Swarm of Drones for Aircraft Maintenance Inspection

Final Report

Group 02

Technische Universiteit Delft

TUDelft

This page was intentionally left blank.

Swarm of Drones for Aircraft Maintenance Inspection

Final Report

by

Group 02

Student Name	Student Number
Aras AKAR	4681843
Justas Andriuškevičius	5062632
Stepan Baidachny	4999940
Lennart Bult	5010691
Wing Chan	5002265
Servaas Clerckx	5081645
Maurice de Bruin	5082250
Andrea Stanziale	5020751
Seppe Van den Bergh	4998758
Andreas Verbruggen	4498933

Tutor:	A. Bombelli and B. F. Santos
Coaches:	C. Nastos Konstantopoulos and M. Morão Patrício
Institution:	Delft University of Technology
Place:	Faculty of Aerospace Engineering, Delft
Project Duration:	April, 2022 - June, 2022

Preface

About 10 weeks ago, this project group undertook the assignment of designing a swarm of aircraft maintenance inspection drones. Having arrived at the end of this DSE, we would like to proudly present this report before you, which represents the essence of the hard work that this group dedicated to this project.

The list of people whom we need to dedicate our thanks to is extensive. But first and foremost we have to give thanks to our tutors Dr. Alessandro Bombelli and Dr. Bruno Santos, whose sage advice and astute guidance in times of indecisiveness were invaluable, especially to young burgeoning engineers such as ourselves. Next, we are much obliged to the vigilant efforts from our coaches, Maria Morão Patricio and Dr. Christos Nastos Konstantopoulos. With our tutors and coaches, we were always able to engage in conversations that were both fun and insightful to the project. Whether it be during a formal midterm review, or an informal weekly status meeting.

We would also like to mention our external advisors, Vis Dhanisetty and Jasper Steringa from NLR, who were able to provide us with indispensable knowledge and insights about aircraft inspections. Without their expertise, our project would have encountered much more difficulty than it did. Finally, this project would not have been complete without the help from a long list of internal and external experts that gave our group consult: Dr. Roger Groves from TU Delft , Bas de Glopper from KLM, Cassio Wallner and Matheus Fessel from Embraer, Dejan Borota from Mainblades.

Sincerely from DSE Group 2,
Delft, 21st June 2022

Nomenclature

“CHIP” drone	large drone for inspection of upper surfaces
“DALE” drone	small drone for inspection of lower surfaces and hard to reach areas
ABS	Acrylonitrile Butadiene Styrene
AI	Artificial Intelligence
AR	Augmented Reality
ATE	Absolute Trajectory Error
BER	Bite Error Rate
BLDC	Brushless DC Motor
CAGR	Compound Annual Growth Rate
CFRC	Carbon Fiber Reinforced Composites
CBS	Conflict Based Search
CG	Centre of Gravity
CNN	Convolutional Neural Network
COTS	Commercial-Off-The-Shelf
CPI	Cost Per Inspection
CPU	Central Processing Unit
CV	Computer Vision
DNG	Digital Negative
DOT	Design Option Tree
DPX	Digital Picture Exchange
DSE	Design Synthesis Exercise
EASA	European Union Aviation Safety Agency
EIRP	Effective Isotropic Radiated Power
EM	Electromagnetic
EOL	End of Life
ESC	Electronic Speed Controller
FAA	Federal Aviation Administration
FBS	Functional Breakdown Structure
FC	Flight Controller
FFBD	Functional Flow Block Diagram
FIFO	First In First Out
FOM	Figure of Merit
FOV	Field of View
FO	fleet operator
GB	Gigabytes
GVI	General Visual Inspection
IC	Integrated Circuit
IMU	Inertial Measurement Unit
INDI	Incremental Nonlinear Dynamic Inversion
JPEG	Joint Photographic Experts Group
LI/I-SLAM	LIDAR/Inertial Simultaneous Localisation and Mapping
LIDAR	Laser Imaging, Detection, And Ranging
LiPo	Lithium Polymer
MB	Megabytes
MCU	Micro Controller Unit
MEMS	Micro-Electromechanical System
ML	Machine Learning
MNS	Mission Need Statement
MOE	Maintenance Organisation Exposition
MO	maintenance operator
MPC	Model Predictive Control
MRO	Maintenance, Repair, and Overhaul
MTOM	Maximum Take-Off Mass
MTSP	Multi Travelling Salesmen Problem
MTTF	Mean Time to Failure
MTTR	Mean Time to Repair
NDT	Non-Destructive Testing
NN	Neural Network
OEM	Original Equipment Manufacturer
ORCA	Optimal Reciprocal Collision Avoidance
PCB	Printed Circuit Board
PC	Polycarbonate
PDB	Power Distribution Board
PD	Proportional-Derivative
PID	Proportional-Integral-Derivative
PLA	Polylactic Acid
PNG	Portable Network Graphics
POS	Project Objective Statement
PP-GF30	Glass Fibre Reinforced Polypropylene
PWM	Pulse Width Modulation
RAMS	Reliability, Availability, Maintainability and Safety
RA	regulation authorities
RGB-D	Red, Green, Blue-Depth
RGB	Red Green Blue
RMSE	Root Mean Squared Error
RMS	Root Mean Squared
ROI	Return On Investment
RPM	Revolutions per Minute
SD	Secure Digital
SLAM	Simultaneous Localisation and Mapping

SLA	Service Level Agreement
SOFT-SLAM	Stereo Odometry and Feature Tracking Simultaneous Localisation and Mapping
SPI	Serial Peripheral Interface
SPL	Sound Power Level
SWOT	Strengths, Weaknesses, Opportunities, Threats
TIFF	Tagged Image File Format
TRL	Technology Readiness Level
T/W	Thrust-to-Weight Ratio
UART	Universal Asynchronous Receiver-Transmitter
USB	Universal Serial Bus
VI-SLAM	Visual Inertial Simultaneous Localisation and Mapping
VO	Velocity Obstacles
A-INDI	Adaptive Incremental Nonlinear Dynamic Inversion
GBV	Ground-Based Vehicle
EKF	Extended Kalman Filter
FFBD	Functional Flow Block Diagram

List of Symbols

Latin Symbols

$\dot{x}$	Velocity in x direction	$[\mathrm{m/s}]$
$\dot{y}$	Velocity in y direction	$[\mathrm{m/s}]$
$\dot{z}$	Velocity in z direction	$[\mathrm{m/s}]$
$\omega_{prop_{hov}}$	Rotational speed of the propeller during hovering	$[\mathrm{rad/s}]$
ω_{prop}	Rotational speed of the propeller	$[\mathrm{rad/s}]$
v	Velocity vector	$[\mathrm{m/s}]$
$A(f)$	A-weighing correction factor	$[\mathrm{dB}]$
AR	Aspect ratio	$[-]$
a	Acceleration	$[\mathrm{m/s^2}]$
A	Aircraft surface area	$[\mathrm{m}^s]$
A	Area of the propeller disk	$[\mathrm{m}^2]$
B_{type}	Factor accounting for the drone type	$[-]$
B	The number of blades of the propeller	$[-]$
C_P	Power coefficient	$[-]$
C_T	Thrust coefficient	$[-]$
$C_{battery}$	Capacity of the battery	$[\mathrm{Ah}]$
C_d	Distance cost of the cost function	$[-]$
C_D	Drag coefficient	$[-]$
C_r	Ray function cost of the cost function	$[-]$
C_{total}	Total cost of the cost function	$[-]$
C_v	Velocity cost of the cost function	$[-]$
cg	Centre of gravity	$[\mathrm{m}]$
cmd_{Mi}	Percentage of maximum motor i speed	$[-]$
C	Capacitance	$[\mathrm{F}]$
c	Speed of sound	$[\mathrm{m/s}]$
d_C	Minimum distance to other objects	$[\mathrm{m}]$
$d_{aircraft}$	Distance between the aircraft and the drone	$[\mathrm{m}]$
d_{FS}	Diagonal length of the frame size	$[\mathrm{m}]$
D_{prop}	Propeller diameter	$[\mathrm{m}]$
D	Distance in between pictures	$[\mathrm{m}]$
E	Electrical energy	$[\mathrm{Wh}]$
E	Young's modulus	$[\mathrm{MPA}]$
f_r	Ray function	$[-]$
FOM	Figure of Merit	$[-]$
f	Frequency	$[\mathrm{Hz}]$
G	Antenna gain	$[\mathrm{dB}]$
g	Gravity	$[\mathrm{m/s^2}]$
I_0	No load current	$[\mathrm{A}]$
I_{load}	Current drawn by the load	$[\mathrm{A}]$
I_{motor}	Input current to the motor	$[\mathrm{A}]$
I_{stall}	Stall current	$[\mathrm{A}]$
I_{xx}	Mass moment of inertia with respect to rotation around the x-axis	$[\mathrm{kgm^2}]$
I_{yy}	Mass moment of inertia with respect to rotation around the y-axis	$[\mathrm{kgm^2}]$
I_{zz}	Mass moment of inertia with respect to rotation around the z-axis	$[\mathrm{kgm^2}]$
I	Current	$[\mathrm{A}]$
K_T	Torque constant	$[\mathrm{Nm/A}]$
K_v	Speed constant	$[\mathrm{rad/sV}]$
k_p	Motor's gain	$[-]$
k_q	Torque coefficient	$[\mathrm{Nm/(rad/s)^2}]$
k_T	Thrust coefficient	$[\mathrm{N/(rad/s)^2}]$
k	Overlap ratio between neighbouring pictures	$[-]$
L_1	Base level for noise calculation	$[\mathrm{dB}]$
$L_2 - L_6$	Correction levels for noise calculation	$[\mathrm{dB}]$
l_{damage}	Damage size	$[\mathrm{m}]$
L_{FM}	Fade margin accounting for other losses	$[\mathrm{dB}]$
L_{FS}	Free space loss	$[\mathrm{dB}]$
l_{pix}	Pixel size projected on the aircraft surface	$[\mathrm{m}]$
l_{view}	Picture size projected on the aircraft surface (horizontal or vertical)	$[\mathrm{m}]$
L	Inductance	$[\mathrm{H}]$
M_t	Tip Mach number of the propeller	$[-]$
n_C	The amount of points the camera can see	$[-]$
n_{pix}	Number of pixels (horizontal or vertical)	$[-]$
n	Number of revolutions per second	$[-]$
$O(n_C)$	Overlap function	$[-]$
P_{copper}	Copper power loss	$[\mathrm{W}]$
$P_{hov_{ideal}}$	Ideal hovering power	$[\mathrm{W}]$

P_{HR}	Headroom loss	[W]
$P_{in_{ESC}}$	Power input of the Electronic Speed Controller (ESC)	[W]
$P_{in_{motor}}$	Input power of the motor	[W]
P_{iron}	Iron power loss	[W]
P_{load}	Power consumed by the load	[W]
$P_{out_{motor}}$	Output power of the motor	[W]
P_{out}	Power output of the motor	[W]
P_{req}	Power required to drive the propeller	[W]
P_{RX}	Received power	[W]
P_{TX}	Transmitted power	[W]
P	Power	[W]
p	Rotational speed around x axis	[rad/s]
q_i	i-th quaternion	[−]
q_{w1}	Wind heading	[°]
q_{w2}	Wind elevation	[°]
q	Rotational speed around y axis	[rad/s]
$R_A(f)$	A-weighing	[−]
R_{motor}	Internal resistance of the motor	[Ω]
r	Distance between the observer and the centre of the propeller	[m]
R	Resistance	[Ω]
r	Rotational speed around z axis	[rad/s]
S_{amount}	Score for the amount of camera view points	[−]
$S_{distance}$	Score for the distance	[−]
$S_{horizontal}$	Score for the angle of the camera with the horizontal	[−]
S_{normal}	Score for the normal lines	[−]
$S_{overlap}$	Score for the overlapping	[−]
S_{total}	Total score	[−]
SPL	Sound power level	[dB]
T/W	Thrust to weight ratio	[−]
T_f	Friction torque	[Nm]
t_{exp}	Exposure time	[s]
t_{insp}	Total inspection time	[s]
T_{Mi}	Motor i thrust	[°]
T_{stall}	Stall torque	[Nm]
T	Thrust of the propeller	[N]
T	Time in between pictures (stopping method)	[s]
$V_{applied}$	Applied voltage	[V]
$V_{battery}$	Voltage of the battery	[V]
V_{in}	Input voltage	[V]
V_{motor}	Voltage of the motor	[V]
V_{out}	Output voltage	[V]
V_w	Wind velocity	[m/s]
V	Speed	[m/s]
V	Voltage	[V]
w_{Mi}	Rotational speed of i motor	[rad/s]
X_L	Reactance of an inductor	[Ω]
x_{blur}	Distance travelled during the exposure time	[m]

Greek Symbols

α_{FOV}	Field of View (FOV) angle (horizontal or vertical)	[°]
η_{esc}	Efficiency of the ESC	[−]
η_{motor}	Efficiency of the motor	[−]
η	Efficiency	[−]
ρ	Density	[kg/m^3]
σ_y	Yield strength	[MPa]
τ	Motor’s second order time constant	[−]
ξ	Motor’s damping factor	[−]
ϕ	Drone orientation with respect to x axis	[rad]
θ	Drone orientation with respect to y axis	[rad]
ψ	Drone orientation with respect to y axis	[rad]
$\Delta\theta$	Angle between the camera view and the points it can see	[rad]
α_c	The angle of the camera with the horizontal	[rad]

Executive Overview

Visual inspection is a fundamental procedure to ensure that an aircraft is airworthy. Traditionally, visual inspections have been carried out manually. These visual inspections require a significant amount of time to complete. Currently, time-intensive manual inspections are starting to be replaced by single drone inspections, improving speed, safety, reliability of results, and traceability. The next logical step to explore is the use of a swarm of drones, which could offer additional advantages. They are listed in the list below:

- One drone has a single point of failure. In case of a failure, inspection has to be suspended until the failure has been resolved. When a swarm is used, the failure of one drone does not necessarily end the inspection procedure.
- If a single, relatively big drone is used (e.g. Mainblades[1]) some regions of the aircraft may not be accessible. A swarm existing completely or partially of smaller drones can potentially reach those regions without reducing the performance of the system.
- Increasing the number of drones can allow for work to be parallelised, decreasing inspection time.
- Inspection roles for each drone can more specialised. Alternative inspection techniques such as thermography or laser scanning can be introduced for use in the system.

It should be noted that these advantages are of technical nature. However, it is not immediately clear whether these advantages add value from a commercial perspective. The additional saved time and increased reliability may not outweigh the increased investment costs and complexity for a swarm compared to a single drone. To investigate these aspects, this project aims at designing a swarm of drones for inspection of a commercial aircraft while analysing the advantages and disadvantages of such a concept compared to the use of a single drone. The Mission Need Statement (MNS) of this project has been formulated as

To develop an aircraft visual inspection method, which saves aircraft maintenance time, costs, and effort.

In order to meet this MNS, the Project Objective Statement (POS) has been formulated as

To reduce the cost and time needed for aircraft visual inspection, having a swarm of inspection drones designed by 10 students in 10 weeks.

In order to provide a basis for the design of a swarm of drones, the stakeholder requirements have been collected. A stakeholder is defined as any party which will be affected by, or affects, the operations of the inspection drones. They may pose requirements or restrictions on characteristics, behaviour or operation of the system. The possible stakeholders were divided into four different groups; the first group being the maintenance operator, inspection technicians, and maintenance technicians, combined into maintenance operator (MO); the other three stakeholder groups are the airport (AR), the fleet operator (FO), and regulation authorities (RA). The stakeholder requirements can be found in Table 1. A collection of requirements have been defined as key or driving requirements; key requirements are those that are particularly important for the stakeholders. Driving requirements are those that are expected to have a major influence on the final design. Note that under requirement StRS-MO-01, the aircraft inspection is specified to be performed on a single-aisle narrow-body aircraft. This has been included to provide a specific reference for the design of the aircraft. The final system can also be usable for wide-body aircraft, although not necessarily within the limits of some requirements, e.g. inspection time.

[1] Retrieved from: `https://mainblades.com/` (cited 15 June 2022)

Table 1: Stakeholder requirements including classifier and related system requirements.

Identifier	Requirement	Classifier	Related System Requirement
StRS-AR-01	The system shall have a minimum additional influence on its environment in terms of noise and obstructions		SYS-SAF-06
StRS-FO-01	The system shall have a risk of collision lower than one in a thousand inspections	Key	SYS-REL-07
StRS-FO-02	The system shall be able to finish an inspection in the nominal time in case of a single drone malfunction		SYS-REL-01/02/03, SYS-TIM-04
StRS-FO-03	The system shall cost less than €48 per aircraft flight hour, per aircraft for a fleet operator	Key	SYS-COS-03
StRS-MO-01	The system shall perform a full inspection on a single-aisle narrow-body aircraft	Key	SYS-PER-01/02/03
StRS-MO-02	The system shall consist of a swarm of inspection drones	Driving	SYS-SIZ-03
StRS-MO-03	The system shall operate fully autonomously, after being deployed by a human operator	Driving	SYS-PER-04
StRS-MO-04	The system shall be deployed and monitored by one person with minimal (drone specific) training		SYS-COS-02, SYS-PER-06
StRS-MO-05	The system shall perform the inspection in less than 2 hours		SYS-TIM-03/04/05/06
StRS-MO-06	The system shall perform the specified inspection types at a similar or better level than the current industry standard for drone inspection	Driving,Key	SYS-PER-04, SYS-TIM-04
StRS-MO-07	The system shall be carried in up to 4 suitcases (103x67.5x37.5cm), i.e. to be easily transportable by 2 persons	Driving,Key	SYS-SIZ-01
StRS-MO-08	The swarm of drones shall have a probability of detection equal to at least 90% for structural damages of diameter $>10\,\mathrm{mm}$	Driving	SYS-REL-04/05/06/07
StRS-MO-09	The systems' structures shall be at least 70% recyclable	Driving,Key	SYS-SUS-01/05
StRS-MO-10	The systems' components shall be at least 80% repairable or easy-to-replace	Driving,Key	SYS-SUS-06/07
StRS-MO-11	End-of-life solutions for the drones and their batteries should be provided		SYS-SUS-02/03/04/08
StRS-MO-12	Inspections using the swarm of drones shall be cheaper than current human visual inspections, including depreciation costs from the drones and batteries	Key	SYS-COS-01/03
StRS-MO-13	The individual drones shall have the capability of changing the sensor module during the inspection	Driving	SYS-PER-05
StRS-MO-14	The system shall take a maximum of 10 weeks to reach its final design		SYS-TIM-01/02
StRS-MO-15	The system shall have a cost per inspection lower than €287.5	Driving, Key	SYS-COS-01
StRS-RA-01	The system shall be operable both indoors and outdoors, with wind speeds up to $5\,\mathrm{m\,s^{-1}}$ and visibility conditions of more than $100\,\mathrm{m}$	Driving	SYS-REL-02, SYS-PER-07/08
StRS-RA-02	The system shall be safely deployable in all non-movement ground phases (i.e. at the hangar, gate, storage)		SYS-SAF-01/02/03/04/05

From the stakeholder requirements, system requirements were derived. For the identification of system requirements, tools such as a functional flow block diagram (Functional Flow Block Diagram (FFBD)), functional breakdown structure (Functional Breakdown Structure (FBS)), and requirement discovery tree (Design Option Tree (DOT)) were used [22].

The system requirements were used as the foundation for the generation of design options. For this, a DOT was generated [22]. From the DOT, the concepts that were clearly unfeasible, unpractical, or contained an unacceptable development risk were scrapped. Eventually, five design options remained which are listed below. For more information on the design option generation process, the reader is referenced to the baseline [22] and midterm report [23].

1. Small swarm of visual inspection drones with a high-quality camera and relatively heavy drones. This option consists of three drones, equipped with a Laser Imaging, Detection, And Ranging (LIDAR) sensor. An advantage of this option is that there is a limited need for collision avoidance. However, due to the size of the drones, the drones cannot inspect the underside of the aircraft. This limits the inspection scope.
2. Swarm of visual inspection drones with both high and lower quality cameras. This will be a mixed swarm of high and medium resolution visual inspection drones. The drones equipped with the high quality camera are also larger and heavier than their medium resolution counterparts. There will be two high-resolution inspection drones, which will inspect the upper side of the aircraft. Additionally, there will be three medium-resolution drones which will inspect the lower side of the aircraft. The medium resolution drones will be equipped with a stereo camera sensor, instead of a LIDAR module. Therefore, collision avoidance will be a bit more complex. An advantage of this design option is the broadened inspection scope relative to option 1.
3. Large swarm of visual inspection drones with a lower quality camera and relatively light drones. This design option employs seven medium resolution drones, capable of inspecting the entire aircraft. However, the need for collision avoidance is higher relative to options 1 and 2.
4. A swarm of inspection drones for visual and passive thermography inspection. In this option, seven medium resolution visual inspection drones will be employed for visual inspection of the aircraft. One additional large drone equipped with a thermal camera is added for a thermography inspection of the aircraft. An advantage is the broader inspection scope relative to options 1,2, and 3. However, this advantage is limited, as a complete thermography inspection of the aircraft cannot be completed within the inspection time of 30 minutes. Furthermore, the thermal inspection done relies on a passive heat source, such as the Sun.
5. A swarm of inspection drones for visual and active thermography inspection. This design option employs drones with three different types of payload. Firstly, there are seven drones equipped with a medium resolution camera for visual inspection. Secondly, one larger drone carries a thermal camera for thermographic inspection. Finally, one drone is carrying the heat source, to heat up the aircraft's surface. Some advantages of this option are that this option has the same broad inspection scope as design option 4 and that the thermal inspection drone does not rely on a passive heat source. A disadvantage is that the design of this design option is quite complex, as there is a higher need for a more advanced anti-collision software and there are three types of drones to design.

Based on these design options, a trade-off was performed. A summarised overview can be found in Table 2. The criteria for this trade-off were the ease of deployment, cost, development risk, and the scope of the inspection. During this trade-off, it was found that options 4 and 5 featured quite significant development risks; both passive and active thermography inspection featured a low technology readiness level. Furthermore, due to the high number of drones deployed for options 4 and 5, collision avoidance becomes a concern. Moreover, particularly for option 5, the high number of drone types to be designed poses additional significant design uncertainties. Also, options 4 and 5 scored low on ease of deployment. This is due to the added complexity and weight of the thermal inspection drones, on top of the seven medium resolution visual inspection drones. Design option 1 scored low on inspection scope. This is primarily due to the fact that the high-resolution drones, which are relatively large, cannot go under the aircraft fuselage and wing. Also, option 3 scored low on inspection scope. This is due to the fact that the medium resolution drones do not feature LIDAR sensors and therefore it is difficult to create a 3D model of the aircraft. On the other hand, design option 2 scored high on inspection scope, due to the presence of LIDAR sensors and the ability to inspect the entire aircraft. Next to that, options 4 and 5 scored high inspection scope, due to the ability to detect internal cracks. Finally, the costs of each option were considered. It was found that option 3 was the cheapest option and options 1 and 5 are the most expensive.

Table 2: Final trade-off for all options.

		Ease of deployment	Costs	Inspection scope	Development risk	Final score
Option 1	Score	3.3	2	2.3	3.8	**2.84**
	Colour	GREEN	YELLOW	YELLOW	GREEN	BLUE
Option 2	Score	2.8	3	3.8	3	**3.25**
	Colour	BLUE	BLUE	GREEN	BLUE	GREEN
Option 3	Score	3	4	3	2.4	**3.02**
	Colour	BLUE	GREEN	BLUE	YELLOW	BLUE
Option 4	Score	2.3	3	3.8	1.6	**2.76**
	Colour	YELLOW	BLUE	GREEN	RED	BLUE
Option 5	Score	1.5	2	3.8	1.4	**2.38**
	Colour	RED	YELLOW	GREEN	RED	YELLOW

As a result from the trade-off process, design option 2 was considered the most suitable concept. The final swarm consists of five drones, 2 high-resolution visual inspection drones and 3 medium resolution visual inspection drones, which were estimated to perform the visual inspection of a single-aisle aircraft in 20 minutes. These drones are further referenced as large drone for inspection of upper surfaces ("CHIP" drone), and small drone for inspection of lower surfaces and hard to reach areas ("DALE" drone), respectively. A sensitivity analysis was performed on the trade-off to ensure the robustness of the final concept choice.

This design option was the basis for the detailed design phase. In the detailed design phase, the system has been divided into six subsystem. Underneath, the different subsystems with a description of the performed design and analysis activities are given in the list below.

- Inspection interface and procedure subsystem: the camera for both "CHIP" drone and "DALE" drone to gather the visual data was selected. Data handling procedures according to literature were discussed. This includes possible damage detection algorithms, considerations on training data, time needed for processing, as well as future possibilities for traceability of data.
- Swarm control subsystem: a task generation algorithm based on the 3D model of the aircraft generated by the inspection and data handling subsystem was designed to get the points were pictures need to be taken. Then, a task allocation greedy algorithm was implemented to allocate each inspection point to one drone. Positioning sensors were selected starting from the minimum accuracy and mass requirements, then a Simultaneous Localisation and Mapping (SLAM) was designed for each drone. An A* path planning algorithm was chosen and implemented to determine the optimal path between each inspection point. An Optimal Reciprocal Collision Avoidance (ORCA) algorithm was finally implemented. The algorithms were then tested in a simulation.
- Flight control subsystem: various control methods were evaluated and a trade-off was performed. The most suitable method was determined to be Proportional-Integral-Derivative (PID). An open-source PID implementation, including simulation, was decided to be used, after which verification tests were performed. The controller demonstrates stable behaviour for trajectory following, hovering, and wind disturbance rejection.
- Power and propulsion subsystem: including the selection of the batteries, motors, and propellers. This was done by analysing the thrust and power required of several propellers calculating the Figure of Merit (FOM) of every propeller. The propeller with the highest FOM was selected for each drone. Furthermore, the Revolutions per Minute (RPM) needed to sustain a Thrust-to-Weight Ratio (T/W) of 1 and 2 were estimated. The power required data and the RPM of the propellers served as input for the motor selection. The motor selection was performed based on torque-RPM curves and the power input to the motor that was required to sustain a certain T/W. After the motor selection, the battery capacity could be estimated for a flight time of 30 minutes and the total power consumption of all components on the drone. Finally, an Electronic Speed Controller (ESC) was selected, based on the maximum continuous current.
- Structure subsystem: starting with the size and weight constraints given in the system requirements, and considering additional constraints, the structural material PP-GF30 was chosen and one "CHIP" drone

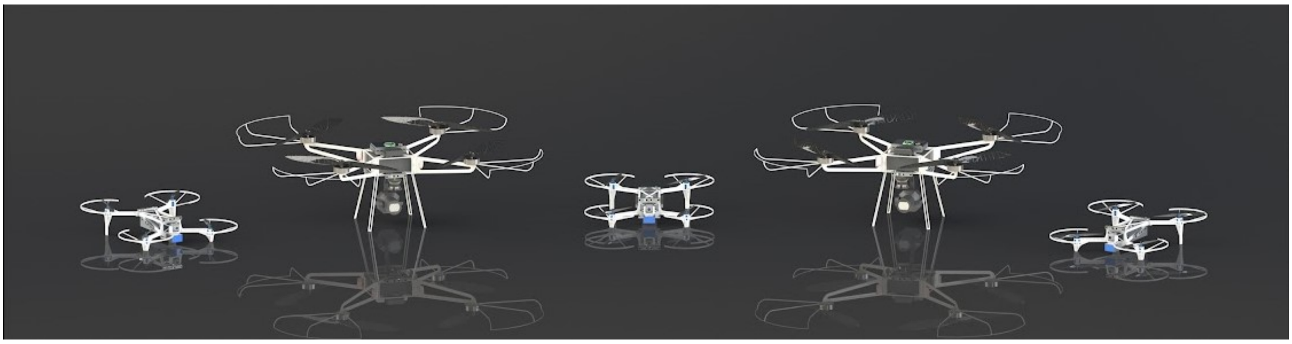


Figure 1: Render of the mixed swarm of inspection drones.

and “DALE” drone were designed. These drones were then topologically optimised to reduce weight, and the final specifications were presented.

- Electrical subsystem: for the required hardware of the different subsystems, an electrical layout has been designed. It included the provision of power and communication capabilities to the different electrical components. The power and communication lines were analysed for capacity. Next to that, voltage regulators have been selected to provide the voltage requirements of the different components.

These subsystems combined resulted in the preliminary detailed design of the two drones’ hardware and software. The visual representation of the swarm is provided in Figure 1. Verification and validation procedures were then performed on all the system and subsystems requirements. A sensitivity analysis was also performed on the propulsion and power subsystem. The system complied with all the set requirements and was found to be able to perform a full visual inspection of a narrow-body aircraft in 16 minutes.

The operations and logistics of the system were also addressed. The system is to be deployed in indoor or outdoor spaces by one operator. After the visual inspection of the aircraft, a digital report based on a 3D model of the aircraft is produced. On this digital report, the locations, types, and sizes of the damages are reported. The damage report aims to improve the traceability and visualisation of damages, making maintenance procedures easier and safer. It was calculated that each swarm of five drones can perform six visual inspections per day.

A market analysis was conducted in order to position the swarm of inspection drones in the current market for aircraft inspection. The design team decided to position itself as a Hardware-as-a-Service provider, supplying the interested customers with the full inspection swarm, together with the training and support needed to operate it. The system was found to be extremely competitive for customers cost wise, saving up to €28 million per year to an airline such as KLM. The system was also found to be potentially profitable for the developing company, with an operational profit of 5.36% in the fifth year of service.

Another fundamental part of the design process, consisted in ensuring that the design and design process were sustainable. Therefore, a sustainable development plan was defined. In order to achieve a sustainable design, both the organisational approach and the product itself took sustainability into account. Regarding the organisational approach, the team did its best to work in a green and sustainable manner. This included using reusable cups and bottles for water and riding the bike or using public transport to arrive at the Fellowship. Sustainable aspects of the product included materials, manufacturing, lean manufacturing, operations, End of Life (EOL) solutions, and some social considerations. For the material selection, the recyclability of the material, the degree to which the material can be used in additive manufacturing, and the durability of the material were taken into account. The chosen material for the structure was PP-GF30, which is recyclable. For the manufacturing process, additive manufacturing was selected for producing the structure. For example, using additive manufacturing, the topology of the structure can be optimised. This helps to reduce material usage as much as possible. Furthermore, the production plan was developed to favour the implementation of lean manufacturing. This be will done by reducing transportation time and by optimising the production of specific parts of the drones, reducing waste. During operations, noise pollution due to the drones will be kept between limits in order to ensure a workable environment for the people involved. Furthermore, it is expected that the drones will be powered by sustainable energy sources. Finally, the end of life of the drone should be postponed as much as possible, by emphasising repairability and interchangeability of the drone system. Drone parts should be reused, or if reusing is not possible, such as for batteries at their EOL then they should be recycled. Finally, from a social point of view, the drones will increase sustainability by making the working environment safer. This safer working environment is achieved by reducing risky activities, such as building the scaffolding around the aircraft.

Contents

1 Introduction

During the last century, aircraft have evolved from relatively simple systems, consisting of only an engine, structural elements, and some rudimentary control surfaces, into complex systems made of millions of components working together to provide essential functions. On top of that, aircraft have to endure extreme conditions that impose considerable strain on multiple subsystems. Some examples are high-pressure differences, bird and debris collisions, wide temperature ranges, and a high number of loading cycles. To assure nominal operations while experiencing this wide variety of operational circumstances, frequent inspections are essential. Commercial aircraft require scheduled visual inspection every 400 to 600 flight hours[1]. Additionally, unscheduled inspections have to be performed following accidents such as bird or lightning strikes. These visual inspections incur significant aircraft downtime, rely on availability of airport and maintenance infrastructure, and require approximately 30 man hours[2]. Introducing drones into the process can lead to significant decreases in inspection time and costs.

Current methods for automated drone inspections of aircraft employ one drone capable of inspecting a certain part of the external surface in an x-amount of time. Although these methods offer significant advantages over current visual inspection, they are not capable of fully catering to customer need. For example, one customer requires inspection at the gate in-between flights, whilst another sticks to a conventional approach of in-hangar inspection during maintenance tasks[3]. Performing a wide range of tasks requires flexibility (e.g. inspection duration, inspection environment); a property swarms naturally posses. Therefore, the aim of this report is to present a swarm of visual aircraft inspection drones that reduces aircraft downtime, infrastructure required, and man hours spent.

A design outline is first provided in chapter 2. The market for the swarm of inspection drone is evaluated in chapter 3 through market definition, positioning, and competitor analysis. The system consists of six subsystems, each interfacing with the other subsystems. Inspection subsystem design is discussed in chapter 4, followed by swarm and flight control design in chapter 5 and chapter 6, respectively. Chapter 7 discusses power and propulsion subsystem design, and is followed by structures subsystem design in chapter 8. Lastly, chapter 9 discusses electrical subsystem design. Chapter 10 provides manufacturing, assembly, and integration plan, as well as an operations and logistics plan of the system. It includes further elaboration on deployment, operating cycles, data handling, and maintenance. The sustainable development strategy is discussed in chapter 11. The cost analysis is discussed in chapter 12, and consists of a cost breakdown, and financial analysis (Return On Investment (ROI), break-even, and profit). A risk analysis, and corresponding mitigation strategy is performed in chapter 13. A Reliability, Availability, Maintainability and Safety (RAMS) analysis is performed in chapter 14. The final design and design compliance matrices are provided in chapter 15. Lastly, a conclusion and recommendations for future work are provided in chapter 16.

[1] Retrieved from: `https://www.aerotime.aero/articles/28347-aircraft-maintenance-abcd-checks` (cited 13 June 2022)

[2] Retrieved from: `https://mainblades.com/save-costs/` (cited 13 June 2022)

[3] Private communication with Embraer, June 2022

2

Design Outline

In this chapter, the general outline of the design of a swarm of drones is provided. First, a definition of a swarm is given and how it applies to drones in Section 2.1. Next, the function of the swarm is discussed in Section 2.2. Lastly, an overview of the swarm configuration is given in Section 2.3.

2.1. Definition of a Swarm of Drones

A swarm is defined as "a large group of insects all moving together"[1]. This definition can be split up into two parts; on the one hand, it requires a number of individuals together. On the other hand, a level of coordination between the individuals is required. Translated to drones, a swarm can be understood to be composed of at least two drones which move in coordination with each other to fulfil a specific function. This interpretation has been used for the design of the swarm of drones.

The designed system complies with this interpretation. It contains several forms of cooperation that make it more than just a group of drones. First of all, path planning is done for the swarm as a whole, taking advantage of the different specialities of different types of drones (see Section 2.3). Secondly, when an individual drone experiences difficulties, its tasks can be redistributed to the rest of the swarm, limiting the impact of a failure. Lastly, the design of the swarm provides the flexibility of experimenting with alternative inspection techniques such as thermography and laser scanning later on. When defects or anomalies are detected on the surface by visual inspection, a dedicated drone with alternative inspection sensors can come in for a closer examination. It should be noted that alternative inspection methods have not been taken into account in the current design due to a low Technology Readiness Level (TRL) or impractical application on a drone. For more information the reader can reference to the midterm report [23].

2.2. Function of the Swarm

The primary function of the swarm is to perform a complete visual inspection of the external structure of an aircraft. The FFBD and FBS can be found in the baseline report [22]. The system has been designed specifically for the inspection of airliners with a narrow body aircraft as the benchmark. During the inspection, all the external structural components are covered. Visual data is collected autonomously following an optimised inspection path. The collected data is processed with dedicated algorithms trained for identification of defects on the external structure.

The system is able to operate in both the controlled environment of a maintenance hangar, as well as in less predictable conditions found at and around an airport gate. Drones can be added to the swarm to decrease inspection time if required. The inspection algorithms have been designed in such a manner that the system can be easily deployed for inspection of other structures such as a bridge or a rocket.

2.3. Configuration

Following a thorough trade-off process, a swarm configuration has been chosen. More information on the trade-off can be found in the midterm report [23]. The configuration consists of a total of five drones: two "CHIP" drones and three "DALE" drones. The swarm can be seen in Figure 2.1. The drones are connected to the ground station which acts as the central command station; it supplies the drones with their flight path, monitors the inspection operation, and redistributes the tasks in case of a drone failure.

[1] Retrieved from: `https://dictionary.cambridge.org/dictionary/english/swarm` (cited 14 June 2022)

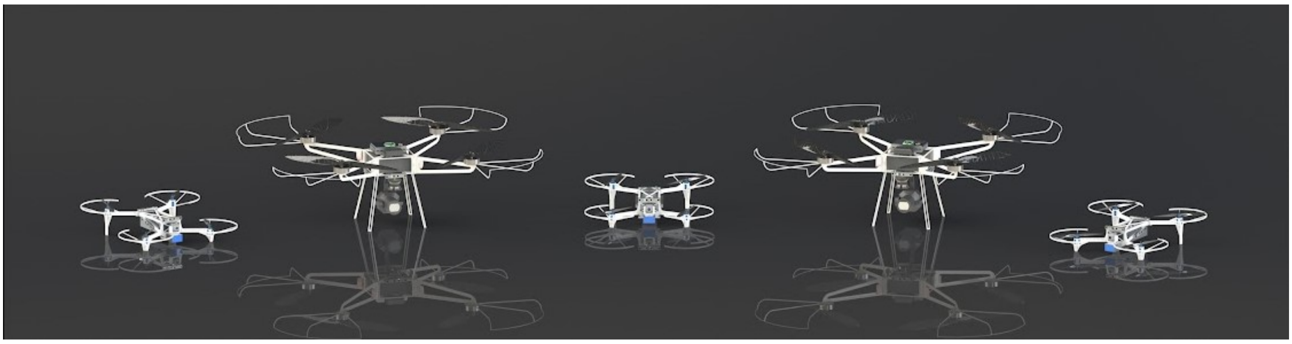


Figure 2.1: Side-by-side render of the swarm configuration, two “CHIP” drones and three “DALE” drones.

The swarm of drones has been composed of different types, i.e. big and small ones, to increase flexibility and performance. The small drones can cover areas of the aircraft that are too confined for the big drones, e.g. the nacelles and bottom side of the wings. The big drone can deliver superior performance in data collection in areas where reference points are sparse, such as the wingtip. More information on this can be found in chapter 5.

3

Market Analysis and Positioning

The swarm of drones will be developed and operated within the context of a company. The company has to position itself in a market. In this chapter, a discussion is provided on the market in which the company will operate. The market is defined, and analysed in Section 3.1. This includes a definition, size estimate, market demand, customer concerns, and competitors. Next, it is discussed how the company will be position itself in this market in Section 3.2. This includes a discussion of the competitive advantage, and business model. Section 3.3 provides a Strengths, Weaknesses, Opportunities, Threats (SWOT) analysis, and further elaborates on market opportunities and limitations.

3.1. Market Definition and Analysis

In this section, the market in which the company will operate is discussed. The market definition, size, and growth foresight are given. Next to that, customer demand and concerns are discussed. An overview of competitors is given, which performance measures have been used as a benchmark for the design of the swarm of drones.

3.1.1. Market Definition

The high level market that the company will operate in is defined as the aerospace market. This includes sales of aerospace equipment, and related services such as Maintenance, Repair, and Overhaul (MRO) of aircraft. The latter can be split up further into A checks on the one hand, and C and D checks, and overhaul on the other hand. A checks are performed frequently and require only one to two days in hangar[1]. This mostly includes visual inspection of the structure, applying lubrication, changing filters, and checking critical systems. C and D inspections, require significantly more time, taking several weeks for one aircraft. Frequently, overhaul is executed in combination with C or D type check[2]. The company will be active in the first market segment involving A checks. Here, the most value can be added initially due to the frequency and relative importance of visual checks. The market definition is presented visually in Figure 3.1.

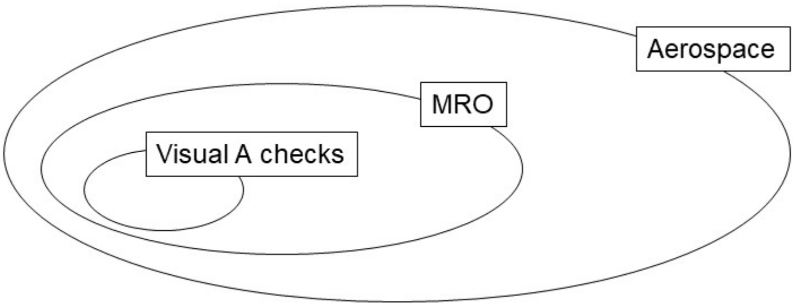


Figure 3.1: Venn diagram of visual A checks market, as a subset of the MRO and aerospace market.

3.1.2. Market Size and Growth

The global aerospace market[3] has seen a minor contraction during the period 2015-2020 with a Compound Annual Growth Rate (CAGR) of -0.5%. The decline during this period can be explained by a mixture of political uncertainties, volatile raw material prices and grounding of planes, i.e. the grounding of the Boeing 737 MAX. The value of the aerospace market in 2020 has been estimated to be $298b. Most recently, the COVID pandemic has been impacting the aerospace industry significantly. Worldwide supply and demand for air travel

[1] Retrieved from: `https://simpleflying.com/aircraft-maintenance-checks/` (cited 13 June 2022)

[2] Private conversation with Fokker Services

[3] Retrieved from: `https://www.thebusinessresearchcompany.com/report/aerospace-market#:~:text=Aerospace%20Market%20Size,at%20a%20rate%20of%207.7%25.` (cited 13 June 2022)

plummeted by more than 50%[4]. Demand and supply has been recovering since but remains uncertain due to new outbreaks as seen in the last months in China. On top of that, inflation, high energy prices, and a potential oncoming recession may put more pressure on the industry in the short term. However, the long-term view remains positive with the industry expected to reach $573b by 2030.

Market circumstances such as the pandemic, high energy prices, and governmental initiatives[5] are pushing airliners to regard MRO as a strategic activity instead of a necessary evil. The MRO market is estimated to be worth $90b in 2022 and is expected to grow with a CAGR of 4.57% during the period 2022-2030 despite the hardships faced by the general aerospace industry. Engine servicing is expected to dominate MRO market growth, whilst global fleet growth is expected to be dominated by narrow body aircraft[6] [7]. The global commercial transport aircraft fleet in 2022 consists of 25,578 aircraft, where the Western European fleet consists of around 6000 aircraft. The global fleet is projected to grow to 38,189 by 2032 [8].

Embraer[9] states that there is on average one lightning visual inspections per aircraft per year, with an estimated current industry standard cost per visual inspection of €10,000. Aircraft fleet operator Air New Zealand[10] daily utilisation figures provide an indication on the average number of flight hours per aircraft per year. With most figures hovering around the 7 to 10 hour daily utilisation mark, a reasonable estimate would be five General Visual Inspection (GVI) per year (assuming a A-check every 600 flight hours[11]). The total market for commercial aircraft visual inspection in Western Europe is therefore estimated to be worth €360m annually.

3.1.3. Market Demand

A collection of market conditions shape the demand for visual inspections of aircraft. Recent economic woes compel airliners to explore strategic benefits in new business areas. They are looking for more efficient maintenance practices leading MRO providers to explore innovative technologies. An increasing demand combined with a shortage of labour in the industry[12] calls for improvements in efficiency.

One area where efficiency can be improved is visual inspections. Aircraft fleet operators such as Air France-KLM are expected to gain significant benefits from novel inspection approaches. Current manned visual inspection requires approximately eight hours of hangar time[13], resulting in high overhead and personnel costs, as well as a lower fleet utilisation. Visual drone inspections can significantly decrease this time to less than an hour. It must however be noted that decrease of inspection time in a maintenance hangar is only beneficial to a certain point since other tasks have to be performed in parallel, e.g. oil changes, filter replacements. This point lies around the Mainblades benchmark of 45 minutes[14].

Next to decreasing inspection time, aircraft fleet operators and MRO's are constantly looking to cope with rising costs, e.g. increasing cost of materials and upwards technician wage pressure. One of the top mitigation strategies for decreasing the impact of rising material costs is to further "leverage data analytics, aircraft health monitoring and predictive maintenance to reduce material usage"[15]. Automated solutions are capable of documenting every step of the inspection process, and are eminently suited for data driven approaches, such as predictive maintenance. A recent example is the partnership of Mainblades and KLM, which aims to monitor paint peeling issues with Boeing 787 and Airbus A350 aircraft[16].

[4] Retrieved from: https://www.icao.int/sustainability/Documents/Covid-19/ICAO_coronavirus_Econ_Impact.pdf (cited 13 June 2022)

[5] Retrieved from: https://www.grandviewresearch.com/industry-analysis/aircraft-mro-market (cited 13 June 2022)

[6] Retrieved from: https://www.oliverwyman.com/content/dam/oliver-wyman/v2/publications/2018/october/Europe-MRO-2018-Trends-&-Forecast-preso.pdf (cited 3 June 2022)

[7] Retrieved from: https://www.mordorintelligence.com/industry-reports/global-aircraft-maintenance-repair-and-overhaul-market-industry/ (cited 25 April 2022)

[8] Retrieved from: https://www.statista.com/statistics/282237/aircraft-fleet-size/ (cited 25 April 2022)

[9] Private communication with Embraer, June 2022

[10] Retrieved from: https://p-airnz.com/cms/assets/NZ/PDFs/airnewzealand-databook-2015.pdf (cited 3 June 2022)

[11] Retrieved from: https://www.aerotime.aero/articles/28347-aircraft-maintenance-abcd-checks (cited 3 June 2022)

[12] Retrieved from: https://www.grandviewresearch.com/industry-analysis/aircraft-mro-market (cited 13 June 2022)

[13] Retrieved from: https://mainblades.com/save-costs/ (cited 14 June 2022)

[14] Private communication with Mainblades, May 2022

[15] Retrieved from: https://www.oliverwyman.com/content/dam/oliver-wyman/v2/publications/2018/october/Europe-MRO-2018-Trends-&-Forecast-preso.pdf (cited 3 June 2022)

[16] Retrieved from: https://aviationweek.com/mro/safety-ops-regulation/tackling-787-a350-paint-headaches-through-drone-inspections (cited 3 June 2022)

3.1.4. Customer Concerns

The market demand creates opportunities for introducing new technologies into the market such as the swarm of drones which is discussed in this report. However, some potential concerns of customers have to considered. These are discussed in this subsection.

The leading cause for the upwards technician wage pressure is a lack of labour supply, coupled with technician retirements. Automated inspection solutions have the potential to increase the inspection productivity of individual technicians, reducing labour demand[17]. This however does assume that technicians adopt the novel technique. This is not always evident, with technicians having to change their way of working, and citing concerns of overall job replacement[18]. Therefore, significant attention shall be paid to proper integration of the novel inspection technique with status quo working methods of technicians[19].

Secondly, as a yet unproven and uncertified technology, it poses a substantial risk for a customer to make an upfront investment in the system[20]. Additionally, system bugs and subsequent upgrades are not uncommon in initial phases of operation. These factors are important to take in consideration when formulating the business model [33].

Third of all, MRO's communicate a hesitancy in allowing automated machines in the vicinity of aircraft. Fokker Services[21] mentions that customers prefer to not have novel technologies in the vicinity of their aircraft. Additionally, it was brought to attention that during initial testing with KLM, Mainblades was not allowed to fly directly above aircraft services, and that the drone pilot was not comfortable with flying closer than around two meters[22]. It is therefore important to address these concerns at an early stage.

Finally, regulatory considerations have to be taken into account. Flying in the hangar of a client with closed doors only requires permission of the customer. However, if inspections operations are expanded to outdoor inspections, e.g. at the gate, approval is needed from multiple parties including airport authorities, aircraft fleet operators, and MRO's[23].

Next to that, in order to operate within the aviation industry, inspection methods and systems have to be certified, which is a lengthy process; Mainblades' method has not yet been certified for visual aircraft inspection after five years of development[24]. There are three main ways for a novel techniques to get certified; regulatory agency certification (European Union Aviation Safety Agency (EASA), Federal Aviation Administration (FAA), Maintenance Organisation Exposition (MOE)), certification through an aircraft fleet operator, or inclusion in the Original Equipment Manufacturer (OEM) part and/or system manual. The long time to certification for Mainblades' system brings forward concerns that certification of the system may be a long process for a swarm of inspection drones, given the additional novelty of the swarm concept.

One mediating factor is that current drone aircraft inspection competitors have likely shortened the way to certification for new actors. An additional consideration with respect to certification is that of the drones themselves. Mainblades uses modified DJI drones that have previously been tested and certified[25]. The certification/regulation of the self-designed drones is thus an additional consideration[26]. For example, an aircraft fleet operator will naturally require that the drones themselves are certified, before even considering a collaboration such as the one between KLM and Mainblades.

3.1.5. Competitors

A detailed analysis of competitors and the capability of their drone inspection solutions is vital in ensuring that the proposed swarm of inspection drones is market-capable, i.e. it outperforms competitors in the space of aircraft visual inspection. A distinction is made between competitors acting in the same market segment, and competitors acting in a different market segment (e.g. wind turbine blade inspection).

[17] Retrieved from: `https://www.oliverwyman.com/content/dam/oliver-wyman/v2/publications/2018/october/Europe-MRO-2018-Trends-&-Forecast-preso.pdf` (cited 3 June 2022)

[18] Private communication with Fokker Services, June 2022

[19] Retrieved from: `https://en.customdrone.com/cases/lightning-strike-inspecties-met-een-drone-met-ai-binnen-30-minuten-alle-schades-in-beeld` (cited 7 June 2022)

[20] Private communication with Mainblades, May 2022

[21] Private communication with Fokker Services, June 2022

[22] Private communication with R.M. Groves, May 2022

[23] Private communication with Mainblades, May 2022

[24] Private communication with Mainblades, May 2022

[25] Retrieved from: `https://mainblades.com` (cited 7 June 2022)

[26] Retrieved from: `https://www.easa.europa.eu/domains/civil-drones/drones-regulatory-framework-background/open-category-civil-drones` (cited 7 June 2022)

Aircraft Visual Inspection Market Segment

This section discusses competitors in the market segment drones for aircraft visual inspection. Four additional competitors were found and are included for reference: MRO Drone[27], Luftronix[28], Embraer[29], and Korean Air[30]. Note that these competitors are not included in the table due to a lack of information found.

Table 3.1: Summary table of competitor capabilities and features. Columns are as follows: competitor name, inspection types that can be performed, information on data processing, partner companies, and inspection time in minutes.

Competitor	Inspection Types	Data Processing	Partners	Time [min]
Mainblades[31] [32]	upper surface, GVI, lightning strike, paint	off-board processing, based on the structural repair manual, accurate dimension estimates, locations in stringers and stations	KLM	45
Donecle[33]	full external surface, narrow-body, GVI, lightning strike, paint, markings	off-board processing, minimum defect size of $1\,\mathrm{mm}$, defects located on an aircraft structural model, a pre-filled inspection report is generated	n.a.	<60
CustumDrone[34]	single-aisle/narrow-body, GVI, lightning strike, bird strike, hail, icing	off-board processing, drone inspection with artificial intelligence	KLM City-hopper, Regional Jet Center	<30

Competitors in Other Market Segments

Google visual inspection AI provides a service to quickly and easily train and deploy a visual damage detection algorithm[35]. It has the possibility to provide a method to perform image analysis without in-house design and development (and costs associated with it). Perceptual robotics employs inspection drones to inspect wind turbines. It provides an all-in-on solution for inspection and defect analysis, as well as offering certain software modules as individual products[36].

XYREC provides an automated Ground-Based Vehicle (GBV) solution for coating and de-coating of aircraft surfaces. Its robot can reach any surface on fixed- and rotary wing aircraft[37]. Extrapolating its capabilities to the segment of aircraft inspection, it may be a possibility to swap the current tools with (non-)contact inspection sensors. For example, thermography is deemed non-feasible at this stage for drone inspection due to mass and power constraints present. Mounting the thermography hardware on a ground-based platform could add significant value in terms of inspection scope, whilst leading to only minimal increases in swarm control complexity.

3.2. Market Positioning

This section discusses the market positioning of a swarm of inspection drones. Subsection 3.2.1 presents strategic advantages of a swarm with respect to single drone solutions. Subsection 3.2.2 mentions the business model chosen, and elaborates on its advantages for the customer.

3.2.1. Strategic Advantages of a Swarm

As discussed in Subsection 3.1.5, most competitors offer single drone visual inspections. These solutions already reduce the inspection time significantly. However, the swarm of drones has some advantages that can prove strategically superior to single drone solutions in the long run. These are discussed in this subsection.

First of all, a swarm of drones can reduce the inspection time even further. Once the system has been designed, tested, and verified, it allows for rapid scaling; although the swarm has been sized to complete an inspection

[27] Retrieved from: `http://www.mrodrone.net` (cited 7 June 2022)

[28] Retrieved from: `https://luftronix.com` (cited 7 June 2022)

[29] Private communication with Embraer, June 2022

[30] Retrieved from: `https://aviationweek.com/air-transport/maintenance-training/korean-air-inspects-aircraft-using-drone-swarm` (cited 7 June 2022)

[35] Retrieved from: `https://cloud.google.com/solutions/visual-inspection-ai` (cited 7 June 2022)

[36] Retrieved from: `https://www.perceptual-robotics.com/solutions/` (cited 7 June 2022)

[37] Retrieved from: `https://www.xyrec.com/technology/` (cited 7 June 2022)

of a narrow body aircraft in half an hour, this can be scaled by adding drones. In the current setting where an aircraft is inspected within a maintenance hangar, this scaling offers limited added value. However, it makes the system future proof; once regulatory hurdles and safety issues are tackled, a swarm opens up possibilities that can increase efficiency even further. The airport of the future could feature a swarm of drones for every gate or group of gates. In the time that passengers embark and disembark, and cargo is being loaded and unloaded, the swarm can autonomously perform an inspection of the aircraft. The consistently collected data allows for data analytics, improved aircraft health monitoring, and analysis of defects beforehand without requiring the aircraft to be taken out of service. Ultimately, this approach should lead to improved efficiency and decreased costs of MRO practices.

Secondly, better coverage of the aircraft by a swarm is a short-term advantage. Single drone systems require a relatively large drone in order to cover the whole aircraft with one battery load. A swarm can consist of multiple smaller drones that each individually cannot inspect the whole aircraft within the allotted time. The smaller size allows the drones to cover confined spaces such as the nacelles. A swarm can potentially provide a more complete picture of the state of the aircraft.

Lastly, since the swarm provides better coverage, it can also provide data on areas of the aircraft that are still inspected manually such as the bottom side of the wing.The direct added value is minimal since these areas can easily be inspected by a human operator without using digital tools. However, consistently collecting and storing data allows for improved health monitoring and data analysis in the long term.

3.2.2. Business Model

In order to bring the customer maximum value, the company has chosen for a Hardware-as-a-Service business model. The hardware is provided on the basis of a Service Level Agreement (SLA). The swarm of drones is provided to the customer for use with continuous support while the company receives monthly payments.

The use of this business model provides advantages to the customer. First of all, the swarm is provided to the customer with the warranty that their system is up-to-date with the newest hardware and software improvements. In this manner, they are assured of continuously using the latest innovations. Next to that, the low investment requirements improves the adoption rate for potential customers. There is no need for long decision processes involving major hardware investments. Ultimately, using this business model reduces the investment risk of the customer. Finally, the customer receives continuous support from the company such as personnel training, troubleshooting, and consultancy services, lowering their operational risk.

This format also provides advantages to the company. First of all, a durable relationship with the customer is cultivated. This improves the customer retention rate and offers potential for a solid customer feedback system for continuous improvements to the products and services the company offers. Next to that, it provides a continuous operational cashflow which strengthens the company fundamentals.

This business model also introduces some inherent risks and challenges. First of all, hardware production has to be financed. In classic manufacturing, this can be done by upfront payments of the customer. Due to the nature of the business model, this is not possible. Two main options can be considered. Financing can be done by the company itself through company funds, or an external equipment financing company can be considered. Next to that, notwithstanding the advantages, providing continuous support can weigh on profit margins; if a customer requires an elevated amount of support, it can occupy a considerable amount of labour hours. This risk is mitigated by offering different SLA packages with differing amounts of included support.

On top of the hardware, a software platform is provided to the customer. In this platform, the collected data is processed and presented to the users of the system. It provides Artificial Intelligence (AI) data processing, easy-to-understand visualisations, and data analytics. The company regards this platform as the basis for its strategic plans. Customer input will be leveraged to add features, services, and products through time. Examples include the use of a digital twin for data consolidation, remote inspections through video connections[38], and further robotisation of the inspection and maintenance process[39].

3.3. Market Opportunities and Limitations

This section will describe market opportunities and limitations for a hybrid swarm of visual inspection drones. To assess the current market position, a SWOT analysis was performed in Table 3.2 below.

[38] Private communication with Embraer, June 2022

[39] Private communication with NLR, June 2022

Table 3.2: SWOT analysis for the mixed swarm of visual inspection drones. Refer to the report main body for a further elaboration.

	Helpful	**Harmful**
Internal	• Added drone redundancy • Scalable number of drones • Flexible deployment • Novel maintenance strategies	• Increased collision risk • Increased design complexity • Increased development risk • Increased certification risk
External	• Competitor solutions uncertified • Suitable to multiple market segments • Global aircraft fleet growth • Increased damage traceability • Inspection at the gate	• General hesitancy in technology adoption • Existing solutions already have inspection data • Solely aircraft visual inspection • Existing solutions are working with fleet operators

Opportunities

There is additional market opportunity beyond the aforementioned A and B-check visual inspection. The FAA mandates aircraft inspection every 100 flight hours[40], and some aircraft operators even perform inspection at the 50 flight hour mark[41]. Flexible deployment of a swarm of inspection drones, i.e. not limited to the hangar, could provide opportunity for visual inspection beyond both mandated, and industry standard checks. Increase of visual inspection beyond mandated increases the amount of data, and thus traceability, and potency of adopting state-of-the-art predictive maintenance.

A swarm of inspection drone provides the added flexibility of increasing both the number and type of drones. The current solution consists of two types of drones, and thus automatically incorporates design properties for swarms consisting of multiple drone types (e.g. two types of visual and one for thermography). One specific area which could be researched is the addition of smaller drones for interior and exterior documentation inspection (assuming that the drones document damage that they themselves might do). This process is currently performed manually before every maintenance task to shelter MROs such as Embraer from liability in case of damages[42]. Furthermore, if the TRL of thermography inspection using drones is sufficient, it will be possible to add these drones to the swarm with minimal design changes.

Further away into the future it may become possible to keep drones in a central hub on an airport, with the possibility of dispatching them to certain gates/regions where they can fly from aircraft to aircraft. Such a strategy would even further reduce the amount of time and resources spent on external surface visual inspection.

Limitations

Although competitors such as Mainblades have not yet been certified, there are two significant advantages that they have over this Design Synthesis Exercise (DSE) group. Firstly, competitors have existing collaborations with fleet operators to test and demonstrate their system in real-world inspection environments (see Subsection 3.1.3). New collaborations of the swarm of inspection drones with aircraft fleet operators are possible, though companies are generally time and resource constrained. Secondly, competitors have already gathered a great amount of data during testing. This provides a significant knowledge advantage, which shall be reduced through fast development of the swarm of inspection drones.

Furthermore, the proposed solution of a swarm of inspection drones results in an added collision risk. This collision risk not only includes drone-aircraft, but also drone-drone collisions. This added collision risk shall be mitigated, increasing the design complexity. Lastly, the swarm aspect on top of the drone inspection adds even more novelty in terms of regulation/certification. Attention shall be paid to mitigating this certification risk as much as possible.

[40] Retrieved from: `https://www.law.cornell.edu/cfr/text/14/91.409` (cited 11 May 2022)
[41] Retrieved from: `https://aerocorner.com/blog/types-of-aircraft-inspections/` (cited 11 May 2022)
[42] Private communication with Embraer, June 2022

4

Inspection Interface and Data Handling

This chapter elaborates on the design of inspection interface and data handling subsystem. This subsystem includes the interface used to gather the visual data of the aircraft surfaces, as well as processing of that data. The chapter starts with listing the requirements of the subsystem in Section 4.1, following with the selection of the camera in Section 4.2. Then the coverage maps of the inspection are shown in Section 4.3. Lastly, data handling is explained in Section 4.4.

4.1. Requirements

The subsystem requirements for inspection are derived from the system requirements related to the inspection procedure and data handling, and are presented in Table 4.1.

Table 4.1: Inspection Requirements.

Identifier	Requirement	Related System Requirements
SUB-INS-01	The system shall be able to carry out an inspection, defined from take-off until landing, within 30 minutes (i.e. excluding system set-up and pack-up).	SYS-TIM-04, StRS-MO-05, StRS-MO-06
SUB-INS-02	The system shall be able to detect defects of size 1-10 mm with a probability of 80% or higher.	SYS-REL-08, StRS-MO-06
SUB-INS-03	The system shall be able to detect defects of size minimum 10 mm with a probability of 90%.	SYS-REL-04, StRS-MO-09
SUB-INS-04	The system shall have a maximum false-positive rate of 10% for defects of minimum 10 mm.	SYS-REL-05, StRS-MO-09
SUB-INS-05	The system shall have a maximum false-positive rate of 20% for defects of size 1-10 mm.	SYS-REL-09, StRS-MO-06
SUB-INS-06	The system shall not come closer than 0.5 meters to the inspected aircraft unless it is doing a contact inspection.	SYS-SAF-01, StRS-RA-02
SUB-INS-09	The system shall take up to 15 minutes for processing inspection data after the landing	SYS-TIM-05, StRS-MO-05.

4.2. Camera selection

The data used in the damage detection algorithm is generated by the camera on the drone. Therefore, the camera selection is an important decision, and an extensive analysis has been performed to determine which camera has to be used. To start the camera analysis, multiple Commercial-Off-The-Shelf (COTS) cameras used in commercial drones were compared. These cameras are shown in Table 4.2.

Table 4.2: Drone cameras used in the camera selection.

Camera	Mass [g]	FOV [°]	Resolution [pixels]	Exposure time [s]	Minimum focus distance [m]
Zenmuse P1[1]	800	38.94×25.95	8192×5460	1/2000	0.93 [2]
EVO II Pro camera[3]	380[4]	67.84×46.07	5472×3648	1/8000	1
EVO II camera[5]	350[6]	63.28×47.46	8000×6000	1/8000	0.5
EVO Nano camera[7]	82 ± 10%	67.2×50.4	8000×6000	1/8000	1
EVO Nano + camera[8]	82 ± 10%	68.0×51.0	8192×6144	1/8000	0.5
DJI Mini 3 pro camera[9]	82[10]	65.68×49.26	8064×6048	1/8000	1
DJI Mavic 3 tele camera[11]	90 ± 10%	13.07×7.35	3840×2160	1/8000	3
DJI Mavic 3 Hasselblad camera[12]	90 ± 10%	67.22×50.37	5120×2700	1/8000	1
DJI Air 2S camera[13]	119[14]	73.22×48.81	5472×3648	1/8000	0.6

All the cameras shown in Table 4.2 have a gimbal. The EVO nano and EVO nano+ are $250\,\mathrm{g}$ drones, same as DJI mini 2 and 3, therefore it was assumed that the weight of the cameras would most likely be the same or very close to each other. To account for uncertainty a 10% margin was taken into account.

4.2.1. Inspection Time Calculator

To be able to determine the best available camera to be used on the drone, the inspection time was calculated for every camera. Two methods for carrying out the inspection were taken into account. In the first method, the drones move at a constant speed while for the second method, the drones stop when taking a picture. For the calculation of the inspection time, the following assumptions were used:

1. The projected pixel size on the surface is based on the assumption that, to be able to detect damages of size l_{damage}, the pixel size should be maximum $\frac{l_{damage}}{3}$ [15].
2. The inspected surfaces are approximated by simple geometric shapes, e.g. cylinders and triangles.
3. The dimensions of an Airbus A320[16] are used to approximate the area of the inspected surfaces.
4. The time it takes for a drone to move between different parts of the aircraft (while it is not inspecting the surface) is not taken into account.
5. The gimbals are perfect stabilisers.
6. A picture overlap of 10% is used for ease of post processing and uncertainty in the position of the drone.

Using the above assumptions, the total area, to be inspected, of an Airbus A320 was calculated to be $1306\,\mathrm{m}^2$.

[1] Retrieved from: https://www.dji.com/nl/zenmuse-p1/specs (cited 31 May 2022)

[2] Retrieved from: https://dl.djicdn.com/downloads/Zenmuse_P1/20210510/Zenmuse_P1%20_User%20Manual_EN_v1.2_3.pdf (cited 31 May 2022)

[3] Retrieved from: https://auteldrones.com/pages/evo-ii-pro-specification (cited 31 May 2022)

[4] Retrieved from: https://www.bhphotovideo.com/c/product/1561108-REG/autel_robotics_600002067_evo_ii_6k_gimbal.html/specs (cited 31 May 2022)

[5] Retrieved from: https://www.autelrobotics.com/productdetail/1.html (cited 31 May 2022)

[6] Retrieved from: https://www.bhphotovideo.com/c/product/1561107-REG/autel_robotics_600002066_evo_ii_8k_gimbal.html/specs (cited 31 May 2022)

[7] Retrieved from: https://www.autelrobotics.com/productdetail/24.html (cited 31 May 2022)

[8] Retrieved from: https://www.autelrobotics.com/productdetail/24.html (cited 31 May 2022)

[9] Retrieved from: https://www.dji.com/nl/mini-3-pro/specs (cited 31 May 2022)

[10] Retrieved from: https://www.amazon.com/Gimbal-Assembly-Genuine-Replacement-Runchicken/dp/B08QV4VD39 (cited 31 May 2022)

[11] Retrieved from: https://www.dji.com/nl/mavic-3/specs (cited 31 May 2022)

[12] Retrieved from: https://www.dji.com/nl/mavic-3/specs (cited 31 May 2022)

[13] Retrieved from: https://www.dji.com/nl/air-2s/specs (cited 31 May 2022)

[14] Retrieved from: https://www.amazon.com/Original-Gimbal-Camera-Assembly-runchicken/dp/B097JNX2D1 (cited 31 May 2022)

[15] R.M. Groves Private Communication, May 2022

[16] Retrieved from: https://modernairliners.com/airbus-a320-introduction/airbus-a320-specs/ (cited 10 May 2022)

Constant Motion Method

The first method will result in motion blur. Therefore, the exposure time of the camera is used to calculate this blur. As the drone moves at a constant speed V, the camera sensor will be exposed to light for the duration of the exposure time, t_{exp}. Sort of stretching the pixel. To be able to detect the damages, this stretched pixel size, gathered over the exposure time, should have a maximum size of $\frac{l_{damage}}{3}$. Therefore, the size of a pixel projected on the surface has to be:

$$l_{pix} <= \frac{l_{damage}}{3} - x_{blur} \tag{4.1}$$

where x_{blur} is the distance travelled during the exposure time.

$$l_{pix} <= \frac{l_{damage}}{3} - V * t_{exp} \tag{4.2}$$

The relation in Equation 4.2, between motion blur and pixel size, is visualised in Figure 4.1.

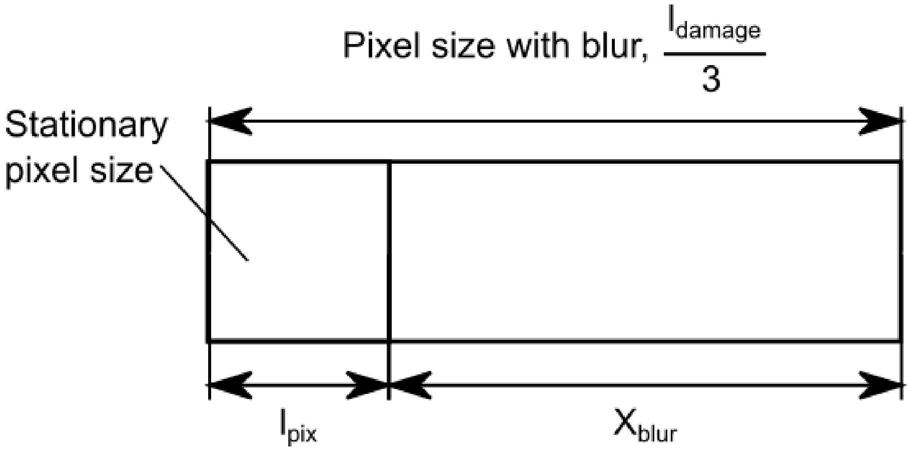


Figure 4.1: The effect of motion blur on pixel size.

Multiplying the pixel size by the horizontal resolution of the camera, n_{pix_H}, will result in the projected width of the picture on the aircraft surface, l_{view_H}, as shown in Equation 4.5. The projected height of the picture can then be obtained by dividing the projected width by the aspect ratio of the picture, Equation 4.4. The total inspection time for one drone is then calculated by dividing the area of the aircraft, A, by l_{view_V} and V, as shown in Equation 4.3. This assumes that the area of the aircraft is laid out into a single horizontal strip, which has the same height as the picture, and the drone will scan this strip in one motion at a constant speed.

$$t_{insp} = \frac{A}{l_{view_V} * V} \tag{4.3}$$

$$l_{view_V} = \frac{l_{view_H}}{AR} \tag{4.4}$$

$$l_{view_H} = l_{pix} * n_{pix_H} \tag{4.5}$$

$$t_{insp} = \frac{A * AR}{V * n_{pix_H} * \left(\frac{l_{damage}}{3} - V * t_{exp}\right)} \tag{4.6}$$

The

where AR, is the aspect ratio of the camera resolution ($\frac{n_{pix_H}}{n_{pix_V}}$). Finally the distance between the drone and the aircraft has to be determined to make sure that the drone will not crash into the aircraft, while the pictures will still be in focus. Through simple geometric relations, the distance to the aircraft is calculated using Equation 4.7.

$$\begin{aligned} d_{aircraft} &= \frac{l_{view_H}}{2 * tan\left(\frac{\alpha_{FOV_H}}{2}\right)} \\ &= \frac{n_{pix_H} * \left(\frac{l_{damage}}{3} - V * t_{exp}\right)}{2 * tan\left(\frac{\alpha_{FOV_H}}{2}\right)} \end{aligned} \tag{4.7}$$

Because the drone is taking pictures at constant intervals,t_{int}, and moving at a constant speed, a maximum speed can be determined at which the pictures have 10% overlap, as mentioned in the assumptions above. First, the distance between the centers of neighbouring pictures is defined as follows,

$$D = l_{view_H} * (1 - k)$$

where k is the 10% overlap as described above. The maximum speed, V_{max}, is then determined by,

$$\begin{aligned} V_{max} &= \frac{D}{t_{int}} \\ V_{max} &= \frac{n_{pix_H} * (1 - k)}{t_{int}} * \left(\frac{l_{damage}}{3} - V_{max} * t_{exp}\right) \\ V_{max} &= \frac{\frac{l_{damage}}{3}}{\frac{t_{int}}{n_{pix_H} * (1-k)} + t_{exp}} \end{aligned} \tag{4.8}$$

Stopping Method

For the second method, where the drones will stop every time they take a picture, there will be no motion blur. This means that the drones will be able to fly further away from the aircraft. Because the drones will not be flying at a constant speed, the acceleration and deceleration times have to be accounted for. The following assumptions are taken into account.

1. The drones are assumed to have no relative velocity with respect to the aircraft when taking a picture, therefore no motion blur is generated.
2. Between pictures, the drone will accelerate and decelerate at a constant rate for equal amounts of time.
3. The drone has a maximum tilt angle of $30°$, according to the information in chapter 6 (maximum acceleration of $0.58\,\mathrm{m\,s^{-2}}$).

The time it takes to travel between two points, at a constant acceleration and deceleration each for half the time, can be derived from the following relation:

$$x = \frac{a}{2} * t^2$$

The motion is symmetric, this means that the distance travelled at half the total time, will be half the total distance. This results in:

$$T = 2 * \sqrt{\frac{D}{a}}$$

where T and D are the total time and distance between pictures respectively. The distance between pictures, including overlap, is determined as follows:

$$D = l_{view_H} * (1 - k)$$

where k is the overlap as a ratio between picture width and overlap width.

As there is no motion blur, the projected size of a pixel has to be less than or equal to one third of the damage size:

$$l_{pix} = \frac{l_{damage}}{3}$$

The total inspection time for one drone is then determined by calculating the strip length, as was discussed in the previous section, divide it by the distance between pictures, and multiplying it by the time between pictures. This results in Equation 4.9 below.

$$
\begin{aligned}
t_{insp} &= \frac{A * T}{l_{view_V} * D} \\
&= \frac{2 * A * AR}{\sqrt{l_{view_H}^3 * (1-k) * a}} \\
&= \frac{2 * A * AR}{\sqrt{\left(\frac{l_{damage}}{3}\right)^3 * n_{pix_H}^3 * (1-k) * a}}
\end{aligned}
\tag{4.9}
$$

For this method, the distance between the drone and the aircraft is calculated using the same method as for the drones in constant motion, but neglecting the motion blur.

$$
\begin{aligned}
d_{aircraft} &= \frac{l_{view_H}}{2 * tan\left(\frac{\alpha_{FOV_H}}{2}\right)} \\
&= \frac{n_{pix_H} * \frac{l_{damage}}{3}}{2 * tan\left(\frac{\alpha_{FOV_H}}{2}\right)}
\end{aligned}
\tag{4.10}
$$

4.2.2. Comparison

Using the equations derived in Subsection 4.2.1, the inspection time with respect to the drone speed for each camera can be determined. In this section, the relation between inspection time and drone speed for the constant motion method is explained for one camera. Afterwards, a comparison between the constant motion and stopping method is given, after which the final cameras, for the “DALE” drone and “CHIP” drone are selected.

To explain the effect of drone speed on the inspection time and distance to the aircraft, this relation is plotted in Figure 4.2 for the EVO nano+ camera. The red curve is the time for one drone to inspect the whole aircraft in seconds and the blue curve is the distance to the aircraft in meters.

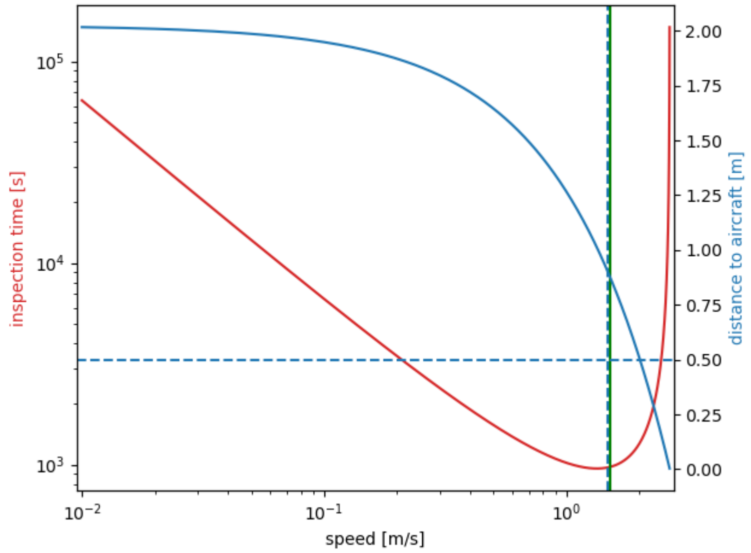


Figure 4.2: Inspection time and distance to the aircraft with respect to the fly speed of the drone for the EVO Nano+ camera.

Looking at the inspection time curve, the inspection time first decreases with increasing speed until it reaches an optimum. After this optimal value it starts to increase again. This can be explained by the motion blur; the faster the drone is moving, the more motion blur is created. When this motion blur becomes too high, the drone has to be very close to the aircraft to still be able to detect the damages. Because the drone is close to the aircraft, the projected picture size will be very small and the drone will have to do more sweeps of the aircraft to be able to inspect everything. For this reason, the inspection time starts increasing again after the optimum until the motion blur becomes too much, and it is impossible to detect the damages.

The horizontal blue line is the minimum distance to the aircraft. For this graph, a minimum distance of $0.5\,\mathrm{m}$ was used, taken from the distance requirement for the “DALE” drone (SUB-INS-06). From the flight control subsystem in Table 6.5, “CHIP” drones need a distance range of $0.58\,\mathrm{m}$ (maximum diagonal displacement) to account for wind gusts. This means that minimum distance to the aircraft should be $1.08\,\mathrm{m}$. The vertical dashed blue line in Figure 4.2 displays the maximum drone speed for which this condition holds.

The maximum drone speed based on the minimum picture interval and picture overlap (Equation 4.8) is also taken into account. In Figure 4.2 this speed is shown as the solid green vertical line, in which an overlap of 10% was used. The drone has to fly at a speed slower than this maximum speed to have enough picture overlap. In conclusion, the optimal inspection time and distance to the aircraft for a certain camera, is determined by taking the lowest inspection time at which the distance to the aircraft still fulfils the requirement, including the distance range determined by wind gusts, and still has enough picture overlap.

This inspection time is calculated for every camera listed in Table 4.2 and shown in Figure 4.3 and 4.4 for the constant motion and stopping method respectively. The left graph shows the inspection time versus mass for each camera and the right graph shows the operating range for each camera in terms of distance to the aircraft. The precise operating range of each camera is also shown in the legend. The cameras for which the maximum distance to the aircraft, due to blur, is smaller than the minimum focus distance are not shown in the right graph and are crossed out in the left graph. The cameras which have an uncertainty in their mass are shown as a horizontal bar (ranging from minimum to maximum mass) instead of a dot.

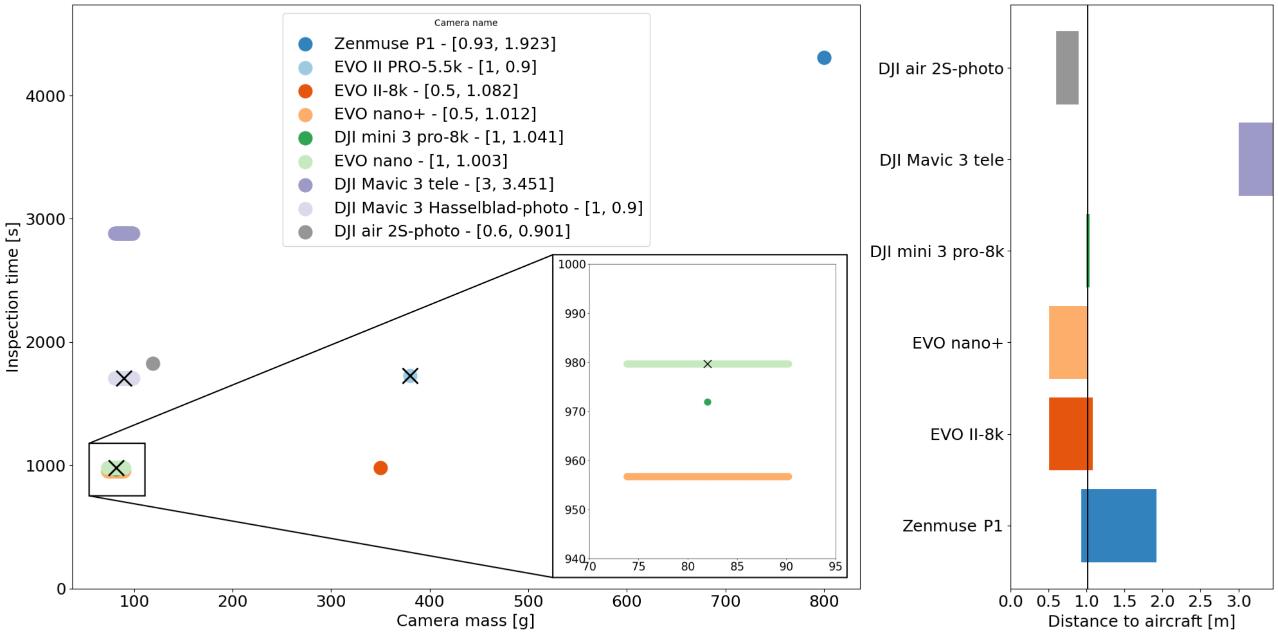


Figure 4.3: Comparison of inspection time, mass, and distance to the aircraft for different cameras using the constant motion method.

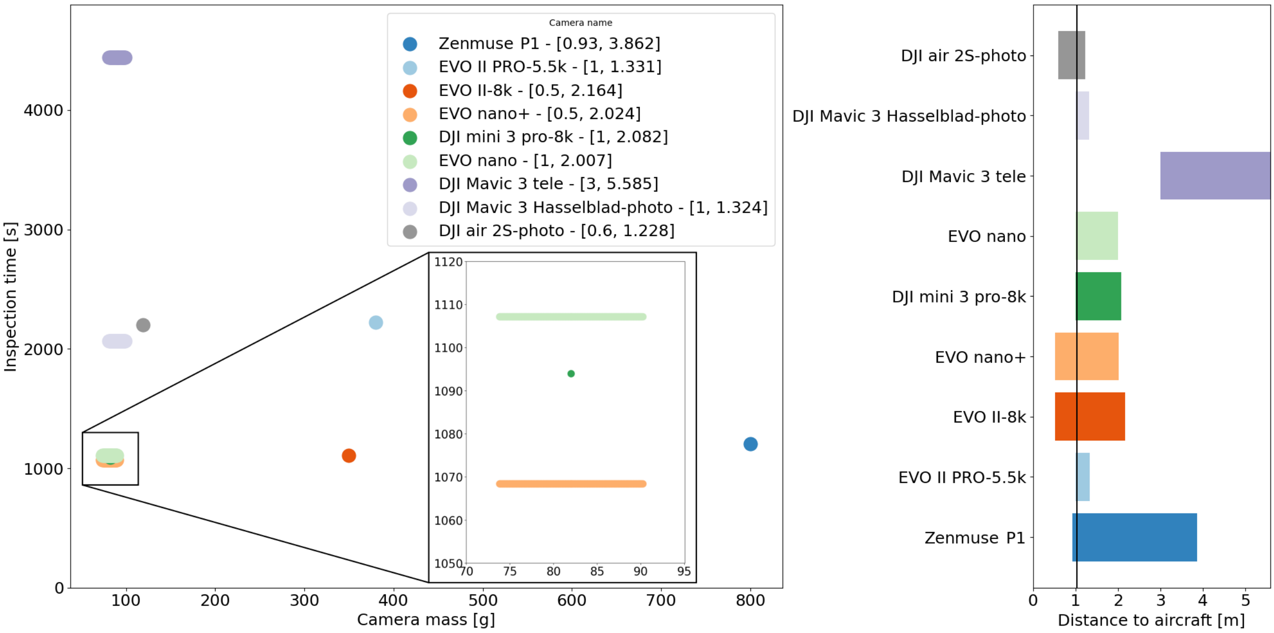


Figure 4.4: Comparison of inspection time, mass, and distance to the aircraft for different cameras using the stopping method.

The inspection times shown in Figures 4.3 and 4.4 above are a first estimate of how long it will take for one drone to inspect the whole aircraft. As this inspection time can differ significantly from the actual inspection time, these values should only be used for comparisons and not for absolute reference.

Comparing Figure 4.3 and 4.4, shows that the constant motion methods results in a very similar inspection time as the stopping method. Although both methods have similar inspection times, the operating ranges (in terms of distance to the aircraft) are significantly lower for the constant motion method than for the stopping method, due to motion blur. For the constant motion method, some cameras even have to be so close to account for motion blur that they can no longer focus. Of the cameras that are able to focus, only four are able to operate at a distance of $1.08\,\mathrm{m}$ or higher. These are the DJI Mavic 3 tele camera, EVO nano+ camera, EVO II-8k camera and the Zenmuse P1.

The constant motion method assumes the shortest exposure time setting. This setting might result in images that are too dark to detect damages. Increasing the exposure time will increase the brightness but will also increase motion blur. For the stopping method, this will not be a problem as there is no motion blur. For this reason, as well as the larger operating ranges, the stopping method was chosen to be the better option. The larger operating range also makes it easier to handle design changes which increase the minimum operating range and minimum distance to the aircraft.

For the camera selection of the “DALE” drone, the camera should have a low mass, acceptable operating range, low inspection time, and should be able to focus at a distance of $1.08\,\mathrm{m}$. On the right of Figure 4.4, the EVO nano, DJI mini 3 pro-8k, EVO nano+, EVO II-8k, and Zenmuse P1 all have an acceptable operating range and are able to focus at $1.08\,\mathrm{m}$. Looking at the inspection time and mass, the first three are still viable options. To account for improvements in the flight control subsystem (reducing the overshoot), the EVO nano+ camera is chosen to be used on the “DALE” drone. As this camera is able to focus up to a distance of $0.5\,\mathrm{m}$, the “DALE” drone would be able to reach more confined places if the flight control subsystem is improved.

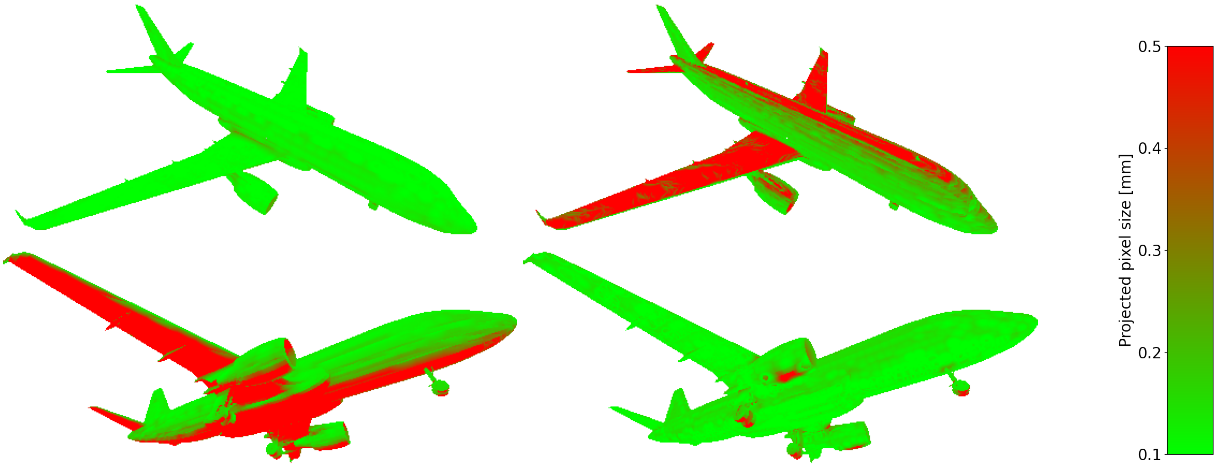
For the “CHIP” drone, the camera should have a large operating range, and be far from the aircraft surface (the “CHIP” drone has a large frame and high mass). The DJI Mavic 3 tele camera, and Zenmuse P1 have the largest operating ranges. The inspection time of the DJI Mavic 3 tele camera is significantly longer than the inspection time of the Zenmuse P1. However, its mass is lower than the Zenmuse P1. The long inspection time for the tele camera will result in more drones for the swarm, significantly increasing the total swarm mass. It was assumed that the high mass of the Zenmuse P1 would still result in a lower total mass for the swarm because of the lower amount of drones needed. For this reason and the fact that it is used by other companies for aircraft inspection with drones[17], the Zenmuse P1 is chosen to be the best option for the “CHIP” drone.

[17] Retrieved from: `https://mainblades.com/` (cited 7 June 2022)

4.3. Coverage map

An estimation of how well the aircraft will be covered by the drones can be made using the cameras selected in Section 4.2. Each drone type will be able to inspect other areas compared to the other drone, due to the drone frame design in chapter 8. This design limits the field of view of the cameras from -90° to 10°, and -30° to 90°, for the "CHIP" drone and "DALE" drone respectively.

To determine what areas of the aircraft can be expected by each drone type and how well they can be expected, a coverage map is generated for both drones on a model of the Airbus A320. This coverage map, shown in Figure 4.5, visualises the projected pixel size on every point of the aircraft. The pixel size is visualised using a colour gradient between red (insufficient) and green (sufficient).



(a) Coverage map of "CHIP" drone.

(b) Coverage map of "DALE" drone.

(c) Colour legend for the coverage maps.

Figure 4.5: Inspection coverage map for "CHIP" drone and "DALE" drone.

As is expected from the field of view limits, the "CHIP" drone is mostly only able to inspect the top surfaces of the aircraft while the "DALE" drone is mostly suitable for bottom surface inspection. The areas around the landing gear and engine nacelles can be better inspected by the "DALE" drone. Vertical surfaces near the root of the wing show difficulty of inspection by the "CHIP" drone. As the "DALE" drone has a better field of view for vertical surfaces and it is smaller than the "CHIP" drone, a "DALE" drone might have to be used to inspect the root of the wing.

Both types of drones are assigned to the locations where they are most useful, inspecting as much of the aircraft as possible. The coverage maps in Figure 4.5 have been combined to form Figure 4.6, showing full coverage of the aircraft.

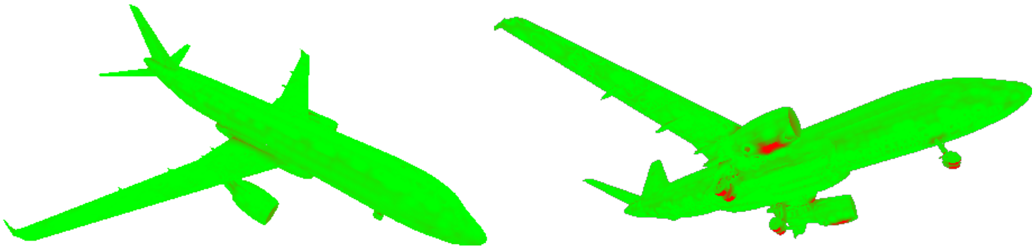


Figure 4.6: The full coverage map of the aircraft.

Based on this analysis, almost the complete aircraft can be inspected by the drones. The only places that still show difficulties are the landing gear wheels, and the underside of the engine nacelles. This is because these areas are located very close to the ground, therefore, the drones are not able to get close enough to inspect these areas.

4.4. Data Handling Analysis

In this chapter the data handling of the inspection process will be discussed. An overview of the data handling process is visualised in Figure 4.7. After the data has been gathered, i.e. a picture has been taken, the data is stored locally on the drone. When the inspection is complete, the Secure Digital (SD) cards are taken out of each drone, and are plugged into the ground station. After all the data has been transferred to the ground station, a check is performed to make sure all the data is present. If data is missing, a sub-selection of the swarm inspects the relevant parts. During packing and transport of the swarm or while waiting for the next aircraft to be inspected, the damage detection algorithm is executed. After all the data has been inspected by the algorithm, a damage report is created, and sent to the operator. Lastly, all the data is compressed and stored on a server for traceability, and will be accessible to the operator. The compression of the data, damage detection algorithm, and traceability will be discussed in more detail in their respective subsection.

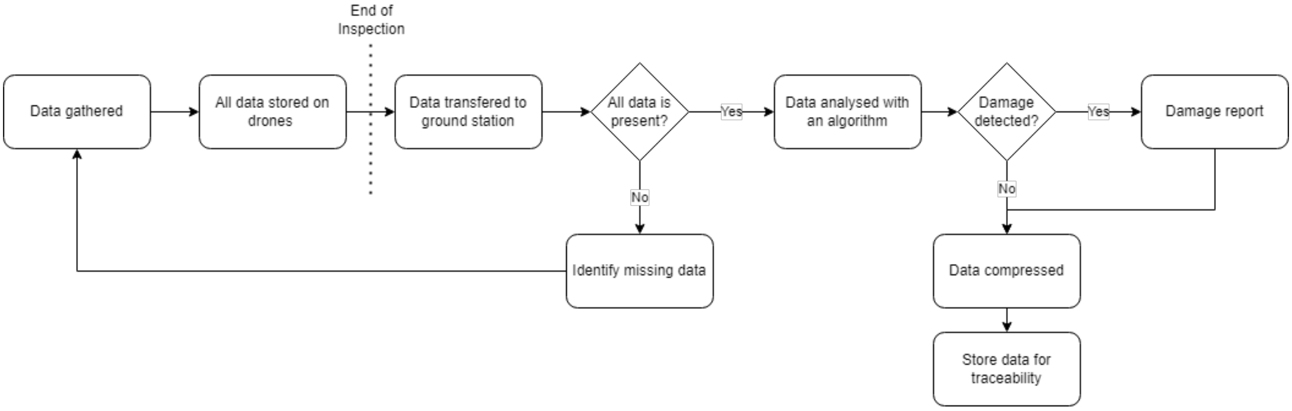


Figure 4.7: Inspection data handling flow chart.

4.4.1. Data Compression

The camera selected is a 10-bit camera, i.e. each pixel has 10 bits of information for each colour channel. An Red Green Blue (RGB) pixel will therefore contain 30 bits of information. As can be seen later in this section (see Table 4.3), it will lead to 105 Gigabytes (GB) of raw data, 72 GB of data is to be gathered by the “DALE” drones, and the rest 33 GB by the “CHIP” drones. This difference comes from the fact that “DALE” drones take almost twice as much images as “CHIP” drones, as was estimated by the simulation Section 5.4. Each “DALE” drone will then have to store at least 24 GB of gathered data, and each “CHIP” drone at least 16.5 GB of gathered data. Apart from that they will also have the collision algorithm, path planning algorithm, and the 3D-model. Therefore the “DALE” drones need 32 GB SD cards, while “CHIP” drones need 24 GB SD cards, to have enough memory for all the necessary data during the inspection. SD cards are lightweight, easy to implement and have very high write and read speed. They are commonly used in drones to store the data. If at some point more data has to be stored on the drone during the inspection it is easy to swap into SD cards with higher capacity.

On one side, 105GB might seem not that much for the whole aircraft being covered. On the other hand, for an algorithm to process the data, and for future storage of all the images for traceability, the amount of data stored for different aircraft will accumulate very fast. Therefore, some compression methods should be incorporated to reduce the amount of data storage needed.

The first decision that has to be taken on compression of the data, is whether a lossy or lossless compression should be incorporated. A lossy compression has higher compression ratios, but looses the quality/resolution of the image, and therefore valuable information. Lossless compression has lower compression ratios, but does not loose the quality of the image. Currently, the whole inspection procedure was based on having a high enough resolution to detect $1\,\mathrm{mm}$ cracks. Therefore, (for the algorithm as well as traceability) the best option is to go with lossless compression, as loosing data can lead to being unable to comply with the 1 to $10\,\mathrm{mm}$ damage detection requirement (SUB-INS-02).

The initial uncompressed image can be stored in different raw formats such as Tagged Image File Format (TIFF), Digital Negative (DNG), and Digital Picture Exchange (DPX). The size of these files depends on the amounts of bits used to represent a pixel. The cameras used in “CHIP” drone and “DALE” drone support both 8-bit and 10-bit colour-depths. As was calculated previously, the 10-bit colour-depth image will have a raw uncompressed size of 188.74Megabytes (MB), and 150.99MB for 8-bit colour-depth. Different methods of encoding for lossless formats exist, mostly relying on redundancy of the data. Therefore, it is difficult to estimate the compression ratios of lossless formats. The most popular compression formats used for images are Portable Network Graphics (PNG) and Joint Photographic Experts Group (JPEG).

The best estimation for the file size found is an online calculator[18] that roughly estimates the average size of the compressed image for a given resolution. The results from that calculator can be seen in Table 4.3.

Table 4.3: File size for different file formats.

Uncompressed Formats			Compressed Formats		
File format	**File size**	**Dataset size**	**File format**	**File size**	**Dataset size**
“CHIP” drone, 8192x5460px, 195 images					
TIFF/DPX/DNG RGB 3x8bit/pixel	134.18 MB	26.28 GB	Lossless PNG 24bit/pixel	25.99 MB	5.07 GB
			Lossless JPEG 24bit/pixel	67.09 MB	13.08 GB
TIFF/DPX/DNG RGB 3x10bit/pixel	167.73 MB	32.71 GB	Lossless PNG 30bit/pixel	32.54 MB	6.35 GB
			Lossless JPEG 30bit/pixel	83.87 MB	16.35 GB
“DALE” drone, 8192x6144px, 383 images					
TIFF/DPX/DNG RGB 3x8bit/pixel	150.99 MB	57.83 GB	Lossless PNG 24bit/pixel	29.25 MB	11.2 GB
			Lossless JPEG 24bit/pixel	75.5 MB	28.92 GB
TIFF/DPX/DNG RGB 3x10bit/pixel	188.74 MB	72.29 GB	Lossless PNG 30bit/pixel	36.62 MB	14.03 GB
			Lossless JPEG 30bit/pixel	94.37 MB	36.14 GB

As can be seen, JPEG preforms much worse than PNG. Unfortunately the calculator only calculates PNG and JPEG with 24 bit/pixel, which is 8-bit colour depth. However the compression ratio for 24bit/pixel and 30bit/pixel will be roughly the same, as the only difference is the amount of bits used to represent the colour. The ratio of the amount of information compressed will be the same. Therefore the data for 30bit/pixel compression is estimated by calculating the space saving of 24bit/pixel compression. For PNG it is $1-\frac{25.99}{134.18} = 0.806 = 80.6\%$, and for JPEG it is $1 - \frac{67.09}{134.18} = 0.5 = 50\%$.

Therefore, it can be seen that the maximum amount of data gathered per one inspection is exactly 105 GB, using 10-bit colour-depth. And it can be compressed with lossless PNG up to 20.38 GB. The actual possible compression should be verified when the system is deployed, as this calculation is an average approximation. The possibility of having 10-bit colour-depth increases the difference between colours and adds more information to process, the benefit of that can only be checked when the algorithm is developed. It is not necessary to store both 8-bit and 10-bit colour-depth images, as it is possible to compress the information from 10-bit to 8-bit colour-depth.

4.4.2. Damage detection algorithm

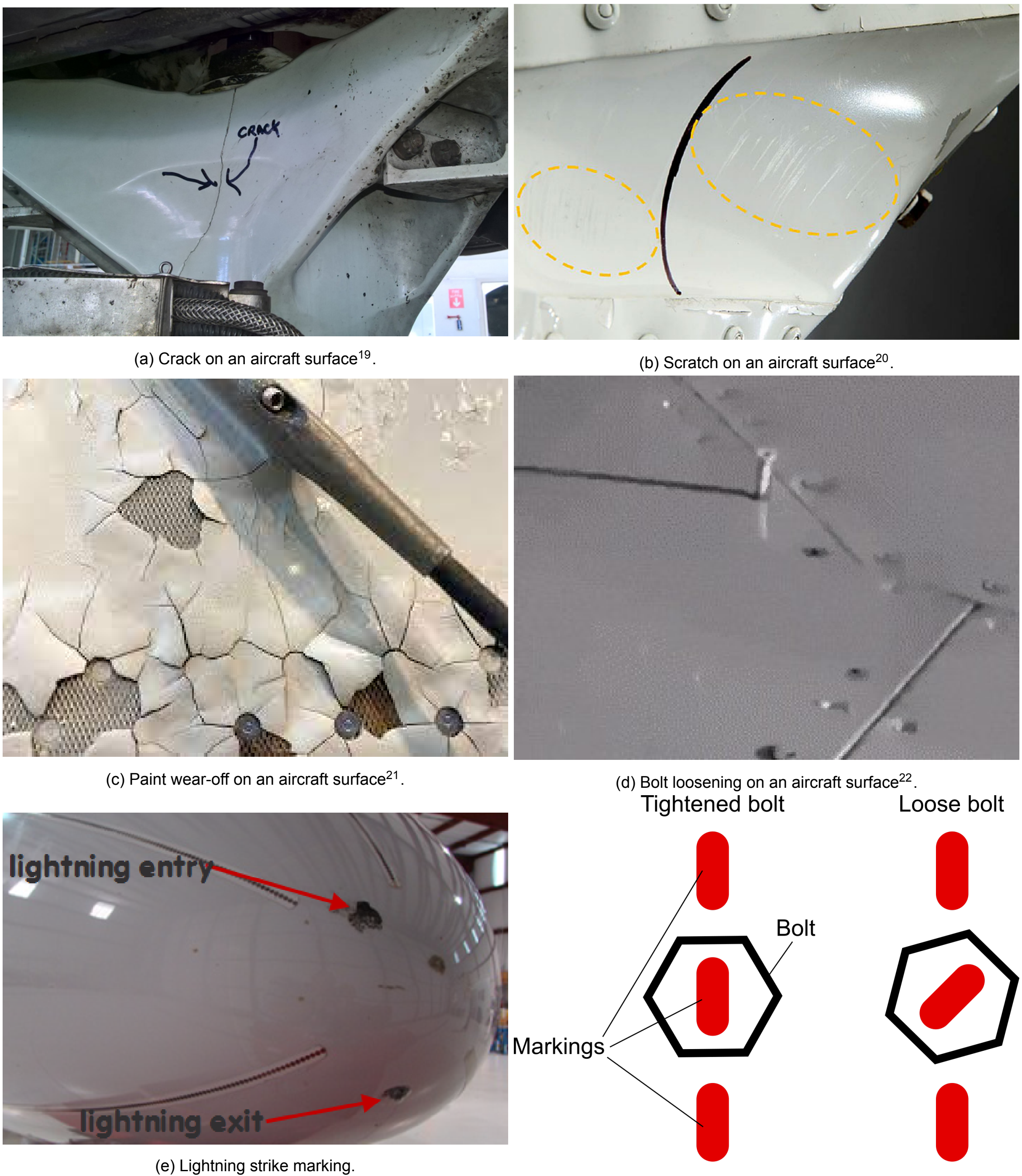
Due to the nature of this project, the unavailability of sufficient training data, the algorithm will not be created by the team. However, a possible implementation on how the damages can be detected is elaborated in this section.

Algorithm scope

First of all, the damages that are going to be detected have to be specified. With visual inspection it is only possible to identify visual damages such as cracks, scratches, paint wear-off, bolt/rivet missing, lightning strike markings, and bolt tightness. Examples of what these damages would look like can be seen in Figure 4.8.

[18] Retrieved from: `https://toolstud.io/photo/megapixel.php?width=8192&height=6144&calculate=compressed&compare=photo#compare` (cited 1 June 2022)

Figure 4.8f shows what is meant by bolt tightness, and how the tightness of a bolt can be detected using markings on the aircraft surface and bolt. The detection of the bolt damages is not verified in literature and therefore will have to be proven when the algorithm is developed.



(a) Crack on an aircraft surface[19].

(b) Scratch on an aircraft surface[20].

(c) Paint wear-off on an aircraft surface[21].

(d) Bolt loosening on an aircraft surface[22].

(e) Lightning strike marking.

(f) Bolt tightness detection.

Figure 4.8: Different visual damages on an aircraft surface.

In order to select a proper algorithm for damage detection, it is vital to first define the inputs, outputs, and overall scope of the algorithm. From the RGB images, current position, and camera specs, the algorithm shall

[19] Retrieved from: `https://www.flightsafetyaustralia.com/wp-content/uploads/2017/11/611750782.jpg` (cited 1 June 2022)

[20] Retrieved from: `https://www.tsb.gc.ca/eng/rapports-reports/aviation/2013/a13c0105/images/a13c0105-photo-04.jpg` (cited 1 June 2022)

[21] Retrieved from: `https://www.reuters.com/business/aerospace-defense/costly-airbus-paint-flaw-goes-wider-than-gulf-2021-11-29/` (cited 1 June 2022)

[22] Retrieve from: `https://www.askacfi.com/914/what-happens-when-aircraft-struck-by-lightning.htm` (cited 21 June 2022)

[23] Retrieved from: `https://i.stack.imgur.com/e06mo.jpg` (cited 1 June 2022)

detect the damages, their size, and position it on the 3D model of the aircraft. All the information on the position of the drone, timestamp, drone identifier, etc. can be stored in the metadata of the image. The damages can be detected from the images via Computer Vision (CV) and Machine Learning (ML) algorithms. Feasible algorithms are explained later in this section. When the damages are detected, the size of the damage can be easily computed by knowing the distance to the aircraft, and the size of the damage in pixels. The former is stored in the metadata, whilst the latter is the output of the damage detection algorithm. Taking FOV and the camera resolution, which are constant, trigonometry, the size of the pixel is defined by Equation 4.11-4.12. Multiplying the pixel size by the amount of pixels leads to the size of the damage in mm.

$$px_{size_h} = \frac{2d \cdot \tan \frac{\text{FOV}}{2}}{res_h} \tag{4.11}$$

$$px_{size_v} = \frac{2d \cdot \tan \frac{\text{FOV}}{2}}{res_v} \tag{4.12}$$

From the path planning and the position of the drone, the part of the aircraft that was captured can be determined, and therefore the image can be projected to the 3D model of the aircraft. This serves as 3D visualisation, which is discussed in Subsection 4.4.3.

Possible algorithms

After conducting a literature study, it was found that there are algorithms that detect cracks in concrete with CV and ML [43, 60], as well as detection of scratches and abrasions on metal surfaces [46]. These are the most similar problems found in literature. As can be seen from both papers, Neural Networks (NNs), and in particular Convolutional Neural Networks (CNNs), can be used. A CNN is a type of artificial neural network used in image recognition, and processing that is specifically designed to process pixel data[24]. It is proven that with enough training data, CNNs can reach a high accuracy in crack recognition [60]. From this paper it can also be seen that different architectures for the CNN can perform differently, with an accuracy range from 96.5-98%. Even though it can be seen that VGG-19 architecture performs the best for crack detection in concrete structures, different damage detection should be tested on aircraft structures. CNNs with different architectures should be trained and tested, to find the best architecture for each specific case.

It is vital to understand that aircraft is a complicated structure with different parts and components. The outer surface of the fuselage also has different coatings. This will affect the algorithms, as the images for different parts would be very different. Some parts also have different connections, as well as retractable parts, such as flaps, ailerons, brakes, windows, and doors. These connections can be seen as “cracks” in the structure, and lead to false positive errors. All these factors lead to a conclusion that different parts of the aircraft should be analysed separately, leading to several CNNs needing to be trained on different sets of data. Therefore the data should be separated into different segments with data for the specific parts. A separate ML algorithm has to be developed for that. The simple flow chart is shown in Figure 4.9.

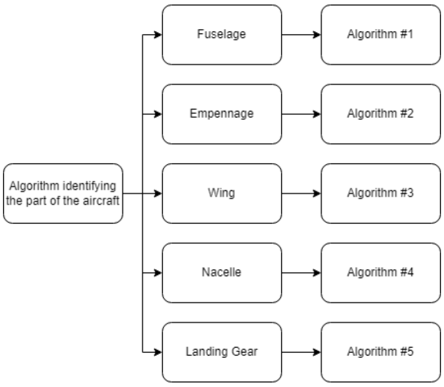


Figure 4.9: Algorithm selection flow chart, different aircraft segments require different algorithms.

As stated previously and as can be seen from literature [46], the majority of errors are going to be false positive errors. These can be easily dealt with by having a person check the images that were labelled damaged. False

[24] Retrieved from: `https://www.techtarget.com/searchenterpriseai/definition/convolutional-neural-network` (cited 31 May 2022)

negatives however are hard to deal with. However the probability of having a false negative error increases with decreasing damage size. Aircraft are designed to withstand damages of small size and currently with human inspection, the damages smaller than $5\,\mathrm{mm}$ are not repaired[25]. Even though the false negatives may be considered less important in this case, it is still a risk that has to be considered. As more data is gathered the algorithm can be developed to decrease the probability of having false negative errors. Identification of false negatives requires a manual inspection of the images. It is also possible to increase the sensitivity of the algorithm towards what is identified as damage. This will make the algorithm more biased towards false positives rather than false negatives, and therefore make it easier to deal with errors by verifying what was identified as damaged.

Before enough data is collected, the images would have to be processed and labelled manually. It is also possible to create an algorithm without ML by extracting features from the images with CV, analysing them manually, and then stating conditions that would help algorithm identify the damage. However this goes against the main purpose of using ML algorithms, which is using AI to find these conditions itself. Therefore the amount of time and efforts spent on such an algorithm would be worthless, as it will not reach the accuracy that CNNs provide.

Training data

In ML, the amount of training data available is the biggest struggle. To be able to train a precise and accurate algorithm, very large datasets are needed. As was stated previously, there is no public data available for aircraft visual inspection. A collaboration with an MRO could provide a solution, though most likely data shall be generated on-the-go. Therefore in order to train the algorithm, during the first inspections, the data would have to be processed manually to create a basis for the algorithm to be trained. When enough training data is collected and the algorithm is developed, the data gathered from further inspections can be used to improve the accuracy of the algorithm. Continuing collection of training data shall continue until the learning curve stabilises.

Several considerations have to be taken into account. As was previously stated, different parts of the aircraft are different in size and colour. This will affect the data, and how algorithm interprets it. Smaller parts will have less images gathered per inspection, for example empennage or the nacelle. More inspections are required to get adequate data on those parts. Probability of damage occurrence may vary over location and type of damage. Therefore even more training data is needed for the algorithm to accurately identify different types of damage. This leads to the conclusion that it is not possible to develop such algorithm at this point due to its complexity and unavailability of data. With more complex and nonlinear relationship problem, the algorithm will require more data.

On the other hand, there is a possibility to speed up the process of acquiring more data with the help of data augmentation. Data augmentation is a set of techniques to artificially increase the amount of data by generating new data points from existing data[26]. After collecting some data, more data with damages can be created by rotating, cropping and copying the already existing images. Another technique of data augmentation that could be used is creating a texture for the 3D model of the aircraft with the damages on it. As was explained in Subsection 4.4.1, by having this 3D model in a software such as Unity, images can be generated from the simulation. If the texture is detailed enough and the damages on it are similar to the real damages occurring on the aircraft structure, this data can be of sufficient quality for use. This generated data is called synthetic data, and as AI evolves, more synthetic data is expected to be used in ML[27].

Time consideration

It is difficult to estimate the time that would be needed to gather all the necessary data. It can be assumed that at least 10,000 images[28] are needed to have a first iteration on a ML algorithm. Assuming there are five different types of damages, and the majority of images having no damages on it, the assumption on the minimum amount of data can be increased to 100,000 images. Having more data with undamaged parts than damaged parts can lead to bias in the model. Therefore, if there is enough information on the damages, data augmentation has to be implemented to increase the amount of images with damages. Knowing that there will be 578 images per inspection (as estimated with the simulation Section 5.4), 173 inspections will have to be conducted to be able to perform a first training of the model. In the best case scenario where six inspection per day are conducted everyday (explained in Section 10.6), it will take at least 29 days to gather that amount of data. In reality it might be not possible to have six inspection per day everyday, and therefore more time will be needed. After the first training of the model, and first iteration on data augmentation it will become clear how

[25] R.M. Groves Private Communication, May 2022

[26] Retrieved from: `https://research.aimultiple.com/data-augmentation/` (cited 31 May 2022)

[27] Retrieved from: `https://research.aimultiple.com/synthetic-data/` (cited 31 May 2022)

[28] Retrieved from: `https://machinelearningmastery.com/much-training-data-required-machine-learning/` (cited 2 June 2022)

much more work shall be performed for model validation.

From the experience of group members in their minor, it is known that a CNN with 14 layers, similar to VGG-16 architecture, can process 127 images/per second with image resolution of 500x500px, with the power of a personal laptop. The images with resolution of 8192x6144px can be split up into 202 images of 500x500px, while images with resolution of 8192x5460px can be split up into 179. This leads to $202 \cdot 383 + 179 \cdot 195 = 112,271$ images in total, dividing that by the speed of the CNN leads to total time of 14.7 min. This is a very rough estimation, in reality the CNN, most likely, will be able to process 1 image with 8192x6144px resolution faster than 200 images with 500x500px resolution. Comparatively, Mainblades process a similar amount of data in maximum 15 min after the inspection is done using Google Cloud Platform[29]. Hence it can be expected that the algorithm would be able to comply with the requirement on 15 min data analysis (SUB-INS-09).

4.4.3. Traceability of Data

After the data has been gathered, analysed, and compressed, it can be stored for traceability. One of the main advantages of the swarm of drones is the ability to digitise all the data gathered during inspection. The 3D model of an aircraft cannot only be used for other inspections of the same aircraft type, but also to create a 3D visualisation of the data collected during the inspection. Visualising the data in 3D will give the operator an intuitive feel on what the current state of the aircraft is. This also provides the possibility to use Augmented Reality (AR). A smartphone or tablet can be used to overlay the picture of the damage on top of the camera output, to show were exactly the damage was detected, even after it has been repaired. This concept of having a digital representation of a system (the aircraft) and its current state is called a digital twin.

Another advantage of traceability is being able to see the damage propagation over time. This can be used to create an algorithm to predict the damage propagation, which in turn opens up the possibility for predictive maintenance. Using the data gathered by multiple inspections, areas of the aircraft that are most vulnerable to damages can be determined. This information can then be sold to the aircraft manufacturer to be able to provide better damage protection in these areas.

4.4.4. Ground Station

Apart from swarm communication (explained in Subsection 5.3.3) the ground station is also responsible for the analysis of the data. It should gather the data from SD cards from the drones, compress the data, run the algorithm to identify damages, and run the software with traceability. Subsequently it shall project the images taken onto the 3D model of the aircraft, and the damage report. The majority of processing power will be used by the algorithm for damage detection, and projecting the images on the 3D model. The exact processing power required is difficult to estimate, and would have to be tested when the software is developed. Most likely, it would be possible to run it on mid/high-performance laptop. The other possibility is to use cloud servers such as the Google Cloud Platform. Use of a cloud server however requires a stable connection with high upload/download speeds, as lots of data will have to be transferred, which will take additional time.

[29] Mainblades private communication, June 2022

5

Swarm Control

In this chapter, the design of the swarm control subsystem is described. The swarm control subsystem is responsible for the control of the drones during the inspection. This entails the task allocation for the drones, the drone algorithms and the communication between the drones. These functionalities are elaborated upon in Section 5.1, where the requirements of the swarm control subsystem are laid out. The task allocation over the drones is discussed in Section 5.2. From the task allocations follow the algorithms and communications that the drones need to execute the tasks. These algorithms and the associated communication these items are categorised as the drone component of the swarm control subsystem and are discussed in Section 5.3. A simulation of the swarm control subsystem is performed and is detailed in Section 5.4. Finally, verification and validation is performed in Section 5.5.

5.1. Requirements

The tasks the swarm control subsystem needs to perform are all the operations that fall under the “aircraft inspection” block of the global operations and logistics flow chart presented in Figure 10.2. A more detailed version of this block which contains all the specific functions of the swarm control subsystem is provided in Figure 5.1. The individual entries of this diagram will be addressed in different sections of this chapter. Here, the main lines of the diagram will be explained to provide an outline of the functioning of the swarm control subsystem.

The diagram is divided into three main blocks. The first two blocks occur when an aircraft type has never been inspected before. When a new aircraft type needs to be inspected a manual fly-around is first performed with a single “CHIP” drone and a single “DALE” drone consecutively. These flights are performed in order to construct a 3D map of the aircraft type that has to be inspected. From this 3D map, the second block, which is the task allocation and trajectory generation, can be entered. The task allocation is the division of the inspection over the drones. The trajectory generation is tightly coupled with the task allocation and generates the trajectories the drones have to follow while performing the tasks. When these trajectories are complete they are handed over to the drones, which can now perform the inspection. The following functions of the swarm control subsystem are then categorised as the drone component of the swarm control subsystem. This drone component covers the positioning, communication and collision avoidance systems of the drones. This should not be confused with the flight control subsystem (discussed in chapter 6). The flight control subsystem will convert the commands from the swarm control subsystem into motor commands for the drones.

The swarm control subsystem needs to ensure the correct execution of this inspection and needs to guarantee that the inspection can be finished if a single drone fails. The requirements that flow from these functions are discussed next, while the design solutions that will meet those requirements (the light blue boxes in Figure 5.1) are discussed in the following sections.

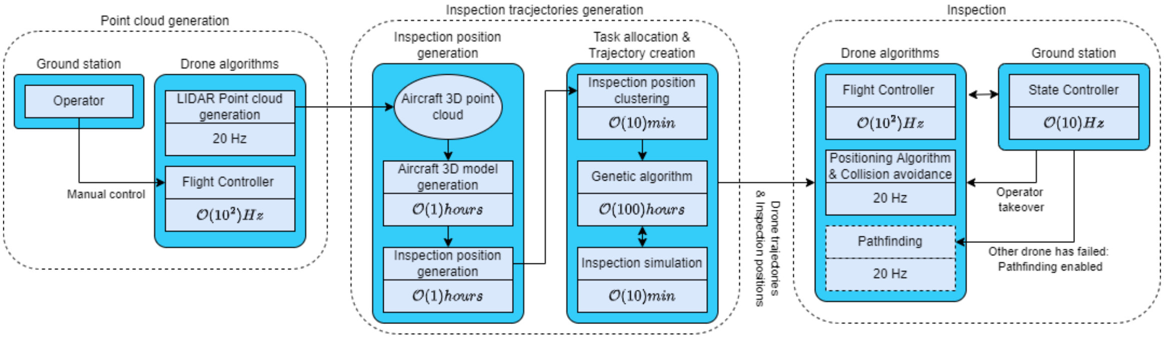


Figure 5.1: Overview of the swarm control subsystem functionality.

The requirements that flow from the functionality are shown in Table 5.1. The requirements and responsibilities of the swarm control system can be categorised into four broad categories. The first category belongs to the task distribution of the inspection over the drones while the other three categories belong to the drone component of the swarm control subsystem. The requirements for these categories will be discussed in this order.

Task Allocation requirements

The task allocation system has a lot of degrees of freedom and uncertainties. Due to these large uncertainties, it was chosen not to include any requirements on the task allocation system at this point in the design.

Positioning Requirements

The positioning system requirements are sized based on the capabilities of already proven systems. For a positioning system on a drone in a GPS-denied environment (such as an aircraft hangar) an accuracy higher than $10\,\mathrm{cm}$ is on the leading edge of what is possible [31]. Therefore, this accuracy is chosen as a requirement that the positioning system will need to achieve (SUB-SWC-POS-1). This is for the *absolute* position of the drones. The positioning system should provide the *relative* position of the drones to objects in its environment at a higher accuracy of $1\,\mathrm{cm}$ (SUB-SWC-POS-2). This requirement is based on the higher accuracy required for the collision avoidance subsystem. The increased accuracy was chosen on one order of magnitude but can be refined later. The positioning system shall also update the absolute and relative position of the drone at a rate of $20\,\mathrm{Hz}$ (SUB-SWC-POS-3). This is similar to systems that are already proven [31].

Collision Avoidance Requirements

The requirements for the collision avoidance system have all been set in previous steps of the design process (see [23]). The requirements are that the drones shall maintain a minimum distance of $0.5\,\mathrm{m}$ to the aircraft and other surfaces (excluding the ground) and a $2\,\mathrm{m}$ distance to humans. The collision requirement is set at an occurrence of less than 1 in 1000 from the stakeholder requirement SYS-REL-07.

Communication Requirements

The communication requirements are given in requirements SUB-COM-1 through SUB-COM-6. The first requirement of the communication subsystem is derived from the requirement that a human operator can take control of a drone. In order to stream HD video from the drone, a link speed of at least $3\,\mathrm{Mbit\,s^{-1}}$ is required using H.256 video encoding [63]. SUB-COM-2 was determined based upon literature where a 99% link availability was deemed necessary for a swarm of drones [1]. SUB-COM-3 is based upon the dimensions of the Boeing 757-300 which is the largest narrow-body aircraft in service. Using the distance between the extreme vertices of the bounding box of the 757-300 gives a required range of communication of at least $67\,\mathrm{m}$. However, this communication link is not guaranteed to have a line of sight but it can be assumed that plenty of other indirect paths are available for the signal after only a single bounce on the environment. Based on this assumption a 40% margin is added to the range to account for indirect paths which makes the final required range $95\,\mathrm{m}$ as in SUB-COM-4. In order to avoid a lengthy regulatory process, the communication system shall be limited to license-free frequencies as in SUB-COM-5.

Requirements one through five limit the frequency ranges to the $2.4\,\mathrm{GHz}$ or $5.8\,\mathrm{GHz}$ frequencies. Using lower frequencies would make it difficult to meet the bandwidth requirements and higher frequencies require licenses to operate in Europe [28]. There are extensive proven communication solutions in the $2.4\,\mathrm{GHz}$ or $5.8\,\mathrm{GHz}$ frequency ranges. The communication subsystem will follow the IEEE802.11g WLAN standard (which provides the basis for the WiFi brand) in these frequency ranges [36]. SUB-COM-6 is derived from this standard [37]. For redundancy purposes, SUB-COM-7 states that a separate communication link will be established for the manual control of the drone.

Table 5.1: Swarm control subsystem requirements.

Identifier	Requirement	Related System Requirements
SUB-SWC-1	The swarm control subsystem shall provide a collision avoidance system.	
SUB-SWC-2	The swarm control subsystem shall provide a positioning system.	
SUB-SWC-3	The swarm control subsystem shall enable a human operator to take control of the swarm.	
SUB-SWC-4	The swarm control subsystem shall distribute the inspection tasks over the drones.	
SUB-SWC-POS-1	The swarm control subsystem shall be able to locate the drones with an Root Mean Squared Error (RMSE) of less than $10\,\mathrm{cm}$.	SYS-SAF-01, SYS-SAF-05

SUB-SWC-POS-2	The swarm control subsystem shall provide distance measurements from objects in the environment with an RMSE of less than $1\,\mathrm{cm}$	
SUB-SWC-POS-3	The swarm control shall update the positions of the drones at $20\,\mathrm{Hz}$.	StRS-FO-02
SYS-REL-07	The swarm control subsystem shall ensure that no more than 1 collision between drones happens every 1000 inspections.	
SYS-SAF-01	The system shall not come closer than 0.5 meters to the inspected aircraft	SYS-SAF-05
SYS-SAF-05	The system shall at all times keep at least 2 meters distance from humans	SYS-SAF-05
SUB-SWC-COM-1	The communication subsystem shall achieve a minimum communication bandwidth of $3\,\mathrm{Mbit\,s^{-1}}$.	SUB-SWC-3
SUB-SWC-COM-2	The communication subsystem shall maintain a link availability above 99%.	
SUB-SWC-COM-3	The communication subsystem shall provide a link over a range of at least $67\,\mathrm{m}$ under no line-of-sight conditions.	
SUB-SWC-COM-4	The communication subsystem shall provide a link over a range of at least $95\,\mathrm{m}$ under line-of-sight conditions.	
SUB-SWC-COM-5	The communication subsystem shall not require a license to comply with regulations.	
SUB-SWC-COM-6	The communication subsystem shall maintain a link Bite Error Rate (BER) below 10^{-5}.	
SUB-SWC-COM-7	A redundant communication link will be provided for drone control	

5.2. Task Generation and Allocation

In this section, the problem of finding the best way to inspect the aircraft is addressed. First, the task generation is covered in Subsection 5.2.1. This task generation covers the creation of the camera positions that will have to be achieved by the drones in order to inspect the aircraft. The second problem is in what order to visit these positions, the task allocation. The problem is essentially an Multi Travelling Salesmen Problem (MTSP) [5] comprised of two parts: allocating each inspection point to a drone and finding the optimal path between those points. The two parts can be addressed in different ways, either together with a Conflict Based Search (CBS) [64] or separately, with a Genetic Algorithm [44] for task allocation, followed by an A* algorithm [1] for pathfinding. The selection between these two algorithms will be addressed with a trade-off in Subsection 5.2.2.

5.2.1. Task Generation

In order to perform the inspection, it must be known at what points the pictures of the aircraft have to be taken. From chapter 4 it is already known which areas on the surface of the aircraft are inspectable by the drones. Now the method of how to turn those surface areas into desired camera positions will be discussed. This is performed using two interlinked algorithms. The first algorithm discretises and manages the area that has to be inspected and the second creates the camera positions for those discretised elements.

The first algorithm is shown in Algorithm 1. It takes the area to be inspected and distributes points on this area with a uniform spacing d. These points are used as samples to determine if the underlying area is fully covered by the inspection. The lower the spacing d the more accurate the resulting camera positions will cover the underlying area but the more intense the computational requirements. In order to start the process of creating the camera positions that cover these points, the points get added into a queue which is a First In First Out (FIFO) structure. The algorithm iterates over this queue until it is empty which means that the entire inspectable aircraft surface will have been covered by the cameras. When the iteration is performed a point is taken from the front of the queue and it is checked how many times it has been considered by the algorithm before. It will be attempted to create a camera position that can see the current point using Algorithm 2.

This second algorithm takes an allowed overlap as an argument. The allowed overlap determines how much

[1] Retrieved from: `https://www.geeksforgeeks.org/a-search-algorithm/` (cited 17 June 2022)

overlap the photo that contains the current point may have with other photos. based upon the number of times a certain point has been visited this allowed overlap is calculated by Algorithm 1. In the current implementation, this calculation is performed based upon a linear function which depends on the maximum visits a point may receive. For example, if the points may be visited a maximum of four times globally, the allowed overlap on the first visit to the point is 0%, on the second 25% and so on. Depending on the success of the second algorithm the points that can be seen by the newly created camera position are removed from the queue. If creating a new camera position for these points with the current constraints is infeasible, the visit to this point is counted and the point is added to the back of the queue again. This results in the entire inspectable area being considered in multiple stages. In the first stage all the points will be visited and only photos which have no overlap with other photos are allowed. When the complete surface area has been considered in this way, a second pass will now allow the creation of photos that have a 25% overlap with the previously generated photos.

Algorithm 1 Algorithm for determining inspection positions from areas which have to be covered

```
$inspectableArea \leftarrow GetInspectableArea(aircraft)$
$points \leftarrow DiscretiseInspectableArea(inspectableArea, d)$
$pointsToDistribute \leftarrow Queue(discretisedInspectableArea)$
**while** not $pointsToDistribute$ is empty **do**
    $pointToCheck \leftarrow pointsToDistribute.Pop()$
    $allowedOverlap \leftarrow GetAllowedOverlap(pointToCheck.visits)$
    $succes, visiblePoints \leftarrow ApplyDroneCamera(pointToCheck, allowedOverlap)$
    **if** succes **then**
        **for all** point in visiblePoints **do**
            $point.claimed \leftarrow True$
            $pointsToDistribute.Remove(point)$
        **end for**
    **else**
        $pointsToDistribute.AddAtTheBack(pointToCheck)$
        $pointToCheck.visits \leftarrow pointToCheck.visits + 1$
    **end if**
**end while**
```

The second algorithm takes the point it receives for the first algorithm and tries to construct the best camera position for this point. The best camera position is defined by a score function given by Equation 5.1. In this equation, $S_{total}(C)$ is the score of a camera-position combination C. This score is a function of five variables. The first function is a function of the minimum distance d to other objects. The score of a point is lower the closer the distance is to the minimum distance and is negative infinity when the minimum distance requirement of a drone would be violated by the given camera position. The second function is a function of the sum of the angles $\Delta\theta$ between the camera view and the points (n_C) it can see. If it is looking at all the points from a very oblique angle, the score that this function returns is low, and it is maximum, when the camera would be at infinity looking at all the points through their normals. The third function is a function of how many points the camera can see. The more points it can see, the better. The fourth is a function of the overlap O with other photos. This overlap should be minimised and cannot be more than the allowed overlap. The fifth and last variable is the angle of the camera with the horizontal α_c. This angle discourages the “CHIP” drone from looking up and the “DALE” drone from looking down. At the end of the score calculation, the score is scaled by a factor B_{type} which depends on the drone type (this is be elaborated upon later).

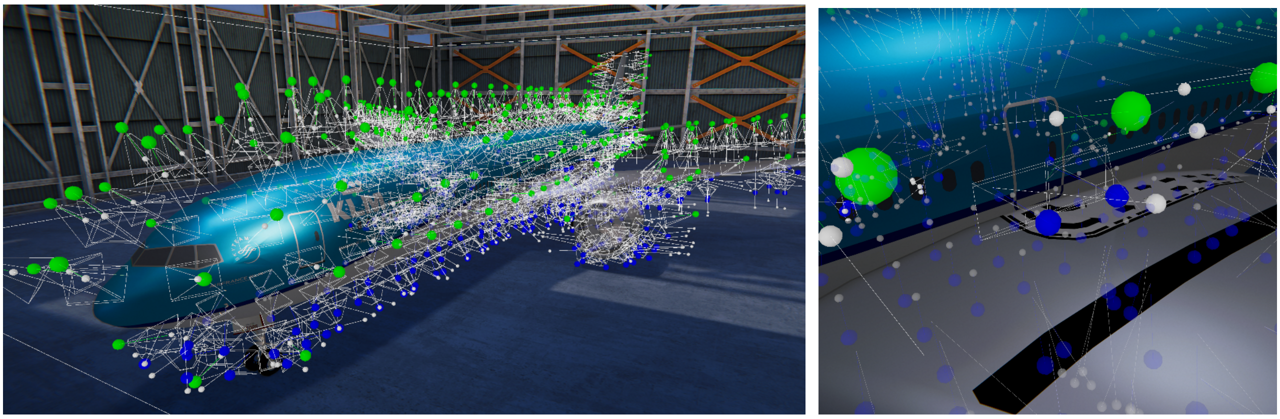
$$S_{total}(C) = \left(S_{distance}(d_C) + S_{normal}(\sum_{i}^{n_C}(\Delta\theta_{(i|C)}) + S_{amount}(n_C) + S_{overlap}(O(n_C)) + S_{horizontal}(\alpha_c) \right) \cdot B_{type} \tag{5.1}$$

The possible camera positions for a given point are evaluated in a conical volume around the normal of the point that has to be checked (the one given by Algorithm 1). For all these potential cameras the costs are calculated. For each position, multiple cameras are evaluated for every drone type given that the cameras for the different types have different properties. When these costs have been calculated at every position for every drone type the best scoring camera position is taken for every drone type. The best camera values of each drone type are then compared and the drone type with the highest camera value is then selected. This is why the factor B_{type} is important, it can give a bias towards a certain drone type when the obtained ratio of points skews too hard in a given direction. For example, if there are ten “CHIP” drones and only one “DALE” drones but there would be more small drone photos than large drone photos the bias would need to be adjusted to favour the “CHIP” drones. This way, some of the large drone photos will be converted to small drone photos.

Algorithm 2 Algorithm $ApplyDroneCamera$ for determining the camera position for a point

```
$positionsToCheck \leftarrow getPositionsToCheck(pointToCheck)$
for all $positionToCheck$ in $positionsToCheck$ do
    for all $droneType$ in $droneTypes$ do
        $visiblePoints \leftarrow GetVisiblePoints(points, positionToCheck, droneType)$
        if CalculateOverlap($visiblePoints$) < allowedOverlap then
            $positionsToCheck.droneType.visiblePoints \leftarrow visiblePoints$
            $positionToCheck.droneType.value \leftarrow CalcValue()$
        end if
    end for
end for
for all $droneType$ in $droneTypes$ do
    $sortedPositions \leftarrow SortByValue(positionsToCheck, droneType)$
    $droneType.bestPositions \leftarrow sortedPositions[0]$
end for
$bestDroneAndPositionCombination \leftarrow SortByValue(droneTypes.bestPositions)$
if $bestDroneAndPositionCombination$ exists then
    $succes \leftarrow True$
    $visiblePoints \leftarrow bestDroneAndPositionCombination.visiblePoints$
else
    $succes \leftarrow False$
end if
```

The resulting distribution of inspection positions is shown in Figure 5.2, where the green points represent the camera positions for the “CHIP” drones and the blue points represent the camera positions for the “DALE” drones. As expected, the large drones inspect the top of the surface of the aircraft while the small drones inspect the points with little spacing or for which an upwards camera angle is required. It is interesting to note that the camera positions along with the fuselage line up in lines aligned with the fuselage direction. There is no functionality in the algorithms that specifically creates this arrangement of the points. This behaviour emerges purely from the symmetries of the geometry of the aircraft. The main strength of this algorithm lies in its flexibility. It can be applied to any general 3D structure which needs to be inspected in combination with any number of types of cameras which will be used for the inspection.



(a) Camera distribution over Boeing777. A subset of 508 “CHIP” drone camera positions are shown in green and a subset of 858 “DALE” drone camera positions are shown in blue.

(b) Here “DALE” drones were preferred by the algorithm due to the tight spacing around the wing.

Figure 5.2: The resulting distribution of “CHIP” drone and “DALE” drone camera positions obtained using Algorithm 1

5.2.2. Task Allocation

Dividing the inspection positions over the drones is a task allocation problem that can be classified as an MTSP. Many solutions exist to solve this problem and a trade-off is performed in order to select the best method for the current application. The algorithms selected for the trade-off are CBS, a genetic algorithm, and a clustering algorithm. The CBS algorithm is based on the extended CBS algorithm adapted from Hönig et al. [34]. This algorithm performs task allocation and pathfinding with collision avoidance by dividing the spatial and time dimensions into a grid and executing the algorithm on this structure. A genetic algorithm similar to the one proposed by Király and Abonyi [44], is also considered. A genetic algorithm works by creating multiple permutations of possible solutions to the task allocation problem and computing their fitness. Based on this fitness the solutions evolve over time. This algorithm purely performs task allocation and any pathfinding needs to then be performed afterwards. The number of conflicts resulting from this algorithm can be accounted for in the fitness measure. A cluster-based approach simply divides the tasks that the agents need to perform into clusters and divides these over the agents (such as in [45]). In the current problem, this would amount to constructing certain regions on the aircraft to inspect for every drone.

To choose what algorithm is the most suitable for the swarm of inspection drones, three trade-off criteria were chosen. These were the ease of implementing the algorithm, the quality of the solution, the obtained solution and the scalability and computational complexity of the algorithm. Each criterion was deemed to be of equal importance, therefore the weight assigned was of $\frac{1}{3}$ for each chosen criterion.

Table 5.2: The trade-off for the swarm control algorithm task distribution algorithm. Meaning of scoring and colours: green (3) = exceeds required performance, blue (2) = meets required performance, red (1) = below required performance. Overall score: 1-1.5 (RED), 1.5-2.5 (BLUE), 2.5-3.0 (GREEN).

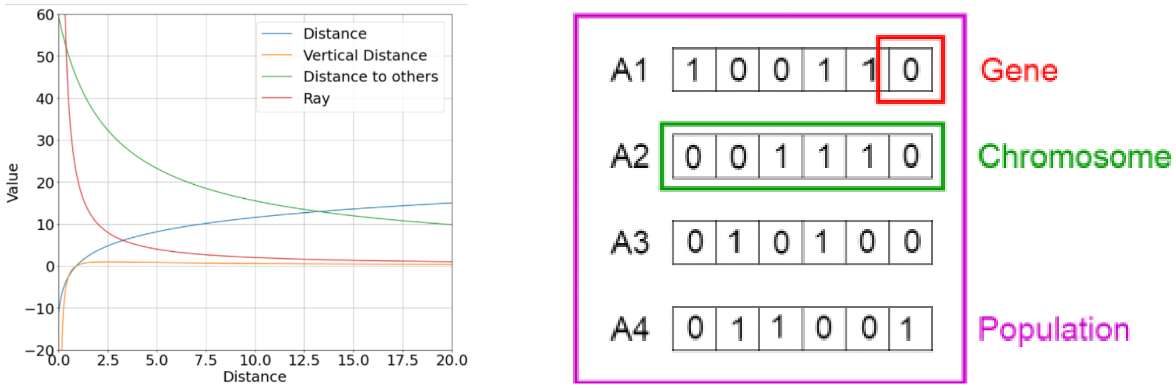
Option	**Ease of implementation**	**Scalability & Computational Complexity**	**Quality of solution**	**Average score**
CBS	Hard to implement in the DSE time	Can be extreme depending on the dimensionality of the problem and the spatial resolution	Takes into account drone paths, task allocation and conflicts.	
Score	1	1	3	**1.67**
Colour	RED	RED	GREEN	BLUE
Genetic Algorithm + A*	Easy to implement	Depends on the desired quality of the solution. The quality of the solution is proportional to the amount of computation time employed.	Converges rapidly to a local maximum. Can take conflicts into account indirectly.	
Score	3	3	2	**2.67**
Colour	GREEN	GREEN	BLUE	GREEN
Cluster Based Approach	Moderately hard to implement	Computationally cheap	Solutions are simple and do not take drone conflicts into account.	
Score	2	3	1	**2**
Colour	BLUE	GREEN	RED	BLUE

From Table 5.2 it appears that the Genetic Algorithm is the best fit for this project. It was mostly chosen due to its relative ease of implementation. The selection was also backed by the literature, as Cheikhrouhou and Khoufi [16] found the Genetic Algorithm the best algorithm to solve the MTSP.

To provide a starting point for the genetic algorithm the task allocation is first performed using a greedy task allocation algorithm. This algorithm aims to minimise the cost represented in Equation 5.2 for every drone during the entire inspection. This is done by computing Equation 5.2 for each point for each drone. All the drones then select the point with the lowest cost for them. During the inspection, this calculation is constantly performed by the drones. The drones then select the cheapest point for them in real time. In, Equation 5.2 the cost function for a drone i relative to one point P is given. In this equation, C is the cost, d the distance, and f_r is the ray function, which has a value of 0 if there is a direct line of sight between the drone and the point P and 1 if there is not. The used cost functions are plotted in Figure 5.3a.

$$C_{total}(i|P) = C_d(d_{(i|P)_{xy}}) + C_v(d_{(i|P)_z}) + \Sigma_{j=0,j\neq i}^{n_{drones}-1} C_d(d_{(j|P)}) + f_r C_r(d_{(i|P)}) \quad (5.2)$$

The ray function is used to favour straight paths over ones that will need to go around the aircraft, it can be seen from Figure 5.3a how the ray value is higher for closer points. This is to account for the fact that a straight line of travel is more important if the destination is close than if the destination is far away. The function also favour points close in vertical distance, as it is more convenient for a drone to travel horizontally rather than vertically.



(a) Local task allocation function for the initialisation of the genetic algorithm.

(b) Genetic algorithm iteration representation.

Figure 5.3: Figures on the local task allocation and the task allocation using the genetic algorithm.

The genetic algorithm is then used to improve the solution. A simulation is run in Unity (see Section 5.4) to get the starting distribution of tasks using the greedy algorithm and the associated inspection time. Then the genetic algorithm is used to shuffle the tasks which generates a population of newly allocated tasks. The simulation is then rerun on this new population, and if a better time is found, it will become the base for successive iterations. A representation of one iteration of a generic genetic algorithm can be found in Figure 5.3b. The genetic algorithm used is based on performing complex mutations as a combination of simple mutations. The simple mutations can be *in-route*, so a mutation that influences only the orders of the genes in a chromosome, or *cross-routes*, meaning that genes switch between different chromosomes. The *in-route* mutations are flip and reverse. Flip switches the position of two genes inside a chromosome while reverse reverses the order of a chromosome. The *cross-routes* mutations are slide, crossover and swap. Slide attaches the last gene of a chromosome to the front of another chromosome. Crossover swaps the first k genes of chromosomes with the first k genes of another chromosome, where k is a random number. Finally, swap is a generalised crossover that swaps sections of genes of different lengths and positions between chromosomes.

Furthermore, some constraints for the algorithm were set. The first constraint is that the points inspectable by “CHIP” drone and “DALE” drone should be disjoint sets such that no *cross-routes* mutation between different types of drones are possible. The second constraint is that each drone shall inspect no less than 25% less points than the total number of points allocated to its type divided by the number of drones of its type. The third constraint is the flip side of the previous one. A drone shall inspect no more than 25% more points than the total number of points allocated to its type divided by the number of drones of its type.

The iterations of the genetic algorithm produced a 15% improvement in inspection time for a number of tasks under 100 after 4000 simulations were run. However, when the number of tasks got scaled above this number the improvements became very sporadic. The full inspection of a Boeing 777 contains 1250 tasks. It is hypothesised that the resulting search space is too large for the genetic algorithm to traverse effectively. The algorithm executed in [16] used 32 000 simulations to optimise 20 tasks. In the current case, a full inspection simulation takes around ten minutes to complete. This means that only 24 simulations can be run in an hour (assuming four parallel simulation executions which is the maximum with the compute capacity available to the authors). By those numbers, it would take 228 years for the current algorithm to find a solution of similar quality (assuming the number of required simulations scales linearly with the number of tasks, it is likely much worse). The only methods to improve the current results without changing the algorithm would be to speed up the simulation or to increase the computational resources.

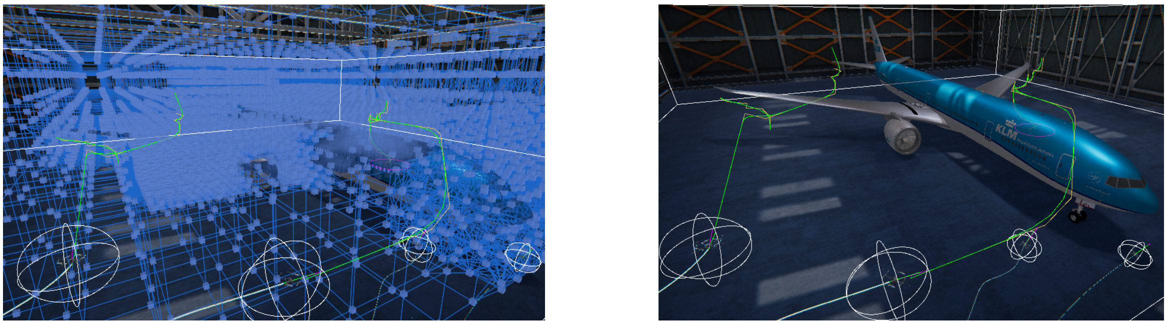
Due to this result, a clustering algorithm will have to be implemented to group neighbouring inspection positions to reduce the number of tasks. The 1250 tasks can then be grouped to scale the number of grouped tasks to be in the range of 50 clusters. It is hypothesised that in this way the genetic algorithm can traverse the search space again.

5.2.3. Trajectory Generation

The algorithm in Subsection 5.2.2 is used to select the best possible task allocation. However, such an algorithm does not take care of the specific path that each drone has to follow between two points. For that, an A* algorithm is used. This algorithm minimises the total cost of the path and is shown in Equation 5.3. Where $f(n)$ is the total cost to be minimised, $g(n)$ is the cost from the start to node n, and $g(n)$ is an estimation of the cost from node n to the end. The function $h(n)$ takes into consideration the cost of the segment to the following node and the distance from the end to that node. The graph is then explored using a greedy algorithm which explores the nodes based upon the minimum cost node at every point.

$$f(n) = g(n) + h(n) \tag{5.3}$$

The A* algorithm was implemented in the simulation based upon the A* pathfinding project [2]. Figure 5.4 shows the results of the pathfinding of the drones around the Boeing 777.



(a) The graph that will be traversed by the drones using the A*. Some nodes were removed for clarity.

(b) The tracks generated by the A* algorithm of the drones (green). The drones are highlighted with white circles.

Figure 5.4: A* algorithm applied to the inspection area around a Boeing 777.

5.3. Drone Components

Now that the task generation methods are known, the drone component of the swarm control subsystem can be addressed. First, the positioning algorithm and architecture for the drones are selected. After this, the collision avoidance algorithm is considered and the communication system is sized. Finally, after the sizing of the drone systems, the hardware that these systems will employ is selected.

5.3.1. Positioning Algorithms

In this section, the positioning systems of the drones will be considered. First, the general architecture of the algorithms will be shown. After this, the algorithms will be considered in detail and components will be selected to perform the positioning.

Positioning Architecture

In the midterm report, it was decided that the positions of the drones will be determined onboard the drones themselves, without relying upon outside information [23]. This requirement narrows down the design space of the positioning system dramatically. There are three levels on which the position of the drone can be measured. These are the three derivatives of the position. The zeroth derivative is the position itself, the first derivative is the velocity and the second derivative is the acceleration. These three levels are shown in Figure 5.5. The positioning systems will have sensors on all these levels. It could seem logical that, if sensors were to be placed on the bottom level, the position would be known instantly and sensors on the higher level would not be necessary. However, this is not the case because the sensors on this level can only measure distance and not position directly. In very simple cases, these two are the same. For more complex geometries, obtaining the position from distance measurements alone is not straightforward. Because the environment in an aircraft hangar is geometrically complex, it is currently intractable to obtain a direct position estimation directly from the distance measurements at a sufficiently fast update rate. This is why the derivatives of the position also need to be sensed. To obtain the position from these higher derivatives, integration is used. This process is called odometry or dead reckoning. There are further advantages to measuring multiple derivatives of the position.

[2] Retrieved from: `https://arongranberg.com/astar/docs/` (cited 14 June 2022)

These are that the sensors for the different derivatives are generally complementary to each other. The higher the derivative the higher the update rates of the sensors that can measure those derivatives.

For onboard measurement on drones, there is only one feasible method to measure the accelerations of the drones. This is measuring the acceleration using Inertial Measurement Unit (IMU)s. IMUs use Micro-Electromechanical System (MEMS) to provide accurate acceleration measurements at update rates of hundreds of Hz. For measuring the bottom two levels there are two established methods for drones. These are measuring the distances and velocities using LIDAR or using optical sensors. A LIDAR uses laser pulses and the speed of light to detect the distance to objects, while optical sensors can use the parallax effect. Optical sensors of this type are generally called stereo cameras or depth cameras. These sensors are sometimes also equipped with an extra structured light sensor to provide even more accurate distance measurements. In the current architecture, both sensor types will be used due to their complementary nature. LIDAR sensors perform really well at range and with great accuracy. They are however heavy, expensive and have a low resolution which makes detecting small or close objects, such as other drones, problematic. Because of this all-around coverage, using only LIDAR is not feasible and would not provide sufficient small object detection in the first place. This is where the stereo cameras shine. While they perform poorer at longer ranges and are generally less accurate, they can provide great all-around coverage and detect small objects at high refresh rates.

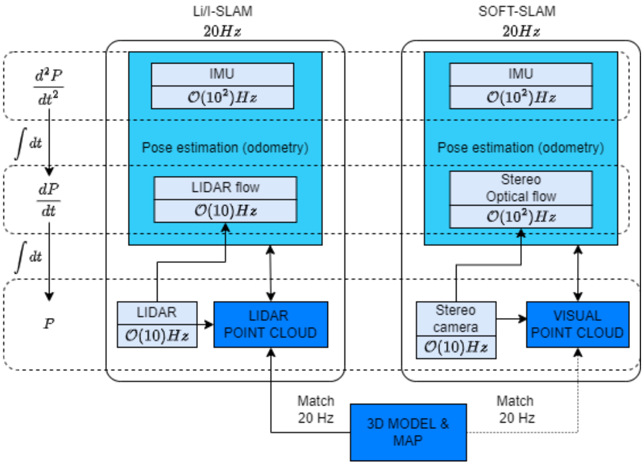


Figure 5.5: The architecture of the positioning system.

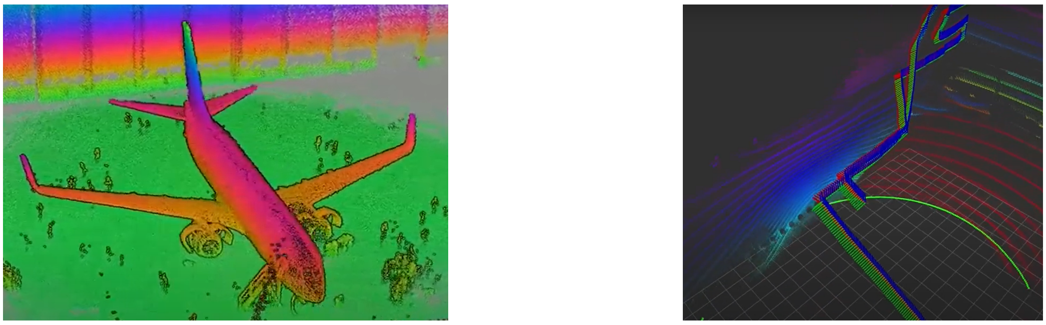
Both drones will have a LIDAR system, an all-around stereo camera system and an IMU. The position and attitude will be computed from the first two derivatives of the position using odometry. The sensor readings from these derivatives are merged to obtain a single state estimation using Kalman filters for both systems. However, the position obtained using this odometry approach is susceptible to large amounts of drift over time due to cumulative errors in the integration steps. To compensate for this drift SLAM is used. When SLAM is applied, a map is constructed of the environment from the distance sensors simultaneously to the localisation inside this map. This map of the environment can then be used to correct for the drift in the odometry in two ways. The first method is called loop closure where previously visited areas or sensor readings are recognised and the odometry is updated accordingly. The second method is by referencing the map to an already known 'ground truth' map of the environment and matching the observations, thereby obtaining the position of the drone (as was discussed in the first paragraph). Applying this SLAM method to the LIDAR measurements will be called LIDAR/Inertial Simultaneous Localisation and Mapping (LI/I-SLAM) while the stereo camera measurements will go through the Stereo Odometry and Feature Tracking Simultaneous Localisation and Mapping (SOFT-SLAM) algorithm. Both these algorithms will be addressed later.

However, due to the large difference in size and payload capacity between the drones, the two drone types will not be able to use identical sensor arrays. The “DALE” drones will receive a smaller LIDAR. The large drones will also be at a further distance from the aircraft making the stereo cameras a lot less accurate. This means that both drone types will use both algorithms but both will lean more on their strongest senses (LIDAR for the “CHIP” drones and stereo vision for the “DALE” drones). This will be addressed in Subsection 5.3.4.

For merging the information of the two algorithms two approaches are explored in the literature. In the first approach, the two algorithms can be executed independently from each other and the resulting position data merged using an Extended Kalman Filter (EKF) [75]. The second approach links the two algorithms and uses data from both sensors to create one centralised point cloud [77]. In the current design the first approach is selected because it allows a clean separation between the two systems that will reduce development complexity.

LI/I-SLAM Algorithm

Using LIDAR and SLAM for positioning inspection drones relative to an aircraft has been demonstrated by Mainblades [3] in a real-world environment. The method works by estimating the position of the aircraft using the odometry as discussed in Subsection 5.3.1 and by generating the associated point cloud of the environment. This point cloud is then matched to a known map of the aircraft to also obtain the position of the drone relative to the aircraft. To obtain this map of the aircraft, Mainblades first performs a manual flight around the aircraft to create the associated point cloud. Figure 5.6a shows the point cloud obtained by the manual fly-around. This point cloud will be processed to obtain the 3D model of the aircraft. which then in turn can be used for every inspection of every aircraft of that type.



(a) LIDAR Point cloud used to construct the aircraft model by Mainblades.[4] (b) Real time point cloud obtained from the LIDAR in the Mainblades drone.[5]

Figure 5.6: LIDAR results from the Mainblades drones.

Figure 5.6b shows the real-time point cloud observed by the Mainblades drone. This is the sensor data that is compared with the 3D model of the aircraft. This comparison process is performed in real-time by Mainblades using Google Cartographer [6]. This approach has also been successfully demonstrated in the literature using a very similar software stack [54]. Using this obtained position relative to the aircraft, the pictures that are taken can then be positioned onto the 3D model of the aircraft.

SOFT-SLAM Algorithm

Using visual and inertial sensors for SLAM is known as Visual Inertial Simultaneous Localisation and Mapping (VI-SLAM). These systems have been proven to provide sufficiently accurate positional information for the use case of indoor drone flight in narrow spaces. A. Merzlyakov et al. compare modern General-Purpose Visual SLAM approaches using three benchmark data sets [51]. From this comparison, ORB-SLAM3 [12] is identified as a candidate for further study after OpenVSLAM has been removed from the public domain. ORB-SLAM3 is demonstrated to have a $0.035\,\mathrm{m}$ Root Mean Squared (RMS) Absolute Trajectory Error (ATE) [58] on the EurocMAV [7] data set [12]. A more recent VI-SLAM algorithm designed specifically for drones, the SOFT-SLAM algorithm proposed by I. Cviˇsi´c et al. [19] seems specifically fit for the current application. This state-of-the-art algorithm scores the highest on the KITTI accuracy benchmark [8] and also generates a map of the surrounding environment from the mapping algorithm (see Figure 5.7b). The accuracy achieved by SOFT-SLAM is impressive, achieving a 0.82% error in position estimation on the KITTI accuracy benchmark and an error measured in single centimetres for tests on the EurocMAV dataset [19] (the accuracy can also be judged from Figure 5.7a).

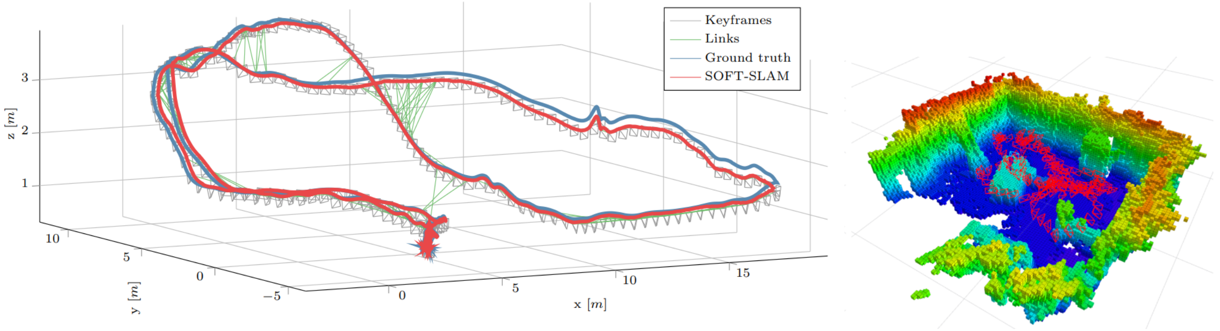
However, for the application of these algorithms to the navigation of inspection drones in aircraft environments a few unknowns remain. These are summarised below:

1. How do the textureless and rounded surfaces commonly found on aircraft and in aircraft hangars influence the accuracy of visual navigation algorithms?
2. How do VI-SLAM algorithms behave in large indoor spaces such as aircraft hangars? Regions such as the aircraft wingtip have very few close by visual reference points.

[3] Retrieved from: https://youtu.be/nqk1w6vzS9k (cited 18 May 2022)
[4] retrieved from: https://youtu.be/nqk1w6vzS9k (cited 18 May 2022)
[5] retrieved from: https://youtu.be/cZgR60m8SM8 (cited 18 May 2022)
[6] Retrieved from: https://github.com/cartographer-project/cartographer (cited 18 May 2022)
[7] Retrieved from: https://projects.asl.ethz.ch/datasets/doku.php?id=kmavvisualinertialdatasets (cited 18 May 2022)
[8] Retrieved from: http://www.cvlibs.net/datasets/kitti/eval_odometry.php (cited 20 May 2022)

3. How robust are VI-SLAM algorithms with regards to lighting changes, glare and reflections that can be expected in an aircraft hangar or outside?
4. How does the drift of the VI-SLAM algorithms increase over a flight time of 30 minutes?

Further uncertainty is created by the accuracy claims of commercial drone manufacturers who use all-around stereo vision for positioning. The DJI Matrice M300 RTK only guarantees a position accuracy of $30\,\mathrm{cm}$ when using the visual positioning system in windless conditions [9]. In regards to uncertainty two, the DJI Matrice also requires a minimum distance of $20\,\mathrm{m}$ to objects on both sides of the drone and a minimum distance of $30\,\mathrm{m}$ to the ground. These minimum distances are not guaranteed to be available in aircraft hangars. All these uncertainties compound to make the employment of a purely visual navigation system a development risk. However, while not sufficient for navigation on its own, the utilisation of visual navigation still provides a wealth of information that is extremely valuable to the situational awareness of the drone. The update rate of this visual information (mainly the optical flow) can also be higher than the information obtained from LIDAR sensors, which is particularly good for rapid pose estimation.



(a) Ground truth of the state estimation compared to the track generated by the SOFT-SLAM algorithm. [19]

(b) 3D map (octomap) generated by the SOFT-SLAM algorithm. [19]

Figure 5.7: Navigation results using the SOFT-SLAM algorithm [19]

5.3.2. Collision Avoidance Algorithms

In order to prevent collisions between the drones in the swarm, a collision avoidance system is needed. The swarm cannot rely upon the trajectories generated ahead of time to avoid collisions. This is because of two reasons. The first reason is that any deviation from the expected behaviour, be it through outside interference or a malfunction, would bring the entire system at risk if no collision avoidance system would be implemented. The second reason is that the system will have to account for unforeseeable influences of other agents that do not belong to the system. The best examples are humans walking around the hangar during the inspection or vehicles driving outside during an outdoor inspection. A collision with these agents will have to be avoided.

Three approaches to collision avoidance were selected to be considered based on prior work in the literature [35]. The first is CBS, where path planning and conflict avoidance are performed in tandem on a grid-based position and time system (see Section 5.2). The second method is Model Predictive Control (MPC) where a combination of path planning and drone dynamics is taken into account for a finite time horizon into the future to obtain drone inputs that will prevent a collision. The main advantage of this method is that the drone dynamics (down to the motor level) are taken into account and as such, there is high certainty that the inputs generated by the MPC model will be safe. An example is implemented in [66]. The third and final method is Velocity Obstacles (VO). With VO, methods the positions of the drones are used to construct velocity obstacles for all drones. If a given drone selects its velocity outside of the velocity obstacles generated by all the other drones it will be on a collision-free trajectory. The main advantages of this method are that it scales well with a large number of drones, that it is computationally inexpensive and that no communication between the drones is needed to obtain a collision-free solution. No communication is needed because the method only uses the relative position and velocity of the other drones, both of which can be sensed onboard the drones without communication. An overview of the advantages and disadvantages of all the methods is provided in Table 5.3. All methods have been demonstrated for drones in the literature, the reader is referred to the review in [35].

From Table 5.3 VO is selected to be implemented. The main advantage of the method is that it is communication-free which increases the robustness of the system to a communication failure and that it allows the system to

[9] Retrieved from: `https://dl.djicdn.com/downloads/matrice-300/20200507/M300_RTK_User_Manual_EN.pdf` (cited 14 June 2022)

Table 5.3: Trade-off table for the collision avoidance algorithm. Meaning of scoring and colours: green (3) = exceeds required performance, blue (2) = meets required performance, red (1) = below required performance. Overall score: 1-1.5 (RED), 1.5-2.5 (BLUE), 2.5-3.0 (GREEN).

Option	**Required Communication & Coordination**	**Scalability & Computational Complexity**	**Quality of solution**	**Average score**
CBS	Real time communication with a centralised system is required	Can be extreme depending on the dimensionality of the problem	Can deliver near-optimal solutions	
Score	1	1	3	**1.6**
Colour	RED	RED	GREEN	BLUE
MPC	Real time communication of intent is required (can be decentralised)	Can be moderate depending on model complexity and the time horizon	Drone dynamics are taken into account delivering near-optimal solutions.	
Score	2	2	3	**2.33**
Colour	BLUE	BLUE	GREEN	BLUE
VO	No communication required	Excellent scaling properties and low computational complexity.	Solutions are not guaranteed to be optimal but are guaranteed to be collision-free [10]	
Score	3	3	2	**2.67**
Colour	GREEN	GREEN	BLUE	GREEN

seamlessly adjust to other agents that are not accounted for inside the system (such as humans).

Velocity Obstacles Implemented with ORCA

VO are an intuitive concept and have been repeatedly rediscovered and developed into multiple different variants. For an overview, the reader is referred to [39]. The variant of VO that was chosen to be implemented is ORCA. This is because ORCA delivers smooth solutions which are close to the drone's preferred velocities and solves all the problems of the previous methods [39]. The algorithm works by creating planes that restrict the velocities to the drone. By picking a velocity closest to the desired velocity in this region of allowed velocities a collision-free path is obtained. No communication is needed between the drones, the only assumption is that all the drones are executing the ORCA algorithm. The specifics of the algorithm can be found in [6] but will also be paraphrased here. Figure 5.8a shows the relative positions of two drones which are represented by two spherical bounding boxes. Figure 5.8b shows the velocity obstacle of A induced by B for a time horizon τ. If the relative velocity of the two drones falls inside this obstacle a collision will occur within the time frame τ. Equation 5.4 shows the equation for the velocity obstacle induced by B on A. In this equation $S(\mathbf{p}, r)$ means a sphere of radius r centred at position **p** and $VO^{\tau}_{A|B}$ is the set of relative velocities between A and B that would cause a collision to occur.

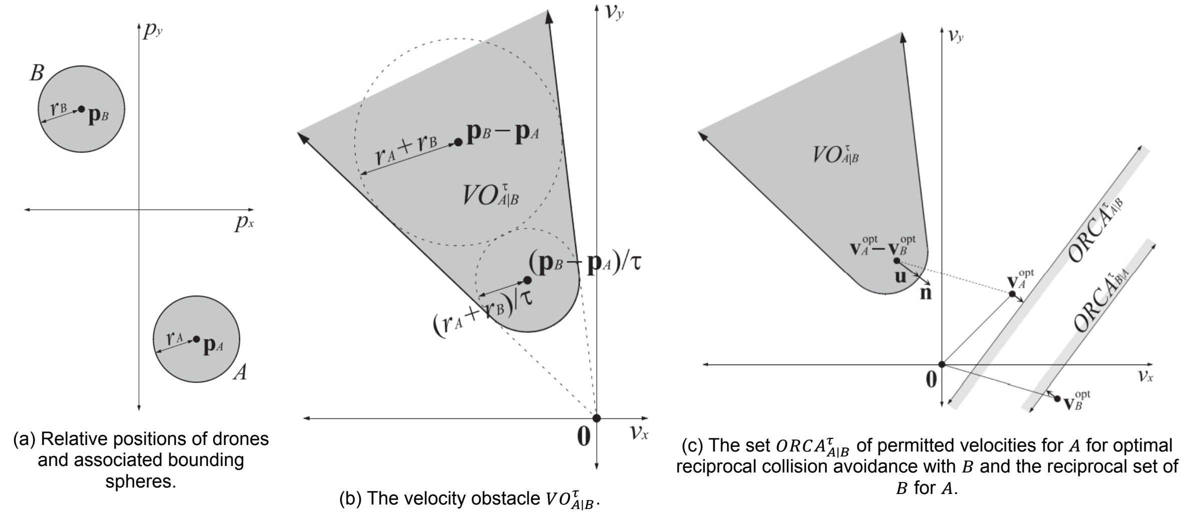
$$VO^{\tau}_{A|B} = \{\mathbf{v} \mid \exists t \in [0,\tau] :: t\mathbf{v} \in S\left(\mathbf{p}_B - \mathbf{p}_A, r_A + r_B\right)\} \tag{5.4}$$

From this velocity obstacle, the vector from the closest point of the velocity obstacle to the relative velocity, can be obtained (see Figure 5.8). This vector **u** is given by Equation 5.5 where v^{opt}_A and v^{opt}_B are the velocities of the drones that have to be optimised for (these will be discussed later). **n** is normal to the velocity obstacle at the origin of **u**.

$$\mathbf{u} = \left(\underset{\mathbf{v} \in \partial VO^{\tau}_{A|B}}{\arg\min} \left\| \mathbf{v} - \left(\mathbf{v}^{\text{opt}}_A - \mathbf{v}^{\text{opt}}_B\right) \right\| \right) - \left(\mathbf{v}^{\text{opt}}_A - \mathbf{v}^{\text{opt}}_B\right) \tag{5.5}$$

Equation 5.6 then gives the equation for the region of space that a new velocity can be selected inside. This region is demarcated by a plane at the position of $\mathbf{v}_A^{\text{opt}} + \frac{1}{2}\mathbf{u}$ with the normal $\mathbf{n}$. All points in the direction of $\mathbf{n}$ can be picked by the drone and will not cause a collision. The factor 1/2 in this equation accounts for the fact that both drones will expend equal effort to avoid a collision. However, the responsibility for avoiding the collision can be distributed over the drones arbitrarily as long as the total sums to 1.

$$ORCA_{A|B}^{\tau} = \left\{ \mathbf{v} \mid \left(\mathbf{v} - \left(\mathbf{v}_A^{\text{opt}} + \frac{1}{2}\mathbf{u} \right) \right) \cdot \mathbf{n} \geq 0 \right\} \tag{5.6}$$



(a) Relative positions of drones and associated bounding spheres.

(b) The velocity obstacle $VO_{A|B}^{\tau}$.

(c) The set $ORCA_{A|B}^{\tau}$ of permitted velocities for A for optimal reciprocal collision avoidance with B and the reciprocal set of B for A.

Figure 5.8: Visualisations of the ORCA algorithm [6].

To select the velocity $\mathbf{v}_A^{ORCA}$ closest to $\mathbf{v}_A^{opt}$ from the ORCA region linear programming is used in [6]. However, in the current implementation, a new (to the best knowledge of the authors) implementation is used where $\mathbf{v}_A^{ORCA}$ is selected by an iterative approach analogous to gradient descent where the gradient is constructed from the error. The algorithm is shown in Algorithm 3. This algorithm is a lot simpler and less computationally expensive than the linear programming used in [6]. However, it does not provide guarantees of finding the optimal solution which would be described by Equation 5.7.

$$\mathbf{v}_A^{\text{ORCA}} = \underset{\mathbf{v} \in ORCA_A^{\tau}}{\arg\min} \left\| \mathbf{v} - \mathbf{v}_A^{\text{opt}} \right\|. \tag{5.7}$$

Algorithm 3 Algorithm for determining $\mathbf{v}_A^{ORCA}$ from $\mathbf{v}_A^{opt}$

```
$selectedVelocity \leftarrow False$
$vORCA \leftarrow vOpt$
while not selectedVelocity do
    $selectedVelocity \leftarrow True$
    for all ORCAPlane in ORCAPlanes do
        if OnWrongSide(vORCA, ORCAPlane) then
            $selectedVelocity \leftarrow False$
            $vORCA \leftarrow MoveToCorrectSide(vORCA, ORCAPlane)$
        end if
    end for
end while
```

The last step is how to determine $\mathbf{v}_A^{opt}$. The choice of this variable has a large influence on the guarantees provided by ORCA. Only when $\mathbf{v}_A^{opt} = \mathbf{0}$ does ORCA guarantee a collision-free solution in high-density environments. However, this choice can lead to a deadlock where all or most drones are stationary and waiting for each other (although this only occurs in complex situations of thousands of agents in complex environments). The second choice is $\mathbf{v}_A^{opt} = \mathbf{v}_{prev}$ where $\mathbf{v}_{prev}$ is the previous velocity of the drone. This choice gives the smoothest results and gives the drones a lot of inertia, expending little energy to avoid each other. The third and final choice is $\mathbf{v}_A^{opt} = \mathbf{v}_{desired}$ which provides a solution where the drones will follow their desired velocity as fast as possible. This can lead to some oscillations between ORCA and the algorithm to determine the

$\mathbf{v}_{desired}$. To account for all these factors $\mathbf{v}_A^{opt}$ is chosen to be $\mathbf{v}_A^{opt} = \frac{\mathbf{v}_{desired}+\mathbf{v}_{prev}}{2}$ which gets scaled down closer to **0** iteratively when no solutions are found. This guarantees that eventually a collision-free solution will be found. For only five drones, this scaling occurs very rarely.

The static objects in the drone environment, such as the aircraft, can not always be reduced to spherical bounding boxes. Equation 5.8 shows the equation for the velocity obstacle generated by static object O for drone A. Here $\oplus$ denotes the Minkowski sum of the static object and the spherical bounding box of the drone.

$$VO_{A|O}^{\tau} = \{\mathbf{v} \mid \exists t \in [0,\tau] :: t\mathbf{v} \in O \oplus -S\,(\mathbf{p}_A, r_A)\} \tag{5.8}$$

However, because this sum is expensive to evaluate and the resulting velocity obstacle is not guaranteed to be convex, a different approach than was employed in [6] was implemented. To compute the required **u**, the actual bounding box of the drone (not spherical) is cast forward in time along the vector $\mathbf{v}^{ORCA}$ from Algorithm 3. The distance of the cast is determined by τ. When the cast bounding box intersects the checked object, $\mathbf{v}^{ORCA}$ is offset in a manner directly analogous to Equation 5.6. This way, a collision will not occur within τ seconds. The resulting behaviour of the system is that the drone will get asymptotically closer to the static object, eventually touching it at an infinite time in the future. An extra constant safety margin **s** can be added to **u** such that the drone comes to a rest at a distance **s** from the static object.

Experimental Results

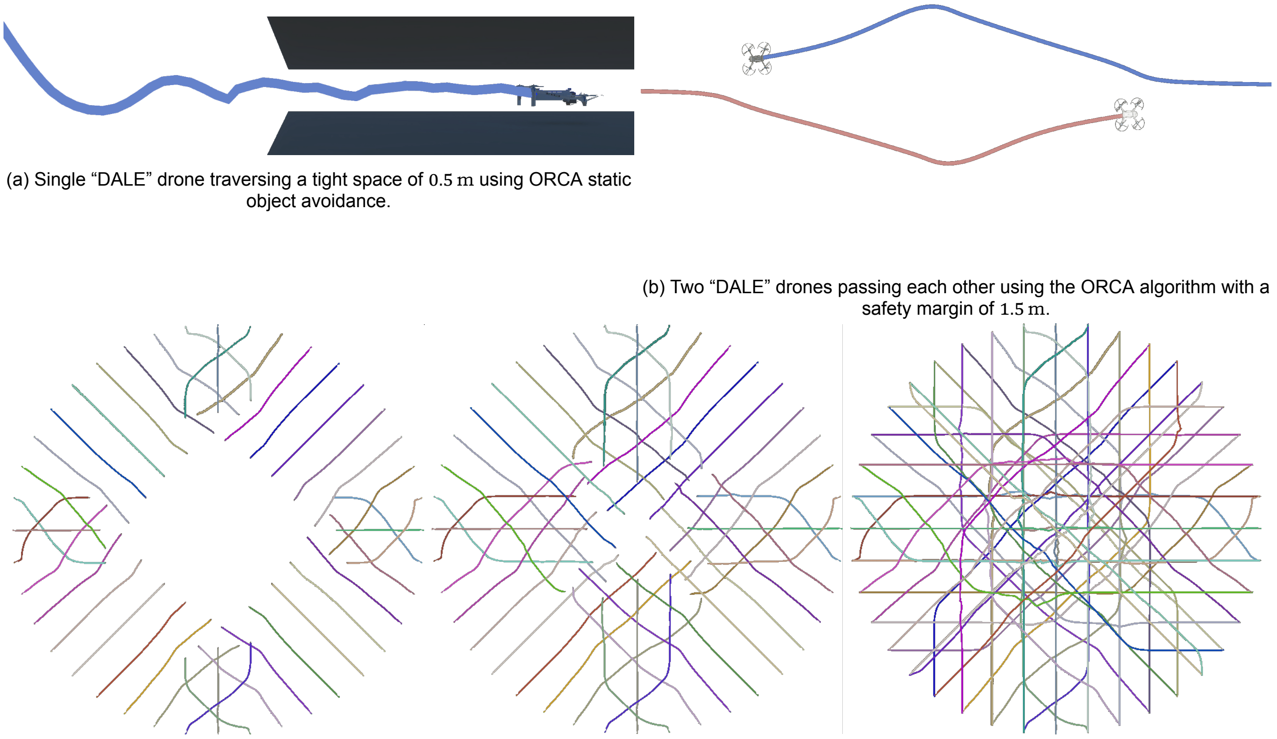
The approach has also been implemented on physical drones with no communication in the literature [18]. This experiment proved that ORCA can prevent collisions between drones with noisy and imperfect sensor and actuator dynamics. Two new factors surfaced in the experiment. The first is that reciprocal dances can occur. Reciprocal dances are also an issue in older VO algorithms and occur when two agents oscillate on which directions to pick to pass by each other (the same situation can also occur with humans trying to pass each other in a hallway). In this case, the reciprocal dance is not caused by the ORCA algorithm but by the uncertainty in the observations of the position and velocity of the other drones. Added to this are the actuation imperfections of the drones. All these uncertainties can cause reciprocal dances to occur. The second factor is that due to sensor blind spots and sensor imperfections, one of the two drones might not detect the other and then not execute the reciprocal part of ORCA. Even in this situation collisions do not occur because the other drone will converge exponentially fast to a solution which does not collide with the other drone.

Simulation Results

A simulation of the algorithm has been implemented in the inspection simulation. In this simulation, the drone dynamics were modelled down to the forces and torques acting on the drones such that no physical laws are broken by the simulated drone behaviour. The control systems for the drones are discussed in chapter 6. In the simulation navigation through tight spaces and collision avoidance for both a small number of drones and a large number of drones were demonstrated (see Figure 5.9 and Figure 5.10). Figure 5.10 shows a visual representation of the ORCA algorithm as implemented.

As is discussed in Section 5.4, numerous complete simulation of the inspection were performed. These simulations used no global optimisation so conflicts between the drones are numerous and random. Therefore, these simulations are suited to test the reliability of the collision avoidance system in a worst-case scenario. In total 800 simulations were run. For a swarm size under 7 drones the occurrence of collisions was 0% over all simulations. However, for swarms larger than this collisions sometimes did occur. As an average over all simulations the probability of a collision during an inspection was 0.34%. This is three times higher than the requirement of 0.1% chance of a collision occurring. Due to the fact that these collisions are concentrated in the higher swarm sizes than the implemented swarm this probability is allowed. By increasing the safety margins and caution exerted by the drones the occurrence of collisions could be reduced with no extra development time. This could be done by increasing the avoidance radii and time horizons of the drones. However, this would come at the cost of an increased inspection time due to the conservative behaviour of the drones. In these situations the global optimisation of the inspection would be a better approach. In this way, the occurrences of conflicts can be reduced which would lower the number of collisions that occur.

The values used for the collision avoidance system during the simulation are stated in Table 5.4. It was discovered that the values for s do not comply with the minimum spacing required by the flight control subsystem (in chapter 6). Changing these values is trivial but rerunning the simulations to determine the outcome of this change will require significant time. It was not possible to do this during the current iteration of the design so this is left for the next iteration.



(a) Single “DALE” drone traversing a tight space of $0.5\,\mathrm{m}$ using ORCA static object avoidance.

(b) Two “DALE” drones passing each other using the ORCA algorithm with a safety margin of $1.5\,\mathrm{m}$.

(c) Trails of 40 drones using ORCA traversing a 100 meter circle early in the manoeuvre.

(d) Trails of 40 drones using ORCA traversing a 100 meter circle in the middle of the manoeuvre.

(e) Trails of 40 drones using ORCA traversing a 100 meter circle at the end of the manoeuvre.

Figure 5.9: Drone simulations using the ORCA algorithm.

Table 5.4: ORCA values used in the simulation.

Drone	τ [s]	r [m]	τ_{static} [s]	**s** [m]
“CHIP” drone	2	2	5	0.5x0.5x0.5
“DALE” drone	2	1	4	0.5x0.5x0.5

Human recognition and avoidance

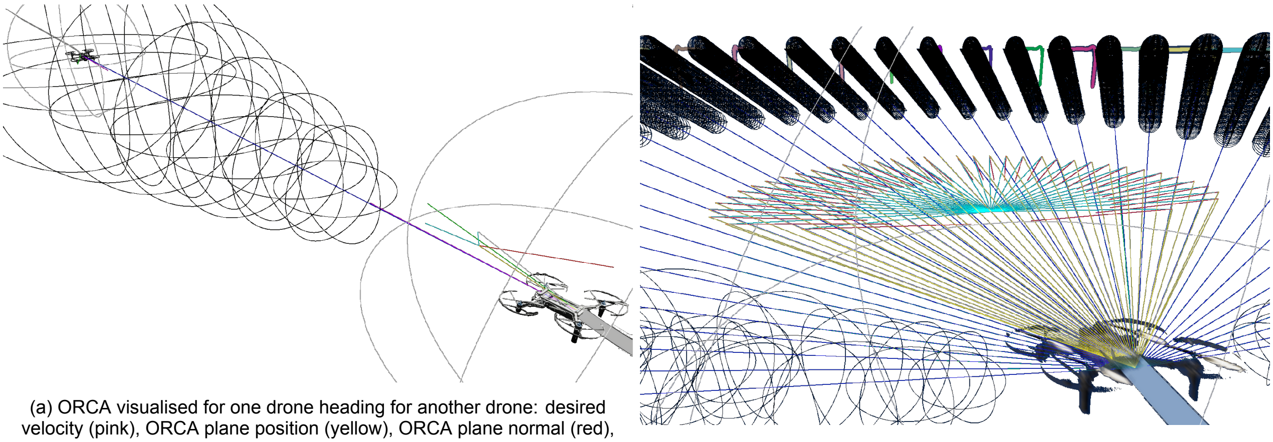
In order for the drones to avoid any humans by a clearance of $2\,\mathrm{m}$ the drones will have to recognise a given object as being a human. If the drones would not recognise the humans they would still be avoided by the drones, as would any other object. However, an extra safety margin is required for the safety of the humans. The drones will use the onboard vision system to identify any humans visible in the all-round vision of the drones. Multiple algorithms exist for this purpose and an algorithm has been experimentally verified to work on the selected hardware in the literature (see Table 5.7).

5.3.3. Communication

For the swarm to efficiently operate communication is needed between the drones. The drones can then update each other and the operator on their respective states and the global status of the inspection. The communication system also needs to give the operator the ability to take over manual control of a drone and to see the live video feed from the drone. The redundant manual control link is sized and discussed in chapter 9. First, the sizing of the communication system will be discussed after which the architecture of the underlying network will be considered.

Communication Subsystem Sizing

From the requirements, it was determined that an IEEE 802.11g compliant communication system would be sized. 802.11g receivers can go to a sensitivity as low as $-82\,\mathrm{dBm}$ while maintaining a sufficient Bite Error Rate (BER) [37]. At this sensitivity the achievable bandwidth is $6\,\mathrm{Mbit\,s^{-1}}$. Reportedly, modern receivers can go



(a) ORCA visualised for one drone heading for another drone: desired velocity (pink), ORCA plane position (yellow), ORCA plane normal (red), velocity iterations (cyan), new velocity (grey), avoidance cone (black), avoidance sphere (white).

(b) ORCA visualised for many drones heading to the opposite side of a circle. Velocity iterations (cyan) point to the new velocity (center).

Figure 5.10: Visualisations of the ORCA algorithm for a single drone and many drones.

to sensitivities as low as $-120\,\mathrm{dBm}$ and receive $11\,\mathrm{Mbit\,s^{-1}}$ at a sensitivity of $-83\,\mathrm{dBm}$, [11]. To be conservative, the values from the original 802.11g standard will be used. The transmission power is limited to an Effective Isotropic Radiated Power (EIRP) of $20\,\mathrm{dBm}$ by the European regulations [38]. Using this transmission power and receiver sensitivity a link budget of $102\,\mathrm{dBm}$ is achieved. This link budget needs to be distributed over the free-space loss and noise factors. The noise factors of relevance to this system are multi-path loss (signals reflecting inside the hangar and creating destructive interference) and interference from other systems operating on the same frequency. This is summarised in the link budget in Equation 5.9 here P_{RX} is the received power, P_{TX} the transmitted power, G the gain, L_{FS} the free-space loss and L_{FM} the fade margin accounting for all other losses. The gain G is identical for the transmitter and the receiver because WiFi is a bidirectional packet-based network. Equation 5.9 was rewritten to calculate the required antenna gain for the system to function.

$$
\begin{aligned}
P_{RX} &= P_{TX} + G + L_{FS} + L_{FM} \\
G &= P_{TX} - P_{RX} + L_{FS} + L_{FM}
\end{aligned}
\tag{5.9}
$$

The free space loss in dB can be calculated from Equation 5.10 with the distance d_{km} in km and the frequency f_{GHz} in GHz. Using this frequency of $2.45\,\mathrm{GHz}$ and required range of $95\,\mathrm{m}$ gives a path loss of $79.8\,\mathrm{dB}$

$$
\begin{aligned}
L_{FS} &= 10log_{10}\left(\left(\frac{4\pi df}{c}\right)^2\right) \\
L_{FS} &= 20log_{10}(d_{km}) + 20log_{10}(f_{GHz}) + 92.45
\end{aligned}
\tag{5.10}
$$

The fade margin of the system needs to account for losses both in destructive interference through the multiple paths the signal will take and for the interference of other systems in the environment. Because this is extremely hard to predict or quantify, experimental data was used to determine the fade margin. Using a Rayleigh noise distribution a fade margin of $18\,\mathrm{dB}$ is required for a link availability of 99% [12]. For indoor systems, a fade margin of $30\,\mathrm{dB}$ accounts for both multi-path losses and noise interference for a link availability of 99% [13].

Using the fade margin of $30\,\mathrm{dB}$ and a path loss of $79.8\,\mathrm{dB}$ in Equation 5.9 an antenna system with a gain of $4\,\mathrm{dB}$ is required. Such antenna systems are widely available in extremely compact and light Printed Circuit Board (PCB) solutions. This communication system with a direct line of sight in a vacuum has a range of $3\,\mathrm{km}$. Two antennas will be installed on the drones to account for the destructive interference and redundancy purposes. If one antenna encounters a deep fade due to destructive interference the other antenna will likely

[11] Retrieved from: `https://support.huawei.com/enterprise/en/doc/EDOC1000077015/bc2e25db/receiver-sensitivity#:~:text=Receiver%20sensitivity%20of%20a%20common,power%20increases%20by%203%20dB` (cited 9 May 2022)

[12] Retrieved from: `https://www.fishercom.xyz/voice-channels/fades-fading-and-fade-margins.html` (cited 10 June 2022)

[13] Retrieved from: `https://community.element14.com/technologies/wireless/b/blog/posts/real-world-link-budget-and-fade-margin` (cited 10 June 2022)

not be in this 'dead zone'. The transmitter and receiver together with an overview of the communication system components are provided in Table 5.8.

Communication Architecture

Figure 5.11 shows the communication diagram. The communication diagram shows the architecture of the communication network. Because the IEEE 802.11 protocol uses a packet-based network, the specific implementation of the communication logic can be software-defined which provides a lot of freedom. A possibility for the architecture is provided but can easily be modified if this is deemed appropriate. In Figure 5.11, the state communication entails the communication between the drones on the status of the inspection, the position and velocities of the drones and other drone variables such as battery charge. This communication rate was sized on the communication of 100 *double float* (8 bytes) values at a rate of $20\,\mathrm{Hz}$. This would require about $100\,\mathrm{kB\,s^{-1}}$ in bandwidth between the drones and $0.6\,\mathrm{MB\,s^{-1}}$ between the complete swarm and the ground station. The remote control command was based upon the transmission of 10 *double float* values at a rate of $200\,\mathrm{Hz}$. This transmission requires a similar data rate of $100\,\mathrm{kB\,s^{-1}}$ but can occur over the redundant control link. Before and after the inspection the ground station will communicate with a cloud server to upload the inspection results and download the inspection trajectories before the inspection. In reality, due to the extremely conservative estimates of the sizing, it is expected that the available data rates for the communication subsystem will be much greater and a lot more data will be able to flow between the drones and the ground station (see Section 5.5). Due to the software-defined nature of the network, this can easily be accommodated.

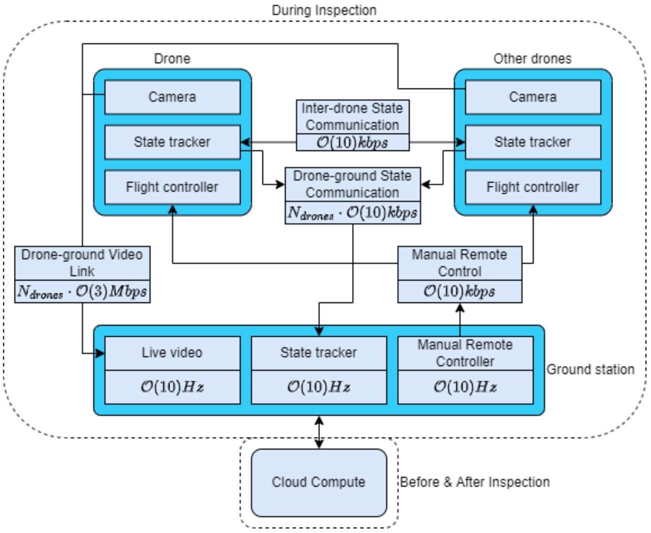


Figure 5.11: Communication flow diagram before and during the inspection.

5.3.4. Drone Hardware Selection

Now that the algorithms have been established and the communication system has been sized, the hardware required to implement these concepts can be selected. First, the Positioning and collision avoidance hardware will be selected. After this, the computer hardware required to handle the output of the selected sensors and the computational requirements of the algorithms will be selected. Then the communication hardware will briefly be considered and finally, a point about the interchangeability of the drones will be discussed.

Positioning and collision avoidance hardware

The first use case that should be addressed by the positioning and collision avoidance hardware is the internal positioning of the drones themselves. Due to the complementary nature of LI/I-SLAM and SOFT-SLAM both algorithms will be used on both drone types. This does not mean that the positioning system of both drone types will be equivalent. Due to the large difference in payload capacity of the two drones, the LIDAR systems that both are able to carry are different. The "CHIP" drone will be able to carry a LIDAR system with a $360°$ horizontal FOV and a $90°$ FOV. While the "DALE" drone will only be able to carry a solid-state LIDAR with a $360°$ horizontal FOV and a $360°$ vertical FOV. Also, the expected flight profiles of the drones are different. The "DALE" drones will always be close to objects in the environment, giving the stereo cameras good parallax information. However, this is not the case for the "CHIP" drones which will fly a lot further away from the aircraft and objects in the environment. This means the two drone types will use their positioning systems in very different ways. The "DALE" drones will rely mainly on the SOFT-SLAM algorithm and use LI/I-SLAM to cancel out any drift and provide accurate height and navigational information. The large drones will navigate mainly by the LI/I-SLAM algorithm while using the visual information from the sensors for pose and attitude estimation.

The second use case of the hardware is the relative positioning of the drones relative to each other. The ORCA algorithm running on a given drone needs the relative positions and velocities of the other drones. During normal operations, these variables can be communicated across the drones using the communications subsystem. However, in the case where communication is unavailable or there are other agents in the system the drones need to be able to determine these variables themselves.

First, the selection of the LIDAR sensor is addressed. For the “CHIP” drone a wide variety of $360°$ LIDAR sensors are considered, while for the “DALE” drone only one suitable, lightweight option was found. Therefore, the selection of the LIDAR sensor for the “CHIP” drone is the one to be addressed in more depth.

Table 5.5: LIDAR sensor options table including specifications.

Sensor	Mass [g]	Power [W]	Vertical FOV [°]	Frequency [Hz]	Vertical Resolution [°]	Maximum distance [m]	Cost [€]
Ouster OS0 [14]	447	14-20	90	10-20	0.7	50	13000
Quanenergy M8 Plus [15]	900	16	20	5-20	0.033-0.132	150	5000
Velodyne Ultra Puck [16]	925	10	40	5-20	0.33	200	8000
Velodyne HDL-32E [17]	1000	12	43	5-20	1.33	100	1325

The LIDAR options for the “CHIP” drone are presented in Table 5.5. The need for a wide vertical FOV and a low mass, directed the choice towards the Ouster OS0 drone, as it is, by a good margin, the best performing sensor in such categories. For the “DALE” drone, the only suitable component available in the market is the Velodyne Velabit [18], which was therefore selected.

Second is the selection of the visual depth sensors. These are Stereo and Red, Green, Blue-Depth (RGB-D) cameras. The same model will be chosen for both types of drones. The options are presented in Table 5.6. In this case, the selected component the Intel® RealSense™ D450 Depth Module is chosen, due to its acceptable range and weight. Given the fact that 6 cameras will be needed, the high weight of the Stereolabs ZED2 rules it out, while the range of the RealSense™ D430 is too low to be used in aircraft inspections.

Table 5.6: Depth sensor options table including specifications.

Sensor	Mass [g]	Power [W]	Vertical FOV [°]	Frequency [Hz]	Resolution [Pixels]	Maximum ideal range [m]	Cost [€]
Intel® RealSense™ D450 Depth Module [19]	29.5	1.5	65	90	1280x720	6	400
Intel® RealSense™ D430 Depth Module [20]	14.5	2	52	90	1280x720	3	300
Stereo Labs ZED2 [21]	125	3	70	60	1280x720	20	700

[14] Retrieved from: https://ouster.com/products/scanning-lidar/os0-sensor/ (cited 31 May 2022)
[15] Retrieved from: https://quanergy.com/products/m8/ (cited 31 May 2022)
[16] Retrieved from: https://velodynelidar.com/wp-content/uploads/2019/12/63-9378_Rev-F_Ultra-Puck_Datasheet_Web.pdf (cited 31 May 2022)
[17] Retrieved from: https://www.mapix.com/wp-content/uploads/2018/07/97-0038_Rev-M_-HDL-32E_Datasheet_Web.pdf (cited 31 May 2022)
[18] Retrieved from: https://velodynelidar.com/products/velabit/ (cited 31 May 2022)
[19] Retrieved from: https://www.intelrealsense.com/wp-content/uploads/2020/06/Intel-RealSense-D400-Series-Datasheet-June-2020.pdf (cited 31 May 2022)
[20] Retrieved from: https://www.intelrealsense.com/wp-content/uploads/2020/06/Intel-RealSense-D400-Series-Datasheet-June-2020.pdf (cited 31 May 2022)
[21] Retrieved from: https://cdn.stereolabs.com/assets/datasheets/zed2-camera-datasheet.pdf (cited 31 May 2022)

Computer selection

The swarm control subsystem requires a lot of algorithms to be executed and computations to be run. These computations will mainly occur on the onboard computer. Two candidates were identified for the onboard computer. The DJI Manifold 2-G and the Manifold 2-C [22]. These computers are built around the NVIDIA Jetson TX2 [23] and the Intel® Core™ i7-8550U [24] respectively. There are other embedded frameworks such as the Intel® NUC boards or even custom Field Programmable Gate Arrays (FPGA) (see [50]). However, the DJI boards are fit for purpose, are custom-tailored for drones and have been proven in countless commercial products. As such the Manifold board will be selected for the onboard computer. The remaining question is whether to select the GPU model or the CPU model. The Manifold 2-G module includes a 256-core NVIDIA Pascal™ GPU architecture with 256 NVIDIA CUDA cores and a Quad-Core ARM® Cortex®-A57 MPCore running at $2\,\mathrm{GHz}$ while the Manifold 2-C's contains an Intel® Core™ i7-8550U Quad-Core processor running at $4\,\mathrm{GHz}$.

Table 5.7 provides an overview of the computation jobs that need to be executed and their measured execution times. The first task that will be discussed is the processing of the raw data from the optical sensors. Especially the processing of the stereo cameras with the structured light depth sensors is intensive. In order to not slow down the onboard computer, this processing is performed on dedicated PCBs for each sensor. The layout of this architecture is further discussed in chapter 9. The further jobs that need to be processed are the processing of the LIDAR data, running Cartographer for the LIDAR SLAM, the SOFT-SLAM algorithm, the object (human) recognition and finally the A* pathfinding and the ORCA algorithm. The LIDAR processing in combination with Cartographer SLAM were demonstrated in the literature to run at $20\,\mathrm{Hz}$ on an Intel® NUC [54]. The exact specifications of the module are not published in the paper. However, the fastest Intel® NUC available at the publication date contains an Intel® Core™ i5-5300U running at $2.90\,\mathrm{GHz}$. Running the algorithm on the Quad-Core ARM processor would give it an execution time of $70\,\mathrm{ms}$. The SOFT-SLAM algorithm was demonstrated to run at an update rate of $20\,\mathrm{Hz}$ in the literature on two cores running at $3.4\,\mathrm{GHz}$. This process can be accelerated at least up to three times using the onboard GPU of the Manifold [49] and potentially even accelerated with an order of magnitude [25]. The GPU acceleration will provide the headroom for a large amount of input data from the surround stereo cameras. The human object recognition detection was demonstrated to run at an update rate of $24\,\mathrm{Hz}$ on the NVIDIA Jetson [26]. This process can also be run on the CPU of the Manifold 2-C but would take a relatively long time to execute. The remaining two algorithms were measured by the authors and were relatively light to execute. The A* pathfinding algorithm takes on average $10\,\mathrm{ms}$ to compute a path for the drone, with spikes up to $20\,\mathrm{ms}$. The execution time of the ORCA algorithm is extremely sensitive to the complexity of the situation. In a simple situation, the algorithm returns in less than $1\,\mathrm{ms}$ but in complex situations, it can take as many as $40\,\mathrm{ms}$ to return. The complex situations should never occur during a real inspection but will nevertheless be sized for.

Summing the execution times and dividing the workload over the four cores of the ARM® Cortex gives a worst-case total update rate of $18\,\mathrm{Hz}$. This is deemed acceptably close to the required $20\,\mathrm{Hz}$ for the positioning algorithm. This fast execution is made possible due to the GPU acceleration of the SOFT-SLAM and object detection algorithm. Because of this advantage, the Manifold 2-G is selected. If the GPU acceleration is deemed infeasible at later stages of the development process the compute module can be replaced by the Manifold 2-C which would run at an update rate of $2\,\mathrm{Hz}$ with the current object detection software and $28\,\mathrm{Hz}$ without the object detection algorithm. A better solution would have to be found for object detection (solutions exist for this use case [27]). The Manifold 2-C would also have less headroom for extensive visual SLAM and processing due to the impossibility of GPU acceleration.

Communication Hardware Selection

The onboard computer selected in the previous section contains an IEEE 802.11a/b/g/n/ac transceiver with a maximum transmission speed of $866.7\,\mathrm{Mbit\,s^{-1}}$ and an EIRP of $20\,\mathrm{dBm}$ for $2.4\,\mathrm{GHz}$. This transceiver meets all the communication requirements of the communication subsystem. Appropriate on-PCB antennas were

[22] Retrieved from: https://www.dji.com/nl/manifold-2/specs (cited 13 June 2022)

[23] Retrieved from: https://developer.nvidia.com/embedded/jetson-tx2 (cited 13 June 2022)

[24] Retrieved from: https://ark.intel.com/content/www/us/en/ark/products/122589/intel-core-i78550u-processor-8m-cache-up-to-4-00-ghz.html (cited 13 June 2022)

[25] Retrieved from: http://www.gipsa-lab.grenoble-inp.fr/summerschool/gpu2011/fichiers/6_Obukhov_OpenCV_rev5.pdf (cited 14 June 2022)

[26] Retrieved from: https://techcommunity.microsoft.com/t5/educator-developer-blog/high-performance-real-time-object-detection-on-nvidia-jetson-tx2/ba-p/917067 (cited 14 June 2022)

[27] Retrieved from: https://towardsdatascience.com/picking-fast-people-detector-working-with-opencv-on-cpu-in-2021-ff8d752088af (cited 14 June 2022)

[28] Retrieved from: https://techcommunity.microsoft.com/t5/educator-developer-blog/high-performance-real-time-object-detection-on-nvidia-jetson-tx2/ba-p/917067 (cited 14 June 2022)

Table 5.7: Processing and compute execution times, location and measured execution time.

Job	Compute location	Execution time [ms]	Measured on	Utilised cores / available cores	Execution time on Manifold 2-G CPU [ms]	Execution time on Manifold 2-C CPU [ms]
Optical sensor processing	Dedicated sensor PCB	< 50	Dedicated sensor PCB	-	-	-
LIDAR processing and Cartographer SLAM [54]	Onboard computer (CPU)	< 50	Intel® Core™ i5-5300U @2.90 GHz	Unknown / 2	< 72	< 36
SOFT-SLAM [19]	Onboard computer (CPU + GPU)	< 50	Intel® Core™ i7-5557U @ 3.4 GHz	2 / 4	< 28	< 42.5
A* Pathfinding	Onboard computer (CPU)	10 - 20	Intel® Core™ i7-8750H @ 3.9 GHz	1 / 6	19 - 39	10 - 20
ORCA	Onboard computer (CPU)	1 - 40	Intel® Core™ i7-8750H @ 3.9 GHz	1 / 6	2 - 80	1 - 40
Object (Human) detection [28]	Onboard computer (GPU)	< 41	Jetson TX2 GPU	256/256	0	330
Inertial sensor processing	Flight computer	-	-	-	-	-
Inner flight control loop	Flight computer	-	-	-	-	-

also sought for. There are a lot of options in this segment and a suitable antenna was easily identified. The components are shown in Table 5.8.

Table 5.8: Communication system components. The variables with marked with a star (*) are budgeted for on the onboard computer.

Component	Mass [g]	Power [W]	Cost [€]	Frequency [GHz]	Gain
IEEE 802.11 Transceiver	*	*	*	2.4 - 5.8	20 dBm
Antenna x2 [29]	<10	0	5.6	2.4 - 5.8	4 dB

Drone Interchangeability

An enticing possibility that arises from the commonality between a lot of the hardware inside the drones is the usage of purely "DALE" drones to also complete the tasks of the "CHIP" drones. This possibility has been considered and could be a feasible solution in the future. When LIDAR technology improves and becomes smaller and cheaper, the "DALE" drones could be equipped with a better sensor. As it is right now, the LIDAR sensors on the small drone will not be very useful outside of their normal flight region underneath and close to the aircraft. This is due to the narrow FOV of the sensor. If the drones would not be underneath or close to the aircraft, the narrow LIDAR beam would be pointing into far away spaces a lot of the time, not contributing any useful information. Thus, without improvements to the LIDAR sensors the small drones would have to rely mainly on their visual system to go where the "CHIP" drones goes. The feasibility of using the "DALE" drones in this configuration would have to be studied in detail. The uncertain factors mentioned in Subsection 5.3.1 surface again with the added uncertainty of not having a LIDAR to fall back upon (because it is not contributing

[29] Retrieved from: `https://www.renhotecic.com/1-Piece-2.4GHz-5.8GHz-Dual-Frequency-4dBi-High-Gain-Built-in-FPCB-FPV-Omnidirectional-Antenna-With-MMCX-Connector-For-RC-Drone` (cited 10 June 2022)

information a lot of the time in this situation). To the best knowledge of the authors, no system has been used in literature or by competitors that demonstrate this purely visual method of navigation. However, it does remain an alluring avenue for future research. Methods for positioning the drone relative to the aircraft purely based purely upon the monocular information of the vision system (without depth information) could be explored. Another option would be to develop an algorithm which makes sure the small FOV beam of the “DALE” drone LIDAR is always pointing at a region in space from which useful information can be extracted. There is a further problem with this method that would have to be overcome which is that the obtained data points from the LIDAR sensor would be ’flatter’ because the subjects the LIDAR sensor would be looking at would be further away.

5.4. Simulation of the swarm control subsystem

The functioning of the swarm control subsystem was implemented in a full 3D physics-based simulation in the Unity engine [30]. This simulation serves three purposes. The first purpose is the verification of all the algorithms discussed in this chapter. The capabilities of these algorithms are proven through the simulation. The second purpose of the simulation is the measurement of the thrust required by the drones during the inspection. This serves as an input to the propulsion subsystem in chapter 7. The third and final purpose is the verification of the sizing of the swarm, it is checked using the simulation results that the configuration of the swarm is the optimum configuration for narrow-body inspection. These subjects will be addressed in the following text

5.4.1. Verification of the swarm control subsystem

One of the largest values of implementing the simulation is that it can create a confrontation with unknown unknowns that were overlooked during the design of the swarm control subsystem. In this way the simulation serves as a method of verification for the methods discussed in this chapter. The verification of the ORCA algorithm, task generation and distribution algorithms was discussed in earlier sections. The component of the swarm control subsystem that could not be verified through simulation was the positioning algorithm for the drones. This will further be discussed in Section 5.5.

5.4.2. Required thrust measurement through the inspection

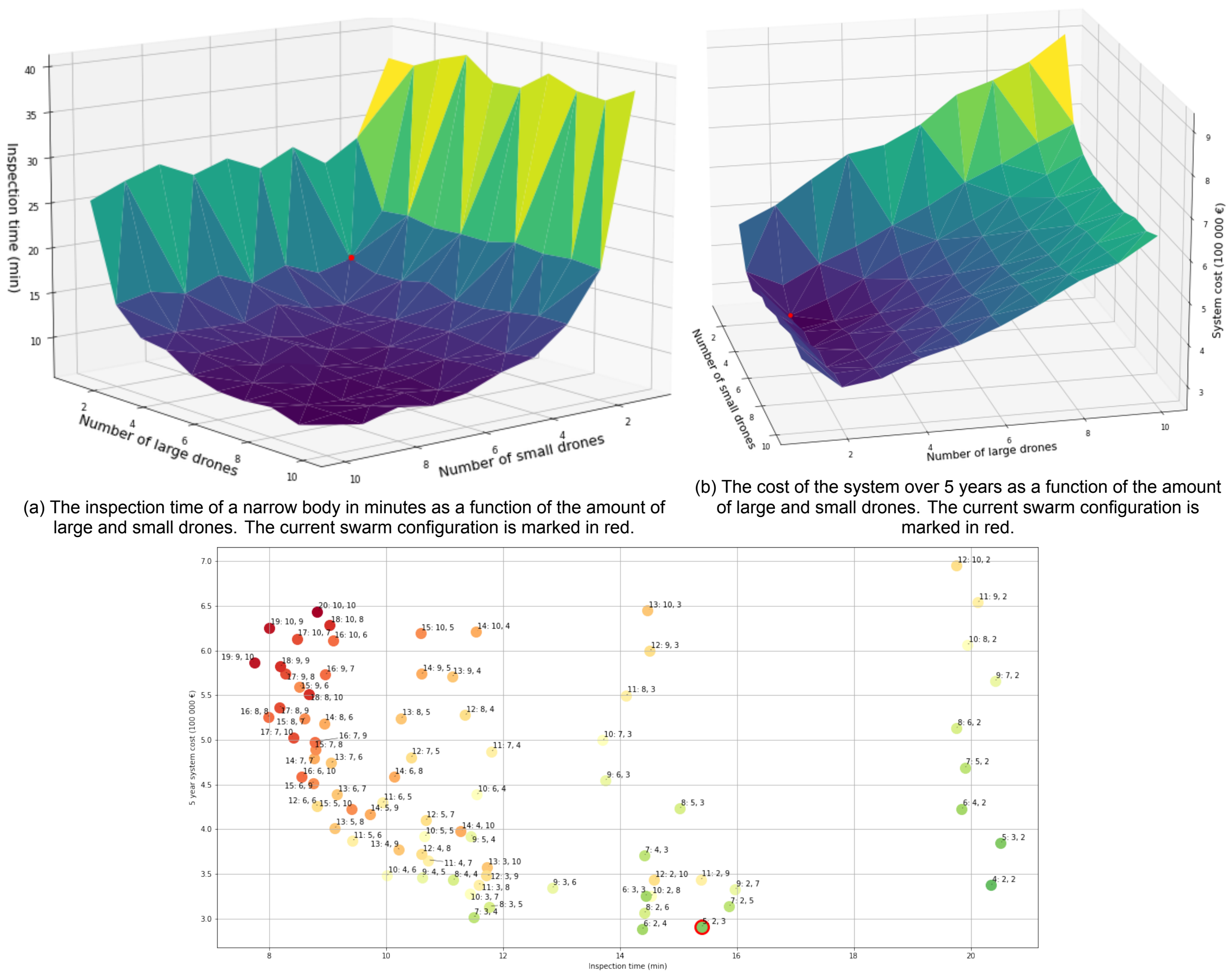
In the simulation the drones were simulated down to the forces and torques acting on the drones. Because of this the thrust levels required by the drones could be measured through the simulation. This measurement could retire an assumption that was problematic for the propulsion and power subsystem (see chapter 7). It was assumed that the average thrust to weight ratio required by the drones during the inspection would be significantly higher than one due to the accelerations and decelerations of the drones. However, from the simulation it could be determined that the average thrust level required by the drones was close to a thrust to weight ratio of one during almost the entirety of the inspection.

5.4.3. Swarm sizing using the simulation results

Using the simulation, the time required to inspect an aircraft by a swarm of arbitrary size could be generated. Figure 5.12a shows the inspection time of a narrow body in minutes as a function of the amount of large and small drones. With a swarm size of just one large drone and one small drone the inspection of the full aircraft takes 38 minutes. The critical factor here are the small drones. The large drones can finish the inspection of the top of the aircraft in 25 to 30 minutes. This corresponds well with the inspection time of Mainblades who inspect a 757 in 45 minutes. The 757 is about 40% larger than the A320 the current simulation used. Taking into account that size difference the results correspond almost perfectly.

From Figure 5.12a it can be seen that the current swarm configuration is situated in the ’sweet spot’ of the inspection time distribution. This swarm configuration obtains a low inspection time without going far into the territory of diminishing returns. This qualitative analysis is confirmed by Figure 5.12b which shows the operating cost of the system over the expected usage time of 5 years. This cost is calculated using the investment cost of the swarm and the cost for operating the swarm over 5 years with the associated inspection per inspection. Figure 5.12c shows the same data in 2D. From this it can be seen that the chosen system is one of the cheapest options together with a swarm of 2 “CHIP” drones and 4 “DALE” drones. The chart also shows the Pareto front of the inspection time versus system cost for the different swarm configurations. By choosing configurations on the front one can trade a higher system cost for a faster inspection time. With the current task distribution method negative returns are reached at a total swarm size of 17 drones.

[30] Retrieved from: `https://unity.com/products/unity-simulation-pro` (cited 17 June 2022)



(a) The inspection time of a narrow body in minutes as a function of the amount of large and small drones. The current swarm configuration is marked in red.

(b) The cost of the system over 5 years as a function of the amount of large and small drones. The current swarm configuration is marked in red.

(c) The distribution of the 5 year system cost versus the inspection time of different swarm combinations. The swarm drone numbers are depicted as follows: Total number of drones in the swarm: Number of large drones, Number of small drones. The current swarm configuration is circled in red.

Figure 5.12: Outcomes of the simulations for multiple swarm configuration in the simulated inspection of an Airbus A320. Depicted are the inspection time and 5 year system costs for the different swarm configurations.

5.5. Subsystem Verification and Validation

Outside of the simulation and verification discussed earlier in the chapter, the swarm control subsystem is hard to verify or validate outside of actual testing of the hardware and algorithms. Earlier in the chapter, as many references as possible were made to similar systems to provide the reader with a broader reference frame to position the proposed solutions inside. The specific verification and validation that could be performed on top of this is discussed in the next paragraphs.

Task Generation and Allocation

The task generation algorithm could be verified by comparing the results to the operational results of Mainblades. When applying the task generation algorithm to a narrow-body aircraft (A320) with the same drone and camera combination as employed by Mainblades, the task generation algorithm creates 195 camera positions that have to be visited. This is very similar to the number of images generated by Mainblades for an aircraft of the same type (in the range of 200 images). For a wide-body aircraft (Boeing 777) the task generation algorithm generates 568 images. This is also very similar to Mainblades which generates 600 to 700 images for the slightly larger Boeing 787.

The second method of verification for the generation and allocation algorithm is the visual inspection of the algorithm when the simulation is running. Because the simulation is completely visualised and interactive, mistakes or wrong results are easy to spot.

Positioning

The positioning algorithm is verified by comparing it to similar systems. The visual architecture is very similar to the positioning architecture employed by modern commercial drones [31]. The LIDAR positioning algorithm is also used by at least two direct competitors for aircraft inspection. These are Mainblades and Donecle [32]. The next steps for the design of the positioning system would be the experimental validation of the actual hardware and the algorithms.

Communication

This sizing assumes the most conservative scores for all the unknown variables. It is expected that when the actual hardware that is selected would be tested during an actual inspection, the available data rates could be an order of magnitude higher. This is supported by the capabilities of similar systems available on the market. For example, the DJI communication system for the Air 2S (a drone even smaller than the “DALE” drone) is able to achieve data rates of $12\,\mathrm{Mbit\,s^{-1}}$ over a range of $8\,\mathrm{km}$ [33] using almost identical transceivers and antennas. However, it is hard to size a system of similar capability with just publicly available information. Therefore, the authors are highly confident in the capability of the current system to meet the minimum requirements and expect the system to outperform the expectations significantly during hardware validation.

Integration with Fight Control Subsystem

In order to verify the correct functioning of the swarm control subsystem in combination with the flight control subsystem a software-in-the loop simulation could be performed where the flight control subsystem would perform the inner loop control of the drone response while the swarm simulation would take this inner loop response as input. Due to time limitations this was not feasible in the current design cycle. As a substitute, the emulated flight control system in the swarm control simulation was executed at an update rate of $200\,\mathrm{Hz}$ to verify stability and correct functioning.

One Drone Inoperative

It is required that the system can complete the full inspection of an aircraft after the failure of one drone. This requirement was verified in Subsection 5.4.3, as it can be seen from Figure 5.10 that the full inspection is completed under 30 minutes with just one “CHIP” drone or with just two “DALE” drones. This is the maximum possible inspection times, as it regards the case in which a drone fails at the beginning of the inspection. The times in case of failures of drone are reported in Table 5.9.

Table 5.9: Maximum inspection time after the failure of one drone.

Inoperative Drone	Inspection Time
“CHIP” drone	20 min 30 sec
“DALE” drone	26 min

[31] Retrieved from: `https://www.dji.com/nl/guidance/features` (cited 14 June 2022)

[32] Retrieved from: `https://www.donecle.com/solution/#howItWorks` (cited 14 June 2022)

[33] Retrieved from: `https://www.dji.com/nl/air-2s/specs` (cited 14 June 2022)

6 Flight Control

This chapter describes the flight control subsystem. Flight control is essential in ensuring stability in a wide range of flying conditions. Section 6.1 discusses flight control subsystem requirements. Section 6.2 provides background information on design aspects, followed by a trade-off between the different controller options. Section 6.3 analyses the simulation used, and evaluates the flight controller. Lastly, Section 6.4 discusses verification and validation of the flight control subsystem.

6.1. Requirements

Subsystem design requires knowledge of the subsystem's desired and required performance. To this end, subsystem requirements shall be specified to provide the designer with a performance benchmark. SUB-FC-01 and SUB-FC-10 have been set in accordance with system requirements on minimum clearance distance and wind requirements. Inspection system considerations were also extremely important since a too large distance would make adequate imaging impossible. SUB-FC-02 and SUB-FC-03 were set by industry-standard flight control capabilities, whilst also complying with system requirements (see for example the DJI Matrice 300). SUB-FC-04 flows directly from swarm control subsystem outputs.

Table 6.1: Flight control subsystem requirements derived from system and stakeholder requirements.

Identifier	Requirement	Related (Sub)System Requirements
SUB-FC-01	The system shall limit horizontal deflection due to wind, gusts up to $5\,\mathrm{m\,s^{-1}}$ to lower than $1\,\mathrm{m}$	SYS-PER-07
SUB-FC-02	The system shall have an indoor hovering accuracy of less than +- $(0.3\,\mathrm{m}\ ,\ 0.1\,\mathrm{m})$ [horizontal, vertical]	SUB-FC-07
SUB-FC-03	The system shall have an outdoor hovering accuracy of less than +- $(0.5\,\mathrm{m}\ ,\ 0.3\,\mathrm{m}\ )$ [horizontal, vertical]	SYS-PER-07, SUB-FC-07
SUB-FC-04	The system flight control response to velocity setpoint changes shall be sufficient to satisfy the requirement for inspection time	SYS-TIM-04
SUB-FC-05	The flight control system shall operate fully autonomously without manual human input during nominal operating conditions	SYS-PER-04
SUB-FC-06	The flight control system shall contain manual override for off-nominal operating conditions	StRS-RA-02
SUB-FC-07	The system shall not come closer to humans, and objects than specified in SYS-SAF-01, SYS-SAF-03, SYS-SAF-04, SYS-SAF-05 with an added margin for hovering accuracy (SUB-FC-02, SUB-FC-03)	SYS-SAF-01, SYS-SAF-03, SYS-SAF-04, SYS-SAF-05
SUB-FC-08	The flight control system control method shall have a TRL of at least 8	StRS-RA-02
SUB-FC-09	The flight control system shall be developed within maximum 5 weeks	SYS-TIM-02
SUB-FC-10	The system shall not come closer than $0.5\,\mathrm{m}$ to the inspected aircraft	SYS-SAF-01

6.2. Flight Control Design

This section describes flight control subsystem design and selection. Important criteria in selection of the control method are the performance and robustness in the field of application, TRL, and feasibility of implementation. Performance and robustness is the most important, since the drone will be operating near both multi-million dollar aircraft, and humans. Background information on quadcopter dynamics is provided in Subsection 6.2.1, an overview of control methods is provided in Subsection 6.2.2, the subsequent control methods trade-off is presented in Subsection 6.2.3. Subsection 6.2.4 discusses the control method used, and subsequent controller tuning.

6.2.1. Quadcopter Drone Dynamics

The state of a drone can be described using twelve state variables, namely six for translation and six for rotation. The state vector is thus as follows: $\vec{x} = [x, y, z, v_x, v_y, v_z, \phi, \theta, \psi, p, q, r]$. x, y, z specify the location of the drone in the inertial frame, and v_x, v_y, v_z describe its velocity in the inertial frame. The attitude state is generally described using the Euler angles ϕ, θ, ψ (rotations around the x, y, z-axes respectively). An alternative representation is provided by quaternions (see Equation B.1). Lastly, p, q, r represent the attitude rate (Ω) of the drone around the x, y, z-axes respectively. A visualisation of the drone body and inertial reference system, as well as motor designation, and geometrical parameters can be seen in Figure 6.1.

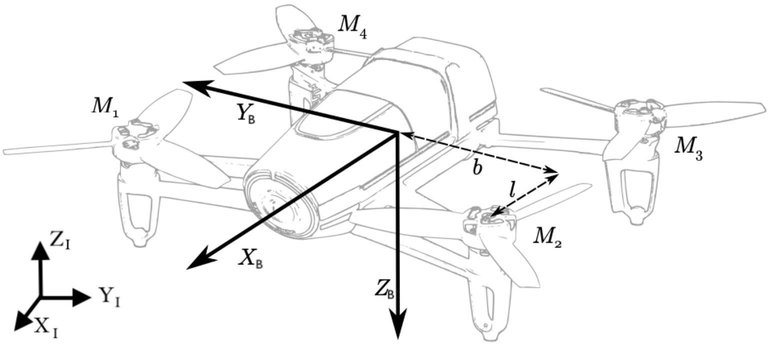


Figure 6.1: Quadcopter motor numbering, with motor distances from Centre of Gravity (CG) in x-direction (l), and y-direction (b) [65]. Positive directions of the x, y, z-axes are shown for both the inertial and body reference frames. A right-handed coordinate system is used, rotations are taken ccw+.

Basic drone dynamics can be divided into one for translation, and one for rotation. The equation for translation in the inertial frame (see Equation 6.1) consists of three main terms; propeller forces, gravity, and aerodynamic forces. See Equation B.2 for the full expression of these constituent terms.

$$\mathbf{F}_I = F_{prop} + F_g + F_{aero} = m\vec{a}_I \tag{6.1}$$

Equation 6.2 provides an expression for the rotational dynamics of a quadcopter drone. The left-hand side describes the dynamics of the drone, whereas the right side describes applied moments. $M_c(\omega_r)$ contains the moments due to motor angular rate (see Equation B.3), $M_a(\Omega, v)$ describes the moments due to aerodynamic effects, and $-M_r(\omega_r, \dot{\omega_r}, \Omega)$ describes the gyroscopic effect of the motors. A detailed description of drone dynamics can be found in [65], an implementation of drone aerodynamics can be found in the“Quadcopter Simulation and Control (Quad_SimCon)” by John Bass.[1]

$$I_v\dot{\Omega} + \Omega \times I_v\Omega = M_c(\omega_r) + M_a(\Omega, v) - M_r(\omega_r, \dot{\omega_r}, \Omega) \tag{6.2}$$

where I_v is the inertia tensor, ω_r is the angular rate vector of the motors, $\dot{\omega_r}$ is the angular acceleration of the motors, Ω is the angular rate of the drone, $\dot{\Omega}$ is the angular acceleration of the drone, and v is the velocity of the drone.

6.2.2. Overview Of Control Methods

The first and most commonly used control method is PID control. PID is versatile in application, and is used in many fields of industry (corresponding to a TRL of 9) [72]. PID control works well for most applications, is model-free, and has a large knowledge base for application. The introduction of time-varying disturbances introduces some difficulty. In general, disturbances are counteracted using the integral gain in the PID controller. Increasing this integral gain will increases the speed at which the system reacts to a disturbance, as well as increase the potential overshoot. When the disturbance disappears, the integral term does not immediately follow, leading to overshoot. One possible way to alleviate this problem is the introduction of some form of gain

[1] Retrieved from: `https://github.com/bobzwik/Quadcopter_SimCon` (cited 18 May 2022)

scheduling for no wind, constant wind and gusty conditions [8]. However, such an implementation might not be as robust and practical, as the use case might not allow for the prediction of when to use which gains[2].

MPC uses a model of the quadcopter to calculate an optimal control signal over a specified time horizon. This optimisation problem is recalculated at every time step to provide optimal control over a specified time horizon based on a model of the quadcopter. MPC has been demonstrated in many engineering systems, and therefore has a TRL of 9. The use of MPC was discussed with engineers at Fusion Engineering[3]. It was mentioned that MPC would add a layer of (control) optimisation on top of swarm control algorithms. Questions were raised as to whether this additional optimisation would result in clear performance gains, and whether or not these performance gains would outweigh the disadvantage of requiring additional computational resources. Since MPC is not part of the BSc Aerospace Engineering curriculum, it would also require additional resources to understand and implement as compared to PID.

Deep learning (AI) approaches were considered next. Although these methods are promising, immediate concerns were raised over their robustness. That is, certification of AI systems has proven difficult, considering the sometimes unpredictable nature of neural networks[4]. A brief look was taken at an approach called Neural-Fly [53]. Neural-fly proposes a deep learning approach for learning a representation of aerodynamic disturbances. Demonstrations have shown the technique to work for the control of a quadcopter in variable wind conditions (TRL 6, ”Prototype/engineering model tested in relevant environment”). Although this method shows increased performance over PID, additional concerns were raised over its TRL, as well as the feasibility of implementation within the DSE.

Lastly, Incremental Nonlinear Dynamic Inversion (INDI) was considered. INDI has great potential as a control method, given its strong inherent properties regarding (wind) disturbance rejection[5]. Fusion Engineering mentioned that their method has not yet been qualified for use in drones, corresponding to a TRL of 7[6]. Lastly, no group member initially had any experience with the method, though a lot of knowledge is present at the TU Delft Faculty of Aerospace Engineering [65, 73].

6.2.3. Control Methods Trade-Off

This subsection discusses the control methods trade-off. The trade-off considers three criteria; TRL, performance and robustness, and implementation difficulty. Firstly, the TRL shall give sufficient confidence that the method developed will be able to operate consistently within the requirements. Secondly, performance and robustness requirements shall be met. The drone shall not collide with any party during regular operation, in gusty conditions up to $5\,\mathrm{m\,s^{-1}}$, and in case of component failure. Additionally, the drone shall keep distance according to system requirements specification. Thirdly, implementation/adaptation of the method shall be possible using the knowledge of a 3rd-year aerospace engineering student for verification purposes.

Table 6.2 shows the control method trade-off, where the scoring method is provided in the table caption. PID scores highest with an average score of 2.5. It offers a similar TRL and performance to MPC, whilst also being easier to implement. As visible, INDI and AI control do not satisfy the trade-off criteria, and are thus not viable options.

It must be noted that the performance of the control system is not always the deciding factor in overall system performance. While simulation is capable of instantaneous control inputs, real-world system response depends on actuator dynamics. A small drone might therefore have a quicker actuator response relative to a larger drone. In consultation with experts, it was found that the size of the “CHIP” drone and “DALE” drone will be more of a deciding factor in real-world system response than the relative performance of control methods.

[2] Retrieved from: https://fusion.engineering/intro-to-incremental-non-linear-dynamic-inversion-indi/ (cited 18 May 2022)

[3] Private communication with Fusion Engineering, May 2022

[4] Fusion Engineering Private Communication, May 2022

[5] Retrieved from: https://fusion.engineering/intro-to-incremental-non-linear-dynamic-inversion-indi/ (cited 18 May 2022)

[6] Fusion Engineering Private Communication, May 2022

Table 6.2: Control method trade-off for the flight control subsystem. Trade-off criteria are TRL, performance and robustness, and feasibility of implementation. Meaning of scoring and colours: green (3) = exceeds required performance, blue (2) = meets required performance, red (1) = below required performance. Overall score: 1-1.5 (RED), 1.5-2.5 (BLUE), 2.5-3.0 (GREEN).

Option	**TRL**	**Performance and Robustness**	**Implementation Difficulty**	**Average score**
PID	TRL of PID is 9	PID is tested and provides reliable performance in aviation and drone industry.	Sources and implementations available. Covered in BSc course on control theory.	
Score	3	2	3	**2.5**
Colour	GREEN	BLUE	GREEN	GREEN
MPC	TRL of MPC is 9	Outperforms PID on optimal trajectories. Requires significantly more computational resources as compared to other methods.	Implementations (both Python and Matlab) available. Not covered in BSc program course on control theory.	
Score	3	2	2	**2.25**
Colour	GREEN	BLUE	BLUE	BLUE
INDI[7]	TRL of INDI is 7	Excellent performance in (wind) disturbance rejection. Stability can be be guaranteed for a linearised version.	Difficult. No sources on implementation in Python. Additionally only a few papers were found on quadcopter specific implementation.	
Score	1	3	1	**2.0**
Colour	RED	GREEN	RED	BLUE
AI	TRL of AI control is 6	Technique that has been shown to work gusty conditions. Certification and demonstrating robustness is difficult.	Implementation not feasible for 3rd year bachelor students.	
Score	1	1	1	**1.0**
Colour	RED	RED	RED	RED

An attempt was made to implement fault tolerant control in the form of Adaptive Incremental Nonlinear Dynamic Inversion (A-INDI). Fault tolerant control is extremely useful in preventing collisions in case of component failure. A-INDI is based on dynamic adaptation to quadcopter dynamics, and has demonstrated fault tolerance for one out of four motor failure in $6\,\mathrm{m\,s^{-1}}$ wind conditions[8] [65, 68]. Although promising, there are some concerns about how these methods would scale to larger drones in terms of mass and size[9]. The implementation, which was unfortunately unsuccessful due to time constraints, can be found in the project GitHub [10]. It is recommended to continue implementing A-INDI under supervision of experienced TU Delft staff as a post-DSE activity.

[8] Retrieved from: `https://www.youtube.com/watch?v=ScYDOqFGOhk` (cited 17 June 2022)

[9] Private communication with Fusion Engineering, May 2022

[10] Refer to GitHub > INDI (`https://github.com/DSE-Swarm-of-Inspection-Drones/Swarm-of-inspection-drones-rebase/tree/main/INDI`) for the implementation.

6.2.4. PID For Drone Control

This subsection will discuss in detail the PID implementation used for control of both drones, including the PID gains used for both the "CHIP" drone and "DALE" drone. An open-source flight controller and environment will be used to simulate the drone, and gain insight into its flight performance (e.g. rejection disturbance). The "Quadcopter Simulation and Control (Quad_SimCon)" by John Bass has been used, and modified to provide a realistic simulation[11]. The simulation takes into account basic quadcopter dynamics, as well as gyroscopic precession of the motors, and aerodynamic drag.

PID Implementation Block Diagram

The general form of a PID controller in the time-domain can be seen in Equation 6.3. The error, and its time-derivative and integral are multiplied by control gains (proportional, K_p; derivative, K_d; integral, K_i) to get a resultant control input. These control gains cannot be chosen arbitrarily, and should each be tuned so as to get a stable, well-damped system response.

$$u(t) = K_p e(t) + K_d \frac{de(t)}{dt} + K_i \int e(t)dt \tag{6.3}$$

A general overview of the control implementation used can be seen in Figure 6.2. First, an outer loop PID velocity controller calculates a reference thrust vector ($\mathbf{T}_{ref} = [T_x, T_y, T_z]^T$) based on a reference velocity input from the swarm control subsystem. A reference attitude can then be defined from the reference thrust vector and yaw angle (ψ_{ref}) using a quaternion representation (see Equation B.1). The error quaternion (attitude error) can subsequently be found through quaternion multiplication of the state quaternion conjugate and reference. A reference angular rate is computed by multiplying the attitude (quaternion) error with a proportional gain. Lastly, the reference angular rate is used as an input for the Proportional-Derivative (PD) attitude controller. The full control algorithm is specified in Section B.2. The reference thrust magnitude and attitude controller output can then be input into the motor mixer, which computes the desired motor speeds. This motor mixer is the inverse of the actuator dynamics (see Equation B.3). Note that this control algorithm does not require large computational resource, and can run on both the "CHIP" drone and "DALE" drone Flight Controllers (FCs).

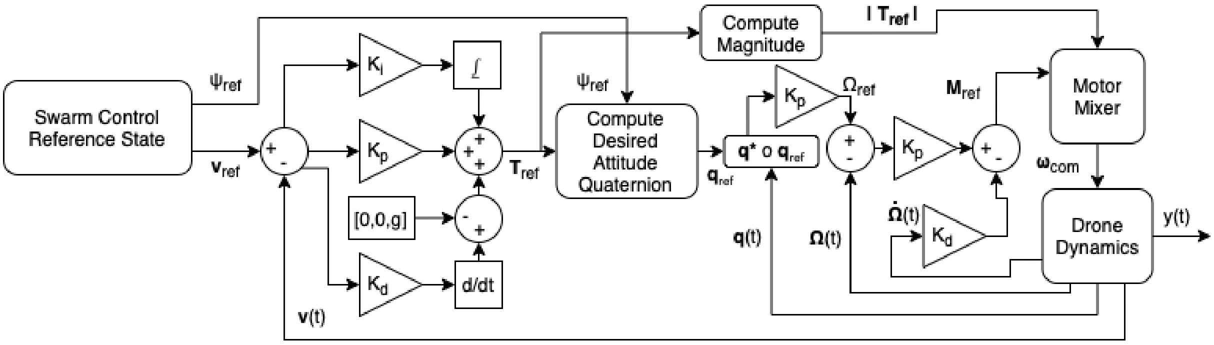


Figure 6.2: Block diagram of the PID control system used to control the drones. Bold commands and states are vector states, and **q*** o $\mathbf{q}_{ref}$ indicates quaternion multiplication of the conjugate of the state quaternion (**q**(t)) with the reference quaternion ($\mathbf{q}_{ref}$). y(t) is the full quadcopter state, ω_{com} is the commanded motor speeds vector, Ω_{ref} is the reference angular velocity, $\dot{\Omega}$ is the angular acceleration, $\mathbf{T}_{ref}$ is the reference thrust vector, ψ_{ref} is the reference yaw angle, and $\mathbf{v}_{ref}$ and **v**(t) are the reference and state velocities.

PID Tuning

Tuning of the PID controller is essential in ensuring a proper response to velocity setpoint inputs, and disturbances. Tuning was performed by inputting the "CHIP" drone and "DALE" drone parameters (see Table B.3) in the control simulation software[12]. The tuned PID parameters for the "CHIP" drone can be seen in Table 6.3, the tuned parameters for the "DALE" drone can be seen in Table 6.4. Gain scheduling will be implemented for altered drone configurations (e.g. a "CHIP" drone without a LIDAR model) [8]. This way changes in variables such as centre of gravity, mass, and mass moment of inertia can be accommodated.

[11] Retrieved from: `https://github.com/bobzwik/Quadcopter_SimCon` (cited 18 May 2022)

[12] Retrieved from: `https://github.com/bobzwik/Quadcopter_SimCon` (cited 10 June 2022)

Table 6.3: Tuned PID gains calculated for the "CHIP" drone control system. Refer to Figure 6.2 for the control system architecture, and Section B.2 for the control gains placement within the control system.

$(K_p)_x$	0.45	$(K_p)_y$	0.45	$(K_p)_z$	0.45
$(K_p)_{\dot{x}}$	9.5	$(K_p)_{\dot{y}}$	9.5	$(K_p)_{\dot{z}}$	9.5
$(K_d)_{\dot{x}}$	0.05	$(K_d)_{\dot{y}}$	0.05	$(K_d)_{\dot{z}}$	0.05
$(K_i)_{\dot{x}}$	4.0	$(K_i)_{\dot{y}}$	4.0	$(K_i)_{\dot{z}}$	4.0
$(K_p)_{\phi}$	8.0	$(K_p)_{\theta}$	8.0	$(K_p)_{\psi}$	1.5
$(K_p)_p$	1.5	$(K_p)_q$	1.5	$(K_p)_r$	1.0
$(K_d)_p$	0.04	$(K_d)_q$	0.04	$(K_d)_r$	0.1

Table 6.4: Tuned PID gains calculated for the "DALE" drone control system. Refer to Figure 6.2 for the control system architecture, and Section B.2 for the control gains placement within the control system.

$(K_p)_x$	0.65	$(K_p)_y$	0.65	$(K_p)_z$	0.6
$(K_p)_{\dot{x}}$	5.8	$(K_p)_{\dot{y}}$	5.8	$(K_p)_{\dot{z}}$	5.8
$(K_d)_{\dot{x}}$	0.15	$(K_d)_{\dot{y}}$	0.15	$(K_d)_{\dot{z}}$	0.5
$(K_i)_{\dot{x}}$	5.0	$(K_i)_{\dot{y}}$	5.0	$(K_i)_{\dot{z}}$	4.0
$(K_p)_{\phi}$	8.0	$(K_p)_{\theta}$	8.0	$(K_p)_{\psi}$	1.5
$(K_p)_p$	1.5	$(K_p)_q$	1.5	$(K_p)_r$	1
$(K_d)_p$	0.04	$(K_d)_q$	0.04	$(K_d)_r$	0.1

6.3. Flight Control Analysis

This section provides flight control system analysis. Subsection 6.3.1 briefly discusses the simulation used for analysing the performance of the flight control subsystem. Subsequently, Subsection 6.3.2 evaluates the flight control subsystem using hovering, and trajectory following metrics under varying wind conditions.

6.3.1. Flight Control Simulation

The quadcopter flight control simulation diagram is presented in Figure 6.3. The simulation starts by selecting the start and end time, and a time step. This will decide how fast the drone should fly. For example, if the difference between the end time and start time is small, the drone speed will increase and vice versa. After that quadcopter, trajectory, control, and wind are initialised. The quadcopter class encompasses structural and propulsion parameters such as mass (m), distance from motor to the drone centre of gravity(dx, dy, dz), mass moment of inertia (I_{ij}), thrust and torque coefficients (k_T, k_q), and thrust (T) and motor rotation speed (w) ranges. It also initialises drone states: position (x,y,z), quaternions (q), velocity ($\dot{x}, \dot{y}, \dot{z}$), acceleration ($\ddot{x}, \ddot{y}, \ddot{z}$), and orientation ($\phi, \theta, \psi$).

The class method "update" updates all the states after a command is sent from the control class. Motor dynamic equations and state derivatives that are used for updating the states can be seen in Table B.1 and Table B.2 respectively. For motor dynamics, rotational speed (w), damping factor (ξ), second-order time constant (τ) and motor gain (k_p) are used. For state derivatives combination of quaternions (q_1, q_2, q_3, q_4), drone rotational speed (p, q, r), wind parameters (q_{w1}, q_{w2}, V_w), thrust (T), moments due to thrust (M) and the inertial tensor (I_{ij}) are used. In trajectory class, firstly the waypoints are initialised. The drone only tries to make it to the first waypoint on the list. Once the drone reaches the point and stays approximately close to that point for a certain amount of time, the next point is then prioritised.

The control class uses a PID controller to control and send desired velocity, thrust, and attitude signal to the quadcopter class to update the states. The working of the PID controller can be seen in Figure 6.2, and is explained in Subsection 6.2.4. The wind class has 3 properties: wind velocity, heading, and elevation. The wind velocity can either be fixed, sine wave, or no wind. It is recommended to implement stochastic wind conditions in a future version to further demonstrate robustness of the control system. The properties from the wind are used in the state derivatives (quadcopter class) to update the drone state. Finally, while running the simulation, the drone state values are constantly being updated at each time step. It can be seen in the "Run simulation" block that it is a loop. It only ends once the time reaches the desired end time.

Multiple verification tests were performed to confirm the correct functioning of the program. Units tests include wind direction unit tests, single Euler angle step inputs, trajectory change unit tests, extreme value tests (e.g. large setpoint changes, and $180°$ roll/pitch angle initial state), and motor mixing unit tests. High-level verification include investigating the effect of including gyroscopic precession, and changes in drone parameters such as thrust coefficient (k_T), mass (m), and moment of inertia (I).

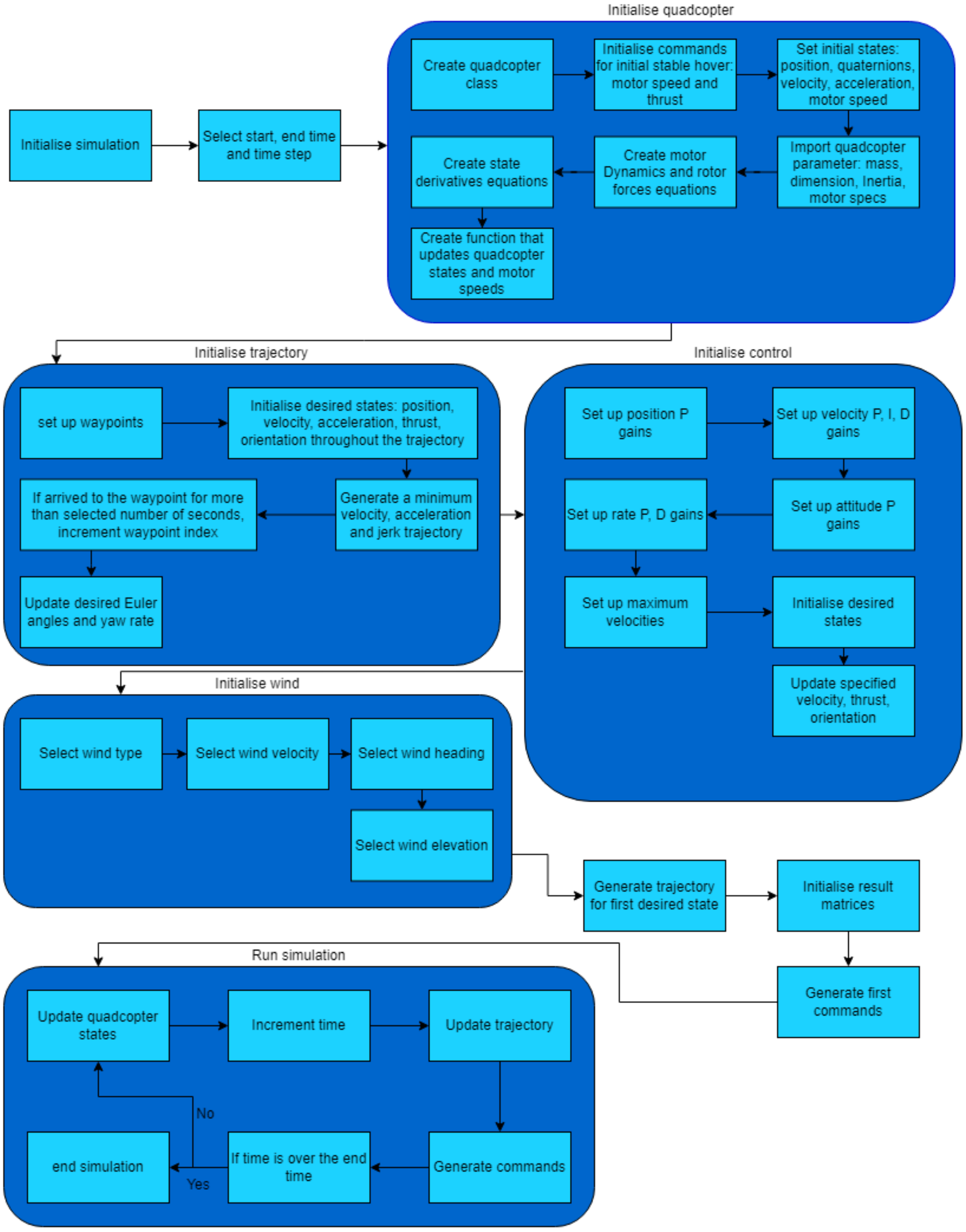
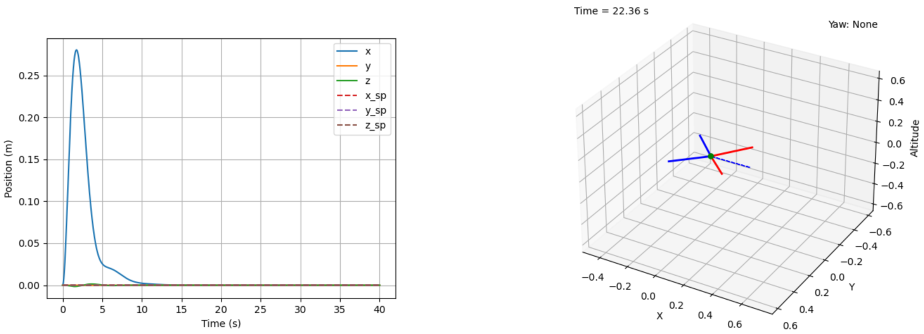

Initialise simulation
Select start, end time and time step
Initialise quadcopter
Create quadcopter class
Initialise commands for initial stable hover: motor speed and thrust
Set initial states: position, quaternions, velocity, acceleration, motor speed
Import quadcopter parameter: mass, dimension, Inertia, motor specs
Create motor Dynamics and rotor forces equations
Create state derivatives equations
Create function that updates quadcopter states and motor speeds
Initialise trajectory
set up waypoints
Initialise desired states: position, velocity, acceleration, thrust, orientation throughout the trajectory
Generate a minimum velocity, acceleration and jerk trajectory
If arrived to the waypoint for more than selected number of seconds, increment waypoint index
Update desired Euler angles and yaw rate
Initialise control
Set up position P gains
Set up velocity P, I, D gains
Set up attitude P gains
Set up rate P, D gains
Set up maximum velocities
Initialise desired states
Update specified velocity, thrust, orientation
Initialise wind
Select wind type
Select wind velocity
Select wind heading
Select wind elevation
Generate trajectory for first desired state
Initialise result matrices
Generate first commands
Run simulation
Update quadcopter states
Increment time
Update trajectory
Generate commands
If time is over the end time
No
Yes
end simulation

Figure 6.3: Drone flight control simulation diagram

6.3.2. Flight Control Accuracy

Firstly the flight control accuracy is tested on the “DALE” drone for hovering in no wind conditions. This is the most simple test, where the drone is instructed to stay in one place throughout the whole simulation time. The drone is perfectly capable of hovering at the specified waypoint with no positional errors. Applying $5\,\mathrm{m\,s^{-1}}$ wind in x direction, the drone drifts by $0.28\,\mathrm{m}$ from the specified waypoint, but then come backs after around $10\,\mathrm{s}$. It stays in the correct position for the rest of the simulated 30 s (Figure 6.4). Applying the wind in the y or z direction yields similar results, with the difference of drifting to the opposite direction to the wind.



(a) Position deflection plotted against time.

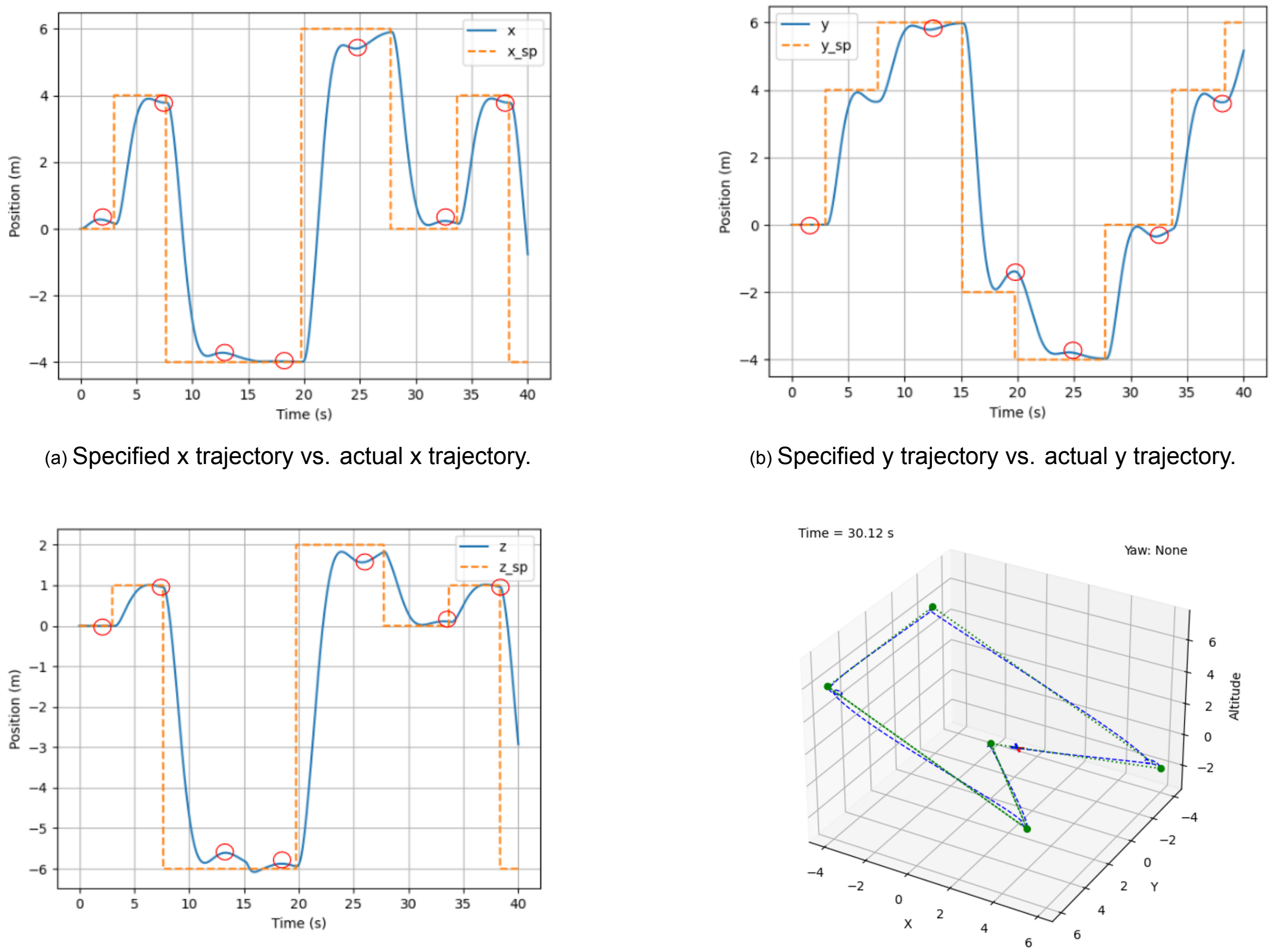
(b) Drone trajectory in 3D space over a simulated time of 22.36 seconds.

Figure 6.4: Hovering “DALE” drone with $5\,\mathrm{m\,s^{-1}}$ wind in x direction.

The last two tests are designed such that the drone tries to reach a specified waypoint by not exceeding its velocity of $5\,\mathrm{m\,s^{-1}}$, and once it stays within the $0.20\,\mathrm{m}$ range the new waypoint coordinates are sent to the drone. For every other point, it needs to maintain the $0.20\,\mathrm{m}$ range for at least $2\,\mathrm{s}$. This introduces more variety to the test. Both environment conditions are applied (no wind and wind) for this test. The specified and actual trajectory are displayed in Figure 6.5. Five waypoints are specified for the simulation. The trajectory errors, defined as the largest overshoot or undershoot for each waypoint interval, are calculated for each waypoint and can be seen as a red circles (Figure 6.5). It takes around $27.3\,\mathrm{s}$ to reach all the specified waypoints for the “DALE” drone. In Table 6.5 can be seen position errors when $5\,\mathrm{m\,s^{-1}}$ wind is applied in x direction. Applying wind in the y and z direction, the maximum deflection can be found among the 3 tests. The maximum horizontal error with the wind is $0.64\,\mathrm{m}$ and the vertical error is $0.44\,\mathrm{m}$. SUB-FC-01 requirement specifies that the drone shall limit horizontal deflection due to wind, gusts up to $5\,\mathrm{m\,s^{-1}}$ to lower than $1.00\,\mathrm{m}$. This requirement is met for the “DALE” drone. When there is no wind, the drone reacts in virtually the same way as with the wind, with just slightly lower maximum positional errors.

Table 6.5: “DALE” drone position errors with wind ($5\,\mathrm{m\,s^{-1}}$) in x direction

Waypoint number	Horizontal error $[\mathrm{m}]$	Vertical error $[\mathrm{m}]$
0	0.28	0.000
1	0.40	0.05
2	0.37	0.40
3	0.63	0.13
4	0.64	0.44



(a) Specified x trajectory vs. actual x trajectory.

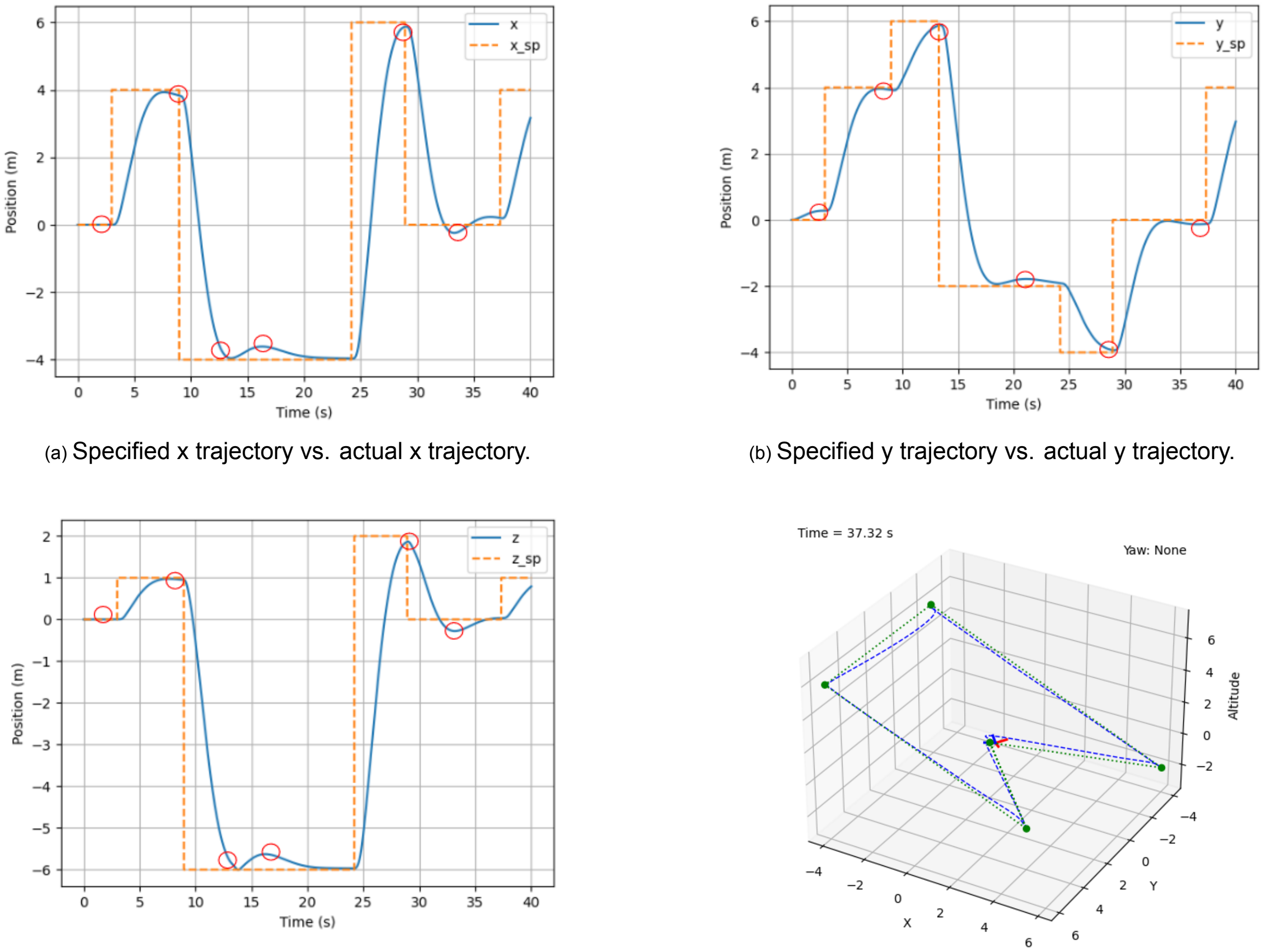
(b) Specified y trajectory vs. actual y trajectory.

(c) Specified z trajectory vs. actual z trajectory.

(d) 3D trajectory.

Figure 6.5: “DALE” drone following specified trajectory with wind ($5\,\mathrm{m\,s^{-1}}$) in x direction.

For the “CHIP” drone same test have been performed. The results for hovering in no wind is same as for “DALE” drone, where both drones stays in the specified waypoint with no deviation. The “CHIP” drone drifts by maximum of $0.28\,\mathrm{m}$ if the $5\,\mathrm{m\,s^{-1}}$ wind is applied (independently on the direction) while hovering, however it takes longer time than “DALE” drone to come back to specified position (18 s). Comparing “CHIP” drone with the “DALE” drone, “CHIP” drone has a better positional accuracy when following specified waypoints (Figure 6.6, Table 6.6). The maximum horizontal error is $0.45\,\mathrm{m}$ and vertical error of $0.37\,\mathrm{m}$. The only difference when no wind is applied is the average vertical error is $1.50\,\mathrm{cm}$ smaller than with applied wind. The maximum errors do not change that much (changes are within mm) when no wind is applied.



(a) Specified x trajectory vs. actual x trajectory.

(b) Specified y trajectory vs. actual y trajectory.

(c) Specified z trajectory vs. actual z trajectory.

(d) 3D trajectory

Figure 6.6: “CHIP” drone following specified trajectory with wind ($5\,\mathrm{m\,s^{-1}}$) in y direction.

Table 6.6: “CHIP” drone position errors with wind ($5\,\mathrm{m\,s^{-1}}$) in y direction

Waypoint number	Horizontal error $[\mathrm{m}]$	Vertical error $[\mathrm{m}]$
0	0.28	0.0
1	0.18	0.05
2	0.16	0.12
3	0.45	0.37
4	0.2	0.134

6.4. Flight Control Verification and Validation

This section provides verification and validation methods for the flight control subsystem requirements. Subsection 6.4.1 discusses verification of the flight control subsystem requirements, and Subsection 6.4.2 discusses future validation efforts. Subsection 6.4.3 discusses the sensitivity analysis performed on the flight control subsystem.

6.4.1. Verification

In Subsection 6.3.2 it has been verified that both drones have no errors in hovering accuracy when the wind is not applied (SUB-FC-02) and a maximum horizontal and vertical deflection due to $5\,\mathrm{m\,s^{-1}}$ of $0.28\,\mathrm{m}$ while hovering (SUB-FC-03). The biggest horizontal deflection a “DALE” drone experienced due to $5\,\mathrm{m\,s^{-1}}$ wind while following trajectory was $0.94\,\mathrm{m}$, where for “CHIP” drone was $0.57\,\mathrm{m}$ (SUB-FC-01). The maximum velocity during the inspection is around $2\,\mathrm{m\,s^{-1}}$, yet the drone complies with the position accuracy requirements while flying at a maximum of $5\,\mathrm{m\,s^{-1}}$, hence it is deemed to comply with the SUB-FC-04. It was shown that both drones can follow specified waypoints autonomously (SUB-FC-05). Since both drones comply with position accuracy requirements (SUB-FC-01, SUB-FC-02, SUB-FC-03), it means SUB-FC-07 and SUB-FC-10 are also met. It is however critical to take wind deflection into account when specifying waypoints, to ensure that the minimum object distance is kept. In the trade-off table, it was shown that the PID controller has a TRL of 9 (SUB-FC-08). The flight control system was developed within the 5 weeks (SUB-FC-09). The only requirement that requires further work is SUB-FC-06 since the manual override was not implemented into the simulation.

Cross-subsystem verification is recommended to be performed by integrating the flight controller into the inspection simulation (software-in-the-loop verification). This method can verify that no instabilities are introduced by combining the swarm and flight control methods. Moreover, it is important that requirements set on minimum object and personnel distance are met using the full software stack. Lastly, this method can be used to make an estimate on the increase of inspection time due to gust-induced inspection path deviations.

6.4.2. Validation

By building and analysing a drone prototype, flight control can be validated. Again multiple waypoints can be set in the environment where the drone could freely move in the horizontal and vertical directions and can be checked if the drone is following the waypoints as desired. During autonomous flying, a human or an outside object shall interface with the trajectory of the drone to see if the drone avoids it with enough distance and continues on its trajectory. Flying a drone outside will also test its control in windy conditions and it can be seen how the drone reacts. An operator will also have a controller, so the override functionality can be tested.

6.4.3. Sensitivity Analysis

A sensitivity analysis is performed on the flight controller, to prove its robustness and reliability to uncertainty. The first sensitivity test is performed by varying the drone mass within 20%. For “DALE” drone reducing the mass by 20% reduces the maximum horizontal error by $9\,\mathrm{cm}$ and the vertical error by $4\,\mathrm{cm}$, however it reaches all the waypoints slower by $6.1\,\mathrm{s}$ in comparison to original “DALE” drone mass. Increasing the “DALE” drone mass by 20% does not change the maximum horizontal error, but increases the vertical error by $11\,\mathrm{cm}$ and reduces the time to cover all the waypoints by $3.7\,\mathrm{s}$. Increasing the mass does not have significant influence on the hovering test. The response time is same as with the default time, and reducing mass reduces the hovering error by $1\,\mathrm{cm}$, while increasing the mass increases hovering error by $1\,\mathrm{cm}$.

Doing the same analysis for the “CHIP” drone, 20% mass reduction leads to no changes in maximum positional errors, however to reach all five waypoints it takes $13.5\,\mathrm{s}$ more. By increasing mass by 20% the maximum horizontal error increases by $13\,\mathrm{cm}$, and the vertical error by $10\,\mathrm{cm}$. Additionally, reaching five waypoints takes $3.1\,\mathrm{s}$ less. Mass change again does not change results for the hovering test, where the maximum errors as well as time to go back to specified waypoint due to wind deflection are the same as before the mass change.

Changing mass moment of inertia matrix by 20% does not affect the control characteristics in any meaningful way, for both “CHIP” drone and “DALE” drone. The hovering, and waypoint tests provide the same results as with the original inertia matrix.

In conclusion, changing the drone mass by 20% has a small influence on the maximu mm positional errors, and the time it takes to go over the specified trajectory. The drone still stays stable even with these changes, and its trajectory looks almost the same as those depicted in Figure 6.5 and Figure 6.6. It is recommended to perform a more extensive sensitivity analysis at a later stage, including control sensitivity to centre of gravity shift, and single motor out.

7
Propulsion and Power

In order to let the drones perform their inspection, a propulsion and power system needs to be designed. Therefore, this chapter elaborates on the design of this system for the “CHIP” drone and “DALE” drone. In Section 7.1, some background information about this system and the selection procedures will be described. Then, in sections 7.2, 7.3, 7.4, and 7.5, the requirements and the selection of the propellers for both drones will be presented.

7.1. Background Information

The design of the propulsion and power subsystem of both drones will be done according to a number of steps. First, the propeller is going to be selected. The selection of the propeller will be done based on the Maximum Take-Off Mass (MTOM) of the drone. Next, the selection of a suitable motor for the propellers will take place. In order to select a motor for the drone, the RPM of the propeller during a T/W equal to 2 needs to be known. This specific maximum T/W allows the drone to hover at 50% throttle, such that the drone is more responsive to disturbances during hovering flight [9, 74]. These drones are to be designed as a quadcopter, such that there are four propellers and motors. After the motor has been selected, the battery can be selected. The battery selection requires an estimate of what the average T/W is during the flight and the power consumption of the motors at that specific T/W, the power consumption of the payload, and the rated voltage of the motor. The selection of the ESC will take place after battery selection. For the ESC selection, the maximum current going into the motors must be known.

7.1.1. Propeller Selection Procedure

In order to select a suitable propeller, the frame size of the drone, and the MTOM must be known. The frame size of the drone defines a maximum allowable diameter of the propeller, such that the propellers do not touch each other when rotating. It is desired that the diameter of the propeller is as large as possible, as this would reduce the power required to rotate the propeller at a specific RPM during hovering flight. This is illustrated by Equation 7.1, which shows the ideal power required for hovering:

$$P_{hov_{ideal}} = \sqrt{\frac{T^3}{2\rho A}} \tag{7.1}$$

in which T is the thrust of the propeller, ρ is the air density ($1.225\,\mathrm{kg\,m^{-3}}$), and A is area of the propeller disk [61].

Therefore, the first step in propeller selection is to determine the maximum allowable propeller diameter. The

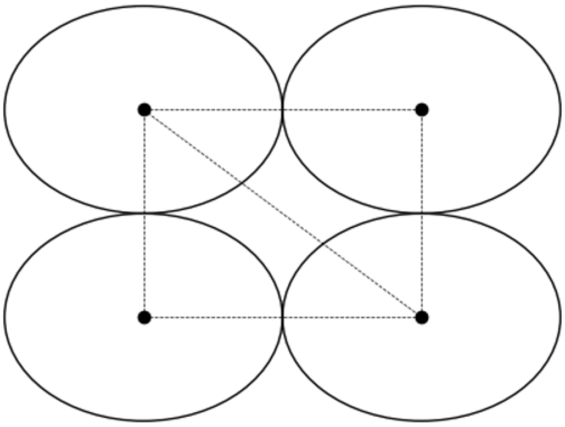


Figure 7.1: The frame size of a quadcopter, represented by the diagonal line.

frame size is defined as the distance between the top left motor and the bottom right motor of the quadcopter. The frame size is therefore the diagonal line in Figure 7.1. Furthermore, the horizontal and vertical distances between the motors are the same. Hence, the angle between the diagonal and horizontal line is $45°$. Therefore, the propeller diameter can be calculated according to Equation 7.2:

$$D_{prop} = \frac{\sqrt{2}}{2} d_{FS} \tag{7.2}$$

in which d_{FS} is the frame size (from motor to motor).

After the maximum propeller diameter is known, data about propellers with a similar diameter can be collected. This data consists of the thrust coefficient (C_T) and power coefficient (C_P) at different RPM values, during hovering, or static conditions. This data is taken from a propeller database from the University of Illinois at Urbana-Champaign [10]. Additional data has been taken from propeller manufacturers[1] [2].

Based on the values of C_T and C_P, the thrust (N) of a propeller, and power required (W) to drive the propeller can be calculated. This can be done according to Equations 7.3 and 7.4:

$$T = C_T \rho n^2 D_{prop}^4 \tag{7.3}$$

$$P_{req} = C_P \rho n^3 D_{prop}^5 \tag{7.4}$$

In these equations, D_{prop} is the diameter of the propeller in meters, and n is the number of revolutions per second [61]. The value of the thrust is then divided by $9.81\,\mathrm{m\,s^{-2}}$, and multiplied by 4 to obtain the total thrust of the drone in kilogram. From Equation 7.3 it can be observed that the thrust is a function of the revolutions per second squared. Therefore, a quadratic relation between the revolutions per second and the propeller thrust can be estimated. Using this quadratic relation, the amount of RPM needed to obtain a T/W of 2 can be estimated. Furthermore, according to a simulation of the flight of the inspection drones, described in Section 5.4, the T/W during the flight will be equal to 1. Therefore, the RPM needed to achieve a T/W of 1 is estimated as well. Note that a T/W equal to 1 represents hovering conditions. Additionally, by setting up a cubic relation between the revolutions per second and the power required to drive the propeller, P_{req} can be estimated. This will be done by substituting the RPM values to achieve a T/W of 1, and 2 respectively in the cubic equation.

Then, the propeller selection will be done based on the FOM of the propeller. The FOM is a measure of the efficiency of the propeller and is defined as the ratio between the ideal power required to hover and the actual power used to drive the propeller. The FOM is calculated using Equation 7.5 [61]:

$$FOM = \frac{C_T^{3/2}}{\sqrt{2} C_P} \tag{7.5}$$

The propeller with the highest FOM at the operating RPM value (T/W = 1) is selected as the most suitable propeller for this drone design. However, the mass of the propeller must adhere to the mass budgets.

7.1.2. Motor Selection Procedure

After the propeller has been selected, the motor operating point can be defined. The operating point of a motor is the torque that a motor needs to provide such that the propeller can rotate at a specific RPM. For the motor selection, the operating point is the RPM needed to achieve a T/W equal to 2. Then, the torque can be calculated using Equation 7.6:

$$\mathrm{T} = \frac{P_{out}}{\omega_{prop}} \cdot 9.554140127 \tag{7.6}$$

where the fraction is multiplied by $9.554\,140\,127$ in order to obtain the torque in $\mathrm{N\,m}$. Note that the power output of the motor, P_{out}, is equal to the power required to drive the propeller, P_{req}.

After the operating point has been calculated, it is necessary to collect data about suitable motors for these drones. This data included the speed constant, K_V (RPMV^{-1}), the internal resistance of the motor, R_{motor} (Ω),

[1] Retrieved from: https://www.apcprop.com/technical-information/performance-data/ (cited 31 May 2022)

[2] Retrieved from: https://www.mejzlik.eu/?gclid=CjwKCAjw7vuUBhBUEiwAEdu2pGjveWOQ1stcvVzW-ehPXxFr7MLLsUL7eiqy3uoTqEpC9p_HLGskJxoCSZoQAvD_BwE (cited 31 May 2022)

the no-load current, I_0 (A), and the applied voltage at no-load conditions. The no-load condition means that there is no propeller connected to the shaft of the motor. Furthermore, I_0 is measured at a specific voltage, which is defined by the motor manufacturer[3]. The list of motor options is compiled with the help of the eCalc tool[4], using the initial configurations of the drones and the selected propeller from Subsection 7.1.1. Using the motor data, it is possible to calculate the no-load RPM of the motor and the stall torque of the motor. During stall conditions, the motor does generate torque, but the RPM is zero[5]. The RPM of the motor can be calculated using Equation 7.7 [67]:

$$RPM = (V_{applied} - IR_{motor})K_V \tag{7.7}$$

If the no-load current is substituted in this equation, then I_0R_{motor} is the voltage drop at no-load conditions. Furthermore, if K_V is converted to SI-units $(\mathrm{rad/(sV)})$, then the torque constant, K_T $(\mathrm{N\,m\,A^{-1}})$, can be obtained by inverting K_V[6].

Furthermore, the stall torque can be calculated using Equation 7.8 [67]:

$$\mathrm{T}_{stall} = K_T I_{stall} - \mathrm{T}_f \tag{7.8}$$

in this equation, I_{stall} is the stall current. During stall conditions, the shaft of the motor is not rotating. Therefore, according to Equation 7.7, $V_{applied}$ equals IR_{motor}, as K_V is non-zero. Thus, the stall current can be obtained by dividing the applied voltage by the motor resistance. T_f is the friction torque. This is the torque necessary to overcome the friction in the motor. The friction torque can be calculated using I_0. If there is no load attached to the motor, then the only element which needs to be overcome is the friction. The friction torque can be calculated using Equation 7.9:

$$\mathrm{T}_f = K_T I_0 \tag{7.9}$$

Now that the no-load RPM and the stall torque are known, an RPM-torque curve can be constructed. The available motors should be able to function at the specific operating point. Therefore, the operating point is also plotted in the RPM-torque curve of a specific motor. Then, the applied voltage will be adjusted such that the RPM-torque crosses the operating point. This adjusted voltage is the voltage needed for the motor to function at the operating point. For all the values of the torque on the RPM-torque curve, the input current to the motor can be calculated using Equation 7.10 [67]:

$$I_{motor} = \frac{\mathrm{T} + \mathrm{T}_f}{K_T} \tag{7.10}$$

Then, the input power to the motor is calculated according to Equation 9.1. The output power of the motor can be calculated using Equation 7.11 [30]:

$$P_{out_{motor}} = (I_{motor} - I_0) \cdot (V_{applied} - I_{motor}R_{motor}) \tag{7.11}$$

Then, the efficiency of the motor is the ratio of the output power and the input power, see Equation 7.12:

$$\eta_{motor} = \frac{P_{out_{motor}}}{P_{in_{motor}}} \cdot 100\% \tag{7.12}$$

The power loss across a motor is caused by copper and iron losses. The copper losses are caused by the resistance in the windings of the motor. The iron losses are due to magnetic effects. The copper and iron losses can be calculated using Equations 7.13 and 7.14 [9]:

[3] Retrieved from: `https://www.radiocontrolinfo.com/brushless-motor-efficiency/brushless-motor-no-load-current/` (cited 8 June 2022)

[4] Retrieved from: `https://www.ecalc.ch/xcoptercalc.php` (cited 8 June 2022)

[5] Retrieved from: `https://icrservices.com/2019/12/12/what-even-is-stall-torque/` (cited 8 June 2022)

[6] Retrieved from: `https://ipfs.io/ipfs/QmXoypizjW3WknFiJnKLwHCnL72vedxjQkDDP1mXWo6uco/wiki/Motor_constants.html` (cited 8 May 2022)

$$P_{copper} = R_{motor}I_{motor}^2 \quad (7.13) \qquad P_{iron} = V_{applied}I_0 \quad (7.14)$$

During these calculations, the efficiency might be overestimated. This is due to the fact that the no-load current is measured at a specific voltage, which is done by the manufacturer. Therefore, at different values of voltage, the no-load current might be different. In order to minimise this difference, the applied voltage at T/W equal to 2 should be close to the voltage at which the no-load current was measured at.

The selection motor will be done according to the assigned mass budgets, and power consumption at a T/W equal to 1. Furthermore, the motor must be compatible with the propeller size, and the applied voltage to achieve a T/W of 2 cannot exceed the rated voltage of the motor.

There are two types of motors available: brushed and brushless motors. Both motor types rely on the same working principle; electromagnetism. When current is applied to a coil, a temporary magnetic field is created. Additionally, there are permanent magnets present. The temporary magnetic field attracts and repels the permanent magnet (the north pole attracts the south pole and vice-versa), such that the shaft of the motor starts to rotate. Then, after half a turn, the direction of the temporary magnetic field flips, such that the motor keeps rotating. Brushed motors use a commutator to flip this magnetic field[7]. A brushless motor does not use a commutator and instead uses the ESC to apply a voltage to different magnets[8]. Hence, a brushless motor eliminates a point of failure, the commutator, and is therefore more reliable and durable. Furthermore, the commutator of a brushed motor generates sparks. This can be an additional safety hazard for the aircraft. Therefore, for this design, brushless motors will be selected[9].

7.1.3. Battery Selection Procedure

Many different types of batteries are available. These battery types include nickel-cadmium batteries, lead-acid batteries, lithium-ion, and lithium-polymer batteries. For drones, the Lithium Polymer (LiPo) battery is commonly used [10]. This is due to their large specific energy ($\mathrm{W\,h\,kg^{-1}}$), which is higher than the specific energy of the aforementioned batteries [30]. Therefore, the LiPo battery will be used for the design of both drones.

Once the motors are selected, the required battery capacity can be estimated. A flight simulation of a drone shall be performed to estimate this capacity. This simulation provides the T/W values during the flight. As discussed in Section 5.4, the T/W during the flight is equal to 1. Therefore, for this T/W condition, the power consumption of the motor must be known. This will be done using a similar procedure for a T/W equal to 2, described in Subsection 7.1.2.

The RPM of the motor is controlled by the ESC. The ESC changes the voltage that is applied to the motor. This changes the rotational speed of the motor. The power output of the ESC is equal to the power input of the motor. However, the are some power losses across the ESC. An ESC has an efficiency of 95%, and therefore the power input to the ESC can be calculated using Equation 7.15 [30]:

$$P_{in_{\text{ESC}}} = \frac{V_{motor}I_{motor}}{\eta_{esc}} \tag{7.15}$$

where $V_{motor}I_{motor}$ is the output power of the ESC, which is the same as the input power to the motor. $P_{in,\text{ESC}}$ is the power consumed by the motor, and ESC combination.

The battery must also provide power to the payload components, and the sensors. Therefore, these components have to be included in the capacity estimation as well. The payload power consumption is reported in Table 9.2 and Table 9.3. From these tables, the total power consumption of the payload can be calculated. After the power consumption of the motor-ESC combination and the payload is known, the total power consumption can be multiplied by the flight time (in hours). This calculation yields the power consumption in $\mathrm{W\,h}$.

Multiple battery cells can be connected in series, which will increase the voltage of the battery. The nominal voltage of one LiPo battery cell is $3.7\,\mathrm{V}$, and the number of cells connected in series is denoted by an ‘S’. It is also possible that some cells are connected in parallel, which increases the capacity of the battery. This is denoted by a ‘P’[11]. The voltage of the battery cannot exceed the rated voltage of the motor or ESC, otherwise,

[7] Retrieved from: https://electronics.howstuffworks.com/motor.htm (cited 9 June 2022)

[8] Retrieved from: https://uav.jreyn.net/quadcopter-design/step-5-motor-selection (cited 9 June 2022)

[9] Retrieved from: https://www.kdedirect.com/blogs/news/brushless-vs-brushed-motors) (cited 9 June 2022)

[10] Retrieved from: https://www.powerelectronicsnews.com/the-need-for-battery-safety-systems-for-drones/ (cited 9 June 2022)

[11] Retrieved from: https://www.genstattu.com/bw/ (cited 9 June 2022)

these components can be damaged. Most COTS batteries have a voltage of 1S to 12S ($3.7\,\mathrm{V}$-$44.4\,\mathrm{V}$).

The required capacity of a battery can be estimated using Equation 7.16 [14]:

$$C_{battery} = \frac{E}{V_{battery}} \tag{7.16}$$

From this equation, it can be observed that the capacity of the battery varies with the voltage of the battery. Therefore, the capacity will be calculated for each S-configuration. According to these capacities, some COTS battery options can be analysed. However, in order to prolong the lifetime of a LiPo battery, the batteries can only be discharged up to 70%-80% of the total battery capacity [3, 30]. Therefore, the capacity of the COTS must be larger than the capacity calculated by Equation 7.16.

The most suitable batteries for this drone design are determined by the size of the battery, as the battery must fit in the drone structure, and by the weight of the battery. Additionally, all batteries have a discharge rating, or C-rating. This rating determines how fast the battery can be discharged in a safe manner[12]. For example, if the battery has a capacity of 2,500$\mathrm{mA\,h}$, and the discharge rating is 25C, then the battery can be discharged at 25 times the capacity ($62.5\,\mathrm{A\,h}$). Therefore, this discharge rating must be higher than the maximum current draw of the motors, considering that the motors will draw the largest current.

7.1.4. Electronic Speed Controller Selection Procedure

As discussed in Subsection 7.1.3, the ESC controls the rotational speed of the motor. Some important parameters for selecting the ESC are the maximum continuous current and the voltage range supported by the ESC. The maximum continuous current refers to the operating conditions at full throttle, the T/W of 2. However, in order to avoid burning of the ESC, a safety margin must be added to the maximum continuous current. Therefore, the maximum continuous current is multiplied with a safety factor of 1.7 [59]. Furthermore, the ESC must be able to withstand the voltage provided by the battery.

7.2. Propeller Selection

In this section, the propeller will be selected for the “CHIP” drone and the “DALE” drone. The results of the propeller selection will be used to select a motor. In order to determine which propellers are the most suitable for both drone types, some requirements will be set up. Based on these requirements, an analysis can be performed. The most suitable propeller follows from this analysis. The analysis follows the procedure as described in Subsection 7.1.1.

7.2.1. Propeller Requirements

From the initial drone configurations, it was determined that the frame size of the “CHIP” drone is 1,000 mm, and that the frame size of the “DALE” drone is $540\,\mathrm{mm}$. According to Equation 7.2, the maximum allowable propeller diameters of the drones are $707\,\mathrm{mm}$, and $381\,\mathrm{mm}$. Most propeller sizes are given in inches, therefore the maximum diameters are 27.8 and 15.0 inch respectively. The requirements for the propellers of both drones are presented in the following table:

Table 7.1: Power and propulsion subsystem requirements for the propeller.

Identifier	Requirement	Related System Requirements
SUB-PROP-1	The propeller diameter of the “CHIP” drone shall not exceed 27.8 inch ($70.8\,\mathrm{cm}$)	
SUB-PROP-2	The mass of one propeller of the “CHIP” drone shall not exceed $133\,\mathrm{g}$	SYS-SIZ-02
SUB-PROP-3	One propeller of the “CHIP” drone shall lift a mass of 2,410 g during hovering	SYS-PER-01, SYS-PER-02, SYS-PER-03

[12] Retrieved from: `https://www.genstattu.com/bw/` (cited 9 June 2022)

SUB-PROP-4	One propeller of the “CHIP” drone shall lift a mass of 4,820 g during T/W equal to 2 conditions	
SUB-PROP-5	The power required to drive one propeller of “CHIP” drone shall not exceed 268 W during hovering conditions	SYS-SIZ-02
SUB-PROP-6	The propeller diameter of the “DALE” drone shall not exceed 15.0 inch (38.1 cm)	
SUB-PROP-7	The mass of one propeller of the “DALE” drone shall not exceed 58 g	SYS-SIZ-02
SUB-PROP-8	One propeller of the “DALE” drone shall lift a mass of 935 g during hovering	SYS-PER-01, SYS-PER-02, SYS-PER-03
SUB-PROP-9	One propeller of the “DALE” drone shall lift a mass of 1.870 g during T/W equal to 2 conditions	
SUB-PROP-10	The power required to drive one propeller of “DALE” drone shall not exceed 125 W during hovering conditions	SYS-SIZ-02
SUB-PROP-11	The RPM of the propellers shall not exceed the maximum RPM defined by the manufacturer	
SUB-PROP-12	While rotating, the propellers shall not collide with another propeller	
SUB-PROP-13	While rotating, the propellers shall not collide with the drone structure	

7.2.2. Propeller Selection “CHIP” drone

In order to minimise the power required during hovering, the diameter of the propeller should be as large as possible. In Subsection 8.5.3 it was shown that if the propeller diameter is more than 27 inch, the propellers will collide. From the available databases, the APC 26x13E, the APC 26x15E, and the Mejzlik 26x8.7 propellers were analysed. Propellers with a diameter lower than 26 inch have not been analysed as the power required during hover was considered too high.

The propeller thrust, power required, and the FOM at the operating points were calculated, using Equations 7.3, 7.4, and 7.5. It was found that the Mejzlik 26x8.7 propeller has the highest FOM value. The FOM was 0.637 for a T/W equal to 1. For the APC 26x13E propeller, the FOM was 0.477. The FOM for the APC 26x15E propeller was 0.382. These values are considerably lower than the FOM of the Mejzlik 26x8.7 propeller. For this propeller, the thrust-RPM curve and the power required-RPM curve are presented in Figure 7.2 and Figure 7.3 respectively.

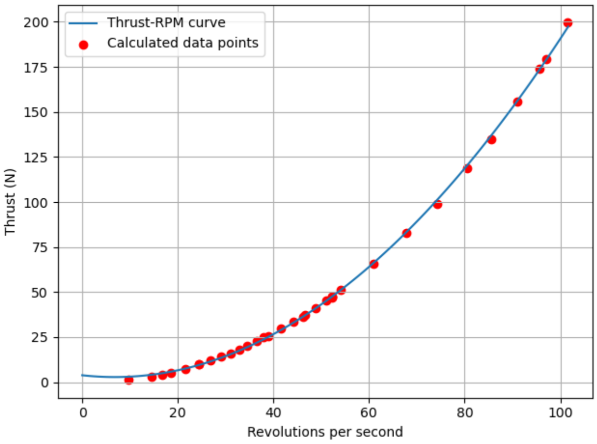


Figure 7.2: The thrust-RPM curve for the Mejzlik 26x8.7 propeller.

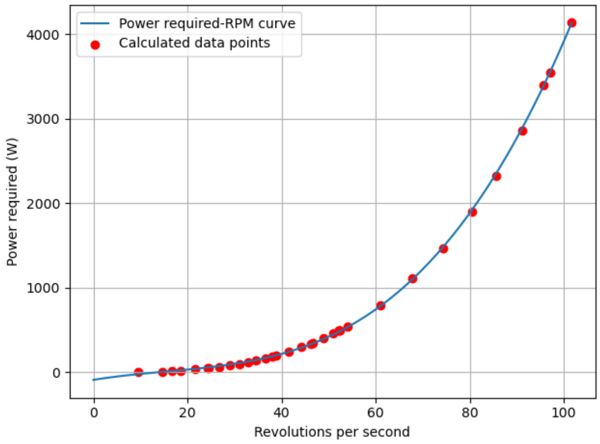


Figure 7.3: The power required-RPM curve for the Mejzlik 26x8.7 propeller.

For this propeller, the RPM at T/W equal to 1 is 2,270, and the power required is 188 W. The RPM is 3,130 for a T/W of 2, and the power required is 481 W. The weight of a single propeller is 80 g. The maximum allowable RPM of this propeller is 6,900, as specified by the manufacturer [57]. Thus, during a T/W between 1 and 2, the maximum allowable RPM will not be exceeded.

7.2.3. Propeller Selection “DALE” drone

The maximum allowable propeller diameter of this drone is 15 inch, according to requirement SUB-PROP-6. However, the two front arms of the “DALE” drone are mounted lower than the core structure. If the propeller diameter is 15 inch, the propellers mounted on the front arms will collide with the drone structure. Furthermore, extending the arms is not possible, due to size constraints. Therefore, the maximum allowable propeller diameter of this drone was determined to be 11 inch. Hence, all available propellers with a diameter of 11 inch in the databases reported in Subsection 7.1.1 were analysed. For these propellers, the FOM at the T/W of 1 was calculated. The results of these calculations are presented in Figure 7.4. From this figure, it can be observed that the propeller with index 3, the Aeronaut CAM 11x6 propeller, has the highest FOM. Therefore, this propeller is selected for the “DALE” drone. The FOM for this propeller is 0.667.

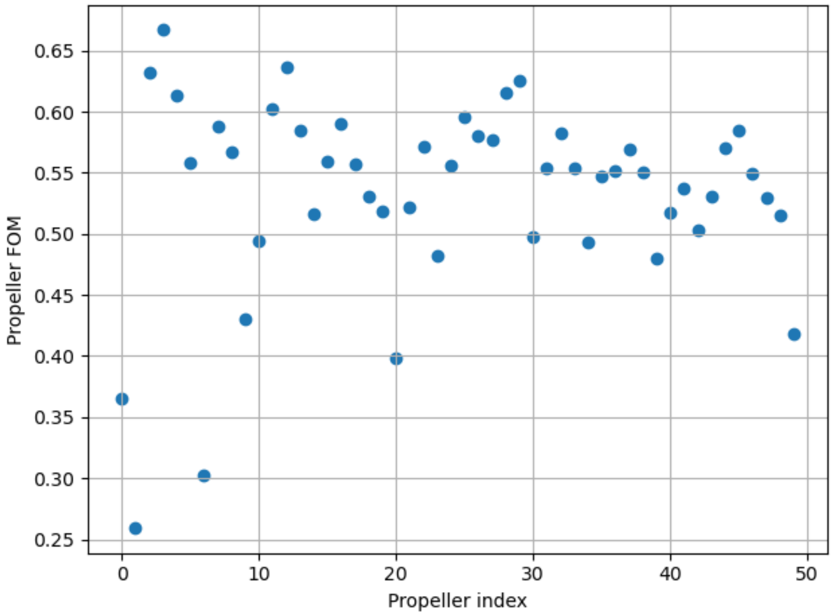


Figure 7.4: The FOM for 11 inch propellers at a T/W equal to 1.

For this propeller, the thrust-RPM and the power required-RPM curve are reported in Figure 7.5 and Figure 7.6. From these figures, it can be observed that at a T/W of 1, the RPM is 6,300 and the power required is 98 W. At a T/W of 2, the propeller RPM is 9,000, and the power required is 291 W. The maximum allowable RPM of this propeller is 13,000[13]. This limit will not be exceeded at a T/W of 1 and 2. Although data about the weight of this particular propeller could not be retrieved, the weight of similar 11 inch carbon fibre propellers is 15 g-20 g[14,15]. Therefore, it is assumed that the weight of this propeller is 20 g.

[13] Retrieved from: `https://www.voltmaster.de/Aeronaut-CAM-Carbon-11x6` (cited 3 June 2022)

[14] Retrieved from: `https://hobbyking.com/en_us/carbon-fiber-propeller-11x4-7-black-cw-ccw-2pcs.html` (cited 3 June 2022)

[15] Retrieved from: `https://robu.in/product/orange-hd-propellers-114711x4-7-carbon-fiber-props-1cw1ccw-1pair-black-2/` (cited 3 June 2022)

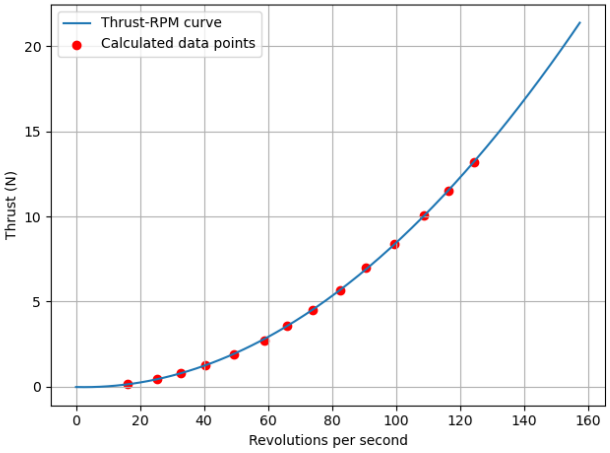


Figure 7.5: The thrust-RPM curve for the Aeronaut CAM 11x6 propeller.

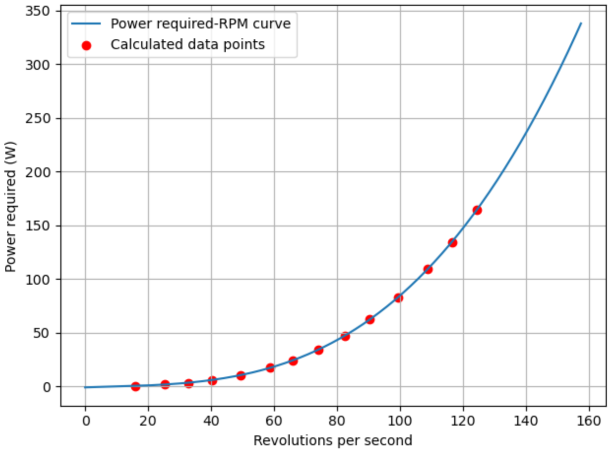


Figure 7.6: The power required-RPM curve for the Aeronaut CAM 11x6 propeller.

7.3. Motor Selection

Now that the propellers have been selected, the motors for both drones can be selected. First, some requirements will be set up. Some of these requirements are derived from the propeller selection. The requirements are reported in Table 7.2. According to these motor requirements, the motors will be selected.

Table 7.2: Power and propulsion subsystem requirements for the motor.

Identifier	Requirement	Related System Requirements
SUB-MOT-1	A single motor of the “CHIP” drone shall provide an RPM of 2,270 during hovering conditions	
SUB-MOT-2	A single motor of the “CHIP” drone shall provide an RPM of 3,130 at a T/W of 2	
SUB-MOT-3	A single motor of the “CHIP” drone shall provide an output power of at least 188 W during hovering conditions	
SUB-MOT-4	A single motor of the “CHIP” drone shall provide an output power of at least 481 W at a T/W of 2	
SUB-MOT-5	A single motor of the “CHIP” drone shall not weigh more than 260 g	SYS-SIZ-02
SUB-MOT-6	A single motor of the “DALE” drone shall provide an RPM of 6,300 during hovering conditions	
SUB-MOT-7	A single motor of the “DALE” drone shall provide an RPM of 9,000 during at a T/W of 2	
SUB-MOT-8	A single motor of the “DALE” drone shall provide an output power of at least 98 W during hovering conditions	
SUB-MOT-9	A single motor of the “DALE” drone shall provide an output power of at least 291 W at a T/W of 2	
SUB-MOT-10	A single motor of the “DALE” drone shall not weigh more than 90 g	SYS-SIZ-02
SUB-MOT-11	The voltage applied to the motor shall not exceed the rated voltage of the motor	
SUB-MOT-12	The motor shall be a brushless motor	
SUB-MOT-13	The motor shall be compatible with the propeller diameter	
SUB-MOT-14	The motor input power shall not exceed the maximum rated power of the motor	

7.3.1. Motor Selection “CHIP” drone

For the motor selection, the T/W of 2 operating point will be used to judge whether the motor is suitable for the “CHIP” drone. The torque at this operating point is calculated using Equation 7.6. At this operating point, the torque is 1.47 N m. For each of the possible motors, a torque-RPM curve was constructed, using Equations 7.7 and 7.8. The T/W equal to 2 operating point was also plotted in these curves. From these curves, the applied voltage was obtained. The current at this operating point is obtained by Equation 7.10. Using the current and voltage, the power consumption is obtained. If the power consumption exceeds the rated power of the motor, then the motor is not suitable for the “CHIP” drone.

Additionally, the voltage supplied to the motor during hovering conditions is obtained. At the hovering operating point, the torque is 0.79 N m. Then, at the different operating points, the current, and the motor input power are calculated. These values are reported in Table 7.3.

Table 7.3: The voltage, current, and input power to the motor at different operating points for the “CHIP” drone.

	Mad Components M8 IPE V2 (R_{motor} = $0.147\,\Omega$, I_0 = 0.99 A, K_V = 100 RPM/V, mass = 280 g)	MAD Components M8 C08 IPE (R_{motor} = $0.071\,\Omega$, I_0 = 1.44 A, K_V = 150 RPM/V, mass = 280 g)	RC Timer HP8108 - 135 (R_{motor} = $0.137\,\Omega$, I_0 = 0.6 A, K_V = 135 RPM/V, mass = 239 g)	MAD Components 8108 EEE (R_{motor} = $0.190\,\Omega$, I_0 = 0.74 A, K_V = 100 RPM/V, mass = 261 g)
Applied voltage (T/W=1)	24.3 V	16.1 V	18.5 V	24.6 V
Current (T/W=1)	9.3 A	13.9 A	11.8 A	9.0 A
Power input (T/W=1)	225 W	223 W	218 W	222 W
Applied voltage (T/W=2)	33.9 V	22.8 V	26.4 V	34.5 V
Current (T/W=2)	16.4 A	24.5 A	21.3 A	16.0 A
Power input (T/W=2)	554 W	558 W	564 W	553 W

From Table 7.3, it can be observed that the power consumption of the motors at different operating points is quite similar. At hovering conditions, the RC Timer HP8108 motor has the lowest power consumption. Furthermore, it can be seen that this motor is much lighter than the other motors. Therefore, this motor is selected for the “CHIP” drone. During hovering conditions, the copper and iron losses are 19 W and 11 W respectively. Additionally, at a T/W of 2, the power consumption of the motor does not exceed the maximum rated power of the motor, which is 750 W.

7.3.2. Motor Selection “DALE” drone

At a T/W of 2, the torque provided by the motor must be 0.31 N m. Using a similar procedure as described in Subsection 7.3.1, the voltage, current, and input power were determined. These are reported in Table 7.4.

Table 7.4: The voltage, current, and input power to the motor at different operating points of the “DALE” drone.

	Aeolian C3536-KV1050 (R_{motor} = $0.055\,\Omega$, I_0 = 1.5 A, K_V = 1,050 RPM/V, mass = 100 g)	Aeolian C3536-KV910 (R_{motor} = $0.075\,\Omega$, I_0 = 1.2 A, K_V = 910 RPM/V, mass = 100 g)	BrotherHobby Avenger 2812-0900 V3 (R_{motor} = $0.0725\,\Omega$, I_0 = 1.12 A, K_V = 900 RPM/V, mass = 79 g)	T-Motor CINE66-925 (R_{motor} = $0.096\,\Omega$, I_0 = 1.05 A, K_V = 925 RPM/V, mass = 77 g)

Applied voltage (T/W=1)	7.0 V	8.2 V	8.1 V	8.4 V
Current (T/W=1)	17.8 A	15.4 A	15.1 A	15.5 A
Power input (T/W=1)	125 W	126 W	123 W	130 W
Applied voltage (T/W=2)	10.6 V	12.3 V	12.2 V	12.8 V
Current (T/W=2)	35.5 A	30.7 A	30.2 A	31.0 A
Power input (T/W=2)	376 W	378 W	368 W	397 W

Table 7.4 also shows that the power consumption of the multiple motor options is quite similar. In order to comply with requirement SUB-MOT-10, either the Brotherhobby Avenger 2812-0900 V3 or the T-Motor CINE66-925 motor can be selected. As the Brotherhobby Avenger motor consumes the least amount of power, this motor is selected for the "DALE" drone. During hovering conditions, the copper and iron losses are 16 W and 9 W respectively. Additionally, the rated power of this motor is 804 W[16], which is not exceeded.

7.4. Battery Selection

Now that the power consumption of the motors are known during hovering and the payload power consumption has been given in Table 9.2 and Table 9.3, the size of the battery can be determined. Furthermore, according to the requirements, the flight time of the drone must be 30 minutes. The requirements for the battery can be seen in Table 7.5.

Table 7.5: Power and propulsion subsystem requirements for the battery.

Identifier	Requirement	Related System Requirements
SUB-BAT-1	The battery of the "CHIP" drone shall provide a flight time of at least 30 minutes	SYS-TIM-04
SUB-BAT-2	The mass of the battery of the "CHIP" drone shall not exceed 4,155 g	SYS-SIZ-02
SUB-BAT-3	The battery of the "CHIP" drone shall provide a power of 230 W to a single motor, and ESC combination of the "CHIP" drone for 30 minutes, during hovering conditions	SUB-BAT-1, SYS-TIM-04
SUB-BAT-4	The battery of the "CHIP" drone shall provide a power of 104 W to the payload of the "CHIP" drone for 30 minutes	SUB-BAT-1, SYS-TIM-04
SUB-BAT-5	The battery of the "DALE" drone shall provide a flight time of at least 30 minutes	SYS-TIM-04
SUB-BAT-6	The mass of the battery of the "DALE" drone shall not exceed 1,900 g	SYS-SIZ-02
SUB-BAT-7	The battery of the "DALE" drone shall provide a power of 129 W to a single motor and ESC combination of the "DALE" drone for 30 minutes, during hovering conditions	SUB-BAT-5, SYS-TIM-04
SUB-BAT-8	The battery of the "DALE" drone shall provide a power of 80 W to the payload of the "DALE" drone for 30 minutes	SUB-BAT-5, SYS-TIM-04
SUB-BAT-9	The voltage provided by the battery shall not exceed the rated voltage of the motor-ESC combination	

[16] Retrieved from: `https://www.brotherhobbystore.com/avenger-2812-v3-motorcw-p0112.html` (cited 3 June 2022)

SUB-BAT-10	The discharge rating of the battery shall exceed the maximum current draw of the motors	
SUB-BAT-11	During a flight, the battery shall not be discharged more than 75% of the total battery capacity	

The power consumption of the motor-ESC combination has been determined by Equation 7.15. The input power to the motor for the "CHIP" drone is $218\,\mathrm{W}$, and the motor input power for the "DALE" drone is $123\,\mathrm{W}$. Then, using an efficiency of 95%, the values reported in SUB-BAT-3 and SUB-BAT-7 are obtained.

7.4.1. Battery Selection "CHIP" drone

The power consumption of all of the components on the "CHIP" drone is $1.024\,\mathrm{W}$. This includes the power consumption of the four motor-ESC combinations and the power consumption of the payload. Taking into account a flight time of (at least) 30 minutes, the total power consumption is $512\,\mathrm{W\,h}$. Furthermore, the rated voltage of the RC Timer HP8108-135 is 6S-12S ($22.2\,\mathrm{V}$-$44.4\,\mathrm{V}$).

Taking into account that the battery will not be discharged more than 75%, the following battery capacities are needed:

Table 7.6: Total battery capacity per cell configuration.

	Battery capacity ($\mathrm{mA\,h}$)
6S	31,000
7S	27,000
8S	24,000
9S	21,000
10S	19,000
11S	17,000
12S	16,000

Furthermore, from the the drone structure, it has been determined that the battery dimensions cannot exceed 220×170×80mm (length×width×height). The most suitable battery was found to be the 'GENX 44.4V 12S 16,000$\mathrm{mA\,h}$' battery. The dimensions of this battery are 185×146×75mm. The resulting battery configuration can be seen in Table 7.7 below.

Table 7.7: Battery configuration for the "CHIP" drone.

Type	**Cell configuration**	**Mass** [g]	**Rated Voltage** [V]	**Total energy** [W h]
Lithium polymer	12S, 1P, 16,000$\mathrm{mA\,h}$	3,810	44.4	710.4

The discharge rating for this battery is 25C. Thus, the maximum continuous discharge that the battery can provide is $400\,\mathrm{A}$. This is higher than the maximum current draw of the motors and therefore complies with requirement SUB-BAT-10.

7.4.2. Battery Selection "DALE" drone

The power consumption of all of the "DALE" drone is $598\,\mathrm{W}$. The total power consumption of the drone is thus $299\,\mathrm{W\,h}$, using a flight time of 30 minutes. The rated voltage of the BrotherHobby Avenger 2812 V3 motor is 5S-8S ($18.5\,\mathrm{V}$-$29.6\,\mathrm{V}$). The battery will not be discharged more than 75%. The required battery capacities can be seen in Table 7.8.

Table 7.8: Battery capacity per cell configuration

	Battery capacity $(\mathrm{mA\,h})$
5S	22,000
6S	18,000
7S	16,000
8S	14,000

From the structure, it has been given that the maximum allowable battery dimensions are 200×100×$100\,\mathrm{mm}$. The most suitable battery for the "DALE" drone is the 'Liperior 16,000$\mathrm{mA\,h}$ 6S 12C 22.2V' battery. The battery dimensions are 187×74.5×$62\,\mathrm{mm}$. The battery configuration for the "DALE" drone is presented in Table 7.9.

Table 7.9: Battery configuration for the "DALE" drone.

Type	**Cell configuration**	**Mass** $[\mathrm{g}]$	**Rated Voltage** $[\mathrm{V}]$	**Total energy** $[\mathrm{W\,h}]$
Lithium polymer	6S, 1P, 16,000$\mathrm{mA\,h}$	1,850	22.2	355.2

The discharge rating for this battery is 12C. The maximum continuous discharge is $192\,\mathrm{A}$, which is higher than the maximum current draw of the motors.

It can be observed that this selected battery is not sufficient to provide a 30 minutes flight time. The usable energy of this battery is $266.4\,\mathrm{W\,h}$, which is less than the required $299\,\mathrm{W\,h}$. Using this battery, the available flight time of this drone is 27 minutes. However, if a battery is selected with a bigger capacity, then the battery weight will increase to at least 2,100 g. This means that SUB-BAT-6 will not be complied with and that the total drone weight will also increase by an unacceptable amount. Furthermore, the inspection time of an aircraft was determined to be 16 minutes, as described in Section 5.4. Therefore, it is unnecessary to take a weight penalty, as the "DALE" drone is capable of flying for more than 16 minutes.

7.5. ESC Selection

Now that the motors and the batteries have been selected, the best performing ESC can be selected. The ESC selection will be performed based on the maximum continuous current going into the motors at a T/W of 2, multiplied by a safety factor of 1.7. This maximum continuous current can be found in Table 7.3 and Table 7.4. The ESC requirements can be seen in Table 7.10 below.

Table 7.10: Power and propulsion subsystem requirements for the ESC.

Identifier	**Requirement**	**Related System Requirements**
SUB-ESC-1	The ESC of the "CHIP" drone shall withstand a maximum continuous current of $36.2\,\mathrm{A}$	
SUB-ESC-2	The ESC of the "CHIP" drone shall be compatible with a 12S battery	
SUB-ESC-3	A single ESC of the "CHIP" drone shall not weigh more than $25\,\mathrm{g}$	SYS-SIZ-02
SUB-ESC-4	The ESC of the "DALE" drone shall withstand a maximum continuous current of $51.3\,\mathrm{A}$	
SUB-ESC-5	The ESC of the "DALE" drone shall be compatible with a 6S battery	
SUB-ESC-6	A single ESC of the "DALE" drone shall not weigh more than $22.5\,\mathrm{g}$	SYS-SIZ-02

7.5.1. ESC Selection "CHIP" drone

For the ESC of the "CHIP" drone, the 'MAD AMPX 40A ESC HV' was chosen. This ESC can support a continuous current of up to $40\,\mathrm{A}$, and is compatible for 5S-14S batteries. This particular ESC has a weight of $22.5\,\mathrm{g}$. In total, there are four of these ESCs needed for the "CHIP" drone. Thus, the total weight of the ESCs is $90\,\mathrm{g}$.

7.5.2. ESC Selection “DALE” drone

For the ESC of the “DALE” drone, the ‘DarwinFPV Betaflight F4 60A 3-6S BLHeli_32 4in1 Dshot1200’ ESC was chosen. This ESC supports a continuous current of up to $60\,\mathrm{A}$ and is compatible with 3S-6S batteries. The weight of this particular ESC is $12.7\,\mathrm{g}$. This is a 4-in-1 ESC and therefore there is one needed for this drone.

7.6. Swarm Noise Estimation

According to requirement SYS-SAF-06, the noise produced by the drone swarm shall not exceed $90\,\mathrm{dB(A)}$, measured at one meter. Therefore, a noise estimation has to be performed for the swarm. This will be done by first estimating the noise produced by one propeller of the “CHIP” drone and one propeller of the “DALE” drone. Then, the total noise produced by the drone can be calculated, and an A-weighting is applied. This A-weighting is applied to correct for the subjective perception of noise by a human ear[17]. The total noise produced by the swarm can be calculated, at a certain reference point. The noise estimation is based on the hovering condition, as the drones will have a T/W equal to 1 throughout the flight. In order to estimate the noise of a propeller, the following steps will be followed [47]:

1. First, a reference level, L_1, has to be obtained. This reference level is based on the power input to the propeller, in horsepower. This reference level is based on figure B-2 of [47] and can be calculated according to the following relation:
$$L_1 = 91.37585 P_{req}^{0.049707}$$
where P_{req} is the amount of power required by the propellers to sustain a T/W of 1. L_1 is the reference level in dB.

2. Next, the reference level will be corrected for the number of blades and the propeller diameter. This correction factor, L_2 will be added to L_1. This factor can be calculated as follows:
$$L_2 = 20\log_{10}\left(\frac{4}{B}\right) + 40\log_{10}\left(\frac{15.5}{D_{prop}}\right)$$
where B is the number of blades of the propeller, which is 2 for both drones. The propeller diameter is in feet. L_2 is calculated in dB.

3. Another correction factor, L_3, has to be obtained, which accounts for the rotational speed of the propeller. The tip Mach number of the propeller can be calculated as follows:
$$M_t = \frac{B\pi\omega_{prop_{hov}}}{60}\frac{1}{c}$$
where c is the speed of sound in feet per second. Then, according to figure B-3 of [47], the correction factor L_3 (dB) can be obtained.

4. Correction factor L_4 will be added for the direction of sound propagation from a propeller. This is based on the angle with the propeller heading. Then, from figure B-8 of [47], the correction factor can be obtained in dB. For this estimation, it is assumed that the observer will be below the propeller. Hence, the heading angle is higher than $90°$. The maximum value of the correction factor is $4\,\mathrm{dB}$, at a heading angle of $120°$.

5. Finally, correction factor L_5 can be added. This correction factor is to account for the spherical spreading of sound and can be calculated according to:
$$L_6 = -20\log_{10}(r-1)$$
where r is the distance between the observer and the centre of the propeller, in feet. L_6 is calculated in dB.

6. The Sound Power Level (SPL) of the propeller can be obtained by summing the reference level L_1 and all the correction factors together. By adding the SPL of the four propellers together, the total SPL of the drone can be calculated:
$$SPL_{drone} = 10\log_{10}\left(10^{SPL_{prop1}/10} + 10^{SPL_{prop2}/10} + 10^{SPL_{prop3}/10} + 10^{SPL_{prop4}/10}\right)$$

[17] Retrieved from: `https://www.ansys.com/blog/what-is-a-weighting` (cited 14 June 2022)

7. In order to apply an A-weighting, the blade passing frequency, f, must be calculated according to:

$$f = B\frac{\omega_{prop_{hov}}}{60}$$

8. Now the A-weighting can be calculated according to the following equations [25]:

$$R_A(f) = \frac{12194^2 f^4}{(f^2+20.6^2)\sqrt{(f^2+107.7^2)\,(f^2+737.9^2)}\,(f^2+12194^2)}$$

   The A-weighting correction factor is obtained by:

$$A(f) = 20\log_{10}(R_A(f)) + 2.00$$

   in which f is the blade passing frequency (Hz).

9. By adding the A-weighting factor to the SPL of the drone, the A-weighted SPL is obtained $(\mathrm{dB(A)})$.

10. The total A-weighted noise of the swarm is obtained by summing all the noise contributions of the individual drones together. This can be done according to:

$$SPL_{swarm} = 10\log_{10}\left(10^{SPL_{CHIP1}/10} + 10^{SPL_{CHIP2}/10} + 10^{SPL_{DALE1}/10} + 10^{SPL_{DALE2}/10} + 10^{SPL_{DALE3}/10}\right)$$

By varying the distance r from 1 to $40\,\mathrm{m}$, and keeping L_4 at $4\,\mathrm{dB}$, the A-weighted SPL can be obtained for various distances between the observer and the drones. This is visualised in Figure 7.7.

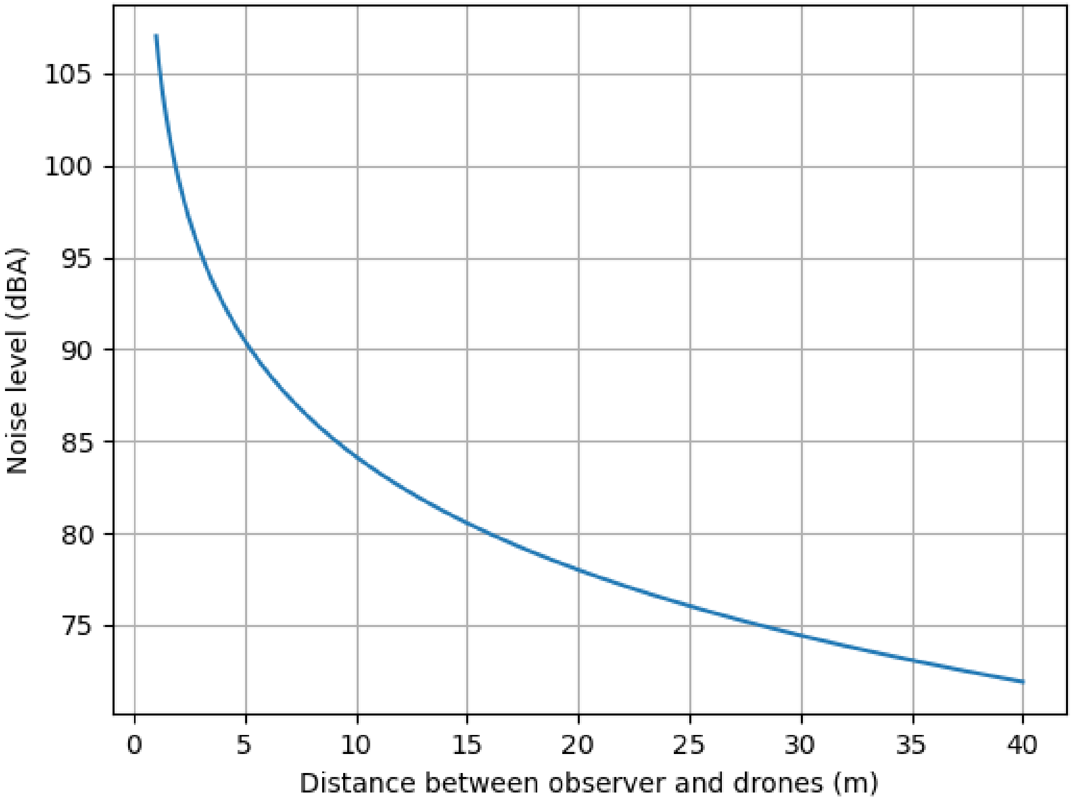


Figure 7.7: The SPL of the swarm calculated at various distances.

7.7. Verification and Validation

In order to ensure that the methods and tools used during the design of the propulsion and power subsystem are correct, some verification and validation will be performed. Validation tests will compare calculated data against real-world data.

7.7.1. Verification

Python and Excel files were created for subsystem design. For each of these files, it was checked that the formulas, variables, and references to variables are correct. Furthermore, during the noise computations, some logarithms were used, which requires extra attention during verification. A number of the computed noise values by the Python program were calculated by hand, and the example calculation in [47] has been performed in the Python program.

7.7.2. Validation

During the propeller selection, relations between thrust and RPM and between power required and RPM were established. As these relations form the basis of the propulsion and power subsystem design, they have to be validated. In addition to the C_T and C_P data, the APC propeller manufacturer also provided the measured thrust, and required power of the propeller. This data has been used to validate the calculations performed by Equations 7.3 and 7.3. The difference between the calculated and measured data was at most 4%, and has been accounted for by selecting a higher RPM than indicated by Equation 7.3, and by selecting a battery with excess capacity.

Furthermore, the noise produced by a single “DALE” drone or “CHIP” drone was compared to measured drone noise data, reported in [62]. This database gives the noise of a drone measured at a distance of one meter. From this database, it has been observed that the noise of the individual drones in the swarm is overestimated in Section 7.6. Therefore, after a prototype of the swarm has been constructed, some in-field noise measurements of the swarm need to be performed to provide a definitive answer.

7.7.3. Verification of Requirement SYS-SAF-06

Currently, the noise produced by the swarm measured at one meter is $107\,\mathrm{dB(A)}$. The noise produced by a single “CHIP” drone is estimated to be $80\,\mathrm{dB(A)}$, and the noise produced by a single “DALE” drone is estimated to be $102\,\mathrm{dB(A)}$. The noise of the “DALE” drone is relatively high, due to the high RPM of the propellers. Therefore, at this stage in the design, it is unknown whether this particular requirement is met. Hence, real-world noise measurements need to be performed to confirm this.

According to Dutch law, the noise measured in an ear (in $\mathrm{dB(A)}$), cannot exceed $87\,\mathrm{dB(A)}$[18]. From Figure 7.7, it can be observed that the noise produced by the swarm reduces below $87\,\mathrm{dB(A)}$ after $7\,\mathrm{m}$. It is expected that during the inspection of an aircraft, that the operator will not be closer than $7\,\mathrm{m}$, and certainly not $1\,\mathrm{m}$ as specified by SYS-SAF-06, to all the drones at any time.

It is recommended to research low-noise blade technologies, such as applying serrations to the trailing edges of the propeller blades. A study has shown that during hovering conditions, the serrations can reduce the noise level up to $1.6\,\mathrm{dB}$. The power and thrust coefficients will not change much by the serrations, but the RPM of the propeller needs to increase slightly due to a drag increase on the blade [52].

[18] Retrieved from: `https://www.arboportaal.nl/onderwerpen/geluid` (cited 15 June 2022)

8
Structural Design

In essence, the structural subsystem contains all the parts that have a weight-carrying function. It enables load transfer from one location to another and is supposed to do so for the duration of the product's life. Furthermore, the structure is preferred to be lightweight, rigid, sturdy, and so on.

This chapter will begin with Section 8.1 covering the principles of structural mechanics, which will provide background understanding for the rest of the chapter. Then Section 8.2 will state the various subsystem requirements by which the structure will have to abide by. Next, the considerations which need to be thought of before any design decisions are explained in Section 8.3. This is followed by Section 8.4, where the first decisions for the structural design are made. With the groundwork covered, the designed drone frame will be presented and several aspects discussed in Section 8.5. Afterwards, the auxiliary aspects such as EOL and operations are discussed in Section 8.6. And the final Section 8.7 will take care of checking the structures' compliance with the subsystem requirements.

8.1. Structural Mechanics Background

When designing a structure, the mechanics that occur when the structure is placed under loads are important to consider. To understand the theory which underpins the mechanics of a structure, it would be convenient to have some knowledge of the mechanics of materials. This first section will serve to provide the reader with an overview of some fundamental concepts in the mechanics of materials.

The foremost important concept which should be understood is stress. When an undeformed body is subjected to a load, for example, a force, the body will deform until it reaches an equilibrium. At this equilibrium, a state of internal forces has built up, and these forces exert internal stresses on the material in the body. These stresses are expressed in the SI unit Pascals (Pa), and come in 6 components: 3 tensile stresses and 3 shear stresses. For analysis purposes, it is convenient to combine all these stress components into a single criterion which can provide an easily understandable number. One such value is the von Mises criterion, which will be used as the standard stress criterion throughout this chapter. So note that whenever stress is mentioned, it is implicitly referring to the von Mises stress. To determine whether a structure yields or not, the internal stress is compared with the materials' yield strength. Note, however, that von Mises is more applicable to ductile materials, and not for the polymer composite material which is used later on. But due to the difficulty of deriving other stress criteria, and the small gain in effectiveness in using another criteria, it is more convenient to simply use the von Mises stress[1].

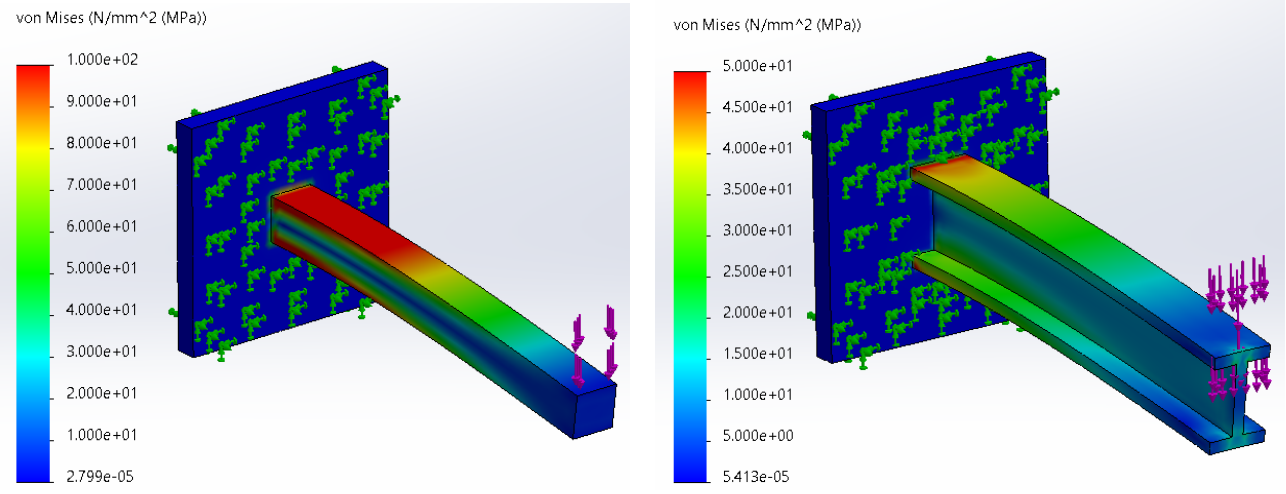
Depending on the structures' geometry and the loading conditions, the distribution of stress throughout the structure will be different. Some examples of stress distribution in structures are shown in Figure 8.1. Notice the difference in stress between the two pictures. Even though the beam in both pictures spans the same length, use the same amount of material, and experience identical forces, the magnitude and distribution of stresses are vastly different.

This is an excellent example of how clever design can efficiently reduce the magnitude of stress experienced. Referring to Figure 8.2, note how the cross-sectional area of the I-beam has more material further away from the red axis than the square beam. This redistribution of the material allows the section to have a higher moment of inertia, which improves its ability to resist bending.

This concept of redistributing material away from the axis of bending is widely used in structural mechanics, to improve the weight efficiency of a structure. In the context of drone design, one case where this would be useful is in making the propeller arms as light as possible.

Another concept that can help in improving the efficiency of a structure is the concept of stress concentrations. Stress concentrations are zones in a structure where the stress rises significantly compared to its' surrounding. This can happen when cutouts are created in a structure, or when there are abrupt changes in cross-sectional area. Looking back at Figure 8.1, at the root of both beams, there are sharp angles between the beam and the wall. The effect that this has on stress concentrations is most apparent in Figure 8.1b, where the area of

[1] C. Nastos Konstantopoulos Private Communication, June 2022



(a) Stress concentrations of a square beam.

(b) Stress concentrations of an I-beam.

Figure 8.1: Generic picture of stress concentrations in a bending beam test, it can be seen that stresses in a square beam are overall higher than stresses in an I-beam.

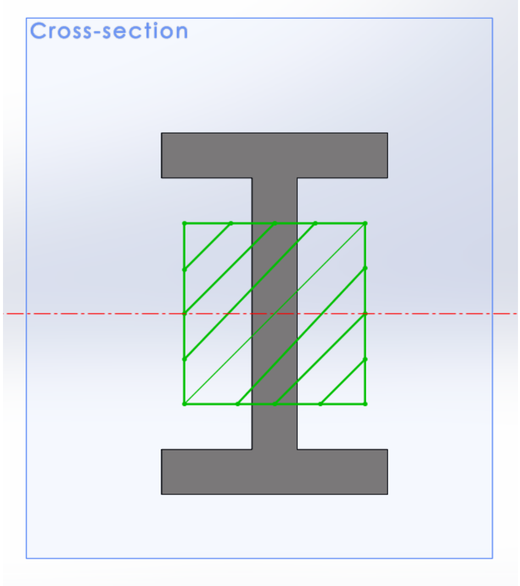
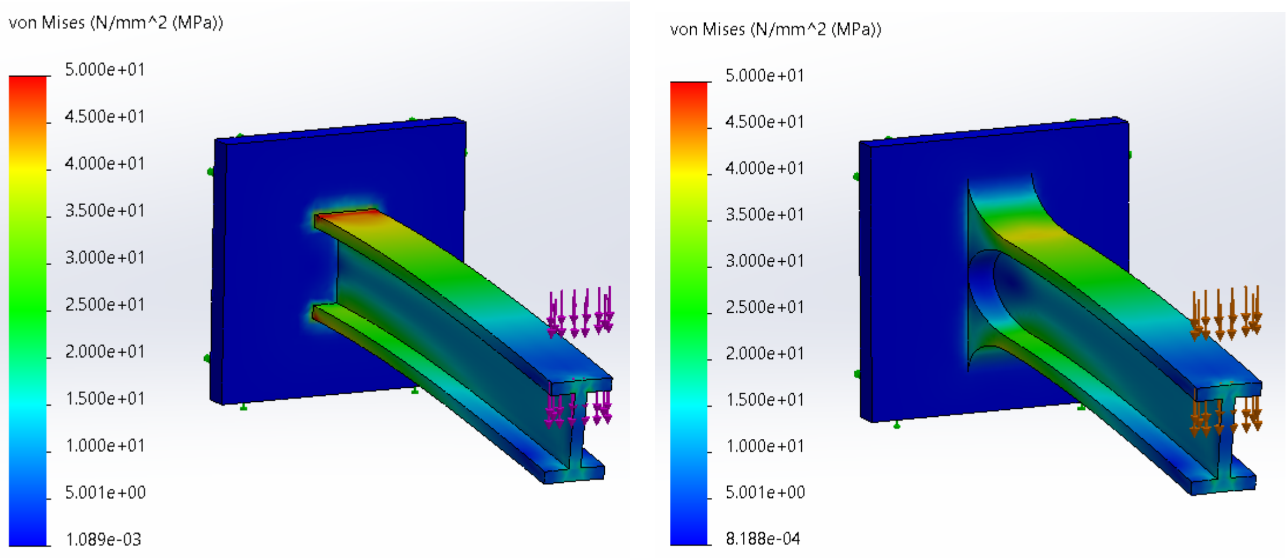


Figure 8.2: Cross section of the I-beam and square beam superimposed. I-beam area in grey and square beam area in green, with the red axis being the axis around which bending occurs.

highest stress is at the right angle between the beam and the wall. To combat this, the presence of right angles in the structure should be minimised, and replaced by smooth corners and edges. This way of changing the structure is known as a ”fillet” in mechanical design.

8.2. Requirements

In this section, the requirements for the structural subsystem are stated. The process of obtaining these requirements involves identifying all the system requirements which influence the structures. Examples of such system requirements have stipulations on characteristics of the material used on the drone, or the weight constraints on the system as a whole. The subsystem requirements can be found in Table 8.1.



(a) Stresses of a normal I-beam.

(b) Stresses of a filleted I-beam.

Figure 8.3: Generic pictures of stress concentrations in I-beam bending tests, it can be seen that rounding the root of the i-beam (known as filleting) can successfully lower the maximum stress present in an object.

Table 8.1: Structure subsystem requirements.

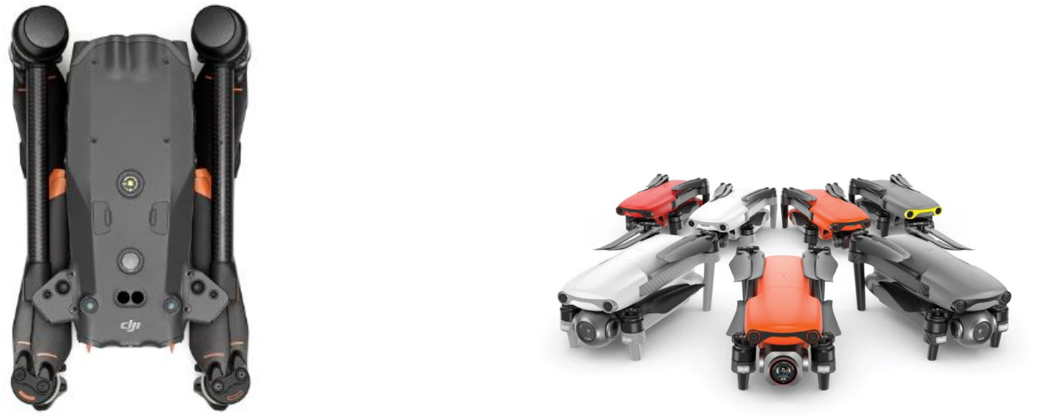
Identifier	Requirement	Related System Requirement
SUB-STR-01	The material datasheet of the material(s) used shall indicate the material is not harmful or toxic.	SYS-SUS-02
SUB-STR-03	1 “CHIP” drone shall be able to fit within a box with a dimension of 103x67.5x37.5cm	SYS-SIZ-01
SUB-STR-04	3 “DALE” drone shall be able to fit within a box with a dimension of 103x67.5x37.5 cm	SYS-SIZ-01
SUB-STR-05	The structural weight of “CHIP” drone shall not exceed 1,050 g.	SYS-SIZ-02
SUB-STR-06	The structural weight of “DALE” drone shall not exceed 400 g.	SYS-SIZ-02
SUB-STR-07	The maximum weight of “CHIP” drone shall not exceed 12 kg.	SYS-SIZ-02
SUB-STR-08	The maximum weight of “DALE” drone shall not exceed 4 kg.	SYS-SIZ-02
SUB-STR-09	At least 70% of the structure by mass shall use materials that are recyclable.	SYS-SUS-01
SUB-STR-10	During 3D printing of the structure, the mass of support material shall not exceed 20% of the total filament mass used to print the structural parts.	SYS-SUS-05
SUB-STR-11	A repairing course of action shall be stipulated for at least 80% of the parts by count.	SYS-SUS-07
SUB-STR-12	Design of the structure in both drones shall not inhibit the removal and attaching of sensor modules.	SYS-PER-05
SUB-STR-13	Each drone shall have a guard around the propellers.	SYS-SAF-03

8.3. Design Considerations

This section will discuss the broader considerations which will need to be heeded when designing the structures. Naturally, the subsystem requirements will have a constraining effect on the design, but there may be other aspects one can look beyond the prescribed requirements to see what other aspects may influence the design. For instance, how exactly should sustainability be accounted for in the design? The application of sustainability can often be very subjective and hard to define in requirements, so designers should have an understanding of what may contribute to a more sustainable design before putting anything on paper. It is thus useful for this chapter to discuss the more abstract design considerations, so that it may give more context to the final design.

8.3.1. Design Constraints

Firstly, some considerations can be directed to the most driving requirements. Referring back to Section 8.2, the more limiting constraints here are the requirements placed on drone size and weight, which would drive the design of the structure more than other requirements. Requirements SUB-STR-03 and SUB-STR-04 stipulate that the drones shall fit inside a suitcase with the specified dimensions of 103x67.5x37.5cm. Following the example of commercially available drones, as seen in Figure 8.4, it would be wise to design the drones to be foldable, or at least detachable. Which will allow their footprint in storage to be smaller than when they are flying. As for the weight constraints, SUB-STR-05 and SUB-STR-06 will place a hard weight limit on how heavy the drone frames should be and will dictate a large part of the design process. While SUB-STR-07 and SUB-STR-08 will require constant communication between the structural design team and the various other subsystem teams to ensure that the combined weight of the frame, payload and other equipment does not exceed these weight limits.



(a) A DJI M30 drone folded [2]

(b) Some evo nano drones folded [3]

Figure 8.4: Commercially available drones with foldable arms.

8.3.2. Material

Another general point to be considered is the material choice. When selecting a material, there are two main considerations which need to be taken into account: the sustainability of the material, and the mechanical and physical properties.

In common language, sustainability is often a buzzword which is thrown around with little concrete definition behind it. In regards to materials, sustainability mainly means whether recycling processes for the material exist, how complicated those processes might be, or if the material itself could utilise some recycled material. Moreover, it will also mean if the material could possibly be biodegraded, which will allow the drone frame to pose minimal danger to the environment at its EOL.

The material's mechanical and physical properties are more definite. One of the more important aspect to consider here is what manufacturing processes the material allow. For example, if the material is suitable for additive manufacturing, a high degree of design freedom is allowed. Moreover, to give even more freedom to how the structure can be designed, using metals would be less preferred. Since there can be complications with hot tearing when metal parts cool, or with residual stresses and warping, downsides are further exacerbated when you try to produce a part that has thin walls with complex geometries. Thus for this detailed design phase, metals will not be considered. Aside from manufacturing, there are also performance properties of interest which influence the structural performance of the design, these properties are:

- Density ρ, the amount of mass per volume of the material, in $\mathrm{kg\,m^{-3}}$. Ideally this value would be as low as possible.
- Young's modulus E, modulus of elasticity of the material loaded in tension or compression, in MPa. Ideally this would be as high as possible.

[2] retrieved from: `https://influentialdrones.com/products/dji-matrice-m30-enterprise` (cited 10 June 2022)
[3] retrieved from: `https://www.autelrobotics.com/productdetail/24.html#jsgg` (cited 10 June 2022)

- Yield strength σ_y, the internal stress at which the material yields, the maximum von Mises stress which is present in the structure should be below this value by a safety factor, in MPa. Ideally the material would have a high yield strength.

8.3.3. Sustainability

While sustainability has been partly addressed in the recyclability of the material, there are still general considerations that can be made to make the drone sustainable. This can come in the form of designing the frame to be robust, which can be achieved by including large safety factors or using durable materials. Achieving this will limit the need for repairs or replacement, thus requiring less resources over the lifetime of the drone.

Conversely, in line with the direction which the European Commission is moving towards[4]: sustainability can also come in emphasising repair-ability of the drone. This would mean ensuring that parts of the drone can be replaced when they fail, instead of discarding the entire drone all at once. One good way of realising repair-ability is to separate the drone into distinct components, which can be assembled together with relative ease. Off-the-shelf components can also be used to as large an extent as possible, which will allow users to easily retrieve repair parts. For example, the flight battery could use commercially available models instead of a proprietary design. All these measures also have the added benefit of allowing the user to extend the lifetime of the drone as long as possible, reducing the need to fully replace a drone system and thus further increasing sustainability.

8.3.4. Trinity Concept

The design of a physical product is broadly made up of three distinct aspects: Physical design, Material choices, and the Manufacturing process. These three facets are tightly interlinked, and decisions made in one aspect need to carefully account for decisions made in other aspects.

The trinity concept can be extremely useful for guidance during a design process, it highlights all the dependencies that exist within a product. For instance, the choice of material needs to account for what kind of manufacturing processes are possible, and what geometries these processes allow for. Thus to produce an effective design for the drone concepts, decisions made in this chapter will bear in mind the trinity concept of interdependence between design, material, and manufacturing.

8.4. Design Choices

Some aspects of the designs can or should be decided before any work can commence, one of the more obvious ones is choosing a material. This section will cover the choices or decisions that were made prior to commencing the design process.

8.4.1. Material Choice

Choosing a material for a product is an intricate and complicated process, and will be done in a strategic manner, as represented in [4]. In essence, the needs for the drone's skeleton architecture will be described. Later, the metric to be maximised (i.e. Young's Modulus to density ratio) will then be depicted on a graph, and candidate materials will be selected from available sources. Following that, a final evaluation will be conducted to determine the best material for this design.

Function:	**Skeleton**	**Objectives:**	*Minimal Mass*
Constraints:	*Recyclable*		*Minimal Bending*
	Foldable / Detachable	Free Variables:	*Material*
	Non Toxic		*Thickness*
	Minimal Waste		
	Length		

The list above represents the general outlook on the structural constraints to be adhered to, the objectives of the structure, and finally the variables that are set to be free to change in order to obtain the optimum out-

[4] Retrieved from: `https://www.europarl.europa.eu/RegData/etudes/BRIE/2022/698869/EPRS_BRI(2022)698869_EN.pdf` (cited June 2022)

come. In a nutshell, The list defines the skeleton structure's restrictions, which are inferred from the subsystem requirements. The table, on the other hand, specifies the design's goals. This include reducing the skeletal structure's mass, and minimising bending. Although the latter goal is not explicitly stated in the requirements list, it is among the most essential design characteristics. High deformation on the frame will affect the inspection quality, but it will also cause spinning propellers to collide, which could result in a crash. As a final remark, the goal of this chapter is to define the optimal solution for the structure by altering the free variables discussed in the list above in order to keep up with the constraints.

Figure 8.5 describes each material family and plots its' Young's Modulus against Density. Beginning the material choice journey, one shall eliminate the unfeasible material groups immediately. For example, the Foams family has a relatively low Young's Modulus but a low density. This would yield a very thick and bulky design so the Foams family would not be a feasible material choice. Similar traits can be found in Elastomers, which also have a very low modulus. Another example of an impractical material group is Ceramics. Although they promise high material properties, they are very brittle. This drawback is not acceptable, especially on an aircraft.

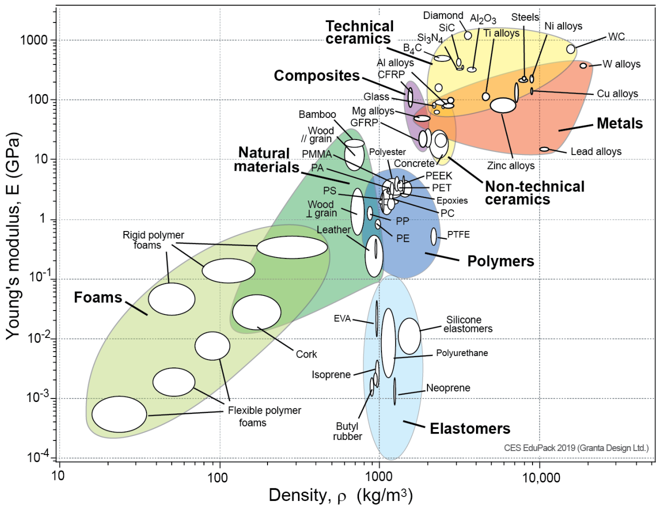


Figure 8.5: Ashby plot showing different material families plotted against their physical properties; Young's modulus and density.

The remaining material groups, such as metals, polymers, and composites, would be promising candidates for a drone structure. Although previous families were eliminated due to their material properties, another trinity concept to be considered is the manufacturing method to be employed. When requirement SYS-SUS-05 is considered, it would be the wisest decision to employ additive manufacturing in the production of the skeletal structure. Where this requirement dictates the need of cutting back waste material in production. That being said, employing material from the Metals or Natural Materials family would not be feasible anymore since they rely heavily on subtractive manufacturing methods, in addition to the reasons stated in Subsection 8.3.2. Furthermore, technologies that integrate additive manufacturing with metals (e.g. Powder Bed Fusion) appeared to be impracticable, owing to their high tendency to leave behind residual stresses in the object, resulting in warping behaviour on thin structures. Moreover, additive manufacturing is widely used in Composites and Polymers. Therefore, they would be an ideal candidate for selection.

In order to have a a light structure that has a high bending stiffness the approach would be to maximising the following material Figure 8.4.1 property according to [4].

$$M = \frac{E^{1/2}}{\rho}$$

The material dictionary may be generated below in Table 8.2 by examining potential material groups and the requirement for optimum stiffness to weight ratios using the above formula. The table shows the proposed material from the Composites or Polymers category, as well as its recycling option, density, Young's Modulus, and the M parameter that was previously established. Furthermore, this table only includes materials which can be additively manufactured and fit within the previously mentioned categories.

Table 8.2: Candidate materials and their material properties.

Name	Recyclable	Density ($\mathrm{kg\,m^{-3}}$)	Young's Modulus (GPa)	M
PAHT-CF15	No	1,250	8.386	73.26
PP-GF30	Yes	1,140	6.000	67.95
PA66-GF30	No	1,335	6.750	61.54
PLA	Yes	1,240	4.107	51.68
PHA	Yes	1,250	3.300	45.96
ABS-PC	Yes	1,070	2.400	45.78
PMMA	Yes	1,190	2.770	44.23
PET-G	No	1,230	2.950	44.16
ABS	Yes	1,020	2.000	43.84
PEI	Yes	1,270	3.100	43.84
PP	Yes	905	1.325	40.22
POM	Yes	1,390	2.600	36.68
PF	Yes	1,400	2.409	35.06
PA	No	1,130	1.000	27.98
PBS	Yes	1,072	0.065	7.52

In terms of parameter M, it is evident that PAHT-CF15 performs best. Despite its high material property, it cannot be recycled, making it unsuitable for a sustainable solution. The following sustainable materials are Glass Fibre Reinforced Polypropylene (PP-GF30) and Polylactic Acid (PLA) respectively. Despite its widespread use, PLA has a low glass transition temperature REF, which violates requirement SYS-REL-02. As a result, choosing PP-GF30, which is commercially accessible, and easily recyclable due to its thirty per cent glass fibre content is the safest alternative for this trade-off.

PP-GF30 stands for 30% Glass Fibre Reinforced Polypropylene. Despite the fact that literature study [27] has demonstrated that the Elastic modulus of Polypropylene has a positive link with the quantity of Glass Fibre in the material, commercially available filaments are restricted to 30%. As a result, the appropriate and realistic quantity for the drone's skeleton structure was determined to be 30%. Contrary to PLA, PP-GF30 has a much higher heat distortion temperature which guarantees a safer operational lifetime compared to its closest competitor.

8.4.2. Structural Features

Choosing the structural material is just one of the many challenges when it comes to a drone design. One shall get the best out of material by smartly exploiting its advantages. A combination of a stiff material such as PP-GF30 and additive manufacturing technology promises the creation of complex and strong structures.

This possibility will be employed on both drones and they will have slender truss-like structures where possible. This would cause a significant weight reduction while maintaining the structural integrity but more importantly its stiffness.

On the other hand, in order to comply with the requirement SYS-SIZ-01 but more importantly SUB-STR-02 where they state the necessity to fit a limited space and the need for foldable arms of the drones respectively, the arms of each drone will be both detachable and foldable. Which will give the operator some flexibility when it comes to storage and transport.

Lastly, another feature that both drone types will employ is the fact that they will have safety fairings around their propellers. As stipulated by SUBS-STR-13. These fairings will be an additional safety measure to prevent any bad results in case of any impact between a plane and drone.

8.5. Drone Designs

This section will go into the design and configuration of both large and small drones, the "CHIP" drone and "DALE" drone, in detail. Although the positioning of components and their rendering pictures are discussed in earlier sections, the remaining aspects of the drones, such as their overall dimension, weight, and payload, will be discussed in this section.

8.5.1. Dimensions

Despite having no strictly defined sizing constraint for either of the drone types, the requirements SYS-SIZ-01 and SUB-STR-02 which deal with the size of the suitcase that drones will be stored have a great impact on the size of the drone. Therefore, the initial diagonal length estimate has been made to be 100cm which was later adopted as the design choice in cooperation with the requirement SUB-STR-02 which dictates the necessity of having foldable or detachable arms for both drone types. A small drone on the other hand has a frame size of 55cm when measured diagonally. When the size of the fairings is considered, where this safety feature is deemed necessary by the requirement SYS-SAF-03, the big drone spans 177cm whereas the small one stands at exactly 82cm.

Thanks to its relatively small size and its configuration, the “DALE” drones can be easily folded and stored in triplet configuration within a single suitcase. as shown in Figure 8.6. The “CHIP” drone offers high flexibility despite its big size. The hinges between arms and the core structure have a hexagonal pin which enables both detachment and also foldability to different angles. These drones will first be disassembled before storing them in the suitcase. A demonstration of the drones stored in their suitcases can be seen in Figure 8.6.

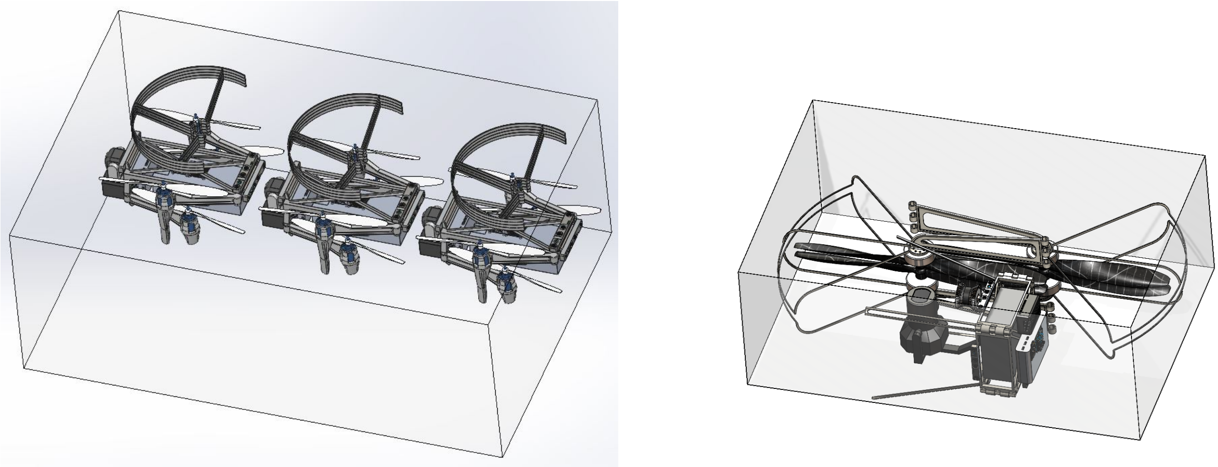


Figure 8.6: “DALE” drones folded and placed in the suitcase volume prescribed in SUB-STR-04.

8.5.2. Structures Mass Budget

The initial mass budget of the skeletal structure on the other hand was assumed to be 1,050g and 400 g for “CHIP” drone and “DALE” drone respectively (requirement SUB-STR-05). Furthermore, initial skeletal design of the big drone with the configured material PP-GF30, was exactly 803 g as given in picture Figure 8.7. Although requirement SUB-STR-05 was met on the first complete design of the “CHIP” drones’ skeleton, the modifications on the design have not stopped there since cutting back every single gram of dead weight on an aircraft would come back as improved flight performance. The modifications and the final design parameters will follow on the.

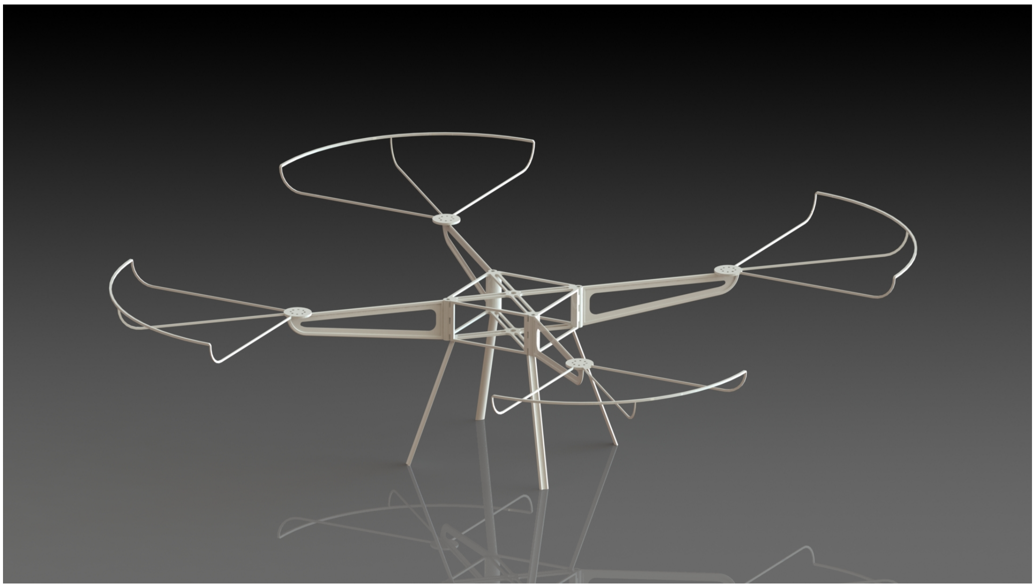


Figure 8.7: Structure of the “CHIP” drone before topology optimisation.

On the other hand, the first complete design of the “DALE” drone had a skeletal mass of $456\,\mathrm{g}$, which was in a clear violation of the SUB-STR-05 criteria. On the contrary to the “CHIP” drone, the modification was more of a necessity for the “DALE” drone than an option at that point. The design iteration followed the current design at that time (illustrated in Figure 8.8) to cut down the overall mass of the structure. Yet, the modifications done after that point is a subject of the Subsection 8.5.6.

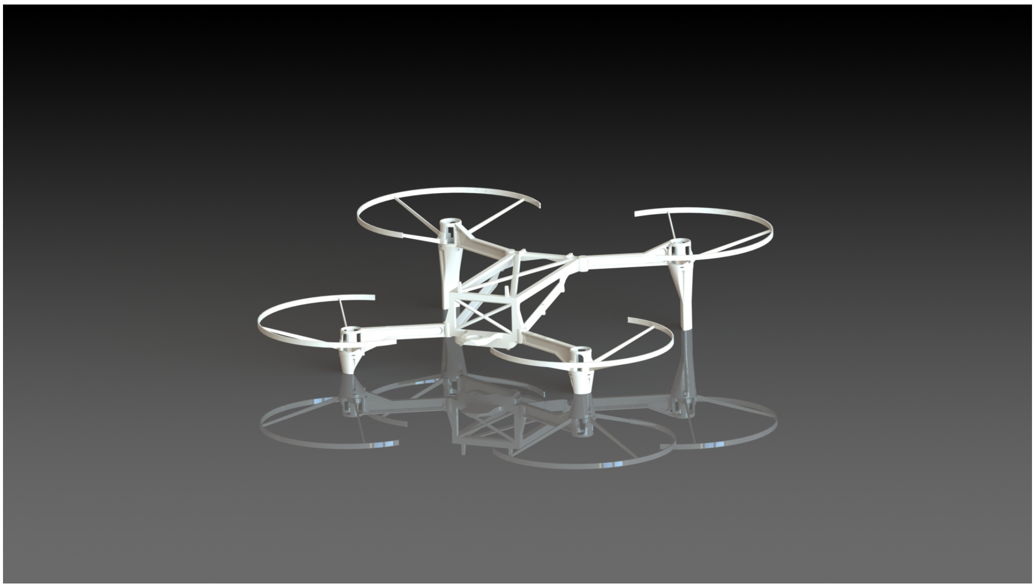


Figure 8.8: Structure of the “DALE” drone before topology optimisation.

8.5.3. Structural Behaviour

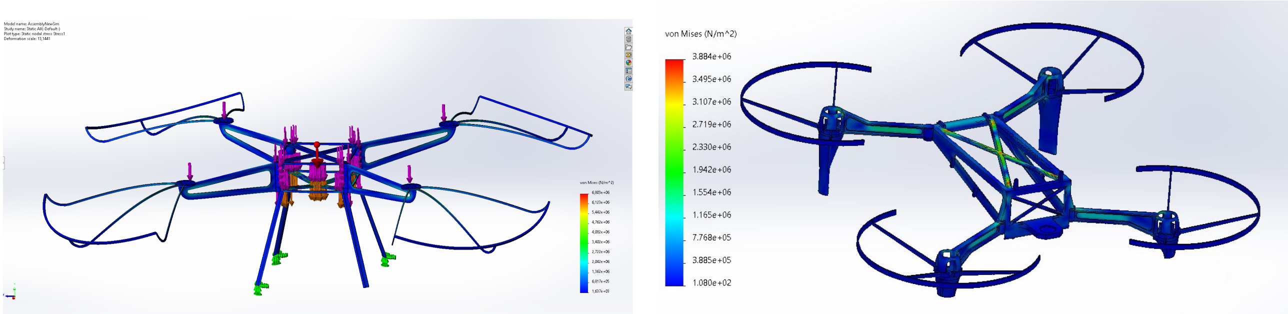
The behaviour of drone structures is divided into two scenarios in this subsection. The first is Nominal Condition, which simulates the drone being on the ground with all rotors idle. The second mode is the Extreme Condition, which describes the momentary acceleration of the drone vertically. These two cases represent the maximum possible deflection and stress that drones shall withstand in each direction. Additionally, the natural or modal frequencies of the structure will be obtained and compared to the induced vibrations from sources such as the motors.

Nominal Condition: fixed boundary conditions at feet, 1g gravity applied.

As was described earlier, the Nominal Condition assumes drones to be standstill. Therefore this would simulate the highest bending moment and stress over the negative Y direction (upwards defined to be positive).

The weight of the structure and onboard components are assumed to be the only forces on this structure. Meanwhile, the drone legs or the contact points are assumed to be fixated to the ground.

For the “CHIP” drones’ case, the highest local stress recorded by the model was $6.8\,\mathrm{MPa}$ and it was recorded at the centre of the core component and the cutouts of the arms, as seen in Figure 8.9a. On the other hand, a maximum stress recorded on “DALE” drone was at $3.9\,\mathrm{MPa}$ magnitude acting on similar locations as on the “CHIP” drone, as seen in Figure 8.9b.

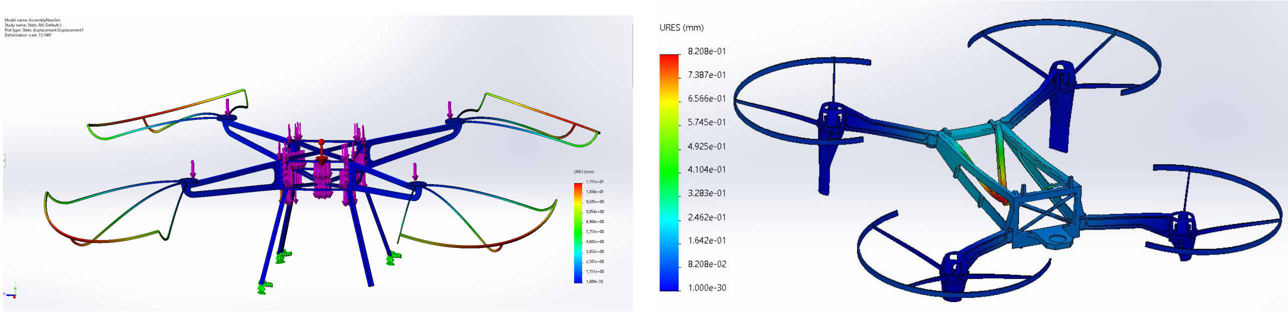


(a) Stress diagram on the “CHIP” drone under nominal conditions.

(b) Stress diagram on the “DALE” drone under nominal conditions.

Figure 8.9: Stress diagram under nominal condition.

Displacement on the other hand, has been highest at the fairings of the big drone and at the centre of the small drone with magnitudes of $1.2\,\mathrm{cm}$ and $0.8\,\mathrm{cm}$ respectively, seen in Figure 8.10. One important remark would be pointing out is that the deflections represented Figure 8.10 are extremely exaggerated for visualisation purposes.



(a) Deflections diagram on the “CHIP” drone under nominal conditions.

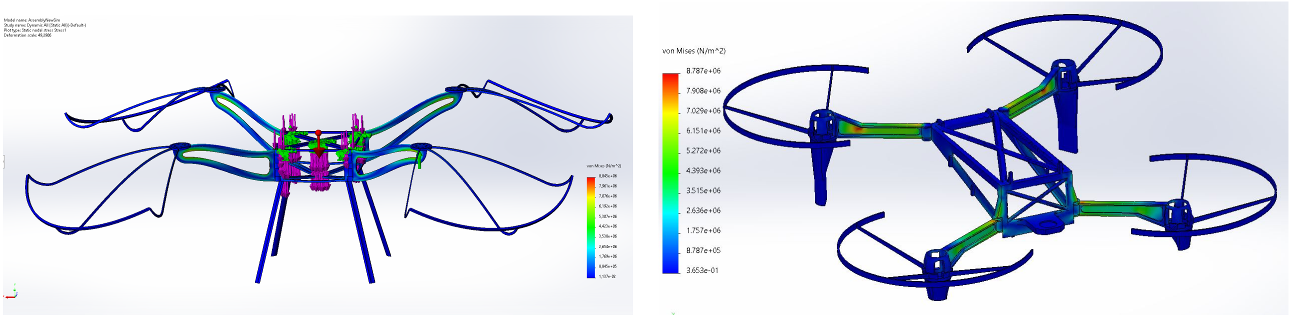
(b) Deflections diagram on the “DALE” drone under nominal conditions.

Figure 8.10: Deflection diagram under nominal condition.

Extreme Condition: fixed boundary conditions at core skeleton, thrust of 2x drone weight applied at motors.

Extreme condition, on the other hand, is a mode for simulating the highest loading on both drones and analysing it following that. The way that this mode works is, that the drone is assumed to be accelerating in a positive Y direction with the highest available thrust from its propellers (which is twice the weight) and the core of the drone is assumed to be fixed. Even though this is a momentary acceleration of an extreme scenario that hardly lasts for a second, the possibility of this happening is not zero and therefore it must be taken into account.

Although the stress levels in this study are remarkably higher, as shown in Figure 8.11, the locations however are identical. The highest stress level recorded on drone arms are both $8.8\,\mathrm{MPa}$ on each drone. This matching stress property was intentional which will be further explained in the next subsection.

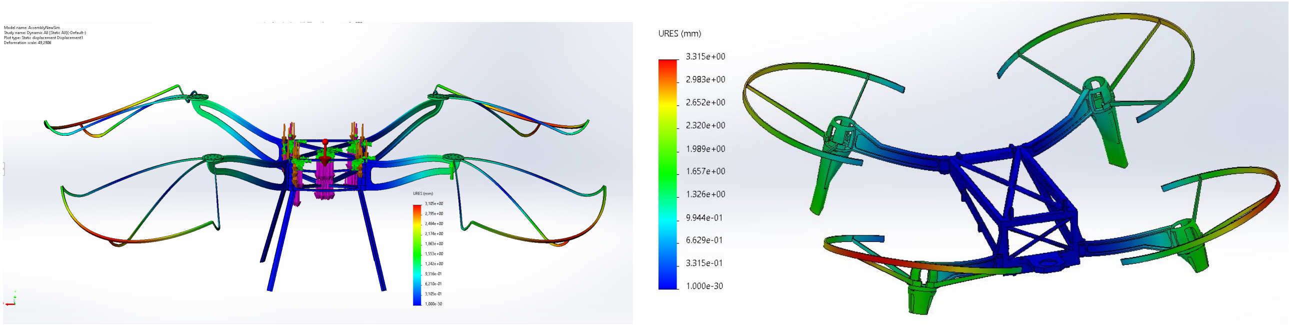


(a) Stress diagram on the “CHIP” drone under extreme conditions.

(b) Stress diagram on the “DALE” drone under extreme conditions.

Figure 8.11: Stress diagrams under extreme condition.

Figure 8.12 shows the deflection results of the two drones under extreme conditions. The highest bending can be noted on both drone arms, which is unsurprising. By displaying very minor deflection at the end, the structure was able to endure a fictitious propeller load. With a magnitude of $3.1\,\mathrm{mm}$, the maximum deflection on “CHIP” drone drone may be noticed at fairing ends. The deflection of the propeller dock is a significant component here. The upward deflection of all drone arms increases the likelihood of propeller paths colliding, which might prove fatal to the drone. As a result, the 27-inch propeller was replaced with a 26-inch propeller for maximised clearance, which was a design option at the time. As a final note, the highest deflection can be seen on fairings of the “DALE” drone with a similar magnitude of $3.3\,\mathrm{mm}$. The fact that the highest deflection on “DALE” drone has an upwards deflection is due to its different structure shape but more importantly its shorter length.



(a) Deflections diagram on the “CHIP” drone under extreme conditions.

(b) Deflections diagram on the “DALE” drone under extreme conditions.

Figure 8.12: Deflection diagrams under extreme condition.

Modal Frequencies

Having displayed the stress and displacement characteristics, the modal or natural frequencies of the drones will now be covered. Using the frequency analysis simulation in SolidWorks, it was possible to obtain all the modal frequencies of the “CHIP” drone and “DALE” drone structures. The modal frequencies of both drones from the simulation are compiled in Table 8.3.

Table 8.3: Modal frequencies of the two drones.

	“CHIP” drone	“DALE” drone
1st mode $[\mathrm{Hz}]$	2.13	15.83
2nd mode $[\mathrm{Hz}]$	2.13	15.86
3rd mode $[\mathrm{Hz}]$	2.13	16.20
4th mode $[\mathrm{Hz}]$	2.14	16.30
5th mode $[\mathrm{Hz}]$	5.23	20.20

To verify that resonance will not occur, the frequency of induced vibrations shall not match any of the modal frequencies. The primary source of vibrations on board a drone are the motors. A rough estimation of motor vibrations could be obtained from literature, which shows that with motors spinning at 5,160 RPM, the vibrations exhibited have peaks at frequencies beyond $100\,\mathrm{Hz}$. Thus since the motors on both drones spin at even higher frequencies, it could be assumed that the vibrations experienced will have peaks at even higher frequencies. So as both drones exhibit modal frequencies at much lower values, they would be safe from resonance.

8.5.4. Additional Remarks

Both drone concepts have passed their rigidity and strength tests on different modes. A good question here would be asking how well did drones passed these tests? Or could we have a better design than this? Unfortunately, the answer to these questions is not straightforward.

Firstly, it is crucial to mention that additive manufacturing is a method that which its products are print direction sensitive especially with regard to its mechanical properties. Moreover, the same material may behave almost five times better if the print direction is flipped by 90%. To account for this variance in behaviour, the drone is assumed to be assembled with parts that are printed in their optimal direction however when assessing the model, the safety factor of 2 and the worst possible print direction has been assumed.

To be more specific, the flexural strength of PP-GF30 is given to be around $95\,\mathrm{MPa}$ according to ULTRAFUSE if printed in XZ direction as can be seen in Figure 8.13, however, this value plunges to $19\,\mathrm{MPa}$ when ZX direction is employed. This is a common problem of additive manufacturing and therefore has to be accounted for in every additively manufacturable material.

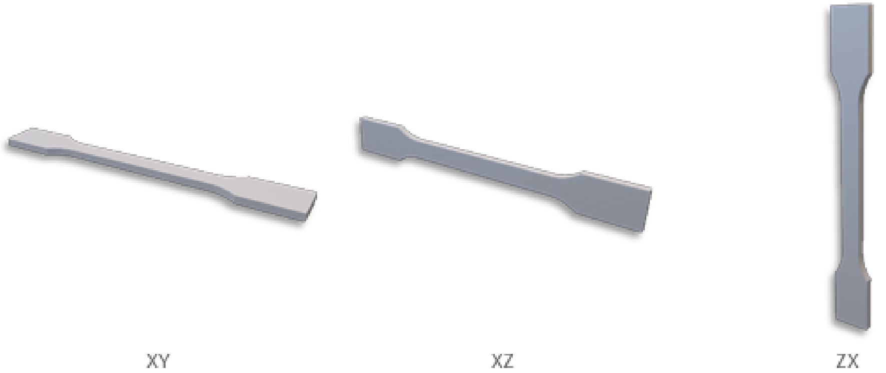


Figure 8.13: Different orientations that a part can be 3D printed in.

To sum up, the simulation assumes that all parts are printed in the XZ direction and uses $95\,\mathrm{MPa}$ as flexural Strength, and later on the design modifications take place to make sure that the design stress is at most the half of the flexural strength of ZX direction to achieve **safety factor of 2** in both designs.

The reason behind using flexural strength instead of any other parameter is solely based on the drones having their highest stress due to bending moment caused by propellers.

8.5.5. Component Placements

The “CHIP” drone will employ two truss-like layers of the core structure. The bottom of the first layer will host the LIDAR and inspection camera by giving both payloads enough space for viewing the environment and taking images in the nadir direction. The middle compartment on the other hand will only host the heaviest component of the drone - the battery - by allowing enough space for attachment and cooling. In the final section, the second layer’s top will host the electronics such as the flight controller, power distribution board, computer and USB hub by giving sufficient flexibility to the inspection operator in terms of accessibility. The stereo cameras will be attached to each side of the drone as well as the top and bottom sections of it to allow the highest range of coverage. Where electronic speed controllers will be housed at the root of each drone arm.

The “DALE” drone will use same placement ideology as the “CHIP” drone when stereo cameras are concerned. In contrast, not only “DALE” drone will have only one layer of the core structure, it will host the battery, computer, flight controller and ESC board under the prime layer. The inspection camera of the “DALE” drone is placed in-front of the drone, with no obstructions above the camera allowing it to gimbal up to 90 degrees above horizontal, and gimbal down to 30 degrees.

8.5.6. Topology Optimisation

Topology optimisation (TO) is a computational approach for improving the performance of a system by optimising material arrangement inside a given design area, for a particular set of forces, and boundary conditions. The idea behind this method, is either to maximise stiffness to mass ratio or to solely achieve a reduction in mass.

For the big drone, three different topology optimisation studies were applied on core, arms and fairings. The arms and fairings had the first aforementioned approach and core part of the skeleton had only the mass reduction requirement. This is because the deflection at arms were critical however the deflection on core structure was acceptable to an extend. Figure 8.14 give the rough idea of which sections of structure need to be removed in blue and the remaining yellow parts shall remain in the structure according to TO study.

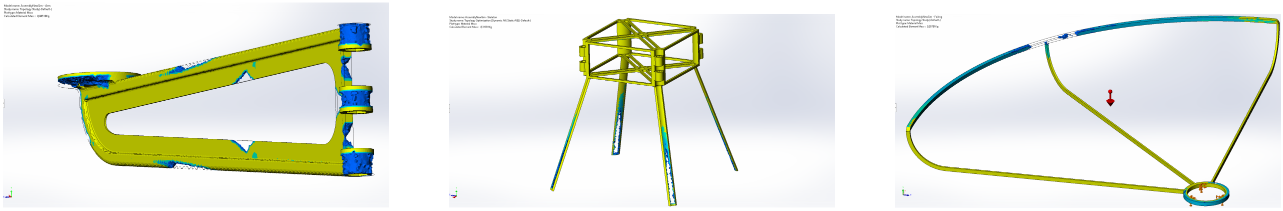


Figure 8.14: Topology optimisation results for components of “CHIP” drone, blue areas indicate material that can be removed.

According to the TO results, a thinner fairing edge, hinge connections, motor dock, and core legs would be a suggested weight reduction proposals for the “CHIP” drone while retaining stiffness. After implementing these suggestions, not only a considerable weight reduction takes place on the design, the local stress relief due to lighter structure and easier drone control due to lower moment of inertia globally. The original design shown in Figure 8.7 had a total structural mass of $803\,\mathrm{g}$ where after the optimisation this mass has diminished to only $624\,\mathrm{g}$ in total.

The topology optimisation has also taken place on “DALE” drone with the similar constraints and same goals. The difference of TO on “DALE” drone than on “CHIP” drone was that the weight reduction on “DALE” drone was a must instead of an preference.

Weight loss was made feasible with the aid of this technique, and a weight criterion that had previously been unmet was finally accomplished, as shown in Figure 8.15. The drone’s weight dropped from $456\,\mathrm{g}$ to $400\,\mathrm{g}$.

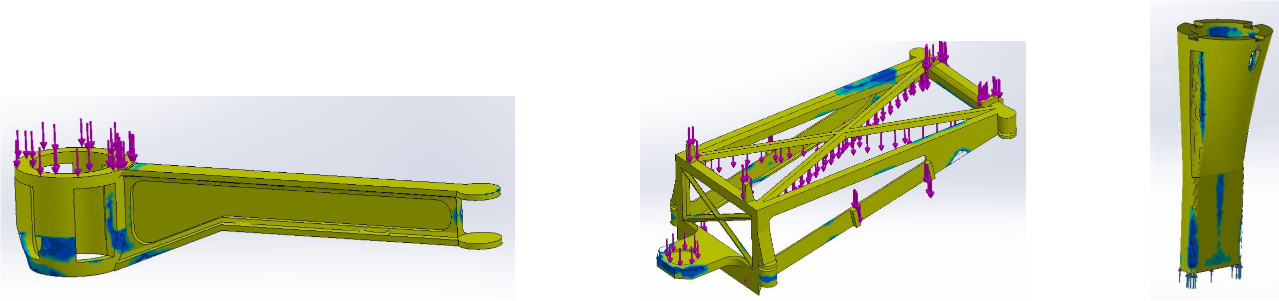


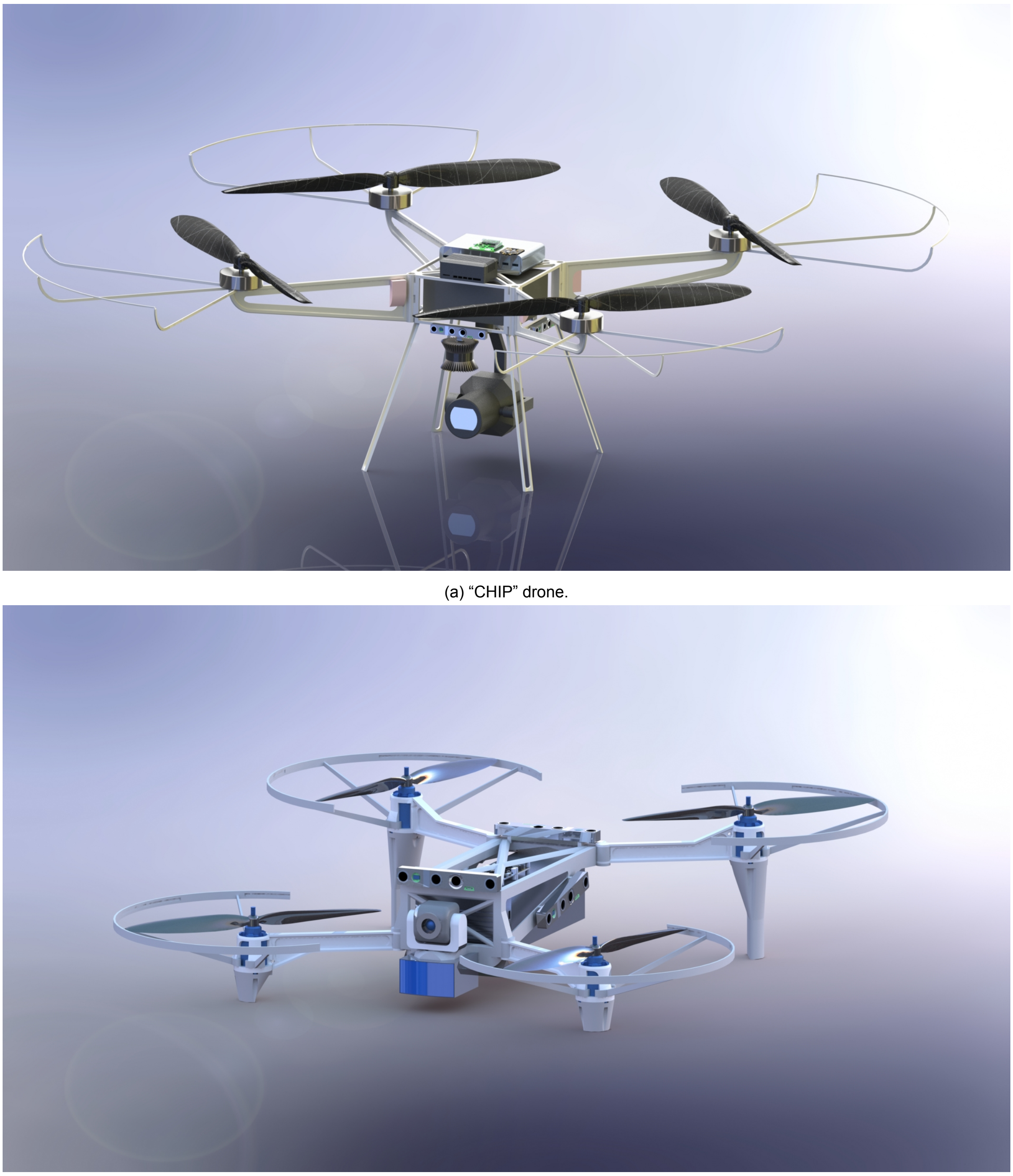
Figure 8.15: Topology optimisation results for components of “DALE” drone, blue areas indicate material that can be removed.

8.5.7. Final Specifications

In summary, this DSE group presents the final specifications of “CHIP” drone and “DALE” drone in Table 8.4, and their respective renders can be seen in Figure 8.16. 2 of the suitcases will hold 1 “CHIP” drone weighing 7.8 kg each, and 1 suitcase will hold 3 “DALE” drones weighing 9.75 kg.

Table 8.4: Final specifications of CHIP and DALE.

	“CHIP” drone	“DALE” drone
Drone Frame Weight [g]	624	400
Total Weight [kg]	8.16	3.74
Dimensions Unfolded (LxWxH) [cm]	140x140x46	65x67x15
Propeller Diameter [cm]	66	28
Maximum Stress [MPa]	8.8	8.8
Design Safety Factor	2	2



(a) “CHIP” drone.

(b) “DALE” drone.

Figure 8.16: Renders of the final drone designs.

8.6. Operational and End of Life Considerations

After the drones have been designed, the needs during operations and possible EOL options will be explored. Thus, the purpose of this section is to look over what needs to be considered under the operational life of the drones, including some possible actions to be taken to repair the drone structure. As well as to explain how the EOL stage of the structure would look like, such as possible life extension options or recycling options.

8.6.1. Operations and Repairs

During the operational life of the drone, no special maintenance would need to be directed at the drone structure. Since plastics are notoriously sturdy and are very tough to degrade, the frame will be able to withstand a wide range of storage conditions. The only event to care for would be when the drone has been damaged, most likely due to high speed collisions.

When the drone comes under forceful impact, a likely outcome would be the deformation of the parts. In which case, depending on how severely deformed the affected part is, the drone may or may not fly. Thus, the part may need to be replaced by a spare. To enable this swap, the drones have been designed for easy disassembly, with the drones being made up of detachable parts. For instance, the “DALE” drone propeller arm can be detached into the fairing, the arm itself, and the legs.

Moreover, all the carried payload and components can be easily dismantled and swapped. The frames of the drones have been especially designed to not obstruct the access to any of the electronics or sensors, with the structure made up of trusses. And to swap any component out would require simple unscrewing or unclipping of the part.

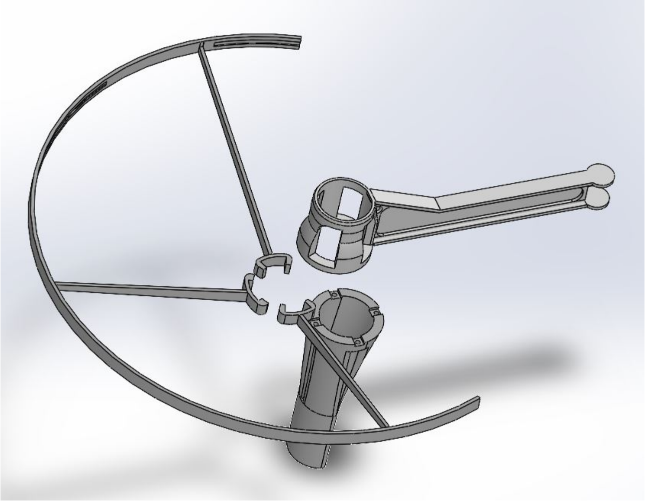


Figure 8.17: Exploded view of the “DALE” drone propeller arm, showing the various parts that it consists of.

8.6.2. End of Life Options

At the drones’ EOL, the structures may be disposed of at a recycling centre. As mentioned in Subsection 8.3.2, the material choice made consciously kept in mind that whatever the material may be, it should be possible to recycle the material. PP-GF30 is a plastic composite which can be recycled, as aforementioned, and as shown in various sources [29] [2]. Thus, a sustainable EOL decision would be to send the parts to be recycled.

However, depending on the circumstance leading up to decommission, it may be wise to store the drones instead of disposing of them. So as to anticipate future needs to use the drones. Or even better, the drone frames can be repurposed to perform tasks other than inspection of aircraft. Examples of potential alternate uses could be aerial photography, or inspection of structures other than aircraft. Both examples would require little to no modifications be made to the drone payload, and at the same time can reduce demand for new purpose built drones to be manufactured.

8.7. Subsystem Verification

Finally, the design choices made in both drones shall now be verified to see if they comply with subsystem requirements. Verification for this is relatively straight forward compared to other subsystem. Since SolidWorks is a validated software for simulations[5], all simulation data gathered can be interpreted readily. Moreover, some verification work has already been carried out earlier in the chapter. Where the maximum stresses of the structures were analysed, and a modal vibration analysis was performed. Thus, the only verification remaining for the design is its' compliance with system and subsystem requirements. To this end, every system and subsystem requirements pertaining to the structures subsystem were reviewed. All the requirements were deemed to have been satisfied, aside from SUB-STR-11 and SUB-STR-10. SUB-STR-11 is deemed to be dubiously complied with. While some repairing actions were outlined in Subsection 8.6.1, there's quite some uncertainty when it comes to the exact nature of damage that can occur, and thus uncertainty in the exact repairs that need to take place. Thus this requirement is only counted as partly complied. While SUB-STR-10 was reviewed to be not satisfied, as “CHIP” drone spends 50% of printed plastic to create supports, while “DALE” drone spends 40 % of material for supports.

[5] Retrieved from: `https://www.solidworks.com/sites/default/files/2019-04/VPCS-English2019.pdf` (cited June 2022)

9
Electrical Subsystem

The inspection drones consist of a variety of electrical and electronic components in order to perform their tasks successfully, including stereo cameras, a Flight Controller (FC), and inspection cameras. Each of these electrical components requires a certain input operating voltage and draws a specific current from the battery, resulting in battery power being consumed by the component. Next to that, information has to be exchanged between the different electronic components, e.g. the FC and the IMU. In this chapter, the design for the drones is provided from an electrical point of view. In Section 9.1, background information that was used during the design process can be found. After that, in Section 9.2, the requirements for the electrical subsystem are stated. Next, the electrical design is outlined in Section 9.3. Design considerations, and a discussion on analysis, can be found in Section 9.4. Lastly, final remarks are made in Section 9.5.

9.1. Background Information

In this section, relevant background information is given on electronics that have been used during the design process of the subsystem. First, the power supply and its efficiency are discussed. Next, practical implementations for voltage regulations are considered. This is followed by a short discussion on controlling a brushless motor. Lastly, an overview is given on relevant communication protocols for exchanging data between components.

Electrical power can be modelled by using two measures: voltage and current. More specifically, electrical power can be calculated using Equation 9.1.

$$P = V \cdot I \tag{9.1}$$

where P is the electrical power in Watt (W), V the voltage in Volt (V), and I current in Amperes (A). In order to deliver a certain power to a component, a range of voltage-current combinations are possible. In terms of power transmission efficiency, high voltage and low current is preferred. This can be seen by combining Equation 9.1, and Ohm's law, resulting in Equation 9.2.

$$P = R \cdot I^2 \tag{9.2}$$

with R the ohms resistance of a transmission line in Ohm (Ω). A lower current through the line results in a lower dissipated power. Next to that, a lower current enables the use of transmission cables with a smaller cross-section, resulting in a lighter design. However, when using COTS components, design freedom on power supply voltage is limited. A large majority of small, electronic components require regulated 3.3V, 5V, or 12V input such as the FRSKY R-XSR radio receiver[1] . More powerful devices, such as the Manifold 2-G[2], regularly contain onboard voltage regulators, allowing a range of unregulated voltage input.

Each drone has one power supply, the battery, from which each component is powered. This supply has a specific voltage, chosen in function of the motors and flying time. More on this can be found in chapter 7. In order to provide power to devices operating on differing voltage levels, voltage regulators are required between the battery and a device. A voltage regulator takes a variable voltage as input, and outputs a steady, regulated voltage. A buck regulator converts the input voltage to a lower output voltage. The difference in input and output voltage is called the headroom voltage. In general, two types of regulators can be used[3]: linear or switching regulator. Linear regulators are cost-efficient, easy to design, and low noise. However, when used for large voltage drops, efficiency can be low. The working principle is based on a variable resistor placed in a closed negative feedback loop. The drop over the variable resistor, i.e. the headroom voltage, determines the amount of dissipated heat, which can be calculated using Equation 9.3 [11].

[1] Retrieved from: `https://www.frsky-rc.com/product/r-xsr/` (cited 7 June 2022)

[2] Retrieved from: `https://www.dji.com/nl/manifold-2/specs` (cited 7 June 2022)

[3] Retrieved from: `https://www.monolithicpower.com/en/voltage-regulator-types#:~:text=A%20voltage%20regulator%20is%20a,with%20the%20other%20electrical%20components.` (cited 8 June 2022)

$$P_{HR} = (V_{in} - V_{out})I_{load} \tag{9.3}$$

with P_{HR} the headroom loss, V_{in} the input voltage, V_{out} the regulated output voltage, and I_{load} the current drawn by the load. For a transformation from 14.6V, the voltage provided by a 4S battery, to 5V, the standard operating voltage for a wide variety of USB devices, the loss amounts to at least 66% when the load draws $500\,\mathrm{mA}$. When using a higher voltage battery, e.g. 12S, this loss increases to even 89%. Next to inefficient use of the provided power, heat dissipation becomes impractical at this stage. For high voltage drops, switching regulators are a more efficient alternative. They can provide efficiencies of 80%, and higher at high voltage drops [13]. The working principle of a switching regulator is switching a switch on and off at a specific frequency and duty cycle[4]. The periodic signal is subsequently filtered by inductors and capacitors to smooth the output voltage. In this manner, both a higher, and lower output voltage can be obtained in comparison with the input voltage. However, switching regulators are less cost-efficient, and require a more complicated Integrated Circuit (IC) in order to function properly.

Next to providing the correct voltage, the noise and stability of the power supply should also be taken into account. Noise can be introduced by environmental Electromagnetic (EM) radiation, transistors, the topology of an electrical system, etc. Instability of the provided voltage can be caused by changes in power consumption by connected loads. To improve stability and noise reduction in ICs, inductors and capacitors can be implemented. The reactance of an inductor X_L is determined using Equation 9.4.

$$X_L = 2\pi f L \tag{9.4}$$

with f the signal frequency in Hz, and L the inductance in Henry (H). The linear relation between reactance and frequency makes the inductor a good choice for filtering high frequency signals, e.g. noise, when placed in series with a load. The reactance of a capacitor X_C is determined using Equation 9.5.

$$X_C = \frac{1}{2\pi f C} \tag{9.5}$$

with f the signal frequency in Hz, and C the capacitance in Farad (F). As can be seen, a capacitor lets higher frequency signals pass more easily. When a capacitor is placed in parallel with a load, the high-frequency component of the signal sees a lower resistance via the capacitor than the load. As can be noted, both capacitors and inductors can provide the same filtering function when placed appropriately in relation to the load. Increasing the number of filtering elements increases the smoothness of the outputted signal[5].

To provide flying capabilities, Brushless DC Motors (BLDCs) are used. Compared to simpler brushed motors, which can be powered by simply applying a voltage over the terminals, BLDCs require a closed-loop controller to be implemented [21]. However, at the price of higher implementation complexity and part count, longer lifetime and higher speeds can be obtained. A BLDC consists of a rotor, the part that rotates, and a stator, the part that remains stationary. The rotor is composed of permanent magnets while the stator consists of three sets of windings placed in a Y or delta configuration[6]. For every set of windings, a power line is provided, leading to three input terminals. To control a brushless motor, an ESC is used. This is an IC consisting of at least a Micro Controller Unit (MCU) and six transistors. The controller activates the transistors in a certain order depending on the motor configuration and the implemented control algorithm. A relatively simple control algorithm is a six-step commutation[7] which provides alternating step inputs to the transistor switches. A more complex but more efficient algorithm is field-oriented control[8]. It provides inputs to the transistors in a manner that generates sinusoidal variations in the magnetic field strengths of the different sets of windings [15].

In order to communicate data between components, e.g. between the FC and IMU, communication protocols have to be selected and accommodated. Two high-level concepts for data transmission exist: parallel and

[4] Retrieved from: `https://www.ablic.com/en/semicon/products/power-management-ic/switching-regulator/intro-2/` (cited 10 June 2022)

[5] Retrieved from: `https://article.murata.com/en-eu/article/basics-of-noise-countermeasures-lesson-3` (cited 10 June 2022)

[6] Retrieved from: `https://www.integrasources.com/blog/bldc-motor-controller-design-principles/#:~:text=A%20BLDC%20motor%20controller%20detects,and%20send%20out%20this%20data.` (cited 10 June 2022)

[7] Retrieved from: `https://nl.mathworks.com/help/mcb/ref/sixstepcommutation.html` (cited 10 June 2022)

[8] Retrieved from: `https://nl.mathworks.com/help/mcb/ref/sixstepcommutation.html` (cited 10 June 2022)

serial communication[9]. In parallel communication, several bits are transmitted simultaneously, providing high speed at the cost of higher interference, and shorter link distances. This way of transmission is dominantly used within integrated IC's such as a MCU. In serial communication, bits are sent sequentially, frequently packed in packets of 8 bits, called a byte. This type of communication provides robustness against interference and longer transmission lines at the cost of lower transmission speeds. The majority of inter-device and network communication, e.g. between an online server and a personal computer, utilises serial protocols. An overview of relevant serial communication protocols for the current application is provided in the following list.

- Serial Peripheral Interface (SPI)[10]: a serial communication protocol frequently used for communication between MCUs and peripheral ICs (e.g. sensors). It is a synchronous protocol, meaning that one device, the controller, provides a clock signal to the peripheral. This clock signal is used for timing the sending and receiving of bits. The protocol is full-duplex. This means that bits can be transmitted in both directions simultaneously. At least three lines are required: one for the clock signal, one for communication from the controller to the peripheral, and one for communication from the peripheral to the controller. Additional lines have to be used in case multiple peripherals are added to the bus.
- Universal Asynchronous Receiver-Transmitter (UART)[11]: technically, UART is not a communication protocol but a physical IC that converts parallel bits into a bit sequence, which subsequently is transmitted over a serial line to a second UART. Communication is performed asynchronously. This means that no shared clock signal is required. Instead, start and stop bits are used for synchronisation. The lack of a shared clock signal reduces the number of signal lines. To connect UART devices, only two lines are required: a transmitting (TX) and receiving (RX) line.
- Universal Serial Bus (USB)[12]: widely used serial, the asynchronous protocol for connecting a wide variety of peripheral devices such as keyboards, cameras, and external memory. Different versions of the protocol exist; 3.0 is the newest and most powerful iteration. However, 2.0 is still widely used. Version 2.0 requires at least four lines: two for power and two for transmission of data. Due to the presence of power lines, USB can also provide power to connected devices on top of half-duplex communication. This means that communication is possible in both directions, though only in one direction at a time.
- SBus[13]: a serial protocol mainly used for RC applications. It makes use of only one line of a UART device making it a simple communication protocol.
- Ethernet[14]: widely used for connecting devices to local networks. It is a key technology that made the foundation and growth of the internet possible. Different, backwards compatible versions have been proposed throughout the years, providing increasingly higher transmission speeds. The most recent version consists of four pairs of transmission lines, providing half-duplex communication. Nonetheless, full-duplex communication is possible in specific applications.

9.2. Subsystem Requirements

In Table 9.1, the requirements for the electrical subsystem can be found.

[9] Retrieved from: `https://www.versitron.com/blog/a-comparison-of-parallel-data-transmission-and-serial-data-transmission` (cited 8 June 2022)

[10] Retrieved from: `https://www.analog.com/en/analog-dialogue/articles/introduction-to-spi-interface.html` (cited 9 June 2022)

[11] Retrieved from: `https://www.circuitbasics.com/basics-uart-communication/` (cited 9 June 2022)

[12] Retrieved from: `https://www.usb.org/` (cited 9 June 2022)

[13] Retrieved from: `https://oscarliang.com/rc-protocols/` (cited 9 June 2022)

[14] Retrieved from: `https://www.techtarget.com/searchnetworking/definition/Ethernet#:~:text=Ethernet%20is%20the%20traditional%20technology,rules%20or%20common%20network%20language.` (cited 10 June 2022)

Table 9.1: Electrical subsystem requirements.

Identifier	Requirement	Related System Requirement
SUB-ELC-1	The electrical subsystem shall provide the power required by all the electronic components	SYS-PER-01
SUB-ELC-2	The electrical subsystem shall provide communication lines with a minimum bandwidth dictated by the specific component	SYS-PER-04
SUB-ELC-3	The electrical subsystem shall provide the capability of monitoring the total power consumption during operation	SYS-REL-03
SUB-ELC-4	The electrical subsystem shall provide the capability of controlling the motors	SYS-PER-01

9.3. Electrical Subsystem Design

In this section, the electronic design outline can be found. In Table 9.2 and Table 9.3, the electrical components for the “CHIP” drone and the “DALE” drone can be found, respectively. For every entry, the commercial name, a general name indicating its function, and the unit count is given. Measures such as the mass, dimensions, and power are provided per individual component. Note that power is the peak power consumed by the component since for the majority of the components only this measure was publicly available. This leads to a rather conservative sizing of the battery. The schematics outlining the connection between all the components for the “CHIP” drone and “DALE” drone, can be found in Figure 9.2 and Figure 9.1. First, the configuration of the “CHIP” drone will be discussed, followed by a discussion on the comparative differences of the “DALE” drone.

In the following discussion, Figure 9.2 will be used as the starting point. The components will be referenced by their electrical reference designator, e.g. C1 for a capacitor. The battery (BT1) provides the required power to all the electrical components. It is connected directly to the Power Distribution Board (PDB) (U12). A capacitor (C1) is placed in parallel with the battery to compensate for potential voltage fluctuations. The PDB provides power lines for the motors. Next to that, it provides voltage rails in $3.3\,\mathrm{V}$, $5\,\mathrm{V}$, $12\,\mathrm{V}$, and battery voltage to power devices with different input voltage requirements. Additionally, input and output signal lines (S1, S2, S3, S4) for control of the motors are included. The input signals are provided by the FC and routed to the ESC’s. Finally, current measurements FC are provided through pin CRT to monitor power consumption.

Every ESC (U15-U18) has three input pins connected to the PDB, two for power and one for the motor Pulse Width Modulation (PWM) control signal . The ESC has three output pins, connected to the motor (M1-M4), for delivering power. The ESC determines the correct sequencing of power signals to the motor in order to provide the required speed. A discussion on control of the motor can be found in Section 9.1.

The required speed and power from the motor are determined by the FC(U9). The PWM output signal is determined by a combination of positional user input data, and inertial data from the embedded IMU. More information can be found in chapter 6. Positional user data can be provided by two sources: the RF antenna (U11) or the flight computer (U4). The choice of input can be determined programmatically. The RF antenna can receive control information from a classical remote controller for manual control by a human operator. The FC and the RF antenna are connected through an SBus interface. During nominal operation, the flight computer will provide positional information. The FC and flight computer are connected through a UART interface. The input to the FC is determined by a combination of the predetermined inspection path and sensor data from the LIDAR sensor (U10) and stereo cameras (U1-U3, U6-U8). More information on this can be found in chapter 5.

Table 9.2: Electronic and electrical component list for the “CHIP” drone.

Name	Component	Nr	Mass (g)	Dimensions (mm)	Power (W)	V	I
Manifold 2-G	Flight computer	1	230	91 x 61 x 35	25	25	1
Intel® RealSense™ Depth Module D450	Depth camera	6	29.5	119.5 x 17.4 x 10.53	-	-	-

Vision Processor D4 USB Type-C V3 Board	Depth camera processor	6	3.96	72.4 x 16 x 3.94	3.5	5	0.7
FRSKY R-XSR	RC antenna	1	1.5	16 x 11 x 5.4	0.35	5	0.07
USB Hub 6 Downstream[15]	USB hub	1	224	100 x 61 x 20	0.3135	3.3	0.095
OS0 LIDAR	LIDAR sensor	1	447	85 x 85 x 73.5	23	25	0.92
FCHUB-12S	PDB	1	21	55 x 50 x 6	0.3465	3.3	0.105
FC F405-STD	FC	1	7	36 x 36	0.792	3.3	0.24
Zenmuse P1	Inspection camera + gimbal	1	800	198 x 166 x 129	20	12	1.67
Rctimer HP8108 135KV	Motor	4	239	86.8 x 86.8 x 26.6	see P&P	-	-
MAD AMPX 40A ESC HV	ESC	4	22.5	51 x 27 x 12.5	see P&P	-	-
25V buck regulator	Voltage Regulator	1	20	107 x 102 x 14	8.5	-	-
5V embedded buck regulator	Voltage regulator	1	-	-	3.9	-	-

The flight computer (U4) is the hardware core of the autonomous capabilities of the inspection system. It performs the calculations required for path finding, collision avoidance, and inspection. It communicates to the flight computer via Wi-Fi. It is interfaced to the LIDAR sensor (U10) through ethernet, and to the inspection camera (U14) through USB. The stereo cameras (U1-U3, U6-U8) are connected through a USB hub (U5). The flight computer receives power from a 25 V switching buck regulator (U13). The inspection camera (U14) includes the gimbal for pointing.

Overall, the “DALE” drone electrical design resembles that of the “CHIP” drone. However, some particular differences exist. These differences will be addressed underneath. Features and connections similar to the “CHIP” drone will not be discussed. For this, the reader can refer to the description of the “CHIP” drone. The schematic of the “DALE” drone can be found in Figure 9.1, and will be used as a reference.

The battery (BT1) plugs into a combined PDB and 4 in 1 ESC (U15). The 3-phase motor power signals are regulated on the board instead of delivering power to four separate ESC’s. The LIDAR sensor (U5) for the “DALE” drone, the Velabit LiDAR, is a more compact version with reduced capabilities. While the sensor for the “CHIP” drone, the OS0 LIDAR, provides a 360° horizontal FOV of the environment, the Velabit only provides a 90° FOV. However, its power consumption is considerably less. While the inspection camera for the “CHIP” drone offers an integrated camera and gimbal solution, this is not the case for the camera of choice for the “DALE” drone. Therefore, a gimbal controller (U14) has been selected to drive the gimbal motors (M1-M3). In order to control the motors correctly, inertial data is provided by an IMU (U16) attached to the camera. The interface uses the SPI protocol. The inspection camera (U18), and the gimbal controller are connected to the flight computer through a USB hub (U17).

Table 9.3: Electronic and electrical component list for the “DALE” drone.

Name	Component	Nr	Mass (g)	Dimensions (mm)	Power (W)	V	I
DarwinFPV Betaflight F4 60A 3-6S BLHeli_32 4in1 Dshot1200 ESC	ESC	1	12.7	30 x 30	0.3465	3.3	0.105
DarwinFPV Betaflight F4 60A 3-6S BLHeli_32 4in1 Dshot1200 FC	FC	1	7.2	30 x 30	0.792	3.3	0.24
Manifold 2-G	Flight computer	1	230	91 x 61 x 35	25	25	1

Intel® RealSense™ Depth Module D450	Depth camera	6	29.5	119.5 x 17.4 x 10.53	-	-	-
Vision Processor D4 USB Type-C V3 Board	Depth camera processor	6	3.96	72.4 x 16 x 3.94	3.5	5	0.7
FRSKY R-XSR	RC antenna	1	1.5	16 x 11 x 5.4	0.35	5	0.07
USB Hub 6 Downstream	USB hub	1	224	100 x 61 x 20	0.3135	3.3	0.095
STEVAL-GMBL02V1	Gimbal controller	1	35.6	50 x 50	0.528	3.3	0.16
Gimbal motor	Gimbal motor	3			5		
Velabit LiDAR	LIDAR sensor	1	125	6 x 6 x 3.5	6	5	
USB hub 2 downstream	USB hub	1	14	65 x 30 x 9	0.3135		
Evo Nano Camera	Inspection camera	1			5		
8V buck regulator[16]	Voltage regulator	1	4.5	12.5 x 13.5 x 3.5	1	-	-
5V buck regulator[17]	Voltage regulator	1	4.5	12.5 x 13.5 x 3.5	3.7	-	-

9.4. Electrical Subsystem Design Choices and Analysis

In this section, some design considerations and basic electrical analysis are discussed. As discussed in Section 9.3, power regulators are required to supply a selection of devices with the correct input voltage. More specifically, a 5V and 8V regulator for the “DALE” drone and a 25V regulator for the “CHIP” drone are required. Linear regulators require less room and are less complicated to implement than switching regulators. However, their efficiency can be low for high voltage drops. For each linear regulator, the efficiency can be calculated by using Equation 9.3 and Equation 9.6.

$$\eta = \frac{P_{load}}{P_{load} + P_{HR}} \tag{9.6}$$

where η is the efficiency, P_{load} the power consumed by the load, and P_{HR} the headroom power. In Table 9.4, efficiencies can be found for the potential linear regulators. Note that these efficiencies are the theoretical upper limit since they are only based on the operating principle; additional physical losses due to resistance, leakages, etc. in other parts of the regulator than the variable resistor are not taken into account.

Table 9.4: Linear regulator efficiencies.

Output voltage	Input voltage	Efficiency
5V	14.6V	34%
8V	14.6V	55%
5V	44.4V	56%

[15] Retrieved from: https://www.microchip.com/en-us/development-tool/EVB-USB2517#Documentation (cited 8 June 2022)

[16] Retrieved from: https://nl.rs-online.com/web/p/switching-regulators/1933956?cm_mmc=NL-PLA-DS3A-_-google-_-CSS_NL_NL_Power_Supplies_%26_Transformers_Whoop-_-(NL:Whoop!)+Switching+Regulators+(2)-_-1933956&matchtype=&pla-301160920730&gclid=CjwKCAjwkYGVBhArEiwA4sZLuFODq5isP2dnHasEMaThKt_b0hfdqjVkP-nrP8f2hkrLJXRRJcyoUBoCGHAQAvD_BwE&gclsrc=aw.ds (cited 8 June 2022)

[17] Retrieved from: https://nl.rs-online.com/web/p/switching-regulators/1933956?cm_mmc=NL-PLA-DS3A-_-google-_-CSS_NL_NL_Power_Supplies_%26_Transformers_Whoop-_-(NL:Whoop!)+Switching+Regulators+(2)-_-1933956&matchtype=&pla-301160920730&gclid=CjwKCAjwkYGVBhArEiwA4sZLuFODq5isP2dnHasEMaThKt_b0hfdqjVkP-nrP8f2hkrLJXRRJcyoUBoCGHAQAvD_BwE&gclsrc=aw.ds (cited 8 June 2022)

Although linear resistors provide several advantages over switching regulators, the maximum obtainable efficiency makes them an unfavourable choice. For example, the 5V buck regulator on the “DALE” drone has to be able to output 21 W of power to the stereo cameras. With an efficiency of 34% for a linear regulator, this would amount to a headroom loss of 41 W. This is a considerable loss compared to the consumption of other components. Next to that, a relatively large heat sink would be required, taking up valuable space and mass. As a result, switching regulators have been selected for implementation. For the first design iteration, 85% efficiency has been assumed. This figure should be reconsidered in later design iterations when a detailed design of the regulators has been provided and implemented. The detailed design of the regulators is planned for later design phases. However, it can already be determined that a design includes at least an input capacitor for smoothing the supply voltage to the transistor switch, and an output capacitor-inductor combination to transform the block voltage signal from the switch into a steady power supply.

Additionally, the capacity of the power supply and communication lines had to be analysed. By analysis, it was determined that the power and communication lines provided sufficient capacity to accommodate the required power for every device. In order to do this, it was chosen to implement a USB hub that features a dedicated power supply instead of using the default power supply through the USB connection between the hub and the flight computer. The total 21 W required by the stereo cameras is too high to be supplied by the USB connection between the hub and the flight computer.

For every component, the required power has been looked up or calculated. These values are reported in Table 9.2 and Table 9.3. The power for COTS components has been taken as the maximum possible power consumed by the component. This was done due to the limited availability of public information. In later iterations, these values can be adjusted after initial implementation tests and measurements of power consumption during nominal operation. The power dissipation by the buck converters has been determined by analysing the total power throughput combined with the assumed efficiency of 85%. The total power consumption excluding the motors and heat dissipation by the ESCs amounts to 103.2 W for the “CHIP” drone and 79.3 W for the “DALE” drone.

9.5. Remarks

In this section, final remarks on the electrical design are made. These remarks can be considered in potential later design iterations. Currently, a heat analysis has not been performed on the electrical design. Due to the use of COTS components, this has not been considered a necessary analysis during the first iteration. However, this has to be considered in later design stages. As a first approach, this could be taken into account by measuring temperature rises in relevant components during tests. If it is shown that heat dissipation poses a problem, this can then be explicitly taken into account during later design iterations.

For a first estimate of power consumption, peak power values have been considered due to the limited availability of public information. For most COTS components, only peak power values are provided by the manufacturer. This leads to a conservative sizing of the battery. In later design stages, this estimate can be refined by performing tests on the physical components by measuring power consumption.

The current electrical design is based on a quick as easy integration of COTS components. It allows for rapid assembly of the drone, decreasing iteration time. However, the custom design of components may lead to power and mass savings. For example, the flight computer features two USB ports. In the design, one of these ports connects to a USB hub which hosts the stereo cameras. If the flight computer would be custom-designed, e.g. by selecting a Central Processing Unit (CPU) specifically selected for the current application or ‘embedding’ the USB hub by providing more ports on the motherboard, power and mass savings can be made. In later design iterations, it should be considered whether the benefits of custom design outweigh the additional development time and costs.

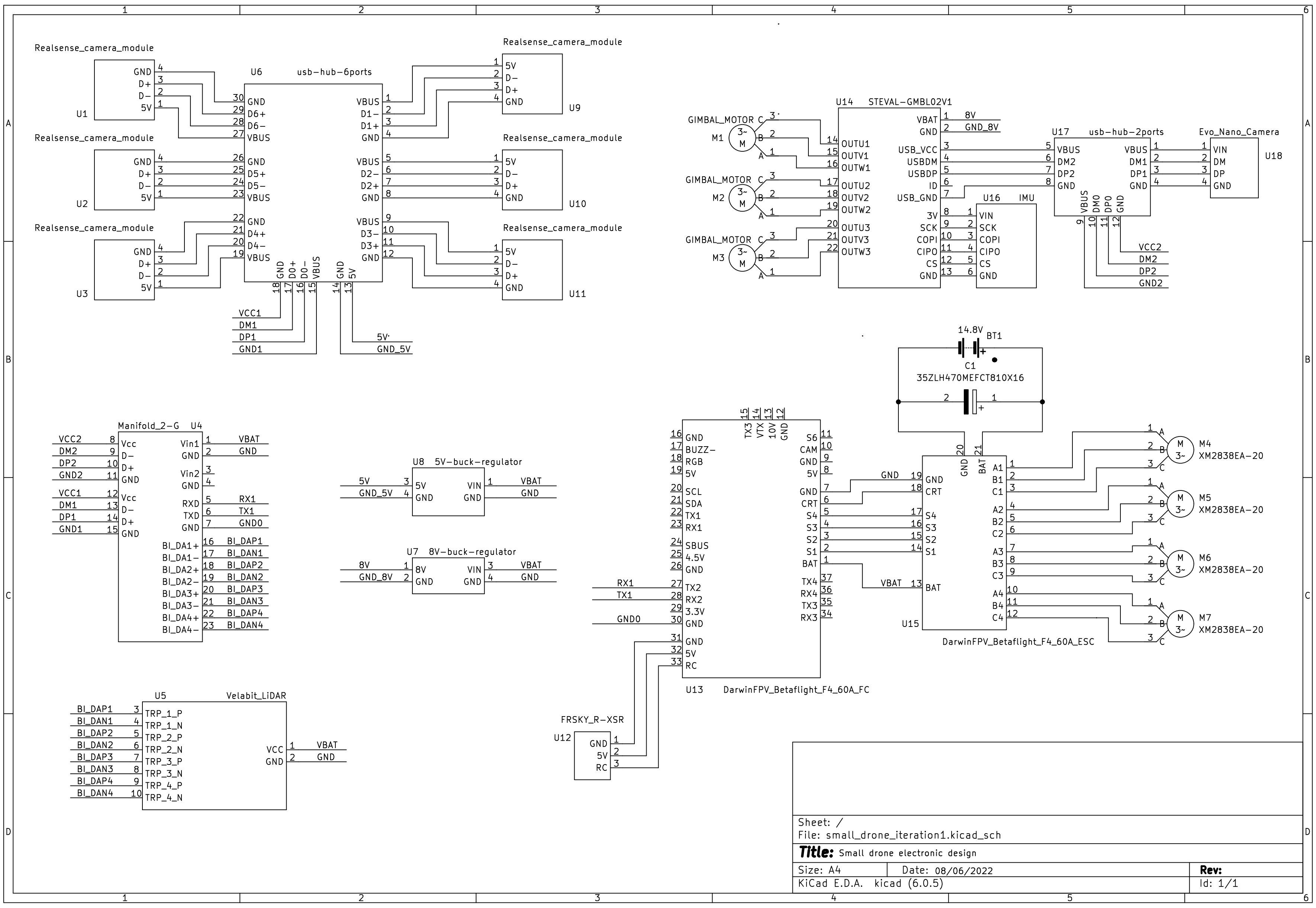


Figure 9.1: “DALE” drone electronic schematic

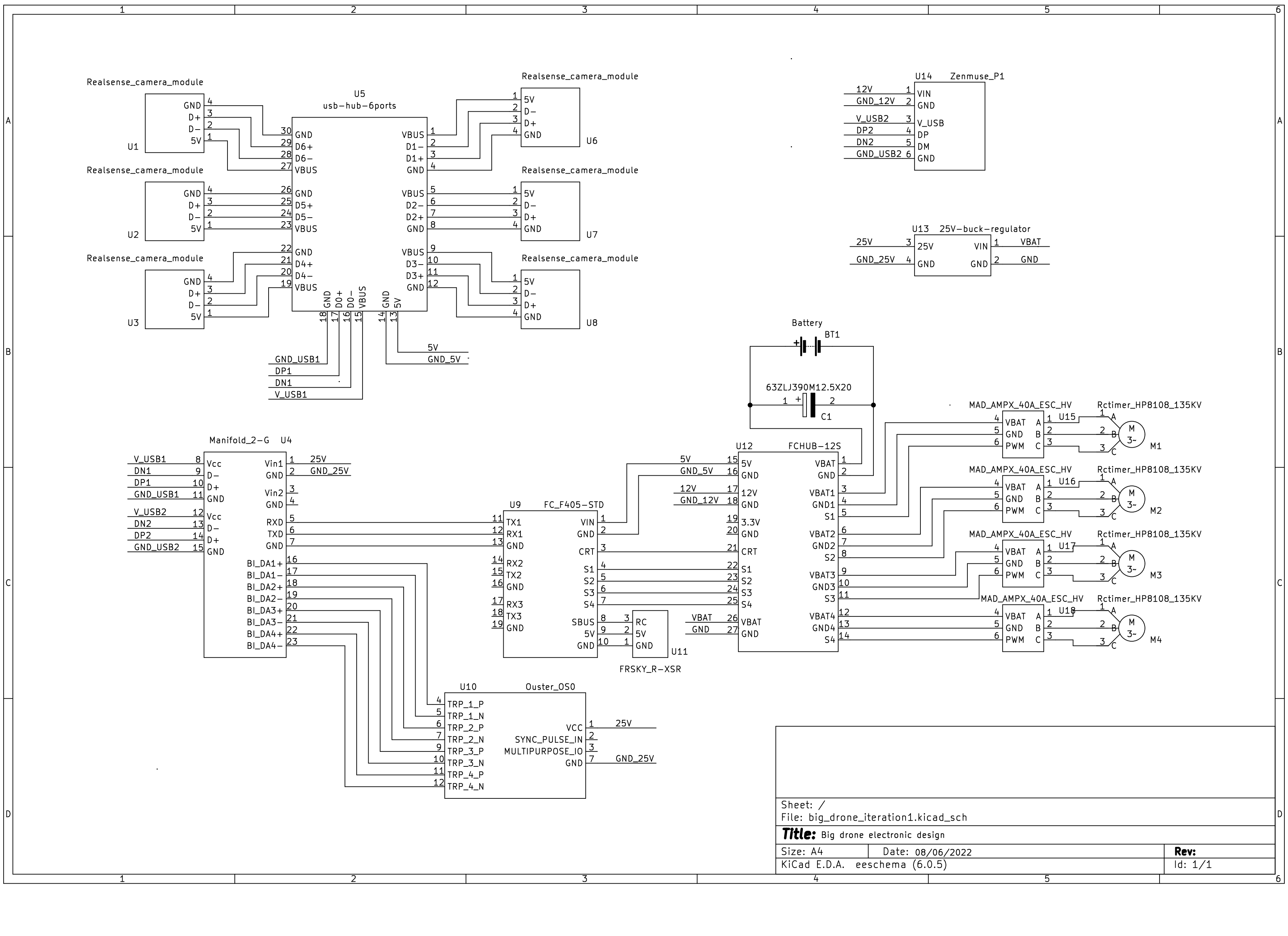
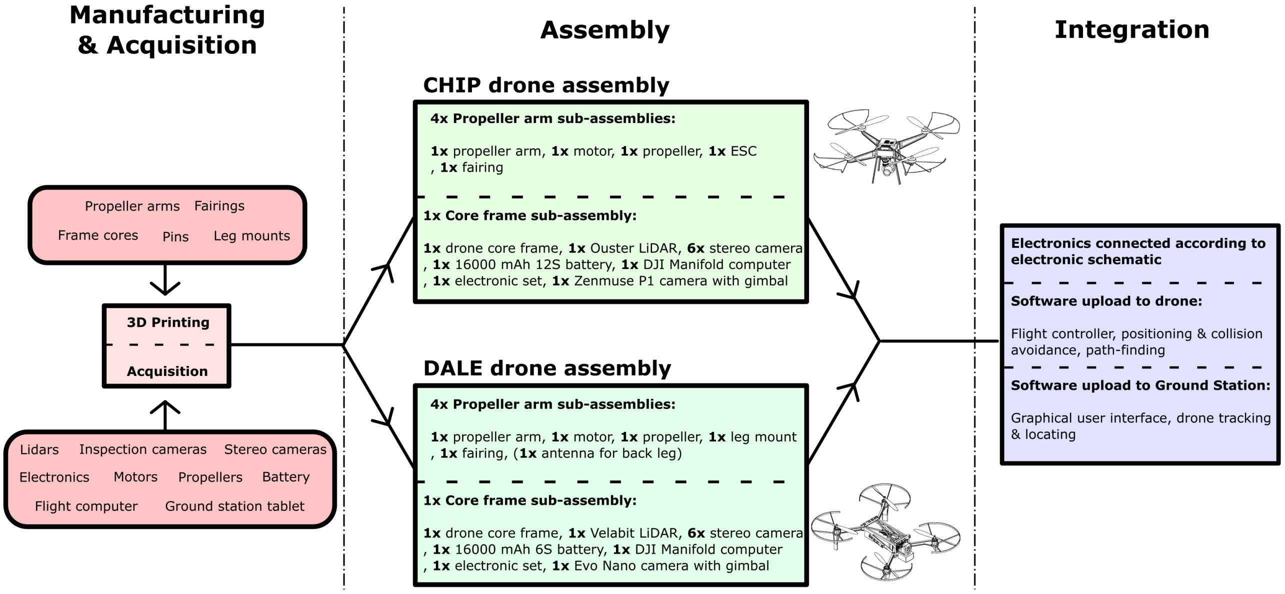


Figure 9.2: “CHIP” drone electronic schematic

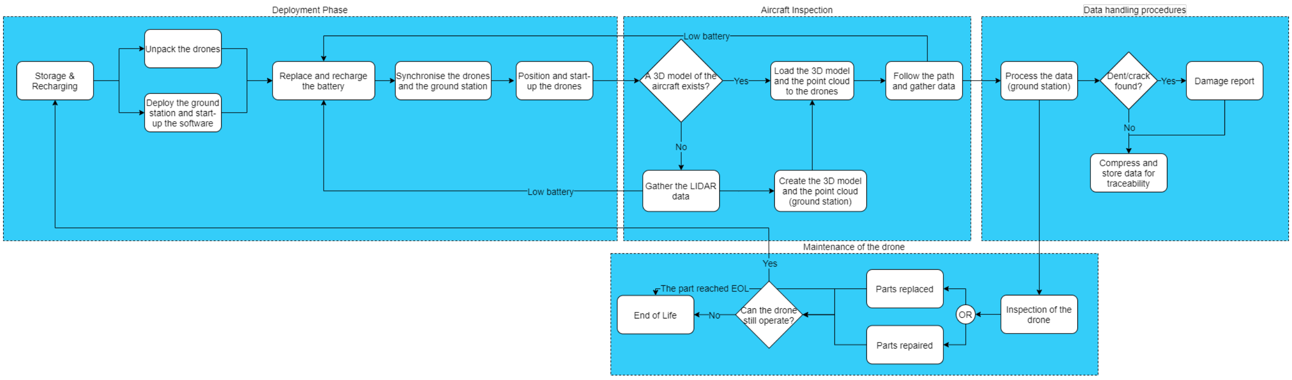
10 Manufacturing, Operations, and Logistics

The first steps to take before commencing operations is to manufacture and deliver the drone systems. Figure 10.1 shows the outline of the production procedures of the drone system, which is split into manufacturing and acquisitions, assembly, and integration. Next, general overview of operations and logistics of the drone in a system of a swarm of drones is shown in Figure 10.2. This operations cycle starts with the storage of the drone, after which the deployment phase starts. This is followed by the aircraft inspection phase, and data handling procedures. After the drone has finished its job, maintenance of the drone has to take place and parts that fail can be either replaced or repaired. If after the maintenance the drone is still operative, the cycle continues. If the drone is inoperative, then it has reached its EOL. All these phases are described in more detail in this chapter.



Manufacturing & Acquisition
Assembly
Integration
Propeller arms
Fairings
Frame cores
Pins
Leg mounts
3D Printing
Acquisition
Lidars
Inspection cameras
Stereo cameras
Electronics
Motors
Propellers
Battery
Flight computer
Ground station tablet
CHIP drone assembly
4x Propeller arm sub-assemblies:
1x propeller arm, 1x motor, 1x propeller, 1x ESC , 1x fairing
1x Core frame sub-assembly:
1x drone core frame, 1x Ouster LiDAR, 6x stereo camera , 1x 16000 mAh 12S battery, 1x DJI Manifold computer , 1x electronic set, 1x Zenmuse P1 camera with gimbal
DALE drone assembly
4x Propeller arm sub-assemblies:
1x propeller arm, 1x motor, 1x propeller, 1x leg mount , 1x fairing, (1x antenna for back leg)
1x Core frame sub-assembly:
1x drone core frame, 1x Velabit LiDAR, 6x stereo camera , 1x 16000 mAh 6S battery, 1x DJI Manifold computer , 1x electronic set, 1x Evo Nano camera with gimbal
Electronics connected according to electronic schematic
Software upload to drone:
Flight controller, positioning & collision avoidance, path-finding
Software upload to Ground Station:
Graphical user interface, drone tracking & locating

Figure 10.1: Manufacturing, assembly, and integration flowchart



Deployment Phase
Storage & Recharging
Unpack the drones
Deploy the ground station and start-up the software
Replace and recharge the battery
Synchronise the drones and the ground station
Position and start-up the drones
Low battery
Aircraft Inspection
Low battery
A 3D model of the aircraft exists?
Yes
No
Load the 3D model and the point cloud to the drones
Follow the path and gather data
Gather the LIDAR data
Create the 3D model and the point cloud (ground station)
Data handling procedures
Process the data (ground station)
Dent/crack found?
Yes
No
Damage report
Compress and store data for traceability
Maintenance of the drone
Yes
The part reached EOL
End of Life
No
Can the drone still operate?
Parts replaced
OR
Parts repaired
Inspection of the drone

Figure 10.2: Operations and logistics flow chart.

10.1. Manufacturing, Assembly, and Integration

Referring to Figure 10.1, the drones parts first need to be 3D printed and off-the-shelf components need to be bought. These parts are then assembled together to produce the two drones, where then all the drone subsystems are integrated so the system can function properly.

10.1.1. Manufacturing

Manufacturing of the skeletal parts will be manufactured by additive manufacturing as stated in Subsection 8.4.1. Despite the limitations of this method, it is the most promising method that enables production of sustainable materials with minimal waste. To be more specific, the skeletal structure will require a 3D printer with a heating bed while material will be obtained from a supplier in a filament wheel form. The individual elements such as Fairings, Arms, Core Structures and pins (only for “CHIP” drone) will be manufactured individually to be assembled later on. Although it cannot be determined from the start, depending on the application quality, the grinding procedure with a sand paper may be required on specific structural surfaces in order of obtaining smooth surfaces. As a last point, each drone must withstand large bending moments because of its weight and propeller force. This suggests that changing the print direction during manufacturing is necessary. Owing to this shape providing the best rigidity when considering bending moments, all pieces will be manufactured in the XZ direction as indicated in Figure 8.13.

10.1.2. Assembly and Integration

After all the parts are manufactured, they will have to be assembled and the components integrated. For a product of this scale, these steps are not very complicated and are fairly quick. Note here that Figure 10.1 mentions an electronic set, this is a metallic case housing a number of small electronic items such as voltage regulators, circuit boards, and a USB hub.

Assembly

Referring to Figure 10.1, both drones are divided into 4 propeller arm sub-assemblies, and 1 core sub-assembly.

Assembly of the propeller arm is simple. For “CHIP” drone, the motor and fairings will be screwed to the propeller arm, then the propeller can be screwed onto the motor, and finally the ESC can be secured to the root of the propeller arm. And for “DALE” drone, the propeller is screwed onto the motor, after which the motor and leg mount can be screwed onto the propeller arm, followed by clipping on the safety fairings to the motor housing.

The “CHIP” drone battery is slid into the drone core frame, where it is subsequently secured. The electronic set along with the flight computer are attached to the top of the frame, next the LIDAR and Zenmuse camera are attached to the bottom of the frame, lastly all 6 stereo cameras can be screwed onto the 6 faces of the frame. For “DALE” drone, the Nano camera and Velabit LIDAR are first screwed onto the front of the core frame, then the Manifold computer and electronics set are screwed onto the interior of the frame, the battery can then be slid into the holder rails inside the frame and secured, and as the last step all 6 stereo cameras can be screwed onto the mounts that are on the surface of the frame.

Once the sub-assemblies are assembled, the four propeller arms can be attached to the four corners of the drone core for both drones, and then here the “CHIP” drone will need 4 hexagonal pins to be slotted into the arm hinges for securing.

Integration

After the assembly is complete, the final integration of electronic components and software upload can take place. Electronic integration involves connecting the electronics according to the schematics given in chapter 9, under Figure 9.2 for the “CHIP” drone and Figure 9.1 for the “DALE” drone. The software which needs to be uploaded to the drones are the flight controller, positioning algorithm and Collision avoidance software, and path-finding software. To finish, the Graphical user interface and drone tracking software needs to be installed onto the ground station tablet, and the system is ready to use.

10.2. Deployment Phase

According to the requirement StRS-MO-07, the swarm of drones with the ground station shall fit in 4 suitcases with specified dimensions. Therefore they are to be stored in the same suitcases. During storage and before the inspection, the drones and the replacement batteries have to be charged. Right before the inspection the drones and the ground station should be unpacked and deployed. The charge of the drones has to be checked. The drones have to be synchronised with the ground station and their position relative to the aircraft has to be specified to the drones, in order for the navigation to work correctly, according to chapter 5. After that the drones are ready to start the inspection of the aircraft.

10.2.1. Deployment time

For the operator, the deployment phase consists of taking the drones out of the suitcases, setting them up, and connecting them to the ground station. It is estimated that will take 3 minutes per “CHIP” drone to unpack it, and 1 minute per “DALE” drone. This sums up to 9 minutes. Another 5-10 minutes are estimated for connection of the drones with the ground station. It is hard to predict the exact time it will take and will have to be verified when the system is developed. This ends up being 15-20 minutes which almost fits under the requirement on the time the deployment phase should take (SYS-TIM-03). Same timing is expected for packing-up the drones.

10.2.2. LiPo Battery Operations

LiPo batteries can only perform during maximum of 200 cycles, it is important to follow several rules according to the battery’s operation, in order for it to last as long as possible. The specified charging and discharging rates should be used. Usually LiPo batteries are charged at 1C rate (full charge in one hour of charging), and are not recommended to be discharged below 3V[1]. In addition, the charging station shall utilise balanced charging mode at all times. In essence, the balanced charging mode employs incremental potential difference increase along the charging process as opposed to the traditional charging modes which supplies fixed potential difference until the battery is fully charged. Meanwhile, due to the balanced nature of charging, the damage caused on batteries is minimal. Consequently, balanced charging method increases the lifespan of a battery and provides an optimal lifetime throughout the operation. On the other hand, It is advised to store the batteries discharged at 3.8V, when they are not used for at a least a day. At the end of the batteries lifetime, it should be recylced as suggested by sustainable development in chapter 11.

10.3. Aircraft Inspection

The aircraft inspection starts with loading up the model of the aircraft and the point cloud to the drone. From these the path for the drones is generated as explained in Subsection 5.2.3. If it is a first inspection of a certain model of the aircraft, a pre-inspection procedure of generating the 3D model and the point cloud should be taken. The drone is manually navigated around the aircraft to gather the data from LIDAR sensor. This data then has to be processed by the ground station to generate the 3D model. As all drones have the LIDAR sensor, it is possible to use the “CHIP” drone to gather the general LIDAR data, and the “DALE” drone to gather the LIDAR data of lower surfaces and hard to reach areas, where the “CHIP” drone cannot reach. According to Mainblades[2], post-processing to generate the 3D model takes approximately 60hr. Therefore even though this procedure is specified as part of aircraft inspection in the flow chart, this is a pre-inspection procedure, that has to be taken once per the model of the aircraft. When the system is deployed, the procedure of generating the model may be optimised to take less time. After the model is loaded into the drones, they can take-off and follow the path to gather the visual data of the aircraft. If during the inspection or gathering the LIDAR data the battery of the drone is discharged, it will land and the battery has to be replaced as specified by the requirement SYS-SUS-06. According to the simulation (Section 5.4), the inspection of the narrow-body aircraft takes 16 minutes.

10.4. Data Handling Procedures

After landing, the data should be transferred from the drones to the ground station. As stated in Section 4.4, the drones store the data on SD cards. Therefore they should be taken out from the drones and inserted into the ground station. After all the data is transferred, the algorithm is run to analyse the data. This is explained in Section 4.4. The damage report is then produced and given to the operator. According to the Section 4.4, the data is to be analysed within 15 minutes.

10.4.1. Damage report

As was stated in the flow chart Figure 10.2, a damage report will be produced. This is a document that will contain all the information regarding the inspection, and be given to the operator (also available in traceability software Subsection 4.4.3). The damage report will have the following information available:

- Model of the aircraft.
- Time taken for inspection and data handling.

[1] Retrieved from: `https://www.robotshop.com/media/files/pdf/hyperion-g5-50c-3s-1100mah-lipo-battery-User-Guide.pdf` (cited 9 June 2022)

[2] Mainblades private communication, May 2022

- The percentage of the area covered (in the software the area not covered will be specified).
- The amount of damages found.
- Types, sizes, and amount of damages found.
- On which parts of the aircraft the damages were found.

This will give the operator enough information on the inspection, and which repair should be conducted during maintenance. Combining that with the software that provides traceability and visualisation of damages (Subsection 4.4.3) should potentially increase the efficiency of maintenance.

10.5. Drone Maintenance

After the inspection is finished, the drone has to be inspected. If damage is found, or some parts of the drone failed, those should either be replaced or repaired. As specified in Subsection 8.6.1 a lot of drone parts are replaceable and in some cases parts can be repaired by gluing with adhesive epoxy. If the drone remains inoperative after the maintenance then it has reached its EOL it should be recycled as suggested by sustainable development in chapter 11. The EOL responsibilities are taken by the development team. After the drone is checked and repaired if needed, it can be stored in the suitcases until the next inspection.

10.6. Daily Number of Inspections

The system shall have a daily work hour utilisation that is as high as possible. System setup, inspection, and pack-up shall take up to one hour per inspection, with data analysis being performed off-board (see SYS-TIM-03, SYS-TIM-04, SYS-TIM-05, and SYS-TIM-06). Charging of a LiPo at 1C takes one hour, with one battery sustaining flight for one full inspection under nominal conditions. Inspection and battery charging both require one hour, and can thus be performed at the same time, given that there are at least two sets of LiPo's available at all times. Thus, a total of eight inspections is possible under perfect operating conditions (one inspection per working hour). This figure was brought down to six inspection per day to account for delays, and possible battery replacements during inspection.

10.7. Operating Conditions

The swarm of drones is designed to be able to operate both inside and outside the hangar. Inside the hangar, a local collision avoidance system is implemented (Subsection 5.3.2) to ensure safety of the workers and damage to the objects in the hangar is avoided. For outside operations, the system is operable with wind gusts up to 5 m/s, and a minimum visibility of 100m. It is not operable outside when there is rain or fog, as some of the electronics are not covered, as can be seen in chapter 8.

11
Sustainable Development

This section details how the team will tackle the technical details of sustainable development. Sustainability during manufacturing and production is detailed in Section 11.1, operations sustainability is discussed in Section 11.2, followed by EOL solutions in Section 11.3. Economical and social considerations are listed in Section 11.4. The chapter is finalised with product contribution to sustainability in Section 11.5. The sustainability aspects mentioned in this chapter have to be taken into account for each subsystem in the detailed design phase, during which specific requirements on sustainability will be formulated.

11.1. Manufacturing and Production

Manufacturing and production can be divided into several topics: choice of material, choice of manufacturing techniques, and lean manufacturing. These topics are discussed in their relative subsections below.

11.1.1. Materials

Three important properties will be accounted for in the selection of the materials for the drone system. These are the recyclability of the material, the degree to which the material can be used in additive manufacturing, and the durability of the material. The recyclability of the material will be important in removing pressure from the end of life solutions. The degree to which the material can be used in additive manufacturing is important to reduce waste. This is discussed in more detail in Subsection 11.1.2. Lastly, it is important to consider the durability of the material. A durable material will have a long lifetime, which results in less waste. The materials that are taken into consideration are Carbon Fiber Reinforced Composites (CFRC), thermoplastics, and metals.

CFRC have a superior strength to weight ratio compared to other materials, which makes them an excellent choice to be used in drone structures. The mass of a drone has to be minimised to decrease required thrust and increase flight time. Although CFRC are very useful in the manufacturing of lightweight drone structures, their production is very energy intensive, expensive, and has a negative impact on the environment [71]. The recycling of these composites is an active research topic, in which the three main methods are mechanical, thermal, and chemical recycling. Mechanical and thermal recycling seem to be better options for industry scale applications [55]. However, continued research is required to ensure whether the recycled fibres have the same properties as virgin fibres. One of the methods is to combine the recycled fibres with Acrylonitrile Butadiene Styrene (ABS), a plastic used in injection moulding and additive manufacturing, to produce carbon reinforced pellets, which can then be converted to filament to be used in 3D printing[1].

Thermoplastics are common in drone applications. They can easily be formed into complex parts using injection molding or additive manufacturing. The two main thermoplastics used are ABS and PLA. ABS is fossil based and as a result, has bad consequences for the environment. PLA is made from corn starch, a renewable source [56]. Thermoplastics are also easily recyclable as they can be melted down after use to be used in the manufacturing of new parts.

Metals are rarely used in the manufacturing of a drone structure due to cost and the complex shapes required to minimise the mass. The latest developments in additive manufacturing of metals might make it possible to manufacture a lightweight structure that can be used in drones. Methods for recycling damaged parts have been proposed. With this, less energy is needed to create the feedstock used for additive manufacturing [76].

As was discussed in chapter 8, metals and CFRC were discarded and PP-GF30 was chosen to be used as the drone frame material. As PP-GF30 is a glass fibre reinforced composite, it has a relatively high Young's modulus to density ratio. Therefore, this material is durable and will result in a reduced structure mass compared to other homogeneous plastics.

When a part fails during operations, it is replaced, and the damaged part is recycled. The most common recycling method for glass fibre reinforced composites is grinding the old parts into granules and use these to create new recycled parts [17]. By grinding the old parts back down into granules, the same production method can be used for the original parts as for the recycled parts.

[1] Retrieved from: `https://www.compositesworld.com/articles/sustainable-inline-recycling-of-carbon-fiber` (cited 11 May 2022)

The recycled parts perform slightly worse than the original parts, e.g., the Young's modulus has a reduction between 8.5 % and 18.8 %. But these are still very well suited to be used in structural applications [17].

11.1.2. Manufacturing

Additive manufacturing has a lot of advantages in terms of reducing waste, increasing replaceability, and topology optimisation. Including recyclable materials, as discussed in Subsection 11.1.1, additive manufacturing for thermoplastics and metals should be used. To improve sustainability, the chosen manufacturing techniques should use as many green energy sources as possible.

Technology is ever-changing, which results in the availability of better and more efficient components after the design of the drone is complete. To be able to easily adapt the design of the drone to incorporate these new technologies, the use of additive manufacturing is the better choice, compared to injection moulding and extrusion which need different moulds after every design change.

Designing a structure that exists out of many small separate parts, increases replaceability such that only a few parts have to be replaced after a structural failure instead of the whole body. Instead of increasing replaceability, the material used can also be reduced by designing a topology optimised structure. Manufacturing such a design usually requires the use of additive manufacturing.

For the parts of the drone that will probably not change over time, the use of injection moulding and extrusion might prove to be a more sustainable option over additive manufacturing. The creation of the mould requires a relatively large amount of energy but in the long run, this will prove to be more sustainable than the energy use that is required by additive manufacturing.

11.1.3. Lean Manufacturing

Lean manufacturing consists in increasing the value of the product and reducing waste in all phases of production. The waste includes both material waste and time waste [69]. When coming up with the manufacturing and production plan, the team will incorporate this philosophy as much as possible. Time waste can be reduced by decreasing transportation time, where possible, by buying necessary materials and components from local providers. This also touches upon a social aspect, since local companies are supported. Material waste can be reduced by optimising the production of specific parts according to their characteristics. This can be done with, for example additive manufacturing, as explained in Subsection 11.1.2. The value will be added to the product by making it as sustainable as possible as well by meeting other requirements that are provided by the stakeholders.

11.2. Operations

During operations, drones produce a lot of noise and have a relatively high power consumption compared to other electronic inspection systems [20].

Noise pollution is important. However, due to the fact that both indoor and outdoor inspections are conducted in either airports or manufacturing sites and all the personal have access to hearing protection, the noise produced by the drones will not cause as much problems as in an urban environment. The team should still try to reduce noise pollution to keep a workable environment for the people involved.

Power consumption can be optimised in several ways in order to improve the sustainability of the drone system. The batteries can be pre-charged with green energy instead of charging the batteries on-site. If green sources of energy are available on-site, these should be utilised. In the worst-case scenario, regular grid power can be used and the carbon emissions can be accounted for through the European Emissions Trading System [26]. Software for both the ground station and each unique drone can be optimised in order to decrease energy usage. Several design options can be incorporated such as implementing a portable solar panel into the suitcases used for carrying drones, to power up the ground station while operating outdoors if the weather is suitable.

11.3. End of Life Solutions

The EOL of the drone system will be postponed as far as possible by emphasising the repairability and replaceability of the drone system. By replacing or repairing parts, the waste of decommissioning other remaining working parts of the system can be reduced. When this is not possible, the end of life process of the system will have to be considered.

Recycling the batteries is the main challenge for finding EOL solutions. Batteries contain hazardous materials that will cause harm to the environment when they are not disposed of carefully. The metals contained in batteries are also expected to become scarcer. For these reasons, it is of great importance to recycle as many elements as possible that make up a battery.

The batteries used for the "CHIP" drone and "DALE" drone are LiPo batteries. These contain, e.g., Co and Ni which are critical metals for battery manufacturing [70]. First the Toxco method is used to obtain lithium salts from the batteries [42]. The Toxco process can be extended by separating Co and Ni using the SX method and recovering them using galvanostatic and potentiostatic processes. The extracted cobalt appears to be particularly suited for cobaltite in new batteries [48]. These methods are currently not implemented nearly enough but are projected to increase substantially in the coming decades[2] [40].

Apart from the batteries, the drones consist of other electronic parts. All these PCBs will eventually reach their EOL and they have to be disposed of in an environmentally friendly way. Recycling of electronic waste and PCBs is extensively discussed in [41], e.g., state of the art separation processes, pyro- and hydro-metallurgical treatment methods, and future developments.

The propellers for the "CHIP" drone and "DALE" drone are made out of Polycarbonate (PC) and CFRC respectively. PC is a thermoplastic, therefore, it can be recycled relatively easily by melting and then reshaping the material into a new product. CFRC is harder to recycle but active research is being done as discussed in Subsection 11.1.1.

11.4. Economical and Social Considerations

Sustainability also involves economical and social aspects. As mentioned in Subsection 11.1.3, local companies should be supported to reduce time waste while also creating a sustainable economical impact. This can be taken further by using fair trade materials to support healthy working conditions. As was previously mentioned in the project plan [24], using drones may replace some jobs (aircraft inspection), while also creating others (manufacturing and supervision of the drones). It also reduces the hard risk of work, which increases the safety of the workers.

The supply chain for electronic and battery input materials can have problematic working, sustainable or ethical conditions. When outsourced, the assembly of these parts is also not guaranteed to occur in a sustainable or ethical manner. Addressing these problems is outside of the scope of the DSE. For battery manufacturing, conditions may improve due to the scale-up of the market, which is likely to lead to the creation of larger incentives for regulation.

11.5. Effects on Sustainability

Apart from using a sustainable design of the drone system itself, using a swarm of drones also can contribute positively to sustainability in general.

Using a swarm of inspection drones will result in a reduced inspection time. This is expected to result in an increase of aircraft utilisation. Therefore, fewer aircraft have to be operational at the same time. The system can also support sustainable aircraft by offering a superior inspection tool from the start of the aircraft lifetimes. This could result in their flight time increase, which in turn should lead to positive effect on sustainability.

It is assumed that the inspection that is performed will also be more thorough than the current inspection methods, thereby leading to better damage detection and increasing the available data for predictive maintenance. It is expected that this will increase the operational lifetime of the aircraft due to decreased degradation.

[2] Retrieved from: `https://www.fortunebusinessinsights.com/industry-reports/lithium-ion-battery-recycling-market-100244` (cited 29 April 2022)

12 Cost Analysis

This chapter discusses various aspects of system economics. A cost breakdown of company and customer expenses is provided in Section 12.1. An analysis on ROI, break-even, and profit for the company, and customer cost savings is provided in Section 12.2.

12.1. Cost Breakdown

This section provides a breakdown of the cost for both the company and the customer. Subsection 12.1.1 breaks down company cost for design and development, system capital, and labour and overhead cost. Subsection 12.1.2 breaks down yearly costs associated with visual inspection using a swarm of inspection drones, including system lease, operator, overhead, and utilisation.

12.1.1. Company Cost

Company cost can be divided into three constituent elements. The first element, design and development cost, is assumed to be the cost associated with the DSE. The second element is labour and overhead cost, and is assumed to be a fixed yearly amount for labour and labour associated overhead cost. The third element is system cost, i.e. the amount of money it takes to produce one system. Battery replacement costs for one operational year are included to provide an estimate on costs for yearly lease of the system.

Design and Development Cost

It is assumed that full design and development of the system has been performed by 10 students in 10 weeks during the DSE. The work performed by these students is valued at €150 per hour, for a total of 4,000 man hours[1]. This results in a total system design and development cost of €600,000.

Labour and Overhead Cost

It is expected that 10 engineers will be working on the system full-time when the system is commercially available. It is assumed that these engineers get paid a competitive engineering salary of €70,000 per year, and the employer has to spend on average an additional 28% of the salary on employer expenses[2] [3]. This brings the total cost per employee to €90,000, resulting in a total labour cost of €900,000 per year. Office space and utilities cost on average €9,100 per employee, resulting in a total cost of €91,000 per year[4].

Production Cost

The production cost is divided in the needed to purchase the materials, the one of the off-the-shelf components used in the production, and the operational cost of the machinery needed to produce the system. The labour cost would also be part of the production cost, however it has already been computed separately. Assembly cost is also included in labour cost and overhead. The total cost for off-the-shelf components amounted at €139,736 as reported in Table C.1 (assuming three years of batteries). The material on the other hand is comprised of 4 fuses 0.7Kg of PP GF30[5], for a cost of €464 per swarm. For producing the structure, 220 hours of 3D printing are required, at a rate of $90\mathrm{h/kg}$ [6], with an operating cost of $1€/\mathrm{h}$ [7] for a total cost €220. System assembly is performed by the 10 company employees. Combining these cost elements leads to a total production cost per system of €140,000. Additional costs such as shipping[8], maintenance, and EOL are neglected at this stage. It is recommended to study these elements a later stage.

[1] DSE introduction lecture, April 2022

[2] Retrieved from: `https://www.glassdoor.nl/Salaries/amsterdam-engineering-salary-SRCH_IL.0,9_IM1112_KO10,21.htm?countryRedirect=true` (cited 9 June 2022)

[3] Retrieved from: `https://www.ondernemenmetpersoneel.nl/orienteren/personeelskosten/werkgeverslasten-berekenen` (cited 9 June 2022)

[4] Retrieved from: `https://www.annexum.nl/nieuws-uit-de-markt/werkplek-op-kantoor-kost-werkgever-e-9-086-per-persoon-per-jaar/` (cited 10 June 2022)

[5] Retrieved from: `https://www.matterhackers.com/store/l/basf-black-ultrafuse-ppgf-30-polypropylene-glass-fiber-3d-printer-filament-285mm-07kg/sk/M52ZZ2H` (cited 10 June 2022)

[6] Retrieved from: `https://printingit3d.com/how-long-does-1kg-of-3d-printer-filament-last/` (cited 10 June 2022)

[7] Retrieved from: `https://all3dp.com/2/3d-printing-cost-calculator-great-websites/` (cited 10 June 2022)

[8] Retrieved from: `https://www.postnl.nl/tarieven/Pakket/NL/20-23kg` (cited 13 June 2022)

System Lease

System lease cost is built up from labour and overhead cost, asset depreciation, and a 'good' 15% profit margin[9]. Asset depreciation is calculated over a period of three years (the same as for batteries)[10], one system therefore has a annual depreciation of €47,000. The total European commercial fleet consists of around 6000 aircraft (see Subsection 3.1.2), with a total estimated 36,000 inspections required annually. A total of 17 systems would thus be required to fulfil market needs. It is assumed that in the first year two systems are leased. A conservative estimate projects a growth of one additional system lease every year. The component of labour and overhead is calculated as follows; total annual labour and overhead cost is divided by the total number of systems leased. The 15% profit margin is calculated for the sum of the labour and overhead, and depreciation cost. Inflation is assumed at an average of 2.1%[11]. The lease cost for one system over the first five operating years can be seen in Table 12.1 below.

12.1.2. Customer Cost

For the customer cost breakdown, Air France-KLM is taken as an example customer. Air France-KLM has a fleet size of 554 aircraft[12]. Assuming a total of six inspections per aircraft per year (see Subsection 3.1.2) results in a total number of inspections of 3,300. One system can perform up to six inspections per day (see Section 10.6), resulting in a total number of inspections of 2,200 per year. Air France-KLM would therefore require two systems to perform visual inspection for their fleet.

A KLM maintenance technician was found to have a monthly pay up to €4,133 [13]. Each system shall have one operator, resulting in a required two operators for Air France-KLM. Including an additional 28% for employer expenses leads to a total annual cost of €128,000 per year.

Overhead cost will likely decrease for the customer with respect to current manned visual inspection, seeing as the time spent in the hangar will decrease with decreasing inspection time. It is also expected that fleet daily utilisation will increase. An estimate value of overhead and utilisation cost will therefore be provided. Current cost of inspection is estimated at €10,000 per visual inspection[14]. A total of 30 man hours is assumed per inspection, resulting in a total man hour cost of €1,000 (assuming a monthly pay of €4,133 for 160 working hours)[15]. The total cost of overhead and utilisation costs per inspection can thus be estimated at €9,000.

Each aircraft inspection currently requires eight hours. Dividing the total cost of overhead and utilisation cost by eight, results in a cost of €1,125 per hour. The expected inspection time of the swarm of inspection drones including setup and pack-up is one hour. This results in a total cost of overhead and utilisation of €3.71m for a total of 3,300 inspections. Note that system storage and electricity costs were considered negligible and have not been included in this figure.

12.2. ROI, Break-Even Point, and Profit

This section discusses the ROI, break-even point, and profit for the company, as well as the ROI for an example customer. The expression for ROI can be seen in Equation 12.1 below.

$$\text{ROI} = \frac{\text{Net return on investment}}{\text{Total value of investment}} \times 100\% \tag{12.1}$$

Customer Savings

The net return of investment for the customer is assumed equal to the current cost of manned inspection over the period of one year for a Air France-KLM size aircraft fleet operator, minus the cost of the swarm of inspection drones for one year. A current cost estimate for the service model (see Section 12.1) yields a cost of €1.25m for two systems, with a total projected cost for visual inspection of €4.96m (including overhead, missed utilisation, and personnel cost). Current cost of inspection is estimated at €10,000 per visual inspection[16]. This

[9] Retrieved from: https://corporatefinanceinstitute.com/resources/knowledge/accounting/profit-margin/ (cited 13 June 2022)

[10] Retrieved from: https://www.droneblog.com/drone-life-expectancy/ (cited 13 June 2022)

[11] Retrieved from: https://www.abnamro.com/research/en/our-research/ecb-to-try-and-dampen-inflation-expectations-and-euro-decline (cited 13 June 2022)

[12] Retrieved from: https://www.airfranceklm.com/en/group/profile (cited 9 June 2022)

[13] Retrieved from: https://nl.indeed.com/cmp/Klm/salaries/Onderhoudsmonteur-(m-v) (cited 10 June 2022)

[14] Retrieved from: http://www.mrodrone.net (cited 9 June 2022)

[15] Retrieved from: https://mainblades.com/save-costs/ (cited 10 June 2022)

[16] Retrieved from: http://www.mrodrone.net (cited 9 June 2022)

results in an annual estimated cost of €33m for Air France-KLM (3,300 annual visual inspections). The result is an estimated cost saving of €28m for the first operational year. Note that this estimation is heavily reliant on assumptions. Actual values for cost are hard to pinpoint, and number of inspections per year not precisely known[17].

Company ROI, Break-even Point, and Profit

The total value of the investment is assumed equal to the design and development cost, and production cost of two initial systems. This results in a total investment cost of €0.88m. Lease cost per system multiplied by the number of systems leased result in a total company income of €1.25m (see Table 12.1). Therefore, the first year ROI is 142%.

It is assumed that the investment comes in the form of a loan of 10 years, at an interest rate of 9.7%[18]. It is assumed that each year an equal amount is paid back, if possible. Figure 12.1 shows the company balance over time, including the break-even point. The break-even point is taken as the point where all income equals company expenses plus the initial investment. Company expenses are limited to labour and overhead costs, loan pay-off, and interest, whilst income is limited to lease agreements with aircraft fleet operators and MRO's (see Table 12.1). It can be seen that break-even is achieved during the sixth year. Note that significant margin is present for unexpected cost increases.

Table 12.1: Annual lease cost for one system, Cost Per Inspection (CPI), loan pay-off plus interest, and profit over the first five years.[19]

Year	Lease Cost [€]	CPI [€]	Loan + Interest [€]	Operational Profit [%]
1	624,000	284	173,360	0.62
2	443,000	201	164,824	1.95
3	353,000	160	156,288	3.18
4	300,000	136	147,752	4.31
5	265,000	121	139,216	5.36

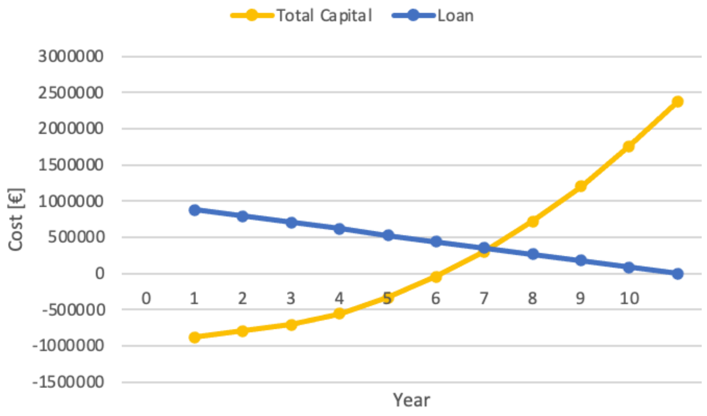


Figure 12.1: Total capital (company savings minus loan) and total loan value over company lifetime [20].

[17] Private communication with KLM, June 2022

[18] Retrieved from: https://www.abnamro.nl/nl/zakelijk/producten/financieren/zakelijke-lening/ (cited 13 June 2022)

[19] See the Excel file "DSE_Group2_Cost.xslx" on the project GitHub page https://github.com/DSE-Swarm-of-Inspection-Drones/Swarm-of-inspection-drones-rebase.

[20] See the Excel file "DSE_Group2_Cost.xslx" on the project GitHub page https://github.com/DSE-Swarm-of-Inspection-Drones/Swarm-of-inspection-drones-rebase.

13 Technical Risk Management and Contingency Analysis

In this chapter, an updated version of the technical risk analysis performed in the midterm report [23] will be presented. Risk is an integral component of any design project, and it is especially relevant when dealing with novel technologies. Risk must be accurately analysed and, when necessary, mitigated to ensure that the project is successful. Technical risk is present in any phase of the project, from the design and the manufacturing, to the actual deployment of the mission. The risk analysis needs to be reassessed in order to be up-to-date with the stage of the current design process.

13.1. Technical SWOT Analysis

Before identifying any risks, it is important to perform a strength, weakness, opportunity and threat (SWOT) analysis. The result of this analysis is presented in Table 13.1. Based on this SWOT analysis, the technical risks will be identified.

Table 13.1: Technical SWOT analysis of a swarm of inspection drones.

	Helpful	**Harmful**
Internal	• The swarm concept allows a modular design • The swarm concept allows for redundancy • The swarm concept allows for shorter (and custom) inspection times • The swarm concept allows for flexible deployment	• The technology is relatively new, increasing development risk • The size of the drones limits dimensions and weight of sensors • The inspection time limits the use of specific Non-Destructive Testing (NDT) methods • Increased risk of collision with aircraft, supervisor, and other drones
External	• Increasing demand for faster and cheaper inspection techniques • The technology is scalable to larger aircraft • There is a clear market gap for this technology	• Disrupted global supply chain might lead to cost increases and scheduling delays • Drone materials are bought-in from a contractor • The inspection method has to adhere to regulations set by aviation authorities

13.2. Technical Risk Identification

The development of a new product is subjected to a number of risks. These risks could have a significant impact on project schedule, costs or quality. For all the risks, a cause, event and consequence are described. All these risks are described in Table 13.2. These risks follow from the (technical) SWOT analysis. With respect to the previous iteration of the risk analysis, produced in the Midterm Report [23], a risk related to the incorrect detection of damages (RT-TECH-16) has been added. Furthermore, more risks related to the development of the system have been drawn. These risks are RT-TECH-15, regarding the expertise of the team, and RT-SCHED-7, regarding the development time. Furthermore, RT-SCHED-8 has been added, due to the selection of the second design concept for the detailed design. Additionally, risk RT-SCHED-9 has been added, which is specific to the smaller drones, due to the stereo vision.

Table 13.2: A description of the technical risks.

Risk ID	Risk title	Cause	Event	Consequence
RT-TECH-1	Collision between drones	The drones are operating in a swarm and are kept apart by an anti-collision software	Two or more drones collide	The drones are critically damaged and cannot proceed with the inspection and it shows that the swarm technology does not work
RT-TECH-2	Collision between drone and aircraft	The drones are operating in a swarm and inspect an aircraft	One or more drones collide with the aircraft	The drones and aircraft are damaged and it shows that swarm technology is not functioning as intended
RT-TECH-3	Collision between drone and supervisor	The drone swarm is supervised by a technician	One or more drones collide with supervisor	The drones are damaged, the supervisor is possibly injured
RT-TECH-4	Uncharged Batteries	The drones' batteries are charged when the drones are inoperative	The charging system fails to correctly charge the drones to their maximum capacity	The drones fail to inspect the entire aircraft
RT-TECH-5	Drones too big	The drones shall fit in 4 suitcases to be carried by 2 operators	The drones are designed to be too large and do not fit in the suitcases	Drones do not have the desired flexibility in deployment and may need special/additional transportation methods
RT-TECH-6	The inspection time is too long	The drones shall carry out the full inspection in 2 hours	The swarm cannot complete the full inspection in the allocated time slot (consistently)	Aircraft are not inspected in time and the client is not satisfied with the product
RT-TECH-7	Insufficient accuracy	The swarm shall detect damages	The swarm cannot detect flaws with the required accuracy	Flaws remain undetected, aircraft operates with these flaws present in the structure
RT-TECH-8	Electronics failure	The drone is powered by an electrical power unit	The system fails to deliver the required power	The drone will stop functioning and crash
RT-TECH-9	Drones too heavy	The drone mass is restricted by the maximum weight of the suitcase	The drone will be heavier than expected	The drone design has to be revised, such that the mass of the drone reduces.
RT-TECH-10	Sensing equipment failure	The drone carries sensors, such as cameras, to conduct inspections	A failure of the sensing equipment of one or more drones	The drones cannot conduct the inspection in the time specified by the client and swarm functioning is unreliable
RT-TECH-11	Lack of redundancy	The system shall be able to perform the whole inspection with one inoperative drone in the allocated time slot	The swarm cannot complete the inspection in the allocated time slot with one drone inoperative	Inspection will be delayed

RT-TECH-12	Excessive vibrations due to the propellers	The drones use propellers to generate lift and move	The vibrations caused by the propellers are excessive	The instruments cannot achieve the required accuracy due to the vibrations and the aircraft flies with beyond acceptable damage
RT-TECH-13	Inaccurate assigned resource budgets	The assigned resource budgets are based on requirements and estimates	Inaccuracies are present in assigning budgets to subsystems	Budgets have to be changed, the drones are heavier than expected
RT-TECH-14	Regulation change	The detection of flaws has some requirements governed by regulations	Airworthiness authorities change regulations on aircraft inspection	Sensors carried by the drone have to be changed such that they adhere to the new regulations
RT-TECH-15	Lack of expertise on technology	The development team has little expertise on drones and inspection techniques	The development team cannot come up with relevant design options	The developed design is unfeasible or ineffective
RT-TECH-16	Algorithm cannot correctly identify damages	The development of the algorithm for the identification of damages is complex	The identification algorithm does not work correctly	Damages are not identified, or pristine parts are recognised are damaged
RT-TECH-17	Operators cannot adapt to new system	Adapting to a new system and procedure is not straightforward	The operators cannot correctly used the system	The inspection is not carried out correctly
RT-SCHED-1	Unfinished technology	Swarm of drones is a new concept	The technology to let the drones operate in a swarm is not ready yet	The deployment of this new inspection method is delayed, possible claims from the client
RT-SCHED-2	Testing	Verification and validation tests must be performed before the product enters the market	The drone swarm does not pass verification and validation tests	The drone swarm design must be changed, which takes additional time
RT-SCHED-3	Lack of funds	The drone swarm is financed by the project owner and/or investors	The investors withdraw their financial support from the design project	The design project cannot continue and will be cancelled
RT-SCHED-4	Lack of support	The drone swarm design project is performed based on a request from the project owner	The priorities of the project owner change and the project is not aligned with their business objectives	Support for this specific design project is cancelled, the project has to be changed or cancelled
RT-SCHED-5	Supply delays	The drones are manufactured from bought-in materials	The material supplier cannot deliver required materials at the desired time	Manufacturing of the drones is delayed, such that the drones cannot enter the market at the specified time

RT-SCHED-6	Defective component in supply chain	The drones are manufactured from bought-in components	The components delivered are flawed	The manufacturing is delayed and the drones cannot be delivered in time
RT-SCHED-7	Incorrectly calculated development time	The development time is constrained by the DSE timeline	The development time is estimated incorrectly	There is no time to properly develop and design the system
RT-SCHED-8	Multiple airframe design	Multiple drone airframe designs have to be developed	The design of these airframes takes more time than allocated	The detailed design cannot be made during the time allocated for the DSE
RT-SCHED-9	Stereo vision unproven	The smaller drones use stereo vision for navigation	Stereo vision is an unproven positioning technology for aircraft inspection	Problems arise when implementing stereo vision on drones, which need to be solved
RT-COST-1	Expensive technology	The drone swarm concept is relatively new	The technology will be more costly than expected	Price of the drone swarm increases, such that inspection using the drone swarm method is more expensive than human visual inspection
RT-COST-2	Expensive materials	The drones shall be made out of 70% recyclable materials	Recyclable materials are more expensive than expected	Price of the drone swarm increases, such that inspection using the drone swarm method is more expensive

13.3. Technical Risk Assessment

Now that all the risks have been identified, a likelihood and impact rating must be assigned to each risk. By multiplying these two, a risk rating can be obtained. Both ratings are on a 1-5 scale, in which 1 means very low and 5 means very high. The likelihood of the risk has been defined according to an estimated Probability of Occurrence (PoO) [32]:

1. Very low: PoO< 1%
2. Low: 1% ≤PoO< 30%
3. Moderate: 30% ≤PoO< 50%
4. High: 50% ≤PoO< 70%
5. Very high: PoO ≥70%

The impact ratings were defined according to the estimated impact on the quality of the product, costs of the inspection method, schedule and customer satisfaction. For example, a failing anti-collision software could result in damaged drones or aircraft and injured people. Furthermore, a collision indicates that the swarm technology does not work properly. Such an impact was decided to be quite severe and these risks were awarded a risk rating of 5. Additionally, RT-SCHED-3 and RT-SCHED-4 also have significance as the project will be cancelled or changed. Such impacts were also awarded an impact rating of 5. Furthermore, the consequences of risk on customer satisfaction and quality were always judged as quite severe and therefore they were awarded a risk rating of 3 or 4. From thisTable 13.3, it can be observed that most risks with an impact on schedule have been awarded a high impact rating. These impact ratings are based on customer satisfaction: if the project is delayed, then the client will not be satisfied and possibly leading to claims from the client. Impacts which could be regarded as an inconvenience, such as one defective product in the supply chain have been awarded an impact rating of 2. In Table 13.3, the risk ratings are presented.

Table 13.3: Technical risk rating of the swarm of inspection drones.

Risk ID	Likelihood rating (1-5)	Impact rating (1-5)	Risk rating	Risk driver
RT-TECH-1	1	5	5	Anti-collision software
RT-TECH-2	1	5	5	Anti-collision software
RT-TECH-3	1	5	5	Anti-collision software
RT-TECH-4	2	4	8	Drone charging system
RT-TECH-5	3	3	9	Drone size
RT-TECH-6	3	4	12	Inspection time
RT-TECH-7	2	5	10	Detection accuracy
RT-TECH-8	1	4	4	Drone power unit
RT-TECH-9	4	3	12	Drone mass
RT-TECH-10	1	3	3	Drone sensors
RT-TECH-11	2	3	6	Drone redundancy
RT-TECH-12	2	4	8	Propeller vibrations
RT-TECH-13	4	3	12	Budget estimate
RT-TECH-14	2	2	4	Regulations
RT-TECH-15	4	5	20	Team expertise
RT-TECH-16	2	4	8	Technology readiness
RT-TECH-17	1	5	5	Operators training
RT-SCHED-1	3	4	12	Drone swarm concept
RT-SCHED-2	4	3	12	Drone testing
RT-SCHED-3	1	5	5	Project funds
RT-SCHED-4	1	5	5	Project support
RT-SCHED-5	5	3	15	Material supply
RT-SCHED-6	2	2	4	Material supply
RT-SCHED-7	2	4	8	Development time
RT-SCHED-8	5	2	10	Design option selection
RT-SCHED-9	4	4	16	Stereo vision positioning
RT-COST-1	3	4	12	Drone swarm concept
RT-COST-2	5	3	15	Drone materials

These risks can be plotted in a risk map, which is done in Table 13.5. In this figure, the cells are green, yellow or red. These indicate the acceptable, medium and unacceptable risks, respectively. For the unacceptable risks, a mitigation strategy should be defined, in order to reduce either the impact or the likelihood of a risk. These mitigation strategies are described in Section 13.4.

13.4. Technical Risk Mitigation Strategies

For the unacceptable risks, a risk mitigation strategy will be defined. There are four risk mitigation strategies available, namely:

- Remove the risk: the PoO is reduced to zero and the risk poses no longer a threat.
- Reduce the risk: either the PoO or the impact of the risk is reduced.
- Accept the risk: it is accepted that this risk event can occur. It should be monitored whether this risk is triggered. If this risk is triggered, then a plan should be ready on how to handle this risk.
- Transfer the risk: the risk is placed on a third party.

For the design of a new product, it is difficult to remove or transfer a risk. Therefore, the risk is either reduced or accepted. The risk mitigation strategies are described in the following table:

Table 13.4: Risk mitigation strategies for unacceptable Technical risks.

Risk ID	Risk rating	Mitigation strategy	Mitigation strategy description
RT-TECH-6	12	Reduce	The target inspection time will include contingencies so that the final inspection time is within the requirements
RT-TECH-7	10	Reduce	Using well-established inspection technology with well-known accuracy will allow reducing the likelihood of an insufficient inspection accuracy
RT-TECH-9	12	Reduce	Using contingencies at the early stage of the design will allow reducing the likelihood of the drones being too heavy.
RT-TECH-13	12	Reduce	Using the necessary contingencies it will be possible to plan for the correct resource budgets in order to fulfil all the requirements.
RT-TECH-15	20	Reduce	Ask for consultations with experts in the relevant fields within and outside of the university.
RT-SCHED-1	12	Accept	The group have not had much influence on the development of the technology, so the risk that the technology is not ready must be accepted. Furthermore, the drone swarm concept is required by the client and is the selling point of this project. Thus, this technology is of high importance.
RT-SCHED-2	12	Accept	It is acceptable that the design will initially fail some verification & validation tests, and the schedule takes into account the time to correct the design in order to comply with these tests.
RT-SCHED-5	15	Accept	The current disruptions to the global supply chain mainly given by political tensions and the effects of the COVID-19 pandemic will impact the manufacturing stage in ways that cannot be controlled by the design team. Furthermore, the drone swarm concept must be implemented, as it is dictated by requirements and the market analysis. It will be monitored whether this risk event is triggered. If this risk event does occur, then a plan will be in place to continue the product development. For example using a backup contractor.
RT-SCHED-8	10	Reduce	The impact of the risk is reduced due to the experience gained by the team members during the design of one drone. When designing a second airframe, the team members are already familiar with the design process.
RT-SCHED-9	16	Accept	Employing the “DALE” drones for aircraft inspection is deemed to be one of the best options, as both the top and the bottom of the fuselage can be inspected. However, if the deployment of the stereo cameras appears to be problematic during detailed design, then these sensors will be replaced by LIDAR sensors or the “DALE” drones will be removed from the design option.
RT-COST-1	12	Reduce	The use of appropriate contingencies will make it more likely that the final cost of the system is within an acceptable range.
RT-COST-2	15	Accept	Just like for the supply chain delay, a current global shortage of materials over which the team has no control will make fluctuations of the materials’ price the norm in this project. However, a plan will be in place in order to continue project development. Such a plan could be similar to the plan reported in the mitigation strategy of RT-SCHED-5.

With these mitigation strategies in place, the risk map is updated. The risk manager will be mainly responsible for managing these risks. The risk manager will be assisted by the chief engineer. The chief engineer has to monitor whether all the assigned budgets to subsystems are adhered to. The communications officer will also be assisting the risk manager by managing relations with the subcontractors. By managing these relations, the communications officer makes sure that the subcontractor delivers the materials as agreed upon by the contract.

Table 13.5: Risk map for the swarm of inspection drones, unmitigated risks.

Impact	5	RT-TECH-1 RT-TECH-2 RT-TECH-3 RT-TECH-17 RT-SCHED-3 RT-SCHED-4	RT-TECH-7		RT-TECH-15	
	4	RT-TECH-8	RT-TECH-4 RT-TECH-12 RT-TECH-16 RT-SCHED-7	RT-TECH-6 RT-SCHED-1 RT-COST-1	RT-SCHED-9	
	3	RT-TECH-10	RT-TECH-11	RT-TECH-5	RT-TECH-13 RT-SCHED-2	RT-SCHED-5 RT-COST-2
	2		RT-TECH-14 RT-SCHED-6			RT-SCHED-8
	1					
		1	2	3	4	5
		Likelihood				

The unmitigated risk map and the updated risk map are presented respectively in Table 13.5 and Table 13.6. The accepted risks would especially have an impact on the development and production of the system and are always given by external factors that cannot be mitigated.

13.5. Contingency Analysis

In this subsection, the mass contingencies for the individual drones' components at different design stages are reported. The contingencies were estimated using the guidelines present in the project management/systems engineering (PM/SE) lecture notes [33]. However, some higher contingencies were planned for the initial stages of the design. This decision was taken as a risk mitigation measure, given the relative novelty of the technologies used. The contingencies are reported in Table 13.7. Once the design matures, then the contingencies will decrease as more details of different subsystems are known. The contingencies determined in the previous stages of the design [22] have been followed and confirmed for the following stages.

Table 13.7: Mass contingency for the individual drones with design stage on the lines and subsystem on the columns.

	Contingencies [%]				
	Structures	Propulsion system	Sensing interface	Batteries	Control system
Conceptual Estimate	25	30	25	30	25
Layout Calculation	20	25	20	25	15
Pre-released Drawings	5	10	10	10	5
Released Drawings	3	5	5	5	3
Specifications	5	5	5	5	5
Actual measurements qualification hardware	1	1	1	1	1
Actual measurements flight hardware	0	0	0	0	0

Table 13.6: Risk map for the swarm of inspection drones, mitigated risks.

Impact	5	RT-TECH-1 RT-TECH-2 RT-TECH-3 RT-TECH-7 RT-TECH-17 RT-SCHED-3 RT-SCHED-4				
	4	RT-TECH-8	RT-TECH-4 RT-TECH-6 RT-TECH-12 RT-TECH-16 RT-SCHED-7 RT-COST-1			
	3	RT-TECH-10	RT-TECH-11 RT-TECH-13	RT-TECH-5 RT-SCHED-1 RT-TECH-15		
	2		RT-TECH-14 RT-SCHED-6		RT-SCHED-2 RT-SCHED-9	
	1					RT-SCHED-5 RT-SCHED-8 RT-COST-2
		1	2	3	4	5
		Likelihood				

14 Reliability, Availability, Maintainability and Safety Analysis

An important aspect of design is RAMS. System effectiveness is determined by those elements, and whether the design can be successful. Each element is closely related to the other. The reliability will be described in Section 14.1, then the availability is described in Section 14.2. In Section 14.3, the maintainability of the drones is discussed. Finally, in Section 14.4, some aspects with regarding safety are discussed.

14.1. Reliability

Reliability is probability of the system carrying the mission successfully. The most probable failures of the system can be seen in the Figure 14.1.

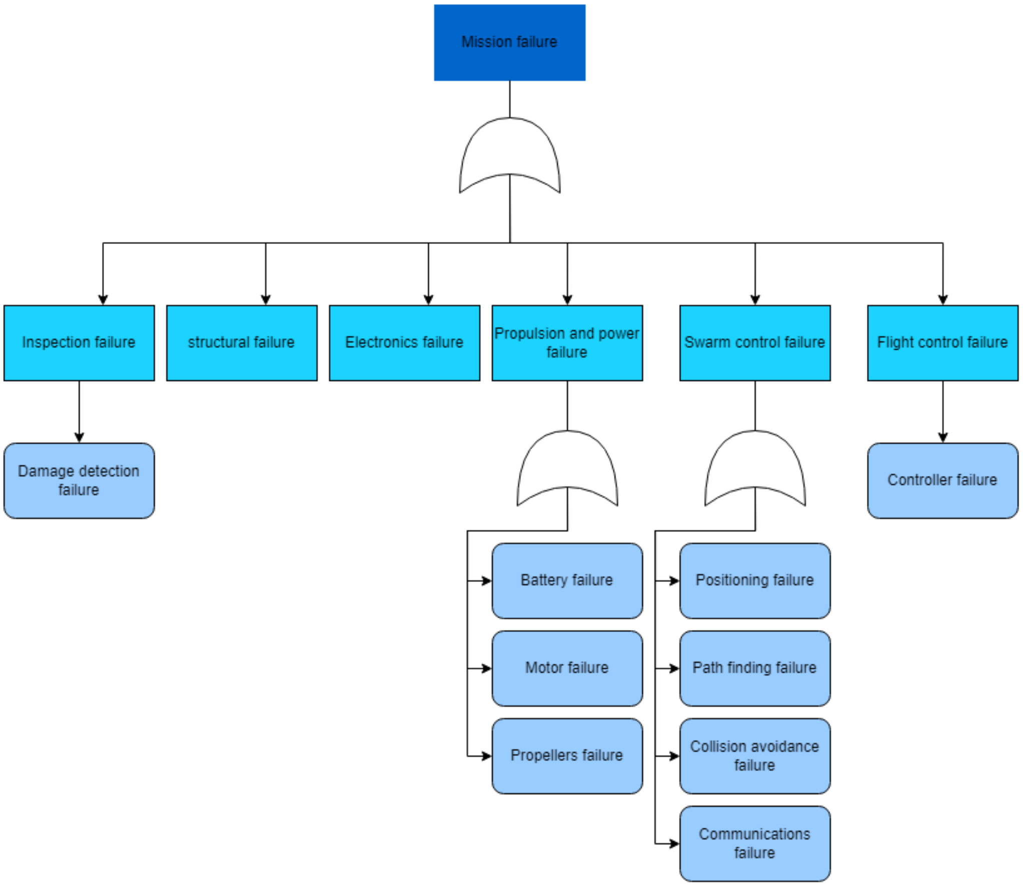


Figure 14.1: System failure tree

Inspection Failure

Starting with the damage detection failure.Computer vision based on CNN architecture works best for detecting damages in the pictures (see Subsection 4.4.2). An accuracy range of 96.5-98% has previously been reached for crack recognition while using CNN in [60]. This is the best estimate for damage detection accuracy so far, and it will be used for reliability calculations later on.

Structural failure

The probability that the structure will fail is assumed to be negligible. The maximum stress the drone can experience is $8.8\,\mathrm{MPa}$, accounting for safety factor of 2. If the structure is 3D printed in the correct direction it can endure stress up to $95\,\mathrm{MPa}$. Even if the structure is printed in the wrong direction, it will only fail at a stress of $19\,\mathrm{MPa}$. Hence the probability of structural failure is deemed to be extremely low.

Electronics Failure

In Table 14.1 the most common drone electronic failures can be seen. The total failure rate is estimated to be around $6.382 \times 10^{-5}\,\mathrm{h}^{-1}$ for a single drone. It is assumed each drone has a same failure rate. The system shall be able to carry out an inspection, defined from take-off until landing, within 30 minutes. Hence the probability that electrical subsystem for 1 drone will fail during the inspection is $P_{e,fail} = 6.382 \times 10^{-5} \cdot 0.5 = 3.191 \times 10^{-5}$.

Table 14.1: Most common electronic systems failure rates based on NPRD-2016 database[1].

Electronics failure	Mean failure rate ($\times 10^{-6}$)
Gyroscope breakdown	12.43
Accelerometer malfunction	3.504
Overcurrent/under-current	37.11
Short-circuit	4.813
High voltage	1.545
Sensors failure	4.418
Total failure rate	63.82

Table 14.2: Most common power and propulsion systems failure rates based on NPRD-2016 database[2].

Power and propulsion failure	Mean failure rate ($\times 10^{-6}$)
Battery physical damage	3.134
Overheating of the battery	44.42
Battery exhaustion	2.891
Motor failure	3.305
No power for cooling	2.891
Total failure rate	56.39

Power and Propulsion failure

Main contributors to power and propulsion system failure are the battery and the motor. Propellers failure is difficult to estimate and it is assumed that this failure happens only if the drone itself crashes. Again using the NPRD-2016 dataset, the total failure rate can be estimated for a single drone. The probability that a single drone will fail during the inspection due to power and propulsion failure is $P_{pp,fail} = 5.639 \times 10^{-5} \cdot 0.5 = 2.8195 \times 10^{-5}$.

Swarm Control and Flight Control Failure

These subsystems are combined into a single category because they share similar failure features. Looking at NPRD-2016 database, the most common failures can be seen in Table 14.3. The swarm and flight control failure rate of a single drone is $2.33215 \cdot 10^{-5}$ and the probability that a single drone will fail because of that during the inspection is $P_{ctrl\ fail} = 2.332 \times 10^{-5} \cdot 0.5 = 1.166 \times 10^{-5}$

Table 14.3: Most common swarm and flight control systems failure rates based on NPRD-2016 database[3].

Power and propulsion failure	Mean failure rate ($\times 1 \times 10^{-6}$)
Inaccurate sensor readings	0.2981
Aerial map inaccuracy	1.567
Signal loss	2.891
Synchronisation error	3.3054
RF interference	15.26
Total failure rate	23.3215

[1] retrieved from: https://www.quanterion.com/product/publications/reliability-online-automated-databook-system-roads-all-databooks-nprd-eprd-fmd-subscription/ (cited 13 June 2022)

[2] retrieved from: https://www.quanterion.com/product/publications/reliability-online-automated-databook-system-roads-all-databooks-nprd-eprd-fmd-subscription/ (cited 13 June 2022)

[3] retrieved from: https://www.quanterion.com/product/publications/reliability-online-automated-databook-system-roads-all-databooks-nprd-eprd-fmd-subscription/ (cited 13 June 2022)

Overall, a single drone can experience structural or electrical or power and propulsion or swarm and flight control failure. The probability of that is $P_{fail} = P_{e\ fail} + P_{pp\ fail} + P_{ctrl\ fail} = 3.191 \cdot 10^{-5} + 2.8195 \cdot 10^{-5} + 1.166075 \cdot 10^{-5} = 7.176575 \cdot 10^{-5}$. Taking into account that there are 5 drones in the swarm, but with 1 “CHIP” drone and 1 “DALE” drone redundancy, the system fails only when 2 “CHIP” drone fails ($P^2_{fail}P^3_{no\ fail}$) or 2 “DALE” drone fails ($P^2_{fail}P^3_{no\ fail}$) or 3 “DALE” drone fails ($P^3_{fail}P^2_{no\ fail}$). $P_{no\ fail} = 1 - P_{fail}$ and ”or” acts as a plus sign. Hence the probability that the system will fail during inspection is equal to $1.03 \cdot 10^{-8}$. It is expected that 2200 inspections will be done per year, hence the probability of experiencing at least 1 failure during a year is $2.266 \cdot 10^{-5}$. In conclusion, the probability that the swarm can fail during inspection is extremely low, and the driving factor in carrying fully successful inspection is the damage detection accuracy. As mentioned before, it is expected to be in a range of 96.5-98%. If the system would not have redundancy, and would fail if only a single drone fails, then the probability of single system failure per year would be 0.158 or 15.8 %. This is significantly more compared to the system with redundancy.

14.2. Availability

Availability is defined as the probability that an item will operate satisfactorily in an ideal environment at a given point in time when used under stated conditions. Availability formula is given as Mean Time to Failure (MTTF) / (MTTF + MTTR) [4]. Mean Time to Failure (MTTF) is calculated by looking how many inspections it takes to complete to finally arrive at failure probability of 1. It takes $\frac{1}{1.03 \cdot 10^{-8}} = 9.7 \cdot 10^7$ inspections or $\frac{9.7 \cdot 10^7}{2200} = 44090$ years. Mean Time to Repair (MTTR) is assumed to be the battary recharge period, which is 1 hour. Using the formula mentioned before, the system is available virtually all every time. If the system would not have redundancy implemented, MTTF would be equal to $\frac{1}{7.176575 \cdot 10^{-5}} = 1.4 \cdot 10^4$ inspections or $\frac{1.4 \cdot 10^4}{2200} = 6.33$ years. MTTR stays the same. The availability this time is 99.9982 %. Hence, implementing redundancy not only increases availability but also reliability.

14.3. Maintainability

Maintenance of drones prevents accidents, lowers costs, and reduces liability risks. Product maintainability refers to its ease of maintenance. For both drones, the payload is modular, meaning the camera or a LIDAR can be easily replaced if needed. One of the main component that needs frequent maintenance is the batteries. The process of maintaining the batteries is design to be as easy as possible. The batteries can be inserted, and removed very quickly not needing any external tools from each drone. It need to be noted that LiPo batteries can only last maximum of 200 cycles, hence after 200 cycles it is crucial to replace the batteries. Usually LiPo batteries should be charged at 1 C rate, and discharged below 3 V. If it is needed to access the motor, it also can be easily done by removing the propellers by simply using the screw. This makes it easy to replace damaged propellers as well. Also the protective casing can be removed as well. For the “DALE” drone it can be done by hand, while for the “CHIP” drone it can be done by using the screw again. Cleaning can also be done quite easily because of the drones designs. Both drones structures are open, so it is not difficult to reach most of the drone places.

14.4. Safety

In order to execute the inspection in the safest manner possible, there are several safety features implemented into the design. In order to assist in the prevention of any bad results caused by an impact between a drone and plane, both types of drones are equipped with safety fairings around their propellers. Safety software is also created to prevent collisions from happening. For the flight and swarm control, a PID controller, collision avoidance, positioning, task allocation, and trajectory generation algorithms are implemented in such a way that it complies with safety requirements. A brushless motor is used because it eliminates a single point of failure, and does not generates sparks as compared to the brushed motor. Besides that, before every inspection it is important to check each battery level, and make sure it is enough for a full aircraft inspection. If the battery level is not checked, it could lead to a drone running out of power and falling on the aircraft or personnel, which is one of the worst case scenarios.

[4] Retrieved from: `https://web.archive.org/web/20140413164657/https://dap.dau.mil/glossary/Pages/2045.aspx` (cited 14 June 2022)

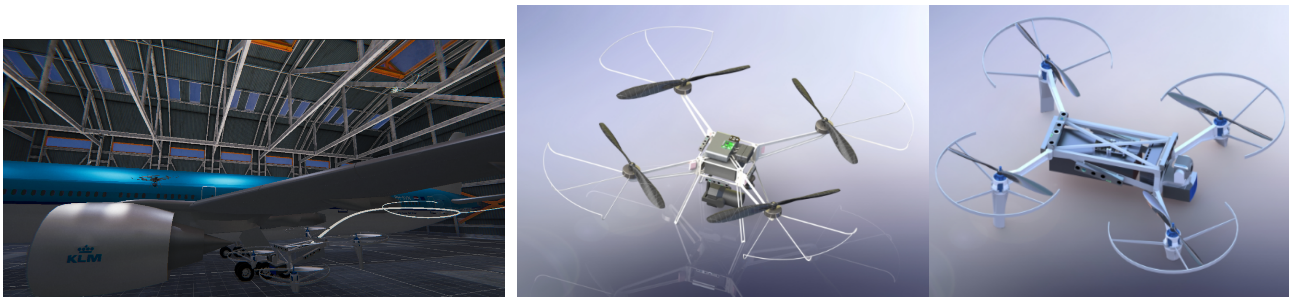
15

Final Drone Design and Requirements Compliance

Having designed all of the subsystems for the two drones, the final specifications of the whole inspection system can be presented.

15.1. CHIP and DALE

The inspection system will host 2 “CHIP” drones and 3 “DALE” drones. It will be able to inspect a narrow-body aircraft in approximately 16 minutes, with the “CHIP” drones and “DALE” drones inspecting the upper and lower surfaces respectively. The inspection routine involves using drones to capture visual images of the entire aircraft surface, and using a ground station to build a 3D model of the inspected aircraft with the captured images tagged onto the surface of this 3D model. Each drone carries the stereo cameras for collision avoidance, and a combination of stereo + LIDAR for localisation. Figure 15.1 shows some example pictures of the inspection drones.



(a) CHIP & DALE inspecting

(b) CHIP and DALE

Figure 15.1: Pictures of the inspection drones

Additionally, the mass and power budgets of the drones are presented in Table 15.1 and Table 15.2. In both mass budgets, the cables are assumed to be 5% of the MTOM [7]. For the EVO Nano camera, the uncertainty about the weight is 10%, as described in Section 4.2. Therefore, this is added as a contingency to the mass of the inspection camera. Furthermore, Figure 15.2 and Figure 15.3 offer a pie-chart of the mass fractions of both drones, expressed in percentages. The ‘Electronics’ part in the pie-chart entails the USB hub, antenna, voltage regulators, the FC, and the PDB.

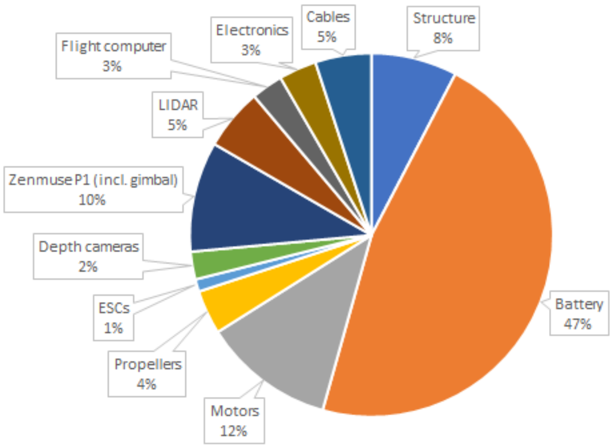


Figure 15.2: The mass fractions of the “CHIP” drone.

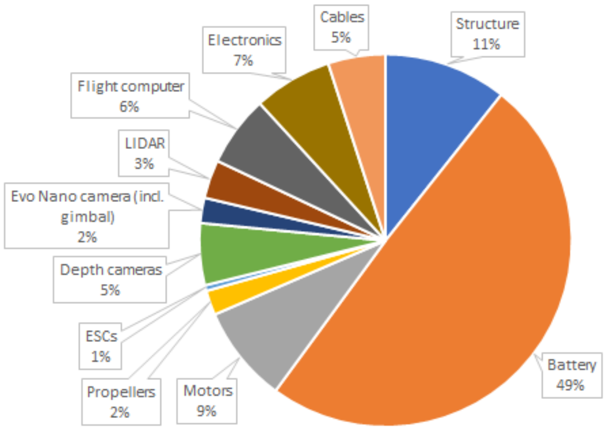


Figure 15.3: The mass fractions of the “DALE” drone.

Table 15.1: Mass, power, repairability, and replaceablity of the "CHIP" drone.

	Amount	**Mass** [g]	**Contingency**[%]]	**Mass** [g] **incl. contingency**	**Power** [W]	**Replaceable**	**Repairable**
Structure	1	624	3	643	-	✓	✓
Battery	1	3,810	5	4,000	-	✓	✗
Motors (hover-maximum)	4	956	5	1,000	872-2,256	✓	
Propellers	4	320	5	336	-	✓	✗
ESCs	4	90	5	95	12	✓	✗
Depth cameras	6	201	5	212	21	✓	✗
Zenmuse P1 (incl. gimbal)	1	800	5	840	20	✓	✗
LIDAR	1	447	5	470	23	✓	✗
Flight computer	1	230	3	237	25	✓	✗
USB hub	1	224	3	231	0.31	✓	✗
Antenna	1	2	3	2	0.35	✓	✗
Voltage regulators	1	20	3	21	12.4	✓	✗
FC	1	7	3	8	0.79	✓	✗
PDB	1	21	3	22	0.35	✓	✗
Cables	-	408	3	420	-	✓	✓
MTOM	-	8,160	-	8,540	-	-	-

Table 15.2: Mass, power, repairability, and replaceablity of the "DALE" drone.

	Amount	**Mass** [g]	**Contingency** [%]	**Mass** [g] **incl. contingency**	**Power** [W]	**Replaceable**	**Repairable**
Structure	1	400	3	412	-	✓	✓
Battery	1	1,850	5	1,943	-	✓	✗
Motors (hover-maximum)	4	316	5	332	492-1,472	✓	✗
Propellers	4	80	5	84	-	✓	✗
ESC (incl. PDB)	1	13	5	14	6.35	✓	✗
Depth cameras	6	201	5	211	21	✓	✗
EVO Nano Camera (incl. gimbal)	1	82	10	90	16	✓	✗
LIDAR	1	125	5	131	6	✓	✗
Flight computer	1	230	3	237	25	✓	✗
USB hub	1	238	3	245	0.62	✓	✗
Antenna	1	2	3	2	0.35	✓	✗
Voltage regulators	1	9	3	10	4.7	✓	✗
FC	1	7	3	8	0.79	✓	✗
Cables	-	187	3	193	-	✓	✓
MTOM	-	3,740	-	3,912	-	-	-

15.2. Requirement Compliance

The stakeholder and system requirements which were posited for this DSE project will need to be reviewed, to check for how well the designed swarm drones comply with these requirements. To this end, all the stakeholder requirements are stated in Table 15.3, along with their compliance status as well as some comments if need be. The compliance status are split into 3 levels, complied, partly complied, and not complied. A partly complied requirement is a requirement that is thought to be achievable from literature studies or from research done, but it is not directly verifiable.

Table 15.3: Stakeholder and System requirements.

Identifier	Requirement	Compliance	Comments
Stakeholder Requirements			
StRS-AR-01	The system shall have a minimum additional influence on its environment in terms of noise and obstructions	Partly Complied	See Section 7.6.
StRS-FO-01	The system shall have a risk of collision lower than one in a thousand inspections	Complied	See Subsection 5.3.2.
StRS-FO-02	The system shall be able to finish an inspection in the nominal time in case of a single drone malfunction	Complied	See Section 5.4
StRS-FO-03	The system shall cost less than €48 per aircraft flight hour, per aircraft for a fleet operator	Complied	See chapter 12.
StRS-MO-01	The system shall perform a full inspection on a single-aisle narrow-body aircraft	Complied	See chapter 4 and Section 5.4
StRS-MO-02	The system shall consist of a swarm of inspection drones	Complied	See Section 5.4
StRS-MO-03	The system shall operate fully autonomously, after being deployed by a human operator	Complied	See chapter 5.
StRS-MO-04	The system shall be deployed and monitored by one person with minimal (drone specific) training	Partly Complied	Carrying the suitcases would take at least 2 people, but setting up the drone and monitoring inspection only requires 1 person.
StRS-MO-05	The system shall perform the inspection in less than 2 hours	Complied	See Section 5.4 and Figure 4.4.2
StRS-MO-06	The system shall perform the specified inspection types at a similar or better level than the current industry standard for drone inspection	Complied	Current standard is being able to visually inspect and detect damages on the surface of aircrafts.
StRS-MO-07	The system shall be carried in up to 4 suitcases (103x67.5x37.5cm), i.e. to be easily transportable by 2 persons	Complied	See Subsection 8.5.1
StRS-MO-08	The swarm of drones shall have a probability of detection equal to at least 90% for structural damages of diameter >10mm	Partly Complied	See chapter 4
StRS-MO-09	The systems' structures shall be at least 70% recyclable	Complied	See Subsection 8.6.2
StRS-MO-10	The systems' components shall be at least 80% repairable or easy-to-replace	Complied	See Subsection 8.6.1
StRS-MO-11	End-of-life solutions for the drones and their batteries should be provided	Complied	See Section 11.3 and Subsection 8.6.2.
StRS-MO-12	Inspections using the swarm of drones shall be cheaper than current human visual inspections, including depreciation costs from the drones and batteries	Complied	See Section 12.2.
StRS-MO-13	The individual drones shall have the capability of changing the sensor module during the inspection	Complied	See chapter 8.
StRS-MO-14	The system shall take a maximum of 10 weeks to reach its final design	Complied	Spirit of the requirement is met, final design meaning DSE final design.
StRS-MO-15	The system shall have a cost per inspection lower than €287.5	Complied	See chapter 12

Table 15.3: Stakeholder and System requirements.

Identifier	Requirement	Compliance	Comments
StRS-RA-01	The system shall be operable both indoors and outdoors, with wind speeds up to 5 m/s and visibility conditions of more than 100m	Complied	The PID controller has been tested for winds of up to 5 m/s, and the simulation assumes 100 m visibility for the drones. See Subsection 6.3.2 and simulation in chapter 5
StRS-RA-02	The system shall be safely deployable in all non-movement ground phases (i.e. at the hangar, gate, storage)	Complied	
System Requirements			
SYS-COS-01	The system shall have a cost per inspection lower than €287.5	Complied	See chapter 12
SYS-COS-02	The system shall be operable by 1 person	Complied	
SYS-COS-03	The system shall cost less than €48 per flight hour, per aircraft for a fleet operator	Complied	See chapter 12
SYS-PER-01	The system shall be able to inspect the fuselage of a single-aisle narrow-body aircraft	Complied	See Section 5.4 and chapter 4
SYS-PER-02	The system shall be able to inspect the main wing of a single-aisle narrow-body aircraft	Complied	See Section 5.4 and chapter 4
SYS-PER-03	The system shall be able to inspect the empennage of a single-aisle narrow-body aircraft	Complied	See Section 5.4 and chapter 4
SYS-PER-04	The system shall operate autonomously after take-off	Complied	See chapter 5
SYS-PER-05	The sensor modules shall be swappable between inspections	Complied	See Subsection 8.6.1
SYS-PER-06	The system shall be operable by a person with basic knowledge of the drone software	Complied	
SYS-PER-07	The system shall be able to operate nominally in wind speeds up to 5 m/s	Complied	See Subsection 6.3.2
SYS-PER-08	The system shall be able to operate nominally with visibility of 100 m or more	Complied	See Section 5.4
SYS-REL-01	An inspection drone shall have a maximum of 0.1% failure rate per inspection	Partly Complied	See chapter 4
SYS-REL-02	The system shall operate nominally between -10°C and 50° C	Complied	
SYS-REL-03	The system shall be able to finish an inspection within the time limit specified in SYS-TIM-04 with 1 drone inoperative	Complied	Demonstrated in the simulation, see Section 5.4
SYS-REL-04	The system shall be able to detect defects of size minimum 10 mm with a probability of 90%	Partly Complied	See chapter 4
SYS-REL-05	The system shall have a maximum false-positive rate of 10% for external defects of minimum 10 mm	Partly Complied	See chapter 4
SYS-REL-06	An end-of-life solution for the batteries shall be provided	Complied	See Section 11.3
SYS-REL-07	The system shall have a probability of collision with the airplane of less than 0.1% per inspection	Complied	See Subsection 6.3.2
SYS-REL-08	The system shall be able to detect defects of size 1-10 mm with a probability of 80%	Partly Complied	See chapter 4
SYS-REL-09	The system shall have a maximum false-positive rate of 20% for defects of size 1-10 mm	Partly Complied	See chapter 4

Table 15.3: Stakeholder and System requirements.

Identifier	Requirement	Compliance	Comments
SYS-SAF-01	The system shall not come closer than 0.5 meters to the inspected aircraft	Complied	See chapter 4
SYS-SAF-02	The system shall at all time keep at least 2 meters distance from personnel	Complied	
SYS-SAF-03	Each drone shall have safety fairings	Complied	See Subsection 8.5.7
SYS-SAF-04	The distance between each inspection drone during the inspection shall be at least 1 meter	Complied	Demonstrated in the simulation, see Section 5.4
SYS-SAF-05	The system shall not come closer than 2 meters to the objects that have not yet been covered in SYS-SAF-1 to SYS-SAF-4, except for the ground	Complied	Demonstrated in the simulation, see Section 5.4
SYS-SAF-06	The system shall produce a maximum noise level of 90 dB measured at 1 meter	Not Complied	See Section 7.6
SYS-SIZ-01	The system shall fit in at most 4 suitcases each with a dimension of 103x67.5x37.5cm	Complied	See Subsection 8.5.1
SYS-SIZ-02	The mass of one filled suitcase shall not exceed 12 kg	Complied	See Subsection 8.5.7
SYS-SIZ-03	The system shall consist of at least three drones	Complied	The system has 5 drones
SYS-SUS-01	The systems' structures shall be at least 70% recyclable	Complied	See Subsection 8.6.2
SYS-SUS-02	The structure shall not contain any materials that have a direct threat to human health at decommissioning	Complied	Various sources show that PP-GF30 is non toxic[1] [2].
SYS-SUS-03	At end of life decommission, the waste shall not exceed 15% of structural mass	Complied	100% of the structure can be recycled.
SYS-SUS-04	The system shall use rechargeable batteries	Complied	See chapter 7.
SYS-SUS-05	During the production of structural parts, waste material shall not exceed 20% of the used raw material	Not Complied	See Section 8.7
SYS-SUS-06	The system shall use interchangeable batteries	Complied	See chapter 8
SYS-SUS-07	At least 80% of system component count shall be repairable	Not Complied	Most of the COTS are not repairable, however, they are easy-to-replace in accordance with StRS-MO-10
SYS-SUS-08	An end-of-life solution for the batteries shall be provided	Complied	Section 11.3 and Subsection 8.6.2.
SYS-TIM-01	The high-level design of the system, phase 3, shall take a maximum of 2 weeks.	Complied	The midterm report [23] was handed in on time.
SYS-TIM-02	The detailed design of the system, phase 4, shall take a maximum of 6 weeks.	Complied	
SYS-TIM-03	System set-up, the time to get the system ready for deployment, shall take a maximum of 15 minutes	Complied	See chapter 10.
SYS-TIM-04	The system shall be able to carry out an inspection within 30 minutes, timed from take-off until landing and excluding system set-up and pack-up time.	Complied	See chapter 5.
SYS-TIM-05	The system shall take up to 15 minutes for processing inspection data after the landing	Complied	See Section 4.4.
SYS-TIM-06	The system shall be able to be packed up by 1 person within 15 minutes	Complied	See Figure 4.4.2.

16 Conclusion and Future Work

16.1. Conclusion

The aim of this report was to present a swarm of visual aircraft inspection drones that reduces aircraft downtime, infrastructure required, and man hours spent. The proposed design is capable of inspecting a narrow-body and wide-body aircraft in 16 and 30 minutes respectively. Use of infrastructure is significantly reduced, and has the potential to be eliminated completely. Man hours spent per inspection are reduced from thirty hours to just one, and hangar time is decreased from eight hours to less than one. As a result, annual visual inspection costs for a company with the size of Air France-KLM will be reduced with 85% to just €5.1m. A Hardware-as-a-Service business model will be adopted, leading to durable relationships without the need for large hardware investments on the side of the customer.

Market analysis has shown that there is significant opportunity for a swarm of inspection drones. Efficiency increases for single drone inspection are severely limited beyond the current benchmarks. A swarm of inspection drones is highly scalable, and capable of improved in-hangar, and novel at-gate inspection. The latter two resulting in reduced need for inspection infrastructure, increased inspection cadence, and increased aircraft utilisation. Indirect effects are the improved data analytics, health monitoring, and adoption of novel strategies such as predictive maintenance.

The swarm consists of two "CHIP" drones and three "DALE" drones (see Figure 16.1), inspecting the upper and lower aircraft surface respectively. The "CHIP" drone uses a Zenmuse P1 optical camera for inspection, and is equipped with one Ouster OS0 $360°$ Field of View (FOV) LIDAR, and six Intel® RealSense^{TM} Depth Module D450 stereo cameras for LI/I-SLAM. The "DALE" drone uses a EVO Nano+ optical camera for inspection, and is equipped with a narrow FOV LIDAR, and six Intel® RealSense^{TM} Depth Module D450 stereo cameras for a combination of LI/I-SLAM and VI-SLAM.

A genetic algorithm will be used for inspection task allocation, after which the A* algorithm is used for path generation. Collision avoidance will be performed using Optimal Reciprocal Collision Avoidance (ORCA). The resulting desired state is fed into a Proportional-Integral-Derivative (PID) controller, providing motor commands to reach the desired state. Adequate stability and performance has been demonstrated in both wind and no-wind conditions. The control system allows for stable hover at the desired inspection locations, ensuring sufficient image quality. Inspection data is subsequently stored on onboard SD cards. It is recommended to use a Convolutional Neural Network (CNN) approach for off-board damage detection.

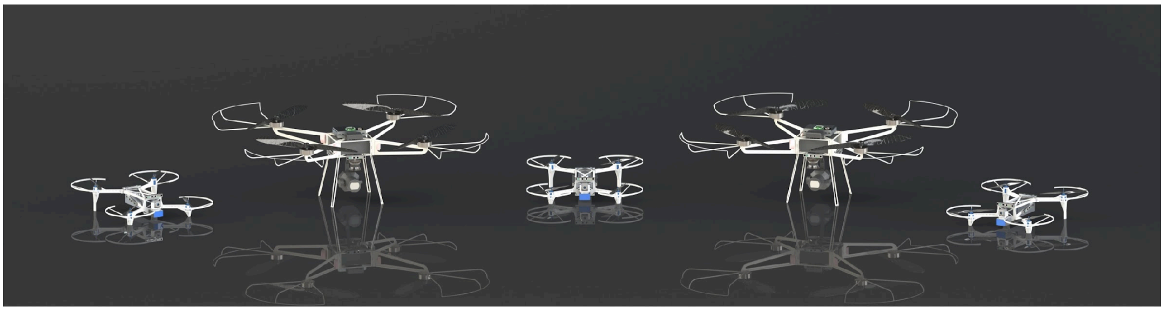


Figure 16.1: Side-by-side render of the mixed swarm of inspection drones.

The total masses of the "CHIP" drone and "DALE" drone are $7.80\,\mathrm{kg}$ and $3.25\,\mathrm{kg}$ respectively. Structures for both drone types have been topology optimised, and have safety fairings for damage prevention. Structural behaviour under expected loading cases has been simulated in Solidworks, and modal frequency analysis was performed. A full component list for electrical, propulsion and power, and sensing systems has been provided. Furthermore, full electrical systems layouts have been designed for both drone types. Production requires 3D printing of the structures, acquisition of Commercial-Off-The-Shelf (COTS) parts, and parts assembly.

16.2. Future Work

The design and development logic (see Figure 16.2) provides an outline of the project continuation after the final DSE hand-in. The project Gantt chart (see Figure A.1) provides a timeline for these activities. Phase 1 consists of finalising detailed design, and defining subsystem, qualification, and acceptance tests. Main considerations are implementation of additional design elements that could not be implemented due to time constraints and/or insufficient knowledge. A selection is provided in the paragraphs below.

The selection of the camera was based on the total time it would take for one drone to inspect the full aircraft exterior. It is recommended to further investigate inspection motion, i.e. alternative beyond constant and stopping motion. Furthermore, an algorithm shall be developed to automate the process of damage detection. The algorithm shall be divided into different segments, to account for the different types of damages and aircraft segments. As there is not enough data publicly available for algorithm training, it is proposed to use data augmentation to speed up the training process. The collected data shall be processed and stored for traceability, such that evolution of damages can be monitored. The location, size, and type of damages can be visualised on a 3D model of the aircraft (a so-called "digital twin"), giving the operator an intuitive feel of the vulnerable areas of the aircraft.

It is recommended to validate the positioning system with physical hardware and integrated algorithms. This is expected to decrease uncertainty related to swarm positioning, and related inspection quality. The current configuration of five drones (two "CHIP" drone and three "DALE" drone) is believed to be optimal considering current design constraints and requirements. It is recommended to investigate the feasibility of using "DALE" drones for tasks that are currently performed by the "CHIP" drone. A decrease in the amount of "CHIP" drones is expected to lead to a decrease in system cost, though performance of the "CHIP" drone is unrivalled in locations where reference points are sparse.

Custom design of components such as the Flight Controller (FC) is expected to result in a decrease in power consumption, and mass, and shall be further investigated. Heat analysis of the electronics shall be performed to validate correct functioning under operating conditions. Reduction of the environmental impact of the swarm is recommended. For this purpose a detailed environmental impact analysis shall be performed, after which factors with significant impact shall be mitigated. For example, the design has significant environmental impact in terms of noise level, it is recommended to perform further research on the use of low-noise propellers.

An Adaptive Incremental Nonlinear Dynamic Inversion (A-INDI) controller implementation for state-of-the-art wind disturbance rejection, and fault tolerance was briefly pursued. It was soon found out that such an implementation would likely not be realisable in the time allotted for the DSE. Implementation of an A-INDI controller is recommended, and is most likely to succeed under supervision of an expert at TU Delft.

Methods for contact inspection (laser depth, thermography) were investigated in-depth in the midterm report. These methods were deemed unfeasible for drone inspection considering limited onboard weight and power resources. Automation of contact inspection would however offer significant benefits for the customer. It it thus recommended to investigate adding GBVs for (automated) contact inspection to the swarm (see XYREC[1]).

Phase 2 consists of all steps necessary in producing a prototype of the system. System assembly is preceded by components manufacturing and testing, as well as ordering of parts. If deemed of sufficient quality, software-hardware integration can take place, otherwise iteration might be necessary. The full function prototype shall subsequently be validated in Phase 3. This validation shall be performed through previously defined subsystem, qualification, and acceptance testing. This validation is crucial in ensuring a smooth transition to Phase 4.

Phase 4 start with initiating a collaboration with an Original Equipment Manufacturer (OEM) and/or a Maintenance, Repair, and Overhaul (MRO) operator. System flaws, and competences can be demonstrated, ways towards certification can become more clear. Certification can then be achieved in three ways; through regulatory agencies, aircraft fleet operators, and OEMs. Phase 5 considers production, quality assurance, and shipping of the product. This phase partially overlaps with Phase 6, operations. With the chosen business model, customer service, support, and feedback is pivotal in the early days of the company. Lastly, Phase 7 considers end-of-life.

[1] Retrieved from: `https://www.xyrec.com/technology/` (cited 7 June 2022)

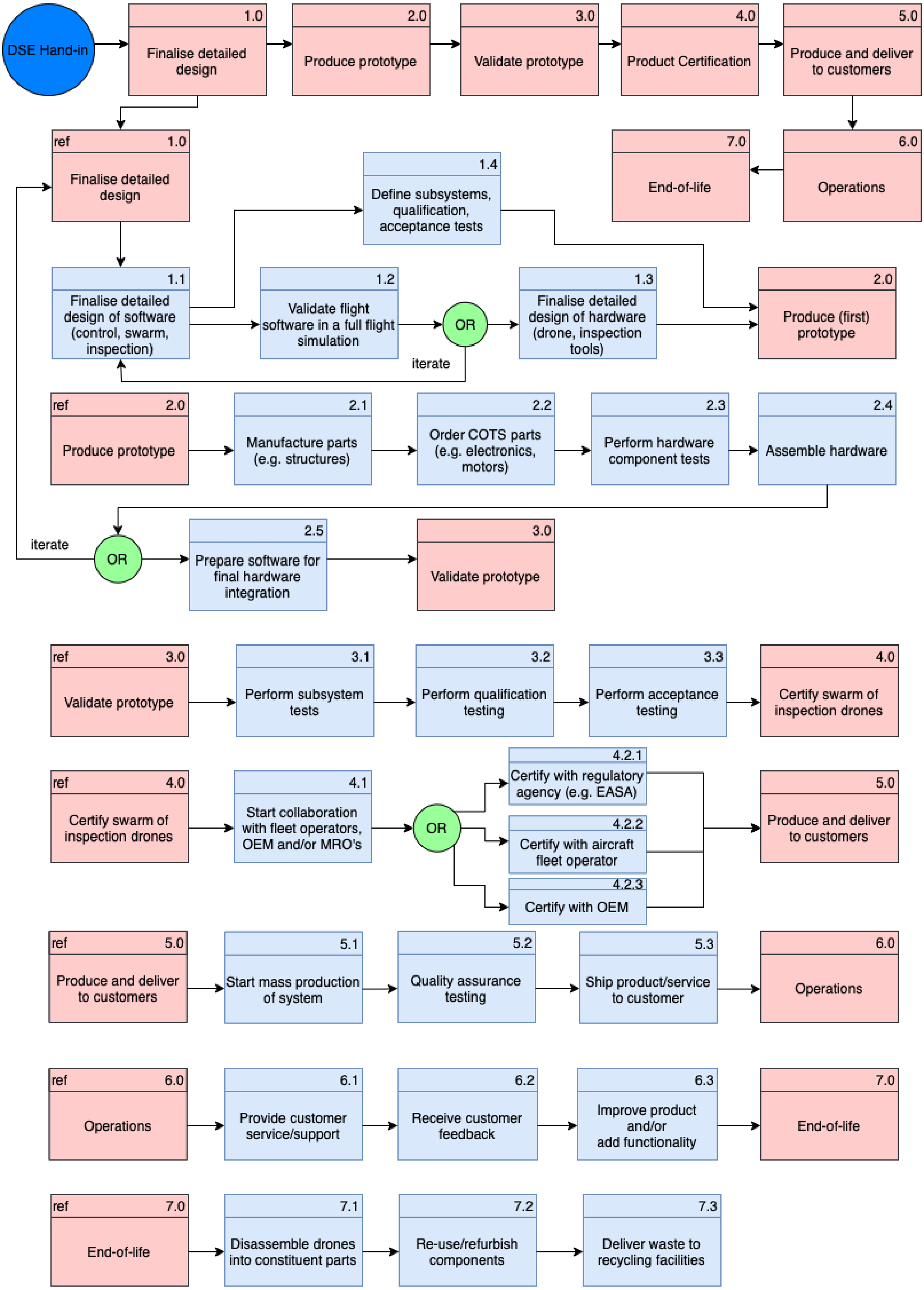

DSE Hand-in
1.0 Finalise detailed design
2.0 Produce prototype
3.0 Validate prototype
4.0 Product Certification
5.0 Produce and deliver to customers
6.0 Operations
7.0 End-of-life
ref 1.0 Finalise detailed design
1.4 Define subsystems, qualification, acceptance tests
1.1 Finalise detailed design of software (control, swarm, inspection)
1.2 Validate flight software in a full flight simulation
OR
iterate
1.3 Finalise detailed design of hardware (drone, inspection tools)
2.0 Produce (first) prototype
ref 2.0 Produce prototype
2.1 Manufacture parts (e.g. structures)
2.2 Order COTS parts (e.g. electronics, motors)
2.3 Perform hardware component tests
2.4 Assemble hardware
iterate
OR
2.5 Prepare software for final hardware integration
3.0 Validate prototype
ref 3.0 Validate prototype
3.1 Perform subsystem tests
3.2 Perform qualification testing
3.3 Perform acceptance testing
4.0 Certify swarm of inspection drones
ref 4.0 Certify swarm of inspection drones
4.1 Start collaboration with fleet operators, OEM and/or MRO's
OR
4.2.1 Certify with regulatory agency (e.g. EASA)
4.2.2 Certify with aircraft fleet operator
4.2.3 Certify with OEM
5.0 Produce and deliver to customers
ref 5.0 Produce and deliver to customers
5.1 Start mass production of system
5.2 Quality assurance testing
5.3 Ship product/service to customer
6.0 Operations
ref 6.0 Operations
6.1 Provide customer service/support
6.2 Receive customer feedback
6.3 Improve product and/or add functionality
7.0 End-of-life
ref 7.0 End-of-life
7.1 Disassemble drones into constituent parts
7.2 Re-use/refurbish components
7.3 Deliver waste to recycling facilities

Figure 16.2: Project design and development logic starting at the hand-in of the DSE final report.

Bibliography

[1] R. Abdallah. “Reliability approaches in networked systems : Application on Unmanned Aerial Vehicles”. PhD thesis. May 2019.

[2] D. Achilias et al. “Chemical recycling of plastic wastes made from polyethylene (LDPE and HDPE) and polypropylene (PP)”. In: *Journal of Hazardous Materials* 149.3 (2007). Pollution Prevention and Restoration of the Environment, pp. 536–542. ISSN: 0304-3894. DOI: `https://doi.org/10.1016/j.jhazmat.2007.06.076`.

[3] M. F. Ahmed, M. N. Zafar, and J. C. Mohanta. “Modeling and Analysis of Quadcopter F450 Frame”. In: *2020 International Conference on Contemporary Computing and Applications (IC3A)*. 2020, pp. 196–201. DOI: `10.1109/IC3A48958.2020.233296`.

[4] M. Ashby, H. Shercliff, and D. Cebon. *Materials: Engineering, Science, Processing and Design*. Vol. 3. Butterworth-Heinemann, 2014.

[5] T. Bektas. “The multiple traveling salesman problem: an overview of formulations and solution procedures”. In: *Omega* 34.3 (2006), pp. 209–219. ISSN: 0305-0483. DOI: `https://doi.org/10.1016/j.omega.2004.10.004`.

[6] J. van den Berg et al. “Reciprocal n-Body Collision Avoidance”. In: *Robotics Research*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011.

[7] D. Bershadsky, S. Haviland, and E. Johnson. “Electric Multirotor UAV Propulsion System Sizing for Performance Prediction and Design Optimization”. In: *57th AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*. Jan. 2016. DOI: `10.2514/6.2016-0581`.

[8] C. J. Bett. “9 - Gain-Scheduled Controllers”. In: *The Electrical Engineering Handbook* PP (2005), pp. 1107–1114. DOI: `10.1016/B978-012170960-0/50086-4`.

[9] M. Biczyski et al. “Multirotor Sizing Methodology with Flight Time Estimation”. In: *Journal of Advanced Transportation* 2020 (Jan. 2020), pp. 1–14. DOI: `10.1155/2020/9689604`.

[10] J. Brandt et al. *UIUC Propeller Database, Vols 1-3*. University of Illinois at Urbana-Champaign, downloaded 17 May 2022.

[11] M. Brown. “Chapter 1 - An Introduction to the Linear Regulator”. In: *Power Sources and Supplies*. Ed. by M. Brown. Burlington: Newnes, 2008, pp. 1–12. ISBN: 978-0-7506-8626-6. DOI: `https://doi.org/10.1016/B978-0-7506-8626-6.00001-6`.

[12] C. Campos et al. “ORB-SLAM3: An Accurate Open-Source Library for Visual, Visual-Inertial and Multi-Map SLAM”. In: *CoRR* abs/2007.11898 (2020). arXiv: `2007.11898`.

[13] A. Castaldo. *Switching regulator fundamentals*. Feb. 2019.

[14] A. Cervone. *Lecture 10: Electrical Power Systems II*. AE2230 Propulsion & Power, Faculty of Aerospace Engineering, Delft University of Technology, Delft. (Accessed 2 Jun., 2022). Feb. 2022.

[15] S. Chattopadhyay, M. Mitra, and S. Sengupta. “Clarke and Park Transform”. In: *Electric Power Quality*. Dordrecht: Springer Netherlands, 2011, pp. 89–96. ISBN: 978-94-007-0635-4. DOI: `10.1007/978-94-007-0635-4_12`.

[16] O. Cheikhrouhou and I. Khoufi. “A comprehensive survey on the Multiple Traveling Salesman Problem: Applications, approaches and taxonomy”. In: *Computer Science Review* 40 (2021). DOI: `10.1016/j.cosrev.2021.100369`.

[17] G. Colucci et al. “Effect of recycling on polypropylene composites reinforced with glass fibres”. In: *Journal of Thermoplastic Composite Materials* 30.5 (2017), pp. 707–723. DOI: `10.1177/0892705715610407`.

[18] P. Conroy et al. *3-D Reciprocal Collision Avoidance on Physical Quadrotor Helicopters with On-Board Sensing for Relative Positioning*. 2014. DOI: `10.48550/ARXIV.1411.3794`.

[19] I. Cvišić et al. “SOFT-SLAM: Computationally efficient stereo visual simultaneous localization and mapping for autonomous unmanned aerial vehicles”. In: *Journal of Field Robotics* 35.4 (2018), pp. 578–595. DOI: `https://doi.org/10.1002/rob.21762`. eprint: `https://onlinelibrary.wiley.com/doi/pdf/10.1002/rob.21762`.

[20] T. Dbouk and D. Drikakis. “Quadcopter drones swarm aeroacoustics”. In: *Physics of Fluids* 33.5 (2021), p. 057112. DOI: `10.1063/5.0052505`.

[21] S. Derammelaere et al. “A quantitative comparison between BLDC, PMSM, brushed DC and stepping motor technologies”. In: *2016 19th International Conference on Electrical Machines and Systems (ICEMS)*. 2016, pp. 1–5.

[22] "DSE Group 2 Baseline Report: A Swarm Of Inspection Drones". In: (May 2022). Faculty of Aerospace Engineering, Delft University of Technology, Delft.

[23] "DSE Group 2 Midterm Report: Swarm of drones for aircraft maintenance inspection." In: (May 2022). Faculty of Aerospace Engineering, Delft University of Technology, Delft.

[24] "DSE Group 2 Project Plan: Swarm of drones for aircraft maintenance inspection." In: (Apr. 2022). Faculty of Aerospace Engineering, Delft University of Technology, Delft.

[25] "Electroacoustics - Sound level meters - Part 1: Specifications". In: (2013).

[26] D. Ellerman and B. Buchner. "The European Union Emissions Trading Scheme: Origins, Allocation, and Early Results". In: *Review of Environmental Economics and Policy* 1 (Dec. 2007), pp. 66–87. DOI: 10.1093/reep/rem003.

[27] M. Etcheverry and S. E. Barbosa. "Glass fiber reinforced polypropylene mechanical properties enhancement by adhesion improvement". In: *Materials* 5.12 (2012), pp. 1084–1113. DOI: 10.3390/ma5061084.

[28] E. C. C. (within the European Conference of Postal and T. A. (CEPT). *THE EUROPEAN TABLE OF FREQUENCY ALLOCATIONS AND APPLICATIONS IN THE FREQUENCY RANGE 8.3 kHz to 3000 GHz*. 2021.

[29] T. Evens et al. "The Influence of Mechanical Recycling on Properties in Injection Molding of Fiber-Reinforced Polypropylene". In: *International Polymer Processing* 34.4 (2019), pp. 398–407. DOI: doi:10.3139/217.3770.

[30] M. Gatti. "Complete Preliminary Design Methodology for Electric Multirotor". In: *Journal of Aerospace Engineering* 30 (May 2017). DOI: 10.1061/(ASCE)AS.1943-5525.0000752#sthash.E2LOdL86.dpuf.

[31] J. V. V. Gerwen et al. "Indoor Drone Positioning: Accuracy and Cost Trade-Off for Sensor Fusion". In: *IEEE Transactions on Vehicular Technology* 71 (2022), pp. 961–974.

[32] E. Gill. *Risk Management & Concurrent Engineering*. AE3211-I Systems Engineering & Aerospace Design, Faculty of Aerospace Engineering, Delft University of Technology, Delft. (Accessed 20 Apr., 2022). Feb. 2022.

[33] R. Hamann and M. van Tooren. "Systems Engineering and Technical Management Techniques. Part I. Lecture Notes". In: (Sept. 2006). Faculty of Aerospace Engineering, Delft University of Technology, Delft.

[34] W. Hönig et al. "Conflict-Based Search with Optimal Task Assignment". In: *17th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2018)*. 2018, p. 9.

[35] S. Huang, R. S. H. Teo, and K. K. Tan. "Collision avoidance of multi unmanned aerial vehicles: A review". In: *Annual Reviews in Control* 48 (2019), pp. 147–164. ISSN: 1367-5788. DOI: https://doi.org/10.1016/j.arcontrol.2019.10.001.

[36] "IEEE Standard for Information technology—Telecommunications and information exchange between systems Local and metropolitan area networks—Specific requirements - Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications". In: *IEEE Std 802.11-2016 (Revision of IEEE Std 802.11-2012)* (2016), pp. 1–3534. DOI: 10.1109/IEEESTD.2016.7786995.

[37] E. T. S. Institute. *ETSI ES 202 131: Specification of Reference Receiver Performance Parameters for Spectrum Planning*. 2003.

[38] E. T. S. Institute. *t ETSI EN 300 328: Harmonised Standard covering the essential requirements of article 3.2 of the Directive 2014/53/EU*. 2016.

[39] D. James, Z. Shiyu, and M. Lyudmila. "Velocity Obstacle Approaches for Multi-Agent Collision Avoidance". In: *Unmanned Systems* 07 (Jan. 2019). DOI: 10.1142/S2301385019400065.

[40] Z. A. Kader, A. Marshall, and J. Kennedy. "A review on sustainable recycling technologies for lithium-ion batteries". In: *Emergent Materials* 4.3 (June 2021), pp. 725–735. ISSN: 2522-574X. DOI: 10.1007/s42247-021-00201-w.

[41] M. Kaya. *Electronic Waste and Printed Circuit Board Recycling Technologies*. Oct. 2019. ISBN: 978-3-030-26593-9. DOI: 10.1007/978-3-030-26593-9.

[42] P. Keller and C. Anderson. "The Production of Critical Materials as By Products". In: 2 (Aug. 2018), p. 14. DOI: 10.31031/AMMS.2018.02.000532.

[43] J. Kim, A.-R. Kim, and S.-W. Lee. "Artificial Neural Network-Based Automated Crack Detection and Analysis for the Inspection of Concrete Structures". In: *Applied Sciences* 10 (Nov. 2020), p. 8105. DOI: 10.3390/app10228105.

[44] A. Király and J. Abonyi. “Redesign of the supply of mobile mechanics based on a novel genetic optimization algorithm using Google Maps API”. In: *Engineering Applications of Artificial Intelligence* 38 (Feb. 2015). DOI: 10.1016/j.engappai.2014.10.015.

[45] P. Kitjacharoenchaia et al. “Multiple traveling salesman problem with drones: Mathematical model and heuristic approach”. In: *Computers & Industrial Engineering* 119 (2019). DOI: 10.1016/j.cie.2019.01.020.

[46] I. Konovalenko et al. “Recognition of Scratches and Abrasions on Metal Surfaces Using a Classifier Based on a Convolutional Neural Network”. In: *Metals* 11 (Mar. 2021), p. 549. DOI: 10.3390/met11040549.

[47] D. W. Kurtz and E. J. Marte. *A review of aerodynamic noise from propellers, rotors, and lift fans*. National Aeronautics and Space Administration. Jet Propulsion Laboratory, California Institute of Technology, Jan. 1970.

[48] C. Lupi, M. Pasquali, and A. Dell’Era. “Nickel and cobalt recycling from lithium-ion batteries by electrochemical processes”. In: *Waste Management* 25.2 (2005). Special Issue - VARIREI 2003, pp. 215–220. ISSN: 0956-053X. DOI: https://doi.org/10.1016/j.wasman.2004.12.012.

[49] T. Ma et al. “Research and Application of Visual SLAM Based on Embedded GPU”. In: June 2021, pp. 3–21. ISBN: 978-3-030-77568-1. DOI: 10.1007/978-3-030-77569-8_1.

[50] L. Mejias et al. “Embedded Computation Architectures for Autonomy in Unmanned Aircraft Systems (UAS)”. In: *Sensors* 21.4 (2021). DOI: 10.3390/s21041115.

[51] A. Merzlyakov and S. Macenski. “A Comparison of Modern General-Purpose Visual SLAM Approaches”. In: *CoRR* abs/2107.07589 (2021). arXiv: 2107.07589.

[52] Z. Ning, R. Wlezien, and H. Hu. “An Experimental Study on Small UAV Propellers with Serrated Trailing Edges”. In: June 2017. DOI: 10.2514/6.2017-3813.

[53] M. O’Connell et al. “Neural-Fly enables rapid learning for agile flight in strong winds”. In: *Science Robotics* 7 (May 2022). DOI: 10.1126/scirobotics.abm6597.

[54] J. Orsulic et al. “Flying with Cartographer: Adapting the Cartographer 3D Graph SLAM Stack for UAV Navigation”. In: *CoRR* abs/2106.02535 (2021). arXiv: 2106.02535.

[55] E. Pakdel et al. “Recent progress in recycling carbon fibre reinforced composites and dry carbon fibre wastes”. In: *Resources, Conservation and Recycling* 166 (2021), p. 105340. ISSN: 0921-3449. DOI: 10.1016/j.resconrec.2020.105340.

[56] I. Palinkas et al. “Analysis and Optimization of UAV Frame Design for Manufacturing from Thermoplastic Materials on FDM 3D Printer”. In: *Materiale Plastice* 58.4 (2021), pp. 238–249. ISSN: 2668-8220. DOI: 10.37358/MP.21.4.5549.

[57] *Product Sheet - Propeller 26x8.7*. Mejzlík Propellers s.r.o. Brno - Židenice, Czech Republic.

[58] D. Prokhorov et al. “Measuring robustness of Visual SLAM”. In: *2019 16th International Conference on Machine Vision Applications (MVA)*. 2019, pp. 1–6. DOI: 10.23919/MVA.2019.8758020.

[59] Q. Quan. *Introduction to Multicopter Design and Control*. Singapore: Springer, 2017.

[60] A. Rao et al. “Vision-based automated crack detection using convolutional neural networks for condition assessment of infrastructure”. In: *Structural Health Monitoring* 20 (Nov. 2020), pp. 1–19. DOI: 10.1177/1475921720965445.

[61] D. Robert and S. Michael. “Static Testing of Micro Propellers”. In: (Aug. 2008). DOI: 10.2514/6.2008-6246.

[62] B. Schäffer et al. “Drone Noise Emission Characteristics and Noise Effects on Humans—A Systematic Review”. In: *International Journal of Environmental Research and Public Health* 18.11 (2021). ISSN: 1660-4601. DOI: 10.3390/ijerph18115940.

[63] L. Sevcik and M. Vozňák. “Adaptive Reservation of Network Resources According to Video Classification Scenes”. In: *Sensors* 21 (Mar. 2021), p. 1949. DOI: 10.3390/s21061949.

[64] G. Sharon et al. “Conflict-based search for optimal multi-agent pathfinding”. In: *Artificial Intelligence* 219 (2015), pp. 40–66. ISSN: 0004-3702. DOI: https://doi.org/10.1016/j.artint.2014.11.006.

[65] E. Smeur, Q. Chu, and G. de Croon. “Adaptive Incremental Nonlinear Dynamic Inversion for Attitude Control of Micro Air Vehicles”. In: *Journal of Guidance, Control, and Dynamics: devoted to the technology of dynamics and control, 39 (3), 450-461* (2016). DOI: https://doi.org/10.2514/1.G001490.

[66] E. Soria, F. Schiano, and D. Floreano. “Predictive control of aerial swarms in cluttered environments”. In: *Nature Machine Intelligence* 3.6 (June 2021), pp. 545–554. DOI: 10.1038/s42256-021-00341-y.

[67] C. Spezia. *Transfer Functions of DC Motors*. ET 438a Automatic Control Systems Technology, Electrical Engineering Technology, Southern Illinois University, Carbondale. (Accessed 1 Jun., 2022). Oct. 2015.

[68] S. Sun et al. "Incremental Nonlinear Fault-Tolerant Control of a Quadrotor With Complete Loss of Two Opposing Rotors". In: *IEEE Transactions on Robotics* PP (Aug. 2020), pp. 1–15. DOI: `10.1109/TRO.2020.3010626`.

[69] R. Sundar, A. Balaji, and R. S. Kumar. "A Review on Lean Manufacturing Implementation Techniques". In: *Procedia Engineering* 97 (2014). "12th Global Congress on Manufacturing and Management" GCMM - 2014, pp. 1875–1885. ISSN: 1877-7058. DOI: `https://doi.org/10.1016/j.proeng.2014.12.341`.

[70] M. Takano, S. Asano, and M. Goto. "Recovery of nickel, cobalt and rare-earth elements from spent nickel–metal-hydride battery: Laboratory tests and pilot trials". In: *Hydrometallurgy* 209 (2022), p. 105826. ISSN: 0304-386X. DOI: `https://doi.org/10.1016/j.hydromet.2022.105826`.

[71] N. van de Werken et al. "Investigating the effects of fiber surface treatment and alignment on mechanical properties of recycled carbon fiber composites". In: *Composites Part A: Applied Science and Manufacturing* 119 (2019), pp. 38–47. ISSN: 1359-835X. DOI: `10.1016/j.compositesa.2019.01.012`.

[72] C. de Visser. "AE2235-I : Aerospace Systems & Control Theory - Dynamics Analysis". In: Faculty of Aerospace Engineering, Delft University of Technology, Delft. 2021.

[73] C. de Visser and E. van Kampen. "AE-4311 Non-Linear And Adaptive Control". In: Faculty of Aerospace Engineering, Delft University of Technology, Delft. 2022.

[74] N. A. Vu, D. K. Dang, and T. Le Dinh. "Electric propulsion system sizing methodology for an agriculture multicopter". In: *Aerospace Science and Technology* 90 (2019), pp. 314–326. ISSN: 1270-9638. DOI: `https://doi.org/10.1016/j.ast.2019.04.044`.

[75] D. Wang et al. "Research on SLAM of Unmanned Platform Based on the Fusion of Lidar and Depth Camera". In: June 2021, pp. 87–91. DOI: `10.1109/WSAI51899.2021.9486369`.

[76] Y. Xia, Z. Dong, and X. Guo. "Towards a circular metal additive manufacturing through recycling of materials: A mini review". In: *Journal of Central South University* 27 (May 2020), pp. 1134–1145. DOI: `10.1007/s11771-020-4354-6`.

[77] J. Zhenga et al. "Research on Laser-Visual Fusion-based Simultaneous Localization and Mapping". In: *Journal of Physics: Conference Series* 1682 (Nov. 2020), p. 012049. DOI: `10.1088/1742-6596/1682/1/012049`.

A
Diagrams

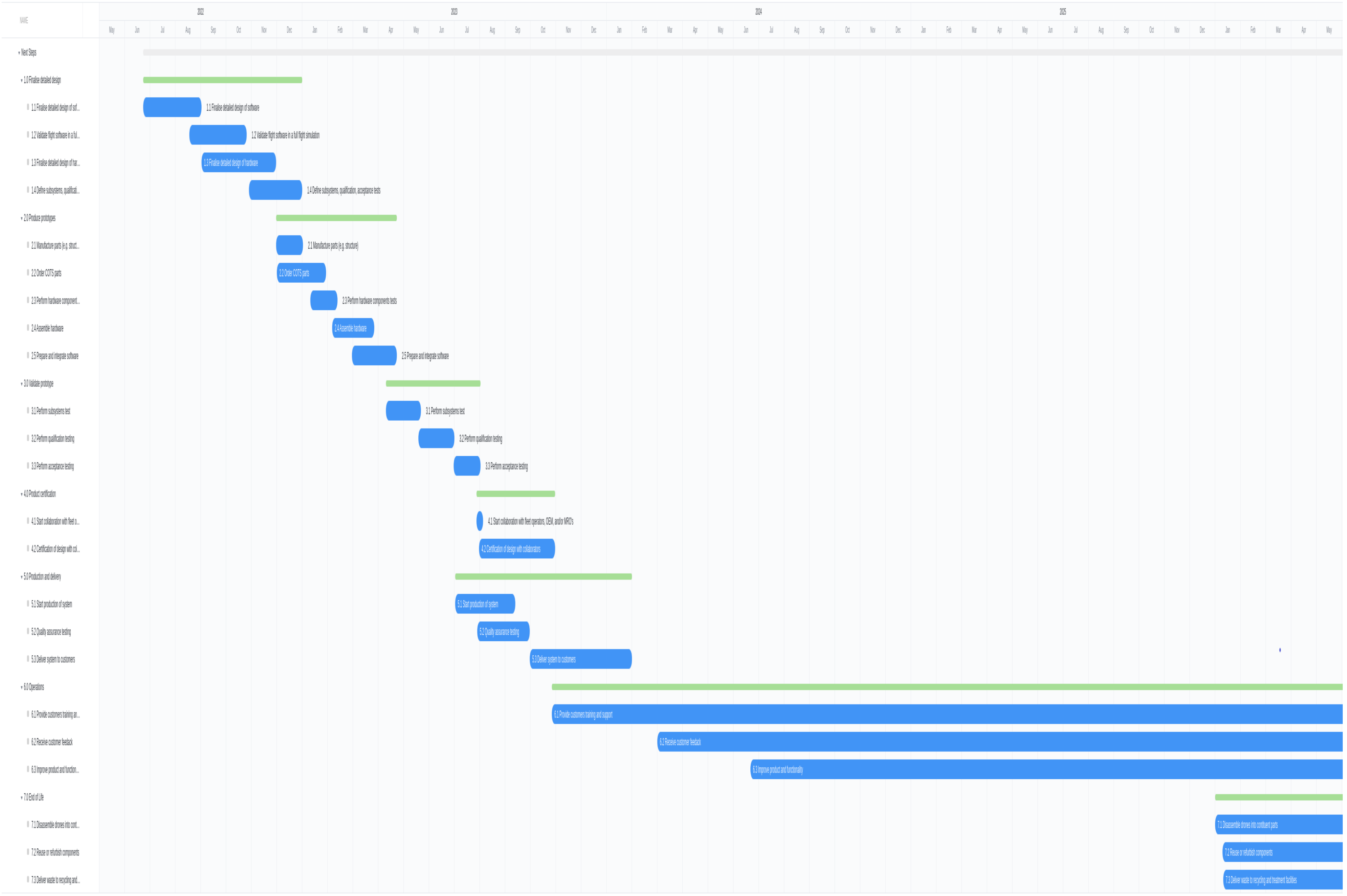


Figure A.1: Project Gantt chart following from the design and development logic.

B

Quadcopter Dynamics and Control

B.1. Additional Quadcopter Dynamics

This section provides additional information on quadcopter dynamics and control. The basics of quaternions are discussed in Subsection B.1.1, followed by a discussion of quadcopter drone forces and moments in Subsection B.1.2.

B.1.1. Quaternions

Quaternions have been used in both the flight (see chapter 6) and swarm control subsystem (see chapter 5). Quaternions provide an alternative method for attitude representation, and have multiple advantages with respect to other methods (e.g. Euler angles). Equation B.1 shows the general form of the quaternion attitude representation.

$$q = \begin{bmatrix} q_0 & q_1 & q_2 & q_3 \end{bmatrix} = \begin{bmatrix} q_0 & \vec{v} \end{bmatrix} = q_0 + q_2\mathbf{i} + q_2\mathbf{j} + q_3\mathbf{k} \tag{B.1}$$

B.1.2. Quadcopter Forces and Moments

Equation 6.1 gives the expression for translational dynamics of a quadcopter drone in the inertial frame. Equation B.2 below provides further detail on the constituents of this expression.

$$\mathbf{F}_I = \begin{bmatrix} 0 & 0 & -m_B g \end{bmatrix}^T + R_{IB} \begin{bmatrix} 0 & 0 & -T_{res} \end{bmatrix}^T + \begin{bmatrix} F_{(x,aero)} & F_{(y,aero)} & F_{(z,aero)} \end{bmatrix}^T \tag{B.2}$$

where R_{IB} is the rotation matrix from the body to inertial reference frame, T_{res} is the resultant actuator thrust, and F_{aero} are the resultant aerodynamic forces. An expression for the aerodynamic forces can be found in the quadcopter dynamics implementation “Quadcopter Simulation and Control (Quad_SimCon)”[1].

Equation B.3 provides an expression for the actuator moment term of Equation 6.2 in Section 6.2. Additionally, it is used by the flight control algorithm to calculate the actuator/motor commands (also called motor mixing). The equation is of the form $Ax = b$, where the flight control algorithm calculates the reference actuator force and moment resultants, i.e. b. The motor commands, x, can subsequently be found by solving for x, or inverting A (such that $x = A^{-1}b$). This method assumes that the parameters l, b, k_T, and k_q are known.

$$\begin{bmatrix} M_c & F_{c,B} \end{bmatrix}^T = \begin{bmatrix} -bk_T & bk_T & bk_T & -bk_T \\ lk_T & lk_T & -lk_T & -lk_T \\ k_q & -k_q & k_q & -k_q \\ -k_T & -k_T & -k_T & -k_T \end{bmatrix} \vec{\omega}^2 \tag{B.3}$$

B.1.3. Motor Dynamics and State Derivatives

The motor dynamics and state derivatives that are used in simulating drone flight dynamics can be seen in Table B.1 and Table B.2 respectively. These equations and states are used for updating drone position, velocity, quaternions, drone rotation speed and motors speed.

[1] Retrieved from: `https://github.com/bobzwik/Quadcopter_SimCon` (cited 18 May 2022)

[0] Retrieved from: `https://www.rtpcompany.com/wp-content/uploads/2013/09/RTPPP30GFFR0BLK.pdf` (cited June 2022)

[1] Retrieved from: `https://forward-am.com/wp-content/uploads/2021/07/ULTRAFUSE%C2%AEPPGF30_11120900_SDS_GEN_EU_EN_3-0_NL07-1.pdf` (cited June 2022)

Table B.1: Motor dynamics equations that are used in simulation

Motor dynamics
$\ddot{w}_{M1} = \frac{-2\xi\tau\dot{w}_{M1} - w_{M1} + k_p cmd_{M1}}{\tau^2}$
$\ddot{w}_{M2} = \frac{-2\xi\tau\dot{w}_{M2} - w_{M2} + k_p cmd_{M2}}{\tau^2}$
$\ddot{w}_{M3} = \frac{-2\xi\tau\dot{w}_{M3} - w_{M3} + k_p cmd_{M3}}{\tau^2}$
$\ddot{w}_{M4} = \frac{-2\xi\tau\dot{w}_{M4} - w_{M4} + k_p cmd_{M4}}{\tau^2}$

Table B.2: State derivatives that are used in simulation

State derivatives
$\dot{x}$
$\dot{y}$
$\dot{z}$
$-0.5pq_1 - 0.5qq_2 - 0.5rq_3$
$0.5pq_0 - 0.5qq_3 + 0.5rq_2$
$0.5pq_3 + 0.5qq_0 - 0.5rq_1$
$-0.5pq_2 + 0.5qq_1 + 0.5rq_0$
$\frac{C_D sign(V_w cos(q_{w1}) cos(q_{w2}) - \dot{x})(V_w cos(q_{w1}) cos(q_{w2}) - \dot{x})^2 - 2(q_0 q_2 + q_1 q_3)(T_{M1} + T_{M2} + T_{M3} + T_{M4})}{m}$
$\frac{C_D sign(V_w sin(q_{w1}) cos(q_{w2}) - \dot{y})(V_w sin(q_{w1}) cos(q_{w2}) - \dot{y})^2 + 2(q_0 q_1 - q_2 q_3)(T_{M1} + T_{M2} + T_{M3} + T_{M4})}{m}$
$\frac{-C_D sign(V_w sin(q_{w2}) + \dot{z})(V_w sin(q_{w2}) + \dot{z})^2 - (T_{M1} + T_{M2} + T_{M3} + T_{M4})(q_0^2 - q_1^2 - q_2^2 + q_3^2) + gm}{m}$
$\frac{(I_{yy} - Izz)qr + (T_{M1} - T_{M2} - T_{M3} + T_{M4})dym}{I_{xx}}$
$\frac{(I_{zz} - Ixx)pr + (T_{M1} + T_{M2} - T_{M3} - T_{M4})dxm}{Iyy}$
$\frac{(I_{xx} - Iyy)pq + (-T_{M1} + T_{M2} - T_{M3} + T_{M4})}{Izz}$

B.2. Control Algorithm

The PID control implementation used can be seen in Algorithm 4 below. The implementation is a modified version of the algorithm presented in “Quadcopter Simulation and Control (Quad_SimCon)”[2].

Table B.3 below shows the parameters that were changed in the controller simulation for PID controller tuning.

Table B.3: Summary table of parameters that are used in the flight control simulation (i.e. for controller tuning).

Parameter	“CHIP” drone Value	“DALE” drone Value	Unit
k_T	1.73×10^{-3}	8.40×10^{-5}	$\mathrm{N\,s^2\,rad^{-2}}$
k_q	1.34×10^{-5}	3.41×10^{-7}	$\mathrm{N\,m\,s^2\,rad^{-2}}$
I_{xx}	2.71×10^{-1}	2.70×10^{-2}	$\mathrm{kg\,m^2}$
I_{xy}	5.84×10^{-4}	5.51×10^{-5}	$\mathrm{kg\,m^2}$
I_{xz}	2.45×10^{-3}	4.44×10^{-3}	$\mathrm{kg\,m^2}$
I_{yx}	5.84×10^{-4}	5.51×10^{-5}	$\mathrm{kg\,m^2}$
I_{yy}	2.70×10^{-1}	4.50×10^{-2}	$\mathrm{kg\,m^2}$
I_{yz}	1.09×10^{-3}	2.36×10^{-7}	$\mathrm{kg\,m^2}$
I_{zx}	2.73×10^{-1}	4.44×10^{-3}	$\mathrm{kg\,m^2}$
I_{zy}	1.79×10^{-3}	2.36×10^{-7}	$\mathrm{kg\,m^2}$
I_{zz}	7.91×10^{-1}	6.68×10^{-2}	$\mathrm{kg\,m^2}$
dxm	3.32×10^{-1}	1.90×10^{-1}	m
dym	3.48×10^{-1}	1.87×10^{-1}	m
dzm	1.20×10^{-1}	1.76×10^{-2}	m
m$_B$	7.80	3.29	kg

[2] Retrieved from: `https://github.com/bobzwik/Quadcopter_SimCon` (cited 18 May 2022)

C_D	1.00×10^{-1}	1.00×10^{-1}	-
ω_{min}	0.00	0.00	$\mathrm{rad\,s^{-1}}$
ω_{max}	3.31×10^{2}	8.87×10^{2}	$\mathrm{rad\,s^{-1}}$
T_{min}	0.00	0.00	N
T_{max}	1.89×10^{2}	66.1	N

Algorithm 4 PID control algorithm for a generic quadcopter drone modified from “Quad_SimCon” by John Bass. β is the maximum drone tilt (pitch, roll). For definitions refer to: ToQuaternion[3], and QuaternionMultiplication[4].

```
if not $| v_{ref} | \leq v_{max}$ then $| v_{ref} | \leftarrow v_{max}$
end if
$v_{z,error} \leftarrow v_{z,ref} - v_z$
$T_{z,ref} \leftarrow (K_p)_z v_{z,error} - (K_d)_z a_z + T_{z,int} - g$
if not ($T_{z,ref} \geq T_{max}$ and $v_{z,error} \geq 0$) or ($T_{z,ref} \leq T_{min}$ and $v_{z,error} \leq 0$) then
    $T_{z,int} \leftarrow T_{z,int} + (K_i)_z v_{z,error} \Delta t$
    $T_{z,int} \leftarrow min(T_{z,int}, T_{max})$
end if
$v_{xy,error} \leftarrow v_{xy,ref} - v_{xy}$
$T_{xy,sp} \leftarrow (K_p)_{xy} v_{xy,error} - (K_d)_{xy} a_{xy} + T_{xy,int}$
$T_{xy,max} \leftarrow min(T_{z,ref} tan(\alpha_{max}), \sqrt{T_{max}^2 - T_{z,ref}^2})$
$T_{xy,ref} \leftarrow T_{xy,sp}$
if $T_{xy,ref}^2 \geq T_{xy,max}^2$ then
    $T_{xy,ref} \leftarrow \frac{T_{xy,ref}}{|T_{xy,ref}|} T_{xy,max}$
end if
$K_{arw} \leftarrow \frac{2.0}{(K_p)_{xy}}$
$v_{error,lim} \leftarrow v_{xy,error} - K_{arw}(T_{xy,ref} - T_{xy,sp})$
$T_{xy,int} \leftarrow T_{xy,int} + (K_i)_{xy} v_{error,lim} \Delta t$
$Z_B \leftarrow \frac{T_{ref}}{|T_{ref}|}$
$Y_C \leftarrow [-sin(\psi_{ref}), cos(\psi_{ref}), 0]^T$
$X_B \leftarrow normalise(Y_C \times Z_B)$
$Y_B \leftarrow Z_B \times X_B$
$q_{ref} \leftarrow$ ToQuaternion($X_B, Y_B, Z_B$)
$q_{error} \leftarrow$ QuaternionMultiplication($q*, q_{ref}$)
$\Omega_{ref} \leftarrow 2.0(K_p)_q \frac{q_0}{|q_0|} q_{1-3}$
if not $| \Omega_{ref} | \leq \Omega_{max}$ then
    $| \Omega_{ref} | \leftarrow \Omega_{max}$
end if
$M_{ref} \leftarrow (K_p)_\Omega(\Omega_{ref} - \Omega) - (K_d)_\Omega \alpha$
```

[3] Retrieved from: `https://nl.mathworks.com/help/robotics/ref/rotm2quat.html` (cited 2 June 2022)

[4] Retrieved from: `https://nl.mathworks.com/help/fusion/ref/quaternion.mtimes.html` (cited 2 June 2022)

C
Cost Table

Table C.1: Components cost per drone.

Component	Quantity	Price [€]
“CHIP” drone		
Manifold 2-G	1	1,300
FRSKY R-XSR	1	24
USB Hub 6 Downstream	1	-
OS0 LIDAR	1	13,000
FCHUB-12S	1	39.15
FC F405-STD	1	49.90
Zenmuse P1	1	6,000
Rctimer HP8108 135KV	4	880
MAD AMPX 40A ESC HV	4	320
25V Buck Regulator	1	3
5V Embedded Buck Regulator	1	3
Mejzlik 26x8.7	8	1,000
GENX 44.4V 12S 16000MAH 25C / 50C PREMIUM LIPO BATTERY	33	23,010
Total	-	45,600
“DALE” drone		
DarwinFPV Betaflight F4 60A 3-6S BLHeli_32 4in1 Dshot1200 ESC	1	40
DarwinFPV Betaflight F4 60A 3-6S BLHeli_32 4in1 Dshot1200 FC	1	40
Manifold 2-G	1	1,300
Intel® RealSense™ Depth Module D450	6	-
Vision Processor D4 USB Type-C V3 Board	6	1,420 (w stereo cams)
FRSKY R-XSR	1	24
USB Hub 6 Downstream	1	-
STEVAL-GMBL02V1	1	100
Gimbal motor	3	-
Velabit LiDAR	1	100
USB hub 2 downstream	1	12
Evo Nano Camera	1	649
8V Buck regulator	1	3
5V buck regulator	1	3
Locosys RTK-1010	1	85
Adafruit 9-DOF Absolute Orientation IMU Fusion Breakout	1	35
Liperior 16000mAh 6S 12C 22.2V Lipo Battery With XT90 Plug	33	4,950
Brotherhobby Avenger 2812-0900 V3	4	120
Total	-	8,880