 (8) (1999). https://hal.inria.fr/inria-00590105.
[34] E. Eisemann. *TI3135TU Visual Data Processing*. 2019.
[35] C. D. Wagter et al. *Autonomous Flight of a 20-gram Flapping Wing MAV with a 4-gram Onboard Stereo Vision System* (2014).
[36] G.C.H.E. De Croon, et al. *Random sampling for indoor flight.* (2010).
[37] *An analytic solution for the perspective 4-point problem*. *Computer Vision, Graphics, and Image Processing* 47.1 (1989), pp. 33–44. ISSN: 0734-189X. https://hal.inria.fr/inria-00589990/file/Horaud-P4P-CVPR89.pdf.
[38] H. van Hasselt, A. Guez, and D. Silver. *Deep Reinforcement Learning with Double Q-learning*. 2015. arXiv: 1509.06461.
[39] T. P. Lillicrap et al. *Continuous control with deep reinforcement learning*. 2015. arXiv: 1509.02971.
[40] S. Shah et al. *AirSim: High-Fidelity Visual and Physical Simulation for Autonomous Vehicles*. 2017. eprint: arXiv:1705.05065. https://arxiv.org/abs/1705.05065.
[41] D. W. Mellinger. *Trajectory Generation and Control for Quadrotors*. Publicly Accessible Penn Dissertations, Retrieved June 20, 2020. 2012. https://repository.upenn.edu/edissertations/547.
[42] D. Sunter et al. *The manufacturing energy intensity of carbon fiber reinforced polymer composites and its effect on life cycle energy use for vehicle door lightweighting*. *20th International Conference on Composite Materials*. 2015.
[43] T. G. Gutowski et al. *The energy required to produce materials: constraints on energy-intensity improvements, parameters of demand*. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences* 371.1986 (2013), p. 20120003.
[44] C. de Wagter. *Project Guide Design Synthesis Exercise Open-Source Hardware Artificial Intelligence Robotic Racing Drone for Future AI Competitions*. 2020.
[45] F. A. Administration. *Fact Sheet – Small Unmanned Aircraft Regulations (Part 107)*. Retrieved May 3, 2020. https://www.faa.gov/news/fact_sheets/news_story.cfm?newsId=22615.