DronebezORgd

Designing a Fleet of Drones for Last-Mile Food Delivery

Final Report

Group 18, AE3200: Design Synthesis Exercise

Delft University of Technology

DronebezORgd

Designing a Fleet of Drones for Last-Mile Food Delivery

Final Report

Group 18, AE3200: Design Synthesis Exercise

Name	Student Number
Andreas Bart Carnine Swaan	5524237
Andrei Negureanu	5817978
Ching Ting (Andy) Wong Lu	5821290
Felice Verbeek	5525314
Ishaan Jain	5778158
Jan Cywiński	5779464
Marcos Alexandre Talocchi Barbosa	5524660
Pablo Raichs Fernandez	5742633
Subramanian (Siddarth) Ramasubramanian	5710065
Simonas Strasunkas	5534348

Principal tutor:	Dr. Alessandro Bombelli
Coaches:	Dr. Nikhil Garg, Dequan Ou
Teaching Assistant:	Alfonso María de Rato Pueyo
Date:	Tuesday 24th June, 2025
Version:	Final
Faculty:	Faculty of Aerospace Engineering, Delft

Cover:	Old Canal in Delft City in the Netherlands by Jan van der Wolf through Pexels
Style:	TU Delft Report Style, modified by Daan Zwaneveld

Executive Overview

This overview aims to provide a summary of the progress made by Group 18 of the Design Synthesis Exercise (DSE) on their project until the Final Review, the final of three milestones in this graduation project.

- **Project title:** DronebezORgd: Designing a Fleet of Drones for Last-Mile Food Delivery
- **Principal tutor:** Dr. Alessandro Bombelli
- **Coaches:** Dr. Nikhil Garg, Dequan Ou
- **Teaching Assistant:** Alfonso María de Rato Pueyo

With the increasing demand for fast and contactless delivery services, especially in urban environments, drones offer a promising solution for last-mile logistics. As such, the Mission Need Statement (MNS) for this project was formulated:

"Restaurants need a cheaper, faster, safer, more reliable, and more sustainable last-mile delivery system in urban environments."

This project explores the feasibility, design, and implementation of a drone-based system tailored for pizza delivery, addressing logistical, technical, and regulatory challenges in the Netherlands. So, the following Project Objective Statement (POS) was defined:

"To design a fleet of drones capable of delivering at least 200 pizzas in one evening in a Delft-like urban environment with minimal operational cost and environmental impact by 10 students in 10 weeks."

Elaborating on the POS, the primary objectives of the project are to:

- Design a cost-effective and reliable drone capable of delivering pizzas autonomously
- Model the operations of the drone fleet and develop an optimal mission-planning algorithm

The final design resulting from this project is presented in this report. The design is a fleet of drones which combine the vertical take-off and landing capabilities of traditional multicopters by utilising vertical propellers and the efficient cruise of airplanes by having a fixed-wing, a horizontal propeller and a V-tail configuration. Within the drone are two payload bays, each capable of holding three pizzas. This payload is loaded and unloaded by the drone landing at a restaurant's or customer's location, with the drone folding its wings in-air to decrease its landing footprint. On a fleet level, 10 drones are computed to be needed for Delft in order to satisfy the requirement of delivering at least 200 pizzas per evening. These drones are based at a depot and make regular trips to one or multiple restaurants and customers before returning to recharge.

Summary of Previous Work

By the first milestone of the project, the Baseline Review [1], a foundation for the project work had been made. The scope of the project was defined as that of a hypothetical engineering startup to better cover the challenges faced by engineers in the industry. On a technical level, the scope was limited to designing and planning fleet operations in a small Dutch urban area of the size of Delft while developing a city-agnostic optimisation algorithm. Next to this, stakeholders and key functions of the DronebezORgd system were identified using a stakeholder matrix and functional analysis to arrive at a comprehensive list of requirements. A sustainable development strategy was also made to aid the requirement generation process. This phase of the project was concluded by developing technical budgets for resources such as mass, schedule, and emissions.

By the second milestone of the project, the Midterm Review [2], the previous foundation was used to choose a design concept and perform preliminary sizing of it. The choice of design concept was done through the means of two trade-offs. In one, four lifting configuration concepts were evaluated and compared on criteria such as safety, payload handling abilities, and operation costs to arrive at a **fixed-wing electric vertical take-off and landing (eVTOL)** configuration. A second trade-off covered the choice of payload delivery methods. From five concepts, the method of **landing before unloading the payload** was chosen based on its interface with the customer and restaurants, payload protection, and energy efficiency. Based on these choices, the sustainable development strategy and technical risk management plan were updated to better reflect the chosen design concept.

Central to this project phase was the preliminary design of the drone and operations framework. Initial estimates for single- and multi-delivery mission profiles were made, from which an initial battery sizing was performed, indicating multi-delivery as the most demanding case. An iterative procedure was used to arrive at a converged maximum take-off weight of 9.4 kg, and a wing span of 2.5 m, assuming 20% of the lift in horizontal flight was generated by vertical propellers. The design concept was also further defined as a high-wing configuration with propellers attached to the lower fuselage, using removable, rechargeable batteries as a power source.

An operational and logistical framework was also defined. The route management problem is split into two: mission planning covers the optimisation of high-level scheduling of drones, and path planning accounts for real-world complexity like obstacles. The day-to-day operations of the theoretical DronebezORgd service are conceptually split into four units: the drones themselves, the depot(s), the control centre, and the administration centre. A detailed financial model was also developed, starting with the classification and quantification of costs and combining this with a variable-demand revenue estimation to estimate the return on investment. A sensitivity study was conducted to verify that a break-even time of less than five years was possible.

Progress Overview

This report, part of the Final Review, covers the subsystem-level design of DronebezORgd and its fleet of drones, each called BezORger. Building on the preliminary design results of the previous milestone, subsystems are defined and designed in more detail. A look at the future of the design is also taken, assuming the continuation of the development of the food delivery service beyond the DSE.

Project Overview

In Chapter 2, the goals of the project are defined in terms of which functions have to be performed, which risks have to be managed, and what requirements the final system must comply with. Included in this overview is a revised functional flow diagram and functional breakdown structure, detailing all of the functions of the DronebezORgd fleet. Compared to the earlier phases of the project, a final list of system requirements is presented with all values, categories, and verification methods specified.

Business Overview

Building on the requirements and technical descriptions of the system, Chapter 3, the organisation of DronebezORgd as a company is planned. A market analysis is performed, looking at the current state of the market, and it is found that DronebezORgd should aim to have a delivery cost no higher than €7.81 per order on average. Strengths, weaknesses, opportunities, and threats for drone delivery of food are identified, indicating a possibility for the company to achieve a 10% market share due to the nascent stage of the food delivery market. This early phase presents a unique opportunity due to the limited number of established competitors, allowing for a greater potential to capture market share. A stakeholder analysis is also performed, grouping all identified stakeholders by their required interest and power in the project.

A business plan and financial model are also presented. The business plan highlights how the service is expected to generate income through using a dedicated portal to receive orders and taking a 25% fee.

The portal is to be designed such that an order can be accepted or declined within a small time frame, depending on the availability of the fleet. The business is built around the application of the service in a Delft-like city, although it is expected to be applicable and profitable in larger cities. Constraining the design to Delft serves as a case study where tailored choices of variables, such as depot locations and locations can be made. In this business plan, it is also assumed that airspace regulations would loosen as current regulations render DronebezORgd infeasible in most of the country. A choice is also made to size the fleet for optimised profitability, resulting in potential under-capacity in peak periods.

Conceptual Design

Chapter 4 covers the establishment of the design process itself. It defines and describes how trade-offs are used to aid decision-making. Technical resource budgets for the design in terms of mass, energy, and cost are presented, together with contingencies to account for uncertainties due to the lack of existing low-MTOW eVTOLs as reference. This chapter also presents the division of the system into subsystems. Their interdependencies are also identified and discussed using a design structure matrix. By establishing and managing these interfaces early, the team ensured a design process with cohesive evolution of the subsystems and the system as a whole. Lastly, subsystem requirements are derived from the set of system-level requirements, providing sufficient detail to design each subsystem.

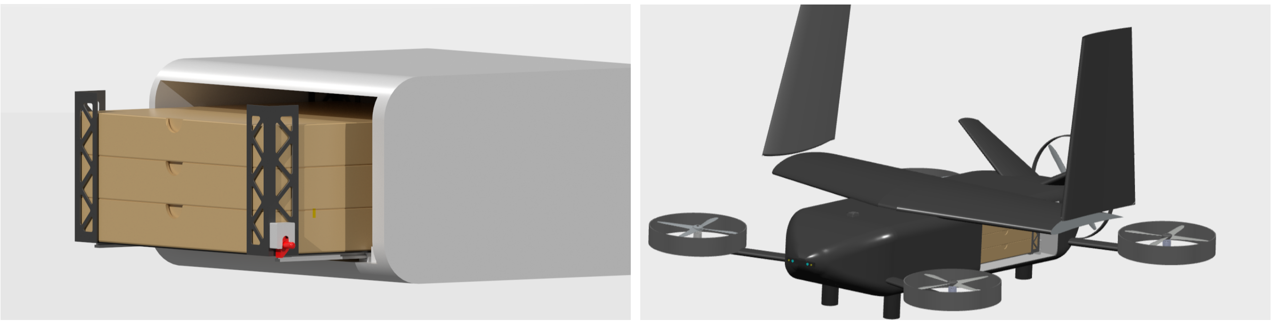
Structures & Materials

Chapter 5 is the first subsystem-specific chapter, covering the structural modelling and design of the BezORger. This subsystem makes sure the drone can withstand all the loads and not fail under any conditions. The whole structure of the drone was idealised to simplify calculations. This chapter also covers the material trade-off for the mainframe. This trade-off included: the material strength, stiffness, manufacturability, ease of maintenance, and material cost. Sustainability has also had an influence on this subsystem as the material should be recyclable, and the mass should be minimised to reduce the power needed. The idealisation used was the boom theory. This theory made some assumptions, including one that was unacceptable for the validity of the structure. To solve this problem a finite element method (FEM) was used in ANSYS to use the stress superposition principle to complete the boom idealisation, rendering the structure valid.

Payload Handling & Avionics

The focus of Chapter 6 is the design of four elements which guarantee the safe, reliable, and efficient handling of the pizza payload. The first of these is the payload delivery mechanism, for which a trade-off of four concepts is performed to arrive at the most suitable concept. The four concepts are referred to as the traditional stack, actuated push, actuated drop, and cyclical stacks, covering a variety in degree of automation and combination of sideways and downwards delivery directions. The most important trade-off criteria were mass, risk, and customer interface. Ultimately, none of the concepts were chosen alone, and the trade-off was used to identify strong points in each concept and create a unique, best performer. This mechanism was designed in two steps: first, doing mass and cost estimations based on simple sketches, and then refining these by making a 3D model in *3DExperience CATIA*, which is shown in Figure 1a. Also shown is the thermal insulation structure responsible for keeping the payload warm and sanitary throughout the mission. The design of this was based on literature comparing different materials for a pizza delivery bag for a similar amount of payload. It was finally decided to use a multi-layer insulation system, mainly consisting of polyurethane foam.

In addition to this, a wingspan reduction mechanism was designed to enable the drone to land in a smaller footprint. A literature study was conducted, indicating that folding the wing along an axis parallel to the drone's symmetry axis is the best-established concept. A mechanism was sized to rotate the wing based on the moment caused by the structural mass of the outboard wing element, applying a design factor to be conservative. In order to safely lock the motion of the wing on the ground and in cruise, a locking mechanism was sized considering the ultimate lifting moment generated by the wing. Lastly, a set of avionics equipment is selected, including a flight computer, which is traded off between popular off-the-shelf components, perception sensors, and notification devices.



(a) Extended payload delivery mechanism relative to thermal insulation structure

(b) Wingspan reduction mechanism in folded state when integrated on the drone

Figure 1: Renders of payload delivery mechanism, thermal insulation (a), and wingspan reduction mechanism (b)

Propulsion & Powertrain

Chapter 7 covers the design of the propeller, motor, and battery of the BezORger. A special focus is put on the propeller design, which is custom-made using the Blade Element Momentum Theory (BEMT) as an analysis tool. Four propellers, each with four blades, were chosen, as this offered a good balance of footprint and controllability. An airfoil selection using high-cambered airfoils was performed, from which the **Clark Y airfoil** was chosen for its relatively low power draw for generating the required amount of thrust. After this, simulations were run to arrive at a pitch angle that maximises thrust generation without stalling, that being 15.5°. Twist and chord distributions were chosen similarly, and the same process was repeated for the horizontal propeller as well, with the only difference being the required thrust and accompanying rotation rate. For ease of manufacturing, all design parameters were kept the same for both the horizontal and vertical propellers, resulting in a **final horizontal propeller radius of 22.2 cm and vertical propeller radius of 20.6 cm**.

Following the propeller design, the motor and battery were selected. The former was done on the basis of required torque and rotation rate of the rotor blades, and the latter, using the voltage and power draw of the motor and propeller. For ease of procurement, the same motor was selected for both the horizontal and vertical propellers, that being the 1,800 W *EVCraft High Torque 6016 Electric Brushless DC Motor*. Five battery chemistries were evaluated in a trade-off, where energy density and safety were the highest-weighted criteria. Finally, the *DJI Mavic 2 Intelligent Li-Po Flight Battery* was chosen. With its charge of 3,875 mh and voltage of 15.4 V, 8 units were required to provide the required energy, with every two batteries connected in series to supply the required voltage.

Noise

Building on the propulsion subsystem design, an analysis of the drone's noise characteristics was performed; this is presented in Chapter 8. This was done in order to verify that the drone's noise emissions are at acceptable levels to be appropriate for use in urban environments. From literature, it was found that tonal noise is dominant compared broadband noise during take-off and landing, meaning the Hanson model could be used. This model estimates the tonal noise generated by rotating propeller blades. To obtain a measure of actual perceived disturbance for humans, A-weighting was applied onto the results of this model. In the end, it was found that the maximum sound pressure level occurs at take-off and is estimated to be 69 dB or 55 dBA, when measured 2 m from the drone, satisfying the identified noise requirements.

Aerodynamics

Chapter 9 covers all design considerations made with regard to the aerodynamic performance of the drone. First in this process is the selection of a suitable airfoil. This choice was done among the 4-digit NACA profiles as these have a relatively favourable stall behaviour, suitable to the BezORger's transition phase between take-off and cruise. From these, the **NACA-6412 airfoil was chosen** as it had the highest lift coefficient. Due to the complicated and unconventional shape of the drone, computational fluid dynamics (CFD) was utilised to simulate the aerodynamic performance. *3DExperience SIMULIA* software was chosen, together with the shear stress transport k-ω model, to run simulations at angles of attack at multiples of 3° between 0° and 12° to construct lift and drag curves. Next to this, a sensitivity analysis was performed by varying the wing sweep from 0° to 10° and the tip and root chords by ± 0.10 m. This analysis was used both as a verification of the method and for the development of an iterative wing planform sizing method by interpolating the data.

Stability & Control

For the analysis of the stability and controllability of the drone in Chapter 10, the dynamics of the drone were split in vertical and horizontal flight. For vertical stability and controllability, the drone is treated as a quadcopter. Using force and moment equilibria of the drone during vertical flight, different one-engine-out scenarios can be analysed to find the maximum required rotation rate of a propeller. Out of the four vertical propellers, it is the failure of one of the front ones, though this is manageable under the selected powertrain. In horizontal flight, the drone behaves analogously to an aircraft. Because of this, a scissor plot is created to evaluate the longitudinal stability, relating the ranges of stability and controllability to the tail area and the range of centre of gravity locations. In order to minimise the required tail area, an iterative procedure of moving the wing with respect to the fuselage was performed. The required vertical tail area of the V-tail configuration is found from the lateral force required to control the yawing of the drone.

Performance

Chapter 11 covers an analysis of numerous aspects of the performance of the drone. Using simulations of the operational mission planning, the mission profile of the BezORger was chosen to include two pick-ups and three deliveries. The distance of an average mission is quantified through a probabilistic spatial simulation over a circular city map of diameter 6 km. Additionally, different phases of flight are defined using vertical take-off and landing for the lower 20 m, to avoid most obstacles, and cruising at 60 m, as this was found to reduce ground-level noise without using too much energy during ascent. Building on this, the mission energy is estimated phase-by-phase, with a special emphasis on the transition phase, going from vertical to horizontal flight. Lastly, the flight envelope is defined based on the stalling speed and the influence of gusts and manoeuvring. Operational limitations regarding path planning to prevent damage to the payload are also considered, preventing slipping of the pizza with respect to the pizza box.

Operations

In Chapter 12, the design of the operations of the fleet is presented. The main goal of this subsystem is to design a system to control the operational fleet utilisation for maximised profit within given constraints. Additionally, profitability shall be ensured by continuing the development of the return on investment calculations from previous design phases. The operational problem is split into three modules. The first of these is mission planning, responsible for the high-level scheduling of the fleet. After a comparison of solution methods, it was decided to test **k-means clustered mixed-integer linear programming (MILP)** and a greedy algorithm, and a full formulation of the Dial-A-Ride Problem (DARP) is presented. The second module is path planning, concerned with finding the shortest and least public-disturbing path between two points as received by mission planning. Also, for this part a comparison of algorithms is performed, resulting in the application of the **A* algorithm** due to its relatively fast runtime without ruling out optimality.

As a third part, a simulation framework is developed for the testing and improvement of the preceding parts, and to provide profit estimates. In the simulation, as orders are placed, mission planning is called to assign drones to deliveries, and then path planning is called to calculate their exact paths. A digitised version of Delft is created using neighbourhood-wise population densities and real restaurants and no-fly zones. From this simulation, it is ultimately found that the greedy algorithm outperforms the clustered MILP in runtime and profit, but it does not allow for scheduling orders far enough ahead in time, leading to the **final choice of clustered MILP**. Simulations are also run to optimise the maximum payload size, leading to the choice of six pizzas instead of the required five. An analysis of the effect of the number of drones concludes that 10 drones are optimal for the return on investment.

Design Integration

In Chapter 13, the subsystem-level methods are integrated in a design iteration framework to arrive at a converged value for the MTOW. The Class II weight estimation method is also repeated, arriving at a **final MTOW of 13.06 kg**. Following this, a sensitivity analysis is performed to get more insight about the final design, the results of this being that the iteration framework exhibits expected behaviour, and supporting the choice of the six-pizza payload instead of the original five.

Additionally, the configuration and architecture of the BezORger is presented, including an overview of the external configuration, presenting the location and orientation of major components and sensors. The internal layout is also presented, highlighting how the internal space has been utilised to efficiently carry the designed payload and required sensors and other equipment. In addition to this, the interfacing between electrical components, hardware, software, and data handling elements is described through a set of informative diagrams.

Design Overview

This chapter, Chapter 14, contains an overview of the most important design parameters and their values after the successful application of the design integration iteration method. A cost breakdown of the different subsystems is also presented. Also central in this chapter is the compliance matrix, an overview of the compliance of the BezORger in terms of its adherence to design goals of the system, including functions, requirements, and sustainable development goals. Finally, a recap of the project's sustainable development strategy is presented, focusing on how this strategy has been defined and implemented in the subsystem- and system-level design of the drone and the DronebezORgd system. Next to a subsystem-level overview of sustainability implementation, an overview of sustainability across the system's lifecycle is also presented.

Manufacturing & Maintenance

In Chapter 15, the production plan and RAMS (Reliability, Availability, Maintainability, and Safety) principles are explored. The production plan includes the purchasing of the components, the assembly of the different subsystem groups: fuselage, tail, and wings. It also includes the assembly of the previously mentioned groups. After the assembly, testing will be performed on the drone, such as vibration tests, structural tests and fatigue tests. The RAMS has a clear plan for testing the safety of the drone. RAMS also includes the consideration of a waterproof design such that the drones can operate in a larger variety of weather, such that the profit is maximised.

Future Outlook

The last chapter of this report is Chapter 16. This chapter includes the future of the project after the DSE. It includes a detailed plan for the future design phases. After the DSE, the design shall be further developed, which includes crafting rigorous V&V procedures and a more detailed subsystem design. The following milestone is the manufacturing of the drones and the depot. This includes the production of the parts, the hiring of personnel and assembling the subsystems. Assembly, integration & testing (AIT) includes the final assembly of the drone and is the last step before launching the operations. This chapter also includes the most important dates with a Gantt chart.

Contents

Nomenclature

Symbols

Symbol	Definition	Units	Symbol	Definition	Units
γ	Specific heat ratio	-	ω	Angular velocity	1/s
ρ	Air density	kg/m^3	T	Thrust	N
BPF	Blade passing frequency	Hz	F_e	External force	N
c	Chord to diameter ratio	-	$\ddot{z}$	Vertical acceleration	m/s^2
C_l	Lift coefficient distribution along span	-	$\dot{z}$	Vertical velocity	m/s
C_d	Drag coefficient distribution along span	-	M_e	External moment	N m
C_L	Total lift coefficient of wing	-	r_f	Distance from C.G. to frontal propeller	m
M_{IN}	Inflow Mach number	-	r_b	Distance from C.G. to rear propeller	m
M_{FL}	Flight Mach number	-	r_{D_x}	Distance from C.G. to drag action point	m
N_{rotors}	Number of rotors	-	$\ddot{\Psi}$	Yaw angular acceleration	1/s^2
p_{ref}	Reference sound pressure	Pa	I_{zz}	Mass moment of inertia around Z axis	kg m^2
r_{hub}	Propeller hub radius	m	$\overline{x_{cg}}$	Normalized C.G. location w.r.t. MAC	m
r_{tip}	Propeller tip radius	m	$\overline{x_{ac}}$	Normalized A.C. location w.r.t. MAC	m
s	Sweep	m	$C_{L_{\alpha_h}}$	Tail lift constant slope	
c_o	Speed of sound	m/s	$C_{L_{\alpha_{A-h}}}$	Wing lift constant slope	
L_P	Sound pressure level	dB	$C_{L_{A_h}}$	Wing lift constant	-
$L_{P,\text{total}}$	Total sound pressure level	dB	C_{L_h}	Tail lift constant	-
L_{PA}	Adjusted sound pressure level	dBA	$C_{m_{ac}}$	Wing moment constant around A.C.	-
$L_{PA,\text{total}}$	Total adjusted sound Pressure level	dBA	$\frac{d\epsilon}{d\alpha}$	Downwash gradient	-
t/c	Max thickness to chord ratio	-	S_h	Tail area	m^2
V_{flight}	Flight velocity	m/s	l_h	Tail arm	m
n_{items}	Number of items	-	$\overline{c}$	Mean aerodynamic chord	m
$x_{2\text{ depots}}$	Boolean indicating whether there are 2 depots	-	$\frac{V_h}{V}$	Tail to wing freestream velocity ratio	-
r_{depot}	Distance depot to city center	m	S.M.	Safety Margin	-
n_{max}	Design load factor	-	β	Sideslip angle	°
b	Wingspan	m	α_{tail}	V-tail angle w.r.t. XY plane	°
L(y)	Spanwise wing lift	N/m	P	Power	W
t	Thickness	m	E	Energy	J
σ	Stress	Pa	$\frac{T}{W}$	Thrust to weight ratio	-
c_t	Tip chord length	m	η	Efficiency	-
c_r	Root chord length	m	R	Range	km
W_{wing}	Wing weight	kg	g	Acceleration due to gravity	m/s^2
τ	Shear stress	N/kg	U_g	Gust Velocity	m/sec

d	Pin diameter	m	X_{LEMAC}	X-position of LEMAC	m
r_{blade}	Propeller blade radius	m	Y_{LEMAC}	Y-position of LEMAC	m
AR	Aspect ratio	-			
S	Wing surface area	m^2			
λ	Taper ratio	-			
α	Angle of attack	°			
C_t	Propeller efficiency constant	-			
C_q	Aerodynamic drag for blade rotation	-			

Abbr	**Definition**	**Abbr**	**Definition**
AI	Artificial Intelligence	GNSS	Global Navigation Satellite System
AOA	Angle of Attack	IR	Infra-red
AIT	Assembly, Integration, and Testing	LED	Light Emitting Diode
AS	Assumption	MILP	Mixed-Integer Linear Programming
BR	Baseline Review	MRO	Maintenance, Repair, and Overhaul
C	Customer	MTOW	Maximum Take-Off Weight
CAD	Computer-Aided Design	MR	Midterm Review
CFD	Computational Fluid Dynamics	NACA	National Advisory Committee for Aeronautics
CFRP	Carbon Fiber Reinforced Polymer	NVM	Normal Force, Shear Force and Moment
C.G.	Centre of Gravity	NVWA	Dutch Food and Consumer Product Safety Authority
D	Depot	OPS	Operations
DARP	Dial-A-Ride-Problem	OSM	OpenStreetMap
dBA	A-weighted decibels	PCM	Phase Change Material
DES	Design System	PET	Polyethylene Terephthalate
DSE	Design Synthesis Exercise	PL	Payload
DSM	Design Structure Matrix	PP	Propulsion and Powertrain
EASA	European Union Aviation Safety Agency	PU	Polyurethane
EOL	End of Life	PDP	Pick-up and Delivery Problem
EU	European Union	R	Restaurant
eVTOL	electric Vertical Take-Off and Landing	ROI	Return On Investment
FBS	Functional Breakdown Structure	RQ	Requirement
FFD	Functional Flow Diagram	SAD	Structural Analysis and Design
FM	Financial Model	SC	Stability and Control
FEA	Finite Element Analysis	SM	Structures and Materials
VALID	Verifiable, Achievable, Logical, Integral, and Definitive	SST	Shear Stress Transport
VRP	Vehicle Routing Problem	SWOT	Strength, Weakness, Opportunity, Threat
VRPTW	Vehicle Routing Problem with time windows	TBD	To Be Determined
VRPPD	Vehicle Routing Problem with pick-up and delivery	ToF	Time of Flight
V&V	Verification & Validation	UAV	Unmanned Air Vehicle
USD	United States Dollars		

List of Figures

List of Tables

1 Introduction

Food delivery is one of the fastest-growing markets in the world, offering a variety of opportunities for startups to innovate within it. In 2024, the market was valued at 288 billion USD and is expected to rise to 505.5 billion USD by 2050[1]. Companies like Uber Eats and Thuisbezorgd are examples of successful food delivery services, owning a majority of the market share. The main advantage that food delivery services bring to the customer is saving time on cooking or travelling for food.

While the food delivery market is very profitable, the entry into such a market is not very accessible to a lot of innovative startups. This is because companies like Uber Eats capture a large portion of the market share, thus they can dictate the usual practices within it. For example, in the Netherlands, the main food delivery practices are to use bikes to deliver the food, which requires a low initial investment, but suffer from inconsistent delivery times due to traffic. Moreover, most of the personnel used to deliver food are paid only minimum wage, while they have to endure rainy and windy days. Furthermore, the monopoly some of these companies have in different regions results in unjustified price increases without much risk of competition.

Over the last few years, a new subsidiary of the food delivery market has experienced an increased interest in the eyes of customers and restaurants, namely, drone food delivery. This emerging market offers easier access for innovative startups to kick off their business, as it brings significant improvements over the standard business model. The price per delivery is drastically reduced due to the automation of the drones, not requiring human personnel to deliver, while the delivery consistency is also increased as drones can bypass traffic. These advantages have the potential to close the gap between startups and the larger corporations in the food delivery market.

The objective of this report is to design a fully autonomous fleet of drones capable of delivering at least 200 pizzas in one evening in a Delft-like urban environment with minimal operational cost and environmental impact by ten students in ten weeks. This aim comes from the restaurant's need for a cheaper, faster, safer, more reliable, and more sustainable last-mile delivery system in urban environments. This is going to be achieved by creating a business, manufacturing, and operations plan for the conceptual design, in which each required subsystem is explored. Literature study of other companies' business models and drone design will be used to build the preliminary tools used in designing both the drone and its operations. While the scope of the project is ambitious, the time limit of ten weeks will limit the advancement towards the final aim; thus, only a preliminary design is expected as the end result of the DSE.

[1] URL `https://www.grandviewresearch.com/industry-analysis/online-food-delivery-market-report` [cited 13 June 2025]

2
Project Overview

This chapter presents an overview of the work done during the project baseline phase. This is centred around establishing a strong systems engineering foundation upon which the following design phases can expand. This is done through a functional analysis as presented in Section 2.1, a technical risk assessment as described in Section 2.2, and the formulation of the system requirements as seen in Section 2.3.

2.1. Functional Analysis

Before the conceptual design phase, a functional analysis of the system was conducted as a starting point in the design and as a tool for the formulation of specific system requirements. This entailed the design of the functional flow diagram presented in Subsection 2.1.1, and the functional breakdown structure presented in Subsection 2.1.2.

2.1.1. Functional Flow Diagram

A functional flow diagram (FFD) was set up for this design to aid the design process. Every block in the FFD represents a function that the design has to be able to perform, as derived from the stakeholder requirements. It shows all the steps of the design, starting with the design of the systems, followed by the manufacturing of the drone. The next step is the assembly, integration, and testing, where the drone is mainly assembled, combining the interfaces of the subsystem, and subsequently performing testing for verification and validation. The next block is the operational life of the drone. Finally, the last two blocks are the maintenance, repair, and overhaul, grouped together, and the decommissioning of the system. The whole diagram can be found in Figure 2.2.

2.1.2. Functional Breakdown Structure

The Functional Breakdown Structure (FBS) is a more detailed FFD, meaning that this diagram goes one level further in the definition of functions, but the main difference is that the FBS does not capture any sequential order. This means that from the FBS, it can not be seen which function follows upon the other. The whole diagram can be found in Figure 2.3.

2.2. Technical Risk

This section builds upon the technical risk assessment performed in the previous reports [1][2] by identifying and mitigating additional risks specific to subsystems in Tables 2.2 and 2.3, respectively. Figure 2.1 graphically shows the effect of mitigation and contingency plans on a risk map.

Table 2.1: Likelihood and severity score descriptions

Likelihood (L)	Probability (P)	Severity (S)	Description
1	$0.0 \leq P < 0.1$	1	Negligible: Minor inconvenience or non-operational impact
2	$0.1 \leq P < 0.3$	2	Marginal: Degradation of secondary mission or minor reduction in technical performance

3	$0.3 \leq P < 0.5$	3	Moderate: Non-negligible but acceptable reductions in performance
4	$0.5 \leq P < 0.7$	4	Critical: Mission success is questionable or significant reduction in performance
5	$0.7 \leq P < 1.0$	5	Catastrophic: Mission failure or significant non-achievement of performance

Table 2.2: Risks identified for the all functional phases; highlighted risks have been managed (Table 2.3)

Identifier	Description	L	S
TR-DES-P-06	Inaccurate aerodynamic performance estimation leads to reduced range	2	4
TR-DES-P-07	Noise levels exceed acceptable urban limits	3	3
TR-MAN-P-05	Manufacturing defects in custom-designed propellers	2	3
TR-AIT-P-05	Improper sensor calibration leading to general loss of performance	2	3
TR-OPS-P-21	GPS spoofing/jamming in urban area disrupts autonomous navigation	2	4
TR-OPS-P-22	Battery overheating in hot weather during multi-delivery missions	2	4
TR-MRO-C-04	Incompatible software version during routine maintenance update	1	2
TR-EOL-S-02	Improper battery disposal causing environmental damage	2	3

Table 2.3: Risk control measures and responsible member *(initials)* for the most critical risks; number of arrows indicate point reduction in score

Identifier	Mitigation plan	Contingency plan	NL	NS
TR-DES-P-06	Perform high-fidelity CFD simulations *(AW)*	-	1↓	4
TR-DES-P-07	Use high-accuracy noise estimation models *(FV)*	-	2↓	3
TR-OPS-P-21	-	Define emergency override and landing protocols	2	3↓
TR-OPS-P-22	-	Design active/passive thermal management system *(SS)*	2	3↓

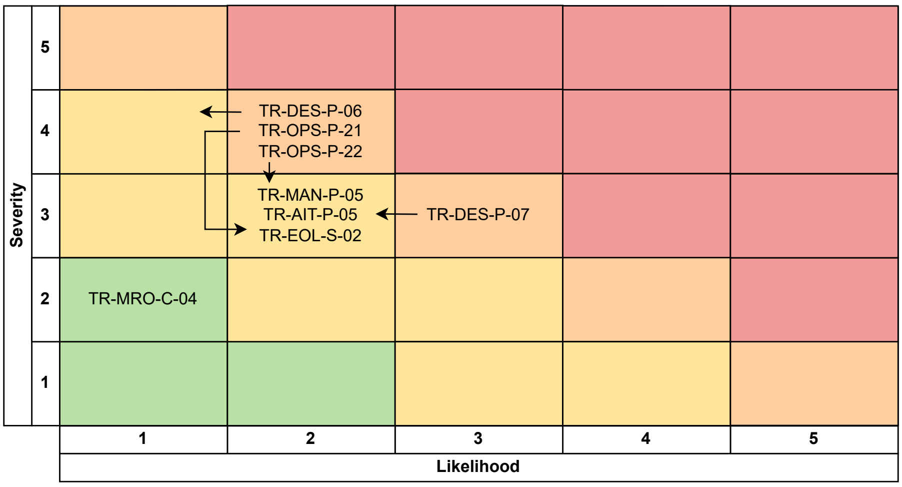


Figure 2.1: Technical risk map before and after implementing risk mitigation and contingency plans; arrows indicate the effect of risk control

2.3. System Requirements

At this stage in the design, more details of the mission are known, leading to a revised list of system requirements. Most of the TBD placeholders were replaced with more accurate numbers, and some requirements have been added, removed, or modified. Additionally, for clarification, some definitions used in the requirements are provided:

- **Mission planning:** The planning for which drone picks up which order at what time. See Section 12.2 for more information.
- **Path planning:** The planning for how exactly a drone goes from point A to point B. See Section 12.3 for more information.
- **Computing module:** The subsystem in charge of calculating path and mission planning, as well as all the communication between drones and the depot and/or control centre. This may be inside the drone, outside the drone, or split between the two.
- **User interface:** The interface used by the customers to place orders.

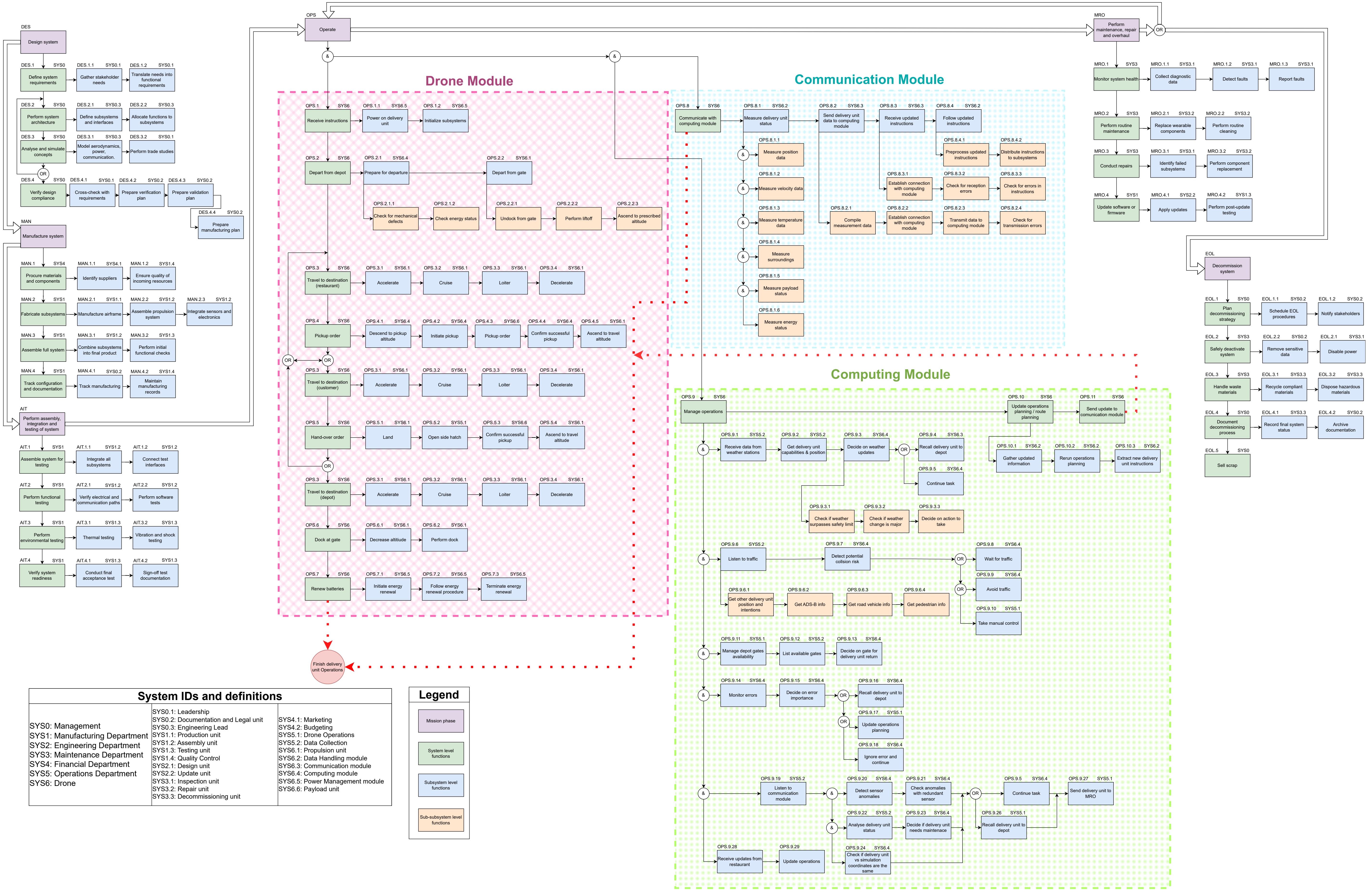
Table 2.4: System requirements (CT: category, F: functional, O: operational, C: constraint) and verification methods (VM: verification methods, T: test, A: analysis, I: inspection, D: demonstration)

Identifier	System Requirement	CT	VM
RQ-01-01-01	Each requirement shall be verifiable by means of testing, inspection, analysis, or demonstration.	F	I
RQ-01-01-02	Each requirement shall be written in accordance with the VALID protocol.	O	I
RQ-01-01-03	The design of the drones shall be verified for compliance against each requirement by means of testing, inspection, analysis, or demonstration.	O	I
RQ-01-02-01	The production plan shall specify materials to be used for each component.	F	I
RQ-01-02-02	The production plan shall specify all manufacturing processes required.	F	I
RQ-01-02-03	The production plan shall specify the interfaces of components.	F	I
RQ-01-02-04	The production plan shall specify assembly of components.	F	I
RQ-01-02-05	The production plan shall specify testing procedures for the design.	F	I
RQ-01-03-01	A drone shall be able to collect pizzas at the restaurants.	F	D
RQ-01-03-02	A drone shall be able to accommodate a pizza with a diameter of at least 45 cm.	F	D
RQ-01-03-03	A drone shall be able to deliver pizzas at specified locations.	F	D
RQ-01-03-04	The drone shall remain intact throughout all planned mission profile phases.	F	D
RQ-01-04-02	A drone shall have a range of at least 5 km when at maximum payload weight.	F	T
RQ-01-04-03	A drone shall have a total range of at least 10 km.	F	T
RQ-01-04-06	A drone shall be able to carry at least 45 by 45 by 30 cm^3 of volume.	F	I
RQ-01-04-07	A drone shall be able to carry at least 2.5 kg of payload.	F	D
RQ-01-04-08	The drone shall be able to land in a 2 by 2 m^2 area.	F	D
RQ-01-04-09	The drone shall be able to land at an incline of at least 10°.	F	T
RQ-01-05-01	The computing module shall be able to perform mission planning in compliance with other requirements given the current state information.	F	A
RQ-01-05-02	The computing module shall be able to perform path planning in compliance with other requirements given the current state information.	F	A
RQ-01-05-03	A drone shall not deviate more than 1 m from the planned path at any point during nominal operations.	O	T
RQ-01-05-04	A drone shall be able to recognise obstacles in a 20 m radius	F	D
RQ-01-05-06	The drone path shall keep a distance of at least 3 m from any obstacle, barring proximity operations at the depot, the restaurant, and the customer locations.	O	T
RQ-01-05-08	A drone shall be able to communicate with the depot during nominal operation.	F	D
RQ-01-05-10	A drone shall be able to estimate position with a 1.5 m accuracy.	F	T, A
RQ-01-05-11	A drone shall be able to estimate velocity with a 1 m/s accuracy.	F	T, A

RQ-01-05-12	A drone shall be able to perform acceleration estimation with a 0.5 m/s^2 accuracy	F	T, A
RQ-01-05-13	A drone shall be able to estimate battery charge percentage with a 3% accuracy.	F	T, A
RQ-01-05-15	A drone shall maintain an energy reserve level of at least 20% at all times during nominal operations.	F	D
RQ-02-01-01	The path planning algorithm shall adapt paths to account for weather updates.	F	A
RQ-02-01-03	The fleet shall be recalled to the depot if the average temperature in the city goes below 0 °C.	O	D
RQ-02-01-04	The fleet shall be recalled to the depot if average wind speed in the city exceeds 7 m/s.	O	D
RQ-02-02-01	Each sensor group and safety-critical mechanism shall have N+1 redundancy.	C	I
RQ-02-02-02	Each drone depot shall have at least 1 drone operator personnel on standby.	O	I
RQ-02-02-03	A detailed sensor malfunction protocol shall be defined for each sensor.	O	I
RQ-02-03-02	A drone shall communicate detected collision risks with the computing module.	F	D
RQ-02-03-03	The computing module shall be able to compute the path plan in less than 12 s.	F	T
RQ-02-03-04	The computing module shall be able to compute the mission plan in less than 30 s.	F	T
RQ-03-01-01	The cost of each drone shall not exceed €10,000.	C	I
RQ-03-01-02	The initial set-up cost of the business, including the cost of the drone fleet, shall not exceed €500,000.	C	A
RQ-03-02-01	The profit of the business shall exceed 0% in 5 years of operation.	F	A
RQ-03-02-02	The objective function of the operations planning algorithm shall maximise profit given the parameters of the model.	O	A
RQ-03-03-01	The operational costs of the business shall be lower than the operational costs of current delivery systems for equal revenue generated.	F	A
RQ-03-03-02	The drone delivery system shall have fewer delays than current pizza delivery systems for an equal number of orders delivered.	F	A
RQ-03-04-01	Fewer than 1 in 100 deliveries shall incur minor accidents, defined as incidents that cause damage but allow continuation of drone operation.	F	A
RQ-03-04-02	Fewer than 1 in 100,000 deliveries shall incur major accidents, defined as incidents that cause damage and disallow continuation of drone operation and/or injury to people.	F	A
RQ-04-01-01	At least 70% of the structural mass of a drone shall be recyclable by currently available technology.	F	A
RQ-04-01-02	The operational energy consumption shall be lower than that of current alternatives for equal operational revenue.	F	A
RQ-04-02-01	Employees shall be compensated in accordance with local union rates.	O	I
RQ-06-01-01	The mission planning shall constrain the earliest delivery time to the scheduled delivery time.	O	I
RQ-06-01-02	Loitering protocols shall be defined.	O	I
RQ-06-02-01	The path planning shall provide delivery time estimates with a 90% confidence interval of 120 sec.	F	A
RQ-06-02-02	The path planning shall update delivery time estimates every 60 s.	F	D
RQ-07-01-02	The maximum tilt angle of a pizza container during delivery shall not exceed 15° from the axis with no lateral force.	C	T
RQ-08-01-01	The user interface shall display all restaurants available for delivery.	F	I
RQ-08-01-02	The user interface shall display approximate available delivery times for each restaurant.	F	I

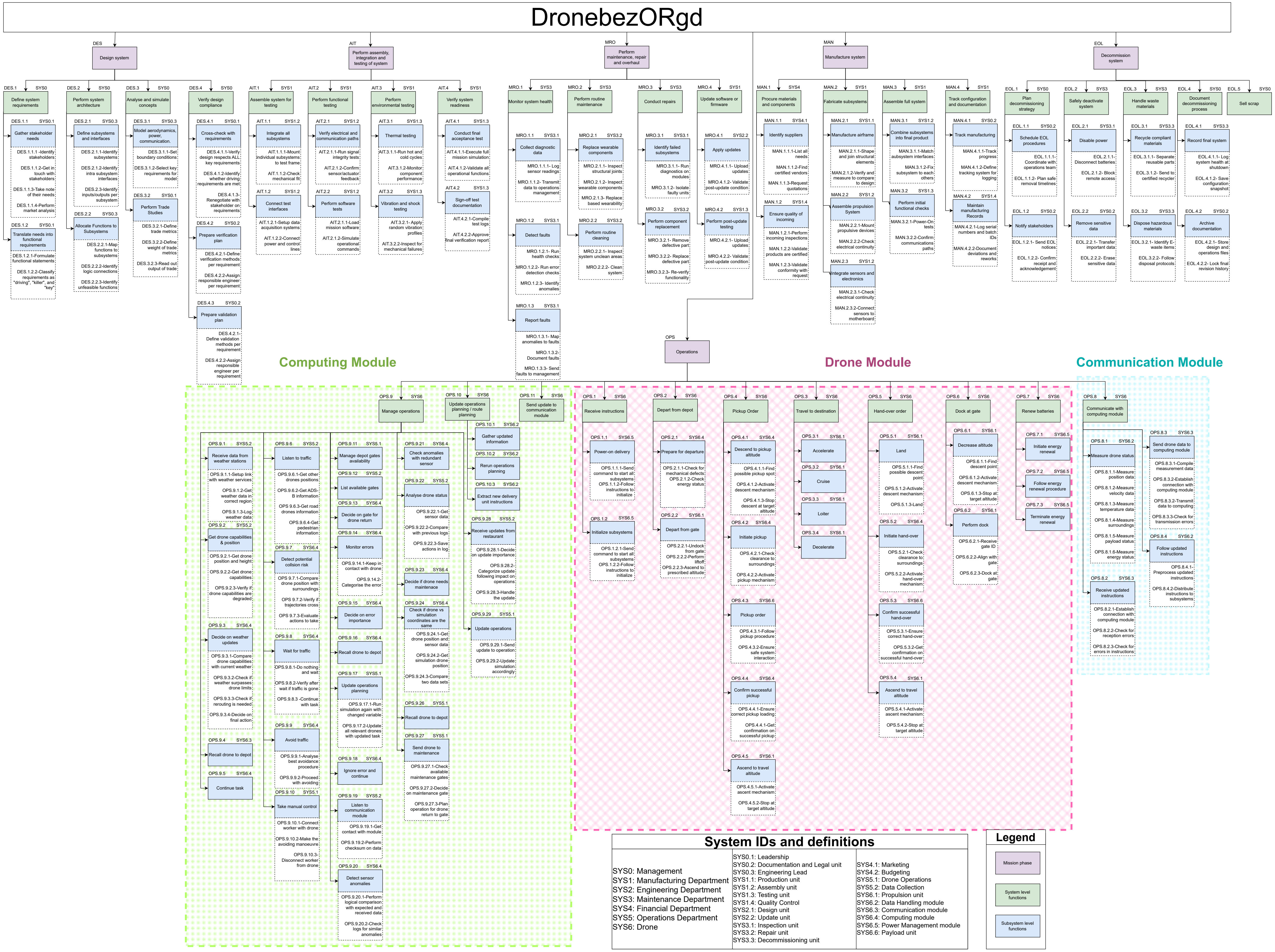
RQ-08-02-01	The user interface shall provide live order tracking with location updates every 60 s.	F	I
RQ-08-02-02	The user interface shall provide a live delivery time estimate with updates every 60 s.	F	I
RQ-08-02-03	The user interface shall provide a confirmation upon delivery.	F	I
RQ-09-01-01	A drone shall emit no more than 50 dBA of sound pressure level as measured on the ground during cruise flight.	C	T, A
RQ-09-01-03	The mission and path planning algorithm shall apply a penalty to the objective function for noise emissions near residential buildings.	F	I
RQ-09-02-01	The drones shall only collect video footage essential for guidance, navigation, control, and obstacle avoidance.	O	I
RQ-09-02-02	A drone shall not collect audio measurements at any point during operation.	C	I
RQ-09-02-03	The video footage collected for guidance, navigation, and control shall be handled strictly in accordance with TBD protocol.	O	I
RQ-09-02-04	The system shall allow residents to request a drone path audit or raise a privacy concern via an online portal.	F	I
RQ-09-03-01	The number of drones within a 10 m radius at any location other than the depot shall not exceed 2.	C	T, A
RQ-09-03-02	The drone shall provide a 2 m safety clearance when the customer picks up the order.	C	T, A
RQ-09-03-03	The drone shall verify an unobstructed landing site prior to initiating landing.	C	T, A
RQ-10-01-01	The path planning shall not plan paths through designated no-fly zones.	C	I
RQ-10-02-02	A drone shall emit no more than 70 dBA of sound pressure level at any point during its operations.	C	T, A
RQ-10-03-01	The location(s) of the depot(s) shall be restricted to legally permissible locations.	C	I
RQ-10-03-02	Employees shall be paid a compensation of at least the legal minimum wage.	C	I
RQ-12-03-01	The drone shall be equipped with a remote identification system as per EASA regulation.	F	I
RQ-13-01-01	The payload bay shall be washable.	F	D
RQ-13-01-02	The payload bay shall ensure the payload is only in contact with food-safe materials as defined by the Dutch National Institute for Public Health and the Environment.	F	I
RQ-13-01-03	The payload shall be protected from ingress of dust, insects, and moisture while in flight.	F	T
RQ-13-02-01	The payload shall maintain an internal temperature of at least 63 °C throughout a mission.	F	T, A
RQ-13-02-02	Temperature logs of the payload bay shall be recorded and stored for at least 30 days.	O	I
RQ-13-02-03	Food safety audits shall be performed at least once every 30 days.	O	I
RQ-14-01-01	The project shall be thoroughly planned in the first week of the DSE in accordance with the given project management guidelines.	O	I
RQ-14-01-02	A dedicated Project Manager shall ensure adherence to TU Delft's project management guidelines.	O	I
RQ-14-02-01	The design process shall be structured in accordance with given systems engineering practices.	O	I
RQ-14-02-02	A dedicated Systems Engineer shall ensure adherence to given systems engineering practices.	O	I

RQ-14-03-01	The design activities carried out during the DSE shall result in a preliminary design of an aerospace system from scratch as outlined in the Project Guide.	O	I
RQ-14-03-02	The design activities carried out during the DSE shall involve the application of aerospace disciplines covered in the bachelor curriculum.	O	I
RQ-14-04-01	The group shall define team operational procedures with regards to scheduling and communication.	O	I
RQ-14-04-02	The group members shall strictly follow team operational procedures.	O	I
RQ-14-04-03	A dedicated Chairman shall ensure adherence to team operational procedures.	O	I


DES
Design system
DES.1 SYS0 Define system requirements
DES.1.1 SYS0.1 Gather stakeholder needs
DES.1.2 SYS0.1 Translate needs into functional requirements
DES.2 SYS0 Perform system architecture
DES.2.1 SYS0.3 Define subsystems and interfaces
DES.2.2 SYS0.3 Allocate functions to subsystems
DES.3 SYS0 Analyse and simulate concepts
DES.3.1 SYS0.3 Model aerodynamics, power, communication.
DES.3.2 SYS0.1 Perform trade studies
OR
DES.4 SYS0 Verify design compliance
DES.4.1 SYS0.1 Cross-check with requirements
DES.4.2 SYS0.2 Prepare verification plan
DES.4.3 SYS0.2 Prepare validation plan
DES.4.4 SYS0.2 Prepare manufacturing plan
MAN
Manufacture system
MAN.1 SYS4 Procure materials and components
MAN.1.1 SYS4.1 Identify suppliers
MAN.1.2 SYS1.4 Ensure quality of incoming resources
MAN.2 SYS1 Fabricate subsystems
MAN.2.1 SYS1.1 Manufacture airframe
MAN.2.2 SYS1.2 Assemble propulsion system
MAN.2.3 SYS1.2 Integrate sensors and electronics
MAN.3 SYS1 Assemble full system
MAN.3.1 SYS1.2 Combine subsystems into final product
MAN.3.2 SYS1.3 Perform initial functional checks
MAN.4 SYS1 Track configuration and documentation
MAN.4.1 SYS0.2 Track manufacturing
MAN.4.2 SYS1.4 Maintain manufacturing records
AIT
Perform assembly, integration and testing of system
AIT.1 SYS1 Assemble system for testing
AIT.1.1 SYS1.2 Integrate all subsystems
AIT.1.2 SYS1.2 Connect test interfaces
AIT.2 SYS1 Perform functional testing
AIT.2.1 SYS1.2 Verify electrical and communication paths
AIT.2.2 SYS1.2 Perform software tests
AIT.3 SYS1 Perform environmental testing
AIT.3.1 SYS1.3 Thermal testing
AIT.3.2 SYS1.3 Vibration and shock testing
AIT.4 SYS1 Verify system readiness
AIT.4.1 SYS1.3 Conduct final acceptance test
AIT.4.2 SYS1.3 Sign-off test documentation
OPS
Operate
&
&
Drone Module
OPS.1 SYS6 Receive instructions
OPS.1.1 SYS6.5 Power on delivery unit
OPS.1.2 SYS6.5 Initialize subsystems
OPS.2 SYS6 Depart from depot
OPS.2.1 SYS6.4 Prepare for departure
OPS.2.2 SYS6.1 Depart from gate
OPS.2.1.1 Check for mechanical defects
OPS.2.1.2 Check energy status
OPS.2.2.1 Undock from gate
OPS.2.2.2 Perform liftoff
OPS.2.2.3 Ascend to prescribed altitude
OPS.3 SYS6 Travel to destination (restaurant)
OPS.3.1 SYS6.1 Accelerate
OPS.3.2 SYS6.1 Cruise
OPS.3.3 SYS6.1 Loiter
OPS.3.4 SYS6.1 Decelerate
OPS.4 SYS6 Pickup order
OPS.4.1 SYS6.4 Descend to pickup altitude
OPS.4.2 SYS6.4 Initiate pickup
OPS.4.3 SYS6.6 Pickup order
OPS.4.4 SYS6.4 Confirm successful pickup
OPS.4.5 SYS6.1 Ascend to travel altitude
OR
OR
OPS.3 SYS6 Travel to destination (customer)
OPS.3.1 SYS6.1 Accelerate
OPS.3.2 SYS6.1 Cruise
OPS.3.3 SYS6.1 Loiter
OPS.3.4 SYS6.1 Decelerate
OPS.5 SYS6 Hand-over order
OPS.5.1 SYS6.1 Land
OPS.5.2 SYS5.1 Open side hatch
OPS.5.3 SYS6.6 Confirm successful pickup
OPS.5.4 SYS6.1 Ascend to travel altitude
OR
OPS.3 SYS6 Travel to destination (depot)
OPS.3.1 SYS6.1 Accelerate
OPS.3.2 SYS6.1 Cruise
OPS.3.3 SYS6.1 Loiter
OPS.3.4 SYS6.1 Decelerate
OPS.6 SYS6 Dock at gate
OPS.6.1 SYS6.1 Decrease altitiude
OPS.6.2 SYS6.1 Perform dock
OPS.7 SYS6 Renew batteries
OPS.7.1 SYS6.5 Initiate energy renewal
OPS.7.2 SYS6.5 Follow energy renewal procedure
OPS.7.3 SYS6.5 Terminate energy renewal
Finish delivery unit Operations
Communication Module
OPS.8 SYS6 Communicate with computing module
OPS.8.1 SYS6.2 Measure delivery unit status
OPS.8.2 SYS6.3 Send delivery unit data to computing module
OPS.8.3 SYS6.3 Receive updated instructions
OPS.8.4 SYS6.2 Follow updated instructions
&
OPS.8.1.1 Measure position data
OPS.8.1.2 Measure velocity data
OPS.8.1.3 Measure temperature data
OPS.8.1.4 Measure surroundings
OPS.8.1.5 Measure payload status
OPS.8.1.6 Measure energy status
OPS.8.4.1 Preprocess updated instructions
OPS.8.4.2 Distribute instructions to subsystems
OPS.8.3.1 Establish connection with computing module
OPS.8.3.2 Check for reception errors
OPS.8.3.3 Check for errors in instructions
OPS.8.2.1 Compile measurement data
OPS.8.2.2 Establish connection with computing module
OPS.8.2.3 Transmit data to computing module
OPS.8.2.4 Check for transmission errors
Computing Module
OPS.9 SYS6 Manage operations
OPS.10 SYS6 Update operations planning / route planning
OPS.11 SYS6 Send update to comunication module
OPS.10.1 SYS6.2 Gather updated information
OPS.10.2 SYS6.2 Rerun operations planning
OPS.10.3 SYS6.2 Extract new delivery unit instructions
&
OPS.9.1 SYS5.2 Receive data from weather stations
OPS.9.2 SYS5.2 Get delivery unit capabilities & position
OPS.9.3 SYS6.4 Decide on weather updates
OR
OPS.9.4 SYS6.3 Recall delivery unit to depot
OPS.9.5 SYS6.4 Continue task
OPS.9.3.1 Check if weather surpasses safety limit
OPS.9.3.2 Check if weather change is major
OPS.9.3.3 Decide on action to take
OPS.9.6 SYS5.2 Listen to traffic
OPS.9.7 SYS6.4 Detect potential collsion risk
OR
OPS.9.8 SYS6.4 Wait for traffic
OPS.9.9 SYS6.4 Avoid traffic
OPS.9.10 SYS5.1 Take manual control
OPS.9.6.1 Get other delivery unit position and intentions
OPS.9.6.2 Get ADS-B info
OPS.9.6.3 Get road vehicle info
OPS.9.6.4 Get pedestrian info
OPS.9.11 SYS5.1 Manage depot gates availability
OPS.9.12 SYS5.2 List available gates
OPS.9.13 SYS6.4 Decide on gate for delivery unit return
OPS.9.14 SYS6.4 Monitor errors
OPS.9.15 SYS6.4 Decide on error importance
OPS.9.16 SYS6.4 Recall delivery unit to depot
OPS.9.17 SYS5.1 Update operations planning
OPS.9.18 SYS6.4 Ignore error and continue
OPS.9.19 SYS5.2 Listen to communication module
OPS.9.20 SYS6.4 Detect sensor anomalies
OPS.9.21 SYS6.4 Check anomalies with redundant sensor
OPS.9.5 SYS6.4 Continue task
OPS.9.27 SYS5.1 Send delivery unit to MRO
OPS.9.22 SYS5.2 Analyse delivery unit status
OPS.9.23 SYS6.4 Decide if delivery unit needs maintenace
OPS.9.26 SYS5.1 Recall delivery unit to depot
OPS.9.24 SYS6.4 Check if delivery unit vs simulation coordinates are the same
OPS.9.28 Receive updates from restaurant
OPS.9.29 Update operations
MRO
Perform maintenance, repair and overhaul
OR
MRO.1 SYS3 Monitor system health
MRO.1.1 SYS3.1 Collect diagnostic data
MRO.1.2 SYS3.1 Detect faults
MRO.1.3 SYS3.1 Report faults
MRO.2 SYS3 Perform routine maintenance
MRO.2.1 SYS3.2 Replace wearable components
MRO.2.2 SYS3.2 Perform routine cleaning
MRO.3 SYS3 Conduct repairs
MRO.3.1 SYS3.1 Identify failed subsystems
MRO.3.2 SYS3.2 Perform component replacement
MRO.4 SYS1 Update software or firmware
MRO.4.1 SYS2.2 Apply updates
MRO.4.2 SYS1.3 Perform post-update testing
EOL
Decommission system
EOL.1 SYS0 Plan decommissioning strategy
EOL.1.1 SYS0.2 Schedule EOL procedures
EOL.1.2 SYS0.2 Notify stakeholders
EOL.2 SYS3 Safely deactivate system
EOL.2.2 SYS0.2 Remove sensitive data
EOL.2.1 SYS3.1 Disable power
EOL.3 SYS3 Handle waste materials
EOL.3.1 SYS3.3 Recycle compliant materials
EOL.3.2 SYS3.3 Dispose hazardous materials
EOL.4 SYS0 Document decommissioning process
EOL.4.1 SYS3.3 Record final system status
EOL.4.2 SYS0.2 Archive documentation
EOL.5 SYS0 Sell scrap
Legend
Mission phase
System level functions
Subsystem level functions
Sub-subsystem level functions

System IDs and definitions		
SYS0: Management SYS1: Manufacturing Department SYS2: Engineering Department SYS3: Maintenance Department SYS4: Financial Department SYS5: Operations Department SYS6: Drone	SYS0.1: Leadership SYS0.2: Documentation and Legal unit SYS0.3: Engineering Lead SYS1.1: Production unit SYS1.2: Assembly unit SYS1.3: Testing unit SYS1.4: Quality Control SYS2.1: Design unit SYS2.2: Update unit SYS3.1: Inspection unit SYS3.2: Repair unit SYS3.3: Decommissioning unit	SYS4.1: Marketing SYS4.2: Budgeting SYS5.1: Drone Operations SYS5.2: Data Collection SYS6.1: Propulsion unit SYS6.2: Data Handling module SYS6.3: Communication module SYS6.4: Computing module SYS6.5: Power Management module SYS6.6: Payload unit

Figure 2.1: Functional flow diagram


DronebezORgd
DES
Design system
DES.1 SYS0
Define system requirements
DES.1.1 SYS0.1
Gather stakeholder needs
DES.1.1.1 -Identify stakeholders
DES.1.1.2-Get in touch with stakeholders
DES.1.1.3-Take note of their needs
DES.1.1.4-Perform market analysis
DES.1.2 SYS0.1
Translate needs into functional requirements
DES.1.2.1-Formulate functional statements
DES.1.2.2-Classify requirements as "driving", "killer", and "key"
DES.2 SYS0
Perform system architecture
DES.2.1 SYS0.3
Define subsystems and interfaces
DES.2.1.1-Identify subsystems
DES.2.1.2-Identify intra subsystem interfaces
DES.2.3-Identify inputs/outputs per subsystem
DES.2.2 SYS0.3
Allocate Functions to Subsystems
DES.2.2.1-Map functions to subsystems
DES.2.2.2-Identify logic connections
DES.2.2.3-Identify unfeasible functions
DES.3 SYS0
Analyse and simulate concepts
DES.3.1 SYS0.3
Model aerodynamics, power, communication.
DES.3.1.1-Set boundary conditions
DES.3.1.2-Select key requirements for model
DES.3.2 SYS0.1
Perform Trade Studies
DES.3.2.1-Define trade metrics
DES.3.2.2-Define weight of trade metrics
DES.3.2.3-Read out output of trade
DES.4 SYS0
Verify design compliance
DES.4.1 SYS0.1
Cross-check with requirements
DES.4.1.1-Verify design respects ALL key requirements
DES.4.1.2-Identify whether driving requirements are met
DES.4.1.3-Renegotiate with stakeholder on requirements
DES.4.2 SYS0.2
Prepare verification plan
DES.4.2.1-Define verification methods per requirement
DES.4.2.2-Assign responsible engineer per requirement
DES.4.3 SYS0.2
Prepare validation plan
DES.4.2.1-Define validation methods per requirement
DES.4.2.2-Assign responsible engineer per requirement
AIT
Perform assembly, integration and testing of system
AIT.1 SYS1
Assemble system for testing
AIT.1.1 SYS1.2
Integrate all subsystems
AIT.1.1.1-Mount individual subsystems to test frame
AIT.1.1.2-Check mechanical fit
AIT.1.2 SYS1.2
Connect test interfaces
AIT.1.2.1-Setup data acquisition systems
AIT.1.2.2-Connect power and control lines
AIT.2 SYS1
Perform functional testing
AIT.2.1 SYS1.2
Verify electrical and communication paths
AIT.2.1.1-Run signal integrity tests
AIT.2.1.2-Confirm sensor/actuator feedback
AIT.2.2 SYS1.2
Perform software tests
AIT.2.1.1-Load mission software
AIT.2.1.2-Simulate operational commands
AIT.3 SYS1
Perform environmental testing
AIT.3.1 SYS1.3
Thermal testing
AIT.3.1.1-Run hot and cold cycles
AIT.3.1.2-Monitor component performance
AIT.3.2 SYS1.3
Vibration and shock testing
AIT.3.2.1- Apply random vibration profiles
AIT.3.2.2-Inspect for mechanical failures
AIT.4 SYS1
Verify system readiness
AIT.4.1 SYS1.3
Conduct final acceptance test
AIT.4.1.1-Execute full mission simulation
AIT.4.1.2-Validate all operational functions
AIT.4.2 SYS1.3
Sign-off test documentation
AIT.4.2.1-Compile test logs
AIT.4.2.2-Approve final verification report
MRO
Perform maintenance, repair and overhaul
MRO.1 SYS3
Monitor system health
MRO.1.1 SYS3.1
Collect diagnostic data
MRO.1.1.1- Log sensor readings
MRO.1.1.2- Transmit data to operations management
MRO.1.2 SYS3.1
Detect faults
MRO.1.2.1- Run health checks
MRO.1.2.2- Run error detection checks
MRO.1.2.3- Identify anomalies
MRO.1.3 SYS3.1
Report faults
MRO.1.3.1- Map anomalies to faults
MRO.1.3.2- Document faults
MRO.1.3.3- Send faults to management
MRO.2 SYS3
Perform routine maintenance
MRO.2.1 SYS3.2
Replace wearable components
MRO.2.1.1- Inspect structural joints
MRO.2.1.2- Inspect wearable components
MRO.2.1.3- Replace based wearability
MRO.2.2 SYS3.2
Perform routine cleaning
MRO.2.2.1- Inspect system unclean areas
MRO.2.2.2- Clean system
MRO.3 SYS3
Conduct repairs
MRO.3.1 SYS3.1
Identify failed subsystems
MRO.3.1.1- Run diagnostics on modules
MRO.3.1.2- Isolate faulty units
MRO.3.2 SYS3.2
Perform component replacement
MRO.3.2.1- Remove defective part
MRO.3.2.2- Replace defective part
MRO.3.2.3- Re-verify functionality
MRO.4 SYS1
Update software or firmware
MRO.4.1 SYS2.2
Apply updates
MRO.4.1.1- Upload updates
MRO.4.1.2- Validate post-update condition
MRO.4.2 SYS1.3
Perform post-update testing
MRO.4.2.1- Upload updates
MRO.4.2.2- Validate post-update condition
MAN
Manufacture system
MAN.1 SYS4
Procure materials and components
MAN.1.1 SYS4.1
Identify suppliers
MAN.1.1.1-List all needs
MAN.1.1.2-Find certified vendors
MAN.1.1.3-Request quotations
MAN.1.2 SYS1.4
Ensure quality of incoming
MAN.1.2.1-Perform incoming inspections
MAN.1.2.2-Validate products are certified
MAN.1.2.3-Validate conformity with request
MAN.2 SYS1
Fabricate subsystems
MAN.2.1 SYS1.1
Manufacture airframe
MAN.2.1.1-Shape and join structural elements
MAN.2.1.2-Verify and measure to compare to design
MAN.2.2 SYS1.2
Assemble propulsion System
MAN.2.2.1-Mount propulsive devices
MAN.2.2.2-Check electrical continuity
MAN.2.3 SYS1.2
Integrate sensors and electronics
MAN.2.3.1-Check electrical continuity
MAN.2.3.2-Connect sensors to motherboard
MAN.3 SYS1
Assemble full system
MAN.3.1 SYS1.2
Combine subsystems into final product
MAN.3.1.1-Match aubsystem interfaces
MAN.3.1.2-Fix subsystem to each others
MAN.3.2 SYS1.3
Perform initial functional checks
MAN.3.2.1-Power-On tests
MAN.3.2.2-Confirm communications paths
MAN.4 SYS1
Track configuration and documentation
MAN.4.1 SYS0.2
Track manufacturing
MAN.4.1.1-Track progress
MAN.4.1.2-Define tracking system for logging
MAN.4.2 SYS1.4
Maintain manufacturing Records
MAN.4.2.1-Log serial numbers and batch IDs
MAN.4.2.2-Document deviations and reworks
EOL
Decommission system
EOL.1 SYS0
Plan decommissioning strategy
EOL.1.1 SYS0.2
Schedule EOL procedures
EOL.1.1.1-Coordinate with operations team
EOL.1.1.2- Plan safe removal timelines
EOL.1.2 SYS0.2
Notify stakeholders
EOL.1.2.1- Send EOL notices
EOL.1.2.2- Confirm receipt and acknowledgement
EOL.2 SYS3
Safely deactivate system
EOL.2.1 SYS3.1
Disable power
EOL.2.1.1-Disconnect batteries
EOL.2.1.2- Block remote access
EOL.2.2 SYS0.2
Remove sensitive data
EOL.2.2.1- Transfer important data
EOL.2.2.2- Erase sensitive data
EOL.3 SYS3
Handle waste materials
EOL.3.1 SYS3.3
Recycle compliant materials
EOL.3.1.1- Separate reusable parts
EOL.3.1.2- Send to certified recycler
EOL.3.2 SYS3.3
Dispose hazardous materials
EOL.3.2.1- Identify E-waste items
EOL.3.2.2- Follow disposal protocols
EOL.4 SYS0
Document decommissioning process
EOL.4.1 SYS3.3
Record final system
EOL.4.1.1- Log system health at shutdown
EOL.4.1.2- Save configuration snapshot
EOL.4.2 SYS0.2
Archive documentation
EOL.4.2.1- Store design and operations files
EOL.4.2.2- Lock final revision history
EOL.5 SYS0
Sell scrap
OPS
Operations
Computing Module
OPS.9 SYS6
Manage operations
OPS.9.1 SYS5.2
Receive data from weather stations
OPS.9.1.1-Setup link with weather services
OPS.9.1.2-Get weather data in correct region
OPS.9.1.3-Log weather data
OPS.9.2 SYS5.2
Get drone capabilities & position
OPS.9.2.1-Get drone position and height
OPS.9.2.2-Get drone capabilities
OPS.9.2.3-Verify if drone capabilities are degraded
OPS.9.3 SYS6.4
Decide on weather updates
OPS.9.3.1-Compare drone capabilities with current weather
OPS.9.3.2-Check if weather surpasses drone limits
OPS.9.3.3-Check if rerouting is needed
OPS.9.3.4-Decide on final action
OPS.9.4 SYS6.3
Recall drone to depot
OPS.9.5 SYS6.4
Continue task
OPS.9.6 SYS5.2
Listen to traffic
OPS.9.6.1-Get other drones positions
OPS.9.6.2-Get ADS-B information
OPS 9.6.3-Get road drones information
OPS.9.6.4-Get pedestrian information
OPS.9.7 SYS6.4
Detect potential collsion risk
OPS 9.7.1-Compare drone position with surroundings
OPS 9.7.2-Verify if trajectories cross
OPS 9.7.3-Evaluate actions to take
OPS.9.8 SYS6.4
Wait for traffic
OPS.9.8.1-Do nothing and wait
OPS.9.8.2-Verify after wait if traffic is gone
OPS.9.8.3 -Continue with task
OPS.9.9 SYS6.4
Avoid traffic
OPS.9.9.1-Analyse best avoidance procedure
OPS.9.9.2-Proceed with avoiding
OPS.9.10 SYS5.1
Take manual control
OPS.9.10.1-Connect worker with drone
OPS.9.10.2-Make the avoiding manoeuvre
OPS.9.10.3-Disconnect worker from drone
OPS.9.11 SYS5.1
Manage depot gates availability
OPS.9.12 SYS5.2
List available gates
OPS.9.13 SYS6.4
Decide on gate for drone return
OPS.9.14 SYS6.4
Monitor errors
OPS.9.14.1-Keep in contact with drone
OPS.9.14.2-Categorise the error
OPS.9.15 SYS6.4
Decide on error importance
OPS.9.16 SYS6.4
Recall drone to depot
OPS.9.17 SYS5.1
Update operations planning
OPS.9.17.1-Run simulation again with changed variable
OPS.9.17.2-Update all relevant drones with updated task
OPS.9.18 SYS6.4
Ignore error and continue
OPS.9.19 SYS5.2
Listen to communication module
OPS.9.19.1-Get contact with module
OPS.9.19.2-Perform checksum on data
OPS.9.20 SYS6.4
Detect sensor anomalies
OPS.9.20.1-Perform logical comparison with expected and received data
OPS.9.20.2-Check logs for similar anomalies
OPS.9.21 SYS6.4
Check anomalies with redundant sensor
OPS.9.22 SYS5.2
Analyse drone status
OPS.9.22.1-Get sensor data
OPS.9.22.2-Compare with previous logs
OPS.9.22.3-Save actions in log
OPS.9.23 SYS6.4
Decide if drone needs maintenance
OPS.9.24 SYS6.4
Check if drone vs simulation coordinates are the same
OPS.9.24.1-Get drone position and sensor data
OPS.9.24.2-Get simulation drone position
OPS.9.24.3-Compare two data sets
OPS.9.26 SYS5.1
Recall drone to depot
OPS.9.27 SYS5.1
Send drone to maintenance
OPS.9.27.1-Check available maintenance gates
OPS.9.27.2-Decide on maintenance gate
OPS.9.27.3-Plan operation for drone return to gate
OPS.10 SYS6
Update operations planning / route planning
OPS.10.1 SYS6.2
Gather updated information
OPS.10.2 SYS6.2
Rerun operations planning
OPS.10.3 SYS6.2
Extract new delivery unit instructions
OPS.9.28 SYS5.2
Receive updates from restaurant
OPS.9.28.1-Decide on update importance
OPS.9.28.2-Categorize update following impact on operations
OPS.9.28.3-Handle the update
OPS.9.29 SYS5.1
Update operations
OPS.9.29.1-Send update to operation
OPS.9.29.2-Update simulation accordingly
OPS.11 SYS6
Send update to communication module
Drone Module
OPS.1 SYS6
Receive instructions
OPS.1.1 SYS6.5
Power-on delivery
OPS.1.1.1-Send command to start all subsystems
OPS.1.1.2-Follow instructions to initialize
OPS.1.2 SYS6.5
Initialize subsystems
OPS.1.2.1-Send command to start all subsystems
OPS.1.2.2-Follow instructions to initialize
OPS.2 SYS6
Depart from depot
OPS.2.1 SYS6.4
Prepare for departure
OPS.2.1.1-Check for mechanical defects
OPS.2.1.2-Check energy status
OPS.2.2 SYS6.1
Depart from gate
OPS.2.2.1-Undock from gate
OPS.2.2.2-Perform liftoff
OPS.2.2.3-Ascend to prescribed altitude
OPS.4 SYS6
Pickup Order
OPS.4.1 SYS6.4
Descend to pickup altitude
OPS.4.1.1-Find possible pickup spot
OPS.4.1.2-Activate descent mechanism
OPS.4.1.3-Stop descent at target altitude
OPS.4.2 SYS6.4
Initiate pickup
OPS.4.2.1-Check clearance to surroundings
OPS.4.2.2-Activate pickup mechanism
OPS.4.3 SYS6.6
Pickup order
OPS.4.3.1-Follow pickup procedure
OPS.4.3.2-Ensure safe system interaction
OPS.4.4 SYS6.4
Confirm successful pickup
OPS.4.4.1-Ensure correct pickup loading
OPS.4.4.1-Get confirmation on successful pickup
OPS.4.5 SYS6.1
Ascend to travel altitude
OPS.4.5.1-Activate ascent mechanism
OPS.4.5.2-Stop at target altitude
OPS.3 SYS6
Travel to destination
OPS.3.1 SYS6.1
Accelerate
OPS.3.2 SYS6.1
Cruise
OPS.3.3 SYS6.1
Loiter
OPS.3.4 SYS6.1
Decelerate
OPS.5 SYS6
Hand-over order
OPS.5.1 SYS6.1
Land
OPS.5.1.1-Find possible descent point
OPS.5.1.2-Activate descent mechanism
OPS.5.1.3-Land
OPS.5.2 SYS6.4
Initiate hand-over
OPS.5.2.1-Check clearance to surroundings
OPS.5.2.2-Activate hand-over mechanism
OPS.5.3 SYS6.6
Confirm successful hand-over
OPS.5.3.1-Ensure correct hand-over
OPS.5.3.2-Get confirmation on successful hand-over
OPS.5.4 SYS6.1
Ascend to travel altitude
OPS.5.4.1-Activate ascent mechanism
OPS.5.4.2-Stop at target altitude
OPS.6 SYS6
Dock at gate
OPS.6.1 SYS6.1
Decrease altitude
OPS.6.1.1-Find descent point
OPS.6.1.2-Activate descent mechanism
OPS.6.1.3-Stop at target altitude
OPS.6.2 SYS6.1
Perform dock
OPS.6.2.1-Receive gate ID
OPS.6.2.2-Align with gate
OPS.6.2.3-Dock at gate
OPS.7 SYS6
Renew batteries
OPS.7.1 SYS6.5
Initiate energy renewal
OPS.7.2 SYS6.5
Follow energy renewal procedure
OPS.7.3 SYS6.5
Terminate energy renewal
Communication Module
OPS.8 SYS6
Communicate with computing module
OPS.8.1 SYS6.2
Measure drone status
OPS.8.1.1-Measure position data
OPS.8.1.2-Measure velocity data
OPS.8.1.3-Measure temperature data
OPS.8.1.4-Measure surroundings
OPS.8.1.5-Measure payload status
OPS.8.1.6-Measure energy status
OPS.8.2 SYS6.3
Receive updated instructions
OPS.8.2.1-Establish connection with computing module
OPS.8.2.2-Check for reception errors
OPS.8.2.3-Check for errors in instructions
OPS.8.3 SYS6.3
Send drone data to computing module
OPS.8.3.1-Compile measurement data
OPS.8.3.2-Establish connection with computing module
OPS.8.3.2-Transmit data to computing
OPS.8.3.3-Check for transmission errors
OPS.8.4 SYS6.2
Follow updated instructions
OPS.8.4.1-Preprocess updated instructions
OPS.8.4.2-Distribute instructions to subsystems
Legend
Mission phase
System level functions
Subsystem level functions

System IDs and definitions

SYS0: Management
SYS1: Manufacturing Department
SYS2: Engineering Department
SYS3: Maintenance Department
SYS4: Financial Department
SYS5: Operations Department
SYS6: Drone

SYS0.1: Leadership
SYS0.2: Documentation and Legal unit
SYS0.3: Engineering Lead
SYS1.1: Production unit
SYS1.2: Assembly unit
SYS1.3: Testing unit
SYS1.4: Quality Control
SYS2.1: Design unit
SYS2.2: Update unit
SYS3.1: Inspection unit
SYS3.2: Repair unit
SYS3.3: Decommissioning unit

SYS4.1: Marketing
SYS4.2: Budgeting
SYS5.1: Drone Operations
SYS5.2: Data Collection
SYS6.1: Propulsion unit
SYS6.2: Data Handling module
SYS6.3: Communication module
SYS6.4: Computing module
SYS6.5: Power Management module
SYS6.6: Payload unit

Figure 2.2: Functional breakdown structure

3
Business Overview

This chapter presents an overview of the business aspect of the project, including a comprehensive analysis of the market and stakeholder ecosystem relevant to drone-based food delivery in the Netherlands, a business plan serving as a guideline for profitability, and a financial model to estimate costs, revenues, and break-even timelines. The chapter begins with Section 3.1, which evaluates the current food delivery market and emerging drone delivery sector, including competitive benchmarking. Next, Section 3.2 identifies and prioritises key stakeholders, offering a management strategy for successful project implementation. Subsequently, Section 3.3 outlines goals and strategies to ensure profitability, while Section 3.4 forecasts costs and revenues across operational stages, estimating the time required to achieve profitability. Collectively, these analyses establish the commercial viability and operational context for the DronebezORgd system.

3.1. Market Analysis

In this section, the market analysis is presented. The section begins with the analysis of the current situation of the food delivery market in Subsection 3.1.1. The size of the current and future market is identified in Subsection 3.1.2. Then all competitors of the project are listed, and their relevance is explained in Subsection 3.1.3.

3.1.1. Food Delivery in the Netherlands

For the DronebezORgd system to become a viable option, it should primarily be an all-around effective option compared to current delivery methods. The focus here is on the end-user. The customer cares mainly about two points: **price** and **speed of delivery**. This section, therefore, evaluates the current market landscape in the Netherlands. At present, Thuisbezorgd is the market leader in the Netherlands for food delivery methods, closely followed by Uber Eats, which, however, has a focus on larger cities.

Although customers typically only see the delivery fee when placing an order, the service fee charged to restaurants plays a significant role in the final price paid by the customer. To maintain profitability, restaurants often pass on these commissions by raising menu prices on delivery platforms. As a result, customers indirectly bear part of the operator's service costs through higher food prices in addition to paying delivery fees.

Profit margins within the HoReCa sector are already tight; therefore, the substantial 30% fee presents a significant challenge for operators, leaving restaurants with minimal profit. Nevertheless, this situation creates an opportunity for services such as DronebezORgd to propose more competitive alternatives. Usually, operators need to charge such a high fee to offset the delivery driver's hourly wage. One of the main benefits of DronebezORgd is that there is no need to rely on human operators, and therefore can charge a lower fee. Restaurants usually pay 30% of the average order value of €26.1[1], meaning €7.81 per delivery, a price point that DronebezORgd aims to compete with.

No precise data is available on the speed of delivery drivers in urban cities such as Delft or Amsterdam. However, it is known that e-bikes, which are the primary mode of transport for deliveries, average around 20 km/h (5.6 m/s) [3]. However, this does not account for peak hours, where this number could decrease significantly.

[1] URL/https://kopvanbrabant.nl/artikelen/thuisbezorgd-kosten-ondernemer-11954.html [cited 2 May 2025]

3.1.2. Current and Future Drone Delivery Market

The current drone delivery market value is estimated at 528.4 million euros in the year 2024, with the estimation that in 10 years, the market value will achieve 10.4 billion euros by the year 2034. From this, 35% of the market resides in US companies, and 65% of the market is capitalised by seven companies, namely DJI, Zipline, Amazon Prime Air, Wing(Alphabet), Matternet, Flytrex, and Manna Aero[2].

The value of the market showcases the high revenue potential of a drone delivery company after achieving the break-even point, considering the high amount of money capitalised by only the seven companies mentioned. Moreover, the market share achievable can be substantial in the present, as the number of successful drone delivery companies is still limited, but that is expected to change in the next 10 years. As more competitors start to better implement drone delivery into their services, starting now can bring an achievable market share of around 10%, which is a great value for a high-profit market such as drone delivery.

Lastly, the high monopoly of drone delivery services due to softer regulations in the US suggests that the concept can be implemented successfully in the market. Thus, the moment when the Netherlands relaxes regulations regarding drone delivery, DronebezORgd can be started as the first food drone delivery company operating in the Netherlands. To measure the strengths and opportunities of this market compared to the weaknesses and threats of this business, a SWOT analysis is conducted.

Table 3.1: SWOT analysis of drone delivery of food [1]

	Helpful	**Harmful**
Internal Origin	**Strengths:** - Speed & efficiency - Reduced traffic congestion (bypass road traffic) - Labor cost reduction (less manual work) - Environmentally friendly (zero emissions) - Innovation appeal (attracts early adopters)	**Weaknesses:** - Limited payload & range (4 kg, 10 km) - Weather sensitivity (rain, wind, snow) - Battery life constraints (frequent recharging) - Noise & public perception (noise pollution, privacy, safety) - High upfront investment (R&D, drone cost, software)
External Origin	**Opportunities:** - Regulatory advancements (regulations softening) - Rural & underserved areas (few viable competitors) - Partnerships with major brands (customer data sharing) - Medical & multipurpose use (diversification options) - Smart cities & AI integration	**Threats:** - Stringent regulations (EASA and global regulators) - Airspace congestion & safety risks (risk of mid-air collisions) - Cybersecurity threats (hacking & data breaches) - Public pushback (privacy, noise safety pushback) - Competition from ground robots (automated ground transport)

3.1.3. Drone Delivery Competition

While the drone food delivery market is still young and growing, there are already a few companies that have capitalised on the need for faster, cheaper, and more reliable food delivery options. Some of these companies can be seen in Table 3.2, and different approaches have been attempted to make drone delivery a profitable business. Companies like DJI focus purely on drone design and performance, optimising for specific missions such as food delivery, while most of the other companies are creating

[2] URL `https://www.gminsights.com/industry-analysis/delivery-drone-market` [cited 2 May 2025]

a service of drone delivery; thus, they have more control over the operations of the drone delivery network.

Some of these drone delivery services, like Flytrex, focus on becoming a replacement for traditional food delivery platforms like Uber Eats or Thuisbezorgd, while services like Manna are working alongside the Thuisbezorgd platform to implement drone food delivery. Finally, companies like Flyzipline are more focused on per-restaurant implementation for food delivery, for restaurants that do food delivery themselves without using food delivery apps.

Even though in the beginning the company will focus on the drone food delivery market, there is always the option of expanding the business beyond that, by starting to compete with existing successful food delivery companies like UberEats and Thuisbezorgd that use bikes and cars in the Netherlands, this being achievable by reducing the cost per delivery towards €1-2 by implementing complete automation of the delivery drone fleet while still offering drone advantages like speed, efficiency, and consistency.

Table 3.2: Financial aspects of drone food delivery competitors [1]

Company name	**Selling points**	**Cost per drone [$]**	**Cost per delivery [$]**	**Investment required[3] [$]**
Flytrex	Suburban on-demand food drone delivery in the US, FAA approved, tether system food drop, local restaurants integration through native app	850[4]	0	3M
Wing (Alphabet)	Urban, suburban, and rural areas food drone delivery worldwide, large volume automatic food delivery, station-based deployment	Unknown	13	Unknown
DJI	Drone hardware seller to food delivery companies and restaurants, one of the most advanced, developer-friendly friendly and cost-effective drone manufacturers	20,000	Unknown	30M
Matternet	Urban drone for medical and food delivery using drone stations for loading and unloading, government approved, focused on an infrastructure model	7,500	Unknown	9.5M
Flyzipline	Urban and remote locations medical and food delivery, autonomous and long range delivery, uses ”droid” drop system for precise delivery, proven track of national-wide delivery networks	Unknown	17	6.6M
Manna	Food and grocery delivery focused on super fast drop-off in low-rise suburban areas, pushing for sustainable food deliveries	Unknown	4	5.2M

Table 3.3: Performance of drone food delivery competitors [1]

Company name	**Max. speed [km/h]**	**Max. payload [kg]**	**Max. range [km]**
Flytrex	50	2.5	3.75
Wing (Alphabet)	104	2.3	6.5
DJI	72	52	16
Matternet	57.6	2	20
Flyzipline	112.65	3.62	16
Manna	129	4	2

[3] The investment required refers to the seed investment necessary for the company to implement the created design into actual drone prototypes and other facilities required to produce revenue.

[4] For the Flytrex company, the very low cost per drone and the free drone delivery fee come from their unique business strategy, which is focused on fast customer acquisition first at the expense of selling their drones and services at a loss, with the idea of becoming the most used food delivery platform before increasing their prices.

3.2. Stakeholder Analysis

To address relevant stakeholders and arrive at guidance for decision-making, stakeholder analysis is performed by understanding their needs and setting their priorities. Key project stakeholders are evaluated based on their interest and influence, both rated 1-10:

- **Investors:** Financial backers with high oversight (7,9)
- **Restaurants:** Service partners (6,7)
- **Customers:** End-users (6,5)
- **General Public:** Community impact (4,2)
- **Local Government:** Municipal oversight (5,4)
- **Airports:** Minimal involvement (1,1)
- **EASA:** Airworthiness authorities (1,9)
- **NVWA:** Food safety regulators (3,10)
- **TU Delft Faculty:** Academic evaluators (8,9)
- **DSE Group 18:** Core implementers (10,7)

By grouping the stakeholders based on their interest and influence, an overview of how to manage them can be created as follows:

- Manage Closely: Investors, regulators, faculty
- Keep Satisfied: Restaurants, customers
- Keep Informed: Local government
- Minimal Contact: General public, airports

This will be used in the project to guide the design and requirements.

3.3. Business Plan

To ensure the financial viability of the project, a business plan is made to fully demonstrate and explain the essentials of the proposed business of DronebezORgd. This includes the explanation of the business model and how DronebezORgd will generate revenue, as well as detailing the infrastructure needed for the implementation of the business in a Dutch city.

Business Model

DronebezORgd is seen as a new alternative to services like Uber Eats. This means that the business is composed of a portal through which customers place orders from partnered restaurants, and DronebezORgd picks up and delivers those orders to the customers, taking a fee of the order value as payment. Specifically, the portal will be designed such that an order is accepted or declined within a very small time frame, according to the availability of the fleet. This implies that the control centre needs to plan which orders can be accepted far into the future, to ensure orders do not go undelivered due to a lack of available drones. Furthermore, the DronebezORgd fee is set at 25% of order value, undercutting the market average of 30%.

The adoption of this business model implies that the portal through which orders are placed is well known, meaning significant monthly marketing costs will be necessary to raise awareness of the solution. It is expected, however, that customer adoption will be swift due to both the superior product of quicker and cheaper deliveries, as well as the novel nature of the product, bringing in demand through curiosity. Furthermore, the proposed business model brings in more revenue than the alternative of working as a contractor for specific restaurants and providing deliveries for a flat fee, which is expected to more than compensate for the extra marketing costs.

To further increase the reliability in the success of the business model, a number of conservative assumptions have been, and will be taken throughout the design of the BezORger and its operations. These will be seen in conservative administrative and drone costs estimates in Section 3.4, conservative design mission profiles in Subsection 11.2.1, and conservative noise estimates in Chapter 8. Together, these are expected to result in a conservative estimate of profits, helping to prove the business's viability.

Business Infrastructure

The proposed business of DronebezORgd is designed to apply to any Delft-like city, that is, any Dutch city of a similar size and population. That being said, it is expected that DronebezORgd would be a viable and potentially very profitable model in much larger cities such as Amsterdam, due to the large advantages that come with airborne delivery, such as the lack of need for road infrastructure, avoiding traffic, as well as allowing for optimised route planning. As an initial business proposal, however, DronebezORgd will design its operations specifically for the Dutch city of Delft. The reason for this is twofold. Firstly, this will act as a case study, allowing for the detailed design of Delft-specific operations, including a tailored selection of business variables such as the number of depots, their locations, and the size of the drone fleet. And secondly, it will provide a more accurate estimate of potential profits than would be obtained through simulated generic city data.

Specifically, the operations of DronebezORgd will be centred around one or two depots placed within the city of Delft. The restriction on depot number is in place both due to the limited size of Delft as well as limited initial investment capital. Each depot will contain all the necessary drone-related infrastructure, including storage location, charging infrastructure, and maintenance infrastructure. All drones will therefore leave from and return to the depot after each trip. The depot will also house at least one drone operator for every 10 drones, able to take control of any drone if the need presents itself. Additionally, the depot will contain the control centre used to give instructions to each drone. Finally, an admin centre, which may or may not be integrated into the depot, will act as an office for all administrative staff of DronebezORgd.

Furthermore, for the implementation of the business in Delft, it is assumed that work will be done with the local airspace regulators to loosen restrictions on drone operations. Currently, the restrictions on drones in almost all Dutch cities render DronebezORgd infeasible. It is therefore expected that regulations will develop favourably, and all estimates are made assuming this is the case.

3.4. Financial Model

In this section, the profitability of the business is further explored. The section starts with a list of assumptions used during the financial analysis and forecasting of the project in Subsection 3.4.1. This is then followed by the estimated costs of the business in Subsection 3.4.2, and finally ends with the return on investment analysis in Subsection 3.4.3.

3.4.1. List of Assumptions

Considering the broad scope of the financial model analysis and forecasting, particularly when long-term performance or market behaviour were involved, it was rather convenient to include several assumptions to lower the complexity of the estimation while maintaining a high fidelity of the future envisioning. To provide a simple yet reliable financial model, the following assumptions have been adopted:

- **AS-FM-01:** The production cost is said to be constant per drone.
  No extra discount due to large order of material to lower the complexity, estimated based on similar drone designs.
- **AS-FM-02:** Maintenance costs per delivery remain constant.
  In fact, this is an average value based on estimation and, taking into account the drone insurance, more severe damages to the drone shall be referred to insurance for recovery.

- **AS-FM-03:** Annual spare parts cost is proportional to fleet size.
  Considering the need for contingency, a 15% buffer of total drone production cost for spare parts is added.
- **AS-FM-04:** Insurance cost per drone is a one-off cost for the useful life of it.
  With regards to both customer safety and business profile protection, an insurance shall be needed for the drone while operating.
- **AS-FM-05:** Late deliveries are assumed to be charged with a penalty, the incident is averaged by the frequency.
  A compensation shall be provided to customers if the order is late.
- **AS-FM-06:** Electricity costs have fixed and variable components.
  Electricity cost is composed of mission consumption, which is variable per mission, as well as utilities in offices, depots and battery charging.
- **AS-FM-07:** The rent cost is linear with respect to both the size, as well as the distance to the city centre.
  This conclusion is based on a regression plane generated from data collected from several advertisement websites, in which more than 80 samples were taken for regression to estimate the rent better.
- **AS-FM-08:** Apart from fixed monthly pay, drone supervisors' pay rate varies with the number of drones this person supervises.
  The number of administrative staff is fixed, drone supervisors are scaled with the number of drones, and the number of maintenance people depends on the number of depots.
- **AS-FM-9:** For both depot construction and equipment cost, they are mostly fixed, with variation due to the number of depots and drones
  Taking into account the construction material cost and temporary freelancer hiring, as well as office and workshop equipment.
- **AS-FM-10:** Licensing and marketing costs remain fixed regardless of operational scale.
  Assuming a fixed drone license cost, monthly software licensing cost, and monthly marketing cost.
- **AS-FM-11:** The administration centre cost is independent of operations size.
  This follows from the number of administrative staff being fixed.

3.4.2. Estimated Costs

DronebezORgd's operational cost structure includes various categories essential for running a drone-based delivery service. Drone production is estimated using data from other competitors and sized for the expected payload. Drone maintenance is divided into routine check-ups, partial maintenance, and major repairs. Spare parts scale proportionally with the fleet size, assuming a percentage of the fleet will require full part replacements. Insurance covers civil liability (injury or property damage) and hull insurance for drone damage. Order reimbursements represent the costs of compensating customers for their delayed deliveries. Electricity includes energy use for the administration centre, depots, control centre, and drone charging, estimated from typical consumption per area and drone battery usage. Facility rent is based on property area and distance from the city centre, calculated using real estate data and regression modelling. The staff costs include administrative workers, drone operators, controllers, and depot personnel, which scale with the number of drones and depots. Depot construction is a one-time cost, depending on the fleet size. Equipment covers office equipment, depot maintenance tools, and drone-related hardware like chargers. Licensing includes recurring software license fees and a one-time license for legal drone operation. Finally, marketing has an annual cost based on industry standards for promotional activities and brand development.

The cost estimations have been tabulated in Table 3.4. For each cost type, the relations have been written out as term-wise expressions showing which variables they depend on. These variables have been given the following symbols: n_{drones}, the integer number of drones; $n_{\text{deliveries}}$, the integer number of deliveries; $x_{2\text{ depots}}$, a boolean which is unitary if a second depot is used and zero otherwise; and

r_{depot}, the distance of the depot to the city centre in metres. The symbol ”-” is used for (the parts of) the cost types that do not depend on any variables.

Table 3.4: Cost estimation model for different cost types and how they depend on free variables. One-time costs are in euros, and recurring costs are in euros per month [2]

Cost type	Unit	Variables	Coefficient
Drone Production	[€]	n_{drones}	10,000
Drone Maintenance	[€]	$n_{\text{deliveries}}$	0.65
Drone Spare Parts	[€]	n_{drones}	1,500
Drone Insurance	[€/month]	n_{drones}	67
Order Reimbursements	[€]	$n_{\text{deliveries}}$	0.25
Electricity	[€/month]	-	113.45
		$x_{2\text{ depots}}$	20.02
		n_{drones}	2.70
	[€]	$n_{\text{deliveries}}$	0.42
Facilities Rent	[€/month]	-	2,738.06
		$x_{2\text{ depots}}$	2,131.46
		n_{drones}	19.92
		$r_{\text{depot 1}}, r_{\text{depot 2}}$	-0.85
Staff	[€/month]	-	14,250
		$x_{2\text{ depots}}$	2,700
		n_{drones}	255[5]
Depot Construction	[€]	-	34,050
		$x_{2\text{ depots}}$	20,430
		n_{drones}	2,724
Equipment	[€]	-	4,000
		$x_{2\text{ depots}}$	1,000
		n_{drones}	140
Licensing	[€/month]	-	68
	[€]	-	10,000
Marketing	[€/month]	-	8,334

3.4.3. Estimated Return on Investment

This preliminary analysis estimates the Return on Investment (ROI) and break-even point for the DronebezORgd system across pessimistic, realistic, and optimistic scenarios. Assumptions include a constant 3% annual inflation, no market share growth for five years, and no delivery performance improvements. Key variables analysed were cruising altitude, a 15-second flight transition, and varying waiting/pickup times. Battery swaps were estimated at five minutes per five deliveries. Projected deliveries per drone per evening were based on weekly pizza orders in Delft, with drone numbers and order volumes varying across scenarios (e.g., optimistic: 19 drones, 1,728 orders/week; realistic: 16 drones, 1,648 orders/week; pessimistic: 13 drones, 1,568 orders/week). Calculated ROI ranges from 152% to 357%, with break-even periods between 1.5 and 3.8 years. The realistic scenario forecasts a 260% ROI and 2.5-year break-even.

[5] In actuality, only an increase in ten drones results in a staff cost increase of €2,550, beyond the first ten drones.

4
Conceptual Design

During the conceptual design, the work from Chapter 2 is expanded upon. Whereas before, the systems engineering foundation was established, this was later built upon to proceed with the design. Specifically, the conceptual design stage began with the selection of a specific design concept through the use of trade-offs as described in Section 4.1. This allowed for a preliminary budgeting to be done in Section 4.2. Next, a subsystem-based design approach could be established as seen in Section 4.3, after which, subsystem requirements could be formulated in Section 4.4 to set a framework for the design of each subsystem.

4.1. Trade-Offs

Trade-offs are used on multiple occasions during the design of DronezeORgd. The purpose and use of trade-off is described in Subsection 4.1.1, the trade-offs from earlier concepts are summarised in Subsection 4.1.2, and the trade-off use in the subsystem design is described in Subsection 4.1.3.

4.1.1. Trade-Offs in the Design Process

Trade-offs were a key part of the systems engineering approach employed in the development of the DronebezORgd fleet. Throughout the project, trade-off analysis was used as a decision-making tool to objectively evaluate design options, balance competing requirements, and guide the design toward a feasible solution.

In complex system design, individual components often interact or place constraints on others, and no single solution optimises all performance metrics. Trade-offs allowed the team to compare alternatives based on relevant criteria, evaluate the robustness of decisions through sensitivity analysis, and ultimately select solutions that performed best when considering the system as a whole. This method is not only useful but also necessary for the preliminary design stage and the scope of the Design Synthesis Exercise, given the limited timeframe.

A consistent graphical trade-off method was adopted across the project. First, relevant criteria were identified and weighted based on their importance to the system requirements, and where relevant, at the subsystem level, making sure that the weights sum up to 1. Each concept was then scored on a 1–5 scale for each criterion. To aid interpretation and reduce ambiguity, colour-coded scores were used, where green indicates a strong performance and red indicates unacceptable performance, decided with respect to the degree of compliance with requirements. A red score would therefore deem the design concept unfeasible and immediately remove it from further consideration in the trade-off.

This has been highlighted in Table 4.1 together with the corresponding scores and colour scales. The best-performing concept was selected, with numerical sensitivity analysis performed on the weights assigned for the trade-off criteria in cases where multiple concepts were found to perform closely. The sensitivity analysis on the final scores of each concept was deemed useful to assess robustness against changes in weighting and select a feasible concept from the trade-off process that complied with all the requirements.

Table 4.1: Ranking categories of the lifting configuration trade-off

Compliance level	Exceeds requirements	Meets requirements	Minor correctable deficiencies	Major correctable deficiencies	Unacceptable deficiencies
Score	5	4	3	2	1

4.1.2. Early Concept Trade-Offs

One of the first major decisions was the selection of a lifting configuration for the drone. Four concepts were evaluated:

- Multicopter
- Slingshot-launched fixed-wing
- Fixed-wing eVTOL with vertical and horizontal propellers
- Fixed-wing with tilt rotors

The graphical trade-off method was used to compare the options based on five weighted criteria: safety (30%), payload handling (25%), operational cost (25%), site/setup cost (10%), and sustainability (10%). These weights were chosen subjectively based on the requirements, but were further supported by the use of the Best-Worst Method. This method simplifies the problem of assigning weights to five criteria simultaneously and instead involves the selection of the most important (best - safety) and least important (worst - sustainability), followed by scoring the other criteria relative to these based on their importance. An optimisation problem can be formulated aiming to minimise the inconsistencies in the scoring and arriving at weights for the trade-off criteria. More information about this method can be found in the [1].

The slingshot-launched concept was eliminated early due to infeasibility in pickup operations, and the tilt-rotor was deemed too complex and unsafe. The two remaining candidates, multicopter and fixed-wing eVTOL, performed similarly overall, but each had distinct advantages: the multicopter excelled in payload handling and safety, while the eVTOL offered superior energy efficiency and sustainability during cruise.

Sensitivity analysis revealed that changing the weight of operational cost or sustainability favoured the eVTOL primarily due to its higher energy efficiency, while increasing the importance of safety or payload handling favoured the multicopter due to its lower complexity. Given the comparable performance and to maintain academic challenge and innovation, the fixed-wing eVTOL was selected as the final lifting configuration.

Table 4.2: Graphical trade-off of the lifting configuration of the delivery drone

Criterion **Lifting configuration**	**Site & setup cost**	**Operational cost**	**Safety**	**Sustainability**	**Payload handling**
Multicopter	nominal	low energy efficiency	nominal	loud, big battery	small size, high manoeuvrability
Slingshot-launched fixed wing	expensive infrastructure	nominal	nominal, high stability but low manoeuvrability	single use parachute	cannot land at restaurants for pickup
Fixed wing with vertical and horizontal propellers	nominal	nominal, high energy efficiency	low manoeuvrability	quiet, smaller battery	nominal

Fixed wing with tilt rotors	high production cost	costly maintenance	unsafe vertical to horizontal rotor position transition, complex	quiet, smaller battery	nominal

A second, critical early trade-off was the selection of the payload delivery method. The five considered options included:

- Landing before unloading
- Tethered release while hovering
- Low-altitude drop
- Parachute drop
- Drop-off/pick-up stations

The low-altitude and parachute drop methods were eliminated early due to safety and reliability concerns. The remaining three were evaluated based on interface with restaurant (25%), interface with customer (25%), payload protection (20%), energy management (20%), and geographic compatibility (10%).

Table 4.3: Graphical trade-off of the payload delivery method

Criterion **Delivery Concept**	**Interface with restaurant**	**Interface with customer**	**Payload protection**	**Geogr. compatab.**	**Energy management**
Landing	nominal, on-ground loading	noise, apparent safety is lower	nominal	larger (flat) space needed	nominal, on-ground while waiting
Pick-up stations	restaurant staff must walk	consumer must walk to get order – no longer home delivery	active heating of order at pick-up station	station construction	drone does not have to wait
Tethered release	trickier and more stressful to load tethered container	appears quieter and safer due to physical distance	order more exposed to environment on descent	smaller, but may tangle in wind	hovers while waiting

While tethered release scored relatively well in interface with the customer and energy management, it introduced significant mechanical complexity. Pick-up stations provide a poor customer experience as the customer would have to walk to their nearest pick-up station, contradicting the fundamental point of a pick-up delivery service. Ultimately, the landing before unloading method was selected for its simplicity and easy integration with existing restaurant loading operations, despite slightly lower performance in customer interface due to its additional noise and perceived safety.

This decision also aligned with the philosophy of reducing mechanical subsystems and simplifying software complexity, key considerations for a startup-phase product. In this case, no further sensitivity analysis was necessary, as the trade-off and qualitative reasoning together supported the final choice.

4.1.3. Trade-Offs in Subsystem Design

Trade-offs were also performed throughout the detailed design phase, including decisions on:

- Material selection (Table 5.1)
- Payload mechanism selection (Table 6.1)
- Flight computer selection (Table 6.5)

- Battery type selection trade-off (Table 7.4)

These trade-offs are discussed in their respective chapters to showcase how subsystem-specific requirements, risks, and design integration shaped final decisions.

4.2. Budget Breakdown

Technical resource budgets and contingency management are highly important in any project to aid the design of the system and its underlying subsystem. At this stage of the project, after the selection of the lifting and payload delivery configurations, an updated budget breakdown is presented.

While at this stage, the general configuration is known, allowing for an understanding of the components present, accurate budgets remain difficult to establish due to a lack of data on low MTOW eVTOLs such as the BezORger. Contingencies are used as a tool to account for these uncertainties. The contingencies are given as a percentage of the subsystem fraction. Together with the contingencies, the found budgets will be used as a starting point for the first iteration of the design. In Subsection 14.1.3, this budget will be updated to allow for reflection of the changes.

To make the budgets, the BezORger was broken down into several categories related to the functions of the system. These categories sometimes varied over the budgets based on available data. Each category was then assigned a fraction of the total budget based on market research, where possible, and engineering judgment where it was not. Contingencies were then applied by taking the element-wise product of individual contingencies.

4.2.1. Mass Budget

The mass budget is calculated through a component-wise approach, based on fractional mass data provided by [4]. The payload mass is assumed to be the design payload of 3 kg (for six pizzas), and the other component masses are derived accordingly. The data includes a pilot mass, which is eliminated by proportionally increasing all other masses. While this approach is not optimal, there is no data available for the fractional masses of small autonomous eVTOL drones, and this approach is expected to be more accurate than using the values derived for simple quadcopters. Additionally, larger contingencies are provided to components that could have different mass fractions at a lower MTOW, such as the drone's mass.

Table 4.4 summarises the mass budget. Based on the payload making up 15%, an estimated MTOW of 20 kg can be directly calculated. By including the component-wise contingencies, an estimated MTOW of **22.1 kg** is found.

Table 4.4: Preliminary Mass budget

Subsystem	Fraction [%]	Contingency [%]	Target Mass [kg]
Payload	15	0	3
Energy Source	35	10	7.68
Structure	24	10	5.28
Propulsion	21	10	4.62
Systems	5	50	1.5
Total	**100**	**10.5**	**22.1**

4.2.2. Energy Budget

The preliminary energy budget of the drone per subsystem can be derived based on preliminary estimates in the literature. As the battery is the main source of power in an eVTOL, the total available energy can be found from the battery specifications. Moreover, most of the energy is consumed by the propulsion system, which means that the other subsystems use negligible amounts of energy. The payload also consumes energy, which greatly varies from mission to mission, and can be reflected in the contingency margin.

The total energy required is estimated by looking at a battery capacity of 22,000 mA h, which is normal for the drone size considered in this project. Furthermore, based on the standard energy use fractions found through market research, the energy use of each system was established as seen in Table 4.5. The payload system was given an unusually high fraction of 5% due to the potentially high energy use of pizza handling. The payload and avionics systems were given the largest contingency. This is the case as systems such as pizza storage mechanisms and autonomous environmental perception sensors may have unusually high energy usage when compared to standard statistical values, causing uncertainties.

Table 4.5: Preliminary Power budget

Subsystem	Fraction [%]	Contingency [%]	Target Energy [mAh]
Propulsion	90	10	21,800
Avionics	5	20	1,320
Payload	5	20	1,320
Total	**100**	**11**	**24,440**

4.2.3. Cost Budget

The cost budget plays a central role in any project funding as it drives the resources required to sustain the project. However, estimating the cost breakdown of the drone per category is highly challenging due to the differences in applications and the limited data on similar drones found in the market. Moreover, cost breakdowns are not found for commercial drones to protect company interests. This means that the preliminary cost budgeting will be done on a broader basis by looking at off-the-shelf component costs. It is important to note that this method is not very reliable; however, large contingencies are put in place to arrive at a cost estimated to be conservative. Operational costs are also not included in this breakdown.

As in the other technical resource budgets, only eVTOL concepts are considered. The cost budget is centred around perception equipment, as this is the subsystem with the most reliable pricing estimates from market research. The perception subsystem is taken to have a cost of €1000. The other subsystem costs are based on scaling this cost, leading to the results found in Table 4.6. Large contingencies are chosen in every subsystem due to high uncertainties in pricing.

Table 4.6: Preliminary Cost budget

Subsystem	Components	Fraction [%]	Contingency [%]	Target Cost [€]
Propulsion	Motors, ESCs, propellers, battery	35	50	2625
Data Handling and Communications	Flight controller, GPS module, telemetry radio	10	50	750
Perception	Cameras, perception sensors	20	0	1000
Power management	Power distribution board, voltage regulators, charging infrastructure	10	50	750
Structure	Frame, landing gear and assembly supplies	30	50	2250
Total		100	40	7375

4.3. Design Approach

At this stage, the conceptual design of DronebezORgd has been completed. A trade-off has been conducted to select the lifting configuration of an eVTOL with vertical and horizontal propellers. Furthermore, the payload delivery method has been determined, further narrowing the design space. With these choices finalised, the preliminary detailed design phase can begin.

The BezORger, along with its operations, is a very complex system. The design of such a system cannot trivially be done in one step. Instead, a systems engineering approach is used to separate the design into modules and allow for concurrent design. This is done by splitting the design into various subsystems separated mostly by the functions that they provide. In the case of DronebezORgd, the next steps entail not only the design of the BezORger but also its interactions with customers and the operational structure and protocols used to run the business. This leads to a special focus in the design on elements such as operations, payload handling, and noise. These elements are therefore also considered as subsystems to be designed concurrently.

While this approach allows for an effective use of resources, as subsystems can be designed by specialised teams for maximal efficiency, the design must then be integrated iteratively to ensure a coherent design. For this to be possible, all subsystems need to be designed in a parametric nature, with the ability to quickly adapt to changing parameters. Iterations can then be performed across all subsystems until a converged design is reached.

As a first step in the design, the subsystems of DronebezORgd are clearly defined in Subsection 4.3.1, and their interfaces and interdependencies are described in Subsection 4.3.2, laying the foundation upon which the next design phase begins.

4.3.1. Subsystem Breakdown

Given the increased complexity present in the following stages of design, the design of the drone was divided into distinct yet interrelated subsystems. Each subsystem, while interrelated with the others, is responsible for the design of distinct parts and functions of the design. As has already been mentioned, additional subsystems specific to DronebezORgd and its success have been added. All the identified subsystems, along with their responsibilities, can be found in the list below.

- **Structures and Materials:**
  The structures and materials subsystem is responsible for providing load transfer on the BezORger while accommodating every other subsystem on board. Specifically, the subsystem designs the dimensions and materials of the airframe structure upon which the drone is built.
- **Payload Handling and Avionics:**
  The payload handling and avionics subsystem is responsible for all the functions related to payload handling, flight control, localisation, and communication. This entails the design of all non-propulsion-related mechanical components of the drone and its avionics. The subsystem is therefore responsible for the design of any payload-related mechanisms and other moving parts, as well as the perception, flight control, and communication systems.
- **Propulsion and Powertrain:**
  The propulsion and powertrain subsystem is responsible for converting stored energy into useful thrust for the BezORger. This entails the design of the propellers, the battery, and accompanying electrical components.
- **Aerodynamics:**
  The aerodynamics subsystem is responsible for ensuring favourable flight characteristics of the BezORger. This is done through the design and analysis of the wing planform and airfoil, as well as the full drone configuration.
- **Stability and Control:**
  The stability and control subsystem is responsible for ensuring the static and dynamic stability of all 6 axes of motion of the drone. This entails the sizing of stabilising surfaces such as a tail, finding the ideal wing position, and ensuring C.G. ranges fall within acceptable bounds.
- **Performance:**
  The performance subsystem can be seen as an integration of those that come before it. Performance is responsible for evaluating the drone's performance and ensuring compliance with mission requirements. To do this, the subsystem is tasked with modelling each flight phase and the performance limits of the BezORger.

- **Operations:**
  The operations subsystem is responsible for the operational fleet utilisation design. This entails the design of a program capable of planning fleet deployment based on real-time orders and the operational protocols accompanying it.

4.3.2. Subsystem Interface Management

In a multidisciplinary system like the DronebezORgd fleet, effective integration across subsystems is crucial to ensure functional, reliable, and efficient design. Each subsystem has its unique design requirements, but their performance and constraints are tightly interconnected.

To facilitate this integration, a structured interface management process was applied. This process involved defining inputs and outputs for each subsystem, identifying dependencies, and enabling communication flows early in the design process. This approach ensured that design decisions in one subsystem accounted for the constraints and capabilities of others, while redundant or conflicting requirements were resolved early.

To map these interdependencies, a Design Structure Matrix (DSM), also known as an N^2 chart, was constructed (Figure 4.1). Each row and column represents a subsystem. A mark at the intersection of row i and column j indicates that subsystem i requires input from subsystem j.

This chart was iteratively reviewed and refined as the design progressed, serving as a critical planning tool throughout both the conceptual and preliminary design phases. In addition to the pairwise interfaces mapped in the N^2 chart, several global system inputs and outputs govern the overall architecture and integration strategy. These system-level inputs are primarily derived from the system requirements analysis and include the mission profile (e.g., range, payload capacity), operational constraints (e.g., ambient noise limits, on-ground handling conditions), and financial targets (e.g., startup budget, break-even targets, sustainability goals).

By establishing and managing these interfaces early, the team ensured a convergent design process, in which all subsystems evolved cohesively under a shared set of performance targets and integration constraints. This systems-level coherence allowed design iterations to remain focused, compatible, and aligned with the project's mission objectives.

4.4. Subsystem Requirements

After the definition of the subsystems in Subsection 4.3.1, a lower level of requirements can be defined, subsystem requirements. Note that not all system requirements are broken down into subsystem requirements, only those that relate specifically to subsystem capabilities. The subsystem requirements for each subsystem can be found below.

Structures & Materials

Table 4.7: Subsystem requirements for the structures and materials subsystem

Identifier	Subsystem Requirement	Arises From
RQ-SM-01	The structure shall allow for quick partial repairs to not interrupting the operating framework on a large scale.	RQ-01-03-03
RQ-SM-02	The structure shall be water-resistant during light rain operations.	RQ-01-03-03
RQ-SM-03	The structure shall withstand the maximum loading factor while maintaining structural integrity.	RQ-01-03-04
RQ-SM-04	The structure shall consider all loading cases with a safety factor of 1.5 for contingency cases.	RQ-01-03-04
RQ-SM-05	The structure shall not fail because of axial loading.	RQ-01-03-04
RQ-SM-06	The structure shall not fail because of shear loading.	RQ-01-03-04
RQ-SM-07	The structure shall not fail because of bending stress.	RQ-01-03-04

RQ-SM-08	The structure shall not fail because of Euler buckling.	RQ-01-03-04
RQ-SM-09	The structure shall not fail because of fatigue throughout all planned mission profile phases.	RQ-01-03-04
RQ-SM-10	The fuselage structure shall be able to accommodate all payloads, including pizzas, batteries, and sensors.	RQ-01-04-06
RQ-SM-11	The drone structure shall withstand a payload of 2.5 kg	RQ-01-04-07
RQ-SM-12	The structure shall have at most a 2 by $2\,\mathrm{m}^2$ of projection area.	RQ-01-04-08
RQ-SM-13	At least 70% of the structural mass of a drone shall be recyclable by currently available technology.	RQ-04-01-01
RQ-SM-14	The clearance of the propellers to the fuselage shall be at least 25 cm.	RQ-09-03-02

Payload Handling & Avionics

Table 4.8: Subsystem requirements for the payload handling subsystem

Identifier	Subsystem Requirement	Arises From
RQ-PL-01	The payload handling subsystem shall allow for the loading of payload for multiple, separate deliveries.	RQ-01-03-01
RQ-PL-02	The payload handling subsystem shall allow for the unloading of payload for multiple, separate deliveries.	RQ-01-03-03
RQ-PL-06	The wingspan reduction mechanism shall retain the structural integrity of the wing in all flight conditions.	RQ-01-03-04
RQ-PL-03	The payload bay shall be able to accommodate a payload volume of at least 45 by 45 by $30\,\mathrm{cm}^3$.	RQ-01-04-06
RQ-PL-04	The payload bay shall be able to accommodate a payload mass of at least 2.5 kg.	RQ-01-04-07
RQ-PL-05	The wingspan reduction mechanism shall reduce the footprint of the drone to fit within a 2 by $2\,\mathrm{m}^2$ square.	RQ-01-04-08
RQ-PL-07	The wingspan reduction mechanism shall be able to lock in both final positions.	RQ-01-04-08
RQ-PL-08	The wingspan reduction mechanism's locks shall have N+1 redundancy.	RQ-02-02-01
RQ-PL-09	Each sensor group of the payload handling subsystem shall have N+1 redundancy.	RQ-02-02-01
RQ-PL-10	The payload handling subsystem shall enable the internal surface of the payload bay to be washable.	RQ-13-01-01
RQ-PL-11	The payload bay shall be made of food-safe materials as defined by the Dutch National Institute for Public Health and the Environment.	RQ-13-01-02
RQ-PL-12	The payload handling subsystem shall prevent ingress of dust, insects, and moisture while in flight.	RQ-13-01-03
RQ-PL-13	The payload handling subsystem shall be able to maintain an internal temperature of at least 63 °C throughout a mission.	RQ-13-02-01
RQ-PL-14	The payload handling subsystem shall be able to record the internal temperature of the payload bay while carrying the payload.	RQ-13-02-02

Propulsion & Powertrain

Table 4.9: Subsystem requirements for the power and propulsion subsystem

Identifier	Subsystem Requirement	Arises From
RQ-PP-01	The vertical propeller diameter shall not exceed 45 cm.	RQ-01-04-08
RQ-PP-02	The horizontal propeller diameter shall not exceed 50 cm.	RQ-01-04-08

RQ-PP-03	The vertical propeller(s) shall be able to provide a thrust of twice the maximum take-off weight under nominal operational conditions.	RQ-01-03-03
RQ-PP-04	The horizontal propeller(s) shall be able to provide a thrust of half the maximum take-off weight under nominal operational conditions.	RQ-01-03-03
RQ-PP-05	The motor for each propeller shall be able to provide the maximum required rotation rate.	RQ-01-03-03
RQ-PP-06	The motor for each propeller shall be able to provide the maximum required torque.	RQ-01-03-03
RQ-PP-07	The motor for each propeller shall be able to operate at the maximum power draw by the propeller.	RQ-01-03-03
RQ-PP-08	The motor for each propeller shall be able to operate at the maximum thrust generated.	RQ-01-03-03
RQ-PP-09	The battery shall be able to provide the required mission energy after accounting for efficiency losses.	RQ-01-04-03
RQ-PP-10	The battery shall be able to provide the maximum required voltage by the propellers.	RQ-01-04-03
RQ-PP-11	The battery shall be able to provide the maximum total power required by all propellers and electrical components combined.	RQ-01-03-03

Aerodynamics

Table 4.10: Subsystem requirements for the aerodynamics subsystem

Identifier	Subsystem Requirement	Arises From
RQ-AERO-01	The aerodynamics shall be optimised for range.	RQ-01-04-03
RQ-AERO-02	The wing shall have a maximum wing span of 3 m.	RQ-01-04-08
RQ-AERO-03	The drone model shall be verified using a CFD software.	RQ-14-02-01

Stability & Control

Table 4.11: Subsystem requirements for the stability and control subsystem

Identifier	Subsystem Requirement	Arises From
RQ-SC-01	The drone shall remain stable in all 6 axes for the C.G. range between 20%-60% MAC from the LEMAC	RQ-01-04-07
RQ-SC-02	The drone shall remain controllable in all 6 axes for the C.G. range between 20%-60% MAC from the LEMAC.	RQ-01-04-07
RQ-SC-03	The drone shall remain stable in all 6 axes for wind gusts up to 7 m/s.	RQ-02-01-04
RQ-SC-04	The drone shall remain controllable in all 6 axes for wind gusts up to 7 m/s.	RQ-02-01-04
RQ-SC-05	The drone shall remain stable in all 6 axes when one of the propellers fails.	RQ-03-04-02
RQ-SC-06	The drone shall remain controllable in all 6 axes when one of the propellers fails.	RQ-03-04-02

Performance

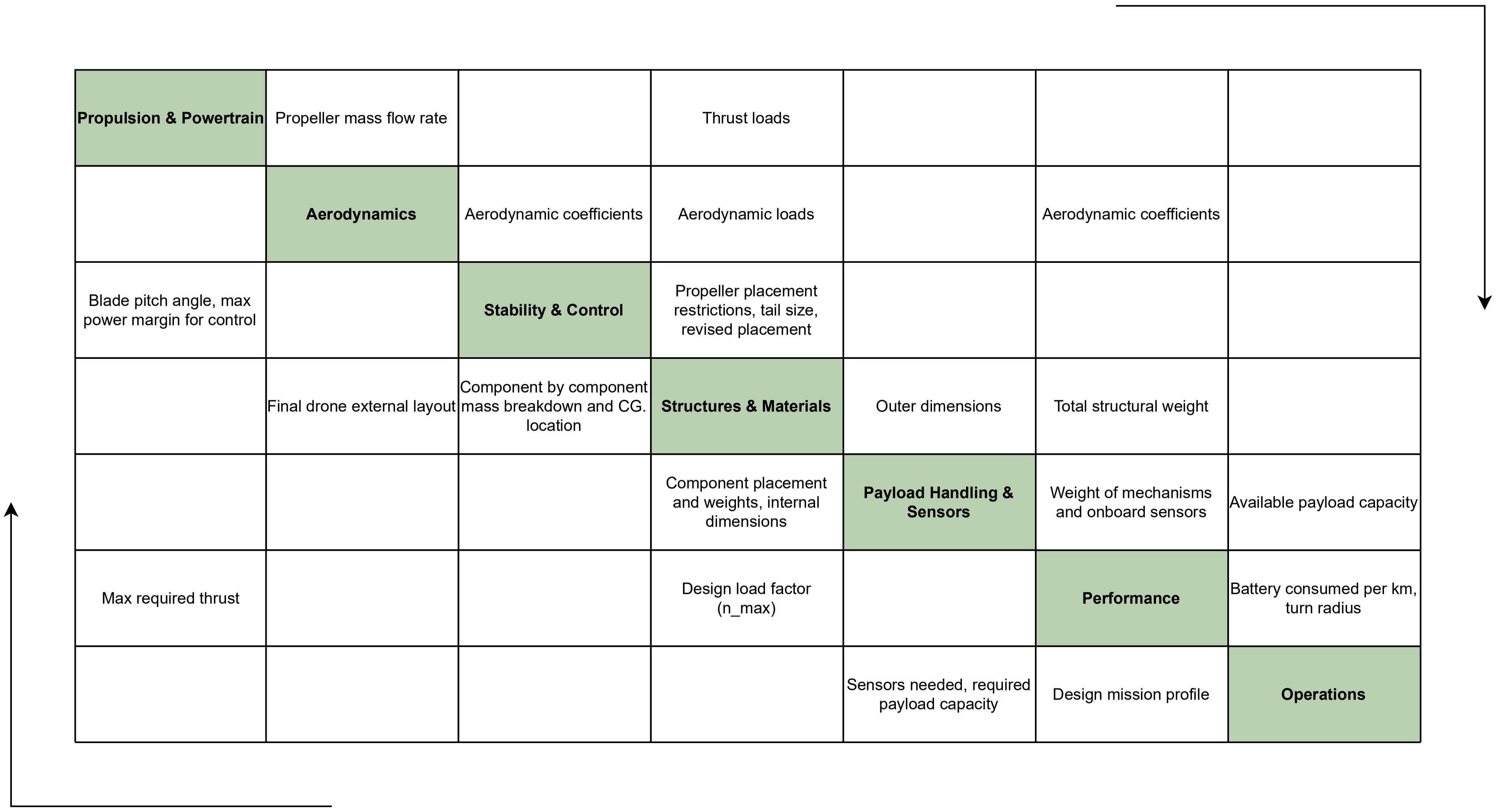
Table 4.12: Subsystem requirements for the performance subsystem

Identifier	Subsystem Requirement	Arises From
RQ-PERF-01	The drone shall have a range of at least 5 km when at maximum payload weight.	RQ-01-04-03
RQ-PERF-02	The drone shall have a total range of at least 10 km.	RQ-01-04-02

Operations

Table 4.13: Subsystem requirements for the operations subsystem

Identifier	Subsystem Requirement	Arises From
RQ-OP-01	Mission Planning shall comply with all other requirements.	RQ-01-05-01
RQ-OP-02	Path Planning shall comply with all other requirements.	RQ-01-05-02
RQ-OP-03	A drone shall not deviate more than 1 m from the planned path during nominal operations.	RQ-01-05-03
RQ-OP-04	Path planning shall plan paths which keep a distance of at least 3 m from any obstacle, barring proximity operations at the depot, the restaurant, and the customer locations.	RQ-01-05-06
RQ-OP-05	Mission planning shall plan routes which keep the drone within communications range with the depot at all times.	RQ-01-05-08
RQ-OP-06	The depot shall recall the fleet if the average temperature in the city goes below 0 °C.	RQ-02-01-03
RQ-OP-07	The depot shall recall the fleet if the average wind speed in the city exceeds 7 m/s.	RQ-02-01-04
RQ-OP-08	Each depot shall have at least 1 drone operator personnel on standby.	RQ-02-02-02
RQ-OP-09	Mission Planning shall compute the high-level drone scheduling in less than 30 s	RQ-02-03-04
RQ-OP-10	Path Planning shall compute the drone routing in less than 12 s.	RQ-02-03-03
RQ-OP-11	The operational profit shall offset start-up costs within 5 years.	RQ-03-02-01
RQ-OP-12	The objective function of Mission Planning shall maximise profit, within the model constraints.	RQ-03-02-02
RQ-OP-13	The objective function of Path Planning shall maximise profit while reducing public disturbance, within the model constraints.	RQ-03-02-02
RQ-OP-14	Operations shall lead to minor accidents, defined as incidents that cause damage but allow continuation of drone operations, in fewer than 1 in 100 deliveries.	RQ-03-04-01
RQ-OP-15	Operations shall lead to major accidents, defined as incidents that cause damage and disallow continuation of drone operations and/or injury to people, in fewer than 1 in 100,000 deliveries.	RQ-03-04-02
RQ-OP-16	The operational energy consumption shall be lower than that of current alternatives for equal operational revenue.	RQ-04-01-02
RQ-OP-17	Mission planning shall constrain the earliest delivery time to the scheduled delivery time.	RQ-06-01-01
RQ-OP-18	Path planning shall provide delivery time estimates with a 90% confidence interval of 120 s.	RQ-06-02-01
RQ-OP-19	Operations shall restrict the number of drones within a 10 m radius at any location other than the depot to 2.	RQ-09-03-01
RQ-OP-20	Path planning shall not plan paths through designated no-fly zones.	RQ-10-01-01



Propulsion & Powertrain	Propeller mass flow rate		Thrust loads			
	Aerodynamics	Aerodynamic coefficients	Aerodynamic loads		Aerodynamic coefficients	
Blade pitch angle, max power margin for control		**Stability & Control**	Propeller placement restrictions, tail size, revised placement			
	Final drone external layout	Component by component mass breakdown and CG. location	**Structures & Materials**	Outer dimensions	Total structural weight	
			Component placement and weights, internal dimensions	**Payload Handling & Sensors**	Weight of mechanisms and onboard sensors	Available payload capacity
Max required thrust			Design load factor (n_max)		**Performance**	Battery consumed per km, turn radius
				Sensors needed, required payload capacity	Design mission profile	**Operations**

Figure 4.1: N^2 chart highlighting subsystem interdependencies

5 Structures & Materials

The structural integrity and material selection of the drone are critical to its performance, safety, and operational feasibility. This chapter presents a systematic approach to analysing and designing the drone's load-bearing components under given flight and payload conditions from project constraints, as well as lowering computational efforts with practical simplifications, taking into account the impact of these assumptions, in Section 5.1. In Section 5.2, material selection is evaluated through a weighted trade-off matrix, prioritising specific strength, manufacturability, and cost.

Given the complexity of full-scale finite element analysis (FEA), the design adopts an idealised structural framework that is explained in Section 5.3: the drone's wings, fuselage, and empennage are discretised into boom elements. Then, computations can be performed in an easier manner using the analytical formulations, as described by Megson [5]. This method, combined with conservative approximations on structural loads, provides insights into stress, bending, torsion, and displacement behaviour. To address limitations of the boom theory, complementary ANSYS simulations are implemented, ensuring a more comprehensive and realistic presentation of load distribution in the structure.

The chapter concludes with verification and validation of the employed methods in Section 5.4. Numerical benchmarks, analytical models, and recommendations for future refinement, including dynamic loading analysis and local stress concentration studies, are presented in this section.

5.1. Design Approach & Assumptions

An idealisation method was adopted to design the structural system of the drone. Namely, each main component of the drone (wing, fuselage, tail) was divided into several cross sections, each one composed of booms. By discretising the whole assembly into several arrays of booms, an easier computation from the formulas in the book by Megson [5] can be performed, and, taking into consideration some underestimates and overestimates of each parameter, a better modelling for the entire structure can be made. Note that the number of sections and booms has a physical meaning; booms are spar caps or stringers, and the sections are, in the case of the wing, the ribs. Meaning that increasing the number of sections to infinity will not result in a better approximation of weight.

- **AS-SM-01:** The drone is symmetric, and joints are modelled as points.
  For simplicity of analysis, unsymmetrical drone designs are not considered, and the structure is simplified so that the connecting points do not introduce extra load.
- **AS-SM-02:** Idealised, discretised boom structures were used for numerical Finite Element Analysis (FEA).
  The drone's load-bearing components are modelled as interconnected 1D boom elements carrying loads.
- **AS-SM-03:** The material used is perfectly manufactured.
  Material properties are identical in the specified direction (considering that composites might be used), ignoring imperfections common in any materials.
- **AS-SM-04:** The lift on the wings follows the empirical formula for elliptical lift distribution.
  Lift is modelled as varying smoothly along the wingspan according to $L(y) = L_0\sqrt{1-(2y/b)^2}$, where b is wingspan and L_0 is root lift.

- **AS-SM-05:** The drag distribution on the wings is constant along the wingspan.
  Drag per unit span is treated as uniform rather than varying with local airfoil characteristics or Reynolds number to simplify the load cases.
- **AS-SM-06:** All remaining loads are modelled as point loads applied on their corresponding CG location.
  Distributed loads in real life are summed into single force vectors acting at component centres of gravity.
- **AS-SM-07:** Wing folding mechanism does not introduce extra structural complexity.
  Hinge joints are treated as perfect pinned connections without play, friction, or misalignment effects.
- **AS-SM-08:** The cross-sections of the structures are assumed to be rectangular.
  All structural members are modelled with rectangular profiles of uniform thickness, ignoring airfoil shape effects for simplicity of boom analysis.
- **AS-SM-09:** The structure is continuous anywhere.
  No discontinuities between components, all connections are mathematically perfect.
- **AS-SM-10:** Bending, shear, and torsion calculations were based on formulas for idealised structures.
  Classical Euler-Bernoulli beam theory and torsion formulas are applied to the boom model.
- **AS-SM-11:** Dynamic and fatigue loads were not considered when modelling the loading conditions of the structures.
  Only static peak loads are analysed, ignoring vibration, gust response, or cyclic loading effects on lifespan and performance.
- **AS-SM-12:** It is assumed that no local stress concentration effect exists for simplification purposes.
  Geometric discontinuities are ignored even though they might exist on a small scale, and nominal stresses are used without concentration factors.
- **AS-SM-13:** For sizing iteration, failure modes such as axial stresses, shear stresses, and Euler buckling are considered, and sized for the most critical mode.
  Structural members are designed to resist the worst-case failure among tensile yield, shear yield, or column buckling.

5.2. Material Selection

The structural design requires careful material selection due to significant variations in mechanical properties, weight, and manufacturability among candidate materials. Several materials for trade-off analysis were chosen based on an evaluation of similar drones in the market and common aerospace material selections. Using a weighted decision matrix, these materials are evaluated against critical design criteria: specific strength σ_y/ρ (30%), specific stiffness E/ρ (25%), manufacturability (15%), maintainability (15%), and cost (15%). The results and comparison are presented in Table 5.1.

Table 5.1: Graphical trade-off of the material selection

Criterion **Material option**	**Specific strength - 30%**	**Stiffness - 25%**	**Manufacturability - 15%**	**Maintainability - 15%**	**Material cost - 15%**
Al 6061-T4	53.7 kN m/kg	25.52 MN m/kg	easy machining, large-scale manufacturing possible	welding/bolt repairs, relatively simple repair process	low, possible discount for large orders

CFRP Uni	1,555.56 kN m/kg	132.22 MN m/kg	skilled labour needed	specialist repair only, delamination risks	very expensive, large order discount not available
CFRP Multi	2,285.64 kN m/kg	133.7 MN m/kg	moderate skill floor required for manufacturing	patch repairs possible with training	expensive, large order discount not available
FFRP Multi	32.25 kN m/kg	15.32 MN m/kg	simple hand layup, however somewhat time-consuming	bonding repairs, limited load capacity	moderate price, volatile supply chain because of niche demand

As can be seen in the trade-off matrix, Aluminium 6061-T4 is the optimal choice for this drone design, with the best combination of strength and stiffness with manufacturability and maintenance. CFRPs, while offering excellent performance, are too expensive and require specialised labour for manufacturing and maintenance. FFRP, on the other hand, in spite of being environmentally sustainable, has a very low stiffness and specific strength. Aluminium is also recyclable, allowing compliance with RQ-SM-13.

5.3. Analysis

In this section, the boom analysis method is described in Subsection 5.3.1, followed by an explanation of the complementing ANSYS procedure in Subsection 5.3.2. Then, the failure modes used for sizing are outlined in Subsection 5.3.3. Finally, the results of the sizing are presented in Subsection 5.3.4.

5.3.1. Boom Analysis

The initial stage of the internal structure analysis involved discretising the airframe by grouping distributed masses into representative point areas. This simplification was justified by the structure's homogeneous material composition. Two different categories of booms were needed and defined: corner booms, which were reinforced to manage critical loads at corners and thus show higher mass density, and standard booms that carry normal loads.

The contribution of the skin panels to the overall structural stiffness was accounted for as described by Megson [5]. The equivalent boom area from skin between any two selected points was calculated using the equation below, where b represents the structural distance between nodes, t for skin thickness, and σ_1, σ_2 the respective stresses at these points. This formulation effectively includes the skin's load-transfer capability between primary structural members.

$$B_1 = \frac{bt}{6}\left(2 + \frac{\sigma_2}{\sigma_1}\right) \tag{5.1}$$

Normal bending stresses and shear stresses at any point in the cross-section of the beam are calculated using Equations 5.2 and 5.3 [5].

$$\sigma_z = \frac{(M_x \cdot I_{yy} - M_y \cdot I_{xy})y + (M_y \cdot I_{xx} - M_x \cdot I_{xy})x}{I_{xx}I_{yy} - I_{xy}^2} \tag{5.2}$$

$$q_s = -\frac{V_y I_{yy} - V_x I_{xy}}{I_{xx} I_{yy} - I_{xy}^2} \left[\sum_{r=1}^{n} B_r y \right] - \frac{V_x I_{xx} - V_y I_{xy}}{I_{xx} I_{yy} - I_{xy}^2} \left[\sum_{r=1}^{n} B_r x \right] + q_{s0} \tag{5.3}$$

Equation 5.2 can be simplified by setting $I_{xy} = 05.1$ and removing M_y (i.e., if there is only one bending moment applied). These simplifications lead to:

$$\sigma_z = \frac{My}{I} \tag{5.4}$$

Similarly, in Equation 5.3, the last term can be removed, since all analysed cross-sections are closed. A combination of these equations is used during the boom analysis to measure the stresses due to different loadings. They are also used in the verification and validation sections of this chapter, Subsection 5.4.1 and Subsection 5.4.2.

To make the computational process easier, all major components, including the fuselage, wings, and tail, were idealised as tapered box structures. This approach simplified the calculation of normal stresses, shear flow distributions, and torsional loading across various cross-sections, while also allowing visualisation of displacement patterns under operational load cases.

Moreover, a mass-optimisation code loop was implemented to systematically reduce the total mass, including material volume and skin thickness, without compromising structural integrity. This optimisation had two main purposes: it addresses the financial requirement of minimising production costs while supporting the project's sustainability objectives through reduced material usage.

Nevertheless, it should be noted that while the boom method provides an efficient means of approximating global load distributions, it is unable to accommodate load and stress transfer between different components of the structure, such as the load from the wings on the fuselage. This limitation is particularly evident at the wing-to-fuselage and vertical propeller attachment points. To address this shortcoming, a complementary finite element analysis was conducted using ANSYS. The combination of these two approaches, namely analytical boom calculations for global behaviour and numerical analysis for localised effects, provides a comprehensive understanding of the structure's performance. The application of the ANSYS method is explored in greater depth next.

5.3.2. ANSYS Procedure

The ANSYS simulation was used to account for the sections intersecting the wing and the propellers. The reason for this approach was to account for the upwards bending moment of the wings/propellers at the left and right sides of the section. With the boom idealisation, forces and moments can only be applied at the centre of the section; this is why ANSYS was needed while designing the fuselage. Furthermore, if the moments of the wing/propeller were to be applied at the centre of the section, they would cancel each other out as symmetry was assumed. The superposition principle was used to add the stresses computed in ANSYS to the stresses already computed using the boom method.

While preparing the FE model within ANSYS, some specific assumptions were made:

1. The section is symmetric.
2. The loads applied are symmetric.
3. The change of loads will result in a linear change of stress.
4. The change of stresses with the section thickness is piecewise linear.

In Figure 5.1, the applied loads on the section can be seen. It shows the moment decomposed into a force distribution; this was done to prevent the moment from being applied to a single element.

The loads only apply at the upper part of the section, as this is where the wing/propellers interact with the fuselage. The magnitude of this load is set to a unitary load, as the assumption was made that the stress increases linearly with the load applied.

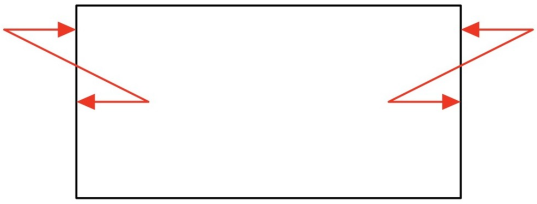
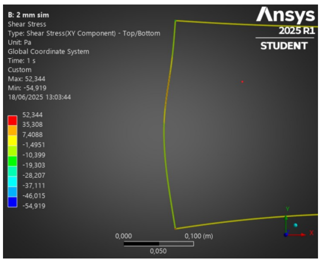


Figure 5.1: ANSYS applied loads



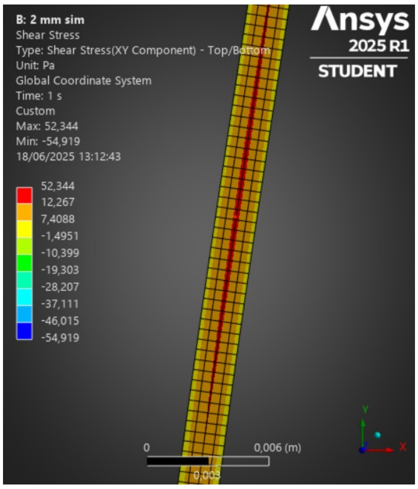
B: 2 mm sim
Shear Stress
Type: Shear Stress(XY Component) - Top/Bottom
Unit: Pa
Global Coordinate System
Time: 1 s
Custom
Max: 52,344
Min: -54,919
18/06/2025 13:03:44

Figure 5.2: ANSYS zoomed out overview

Furthermore, the simulation was repeated for three different thicknesses: 2 mm, 3 mm, and 5 mm. This was done as the section's thickness depends on many factors, and this was the reason for the fourth assumption.

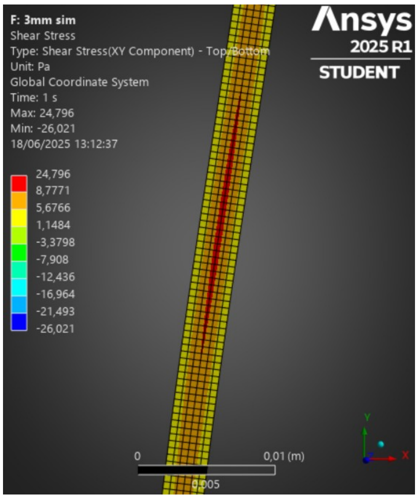
Figure 5.2 shows the left side of a section with the unit load applied. This is to show that the shear stress is concentrated at the corners and the region where the load is applied. In future design, the corners will be rounded, so the only stress that will be taken from this analysis is the one from the applied load region.

Figure 5.3, 5.4 and 5.5 show the shear stress for the section thickness of 2 mm, 3 mm, and 5 mm respectively. These figures are zoomed to the section of interest, i.e., where the load gets applied. Keep in mind that for the following figures, the scale changes on each figure.



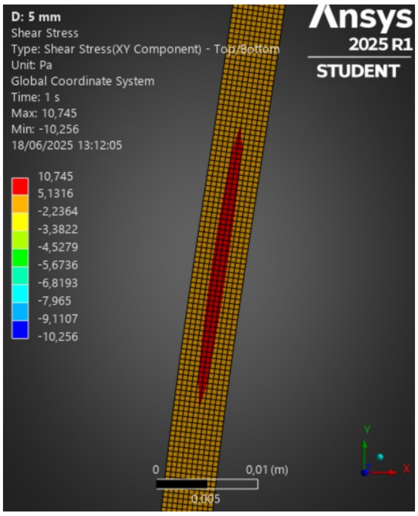
B: 2 mm sim
Shear Stress
Type: Shear Stress(XY Component) - Top/Bottom
Unit: Pa
Global Coordinate System
Time: 1 s
Custom
Max: 52,344
Min: -54,919
18/06/2025 13:12:43

Figure 5.3: ANSYS shear stress, section of 2 mm



F: 3mm sim
Shear Stress
Type: Shear Stress(XY Component) - Top/Bottom
Unit: Pa
Global Coordinate System
Time: 1 s
Max: 24,796
Min: -26,021
18/06/2025 13:12:37

Figure 5.4: ANSYS shear stress, section of 3 mm



D: 5 mm
Shear Stress
Type: Shear Stress(XY Component) - Top/Bottom
Unit: Pa
Global Coordinate System
Time: 1 s
Max: 10,745
Min: -10,256
18/06/2025 13:12:05

Figure 5.5: ANSYS shear stress, section of 5 mm

The maximum stress has been saved and plotted against the section's thickness to quickly estimate the stress with a changing thickness for quick iterations. The explained plot can be seen in Figure 5.6.

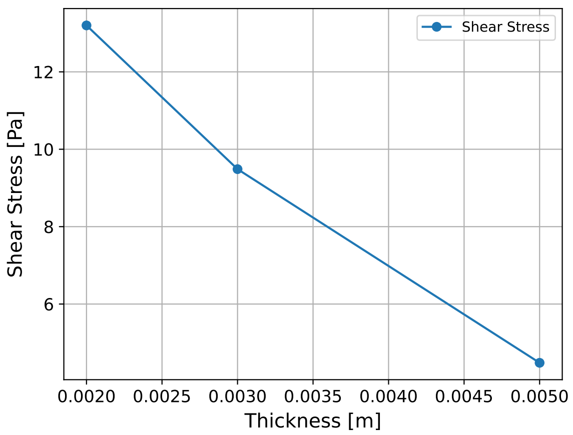


Figure 5.6: ANSYS max shear stress to thickness

This way, the shear stresses can be easily estimated for a certain thickness. This becomes very useful during the iteration loops, which are explained in Subsection 5.3.3.

5.3.3. Failure Modes & Iterations

In order to size the structure for minimal mass, a series of different failure modes was selected. The first two modes are derived from Equation 5.2 and Equation 5.3, checking that the analysed cross-section doesn't have its stress exceeding the yield stress of the material. The third failure mode examined was an Euler-buckling check [5].

For the material failure, the values of stress are compared to the material properties with a safety factor added. The buckling check for booms is described through the following formula:

$$\sigma_{CR} = \frac{\pi^2 E}{(L_e/r)^2} \tag{5.5}$$

The effective length l_e of the boom determines the boundary condition of the beam (whether it is pinned at both ends, one end, fixed, etc). In the case of the wing, where one end is fixed and the other is free hanging, the ratio is $l_e/l = 2$ [5, p. 272]. The r term is the radius of gyration of the cross-sectional area, assumed circular in the simulation.

The following iteration loop has been implemented to minimise the structural weight of the drone. The iteration starts from a given boom area, which is relatively large and certain not to fail under any conditions. If the structure passes all the above-described failure modes, it goes on to reduce the area by a given factor. Once the area has been reduced, the check on failure modes runs again. It repeats this process until it fails and returns the last geometry that did not fail. There are two edge cases, namely, if only the spar caps (corner booms) fail, meaning that the stringers are still over-designed, and the opposite case for stringer failure. So the iterations only stop if both the spars and stringers fail. During the iterations, as the area of the booms changes, the forces on the structure are recomputed every time, as the weight changes.

5.3.4. Results

The results are presented here for both the wing and fuselage. The tail has not been included as its analysis method is the exact same as the wing, but rotated by a certain number of degrees.

Wing/Tail Analysis

As mentioned, the wing was modelled as a tapered rectangle. In Figure 5.7, this wing structure can be seen.

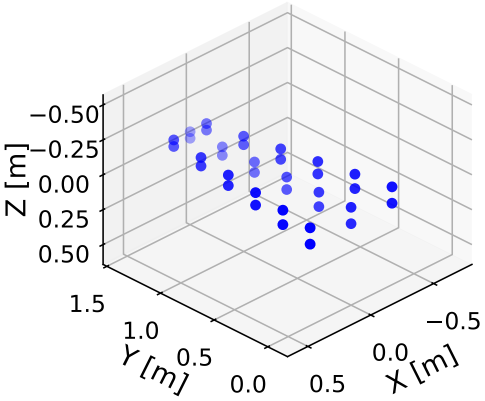


Figure 5.7: Wing structure idealized

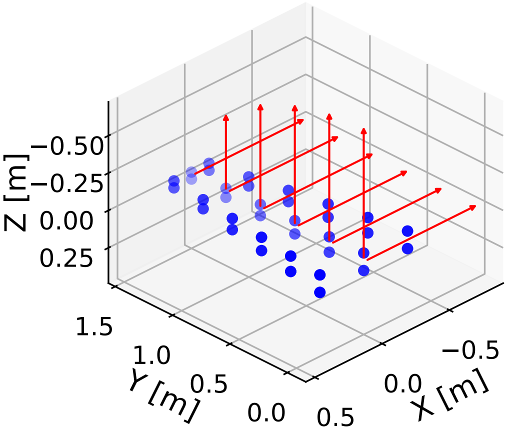


Figure 5.8: Wing structure idealised including applied forces

In the above figure, the positive x-axis is towards the nose, the positive z-axis is downwards, and the coordinate system used is right-handed. In Figure 5.7, the four corners are the spar caps, and the middle booms are two stringers. Note that the size of the boom on the plot does not represent its area; in this plot, only the positions of the dots are relevant.

Applied Loads: In order to perform a correct analysis of the wing, some loads have to be identified. The most obvious one is the lift that the wing generates, which results in a distributed force over the length of the wing. The lift is assumed to be elliptical, and thus stronger at the root of the wing. The next load is the weight of the wing, which will result in a distributed force downwards. The final load on this structure is the drag; this drag has been modelled as uniform along the span. In Figure 5.8, a representation can be seen of the applied forces on the wing structure. Note that the bending relief of the weight is already accounted for, and that the forces are not to scale.

NVM Diagrams: In order to do a per-section analysis, the internal loads have to be computed. This is done by plotting the normal, shear, and moment (NVM) diagrams along the span of the wing. In Figure 5.9, the NVM diagram can be seen. This diagram shows the shear forces and moments over half of the wing, as symmetry was assumed over the root chord.

In Figure 5.9, the polynomial order of the moments is larger by one than that of the shear, increasing the confidence of these plots. The normal force is zero over the span because no normal force is applied on the structure.

Fuselage Analysis

The analysis of the fuselage is more complex, as more loads are applied, and the ANSYS simulation has to be applied three times. The fuselage is also composed of three groups: the nose group, the middle group, and the tail group. The tail and nose groups both have a taper ratio, whereas the middle group's cross-section is constant over their length. The reasoning behind this is explained in Subsection 13.2.2.

Applied Loads

As mentioned before, the fuselage is more complex due to the number of loads that are applied to it. First of all, the reaction forces of the wing get applied to the fuselage, which includes: the normal force due to drag, the moment due to drag, the upwards shear force due to the lift, and again the moments due to this same lift. The ANSYS simulation is used for the upward lift moment. Furthermore, the propellers also induce a lift shear force and moment, and again, the ANSYS simulation will be used for this. The thrust of the horizontal back propeller creates a normal force along the length of the fuselage. The tail is also accounted for by inducing its reaction forces onto the structure. The final load is the weight of the delivery mechanism. Also, the structural weight of the fuselage is accounted for.

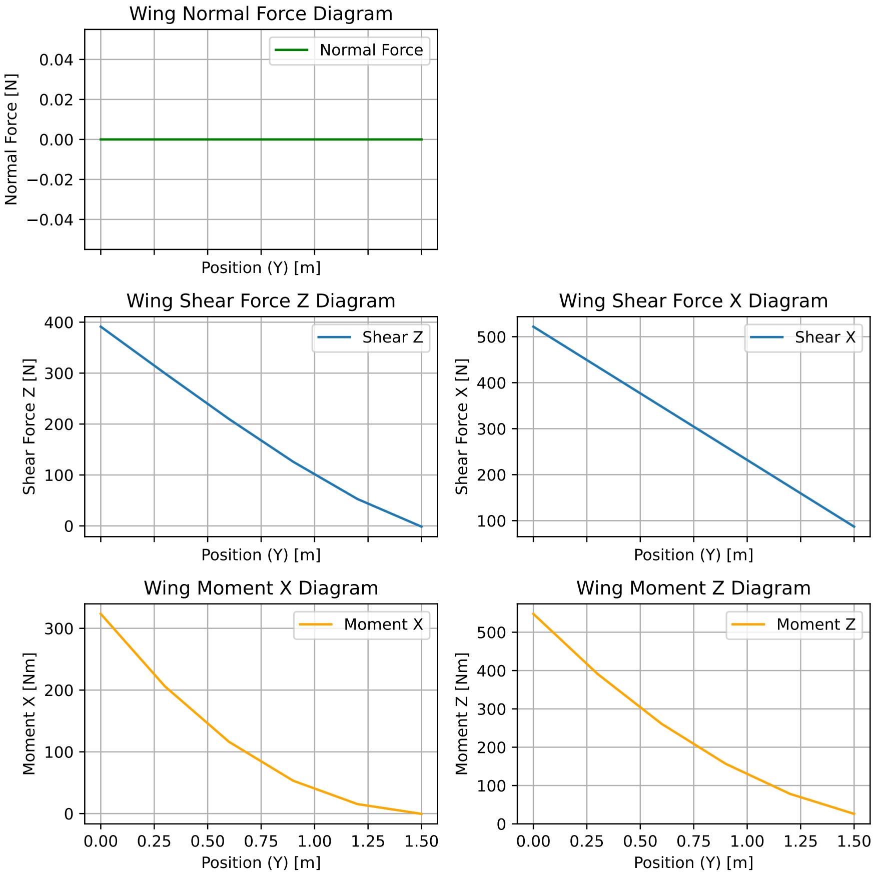


Figure 5.9: Spanwise NVM diagram of the half wing

All mentioned forces can be seen in Figure 5.11, and are represented by the red arrows. Again, their magnitude is not to scale and is only a representation to give the reader an idea of their application point.

NVM Diagram

The NVM diagrams have been set up for the fuselage as well. The boundary condition to restrict movement has been set in the middle of the fuselage. This is also why the peaks occur in the centre. The normal force accounts for the back propeller and fades away due to the drag of the wing. The jump in shear force z is due to the lift of the props and is linearly growing due to the spread-out lift over multiple sections of the wing. The moments are again one polynomial order bigger than the shear, resulting in confidence in these plots. The described plot can be seen in Figure 5.12.

Landing Gear Design

The last part of the result is the landing gear design. Given its simple nature and small size, it did not follow any of the described sizing procedures. Instead, it took a much simpler approach. Its main function is to clear the drone off the ground, so that the fuselage is not damaged by uneven terrain or foreign objects on the ground, while also ensuring the drone maintains a 3° slope, with the rear pointing downwards. This is because the vertical propellers of the drone have the same 3° angle, necessary so that they can, in case of an emergency, be activated during cruise and still propel the drone forward in an aerodynamic manner. Therefore, the landing gear inclination is necessary so that the drone can actually take off vertically, by ensuring that the angled propellers are parallel to the ground when landed.

The landing gear is composed of four support legs, each made out of lightweight rubber to ensure impact absorption while landing. They are cylindrical in shape, with a radius equal to 35 mm and a height of 70 mm for the rear pair and 116 mm for the front pair. The cross-section is hollow, with a

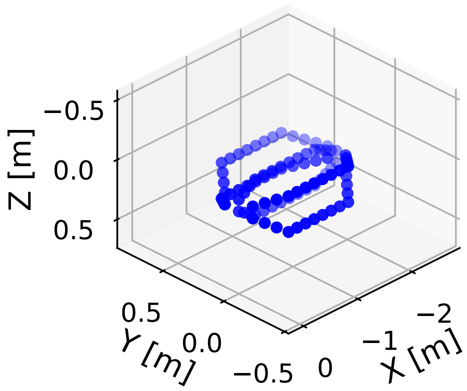


Figure 5.10: Fuselage structure idealized

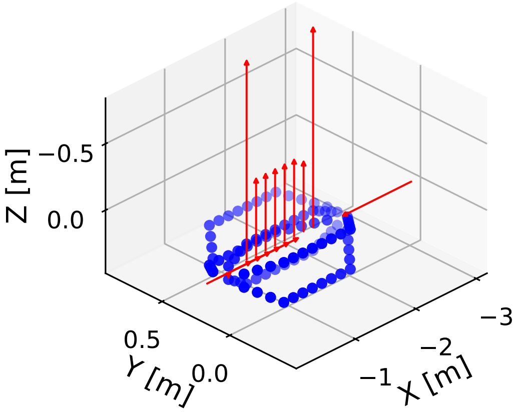


Figure 5.11: Fuselage structure idealised including applied forces

wall thickness equal to 2 mm. The total weight for all four legs is less than 0.07 kg due to the material choice. The landing gear can be seen in Figure 13.7.

5.4. Verification & Validation

In this section, the verification and validation for the boom method are handled in Subsection 5.4.1 and Subsection 5.4.2. Further on, the same is conducted for the ANSYS modelling in Subsection 5.4.3.

5.4.1. Verification for the boom method

To verify the implementation of the boom idealisation method, a simple analytical case was developed with several modifications to the original code in order to simplify the process. The goal was to create a scenario where results could be compared directly between the simulation and an analytical solution. Several assumptions and code modifications were introduced to reduce complexity:

- Wing assumed to have a constant square cross section and no taper.
- Side of the square equal to 0.5 m.
- Equivalent skin thickness equal to 0.5 mm
- No lift or weight on the wing. The only point load is applied at the tip of the wing, pointing downwards.

In order to compare the analytical calculation result with the simulation outcome, the root cross section had its maximum bending stress calculated in both scenarios. For the analytical computation, Equation 5.4 and Equation 5.6 were used:

$$I_{xx} = \frac{a^3 \cdot t}{3} \tag{5.6}$$

Where:

- $M_x = -150\,\mathrm{N\,m}$ is the internal bending moment at the root,
- $y = 0.25\,\mathrm{m}$ is the distance from the neutral axis to the edge of the cross section,
- $a = 0.5\,\mathrm{m}$ is the side of the square,
- $t = 0.5\,\mathrm{mm}$ is the effective skin thickness.

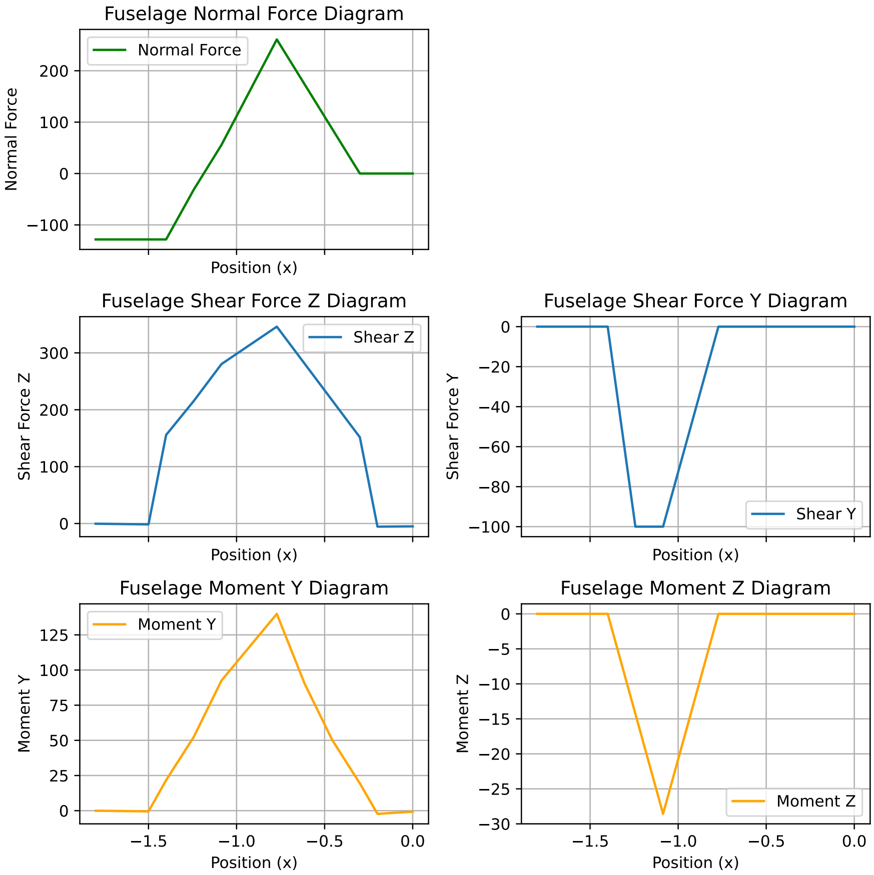


Figure 5.12: NVM diagram of the fuselage

Substituting the values yields $I_{xx} = 2.083 \times 10^{-5}\,\text{m}^4$ and $\sigma_z = -1.8\,\text{MPa}$ is the maximum bending stress at the root of the beam. Meanwhile, the result of the adapted simulation is −1,782,073 Pa for the same loading case and geometry. The deviation between the two values is around 1%, verifying that the basis of the code for the boom methods is working as intended.

5.4.2. Validation for the Boom Method

To validate the implementation of the boom idealisation method, a comparison was made between simulation results and experimental data from literature, namely the paper from Jadhao, Charde, and Dhengle [6]. A cantilever beam test case was selected from their study where steel beams were subjected to static loading, and stresses along the beam were recorded. This setup was chosen due to the high degree of similarity to the boom method simulation of the wing, enabling direct comparison of the code to real-world data.

Validation Setup

The beam in question had the following properties:

- **Geometry:** rectangular solid cross-section
- **Side length:** 25 mm
- **Side height:** 5 mm
- **Length:** 0.395 m
- **Material:** Steel
- **Applied load:** 12.32 N, downwards at the free tip
- **Measured stress:** 41.1 MPa

- **Stress location:** 15 mm from root

The cross-section was idealised in the simulation as an 8-boom configuration, with booms located at the corners of the square and also on each side. The area of each boom was approximated using the expression:

$$A_{\text{boom}} = \frac{a \cdot b}{8} = \frac{25 \cdot 5}{8} = 15.63\,\text{mm}^2 = 1.563 \times 10^{-5}\,\text{m}^2$$

Stress from Experimental Data and Boom Method Result

The experiment directly reported the stresses measured at four different locations along the length of the beam. For this validation, the closest one to the root was selected for comparison. Under the loading of 12.32 N the stress closest to the root is 41.1 MPa [6].

The same case was implemented in the boom method simulation code with the following settings:

- Span: 0.395 m
- Point load: 12.32 N at the tip
- 8 booms, each with an area of $1.562{,}5 \times 10^{-5}\,\text{m}^2$
- No distributed lift, but the weight of the sections is computed

The maximum bending stress at the root was computed as **49.1 MPa**.

Comparison and Conclusion

The result from the boom method code differs from the theoretical value by 19.5%.:

$$\text{Relative error} = \frac{|\sigma_{\text{boom}} - \sigma_{\text{theory}}|}{\sigma_{\text{theory}}} = \frac{8}{41} \approx 19.5\%$$

While both values are in the same order of magnitude, the discrepancy is significant and needs to be further examined. There are several factors that can explain the larger value from the simulation:

- **Idealisation assumptions:** The boom method assumes concentrated stiffeners and zero-thickness skin, which introduces simplifications in stress distribution.
- **Cross-section irregularity:** The experimental beam is not hollow, while the simulation assumes the booms spread around the edges only. This assumption in the code decreases the strength of the beam and partially explains the higher stress from the simulation.
- **Numerical resolution:** Discretisation in the simulation (number of sections) could affect result precision. Similarly, the number of booms in the cross-section also affects how the distribution takes place.
- **Stress not measured exactly at root:** In the experiment, the stress is not measured exactly at the root, but instead 15 mm in front of it. Compared to the code value, the experimental value will naturally be lower, since there are fewer accumulated stresses closer to the tip.
- **Experimental measurement accuracy:** Equipment dis-calibration or human error could have affected the experimental data slightly. Material properties discrepancies, due to manufacturing defects or fatigue, could also explain the difference.

Overall, several of the assumptions in the code explain the disparity between its result and the experiment. And despite the significant difference, the boom method ends up having a higher stress, which would over-design the wing. This is highly preferable over the opposite case of under-designing the structure, potentially resulting in failure.

It is also important to note that in the original paper, the authors also concluded that among experimental (41.1 MPa), analytical (44.25 MPa), and ANSYS (44.94 MPa) measurements, the experiment had the lowest stresses [6]. This further corroborates the simulated boom method, displaying higher stresses than the experiment.

For further validation, the code could be further adapted to better reflect the loading scenario from the experiment, such as more accurate positioning of where the stress is measured and the number of booms. These could help further refine the design and validate the method.

5.4.3. ANSYS Verification & Validation

The verification of the ANSYS simulation was performed during the gathering of the needed data. The gathered data clearly shows an increase in stress with a decrease in thickness, which is clearly the expected result; this plot can be found in Figure 5.6. Furthermore, a sensitivity analysis has been performed on the mesh size. Figure 5.13 clearly shows a convergence of stresses with decreasing mesh element size; a trade-off of mesh compiling time and accuracy was performed. For all simulations, an element size of 0.5 mm has been chosen, which allows to have at least 4 elements over the thickness of the section.

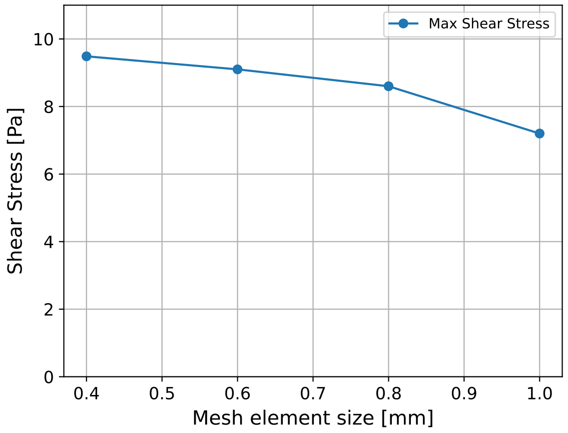


Figure 5.13: ANSYS convergence for mesh size

While experimental validation is outside the scope of this project, the structure can be built and tested under the prescribed loads and see if any failure mode occurs.

5.5. Recommendations

Due to the time constraints of the DSE, the detailed design of the structure was not performed, leaving room for future work building upon current results. For example, the wing box currently consists of only one cell, which has reduced the computation effort significantly, but might not be the most optimal design. Therefore, it is encouraged to look into multi-cell wing boxes for future development. Moreover, the landing gear design is performed at a very preliminary level. As such, a detailed stress and strain analysis of the gear should be performed to investigate failure loads and modes.

Another point for further investigation is fatigue loading: the drone will encounter many take-off and landing sequences, which will affect the structure through fatigue; therefore, a fatigue analysis is preferred before drone manufacturing. Manufacturing the drone will also enable more robust validation of the sizing procedure, since actual tests will be performed on the product before finalising its design.

Finally, the exact geometry of the spars and stringers has not been defined yet; this should also be looked into for further development. ANSYS was used in this report, but only for the section intersecting the wing/propellers; this analysis should be extended to all sections and to the true geometry of the final design.

6
Payload Handling & Avionics

Central to both the design and operations of the DronebezORgd fleet is the safe, reliable, and efficient handling of the pizza payload. This is the focus of this chapter. Not only does the payload have to be sufficiently shielded from the freestream air in terms of particulates or temperature and be easily accessible for the customer, but the drone also has to fit within a footprint that enables compatibility with as many pick-up and delivery locations as possible. These aspects were all reflected in the subsystem requirements in Table 4.8. To comply with these requirements, four crucial elements were identified. These are to be discussed in this chapter, after a presentation of the design approach and assumptions in Section 6.1. First, Section 6.2 covers the design of the payload delivery mechanism. Additionally, a thermal insulation structure is presented in Section 6.3. A wingspan reduction mechanism is then chosen and sized in Section 6.4. Lastly, a set of avionics equipment is selected in Section 6.5 before recommendations for future projects and studies are discussed in Section 6.6.

6.1. Design Approach & Assumptions

The design of the payload handling subsystem was split into four parts: payload delivery mechanism, thermal insulation structure, wingspan reduction mechanism, and avionics selection. Common across these is the use of some off-the-shelf components in the design, resulting in a discrete design space in terms of mass and cost estimates for the subsystem. These components are chosen through performing initial sizing procedures of the mechanisms or parts, and then browsing online catalogues for suitable components. Because of this, each subsection relating to the design of one of these parts is terminated with a summary of all selected components.

The methods in this chapter build on the following list of assumptions.

- **AS-PL-01:** The payload is stored in uniformly sized boxes
  Although pizzas may be served in a variety of sizes, it is believed sufficient for the level of detail of this project to design for the largest allowable size. This is done in analogy to shipping containers or aircraft unit load devices.
- **AS-PL-02:** The initial payload temperature is sufficiently above 63 °C
  Food delivery data shows that initial pizza temperatures are usually in the range between 79 *and* 93 °C *[7]. Pizzas at significantly lower initial temperatures than this would require active heating to meet RQ-PL-13, constraining the design space.*
- **AS-PL-03:** The wing weight is linearly distributed in spanwise direction
  The wing structure is modelled as a three-dimensional trapezoid. As the cross-sectional area reduces linearly going from the root to the tip, so does the weight.
- **AS-PL-04:** The wing lift is elliptically distributed in spanwise direction
  The lift acting on the wing structure is modelled as an idealized elliptical distribution, as the real distribution is difficult to quantify at this stage in the design.
- **AS-PL-05:** The wing produces no lift while hovering
  As the airspeed of the drone is zero during hovering, there is no flow of air particles over the wing in this phase, and thereby no lift.

6.2. Payload Delivery Mechanism

This section covers the design of the payload delivery mechanism, an assembly aimed at getting the payload into the customer's hands in a convenient, efficient, and safe manner. This is the main content of Subsection 6.2.1. Thereafter, Subsection 6.2.2 presents an overview of verification and validation activities related to the design.

6.2.1. Analysis

This design subsection is structured as follows. First, a material choice is made based on a list of criteria. Then, design options for the payload delivery mechanism are generated and presented, before being traded off in a graphical trade-off matrix. Lastly, the final concept is designed in further detail.

Material Choice

Core to the challenge of designing a functional payload delivery mechanism is the choice of a suitable material. This material should satisfy several needs. Being possible to 3D-print using commercially available Fused Deposition Modelling machines is one of them, as this manufacturing technique allows the creation of complicated parts with no up-front mould costs and little material loss. Recyclability is also seen as an advantage, but this was found to often follow from the 3D-printability. As the environment of the payload bay will be rather extreme, with high temperatures and relative humidities, it is of utmost importance that the material used is resistant to this [7]. Lastly, even if the material is not expected to come into contact with the pizzas themselves due to their cardboard packaging, there is still a possibility for migration of material particles to the food, meaning the material falls under EU Commission Regulation 10/2011 [8].

In summary, a material was sought that is:

1. 3D-printable
2. recyclable
3. resistant to ambient temperatures up to 93 °C [7]
4. rated safe in contact with food according to EU Commission Regulation 10/2011 [8].

These criteria narrow the space for material choice significantly. In an investigation of more than ten commercially available materials that fit all four criteria, only one was found – *GreenTEC Pro* by Extrudr. Sold in spools ready for 3D printing, it is a recyclable and biodegradable material made fully from renewable raw resources. It offers good temperature resistance with its heat deformation temperature of 115 °C, sufficiently above the expected temperatures in the payload bay [9], and shows documented compliance with EU Commission Regulation 10/2011 [10]. No material trade-off is performed for this choice, as it was simply found that the criteria narrowed the design space so significantly that no competing material could be found in the allocated time for the design of this subsystem. For instance, *colorFabb_HT* is a material which is tailor-made for high-temperature uses, but fails to have EU 10/2011 compliance, meaning it cannot be used for the DronebezORgd operation [11]. Alternatively, colorFabb's *PA Blue Metal Detectable*[1] and certain colours of Filamentive's *PETg*[2] materials do comply with this regulation, but have inferior moisture or temperature resistance, respectively.

Design Concepts

In order to ensure that the choice of overall architecture for the payload delivery mechanism is not arbitrary and adequately reflects the space of technologically feasible designs, design concepts were systematically brainstormed as a team. From this, a set of numerous possible concepts was chosen and then sized using approximate methods so that they could be assessed and compared. These concepts are described below. For each concept, it is assumed that the loading of payload must be possible in arbitrary order, though the order of unloading may be strict.

[1] URL `https://colorfabb.com/pa-blue-metal-detectable` [cited 15 May 2025]

[2] URL `https://www.filamentive.com/product-category/rpetg-recycled-petg-pet-3d-printer-filament/` [cited 15 May 2025]

1. **Traditional stack, sideways delivery:** Similar to existing last-mile food delivery services, the payload is simply stored in a compartment accessible from one side. There is no mechanism, and the customer has to reach inside to receive their order. Although the delivery order is flexible, as the customer can pick any pizza in the stack, it also introduces additional risks if customers cannot be trusted not to tamper with others' orders.
2. **Actuated push, sideways delivery:** Pizzas are stored on individual racks that are closed behind separate doors, which only open at the respective customer's delivery location. Linear actuators may be used to push the pizzas out so that they are more reachable. Similar to the above, this concept relies on the customer to extract the pizza from the payload bay.
3. **Actuated drop, downwards delivery:** Individual pizzas are held on two sides with mechanical actuators. A controlled number of pizzas can be delivered by triggering a given sequence of actuators, dropping the pizzas a distance of around 10 to 25 cm. The order of deliveries is, however, predetermined, as a pizza higher in the stack has to wait for all lower pizzas to be delivered first. A separate sideways hatch may be needed for loading the payload bay.
4. **Cyclical stacks, downwards delivery:** The payload is stored in two adjacent stacks, and a combination of vertical conveyor belts and actuators is used to cycle the order of pizzas. At the bottom of one of the stacks, pizzas may be dropped to the ground. This concept allows for automated delivery while retaining full flexibility of the delivery order.

Trade-Off

To arrive at the most suitable design concept for the payload delivery mechanism, a trade-off was performed. This was done following the same method as described in Subsection 4.1.1 – that is, a graphical trade-off method. Five criteria were selected on the basis of minimised overlap between the criteria and maximised coverage of the diversity spanned by the different design concepts. The five criteria are described below, before being used in Table 6.1 to evaluate the concepts.

1. **Mass – 30%:** This criterion covers the expected mechanism mass based on simple force equilibrium calculations and online databases of electronic components. This criterion was awarded the highest weight, as large masses for the payload handling subsystem could have snowball effects in other subsystems.
2. **Risk – 25%:** Covering aspects such as design complexity and theft of other customers' payload, this criterion assesses what technological and operational risks the mechanisms may come with. As it also includes considerations on fail-safe design, it was judged to be of high importance.
3. **Customer interface – 20%** It is important for the adoption of the DronebezORgd service that the design of the delivery mechanism is user-friendly. This criterion therefore, evaluates the effort required by the customer in the delivery process, for instance, by valuing a familiar interaction over a more foreign one.
4. **Restaurant interface – 15%:** As the restaurant is also a customer of the service, a good interface with restaurants is appreciated. When comparing the different design concepts, it is mostly just the ease of loading the pizzas into the delivery mechanism that stand out as differences. This criterion is therefore given a lower weight than that of the customer counterpart.
5. **Operational efficiency – 10%:** The efficiency gains or losses in the operation of the drone fleet are taken into account through this last criterion. Elements covered include the degree of automation in the design and how it may cause or hinder in taking off or payload delivery order.

As can be seen from Table 6.1, the four design concepts performed variably in the different criteria. Whereas the cyclical stacks showed unacceptable performance in terms of risk, it excelled at operational efficiency, offering a flexible delivery order and no opportunities for payload handling-related hindering of take-offs. On the other hand, the traditional stack showed good performance in most categories, especially mass, only being outperformed by the actuated push in the customer interface. Overall, the sideways-oriented mechanisms showed better results than downward-oriented mechanisms in most criteria.

Table 6.1: Graphical trade-off of the payload delivery mechanism

Criterion Mechanism Concept	Mass	Risk	Customer interface	Restaurant interface	Oper. efficiency
Traditional stack	no delivery mechanism needed	fail-safe, but remaining pizzas may be stolen or tampered with	tried-and-tested interaction	tried-and-tested interaction	pizza is kept for no-shows
Actuated push	actuators are not load-bearing – can be lighter	fail-safe design	pizza is pushed out for reachability	similar interaction as tried-and-tested	hindered take-off if pizza left out
Actuated drop	actuators are load-bearing	pizzas may fall out on mechanism failure	ground drop may feel robotic, wet ground may damage pizza	awkward loading position without extra hatch	delivery unaffected by external factors
Cyclical stacks	stepper motors and actuators, both load-bearing	tight tolerances needed, and pizza may fall out on mechanism failure	ground drop may feel robotic, wet ground may damage pizza	difficult to load due to moving parts	full control of pizza delivery and order

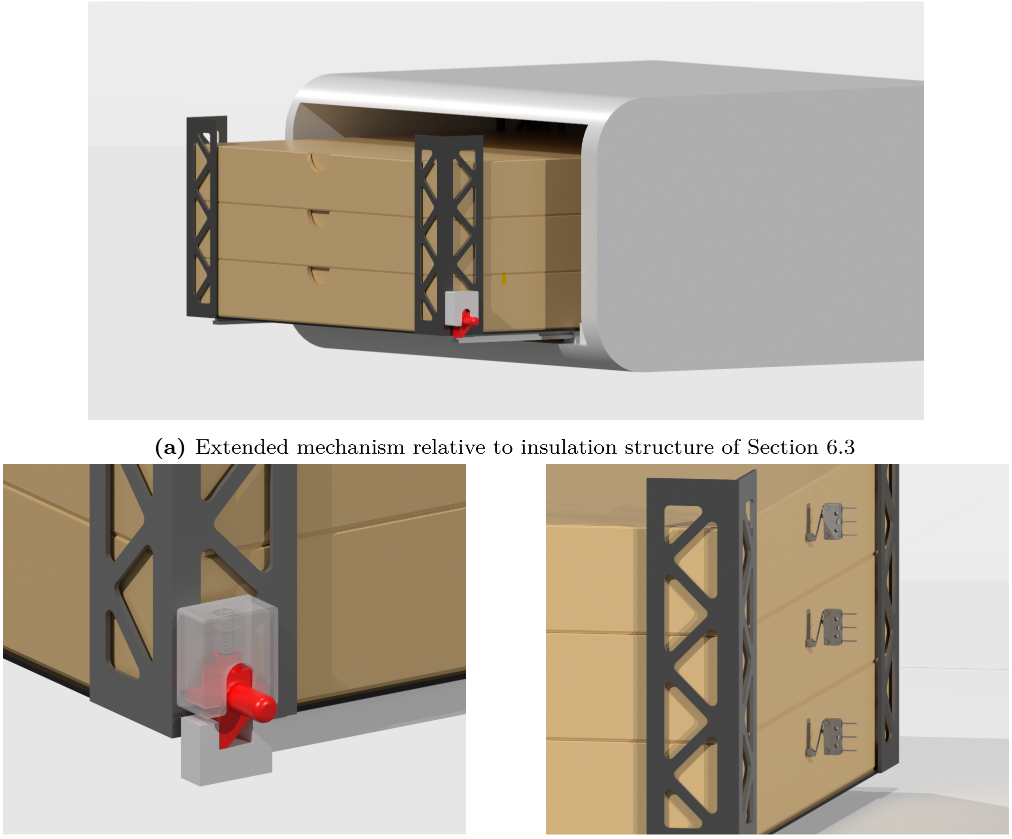
Although the traditional stack and actuated push concepts scored better overall, neither was chosen, and the trade-off was rather used as an opportunity to re-evaluate the design space, as it showed that different options had certain distinct, well-performing aspects that could potentially be combined. This resulted in the final concept, blending the tried-and-tested interaction from the traditional stack with the reachability benefits of the actuated push. The mechanism was also chosen to be manual, removing the need for heavy electrical components. Another benefit of this is the reduced likelihood of hindered take-offs, as the pizza cannot be ejected without the customer present. Also clear from the trade-off was the advantage of providing the restaurant with the same excellent user interface as the customer through a sideways-oriented design.

Detail Design

With a payload delivery mechanism concept chosen, a two-step approach was taken in sizing and designing it. First, a strawman sketch of the design was made, from which some simple volume, mass, and cost calculations could be made, given the preceding material choice. In the second stage, the mechanism was modelled in *3DExperience CATIA*, where more accurate volume estimates could be extracted, providing a verification of the first step and results of increased fidelity and trust.

In both steps, the design was treated as an assembly of multiple parts, each with simplicity and 3D-printability in mind. These are listed below, together with a short description of their main functions and constraints, complementing what is shown in Figure 6.1a and 6.1b.

1. **Payload holder:** basket that protects the pizzas from excessive movements and uses cutouts to save mass
   (a) **Bottom plate:** supports the weight of the pizzas and provides some tolerance for box size
   (b) **Corner pieces:** prevents lateral and longitudinal pizza movement
2. **Sliding mechanism:** allows for partially sliding the payload holder out of the drone
   (a) **Rail with end stop:** continuous T-shaped piece able to stop exaggerated movement



(a) Extended mechanism relative to insulation structure of Section 6.3

(b) Close-up of locking mechanism, made translucent

(c) Close-up of limit switches for capacity sensing

Figure 6.1: Renders of the payload delivery mechanism

   (b) **Complementary rail holder:** C-shaped piece that isolates the sliding mechanism to one degree of freedom

3. **Locking mechanism:** locks the sliding mechanism and provides the customer with a handle to pull with, taking inspiration from typical door latches
   (a) **Slot:** provides a ledge behind which the latch can lock
   (b) **Latch and handle:** spring-loaded latch connected with ergonomic handle
   (c) **Latch enclosure:** protects other parts from possible damage and simplifies user experience

In addition to this, a sensor layout was chosen to aid the computation module to evaluate the state of the payload delivery. Due to the demanding environment of the payload bay, most optical sensors, such as retroreflective or proximity sensors, were ruled out for this purpose [12]. Limit switches, a type of electromechanical component activated with the push of a physical force, were chosen due to their high operating temperature, simplicity, and possibility for an IP67 rating. Specifically, the *83186063* by Crouzet at a mass of 2 g was chosen[3]. Three switches were positioned at the back of the payload bay for the detection of the number of pizza boxes present, as seen in Figure 6.1c, and two for the detection of successful locking of the sliding mechanism; the latter in accordance with RQ-PL-09 as correct closure of the bay was seen as critical for a safe flight.

Finally, the mass and cost of the payload delivery mechanism were found to be around 1.17 kg and €126 for both payload bays in total. This has been summarised per component in Table 6.2.

[3] URL `https://www.digikey.nl/en/products/detail/crouzet/83186063/2415907` [cited 12 May 2025]

6.2.2. Verification & Validation

Each part of the design approach for the payload delivery mechanism was verified in a tailored manner. When choosing off-the-shelf components, multiple catalogues were checked to ensure the associated masses and costs were reasonable for the required performance. When performing the initial sizing and estimation, sensitivity studies were performed to ensure all variables were referenced and used correctly. Lastly, the CAD model was also used to both refine these estimates and verify that the calculations and modelling had been done correctly. Verifying *3DExperience CATIA* and its volume and mass computations was not performed, as it was judged to be outside the scope of this project and expected not to be erroneous due to how established the software is in the industry.

Validating this design still remains as a recommendation. The effect of thermal and mechanical loads on the mechanism components, as well as the effectivity of the chosen sensors, should be investigated through manufacturing prototypes of the system and subjecting them to representative conditions.

Table 6.2: Summary of all selected payload delivery mechanism components

Component Type	Model Name	Count [-]	Cost [€]	Mass [g]	Dimensions [mm]
Bottom plate	-	2	10.20	225.7	454 × 454 × 2
Corner piece	-	8	1.68	37.3	52 × 52 × 210
Rail with end stop	-	4	1.69	37.4	405 × 12 × 9
Rail holder	-	4	2.24	49.7	200 × 22 × 13
Slot	-	2	0.38	8.4	30 × 22 × 12
Latch and handle	-	2	0.38	8.3	26 × 32 × 12
Latch enclosure	-	2	0.45	9.9	31 × 36 × 17
Limit switch	Crouzet 83186063[3]	10	7.33	2.0	20 × 6.4 × 17.5
Spring	RS-PRO 821-368[4]	2	0.46	0.4	20 × 4.6 × 4.6
Total			126.2	1172.2	

6.3. Thermal Insulation Structure

The thermal insulation structure is responsible for keeping the payload warm and clean during the mission. The design of this assembly is discussed in Subsection 6.3.1. Following this, verification and validation efforts are presented in Subsection 6.3.2.

6.3.1. Analysis

The design of the thermal insulation structure is derived from literature on existing methods for payload insulation in last-mile food delivery. According to a study by Hasan and Dincer [7], which investigated the thermal insulation performance of three different insulating materials for a food delivery bag made for three 40 cm pizzas, polyurethane (PU) foam appears to be an effective option. It offers a relatively low thermal conductivity compared to the industry standard bag, and additionally has good structural rigidity while still being a flexible material. In the study, PU sheets of 25 mm thickness were used, for which, experiments showed that the final temperature of each pizza stayed above 70 °C throughout an average 30-minute delivery. Additionally, reusable moisture absorption fabrics such as bamboo and hemp were tested, showing positive results. Next to this, also the effect of phase change materials (PCM) in the heat retention of the delivery bag was assessed, but this showed ineffective results unless the PCMs were heated beforehand, allowing them to release heat during the delivery.

The thermal insulation structure of the BezORger builds on the conclusions of this study. A low-density PU foam[5] similar to that of the study is used due to its promising performance. This insulation

[4] URL `https://nl.rs-online.com/web/p/compression-springs/0821368` [cited 12 June 2025]

[5] URL `https://www.hornbach.nl/p/schuimplaat-softpur-50x80x3-cm/3838954/` [cited 17 June 2025]

material is structured as a cuboid with a wall thickness of 25 mm and a spacing of 10 mm to the payload delivery mechanism inside it. The insulation structure is preferably multi-layer in order to reduce heat loss due to radiation or keep the payload bay cleanable, but the choice of these materials was seen as too demanding for the scope of the current design stage. The usage of PCMs was also seen as promising, but as no thermal simulations could be done for the payload bay, the effectiveness of these in this application is still unknown.

In order to comply with RQ-PL-14, a sensor was chosen that can record the internal temperature in the payload bay throughout a mission. As humidity data was also shown to be important by Hasan and Dincer [7], a combined humidity and temperature sensor is chosen. Two units were chosen per payload bay in accordance with RQ-PL-09. This brought the mass and cost of the thermal insulation structure to around 1.26 kg and €70.22, as shown in Table 6.3.

Table 6.3: Summary of all selected thermal insulation structure components

Component Type	**Model Name**	**Count [-]**	**Cost [€]**	**Mass [g]**	**Dimensions [mm]**
Insulating PU foam structure	-	2	19.43	610.7	543 × 513 × 290
Humidity & temperature sensor	Adafruit Industries 5064[6]	4	7.84	<10	59 × 26.5 × 13.2
Total			70.22	<1261.4	

6.3.2. Verification & Validation

For the verification of this design, it was used that major parts of the design come from established literature and experiments. Validation of the design is outside the scope of this project, but it is highly recommended for a future design stage to use simulation software to better quantify the actual performance of the design. Additionally, a prototype of the thermal insulation structure should be constructed, and proper experiments should be performed to validate it against real-world data.

6.4. Wingspan Reduction Mechanism

This section presents the design of the wingspan reduction mechanism, whose main purpose is to enable the drone to reduce its footprint to be able to land at a customer's location. Subsection 6.4.1 covers the preliminary design of this assembly. Afterwards, in Subsection 6.4.2, verification and validation efforts and plans are presented.

6.4.1. Analysis

The structure of this subsection is as follows. Firstly, a literature study is performed in to establish the current state of wingspan reduction mechanisms, both in larger aircraft as well as on the kilogram-scale. Building on this, the folding and locking mechanisms for the BezORger are designed based on calculations of the expected loads during cruise and transition.

Literature Study

As there are many conceivable methods for decreasing the wingspan of the BezORger's wing, literature was used to identify the most compelling concepts. As identified by Yarygina and Popov [13], existing techniques for reducing the footprint of an aircraft include controllably turning, sweeping or folding along an axis perpendicular to the wing plane, or folding the wings along an axis in the wing plane. Out of these, folding along the airplane symmetry axis was identified as the most common, at least for their dataset of carrier-based aircraft.

[6] URL `https://www.digikey.nl/en/products/detail/adafruit-industries-llc/5064/14625562` [cited 17 June 2025]

Also on a smaller scale, *folding* the wing appeared to be the most common technique. Radio-controlled aircraft models [14] and drones[7] were found, which also specify in detail which mechanisms are used to achieve the folding of the wing. Mills and Ajaj [14] identified three concepts for folding the outer 0.26 m of a modified 2.12 m-span UAV with a conventional aircraft configuration with a take-off mass of 2.7 kg. All three use a rotary actuator (direct current servomotor) in different locations of the wing, connecting them to the hinge line either directly, using a belt drive system, or using a worm drive system, though the second option was ultimately chosen in their trade-off due to its effectiveness and lower complexity. The folding drone[7], on the other hand, is smaller at 1.5 m of unfolded wingspan and a 1.4 kg take-off mass and uses linear actuators within the bottom surface of the wing.

It is based on this literature that a similar belt drive system is chosen for the folding of the BezORger's wing. However, as this drone is of a significantly larger scale than those mentioned above, combined with the ultimate load factor found in Section 11.4, a choice was made to separate the functions of folding the wing during hovering flight and keeping the wing's position locked in all flight phases. This was done to achieve higher degrees of safety both under the in-flight loads and on-ground customer interaction.

Mechanism Sizing

For the design of the belt-driven actuator system, which is only activated while the drone is hovering, it is assumed that the weight of the wing is the dominant opposing force (AS-PL-05). The wing weight is modelled linearly according to AS-PL-03 as follows:

$$w(y) = Ay + B, \text{ with } A, B \text{ such that } w\left(\frac{b}{2}\right) = \frac{c_t}{c_r} w\left(0\right) \text{ and } \int_0^{b/2} w(y)dy = \frac{W_{\text{wing}}}{2}.$$

Here, $w(y)$ is the wing weight distribution at a spanwise position y from the wing root, b the wingspan of the drone, c_t and c_r the tip and root chords, respectively, and W_{wing} the weight of the wing group. From this, the torque required for folding the wing when the wing is in the unfolded position could be found through Equation 6.1, with $y\prime$ the spanwise location of the hinge line.

$$M_W(y\prime) = \int_{y\prime}^{b/2} \left(y - y\prime\right) w(y) dy, \tag{6.1}$$

Given a span of 3 m, tip chord of 0.217 m, root chord of 0.543 m, and a wing group weight of 2.18 kg, the wing weight moment at $y\prime = \frac{b}{4}$ becomes 0.415 N m, or 0.219 N m after applying a design factor of 1.5. The torque required for the servomotor itself may, however, be reduced by using a gear ratio, as long as that still allows for at least 90° of movement. This was, however, deemed unnecessary in this design as the required torque is relatively low and a gear ratio may result in an increased volume for the mechanism, which has to obscure the wing surface minimally. Ultimately, the *S3003* by TianKongRC[8] was chosen for its low mass of 43 g and maximum torque of 0.64 N m, as few components could be found with a smaller, yet still large enough, torque.

In both final positions of the wingspan reduction mechanism, a locking mechanism must be employed to guarantee higher levels of safety, in the event that the actuator used for folding the wing malfunctions (see RQ-PL-08). Here, a simple spring-loaded latching shear pin inside a double shear lug was chosen, though this could be revised at a later stage in the design if testing shows the need for it. During cruise, the largest loads acting on the wing are the lift and wing weight. As the assumption of an elliptical lift distribution is not necessarily conservative, the contribution of the wing weight to the moment at the hinge line was neglected. The moment could then be found using

$$M_L(y\prime) = \int_{y\prime}^{b/2} (y - y\prime) \frac{4L}{\pi b} \sqrt{1 - \left(\frac{2y}{b}\right)^2} dy, \tag{6.2}$$

[7] URL `https://youtu.be/gZQEOjyjwhc?si=Vw2Ok4imO_4B8j86` [cited 04 June 2025]

[8] URL `https://www.tinytronics.nl/en/mechanics-and-actuators/motors/servomotors/s3003-servo` [cited 16 June 2025]

where L is the total lift generated by the wings[9]. Inserting a total lift of 600 N at the ultimate load factor (see Section 11.4) and otherwise the same values as above, results in a wing lift moment at $y\prime = \frac{b}{4}$ of 54 N m after applying a design factor of 1.5. At this spanwise location, the chord length is around 0.38 m, and the thickness around 0.045 m, meaning a single shear pin would have to withstand a load of around 1,200 N.

It is based on this load that the pin and lug were designed. These two components have to satisfy the equations

$$\tau_{\max} = \frac{F_{\text{ult}}}{\frac{1}{4}\pi d_{\min}^2} \quad \text{and} \quad \sigma_y = \frac{F_{\text{ult}}}{d t_{\min}}, \tag{6.3}$$

for shear and bearing failure, respectively. Here, $\tau_{\max}$ and σ_y are the maximum shear and yield stresses of the material, $d_{\min}$ is the minimum required pin diameter, and $t_{\min}$ is the minimum required lug thickness. Using the same aluminium 6061-T4 material as in Section 5.2, this results in values of $d_{\min}$ =3.1 mm and $t_{\min}$ =2.8 mm. Ultimately, a diameter of 4 mm was chosen to be conservative.

In the wingspan reduction mechanism, the shear pin is accompanied by a mechanism consisting of a spring to make it naturally assume a locked position for a more fail-safe design, and a servomotor to push against this spring during rotation of the wing folding hinge. The layout and mounting hardware of the wingspan reduction mechanism were not designed in further detail, and so a contingency is taken on the mass and cost of mounting hardware of 50% of the otherwise total values, as Mills and Ajaj [14] indicates the need for multiple components. All components are summarised in Table 6.4, and a visualisation of the folded wing is shown in Figure 6.2. By folding at the middle point of each wing, the wingspan is reduced from 3 m to around 1.6 m, when accounting for the thickness of the folded wing.

Table 6.4: Summary of all selected winspan reduction mechanism components

Component Type	Model Name	Count [-]	Cost [€]	Mass [g]	Dimensions [mm]
Wing movement servo	TianKongRC S3003[8]	2	6.00	43	40 × 20 × 41
Belt drive gear	OPTIBELT Timing Belt Pulley T2.5 12[10]	4	4.02	3.0	9.55 × 9.55 × 16
Belt drive belt	OPTIBELT 4 T2,5 / 145[11]	2	9.40	6.7	4 × 145
Shear pin	-	2	0.10	3.4	25 × 4 × 4
Locking spring	RS-PRO 821-368[12]	2	0.46	0.4	20 × 4.6 × 4.6
Unlocking servo	SunFounder CN0193	2	13.57	56	40.1 × 18.7 × 48.5
Mounting hardware contingency	-	2	1.26	42.8	-
Total			77.66	377.8	

6.4.2. Verification & Validation

The derivations and computations used to arrive at the preliminary sizing of the wingspan reduction mechanism have all been checked using a computer algebra solver and sensitivity analyses were performed by varying certain input variables and checking for outputs of expected proportionality.

[9] URL `https://web.mit.edu/16.unified/www/SPRING/fluids/Spring2008/LectureNotes/f07.pdf` [cited 04 June 2025]

[10] URL `https://nl.rs-online.com/web/p/belt-pulleys/2362767` [cited 17 June 2025]

[11] URL `https://nl.rs-online.com/web/p/timing-belts/0209700` [cited 17 June 2025]

[12] URL `https://nl.rs-online.com/web/p/compression-springs/0821368` [cited 12 June 2025]

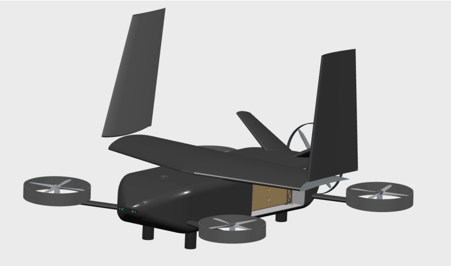


Figure 6.2: Render of wing with folded wingspan reduction mechanism

As the wingspan reduction mechanism brings added complexity and risk to the drone design, validating it is of utmost priority in the coming design phases. Prototypes of the design should be made such that any tolerances can be refined, and thorough test plans should be developed that push the load-bearing capabilities of the mechanism to its limits.

6.5. Avionics Selection

Another important facet of the drone's design is the selection of the avionics. This is necessary not only to ensure compliance with the requirements on obstacle avoidance, positional estimation, and communication, but also to enable more accurate mass estimation and internal layout design. Subsection 6.5.1 will therefore describe the analysis related to the selection of the BezORgers avionics, and Subsection 6.5.2 will explain the verification and validation to accompany it.

6.5.1. Analysis

Due to the limited development time and scale of DronebeORgd, all avionics will be selected off the shelf. While this implies that some components may not be perfectly tailored to the BezORgers' mission, it provides extra reliability due to the use of well-tested commercial products.

Flight Controller

The first step in avionics selection is choosing an appropriate flight controller. This will establish the base of the system, providing the basic necessary components for state estimation, communication, and the integration of components.

This flight controller is selected through a trade-off of three identified potential controllers: the DJI NAZA-M V2, the Pixhawk 2.1, and the Navio2. All three of these controllers provide localisation, state estimation, and communication capabilities. The five criteria are described below, after which, Table 6.5 summarises the trade-off, using the colouring scheme from Table 4.1 to rank each option.

1. **Cost – 15%:** The cost is an important criterion due to drone costs being a significant contribution to business start-up costs, meaning their reduction is important.
2. **Size – 10%:** This criterion covers the flight controller mass. This criterion was awarded the smallest weight, as while size is important, flight controllers are inherently small on the scale of the drone, and so their effects on sizing are limited.
3. **Computational Performance – 20%:** This criterion is especially important to the BezORger due to its autonomous nature. This means a larger-than-normal amount of computation is performed by the drone, leading to a need for high computational performance.
4. **Positional Accuracy – 25%:** Due to the precise nature of navigating in an urban environment, positional accuracy is a very important factor of the flight controller, and is therefore given the second largest weight.
5. **Reliability – 30%:** The largest weight is given to reliability due to DronebezORgd's continued focus on safety. This criterion includes the robustness and reliability of the system, ensuring failures never occur.

Table 6.5: Graphical trade-off of the flight controller

Criterion / Flight Controller	Cost	Size	Computational performance	Positional Accuracy	Reliability
DJI NAZA-M V2	low pricing	large	high computational power	Unacceptable 2.5 m horizontal accuracy	standard GPS only
Pixhawk 2.1	Nominal	compact	nominal	nominal	supports redundancy through expansion
Navio2	low pricing	nominal	high computational power	great 10 cm vertical accuracy	supports redundancy through expansion

As seen in Table 6.5, the DJI NAZA-M V2 is rejected as an option due to its unacceptable 2.5 m horizontal positional accuracy, as this would render landing in the area designated by RQ-PL-05 impossible. Looking at the remaining two options, the Pixhawk and Navio2 are both found to be viable options; however, the Navio2 is selected as the superior flight controller due to its great positional accuracy, high computational power, and low pricing.

To accompany the Navio2 flight controller and strengthen the redundancy of the system to levels compliant with RQ-PL-09, an additional GNSS receiver is used. Specifically, the GNSS chosen is the Navio2 expansion GNSS sensor provided by the same manufacturer to ensure compatibility. Additionally, to comply with RQ-12-03-01, a drone remote identification module is needed. For this, the Dronetag DRI is chosen due to its low mass and easy integration with the flight controller.

Perception Sensors

In addition to the flight controller, the perception sensor selection must be conducted to provide the drone with sufficient perception capabilities. An example use case of this is obstacle detection, where the perception sensors would enable the drone to respond to unexpected obstacles, sufficiently diverge from its assigned path, and then return. Based on market research of autonomous drones with high perception capabilities, the following sensors are deemed necessary: six stereo cameras for omnidirectional depth perception, and two downwards-facing time-of-flight (ToF) sensors for precise ground tracking for proximity operations. Furthermore, it was deemed that the forward-facing stereo camera should be of higher quality and include infrared lighting due to the fast travel speed in the forward direction. The specific camera and ToF sensors selected can be seen below in 6.6.

Notification Devices

Also important for the operation of the DronebezORgd fleet is the ability for a drone to communicate its state, both for diagnostics or maintenance procedures, but also, for instance, in the event that a customer has forgotten to close the payload bay. Because of this, a multi-colour LED strip and an electromechanical buzzer are added to the avionics of the drone.

Avionics Selection Overview

Combining the components selected in the previous three sections, the full list of the BezORger's avionics is established. This list can be seen summarised in Table 6.6, where the model of every included component can be seen accompanied by its count, cost, mass, and dimensions.

6.5.2. Verification & Validation

The verification of the avionics selection is very straightforward and has been done throughout the design of the avionics systems. The functions of collective avionics components have been verified both against the requirements to ensure compliance with the expected precision and reliability of the systems, as well as against similar drones to ensure a market-leading standard of quality and redundancy. Validation, on the other hand, is not in the scope of the project as it is only possible to validate once a prototype of the BezORger has been built. It is assumed, however, that each avionics component presented in 6.6 has been independently validated by the manufacturer. Further validation will be done in the future once prototypes are available.

Table 6.6: Summary of all selected avionics components

Component Type	Model Name	Count [-]	Cost [€]	Mass [g]	Dimensions [mm]×[mm]×[mm]
Flight controller	Navio2	1	199.00	77	65 × 55 × 10
Computing interface	Raspberry Pi	1	44.33	54	85 × 56 × 20
Additional GNSS	Navio2 GNSS Antenna	1	12.90	30	45 × 35 × 15
Remote ID	Dronetag DRI	1	49.00	2	23 × 16 × 5
Standard stereo camera	Waveshare IMX219-83	5	59.99	23	85 × 24 × 20
Front-facing stereo camera	Ultraleap Stereo IR 170	1	283.41	22	145 × 19 × 11
Time-of-flight sensor	Grove VL53L0X	2	8.58	8	20 × 20 × 11
LED strip	Matek 2812 LED Strip Slim[13]	1	5.95	<20	57 × 8 × 4
Buzzer	AI-1223-TWT-3V-2-R[14]	1	1.12	2	12 × 12 × 7.5
Total			912.82	<298	

6.6. Recommendations

This chapter concludes with an overview of the recommendations the team has for any future design phase or project. As is clear from the preceding sections, the testing of the design of all parts of the payload handling subsystem is crucial for validating the design and models used to size them. For this, functional prototypes of all parts should be made and subjected to a controlled and representative environment. Examples of such tests would include assembling the thermal insulation structure and payload delivery mechanism, loading it with pizzas or other representative warm material, and monitoring the rate of heat rejection and any deformation of the mechanism structure.

As the component costs for the payload delivery, thermal insulation, and wingspan reduction are relatively low, multiple prototypes can be made as a form of sensitivity study to see how the performance or functionality is affected by varying certain design parameters. Validating that the correct avionics equipment has been chosen is also important, as its high overall cost should be justified before the whole DronebezORgd fleet is manufactured.

For the continuation of the design process, it is recommended to develop a thermal simulation model to quantify the thermal performance of the payload handling subsystem. Using this, mass usage and material choice can be further optimised. This would also allow the evaluation of various combinations of multi-layer insulation architectures and PCMs and their effect on the final design. Next to this, the use of various perception sensors in low-light or night-like conditions should be investigated to allow the drone to operate for extended parts of the day or year. A further study on wing folding and locking mechanisms is also recommended so that the reliability, mass, and safety of the subsystem can be optimised.

[13] URL `https://droneshop.nl/mateksys-2812-led-strip-slim-2pcs` [cited 17 June 2025]

[14] URL `https://www.digikey.nl/en/products/detail/pui-audio-inc/AI-1223-TWT-3V-2-R/5011391` [cited 17 June 2025]

7 Propulsion & Powertrain

This chapter focuses on the design, analysis, and V&V of the propulsion subsystem, which includes the horizontal and vertical propellers, the motor, the battery, as well as electrical components aiding in the transfer of power.

An overview of the design approach, along with the assumptions and simplifications made, is presented in Section 7.1. The propeller design and analysis are discussed in Section 7.2, covering airfoil choice, rotor layout, and blade design. The design and selection of the remaining powertrain components, including the motor, the battery, and other electronics, is presented in Section 7.3. Lastly, Section 7.4 presents recommendations to optimise the propulsion subsystem design and perform higher accuracy analyses.

7.1. Design Approach & Assumptions

The design of the propulsion subsystem was split into the propeller design and the design of the rest of the powertrain.

The Blade Element Momentum Theory (BEMT) was used to perform propeller analysis and subsequent design. A BEMT analysis tool based on a study by Goyal et. al. was used [15][16]. BEMT is not a high-fidelity method, and the tool used relies on the following assumptions:

- **AS-PP-01:** The flow over each annular (radial) ring is independent of every other annular ring.
- **AS-PP-02:** Wake expansion, yaw, and tip losses are neglected.
- **AS-PP-03:** The flow is non-turbulent and steady.
- **AS-PP-04:** A low free-stream velocity (between 1 and 3 m/s) accurately simulates hover conditions.

The analysis tool was then used to perform parametric studies on the blade geometry and design. The following ordered list of design variables then structured the design process:

- Number of propellers, n_{prop}
- Number of blades, n_b
- Blade airfoil profile
- Reference pitch angle at 70% outer radius, θ_{ref}
- Radial twist distribution with respect to θ_{ref}, $\theta(r)$
- Radial chord distribution, $c(r)$

Parametric studies revealed insights into the relation between these variables and the performance of the propeller (thrust produced and power drawn), which was used to arrive at design values for each variable. Furthermore, for any parameter that was yet to be designed, the values corresponding to the scaled TUD XProp were used instead. This approach implicitly makes the following assumptions:

- **AS-PP-05:** The impact of a design variable on the propeller performance is independent of other design variables.

- **AS-PP-06:** The TUD XProp size and design parameters are similar enough to BezORger's eventual propeller design for it to accurately reflect relations between parameters.

Following the propeller design, the motor and battery were selected. The motor selection was done based on the required rotation rate of the rotor blades and the required torque. The battery design followed from the voltage and power draw of the motor and propeller.

The propeller was designed so as to optimise for efficiency for the specific thrust and performance requirements on BezORger, given its mission profile, while avoiding over- or under-designing. On the other hand, the motor and battery were selected from commercially available options, as they entailed fewer design variables. So, close-to-optimal choices could still be achieved through off-the-shelf components.

7.2. Propeller

Given the limitations on the size of the aircraft, a configuration of **four propellers**, each with **four blades**, was chosen. More than four propellers would require close positioning, resulting in significant interference between propellers and a decrease in performance. Fewer than four propellers hinder controllability of the drone, which is critical for safety in an urban environment. Then, RQ-PP-03 dictates that the maximum thrust generated by each propeller should exceed 56.4 N.

$$T_{max} = \frac{2 \cdot MTOW \cdot 9.81}{n_{prop}} = \frac{2 \cdot 11.5 \cdot 9.81}{4} = 56.4\,\text{N}$$

Furthermore, to ensure safety, **circular ducts** were installed around each propeller. These also have the added benefit of increased thrust and efficiency, albeit at the cost of added mass [17].

7.2.1. Airfoil Selection

The operating Reynold's number for BezORger's propellers will be in the order of $\sim 10^4$ to 10^5. As such, for the propeller blades, it is desirable to use an airfoil with a large camber for the following reasons:

- They produce more lift at low Reynold's numbers.
- They produce more lift at smaller α, requiring a smaller propeller blade pitch.
- They tend to have a higher maximum C_L/C_D, resulting in higher efficiency.

As such, a preliminary selection of cambered airfoils commonly used in propeller blades was made, as presented in Table 7.1 (airfoil data from AirfoilTools[1]). To maintain ease of manufacturing, using multiple airfoils along the span of the blade was rejected, and only **single-airfoil profiles** were investigated.

Table 7.1: Preliminary selection of cambered airfoils for propeller blades

Airfoil	Max camber	Max $\mathbf{C_L/C_D}$	$\alpha_{\mathbf{L/D}}$	$\mathbf{C_{L_{max}}}$	$\alpha_{\mathbf{s}}$
NACA 4412	4.0	36.1	8.50	1.26	14.75
NACA 6409	6.0	36.0	9.00	1.35	9.00
NACA 6412	6.0	29.6	4.25	1.32	12.25
MH114	6.4	26.0	7.00	1.45	12.50
Clark X	3.3	36.4	7.75	1.19	9.75
Clark Y	3.4	36.5	7.50	1.27	12.25
S1223	8.1	42.3	3.25	1.80	8.00

[1] URL: `http://airfoiltools.com/`

To perform a more detailed analysis on the effect of airfoil choice on propeller performance, BEMT simulations were performed for each airfoil for varying radii, with the other design parameters set to the default values of the scaled TUD XProp and the operating conditions set to those during hovering flight [16]. The **radius was fixed to 20.3 cm** — the same as the scaled XProp radius — based on RQ-PP-01. The rotation rate of the propeller was set to **100 Hz**, typical for drones of this size. The dependence between the radius of the propeller and the thrust generated is shown in Figure 7.1.

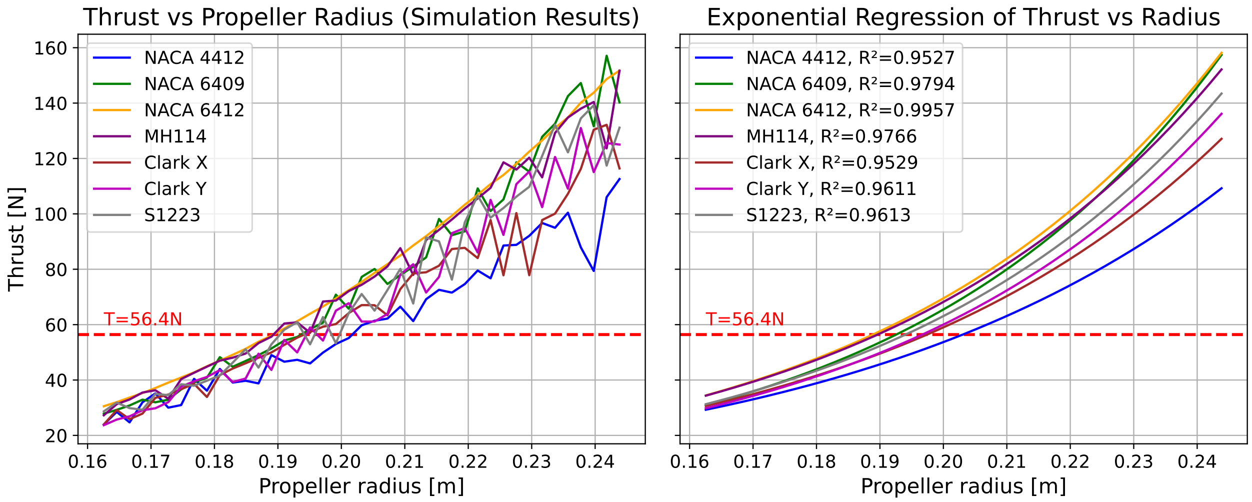


Figure 7.1: Thrust generated vs. propeller radius (and exponential regression)

As shown in Figure 7.1, the airfoil choice does not affect the radius to generate the required thrust of 56.4 N (RQ-PP-03) significantly; the required radii lay between 19.0 cm and 20.2 cm. As such, the relation between the power drawn and the thrust output was also investigated, as shown in Figure 7.2.

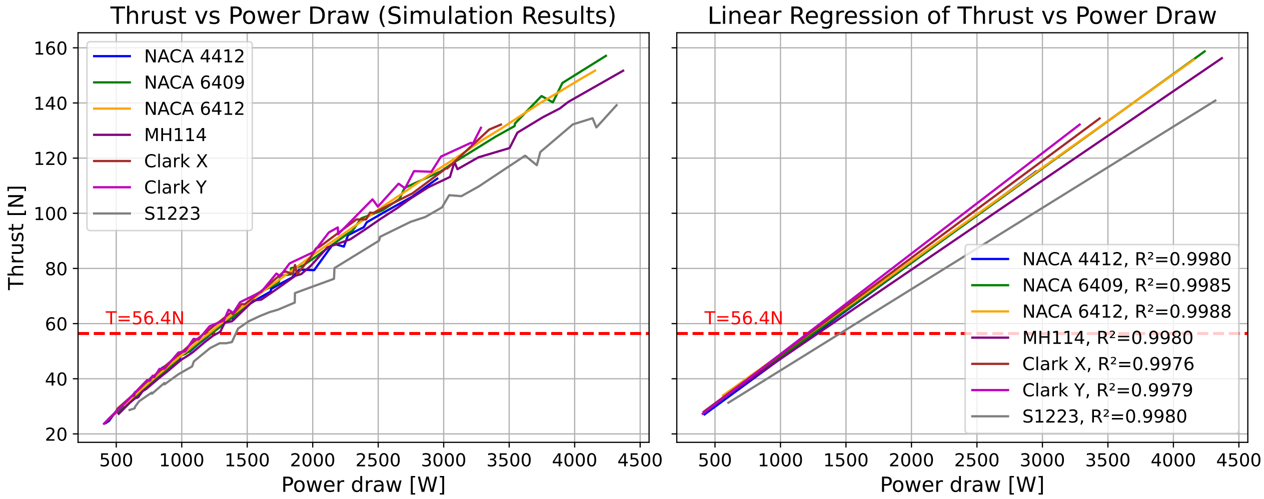


Figure 7.2: Thrust generated vs. power drawn (and linear regression)

It can be observed in Figure 7.2 that the thrust generated depends much more strongly on the power drawn than on the propeller radius. To generate the required 56.4 N thrust, the power draw ranged from 1,205 W for the Clark Y airfoil to 1,454 W for the S1223 airfoil. As such, the **Clark Y airfoil was chosen** for its high thrust-to-power ratio. The selection was done on the basis of power draw at maximum thrust, since the critical mission phases for energy consumption are takeoff and landing as opposed to transition or cruise.

7.2.2. Blade Design

With the Clark Y airfoil selected, the pitch angle and subsequently the twist distribution were defined. The pitch angle is measured at a reference location of $0.7R_{prop}$ from the centre of the hub, and the twist distribution is defined with respect to the pitch. The primary effect of the pitch angle is on the local AoA at each point along the blade, thus affecting the thrust generated. These effects were analysed for the chosen Clark Y airfoil at hovering conditions, still using the twist and chord distribution of the TUD XProp [16][15].

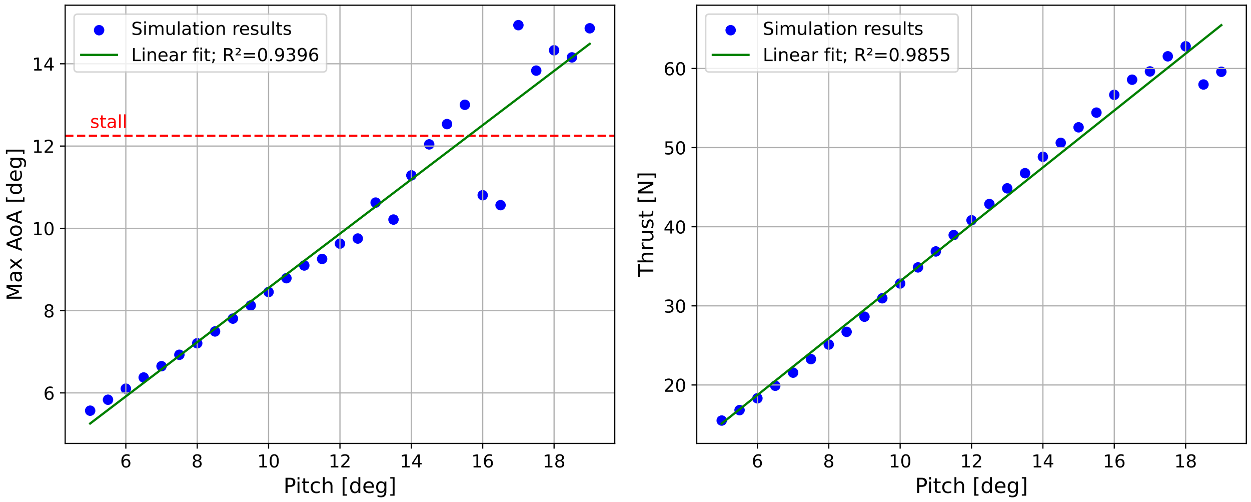


Figure 7.3: Effect of varying pitch on maximum AoA and thrust generated

As observed in Figure 7.3, while the thrust monotonically increases with reference pitch angle for the plotted values, stall is observed locally along the propeller blade at values exceeding 15.5°. So, a **pitch angle of 15.5°** was selected, maximising thrust while avoiding stall.

With the reference pitch fixed, twist distributions were explored. The twist distribution of the TUD XProp blade bilinearly decreased from 26.82° to $-8°$ with distance from the hub centre [15]. As such, fixed multiples of the linear twist profile were explored. However, due to the pitch already being at the stall limit, the twist could only be scaled down. The effect of linearly scaling down the twist on the thrust, power, and figure of merit is shown in Figure 7.4.

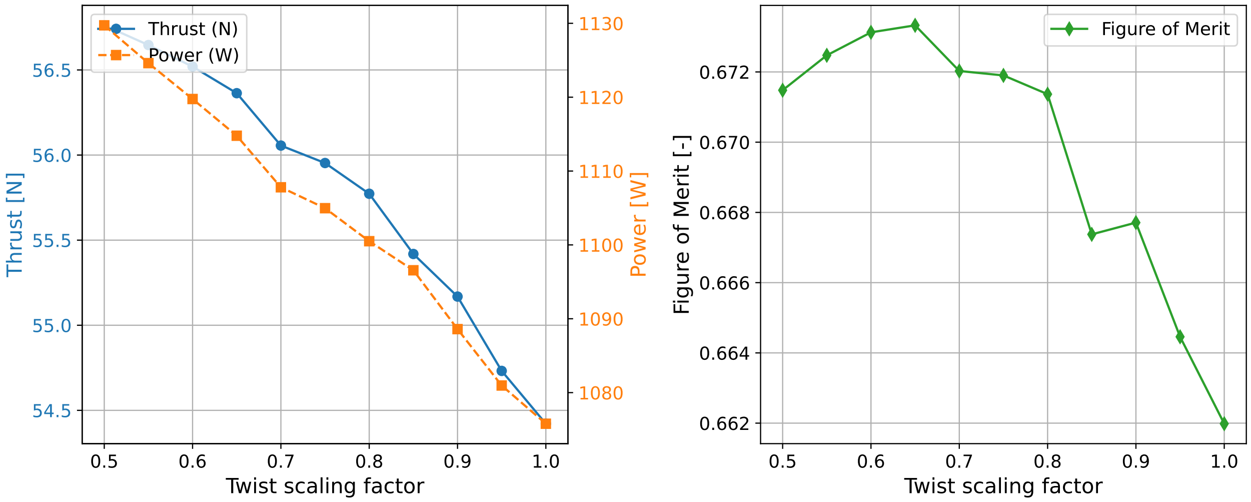


Figure 7.4: Effect of twist distribution on thrust generated, power draw, and figure of merit

As seen in Figure 7.4, for the values of the twist scaling factor explored, the thrust generated (and power draw) monotonically increased with decreasing twist. However, the figure of merit saw a peak

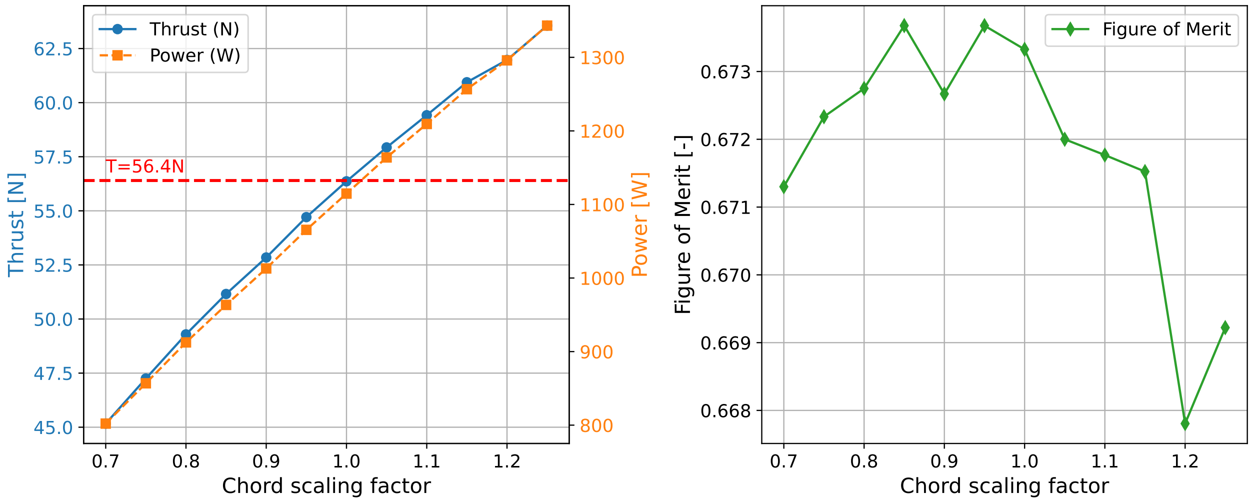


Figure 7.5: Effect of chord distribution scaling on thrust generated, power drawn, and figure of merit

at a factor of 0.65. As such, the **twist distribution** chosen (with respect to the reference location at $0.7R_{prop}$) was **0.65 times the TUD XProp twist**.

Lastly, the effect of the radial chord distribution was investigated. Again, the TUD XProp chord distribution was linearly scaled by different factors to see how the performance varied. The results are shown in Figure 7.5.

Evidently, Figure 7.5 reveals that the thrust increases almost linearly with the chord distribution. Notably, the effect on the figure of merit is much less pronounced than that of twist distribution. As such, the thrust requirement was used to size the chord distribution — the **chord distribution** was chosen to be the **same as that of the TUD XProp**, as it was the smallest distribution that fulfilled the thrust requirement [15].

7.2.3. Horizontal Propeller

A similar approach was followed to design the horizontal propellers. The only difference was in the thrust required and hence the rotation rate. For ease of manufacturing, all design parameters for the horizontal propeller were kept the same as the vertical propeller; only the sizing was redone based on RQ-PP-04 and the cruise operating conditions instead of hover. The former dictates that the horizontal propeller should be able to deliver a thrust of 56.4 N.

$$T_{max} = 0.5 \cdot MTOW \cdot 9.81 = 0.5 \cdot 11.5 \cdot 9.81 = 56.4\,\text{N}$$

Once more, the thrust generated was plotted and regressed against the propeller radius, as shown in Figure 7.6.

It can be seen from Figure 7.6 that a **propeller radius of 22.2 cm** is required to generate the required thrust.

7.2.4. Verification & Validation

The MATLAB BEMT simulation code used was based on peer-reviewed work by Goyal et. al [16]. Even so, verification was performed by means of unit tests, which included the following checks:

- Ensuring the polar data and blade geometry files were read correctly
- Performing sensitivity analyses by slightly varying inputs to ensure stability of the model
- Critically analysing relations obtained between parameters against physical laws

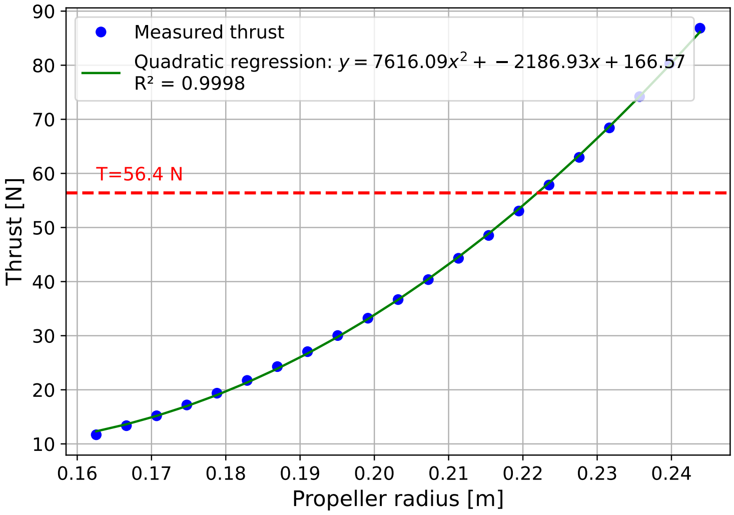


Figure 7.6: Effect of varying horizontal propeller radius on thrust generated

In addition to the model, however, its usage for the design process was also verified. The vertical propeller design is presented in Figure 7.7, and the design parameters and specifications for both propellers are summarised in Table 7.2.

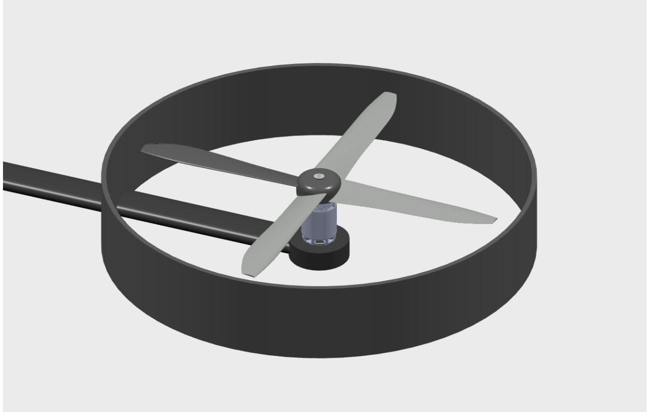


Figure 7.7: CAD render of vertical propeller design

Table 7.2: Vertical and horizontal propeller design summary

Design parameter	Vertical propeller	Horizontal propeller
Number of propellers	4	1
Propeller radius	20.3 cm	22.2 cm
Hub radius	3.2 cm	3.5 cm
Ducts around propeller	Circular	Circular
Number of blades per propeller	4	4
Blade airfoil	Clark Y	Clark Y
Reference pitch angle	15.5°	15.5°
Radial twist distribution	see Figure 7.8	see Figure 7.8
Radial chord distribution	see Figure 7.8	see Figure 7.8
Maximum rotation rate	100 Hz	100 Hz
Maximum thrust generated	56.4 N	56.4 N
Maximum power draw	1,114.8 W	1,598.7 W
Maximum torque	1.77 N m	2.54 N m

Figure of merit	0.673	0.678

As specified in Subsection 7.2.3, the twist and chord distributions of both the vertical and the horizontal propeller were the same along the radial direction (when plotted against the radius-normalised distance from hub centre); Figure 7.8 graphically shows this variation.

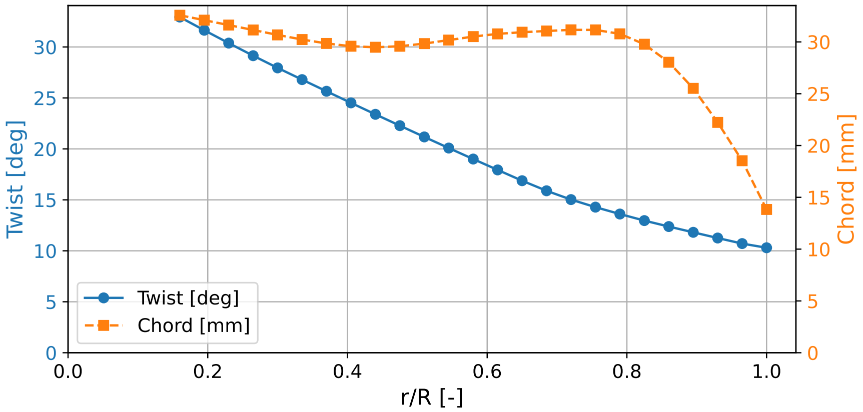


Figure 7.8: Twist and chord distribution of both the horizontal & vertical propellers

To verify that the thrust requirement RQ-PP-04 was met, a regression was made between the propeller setting (rotation rate in Hz) and the generated thrust. To further analyse the performance, the thrust vs. power required relation was also modelled. These results are presented in Figure 7.9.

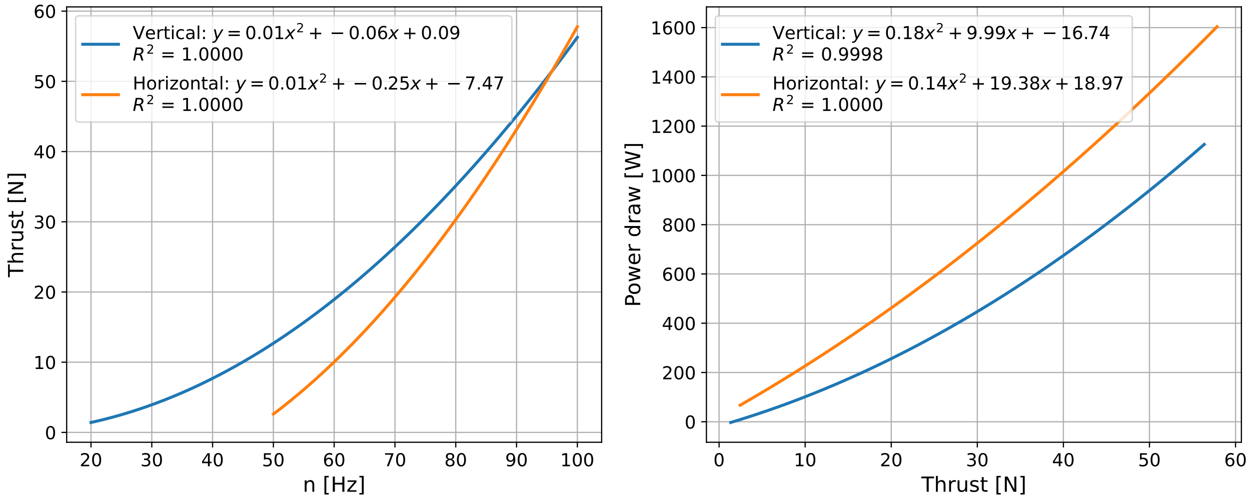


Figure 7.9: Relation between rotation rate, thrust generated, and power draw for vertical and horizontal propellers

It can be seen from Figure 7.9 that at a rotation rate of 100 Hz, the thrust requirement is exactly met, verifying that the propellers are neither under- nor over-designed.

7.3. Powertrain

The propeller design gave way to the remaining powertrain components. Specifically, it defined requirements on the required rotation rate of the blades, the torque, and the power it operates at. This imposed constraints on the sizing and selection/design of the motor and subsequently the battery and other electrical components.

7.3.1. Motor & Electronic Speed Control

The motors for the propulsion subsystem need to meet the following requirements imposed by the propellers:

- Allow for a rotation rate of at least 6000 rpm (= 100 Hz)
- Supply a torque of at least 1.77 N m to the vertical and 2.54 N m to the horizontal propeller
- Operate at a power of at least 1,114.8 W for the vertical and 1,598.7 W for the horizontal propeller
- Operate at a thrust of at least 56.4 N

In the interest of ease of procurement of components during manufacturing, it was decided to:

- Select an off-the-shelf motor (due to the relatively low number of design variables)
- Select the same motor for both the horizontal and vertical propeller (due to similar magnitudes of required RPM, torque, and power)

The **EVCraft High Torque 6016 Electric Brushless DC Motor**[2] was selected as the motor of choice for DronebezORgd as it met each requirement. An overview of its performance limits and specifications can be found in Table 7.3; note that the rotation rate, torque, power, and thrust refer to the maximum values respectively.

Table 7.3: Performance limits and specifications of the EVCraft High Torque 6016 Electric Brushless DC Motor

Rotation rate	Torque	Power	Thrust	Voltage	Weight	Price
125 Hz	3 N m	1,800 W	63.8 N	27 V	0.39 kg	€50

To ensure compatibility with the selected brushless DC motor, the ESC must meet several key requirements:

- Support the operating voltage of 27 V
- Deliver sufficient continuous and peak current to match the motor's 1,800 W peak power
- Provide reliable communication with the flight controller
- Include safety features such as over-current and over-temperature protection

The **XRotor H80A ESC**[3] meets these requirements, offering up to 40 A continuous current and 51.8 V maximum voltage support.

7.3.2. Battery

In compliance with RQ-PP-09, the battery of the drone should have an energy of at least 1.69 MJ, assuming an electric efficiency of 0.95.

$$E_{bat} = \frac{E_{mission}}{\eta_{electric} \cdot (1 - \text{reserve})} = \frac{1.28 \cdot 10^6}{0.95 \cdot (1 - 0.2)} = 1.69\,\text{MJ} = 468\,\text{W h}$$

Moreover, the battery should also be able to generate a potential difference of 27 V for the motors to function as required. A trade-off was conducted to analyse battery types based on energy density (0.4), power density (0.1), safety (0.3), and cost (0.2), shown in Table 7.4.

[2] URL: `https://evcraft.en.made-in-china.com/product/lEipLUzPorYA/China-High-Torque-6016-Electric-Brushless-DC-Motor-27V.html`

[3] https://www.hobbywingdirect.com/products/xrotor-pro-h80a-esc

Table 7.4: Trade-off of battery types for BezORger

Criterion Battery Type	Energy Density [W h/kg]	Power Density [W/kg]	Safety	Cost
Li-ion (NMC)	220	800	moderate; thermal runaway risk	moderate
Li-Po	180	1300	moderate; flammable components	low
LiFePO$_4$	140	1000	high; thermally stable	low
Solid-State Li-ion	300	1200	very high; non-flammable	very high
NiMH	60	300	high; no thermal risk	low

From the trade-off, it was found that LiFePO$_4$, Li-Po, and Li-ion (NMC) batteries were all viable options. As such, all three were explored, and the **DJI Mavic 2 Intelligent Flight Battery**[4], a Li-Po battery, was chosen. The specifications of the battery are presented in Table 7.5.

Table 7.5: Specifications of the DJI Mavic 2 Intelligent Flight Battery

Energy	Charge	Voltage	Weight	Price
59.3 W h	3,875 mA h	15.4 V	0.30 kg	€103

To meet the energy requirement, **8 units** of this battery are needed, resulting in a total energy of 474.4 W h. To meet the voltage requirement, two batteries are connected in series, yielding a resulting voltage of 30.8 V. So, the eight batteries are arranged in a **2 × 4 layout**, with two in series and four such series connections in parallel.

7.4. Recommendations

The current propulsion subsystem design relies on several simplifying assumptions that could be addressed to improve accuracy and performance. Firstly, the BEMT analysis tool neglects wake expansion, yaw, and tip losses (AS-PP-02), which could lead to discrepancies between predicted and actual propeller performance. Future work should incorporate these effects through higher-fidelity computational fluid dynamics (CFD) simulations. Moreover, the assumption of steady, non-turbulent flow (AS-PP-03) may not hold during dynamic flight phases such as transition or gust encounters. Transient analyses or unsteady aerodynamic models should be explored to account for these effects.

A critical limitation of the current approach is the sequential design of parameters under the assumption of independence (AS-PP-05). This does not account for potential interactions between variables such as airfoil selection and twist, and chord distribution. To achieve a more optimal design, a multidisciplinary optimisation should be performed by simultaneously varying all parameters to maximise the thrust-to-power ratio while meeting constraints. Furthermore, the reliance on the TUD XProp as a reference design (AS-PP-06) introduces uncertainties, as BezORger's mission profile and operational conditions differ. Future iterations should validate the scaled parameters through wind tunnel testing or small-scale prototypes.

Lastly, the propulsion subsystem will require resizing following the system-level design iteration in Section 13.1. Currently, the design is based on an estimated MTOW of 11.5 kg; the propeller geometry, motor selection, and battery capacity must be re-evaluated to ensure compliance with revised thrust and power requirements based on the updated MTOW. To improve accuracy, better models should be used to estimate the effect of circular ducts on the thrust and power.

[4] URL: `https://www.amazon.es/ZahoTse-Mavic-Capacity-Intelligent-Replacement/dp/B0CBCT61NN`

8 Noise

For DronebezORgd to become a pioneer in the drone food delivery sector, public perception is key. To that extent, the noise emissions need to be below levels that cause public disturbance. In particular, BezORger has to comply with the requirements specified in Table 2.4. This chapter tackles the estimation of noise emissions, starting off with defining the approach used and the assumptions made in Section 8.1. The Hanson model to estimate tonal noise and its results are discussed in Section 8.2 [18]. The verification and validation of the analysis is discussed in Section 8.3, and a critical examination of the methods used, along with recommendations for future work, is presented in Section 8.4.

8.1. Design Approach & Assumptions

This section outlines the approach used to estimate the noise emissions using an implementation of the Hanson model by Goyal et. al. [19]. This method focuses on the tonal noise generated by the propeller, which consists of discrete frequencies describing pure tones, as opposed to broadband noise, which corresponds to a wide range of frequencies. The following assumptions were made in the Hanson model and its implementation:

- **AS-NOISE-01:** The observer is located in the acoustic far field ($> 10 \cdot r_{prop}$).
- **AS-NOISE-02:** Sound propagation is linear and sound waves do not interact with each other.
- **AS-NOISE-03:** There is a uniform and steady air inflow to the rotor.
- **AS-NOISE-04:** Blade-wake interaction and broadband noise are neglected according to Akiwate et al. [20].
- **AS-NOISE-05:** The model assumes that the propeller blades are uniform, slender, and thin.
- **AS-NOISE-06:** The model assumes the load distribution along the chord is parabolic.
- **AS-NOISE-07:** The total sound pressure level is summed using logarithmic summing.
- **AS-NOISE-08:** The adjusted sound pressure levels L_{PA} are calculated based on Hanson's tonal noise at the 90° angle and the blade passing frequency (BPF) in Equation 8.1.
- **AS-NOISE-09:** The noise estimation assumes that the interaction of the vertical duct is neglected.
  Since the duct around the vertical propellers obstructs the noise sound waves produced by the blades, this assumption results in a conservative noise estimate.

8.2. Hanson Model

The Hanson model analytically predicts the tonal noise generated by rotating propeller blades, based on aerodynamic forces obtained from blade loading distributions [18]. The sound pressure levels L_p, measured in dB, at each harmonic frequency have been computed using blade geometry, aerodynamic load distributions, and operating conditions such as rotational speed and observer location. The total tonal noise $L_{P,\text{total}}$ is obtained by summing the contributions from multiple harmonics, resulting in a frequency-dependent and directional noise prediction.

8.2.1. Analysis

The tonal noise is dominant in the plane of the propeller rotation (at 90°), while broadband noise is dominant along the axis of rotation (at 0° and 180°), as shown in Figure 8.1, and the sound field is nearly symmetrical [19]. As such, during take-off and landing, tonal noise dominates over broadband noise due to the four vertical propellers, allowing for the use of the Hanson model.

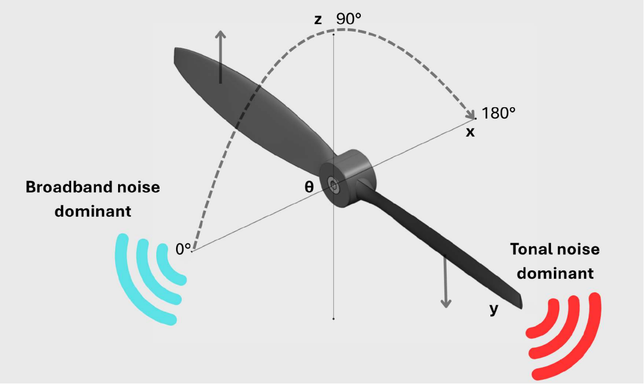


Figure 8.1: Noise propagation around a rotating propeller [19]

The Hanson model calculates the tonal SPL L_p generated by a single rotor in dB. This was first scaled by the number of propellers (using logarithmic summing) and then converted to dBA. The latter is a frequency-dependent scaling known as the A-weighting that reflects perceived sound levels by humans better than dB [21, 22], shown in Equation 8.1.

$$L_{PA}[dBA] = L_P[dB] + A(f) \tag{8.1}$$

Here, $A(f)$ is the frequency scaling term and is closely related to the blade-passing frequency (BPF), which is the product of the rotational frequency of the propeller in Hertz and the number of blades.

8.2.2. Results

The noise generated depends largely on the thrust; as such, estimations were made for cruise, take-off, and landing. Figure 8.2 presents the SPL along different values of the viewer angle θ.

As expected, it can be seen in Figure 8.2, the tonal noise is dominant at $\theta = 90°$, with a sharp drop towards the extremes, serving as a sanity check. Take-off at MTOW is critical for noise emission, with a maximum SPL of 69.0 dB at a distance of 2 m. However, after the A-weighting, take-off with zero payload was critical, with an adjusted maximum SPL of 55.3 dBA. Additionally, the maximum noise during cruise is 20.5 dBA, which satisfies RQ-09-01-01.

8.3. Verification & Validation

This section discusses the verification and validation of the Hanson noise estimation method.

Verification

Goyal's implementation of Hanson's model [19] has been described and applied in a peer-reviewed study [23] published by TU Delft researchers in the journal Aerospace Science and Technology[1]. This implies that it has already been partially verified by the scientific peer review process.

Additional verification tests were performed covering the following aspects: ensuring the polar data and blade geometry files were read correctly, performing sensitivity analyses by slightly varying inputs

[1] URL `https://www.sciencedirect.com/journal/aerospace-science-and-technology/vol/147/suppl/C` [cited June 12, 2025]

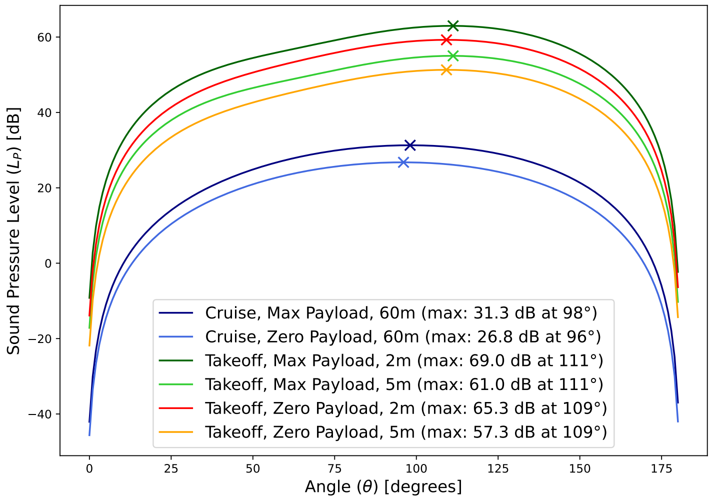


Figure 8.2: Maximum Sound Pressure Levels (L_P) during all critical mission phases

to ensure stability of the model, and critically analysing relations obtained between parameters against physical laws.

Furthermore, the presence of the Doppler effect in Figure 8.2 indicates that the numerical model captures the underlying physics of sound propagation from a moving source. This demonstrates that the mathematical formulation and computational implementation behave as anticipated against theoretical principles.

Validation

The validation of the Hanson noise estimation model could be performed through the comparison of the model results against experimental data. Furthermore, to analyse the effects of AS-NOISE-09, experiments could be performed comparing the effects of the propeller ducts on noise emissions.

8.4. Recommendations

To enhance the noise estimation analysis, several improvements can be considered for future work. To start off, AS-NOISE-01 prevents accurate noise estimation in conditions below $<10\ r_{tip}$. In the future, experimental tests should be conducted to validate the noise levels for these distances. Furthermore, AS-NOISE-03 does not hold during take-off and landing phases when the drone is within approximately two meters of the ground. When the air pushed downward by the rotor hits the ground and reflects upward, it creates turbulent recirculating flows near the blades. Additionally, acceleration and deceleration during transitional flight phases introduce unsteady flow conditions. Therefore, the inflow is no longer uniform or steady in these regimes. Future experimental validation is planned to assess noise levels during these critical flight stages.

The current model only calculates the tonal noise AS-NOISE-04 but neglects the broadband noise. In the future, the broadband interaction should be calculated using the model by Thurman, Zawodny, and Baeder [24]. The outcome shall validate whether tonal noise is the dominant factor.

9 Aerodynamics

Aerodynamics are an essential part of the drone, dictating the stall behaviour, stall angle, wing platform, drag, and lift of the drone. This chapter covers the choices and methods that were used to get to the aerodynamic coefficients of the preliminary design. The design approach and assumptions made are presented in Section 9.1. A preliminary sizing was conducted in Section 9.2, and further design iterations are presented in Section 9.4. Finally, limitations of the current method and recommendations for future work are discussed in Section 9.5.

9.1. Design Approach & Assumptions

At a high level, aerodynamics is tasked with determining the aerodynamic coefficients of the BezORger, which include the lift coefficient and the drag coefficient. Furthermore, an effective wing planform with a wing profile should be designed. This was done in 4 distinct steps:

1. **Preliminary Sizing**: This step is needed to get a starting point for a wing platform.
2. **Airfoil Selection**: The airfoil selection is a crucial step in the design process. The needs of the drone will be traded off to find the optimal airfoil with the current available information.
3. **CFD Analysis**: A CFD analysis is performed to determine aerodynamic coefficients. The used settings and model will be discussed.
4. **Integration**: The final step is to integrate the CFD results into the design and convert all results to a usable data set to be used in the iterations of the design.

Some assumptions are also made in this chapter; a summary is given in the following list.

- **AS-AERO-01:** All CFD analysis will be performed without the propellers.
- **AS-AERO-02:** Thermal effects will be neglected.
- **AS-AERO-03:** Changes in wing platform will result in linear coefficient changes.

9.2. Preliminary Sizing

This section covers the analysis of the preliminary sizing in Subsection 9.2.1, and the verification of these methods in Subsection 9.2.2.

9.2.1. Analysis

This subsection covers the analysis done for the wing position, the preliminary wing platform sizing, and finally, the airfoil selection.

Wing Positioning

The drone has a wing folding mechanism, which influences its wing platform and position. Due to this mechanism, the drone will need a high wing configuration, which will allow more clearance for the mechanism. Furthermore, the high wing configuration will allow more ground clearance when landed, requiring shorter landing gears. Finally, the high wing placement will also help to have clean incoming air that is not disturbed by the propellers, as they are placed on the lower part of the drone, rendering AS-AERO-01 valid.

Wing Parameters

To have a starting point, two parameters have to be calculated: the tip and root chord. To estimate these parameters, the aspect ratio of the wing has to be computed first, which is done with Equation 9.1, where b is the wing span, which is assumed to be 3 m as per requirement 4.10, the S is the wing platform area which was computed from weight estimations.

$$AR = \frac{b^2}{S} \tag{9.1}$$

Subsequently, the root and tip chord can be computed from Equation 9.2, using the wing span and the wing surface area. In Equation 9.2, λ is the taper ratio, which was assumed to be 0.4 as this will give an almost elliptical wing lift distribution[1], which is optimal.

$$c_r = \frac{2S}{(1+\lambda)b}, c_t = \lambda c_r \tag{9.2}$$

After these calculations, the root chord is found to be 0.54 m and the tip chord is found to be 0.22 m.

Airfoil Choice

The airfoil choice is an important aspect of the drone's aerodynamics; it will influence all the coming results and will greatly influence the overall design. The National Advisory Committee for Aeronautics (NACA), has an extensive list of wing profiles and their first order estimations. Based on this database, the wing profile has been selected. This section will go over the procedure used to select the airfoil.

An online database[2] was found in order to perform calculations and narrow down the options. The first choice was to only consider 4-digits NACA profiles. This choice was made due to the stall characteristics of this series — it has favourable stall characteristics, which are crucial for the drone as it will need to fly through a transition zone to go from hovering to normal flight. Furthermore, as the drone has to land in tight spaces, the goal will be to find the airfoil with the highest lift coefficient (C_L). This happened to be the NACA-6412 airfoil, in Figure 9.1, the outline of this airfoil can be seen.

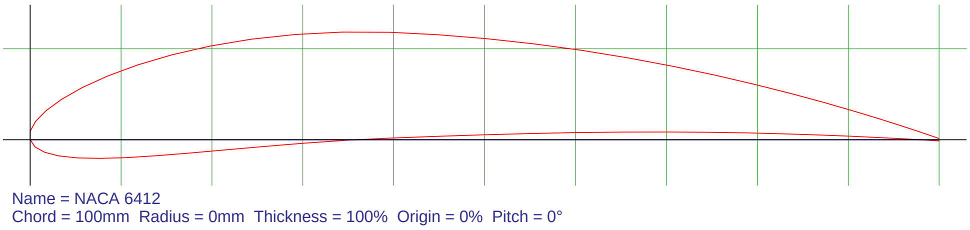

Name = NACA 6412
Chord = 100mm Radius = 0mm Thickness = 100% Origin = 0% Pitch = 0°

Figure 9.1: NACA-6412 airfoil outline

The NACA-6412 has a maximum C_l of above 1.5 before stalling, which is the maximum that was found in the aforementioned database. The C_l vs. α can be seen in Figure 9.2.

9.2.2. Verification & Validation

To verify the preliminary sizing, some simple unit tests were set up to do a sensitivity analysis on the inputs. For example, a check was to see if the wing surface area would rise with an increase of MTOW, and this was the case. Validation should be conducted once the model exists physically.

9.3. Computational Fluid Dynamics

This section will cover the analysis of the CFD analysis in Subsection 9.3.1, and the verification of these methods in Subsection 9.3.2.

[1] https://brightspace.tudelft.nl/d2l/le/content/512607/viewContent/2879495/View [cited 16 June 2025]
[2] https://splinecloud.com/repository/AlinaNeh/NACA_airfoil_database/ [cited 16 June 2025]

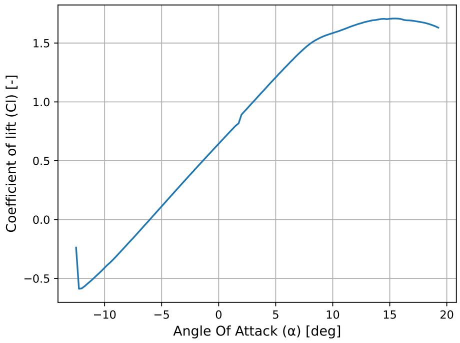


Figure 9.2: C_l vs. α curve for NACA-6412

9.3.1. Analysis

During the design of the drone, Computational Fluid Dynamics (CFD) was used, due to time constraints and the time that a CFD simulation has to run for, the design was simplified to not account for the propellers.

CFD Software Choice

The choice of this software was essential to the efficiency of the drone's aerodynamics; a poor choice could result in a lot of wasted time. The difficulty to learn the software was also taken into account, and in the end, 3 options were available: XFLR5, ANSYS, and 3DExperience. The issue with XFLR5 was that no propellers could be modelled in this software, which initially was still expected to be analysed. Finally, the SIMULIA software from 3DExperience was chosen, as the CAD model of the drone is also made in this software; this would allow for quick iterations and changes as the drone evolves.

Model Settings

CFD is a complex topic, and not all models can be explored. The chosen model was the Shear Stress Transport k-ω (SST k-ω) model. This combines two models, one for the inner boundary layer and the other for the free stream region. The model uses the best of the two models, and due time constraint of the project, this was the chosen model.

The model of the mesh was a hex model; this was used because some tests resulted in the hex mesh building faster than the tetrahedron mesh with no significant differences in results. Furthermore, the minimum size of the elements is 16.236 mm, the maximum size is 162.358 mm, and a boundary layer was added to model this region more accurately with 4 layers, with a linear thickness growth starting at 0.4 mm. The size of these elements was automatically suggested by the software based on the CAD model given to it. In Figure 9.3, the outside of the mesh can be seen in Figure 9.4, where a section is shown where the mesh has been cut through its middle.

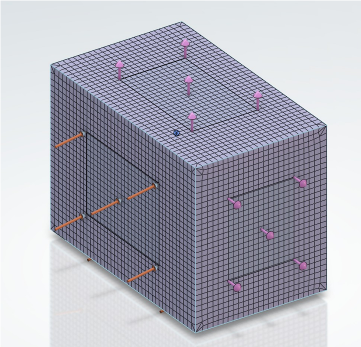


Figure 9.3: Outside view of the CFD mesh

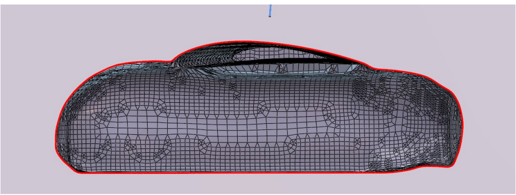


Figure 9.4: Inside view of the CFD mesh

In Figure 9.3, the orange arrows represent the incoming air at a speed of 15 m/s, this was a good approximation of the speed at which the drone would be flying. The pink arrows represent a pressure outlet, which effectively means that the air can flow out of those surfaces. This configuration was also proposed by the 3D Experience education manual and is a great way of having first-order estimations. Furthermore, as per assumption AS-AERO-02, thermal effects will be ignored as this simplifies the model further, and the effect of these will not be significant as the model is flying at low speeds and at ISA sea level conditions.

Procedure and Results

The expected output of the simulation is a C_L vs. α curve and an estimate for the drag. This means that the simulation needs to be run for each angle of attack; these angles have been discretized, and thus the following angles have been analysed: 0°, 3°, 6°, and 12°. Initially, simulations were run for higher angles, but above 12° the wing started to stall, which was not the scope of this simulation. Knowing the initial sizing of the wing platform, the first simulations could run. The CFD simulation only gave lift and drag forces over the whole drone; thus, using Equation 9.3 and Equation 9.4, the coefficients could be calculated as the geometry of the wing was known.

$$C_l = \frac{2L}{\rho S v^2} \tag{9.3}$$

$$C_d = \frac{2D}{\rho S v^2} \tag{9.4}$$

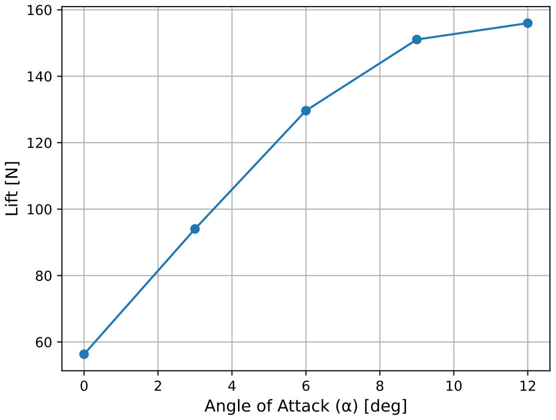


Figure 9.5: Basic configuration lift curve

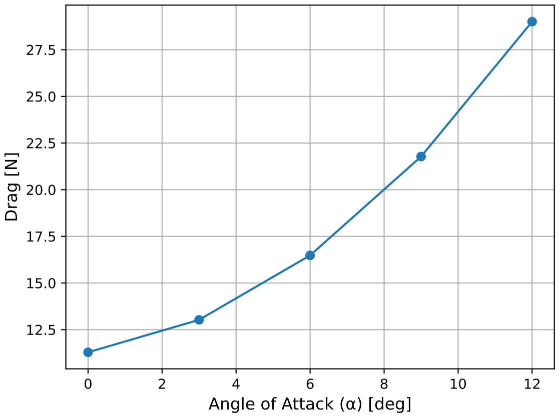


Figure 9.7: Basic configuration drag curve

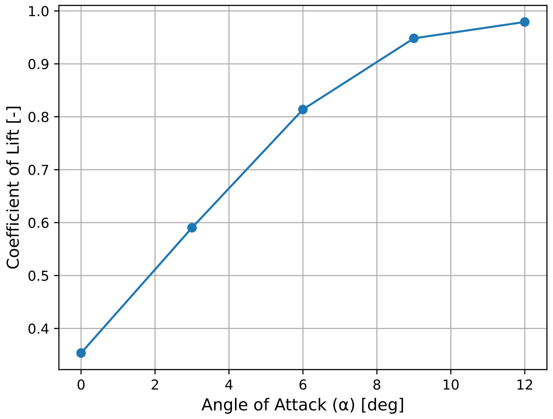


Figure 9.6: Basic configuration lift coefficient curve

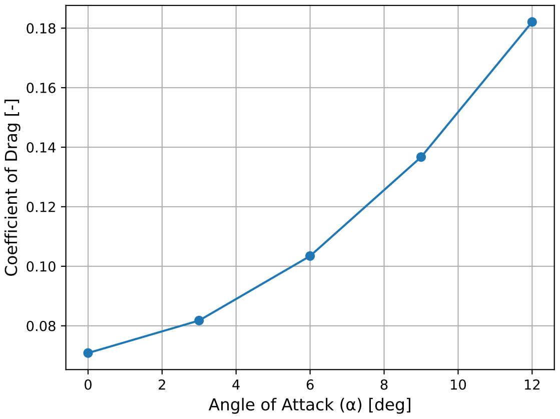


Figure 9.8: Basic configuration drag coefficient curve

In Figure 9.5, 9.6, 9.7, and 9.8, the word ‘basic’ refers to the configuration that was the result of the preliminary sizing; these plots will be used as a reference point for further comparisons. In Table 9.1, an overview of the basic configuration values is given.

Table 9.1: Basic configuration

Parameter	Value	Unit
Root chord (c_r)	0.54	m
Tip chord (c_t)	0.22	m
Span	3.0	m
Leading edge sweep	3.1	degree
Surface area	1.156	m^2

In order to know what parameter influences the curves in what manner, a sensitivity analysis was set up. Two additional simulations are set up, where only the root chord changes by ±0.10 m. This same procedure has been applied for the tip chord, this also changed by ±0.10 m. The final parameter was the wing sweep; the two used values were 0 ° and 10 ° .

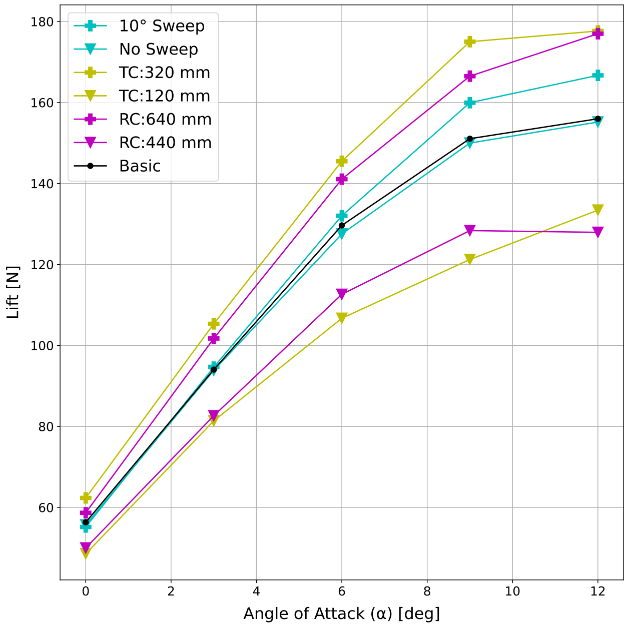


Figure 9.9: Comparison lift curves

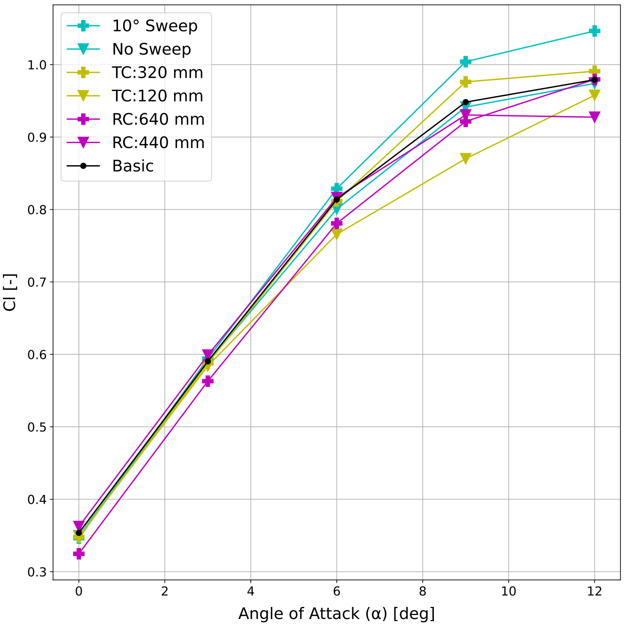


Figure 9.10: Comparison lift coefficient curves

In Figure 9.9, the angle of attack α versus lift force can be seen. The lines with the same colour are the same parameter change; one line will have a plus symbol, meaning that the parameter increased, the other line has a down arrow, meaning that the value of that parameter decreased. Figure 9.10, is the lift coefficient comparison. The difference between these two graphs is that the lift accounts for the increase/decrease in surface area due to the parameter change. The black line represents the unchanged design, named the ”basic” design from the preliminary sizing; its parameters can be found in Table 9.1.

In Figure 9.11 and 9.12, the same styling logic applies as before. One might notice that increasing any parameter results in a lower C_d, but the increase of the parameter results in a higher surface area, meaning that the drag force usually increases; the only exception is the wing sweep, as the sweep does not change the surface area of the wing.

9.3.2. Verification & Validation

Validation of aerodynamics is complicated without having the physical model tested in a wind tunnel. Verification is possible and was performed by doing the sensitivity analysis of all different parameters explained in Figure 9.3.1.

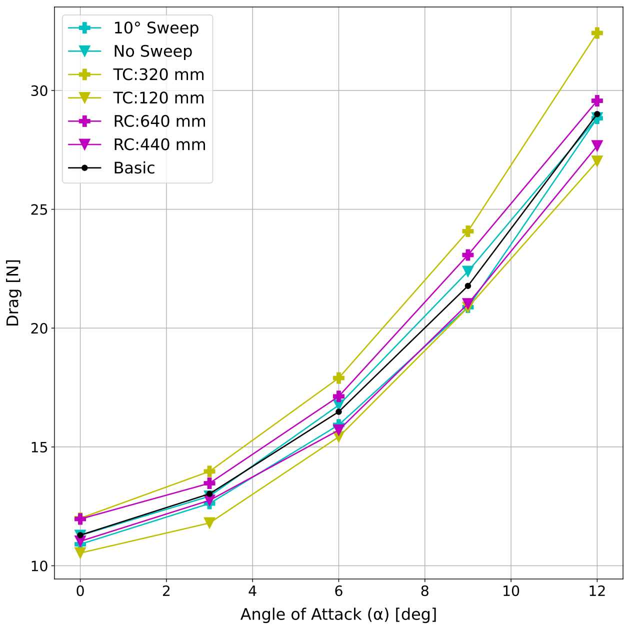


Figure 9.11: Comparison drag curves

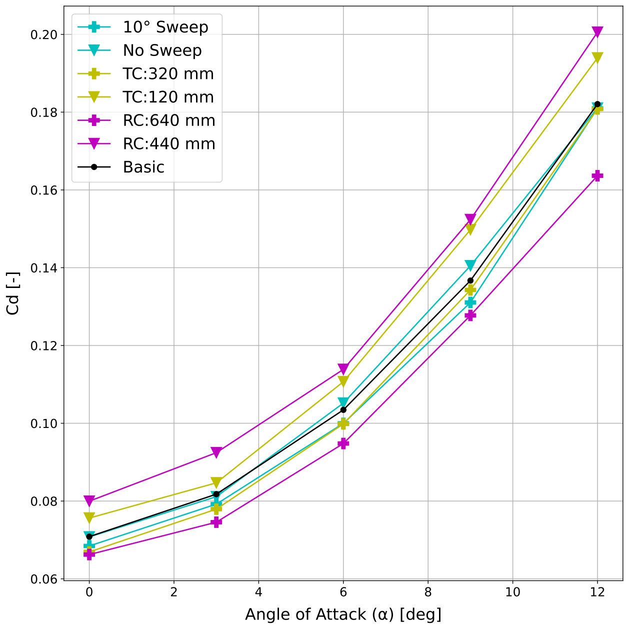


Figure 9.12: Comparison drag coefficient curves

9.4. Integration

This section will cover the analysis of the code integration and iterative loops in Subsection 9.4.1, and the verification of these methods in Subsection 9.4.2.

9.4.1. Analysis

In this analysis, the results of the CFD were transferred to the Python environment to allow for quick iterations of the design, this was performed in Subsection 9.4.1. The design of the wing is an iterative process and has been explained in Subsection 9.4.1.

Code Integration

CFD simulation results are only obtained for discrete values of planform parameters. However, when performing the iterative aerodynamic design of the drone, parameters are treated as continuous values, and aerodynamic coefficients are required at arbitrary values. So, a linear interpolation was used to compute the C_l when varying each of the root chord, tip chord, and sweep, resulting in a C_l vs. α graph for each parameter. Then, the curve for every parameter change was averaged to obtain an approximate C_l vs α curve. This methodology is also reflected in AS-AERO-03.

Design Loop

In order to find a revised wing platform from the initial weight, the lift formula is used with an assumed cruise AOA, the C_l can be found from this AOA using the "basic" lift curve. When the C_l is found the wing area can be found using Equation 9.5. The cruise speed is also the design cruise speed and is set as an input.

$$S = \frac{2MTOW}{\rho V_{cruise}^2 C_l} \tag{9.5}$$

After using Equation 9.5, the wing surface area is computed, assuming a taper ratio (λ), the root chord and tip chord can be computed.

$$c_r = \frac{2S}{b(1+\lambda)} \tag{9.6}$$

$$c_t = c_r \cdot \lambda \tag{9.7}$$

In Equation 9.6, the b is the span of the wing. After performing Equation 9.6 and Equation 9.7, the tip and root chord are found, which again means that a new C_L vs. α curve can be found. This again influences the C_L used in Equation 9.5. Finally, the last assumption is that the wing will have no sweep at all. Wing sweep is often used for planes cruising close to Mach 1, meaning that for this design, a wing sweep is not necessary and would only increase the structural mass. Once the change of the wing surface is less than $0.001\,\mathrm{m}^2$ the wing is considered as converged and will be given to the next steps described in Section 13.1.

9.4.2. Verification & Validation

Furthermore, looking at Figure 9.10, the black line lies approximately in the middle of both the root chord lines and tip chord lines, rendering the assumption made in Section 9.4 valid. Furthermore Figure 9.13, shows a pressure distribution over the drone; this result is the expected one, where high pressure is measured at the nose and lower on the suction side of the wing. Additionally, a simple simulation was run of a 2D airfoil of the NACA-6412 airfoil to verify the results with the one found online[3], which resulted in the CFD rendering correct values.

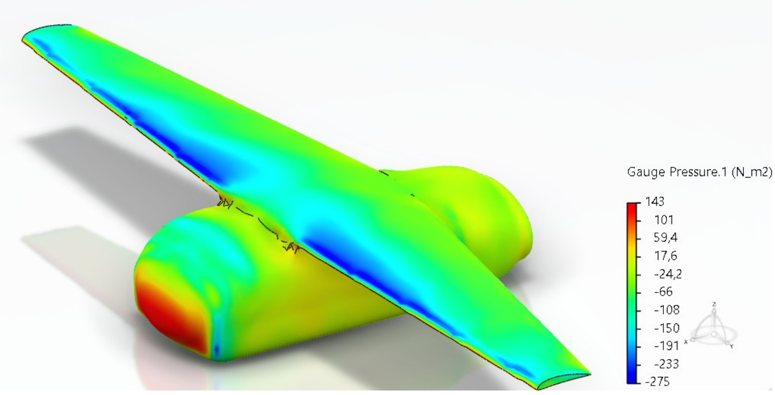


Figure 9.13: Gauge pressure at AoA 12°

9.5. Recommendations

To enhance future work on the aerodynamic analysis, several improvements can be considered. One important step would be to run CFD simulations with multiple airfoil profiles, as the interaction between the wing and fuselage might lead to the identification of a more optimal airfoil shape. Additionally, the CFD mesh should be revisited and refined; due to time constraints, a relatively coarse mesh was used in the current simulations, which may have limited the accuracy of the results. Running the CAD model in an alternative CFD software is also recommended to perform a sensitivity analysis. Comparing results from different solvers would help validate the current model and increase confidence in the simulation outcomes.

Furthermore, propellers should be included in the aerodynamic analysis, with particular attention given to their interaction with the wing. This step is essential to validate the assumption presented in Section AS-AERO-01, regarding the exclusion of vertical propeller effects.

Finally, experimental validation through wind tunnel testing is strongly recommended. This would provide high-fidelity data to verify the accuracy of the CFD results and strengthen the overall reliability of the aerodynamic model.

[3] `http://airfoiltools.com/airfoil/details?airfoil=naca6412-il` [cited 17 June 2025]

10 Stability & Control

This chapter analyses the static and dynamic stability and controllability for the drone during both horizontal and vertical flight, and sizing stability surfaces when needed. The stability & controllability chapter starts with the design approach and the assumptions made in Section 10.1, then discusses the model and V&V for vertical stability and controllability in Section 10.2. Next, the horizontal stability & controllability of the drone is explored in Section 10.3, and finally recommendations for further research on the design are offered in Section 10.4.

10.1. Design Approach & Assumptions

In order to analyse the stability and controllability of the drone, the department first divided the work into two phases: vertical and horizontal flight. For the vertical flight, there are many models for multi-copter drones, but very few for VTOL drones. Thus, the department decided to create a simplified model for vertical flight, with the assumptions presented below. The main inspiration for this model has been taken from the work of Andrew Gibiansky, who created a stability model for the quadcopter[1].

For horizontal flight stability, a quick analysis has been done to check if there is any need for tails or control surfaces. Taking into account the variable number and placement of the pizza payload and batteries during the mission, it would be almost impossible to design a stable drone in horizontal flight without tails. Even using vertical thrusters was initially considered to keep the drone statically stable, but the concept was removed due to constant power consumption, which is less efficient and sustainable, while a tail is much more efficient by using the drone's speed to create stabilising lift. The sizing of the tails has been done using the model presented during the TU Delft course AE3211-I[2].

As for the control surfaces, it has been confirmed that the vertical thrusters can be used for manoeuvres, considering that the manoeuvres take a few seconds, thus the energy consumption is minimal. Not having control surfaces also means reduced mass and material consumption when producing the drone, resulting in a more sustainable design.

A list of the assumptions made during the work of the stability and controllability department is presented below:

- **AS-SC-01:** The action point of the drag force of the drone during vertical movement is at the geometric centre of the drone.
- **AS-SC-02:** The centre of gravity of the drone will be symmetric with respect to the longitudinal axis of the drone.
- **AS-SC-03:** The drone will only move up, down, and rotate around the vertical axis during vertical flight.
- **AS-SC-04:** For symmetry simplifications, the pairs of front and back vertical propellers will be at the same distance from both the longitudinal and lateral axes.
- **AS-SC-05:** Due to low angular velocities expected during yaw rotation, rotational drag that is active during the rotation around the vertical axis can be considered very small compared to

[1] URL `https://andrew.gibiansky.com/blog/physics/quadcopter-dynamics/` [cited 2 May 2025]

[2] TU Delft, *Systems Engineering & Aerospace Design (AE3211-I)*, 2025

the other moments.

- **AS-SC-06:** The vertical propellers can only rotate one way in order to produce lifting thrust, which means reverse thrust is not considered as an option.
- **AS-SC-07:** External forces that interact with the longitudinal-lateral plane during vertical flight will be assumed to always act in the direction parallel to the longitude of the drone, with the idea being that the drone will always create a yaw movement in order to position the drone's horizontal propeller directly in front of the external force.
- **AS-SC-08:** The model assumes that the thrust and torque produced by the vertical propellers are proportional to the squared angular velocity.
- **AS-SC-09:** Using engineering judgment, it is chosen that the vertical propellers that are opposite in both lateral and longitudinal directions have the same angular velocity direction.
- **AS-SC-10:** The model assumes that only one vertical propeller will fail at one time, considering the probability of two vertical propellers failing at the same time is way too small to be even considered.
- **AS-SC-11:** The model assumes that wind gusts that affect stability and controllability have a velocity of up to 7 m/s.
- **AS-SC-12:** The model assumes that the wing will fold enough that the effects of the wing during vertical flight are negligible.

10.2. Vertical Stability & Controllability

This section covers the analysis done for vertical stability & controllability in Subsection 10.2.1, and then discusses the verification and validation done for this analysis in Subsection 10.2.2.

10.2.1. Analysis

The analysis of vertical flight stability and controllability comprises drawing the free body and kinematic diagrams, defining the forces and moments equilibrium, transforming the equations into a linear system, designing for propeller failure, and explaining how the model is used.

Free Body and Kinematic Diagrams

In order to analyse the stability and controllability of the drone during vertical flight, the first step is to consider the drone as a rigid body with a variety of forces and moments acting on it. In some cases, the total sum of the forces and moments must be zero for the axes of motion where equilibrium is expected, while in other cases, the total sum must be different from zero to simulate motion by having translational and rotational accelerations. A free body and kinematic diagrams are shown in Figure 10.1 to showcase the expected forces that interact with the drone and the accelerations that form the movements of the drone.

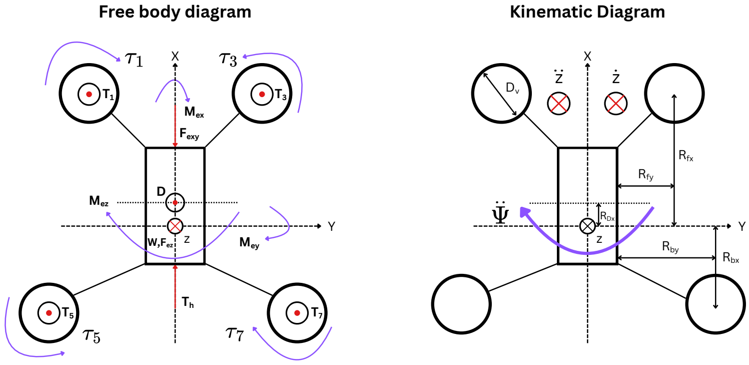


Figure 10.1: Free body and kinematic diagram for vertical flight analysis

Forces and Moments Equilibrium

All the assumptions presented for vertical flight are used in the free-body and kinematic diagrams. With this in mind, some conventions will be chosen for this analysis. First, the angular motions around the X, Y and Z axes are called roll, pitch, and yaw movements, and their positive movement is defined by the orientation of the three axes of the system. Secondly, the angular velocity of the vertical propellers is considered to be positive in the clockwise direction. Now, before the equilibrium analysis starts, it's important to define all the forces expected in the analysis. Some forces such as gravity remain constant, while some external forces and moments are given by the sensors monitoring the drone's environment and it's own movement, like $F_{e_{xy}}$, F_{e_z}, M_{e_x}, M_{e_y}, and M_{e_z}. The drag of the drone during vertical flight will depend on the velocity of the drone squared, while the thrust and torque produced by the propellers will be proportional to the angular velocity of the propeller squared. An overview of the more complex force dependencies is shown in Equation 10.1.

$$\begin{aligned} T_i &= K_t \cdot \omega_i^2 \qquad \text{where} \quad K_t = \frac{C_t \cdot \rho \cdot D^4}{4 \cdot \pi^2} \\ \tau_i &= K_\tau \cdot \omega_i^2 \qquad \text{where} \quad K_\tau = \frac{C_q \cdot \rho \cdot D^5}{4 \cdot \pi^2} \\ D &= K_d \cdot \dot{z}^2 \qquad \text{where} \quad K_d = 0.5 \cdot C_d \cdot \rho \cdot S_{xy} \end{aligned} \tag{10.1}$$

From the following equations, C_t is the propeller efficiency constant, which showcases the efficiency at which electrical power is transformed into mechanical thrust, C_q is the aerodynamic drag for blade rotation constant, which refers to the amount of rotational drag the blades have to overcome to keep rotating, C_d is the body drag coefficient of the drone, and D is the diameter of the propeller. With all the forces defined, the equilibrium and motions on all 6 axes of motion can be described in Equation 10.2:

$$\begin{aligned} \sum F_x &: \quad F_{e_{xy}} = K_{t_h} \cdot \omega_h^2 \\ \sum F_z &: \quad m \cdot \ddot{z} = m \cdot g + F_{e_z} - K_d \cdot \dot{z} \cdot |\dot{z}| - T_1 - T_3 - T_5 - T_7 \\ \sum M_x &: \quad 0 = M_{e_x} + T_1 \cdot r_{fy} - T_3 \cdot r_{fy} + T_5 \cdot r_{by} - T_7 \cdot r_{by} \\ \sum M_y &: \quad 0 = M_{e_y} + K_d \cdot \dot{z} \cdot |\dot{z}| \cdot r_{Dx} + T_1 \cdot r_{fx} + T_3 \cdot r_{fx} - T_5 \cdot r_{bx} - T_7 \cdot r_{bx} \\ \sum M_z &: \quad I_{zz} \cdot \ddot{\Psi} = \tau_1 - \tau_3 - \tau_5 + \tau_7 + M_{e_z} \end{aligned} \tag{10.2}$$

The force equilibrium around the Y axis is missing due to the assumption AS-SC-07, so there are no forces in the Y axis. The torque used in Equation 10.2 in the Z direction comes from the differential torque produced by the propellers. This phenomenon is happening since, initially, the drone's angular momentum is zero before the propellers start to run. After the propellers are active and their blades have non-zero angular momentum, the drone's whole angular momentum must remain zero; thus, the drone shall start having a yaw rotation movement. That is why AS-SC-04 is so important; having a pair of propellers rotating in the same direction means we can control the differential torque produced between the two pairs while still being stable in all other axes of motion.

Transformation to the Linear System

This set of equations can be rearranged so that it is easier to find the angular velocities required for stability and controllability. By replacing the thrust and torque of the propellers with the formulas in Equation 10.1, and rearranging the formulas in Equation 10.2, we obtain the system of equations in Equation 10.3.

$$\begin{aligned}
&\omega_1^2+\omega_3^2+\omega_5^2+\omega_7^2=\frac{m\cdot g+F_{e_z}-K_d\cdot\dot{z}\cdot|\dot{z}|-m\cdot\ddot{z}}{K_t}=A\\
&\omega_1^2-\omega_3^2-\omega_5^2+\omega_7^2=\frac{I_{zz}\cdot\ddot{\Psi}-M_{e_z}}{K_\tau}=B\\
&\omega_1^2r_{fx}+\omega_3^2r_{fx}-\omega_5^2r_{bx}-\omega_7^2r_{bx}=-\frac{M_{e_y}+K_d\cdot\dot{z}\cdot|\dot{z}|\cdot r_{Dx}}{K_t}=C\\
&\omega_1^2r_{fy}-\omega_3^2r_{fy}+\omega_5^2r_{by}-\omega_7^2r_{by}=-\frac{M_{e_x}}{K_t}=D\\
&\omega_h^2=\frac{F_{e_{xy}}}{K_{t_h}}=E
\end{aligned}\tag{10.3}$$

In order to determine the constants A, B, C, and D at any moment of motion, the inputs must be defined. Some of the inputs come from the design of the drone, like $m, K_d, K_t, I_{zz}, K_\tau, r_{Dx}, K_{t_h}$, while others reflect the motion itself, like the external forces $F_{e_{xy}}$,F_{e_z}, M_{e_x},M_{e_y}, velocities and accelerations $\ddot{z}, \dot{z}, \ddot{\Psi}$. The motion inputs themselves are determined by the sensors inside the drone, checking between the expected velocity and acceleration and the actual values, and then defining an external force as the culprit of the equilibrium disturbance. The most common external forces would come from wind gusts, which, from AS-SC-11, it should be expected to have the velocity up to 7 m/s.

Now, removing the last formula from Equation 10.3 due to it being decoupled from the main linear system, as it doesn't depend on all other variables, we can put the other four equations into a linear system, as shown in Equation 10.4.

$$\begin{bmatrix}1&1&1&1\\1&-1&-1&1\\r_{fx}&r_{fx}&-r_{bx}&-r_{bx}\\r_{fy}&-r_{fy}&r_{by}&-r_{by}\end{bmatrix}\begin{bmatrix}\omega_1^2\\\omega_3^2\\\omega_5^2\\\omega_7^2\end{bmatrix}=\begin{bmatrix}A\\B\\C\\D\end{bmatrix}\tag{10.4}$$

Propeller Failure

Solving this system enables all angular velocities to be found by finding the inverse of the matrix and multiplying it by the vector on the right side of the equation. This way, we find all the required angular velocities of the propellers in order to be stable and controllable with all propellers functional.

In case one of the propellers is not working anymore, in order to make the system determinable again, the axis motion would need to become unstable. For the safest emergency landing possible, the rotation around the vertical axis, namely the yaw rotation, has been chosen as the safest one to be left unstable, while still controlling the height drop speed. This translates into our system of equations by removing the second row, thus only three equations remain in the system. Now, due to an engine failure, one of the angular velocities will become zero and not be a variable anymore, thus its corresponding column must be eliminated as well. For example, if $w_1 = 0$, then the first column of the first matrix must be removed to make the system solvable again. This would look like Equation 10.5.

$$\begin{bmatrix}1&1&1\\r_{fx}&-r_{bx}&-r_{bx}\\-r_{fy}&r_{by}&-r_{by}\end{bmatrix}\begin{bmatrix}\omega_3^2\\\omega_5^2\\\omega_7^2\end{bmatrix}=\begin{bmatrix}A\\C\\D\end{bmatrix}\tag{10.5}$$

While it's important for safety reasons to design for propeller failure, it is also a great way to make the drone more sustainable. By reducing the drone's damage from a crash, fewer materials are required to repair or replace the drone with a new one, while also making sure that the non-recyclable parts don't need to be thrown out.

Model Usage

In the end, the model created for vertical stability and controllability can be programmed into the drone, taking as input data from the sensors about the environment and outputting the required angular velocities for each propeller. However, this model can also be used to find the maximum possible angular velocity needed to fly vertically, something that is then used in the propulsion subsystem. For the worst case scenario when one of the front engines fails, the maximum angular velocity comes from the other front propeller and has a value of 5,531 rpm. In this case, the drone descends controllably at a rate of 2 m/s without accelerating, which is safe enough for an emergency landing. This is below the maximum limit of 6,000 rpm given by the propulsion department; thus, the drone is capable of flying vertically without issues.

10.2.2. Verification & Validation

Considering that the model used to simulate vertical stability and controllability was created during the DSE, it is very important to make sure that the model and the software created for it work as intended and reflect real-life scenarios as accurately as possible without any errors.

Verification

For the verification process, a number of unit and system tests have been created. The first unit test will check if all the constants in Equation 10.1 are calculated properly in the software. This is done by comparing the code output with hand calculations done by the department. Checking if the software is solving correctly Equation 10.4 and checking that the inverse matrix and multiplication processes are done in the second unit test. This is done by checking the software output with other reliable inverse matrix calculators and by hand calculations. The third unit test will check if all the variables used in the code's calculations are defined in the expected unit by verifying how all variable inputs are defined, and all parts of the code that deal with unit conversion. The system test will check the output of the whole software for different cases, such as per-engine failure, different velocities, and acceleration scenarios, and check if the magnitude of the angular acceleration is realistic. The test also looks for any negative angular velocity, as the software may output such a value, but the mathematical model does not allow for any negative angular velocities.

Validation

For the validation process, the most optimal way to do it would be to have a prototype and test in real-life scenarios the stability and controllability of the drone. However, at this time in the DSE, this is not a feasible validation method; thus, other ways must be explored.

A valid method of validating the whole model would be to compare the outputs of the software with results from real-life experiments done for different drones' stability. These would be found in a variety of literature studies. While the models may vary with their assumptions and level of accuracy, and the drone designs would vary, of course, checking the order of magnitude of the values would prove enough to validate the model at this stage of the DSE. This can be done in the future stages of the design.

10.3. Horizontal Stability & Controllability

This section covers the analysis done for horizontal stability & controllability in Subsection 10.3.1, and then discusses the verification and validation done for this analysis in Subsection 10.3.2.

10.3.1. Analysis

In order to size the tails required for the stability and controllability of the drone during horizontal flight, the model proposed in the TU Delft course AE3211-I can be used[3]. First, the horizontal tail

[3] TU Delft, *Systems Engineering & Aerospace Design (AE3211-I)*, 2025

is sized, then the vertical tail is sized, then the tail is converted to a V-tail configuration, and the presentation of the final tail sizing is done.

Horizontal Tail Sizing

The first step was to analyse is to create the scissor plot, which can be seen in Equation 10.6, designed for an MTOW of 10 kg. This plot contains two lines, one for stability (on the right) and one for controllability (on the left), using Equation 10.6. These equations were derived by analysing the longitudinal rotational equilibrium of the drone, having as moments the one created by the wing and horizontal tail. More details can be found in the referenced course.

$$\begin{aligned} \text{Stability:} \quad & \overline{x_{cg}} = \overline{x_{ac}} + \frac{C_{L\alpha_h}}{C_{L\alpha_{A-h}}}\left(1 - \frac{d\epsilon}{d\alpha}\right)\frac{S_h l_h}{S\,\bar{c}}\left(\frac{V_h}{V}\right)^2 - \text{S.M.} \\ \text{Controllability:} \quad & \overline{x_{cg}} = \overline{x_{ac}} - \frac{C_{m_{ac}}}{C_{L_{A-h}}} + \frac{C_{L_h}}{C_{L_{A-h}}}\frac{S_h l_h}{S\,\bar{c}}\left(\frac{V_h}{V}\right)^2 \end{aligned} \tag{10.6}$$

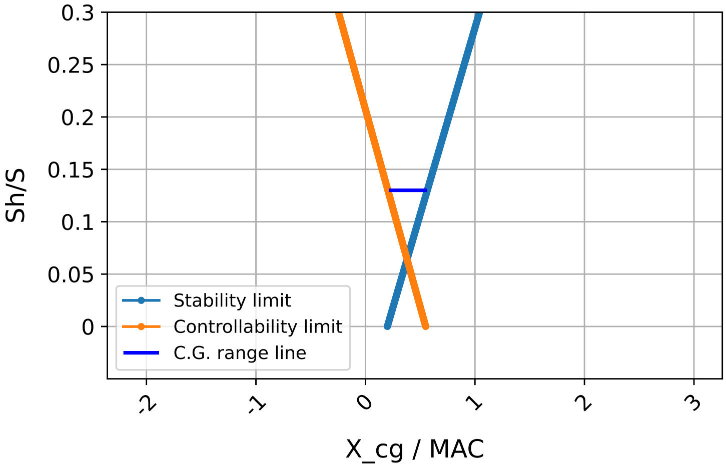


Figure 10.2: Scissor plot for the tail sizing

Looking at the scissor plot, the ratio between the required horizontal tail area and the wing area can be obtained by having the centre of gravity position of the drone be inside the region defined by the two lines. However, the C.G. position will not always remain the same, considering that the drone has a variable number of pizzas. These variables make the C.G. vary per delivery; thus, it is more practical to talk about the maximum C.G. range of the drone for all possible scenarios. In order to find this range, a loading diagram was constructed. To find the maximum shifts of the C.G. along the fuselage, a diagram was constructed so that the weights are added inside the drone from the nose towards the back of the drone. The resulting diagram can be seen in Figure 10.3.

With the C.G. range determined, the area ratio can be determined. While this can be done, it does not mean that the fit will be perfect in the scissor plot. Even though the lengths of the C.G. range are low, it can still go beyond the safe region of the two lines in the scissor plot, thus making the design unstable. Both the C.G. range length and the end points of the range are important in finding the perfect fit, which results in finding the lowest necessary tail area. In order to influence the end points to our advantage, the best method is to change the position of the wing. This creates a shift in the C.G. by moving the wing weight, but more importantly, it changes the location of the leading edge mean aerodynamic chord (LEMAC), which is used as a reference point for the stability and controllability analysis model. The change in the endpoints and lengths of the C.G. range relative to the position of the wing can be seen in Figure 10.4.

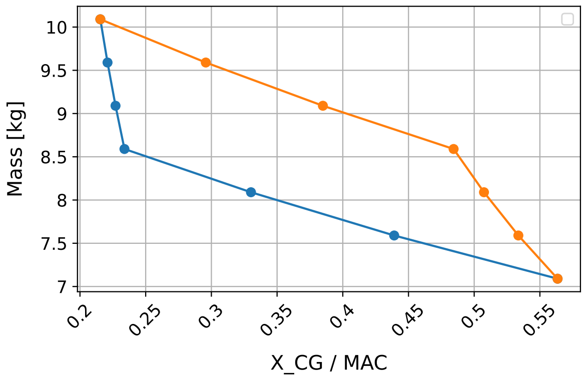


Figure 10.3: Loading diagram for the drone

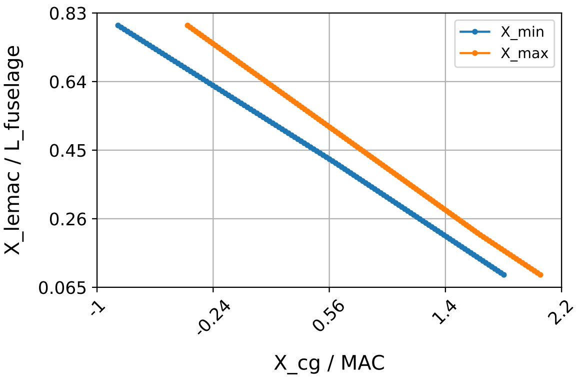


Figure 10.4: C.G. variation due to wing position

Using all this, the best wing position that gives the best C.G. range that results in the lowest horizontal tail area needed for stability and controllability was found. Then the sizing of the horizontal tail was achieved.

Vertical Tail Sizing

For lateral stability, a vertical tail sizing is required. The vertical tail is needed for static stability and controllability for lateral movements, such as yaw rotations. The lateral force needed from the vertical tail can be modelled by considering the drone does a yaw movement around the vertical axis with a constant sideslip angle β.

$$\begin{aligned} M_{tail} &= \frac{\pi b^2 \beta \rho v^2 (L_{fuselage} - x_{cg})}{4}, \quad M_{air} = \left(x_{cg} - \frac{L_{fuselage}}{2} \right) \frac{\rho v^2 \beta S_{lat}}{2}, \\ S_{vertical} &= \frac{(2x_{cg} - L_{fuselage}) S_{lat}}{AR\pi (L_{fuselage} - x_{cg})} \end{aligned} \tag{10.7}$$

The moment generated by the vertical tail and the air force acting on the aircraft due to the sideslip angle is shown in Equation 10.7. Using both equations, the area required for the vertical tail was obtained.

Tail Conversion to the V-tail Configuration

Now that the required horizontal and vertical tail areas have been determined, the ideal tail configuration for the drone was chosen. Due to the thicker design of the drone, a standard configuration of horizontal tail at the base of the vertical tail was not possible due to the airflow did not circulate properly at the base of the vertical tail. Thus, a V-tail configuration has been decided on, which uses a NACA0012 airfoil, as recommended by the aerodynamics department. In order to convert the vertical and horizontal tail areas into a V-tail configuration, Equation 10.8 can be used. The α_{tail} is defined as the angle that the V-tail is making with respect to the XY plane of the drone.

$$S = \frac{\sqrt{S_v^2 + S_h^2}}{2}, \quad \alpha_{\text{tail}} = \arctan\left(\frac{S_v}{S_h}\right) \tag{10.8}$$

Then, using the aspect and taper ratio chosen for the tail configuration, the wingspan and chords of the V-tail configuration can be determined in order to have the full geometry of the tail.

Final Tail Sizing

Finally, after a few iterations, the final airfoil, wing position, and V-tail sizing were concluded. The results gave a wing positioned at 0.92 m, and a tail with a NACA0012 airfoil, root chord of 0.271 m, tip chord of 0.136 m, span of 0.407 m, and a V-tail angle of 31.21°.

10.3.2. Verification & Validation

The model used to size the V-tail must be verified and validated to make sure that the stability and controllability aspects of the drone are preserved.

Verification

For the verification process, a number of unit and system tests have been created for horizontal stability and controllability. The first unit test will check the data conversion of inputs given by the other department's software that is used in this code. This is done by checking for each imported value if the import is correctly defined, up-to-date, and that the units used are SI. The tail geometry sizing part of the code will be checked by the second unit test. This is done by comparing the outputs with hand solutions for the tail sizing. The third unit test will check the code for the centre of gravity range of the drone by checking the outputs with hand-calculated values, and also look at the graphical output and do a sanity check on its accuracy with what is expected. The code for the wing position variation of the drone is verified by the fourth unit test by checking the outputs with hand-calculated values, and also looking at the graphical output and doing a sanity check on its accuracy with what is expected. The final unit test will check the code for the scissor plot of the drone. This is done by checking the outputs with hand-calculated values and also looking at the graphical output and doing a sanity check on its accuracy with what is expected, one example being checking if the centre of gravity range fits the two stability and controllability lines at the expected area ratio. The system test will check the output of the whole software, making sure that all code parts checked in unit tests are connected properly and give the expected output.

Validation

For validation, the most accurate way to do it would be to use a prototype drone using the team's design to test the horizontal stability and controllability. However, at this time in the DSE, this is not an employable validation method. Regardless, it must be noted that the model used to determine horizontal stability has been verified and validated by the teachers at TU Delft, the source of this model, and considering this fact, a level of reliability can be accorded to the model reflecting real-life scenarios.

10.4. Recommendations

While a lot of work has been done in the stability and controllability department, there are still some recommendations that can be made for future research.

First of all, while vertical thrusters are replacing control surfaces when manoeuvres are executed, a proper analysis of how the vertical propellers function for each manoeuvre should be done. A good starting point for this analysis would be from the vertical stability model, with the wing adding lift and drag to the model. Also, some of the axes that were previously only in equilibrium would now be allowed to move around the respective axes, to simulate the manoeuvres.

Secondly, a detailed analysis of the transition phase in regards to stability and controllability could be made. A great starting point for this analysis would be merging both the horizontal and vertical flight stability models and verifying the stability of the path chosen by the performance department.

Last but not least, it is highly recommended that a more throughout validation of the used models is done, ideally by testing the stability and controllability of a prototype drone created from the team's design, thus making sure that the models used reflect real-life scenarios that the drone will encounter while delivering pizza to the customers.

While the mentioned recommendations are left for future research due to time constraints during the DSE, these recommendations are the next step in guaranteeing that the designed drone is stable enough to be considered safe by the government and customers.

11 Performance

This chapter presents the performance analysis of the BezORger drone, which plays a central role in shaping and evaluating the design throughout the project. While performance is not a design output in itself, it is a critical tool for evaluating the drone's performance and verifying whether the drone's configuration meets mission requirements, most notably, its ability to deliver multiple 2.5 kg pizza orders (RQ-01-04-07) across an urban environment with a typical range of 6 km (RQ-01-04-02, RQ-01-04-03).

The chapter is structured as follows: Section 11.1 presents the overall modelling assumptions, followed by a description of the mission overview with flight profile and phases in Section 11.2. Next, the energy modelling across each flight phase is presented in Section 11.3, followed by the analysis of the flight envelope and performance limits in Section 11.4. Finally, recommendations for future work are made in Section 11.5.

11.1. Assumptions

To keep the scope manageable and focus the analysis on the mission's most impactful aspects, several modelling assumptions were introduced. These simplifications allowed for analytical clarity while maintaining a conservative bias in energy estimates and performance margins. The assumptions are summarised below and referred to in subsequent modelling sections.

- **AS-PERF-01:** Atmospheric variables such as air density and pressure are assumed constant throughout the flight.
  This is justified by the low cruise altitude of 60 m*, where deviations from standard ISA conditions are below 1%, making the impact on performance negligible.*
- **AS-PERF-02:** Time estimates for vertical segments like take-off and landing are obtained by modelling at constant velocity.
  The acceleration phases are short and contribute minimally to total energy; assuming steady velocity leads to a conservative estimate.
- **AS-PERF-03:** The energy used in the transition from cruise to landing is assumed equal to the take-off to cruise transition.
  This symmetry simplifies analysis and is conservative by overestimating energy needs for descent.
- **AS-PERF-04:** The final stabilisation phase after transition is assumed to have negligible energy impact.
  This phase typically involves reducing thrust, which decreases rather than adds to energy use, making this a conservative assumption.

11.2. Mission Overview

A mission overview of DronebezORgd is needed for the design of drone its operation. The analysis of the mission overview has been presented in Subsection 11.2.1, followed by verification of the analysis in Subsection 11.2.2.

11.2.1. Analysis

The mission overview consists of the mission profile, which explains an overview of legs flown by the BezORger during a standard mission, and the flight phases that are considered in each leg.

Mission Profile

A mission profile is needed for the design of the drone. This is determined by the business model (Section 3.4), where DronebezORgd is decided to be a pickup-delivery service with multi-delivery capability. This means that the drone flies from the depot to the restaurant and delivers its customers the clustered orders, and returns to the depot upon completion of the trip. Some terms are now defined and will be used for the rest of the report:

- **Abbreviations:** C – customer, D – depot, and R – restaurant
- **Leg:** A leg refers to any flight from A to B, e.g., ‘DR’ corresponds to a leg from depot to restaurant. This includes a standard mission profile of take-off, climb, cruise, loiter, descent, and landing. The loiter time and range vary between different legs, while the cruise altitude is assumed to be 60 m.
- **Pickup:** A pickup refers to any flight to a restaurant, which could, for example, be ‘DR’ or ‘CR’.
- **Delivery:** A delivery refers to any flight to a customer, which could, for example, be ‘RC’ or ‘CC’.
- **Trip:** A trip refers to the entire sequence of flights performed by the drone from departure from the depot to arrival at the depot. An example would be ’DRCCD’, which refers to the sequence for depot-restaurant-customer-customer-depot.

To ensure that the drone remains flexible to complete a wide range of orders within a Delft-like city, both single and multi-delivery cases are considered for the mission profile. The design has to conform to both flight profiles, where it was found that the multi-delivery case is generally more limiting.

The worst-case single-delivery mission considered is for a Delft-like city of 6 km radius. In these conditions, for any extreme restaurant and customer position, a fully-charged drone should be able to complete the trip `D-R-C-D`, representing the worst-case scenario of a single-delivery mission. For the defined city, this would give a worst-case mission for an equilateral triangle inscribed in a circle. Each side of the triangle would be given by $\sqrt{3} \cdot R$, giving around 5.2 km for each leg.

In addition to the single-delivery case, a more realistic multi-delivery mission profile was defined to guide the sizing and operations design of the BezORger drone. This profile assumes each drone departs from a depot, makes a sequence of restaurant–customer–customer–restaurant deliveries, and then returns to the depot. A worst-case scenario for multi-delivery was deemed impractical due to the potential for infinite orders. Therefore, a decision was made to design for an average mission, represented as `D-R-C-C-R-C-D`. This sequence was derived from simulations by mission planning in operations (Section 12.2). Clustering two customers per order pickup from the restaurant is also consistent with the chosen payload handling method, which allows for a maximum of two deliveries to be grouped due to the payload opening mechanism (Section 6.2).

To model distances for each leg of this mission, a probabilistic spatial simulation was conducted over a circular city with a 6 km diameter. Restaurant locations were sampled from a Gaussian distribution centred near the city core, while customer locations followed a ring-shaped density profile to represent residential clustering away from the centre. Using this model, 10,000 sample mission routes were generated. Mean values of each route leg were extracted and used to define the baseline multi-delivery profile. More information about the simulation can be found in the midterm report [2].

The parameters for single- and multi-delivery have been summarised in Table 11.1, including average distances, payloads, and loitering times. Loiter was included in this analysis to account for situations where the drone needs to hover, for example, while waiting for clearance for landing.

Table 11.1: Mission profile parameters for single- and multi-delivery cases

Mission Type	Phase	Leg	Distance [km]	Loitering Time [s]	Payload [kg]
Multi	1	DR	1.00	120	0
	2, 5, 8	RC	1.86	120	2.5
	3, 6, 9	CC	1.24	120	2
	4, 7	CR	1.86	120	0
	10	CD	1.58	30	0
Single	1	DR	5.20	120	0
	2	RC	5.20	120	2.5
	3	CD	5.20	30	0

Flight Phase Modelling

Each mission leg of the BezORger drone includes a sequence of distinct flight phases, which were modelled to account for their unique energy, velocity, and altitude characteristics. An overview of these phases is presented in Figure 11.1.

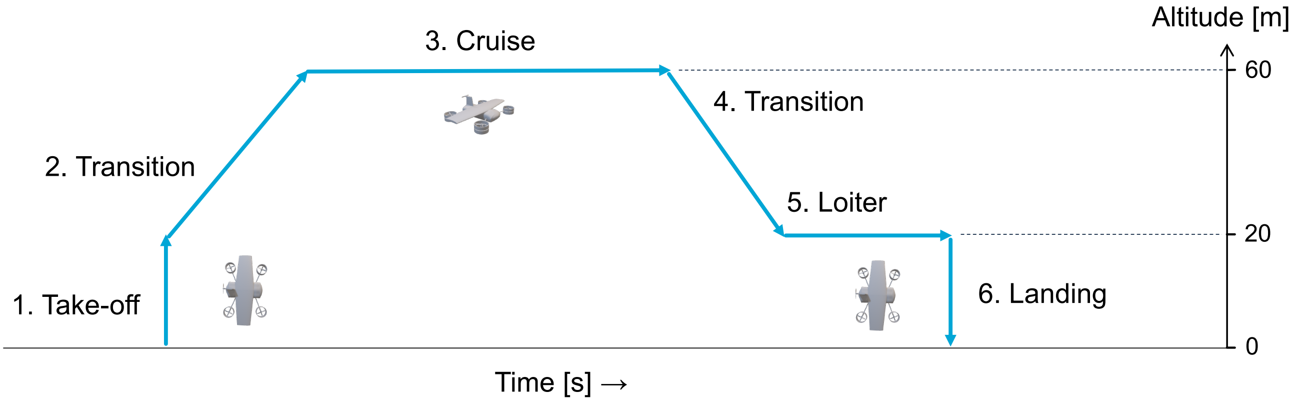


Figure 11.1: Overview of flight phases per leg

To safely clear urban infrastructure and trees while minimising excess energy consumption, a transition altitude of 20 m was selected. This decision was informed by city mapping data for Delft, where most static obstacles were found to be below this height Subsection 12.4.1.

The cruise altitude was chosen as 60 m, balancing several competing factors. On one hand, higher cruise altitude reduces ground-level noise and improves public perception by increasing visual clearance, enhancing social sustainability. On the other hand, it requires more energy during ascent. Most buildings in Delft fall below this height, allowing for simplified path planning and obstacle avoidance.

The drone's flight phase velocities were tuned to reflect the expected performance across different legs depending on restaurant, customer, or depot, with specific values listed in Table 11.2.

Table 11.2: Flight phase and performance parameters

Parameter	Take-off velocity [m/s]	Landing velocity [m/s]	Cruise velocity [m/s]
Depot	3	4	14
Customer	2	2	14
Restaurant	2	2	14

11.2.2. Verification & Validation

The verification of the mission overview primarily revolved around checking the shape of city distributions used to obtain the distances for the different legs of the mission profile, e.g., restaurant-customer

(RC) distance.

This was done by increasing testing for convergence of the distances obtained for an increasing number of simulations. Sensitivity analysis was also performed to verify whether the results made sense in relation to the changes made. For example, the depot location was offset from the centre, and the resulting depot-restaurant and depot-customer distance was found to increase as expected. Due to the nature of the simulation performed a normal distribution would be expected if repeated a sufficient number of times. The resulting distance distribution was found to tend towards a normal distribution as the number of simulations was increased.

The results were validated by comparing them to the distances for real restaurant and customer locations for Delft from operations simulations Subsection 12.4.1.

11.3. Mission Energy Modelling

The mission energy model is needed for the sizing of the battery and power system, and plays a key role in the design iterations of the drone. The mission energy model is explained in Subsection 11.3.1, followed by verification of the model in Subsection 11.3.2.

11.3.1. Analysis

To accurately estimate the total energy consumption across a typical delivery mission, each flight phase was modelled separately with tailored assumptions for thrust and power. An additional 20% of energy was added on top of the calculated mission energy to account for reserve battery capacity. This is important to consider for operational safety in an urban-like environment, as the reserve battery would allow the drone to fly away from high-populated regions and crash-land in a remote area, in case of power failure from the main battery. The total mission energy was then computed as the sum of energy used during take-off, cruise, landing, loitering, reserve, and transition phases.

Take-Off and Landing

For vertical phases, take-off and landing, a thrust-to-weight (T/W) ratio model was used, assuming a value of $T/W = 2.0$ for low-speed vertical flight based on guidelines from UAV handbooks [25]. Although the exact force balance was also computed based on drone weight and expected thrust vectoring capabilities, the T/W ratio model was used for energy estimation as it provided a more conservative estimate.

Power during these phases was calculated based on the thrust and corresponding power-thrust relation obtained from propeller simulations (see Figure 7.2). Time for ascent and descent was estimated from the vertical flight distance of 20 m for transition and vertical climb/descent speeds from Table 11.2. The required energy could then be estimated using Equation 11.1, where P is the average power during vertical flight and t is the climb or descent duration. These durations were phase-specific: take-off and landing each involved a climb or descent from/to ground level to the defined transition altitude of 20 m.

$$E = P_{avg} \cdot t \tag{11.1}$$

Hover and Loitering

Hovering was modelled for phases where the drone remains stationary in the air, such as while waiting at pickup or drop-off locations for landing clearance. The required thrust equals the drone's weight, corresponding to $T/W = 1$. The power needed for hover was based on the propeller thrust-power relation, and computed as for take-off and landing using Equation 11.1.

Cruise Flight Cruise was modelled as steady, level horizontal flight at constant velocity. Energy consumption was estimated using standard propeller aircraft relations, adapted for the drone's electric propulsion system:

$$E_{\text{cruise}} = \frac{W \cdot R}{\eta_{\text{elec}} \cdot \eta_{\text{prop}} \cdot \left(\frac{L}{D}\right)_{\text{cruise}} \cdot \frac{1}{g}} \tag{11.2}$$

where W is the weight, R is the segment range, η_{elec} and η_{prop} are the electrical and propulsive efficiencies, and $(L/D)_{\text{cruise}}$ is the lift-to-drag ratio during cruise. This method enabled realistic estimation of energy used across multiple horizontal segments in both single- and multi-delivery missions.

Transition (Vertical to Horizontal Flight)

In the context of an eVTOL like the BezORger, the transition from vertical take-off to forward cruise is a critical performance phase. While a conservative approximation could model the entire ascent using vertical flight conditions during take-off, a more nuanced energy estimation was desired for this DSE. Importantly, the transition is not only a significant energy-consuming phase but also a key regime where the drone shifts from thrust-supported lift to wing-generated lift.

The goal of the modelling was to estimate the energy consumption during this phase and ensure a feasible trajectory under realistic power, aerodynamic, and safety constraints. Although an energy-optimal transition could, in principle, be formulated as an optimisation problem over thrust inputs and trajectory shape, this would require a detailed dynamical model of the drone and is beyond the scope of this study.

Instead, a simplified yet insightful simulation approach was developed. A fixed time window of 20 s was allocated for the transition, during which the thrust from the vertical propulsion system was gradually reduced, and the horizontal propulsion system was ramped up. The vertical and horizontal thrust profiles were defined as piecewise linear functions with six nodes each, allowing for flexibility while keeping the problem tractable.

The simulation was conducted using explicit Euler integration, with the drone's state defined by horizontal and vertical velocities (v_x, v_z), position (x, z), and pitch angle (θ). The aerodynamic forces were computed using a linear lift and quadratic drag polar model from the aerodynamic design.

The translational equations of motion expressed in Equation 11.3 and 11.4 were then solved for each time step. Here, $\gamma = \arctan(v_z/v_x)$ is the flight path angle, and T_v, T_h are the vertical and horizontal thrust inputs. The accelerations were integrated in time to obtain velocities and positions. Total energy was calculated by summing the power drawn by both systems, using the thrust-power relation for the vertical propellers (see Figure 7.2) and Equation 11.5 for the horizontal component, where η is the total powertrain efficiency, calculated as the product of the motor and propeller efficiency. The total energy for the transition was then found by integrating the total power over the transition time, following Equation 11.6.

$$F_x = -T_v \sin\theta + T_h \cos\theta - D\cos\gamma - L\sin\gamma \tag{11.3}$$

$$F_z = T_h \sin\theta + T_v \cos\theta - mg + L\cos\gamma - D\sin\gamma \tag{11.4}$$

$$P_{horizontal} = \frac{T_h \cdot |v_x|}{\eta} \tag{11.5}$$

$$E_{\text{total}} = \int (P_{\text{vertical}} + P_{\text{horizontal}}) dt \tag{11.6}$$

To qualify as a valid transition, the simulated trajectory was required to meet the following constraints:

- Reach target cruise altitude of 60 m from a starting height of 20 m.
- Avoid negative vertical velocity during the initial climb (safety-critical).

- Maintain pitch angle below 12° to ensure the angle of attack remains below stall conditions.
- Keep the resultant vertical load factor below 1.5g (safety-critical).

The thrust profiles were iteratively tuned to achieve a smooth, continuous trajectory meeting all constraints. A minimum of three control nodes was found insufficient to meet constraints simultaneously, so six nodes were ultimately used to enable more flexible thrust shaping. The thrust profiles used to obtain the transition profile for a drone of MTOW of 10 kg have been shown in Figure 11.3. It was noticed that the shape of the vertical thrust profile strongly resembled a sigmoid function, while the horizontally thrust profile could be approximated by a power law. The transition trajectory obtained for the original piecewise linear thrust profile approximation has been shown together with the transition trajectory obtained for the sigmoid-based vertical thrust and power-law-based horizontal thrust profiles in Figure 11.2.

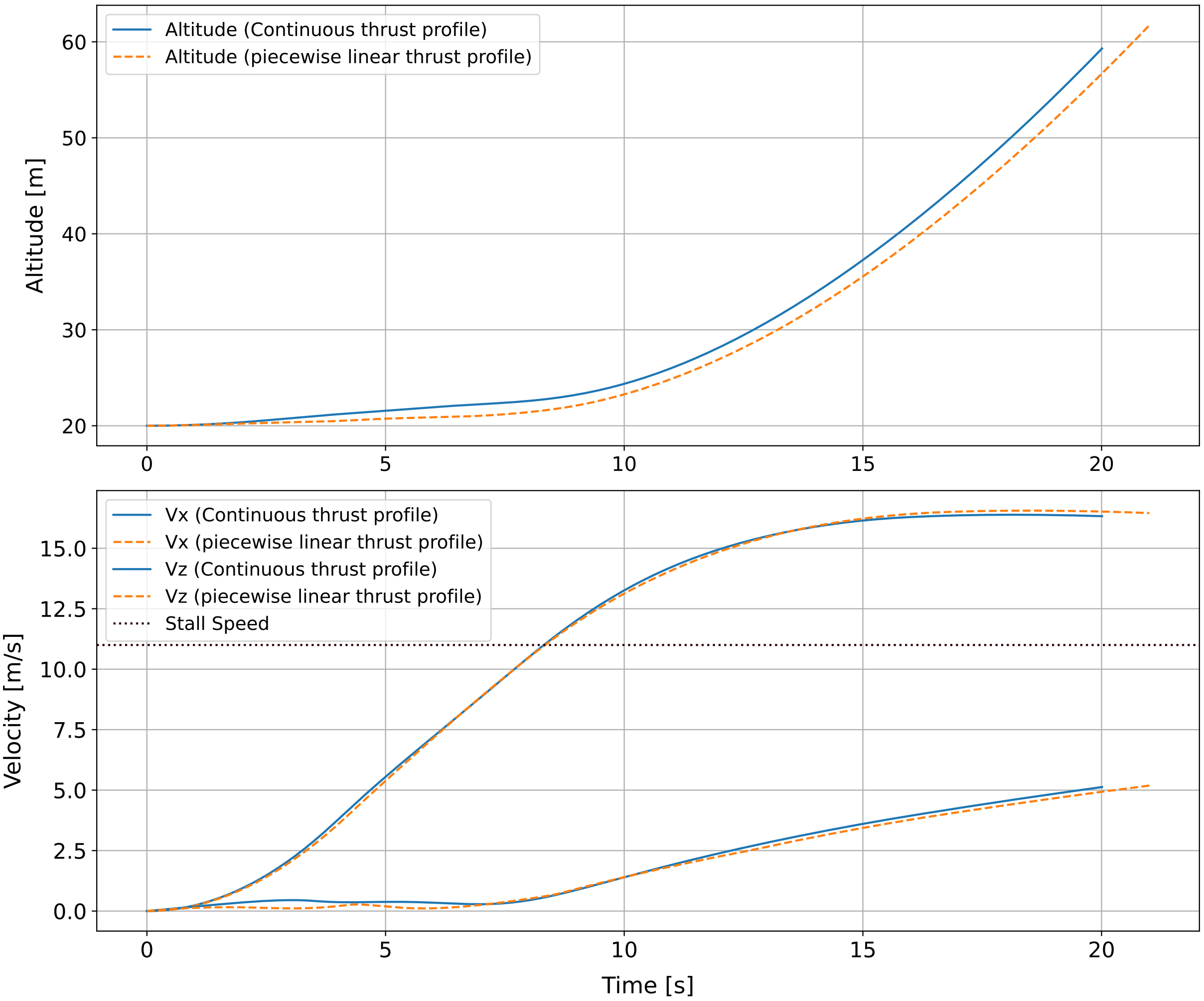


Figure 11.2: Feasible transition trajectory and velocity profile for a drone with MTOW of 10 kg

It can be seen that the continuous curve fits predict transition profiles that are feasible and are remarkably close to those found manually. A particularly interesting outcome is the distinct shapes of the thrust profiles, which warrant further in-depth analysis to fully understand their underlying contributing factors.

There are several limitations to the current transition modelling. Firstly, the stabilisation phase of the transition has not yet been modelled. This is evident as the vertical velocity remains non-zero upon reaching cruise altitude, implying the drone would continue to climb in the current simulation. While this omission means the current energy estimate is conservative, as stabilisation would involve slightly reducing horizontal thrust, it was deemed acceptable for this model, whose primary goal was to identify performance limits and revise energy estimates for the transition phase. Secondly, the optimisation

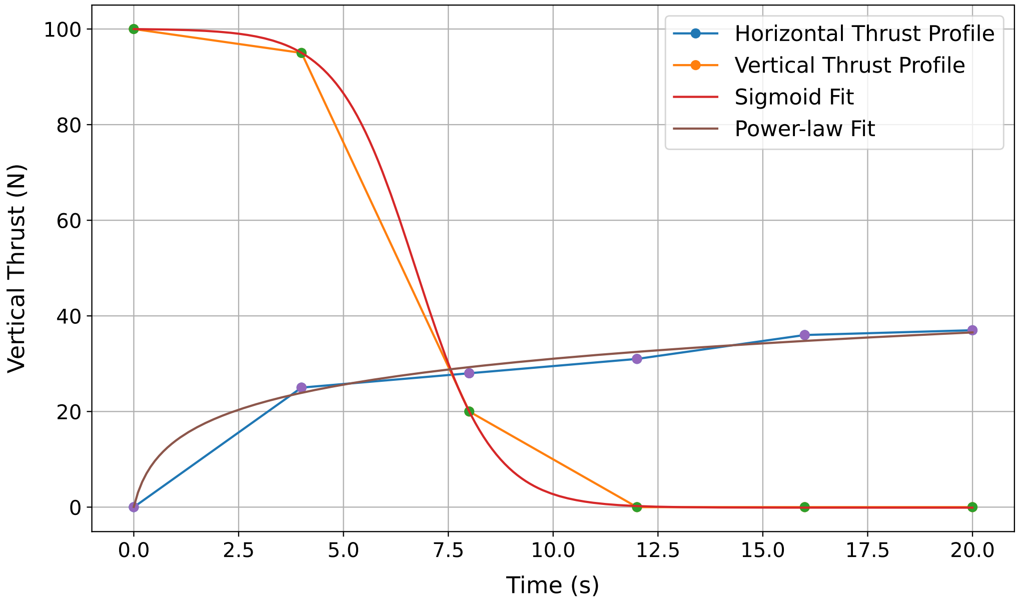


Figure 11.3: Thrust profiles found for feasible transition

was performed using a limited number of nodes. This aspect requires revision to ensure more accurate and robust results. Furthermore, the current framework identifies only a feasible transition trajectory. A significant future extension would be to develop a comprehensive optimisation framework capable of determining the ideal transition for energy minimisation. Moreover, the transition is here modelled between take-off and cruise, while the same energy estimate is assumed for the transition from cruise to landing. This is a conservative assumption as the thrust required during landing would be slightly lower than the thrust required during take-off.

As the transition model required manual iterations, energy estimates were obtained for a variety of MTOWs. The transition energy estimate for any MTOW was then estimated by interpolation, where a linear relationship was found. As an example, using the validated thrust profiles and a transition duration of 20 s, the total energy consumption for transition was calculated as $E_{\text{transition}} = 22{,}380\,\text{J}$.

This value was cross-checked against a sanity baseline: the energy required for a purely vertical climb from 20 m to 60 m. As expected, the transition required slightly less energy, since lift was partially provided by the wings during forward acceleration. This supports the modelling assumption that the chosen trajectory is both realistic and conservative enough for early-stage energy sizing.

11.3.2. Verification & Validation

The first level of verification was done by comparing the energy required for each mission phase. As expected, the energy-intensive phases of take-off and landing were found to be dominant, followed by hovering, which was slightly lower. The transition energy estimate was verified to be slightly lower than the transition as the lift produced by the wing partially relieves the thrust needed from the vertical propellers. The results in the transition regarding the observations on the vertical thrust profile approaching a sigmoid and horizontal thrust profile approaching a power function need further verification by increasing the number of nodes in the linear-piecewise analysis. Sensitivity analysis on the MTOW of the drone was also performed to ensure that the energy estimates of each flight phase varied continuously and exhibited a linear relation with the MTOW.

This can be validated by performing test flights with the drone and monitoring the battery used in each flight phase in the future testing phase of the project, presented in Section 16.2.

11.4. Flight Envelope & Performance Limits

The analysis encompassing the flight envelope and performance limits is presented in Subsection 11.4.1, followed by verification and validation in Subsection 11.4.2.

11.4.1. Analysis

The flight envelope defines the permissible combinations of velocity and load factor under which the drone can safely operate. It captures constraints related to cruise velocity, structural loading, gust tolerance, and manoeuvring capability. This section outlines the derivation of key performance parameters from manoeuvre and gust envelopes that inform both structural design and operational limits. The trade-off between payload and range is also analysed by determining the possible distances the drone is capable of flying with different amounts of payload and battery mass.

Cruise Velocity Determination

A minimum cruise velocity was established to ensure steady-level flight. Using a preliminary MTOW of 12 kg from the Class I weight estimation [2], a wing surface area of $S = 1.3\,\text{m}^2$, and a lift coefficient $C_L = 0.8$ corresponding to a cruise angle of attack of 6°, the required cruise velocity was computed using Equation 11.7.

$$V = \sqrt{\frac{2W}{\rho S C_L}} \tag{11.7}$$

This resulted in a minimum required cruise speed of 13.1 m/s. The cruise velocity for maximum range was also determined, based on the condition of minimum drag (D_{min}). For a propeller-driven aircraft, this occurs for lift and drag coefficients C_L and C_D defined by Equation 11.8 [26].

$$C_{D_{D_{min}}} = 2C_{D_0}, \quad C_{L_{D_{min}}} = \sqrt{\pi A e C_{D_0}} \tag{11.8}$$

Using $C_{D_0} = 0.08$, $A = 5.9$, and $e = 0.8$ from the early aerodynamic design phase, a value of $C_{L_{D_{min}}} = 1.08$ was obtained. However, this lift coefficient was not achievable in cruise, as the selected airfoil limited C_L to approximately 1 for nominal angles of attack below 10°. This showed that the cruise velocity is constrained by available lift rather than by drag minimisation.

The cruise velocity was therefore selected as 14 m/s, to allow for a safe margin from the calculated required cruise velocity of 13.1 m/s and to account for changes during design iterations. The cruise velocity was recomputed at the end of the design iterations to see if the minimum cruise velocity requirement is met.

Gust & Manoeuvring Envelope

To account for external disturbances, a gust envelope was constructed, capturing the additional load factors induced by gusts at three reference velocities:

- V_B: Low-speed, high-lift regime (set as $0.9 \cdot V_C$ remaining above the stall limit)
- V_C: Cruise speed (14 m/s)
- V_D: Dive speed (set to chosen maximum drone speed of $1.5 \cdot V_C = 21\,\text{m/s}$)

The low-speed high lift velocity V_B was chosen close to the stall speed, and the dive speed was set to the maximum flight speed of the drone to encompass critical operational scenarios. This approach is particularly important, anticipating future certification and ensuring operational safety in urban environments, as current CS regulations do not exist for drones.

For each of these speeds, gust velocities U_g were defined, where the magnitude of the gust accounted for reduces as the flight velocity increases, due to decreasing vulnerability of the drone to gusts at increasing flight velocities. The specific gust velocities are usually derived from CS regulations for conventional aircraft, but since regulations do not yet exist for drones gust velocities of 6 m/s for V_B, 5 m/s for V_C, and 3 m/s for V_D were assumed based on forecasted gusts in The Netherlands[1].

The additional load factor due to a gust was calculated using Equation 11.9, where a refers to the lift curve slope.

$$\Delta n = \frac{\rho V S U_g a}{2W} \tag{11.9}$$

The manoeuvring envelope is defined by the stall line (based on $C_{L_{max}}$) and structural load limits. For the BezORger, a positive structural limit of $n = 3.5$ and a negative limit of $n = -1.5$ were adopted based on conventional standards.

Figure 11.4 shows the combined gust and manoeuvre envelope. The maximum load factor of $n_{max} = 3.8$ occurred at cruise velocity V_C, and was multiplied by a safety factor of 1.5 to obtain the ultimate load factor n_{ult}, used for structural design. The valid flight region for the BezORger was therefore defined for $V > V_B$, with the blue region in the diagram indicating manoeuvring loads and the green region showing gust-induced loads.

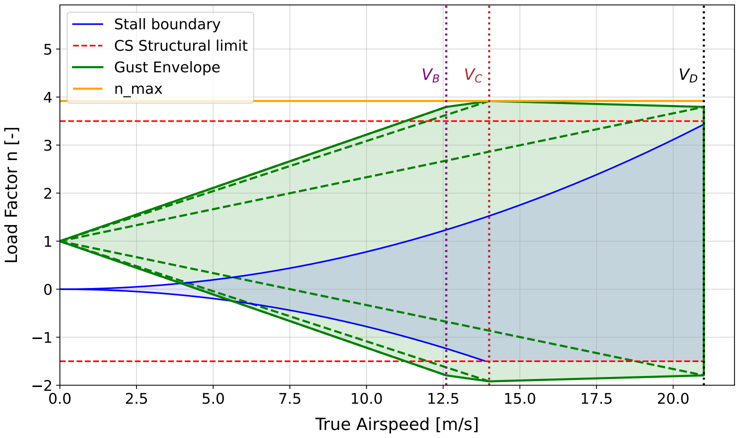


Figure 11.4: Gust & manoeuvre envelope

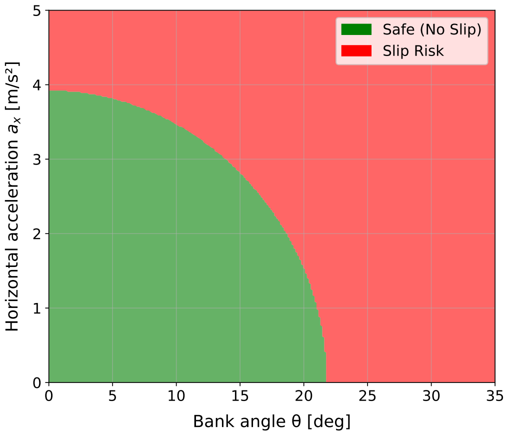


Figure 11.5: Bank angle & horizontal accelerations limits for no-slip of pizza

Operational Limitations due to Payload Sensitivity

Beyond designing the drone structure for specific load factors, it is also desirable for the drone to operate under nominal conditions that minimise high load factors to prevent damage to the payload. For a pizza-delivery drone, several ways of damage can occur, including sudden accelerations, folding, or crushing the pizza against the box edges.

For an initial analysis, operational limits on the load factor arising from horizontal accelerations and banked turns were considered. A key constraint is that the pizza's bottom surface must not slip against the pizza box during flight. The forces acting on the pizza include its weight, the normal reaction force perpendicular to the pizza box surface, and inertial forces from banked turns and horizontal acceleration, which are counteracted by the friction force.

To safeguard the pizza, the drone's flight path should be controlled to prevent slipping and subsequent impact with the pizza box walls, which could lead to folding. A no-slip condition was established by equating the maximum friction force f_{max} to the drone's tangential acceleration with respect to the pizza box, as shown in Equation 11.10. The static friction coefficient μ_s was estimated as 0.4 for a cardboard-to-cardboard interface [27]. By decomposing the forces along the global horizontal direction, the no-slip condition was reformulated as shown in Equation 11.11.

[1] `https://windy.app/forecast2/spot/3681/Amsterdam+Netherlands` [cited 12 June 2025]

$$ma_t \leq \mu_s mg = f_{max} \tag{11.10}$$

$$\sqrt{a_x^2 + (gsin\theta)^2} \leq \mu gcos\theta \tag{11.11}$$

The flight restrictions imposed by this relationship are illustrated in Figure 11.5. While the drone's structure is designed for the flight envelope depicted in Figure 11.4, optimal payload condition is maintained by operating within the green region of combined horizontal accelerations and bank angles. This is particularly crucial during banked turns and also provides insight into the limits on horizontal accelerations to avoid slipping. For straight, horizontal flight, this corresponds to a y-intercept of 3.9.

Considering these limits, the minimum desired turn radius of the drone was determined from Equation 11.12. With a maximum bank angle θ of 22°, achieved under zero horizontal acceleration conditions from Figure 11.5, a minimum turn radius of 52 m was obtained. It is important to note that this is not the absolute minimum turn radius achievable for the drone, which would be significantly lower if calculated based on the maximum lift coefficient of the wing and propeller thrust. However, this calculated minimum desired turn radius can be used for path planning within operations (see Section 12.3) to ensure smooth trajectories that preserve the payload's integrity.

$$R_{min} = \frac{V_{cruise}^2}{g\sqrt{n^2 - 1}}, \quad \text{where} \quad n = \frac{1}{cos\theta} \tag{11.12}$$

Payload-Range

The payload–range diagram is a classical aerospace performance tool used to illustrate the trade-off between payload mass and achievable range, assuming a fixed maximum take-off weight. In conventional aircraft, this typically reflects a single flight mission consisting of take-off, cruise, and landing.

For the BezORger, this interpretation must be adapted. The drone performs multiple energy-intensive phases, including several take-offs, landings, and loitering periods, within a single delivery sortie. However, the payload–range diagram is based on a simplified mission model that includes only one instance of each phase, followed by a single cruise segment, which underestimates total energy consumption and overstates the effective range. As such, it does not represent the true energy profile of a full mission but offers a valuable abstraction of the drone's cruise capability under varying payload and battery configurations.

Figure 11.6 presents the payload–range diagram for the BezORger, assuming a 20% battery reserve. Point A corresponds to the maximum payload case with minimal battery mass. Between points A and B, additional battery mass is added until the drone reaches its MTOW. From B to C, the payload is reduced to allow for further battery capacity, eventually reaching the zero-payload, full-battery configuration.

This diagram highlights the operational flexibility provided by modular battery configurations: short-range missions can be completed with fewer batteries, while long-range operations require sacrificing payload for additional energy. The BezORger demonstrates efficient cruise performance, achieving over 100 km range under optimal conditions.

11.4.2. Verification & Validation

The flight envelope of the drone was verified by performing a sensitivity analysis of the variables used to construct the manoeuvre and gust diagrams, for example the drone MTOW, wing surface area, lift coefficient, and seeing if the curves adapted accordingly. The plotting was verified with by hand calculation for the stall limit, and visual inspection for parameters such as the cruise velocity and max drone speed. The shape of the diagram also matched the theoretical shape expected from literature [26].

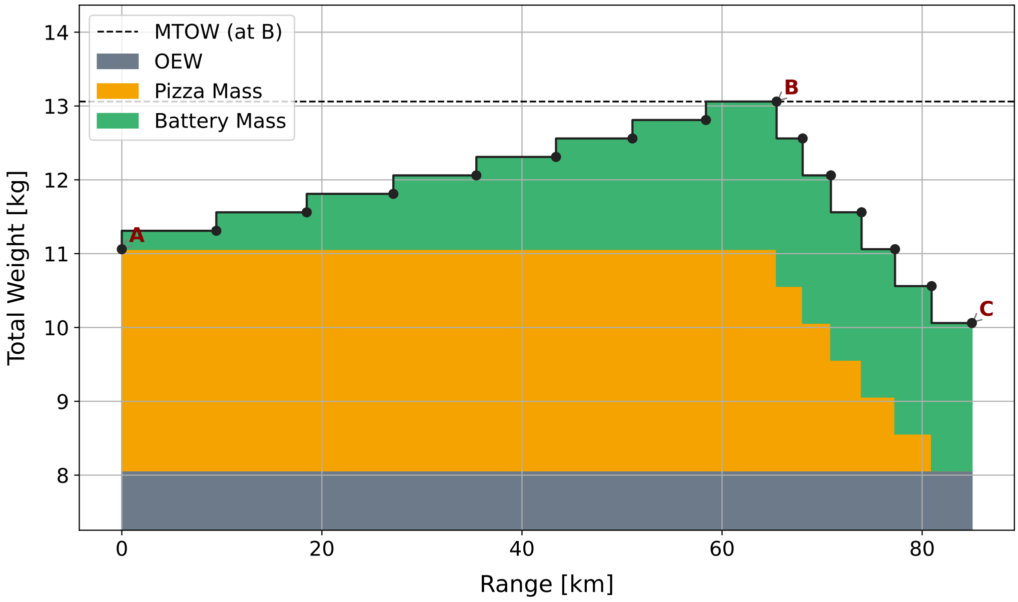


Figure 11.6: Payload–Range diagram for the BezORger

The payload range diagram was verified similarly by visual inspection and also changing parameters such as the pizza mass, number of batteries and seeing whether the graph responded accordingly.

Validation of the flight envelope needs to be done by performing test flights of the drone and checking whether the drone is able to sustain different combinations of load factor and velocity as specified by the diagrams. The payload-range diagram can also be validated by comparing the results with a test flight of the drone at varying payload and battery masses.

11.5. Recommendations

Based on the modelling and analysis performed, several areas have been identified for further improvement and deeper investigation:

- **Transition Modelling:** The transition phase between vertical and horizontal flight should be modelled in more detail. This includes incorporating stabilisation at the end of the transition, using higher-fidelity thrust profiles, and better capturing drone dynamics and aerodynamics, especially at low speeds and high angles of attack. Setting up an optimisation framework for the transition trajectory would allow for more efficient and realistic energy estimates.
- **Payload–Range Diagram Extension:** The current payload–range diagram assumes a single take-off and landing. Extending this analysis to account for multiple delivery legs, each involving additional take-offs and landings, would give more practical insights into the energy demands of real missions. This would support better operations planning, particularly when determining the required number of batteries for a given delivery schedule.
- **Take-off and Landing Power Assumptions:** The thrust-to-weight ratio used for estimating energy during take-off and landing should be revisited. These phases are among the most energy-intensive in the mission, and a refined estimate could improve sizing accuracy and overall energy budgeting.

12 Operations

This chapter presents the simulation framework used to assess the real-world feasibility of the DronebezORgd system. The challenge of planning and optimising the fleet operations has been divided into two broad modules — Mission Planning, which deals with scheduling and allocating orders to drones, and Path Planning, which includes computing the exact trajectory of a drone subject to no-fly zones, minimising noise disturbance, and flight dynamics.

Section 12.1 outlines the approach taken to solve the problem, including all simplifying assumptions made. Section 12.2 covers the Mission Planning module, focusing on the exact mathematical formulation of the problem. Section 12.3 details path planning, including algorithm selection, implementation and smoothing techniques. Then, everything is put together in Section 12.4, and the simulation is run, calculating the revenue and operational profit of the business. Finally, Section 12.5 presents recommendations for future work based on current findings.

12.1. Design Approach

This section designs the approach upon which the operations of DronebezORgd are designed. Subsection 12.1.1 gives an overview of the structure of the operations, and Subsection 12.1.2 describes the assumptions upon which operations design is based.

12.1.1. Structure Overview

A key component of the DronebezORgd platform is its operational design, describing how drones are assigned deliveries and how those deliveries are executed in practice. To manage this, the core challenge is framed as a dial-a-ride problem (DARP), where packages must be picked up from depots and delivered to customers within specific time windows, using a limited fleet of drones. Given the complexity and real-time nature of drone logistics, the operational framework is split into three interacting layers: **Mission Planning**, **Path Planning**, and **simulation**. By structuring the operations in this three-layer framework and adopting a hybrid solving approach, the DronebezORgd platform aims to be both efficient in planning and sustainable in its operations.

Mission Planning

Mission planning focuses on the high-level scheduling of drone activity. It determines which drones handle which deliveries, when they depart, what routes they take, and when they return to the depot. For simplicity, this layer assumes straight-line travel between points, allowing for a cleaner allocation of resources and timing in the early stages of decision-making.

This process is framed as an optimisation problem. The goal is to maximise operational profit, which in the frame of one mission plan, is done by maximising the number of deliveries while minimising the energy used.

Constraints arise naturally from the problem structure: each order must be served exactly once, drone capacity and flight range must not be exceeded, and timing constraints must be respected. These form the foundation of the optimisation formulation in Python, where the objective function, constraints, and decision variable vector are explicitly defined.

Path Planning

While Mission Planning assumes idealised travel paths, the real world introduces complexity. Path Planning addresses this by determining the actual routes flown by drones, factoring in obstacles, no-fly zones, noise pollution, and drone flight dynamics. This is done through the use of the A* algorithm on a finite grid with post-processing to account for flight dynamics.

Simulation

The simulation is created to integrate Mission Planning and Path Planning and provide an environment for their testing. This is done by creating a digital representation of a Delft-like city, including the actual restaurant and customer locations. Orders are then generated randomly from customers to create a realistic testing scenario to verify Mission and Path Planning methods, as well as estimate profits and the return on investment of DronebezORgd. The flowchart in Figure 12.1 shows the main steps taken by the simulation by the simulation over one day.

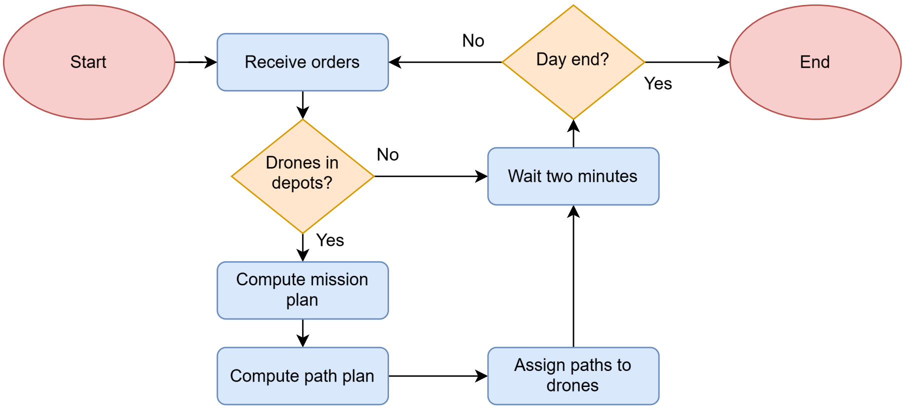


Figure 12.1: Operations flowchart

12.1.2. Assumptions

The following assumptions were made to simplify reality to a level which can be modelled, while retaining as much important information as possible.

- **AS-OP-01:** Mission Planning assumes straight line travel.
- **AS-OP-02:** Mission Planning assumes constant: cruise speed, battery usage rate during cruise, take-off and landing battery usage, order density over time.
- **AS-OP-03:** Mission Planning assumes the drone is always at MTOW.
- **AS-OP-04:** Mission Planning assumes there is always space to land at the restaurants and customers.
- **AS-OP-05:** Mission Planning assumes that ordering frequency at a location is directly proportional to its population density.
- **AS-OP-06:** Mission Planning assumes perfect restaurants: each restaurant always has the food ready at the time of arrival at the restaurant, all restaurants receive the same number of orders on average, all restaurants charge the same amount per pizza, an average of 1000 orders is placed per day over all restaurants.
- **AS-OP-07:** Mission Planning assumes that the number of pizzas per order is a rounded normal distribution with $\mu = 4$ and $\sigma = 1$.
- **AS-OP-08:** Mission Planning assumes that the total number of orders per day is sufficiently high that the variance in the number of orders each day does not affect the number of orders taken by the drones.

- **AS-OP-09:** Path Planning assumes space is defined by a finite grid and every location is quantised to that grid.
- **AS-OP-10:** Path Planning assumes a constant minimum turn radius .
- **AS-OP-11:** Path Planning assumes turning does not affect noise or battery usage.
- **AS-OP-12:** Path Planning assumes public disturbance is proportional to the number of residents flown over in terms of population density.
- **AS-OP-13:** All costs can be derived as a function of simulation variables.
- **AS-OP-14:** Order location probability is proportional to population density.
- **AS-OP-15:** The distribution of orders does not differ by week.

12.2. Mission Planning

Mission Planning concerns itself with the high-level scheduling of drones during operations. Specifically, based on current orders, drone locations and availabilities, Mission Planning is used to create a set of instructions for each drone to deliver as many orders as possible in the most efficient manner. To achieve this, Subsection 12.2.1 describes the design process of the Mission Planning module, including selecting the solution method and its implementation. Subsection 12.2.2 then describes the verification & validation done during that design process.

12.2.1. Analysis

This section describes the design and development of the Mission Planning module. Solution methods are discussed, two of which are developed and tested, and a final selection is made.

Solution Methods

To arrive at an effective Mission Planning solution, various solution methods need to be compared. Table 12.1 describes the traits of four of the methods which were considered in the design process. The first two correspond to Mixed-Integer Linear Programming (MILP) problems, meaning they are fully formulated problems which allow for far-in-the-future planning of the exact problem while complying with all conditions, given that the problem is linear. The greedy heuristic is an easy-to-implement algorithm for one-step-at-a-time planning, and evolutionary algorithms, similarly to MILP, can provide far-in-the-future planning by slightly nudging variables in the right direction over generations in a way which mimics evolution.

Table 12.1: Comparison of Mission Planning solution methods

Method	Runtime	Optimality	Scalability	Comments
Full MILP	High	Optimal	Poor	Too slow for many orders
Clustered MILP	Medium	Medium	Medium	Good trade-off
Greedy heuristic	Low	Poor-medium	Good	Fast but suboptimal
Evolutionary algorithms	Medium-high	Medium	Medium	Needs hyperparameter tuning

Comparing MILP with evolutionary algorithms in Table 12.1, it can be seen that while a standard full MILP may have worse scalability, a clustered approach may fix that by reducing the scale of the problem while not having to deal with the hyperparameter tuning required by evolutionary algorithms. Due to this fact, it was decided not to look into evolutionary algorithms, and instead only into the MILP formulations and the greedy algorithm. Both of these will be explored in the following sections.

Greedy Algorithm

The greedy algorithm is a heuristic that consists of making the locally optimal choice at each step [28]. To apply the algorithm to this particular problem, the following steps are taken by each drone whenever it is ready to leave a node:

1. Get orders.
2. Filter orders by time, whether the drone has enough capacity for the order, and whether the order is already being delivered.
3. Sort orders by distance from the drone to the restaurant the order was placed at from shortest to longest.
4. Plan new mission with first order.
   (a) Make a list of nodes with the orders from the drone's previous mission plan + the new order's restaurant.
   (b) Find the nearest node to the drone. Add that node to the new mission plan and remove it from the list of nodes. If the node is the new order's restaurant, add the order's location to the list of nodes
   (c) Repeat 4b until all nodes have been added.
5. Check if the new mission plan meets time and battery constraints. If yes, update the mission plan to the new mission plan, If not, return to step 4 with the next order in the list.

As seen in this list of steps, it is called in a slightly different way to the method shown in Figure 12.1. The main advantage of this algorithm is its speed; it takes a very short time to compute the mission plan as it essentially computes only one step at a time. However, this also comes with some disadvantages. Besides the obvious disadvantage of not finding an optimal solution, it also only decides whether an order is taken by the drones a few minutes in advance, not leaving enough time for another delivery service to take the order instead and leading to delays.

Dial-A-Ride Problem Formulation

In literature, there exist many standard types of linear delivery optimisation problems. These start with the simplest, the Vehicle Routing Problem (VRP), which attempts to find the optimal delivery routing between one or more depots and several customer locations while satisfying some simple constraints [29]. The VRP can be extended to variants with more constraints, such as the VRP with time windows (VRPTW) or the VRP with pick-up and delivery (VRPPD). These models are further generalised to the standard problem which best describes the DronebezORgzd operations, the Dial-A-Ride Problem (DARP). The Dial-A-Ride Problem is used to design vehicle routing schedules of a fleet of vehicles for a set of customers who specify pick-up and delivery locations.

A standard three-index formulation was taken as a start from Cordeau and Laporte [30]. This formulation is expanded by adding extra variables and constraints to account for battery limits, variable starting locations, and various servicing and waiting times. Furthermore, certain mathematically unnecessary yet logically sound constraints are added to reduce the feasible solution space of the problem.

In the proposed model, V is the set of all vertices in the form $\{Z_{\text{start}}, P, D, Z_{\text{end}}\}$, where Z_{start} and Z_{end} are two copies of the depots and make up the set Z, and P, D are respectively the sets of pick-up and delivery nodes, each with length n. Note that even though multiple orders can come from the same restaurant, they are modelled as two pick-up nodes at the same location. The other constants used to constrain and solve the MP problem are presented in Table 12.2. The values associated with each constant are calculated based on the current state of the simulation, as will be described in Subsection 12.4.1, and passed to Mission Planning to be used in the model.

Table 12.2: Constants of the extended DARP model

Constants	Description
dist_{ij}, t_{ij}	Matrices holding the distances and travel times between any pair of nodes.
trav_{ij}	A boolean matrix which states true if there is a positive distance between i and j, and otherwise states false.

M	A constant, Big M, a large number used to dominate any inequality and is therefore used to deactivate constraints.
s^k	Array holding the starting node of drone k. This has been set to the depot in the used configuration.
e_i, l_i	Arrays holding the start and end bounds of an order's scheduled time.
w_i, $W_{\max}$	Arrays holding the minimum and maximum waiting times at each node. These represent the times it takes to pick up pizza, deliver pizza, or replace batteries at a depot.
$H_{\max}$, $T_{\max}$, $R_{\max}$	Constants representing the maximum hover time at each node, full journey time for each drone, and ride time of each pizza.
E_{meter}, $E_{\text{TO/land}}$	Constants representing the energy used per meter of flight, and per each take-off and landing.
w_1, w_2	Constants representing the relative weight of the number of deliveries and finish time in the objective.
b^k_{depot}	Array holding the battery level each drone leaves the depot with. In the current configuration, this is set to full charge.

The decision variables used in the model can be found in Table 12.3. These include all of the variables which a solver can change the value of to arrive at an optimal solution. While most of the variables are not used in the objective function, they are still present to form the logic of the problem.

Table 12.3: Decision variables of the extended DARP model

Variable	Description	Variable Type
x^k_{ij}	Drone k flies from node i to node j	Binary
u^k_i	Arrival time of drone k at node i	Continuous
v^k_i	Departure time of drone k at node i	Continuous
l^k_i	Load of drone k at node i	Integer
b^k_i	The battery of drone k at node i	Continuous

The full formulation of the extended DARP problem can be seen below. Equation 12.1 is the objective function which should be maximised to find the best solution, and equations 12.2 to 12.19 form the constraints of the model. Each constraint is then given a category and explained in Table 12.4.

$$\text{Maximise} \quad w_1 \sum_{i\in D} \sum_{\substack{j\in V\\ j\neq i}} \sum_{k\in K} x^k_{ij} \; + \; w_2 \sum_{i\in Z_{\text{end}}} \sum_{k\in K} u^k_i \tag{12.1}$$

Subject to

$$\sum_{j\in V} \sum_{k\in K} x^k_{ij} \leq 1 \quad (i \in P), \tag{12.2}$$

$$\sum_{j\in V} x^k_{ji} - \sum_{j\in V} x^k_{n+i,j} = 0 \quad (i \in P,\ k \in K), \tag{12.3}$$

$$\sum_{j\in V\setminus s^k} x^k_{s^k j} = 1 \quad (k \in K), \tag{12.4}$$

$$\sum_{j\in V} x^k_{ji} - \sum_{j\in V} x^k_{ij} = 0 \quad (i \in P \cup D,\ k \in K), \tag{12.5}$$

$$u^k_j \geq v^k_i - M(1 - \sum_{i\in V\setminus\{j\}} x^k_{ij}) \quad (i \in P,\ j \in D,\ k \in K), \tag{12.6}$$

$$u_j^k \geq v_i^k + t_{ij} - M(1 - x_{ij}^k) \quad (i, j \in V,\ i \neq j,\ k \in K), \tag{12.7}$$

$$u_j^k \leq v_i^k + t_{ij} + H_{\max} \cdot \mathrm{trav}_{i,j} + M(1 - x_{ij}^k) \quad (i, j \in V,\ i \neq j,\ k \in K), \tag{12.8}$$

$$u_{2n+1}^k - u_0^k \leq T_{\max} \quad (k \in K), \tag{12.9}$$

$$u_i^k \geq e_i - M(1 - \sum_{j \in V \setminus \{i\}} x_{i,j}^k) \quad (i \in D,\ k \in K), \tag{12.10}$$

$$u_i^k \leq l_i + M(1 - \sum_{j \in V \setminus \{i\}} x_{ji}^k) \quad (i \in D,\ k \in K), \tag{12.11}$$

$$u_j^k - u_i^k \leq R_{\max} + M(1 - \sum_{i \in V \setminus \{j\}} x_{ij}^k) \quad (i \in P, j \in D, k \in K), \tag{12.12}$$

$$v_i^k \geq u_i^k + w_i - M(1 - \sum_{j \in V \setminus \{i\}} x_{ji}^k) \quad (i \in V, k \in K), \tag{12.13}$$

$$v_i^k \leq u_i^k + W_{\max,i} + M(1 - \sum_{j \in V \setminus \{i\}} x_{ji}^k) \quad (i \in V \setminus Z, k \in K), \tag{12.14}$$

$$v_{s^k}^k \leq u_i^k + M(1 - x_{s^k i}^k) \quad (i \in V \setminus s_k,\ k \in K), \tag{12.15}$$

$$l_j^k \geq l_i^k + q_j - M(1 - x_{ij}^k) \quad (i, j \in V, i \neq j, k \in K), \tag{12.16}$$

$$l_i^k = 0 \quad (i \in Z, k \in K), \tag{12.17}$$

$$b_j^k \leq b_i^k - \mathrm{dist}_{i,j} \cdot E_{\mathrm{meter}} - E_{\mathrm{TO/land}} \cdot \mathrm{trav}_{i,j} + M \cdot (1 - x_{i,j}^k) \quad (i \in V \setminus Z_{\mathrm{start}},\ j \in V,\ i \neq j,\ k \in K), \tag{12.18}$$

$$b_j^k \leq b_{\mathrm{depot}}^k - \mathrm{dist}_{i,j} \cdot E_{meter} - E_{TO/land} + M \cdot (1 - x_{i,j}^k) \quad (i \in Z_{\mathrm{start}},\ j \in V,\ i \neq j,\ k \in K), \tag{12.19}$$

Table 12.4: Constraints of the extended DARP model

Equation	Constraint Type	Description
12.2	Problem logic	Each customer is served by at most one drone.
12.3	Problem logic	If a drone serves a customer, it must also pick up the order from the corresponding restaurant.
12.4	Problem logic	Each drone must leave from its starting node.
12.5	Problem logic	The drone must leave every node it visits, except depots.
12.6	Problem logic	The drone must visit the restaurant before the customer.
12.7	Time logic	The drone must arrive at a node at the earliest, the fastest travel time after leaving the previous node.
12.8	Time logic	The drone must arrive at a node at the latest, the maximum travel time after leaving the previous node.
12.9	Time logic	The total route must be shorter than the maximum journey time.

12.10	Time logic	The drone must arrive at a customer after the start of the scheduled delivery time window.
12.11	Time logic	The drone must arrive at a customer before the end of the scheduled delivery time window.
12.12	Time logic	The pizza must be delivered to the customer within the maximum time after pick-up from the restaurant.
12.13	Time logic	The drone can only depart from a node after the minimum waiting time.
12.14	Time logic	The drone must depart from a node before the maximum waiting time.
12.15	Time logic	The drone must leave the starting node before arriving anywhere.
12.16	Load logic	The load of the drone must increase by at least the demand of the node it arrives at.
12.17	Load logic	The load of the drone at any depot must be zero.
12.18	Battery logic	The drone battery at a node is at most its battery at the previous node, less the battery spent travelling, taking off, and landing.
12.19	Battery logic	The drone battery after leaving the depot to a node is at most full, less the battery spent travelling, taking off, and landing.

Some additional constraints were added that are not strictly necessary but reduce the scope of the problem by eliminating suboptimal solutions. For example, setting all direct depot to customer legs to 0, as it would not make sense to perform this trip.

The objective function of the formulation is presented in Equation 12.1, showing that the problem is trying to maximise the number of deliveries while returning the drones to the depots quickly, so they can be assigned new missions as soon as possible.

To implement and solve the posed problem, the *Gurobipy* Python package is used, allowing for easy implementation as well as ready-to-use solution methods and insightful solution details and progress tracking[1]. *Gurobipy* uses the branch-and-cut algorithm combined with several heuristics to reduce run time and efficiently solve the optimisation problem.

The Mission Planning algorithm was run every two minutes of simulation time and would take all active orders to be delivered within the next 45 minutes and all drones currently at a depot with their batteries ready.

Clustering

As mentioned before, the main problem with the MILP formulation is its runtime, specifically for a high number of nodes, due to DARP being an NP-hard problem [31].

Considering how the expected number of orders is in the order of 1000 per day, this solution is not viable. Therefore, a K-means clustering algorithm was used to group orders based on location both of the customer and the restaurant for each order. While this means that an optimal solution is no longer guaranteed, the algorithm should still produce close to optimal solutions as multi-delivery trips should be in the same area and thus the same cluster most of the time.

Even after applying the clustering method, runtime to find optimality within each cluster is too high as the algorithm needs to be run for every cluster and the clusters still need to be large enough that the drones can fully use their battery. To mitigate this, a limit on solution time of 5 s per cluster was implemented. This limit was chosen because it was observed from testing that most of the time, the objective function would stay almost constant after this time.

12.2.2. Verification & Validation

In order to verify the correctness of the code, several checks were performed, particularly for smaller problem sizes. An animation was also made to easily visualize the orders and the drones' movement

[1] `https://docs.gurobi.com/current/` [cited 15 June 2025]

around the city.

Greedy Algorithm

In order to verify the greedy algorithm, the code was run for a reduced problem size of two drones and six orders and the results were compared against the manually solved plan. Then the animation was run to check that the drones actually followed the plan that they were assigned. Finally, edge cases where the drones would not have enough capacity, time or battery to fulfil all orders in one trip were tested.

Dial-A-Ride

Verification of the DARP implementation was also performed on a smaller problem size. Since *Gurobipy* was used to solve the problem, verification of the optimisation method itself was not performed, and the focus was instead on verifying the problem setup and the results. The optimal solution was calculated manually by brute forcing with two drones and four orders and compared against the output of the code. The animation was also used to verify that the drones were being assigned the correct orders. Then each constraint was verified by checking the effect of removing them or making them more constraining where applicable, such as increasing battery usage rate for example. Upper and lower bounds were implemented on all the free variables of the model to ensure the model would not cheat big-M constraints and to reduce the scope of the problem. Finally, many different terms such as arrival time at the depot or number of take-offs and landings and relative weights of the objective function were tested to see their effect on the optimality and time efficiency of the solution found. For example, the weight of the second term was verified by checking that increasing it greatly in magnitude meant the drones would never leave the depots, thus earning 0 revenue.

Clustering

K-means clustering was performed using the *scikit-learn*[2] library, so again the method itself was not verified . In this case the focus of the verification was in ensuring that the clusters met the requirements of the MILP while performing correctly. An intermediate size of ten orders was tested with two drones and compared to the mission plan without clustering. While the mission plan without clustering could not find optimality within 10 min, it seemed to converge to a value which was assumed to be close to the optimum. Then clustering with different criteria such as customer location, restaurant location and delivery time were tested to see which criteria best approximated the unclustered mission plan in most cases. Another verification was to ensure that clusters would not be too large by reducing them to a maximum size since the MILP would not handle large numbers of orders well.

12.3. Path Planning

Path planning concerns itself with finding the shortest and least public-disturbing paths between two points on a map while avoiding obstacles and no-fly zones and complying with drone flight dynamics. In Subsection 12.3.1, the complete Path Planning analysis is presented, including the selection of the path finding algorithm and its smoothing to comply with drone flight dynamics. Subsection 12.3.2 then discusses the verification & validation process.

12.3.1. Analysis

This section describes the design and development of the Path Planning module. This includes describing the selection of the Path Planning algorithm, its implementation, and the smoothing procedure used to make it comply with the BezORger's flight dynamics.

Algorithm Selection

Before analysing the possible solution methods, two mapping approaches are considered: the grid approach, where the map is discretised into an even grid, and the waypoint approach, where waypoints

[2] `https://scikitlearn.org/stable/modules/clustering.html` [cited 16 June 2025]

are added only to specific points on the map, such as next to each obstacle. While it may seem as though the way point approach allows for points to be added only where necessary, leading to fewer possible routes and reducing computational time, this is not necessarily the case. Assuming obstacles do not block most paths, the order of legal moves between control points in the whole map is n^2, where n is the number of nodes. In the grid approach, however, the number of legal moves is in the order of $8n$ as the drone can only move from its current node to one of the 8 nodes around it. Due to the large size of the city maps considered, the grid approach was selected for its better scalability in terms of computational time. Furthermore, the grid approach allows for a better interface with the simulation, which will be discussed in Section 12.4.

Next, a solution method is chosen based on runtime, optimality, and implementability. Table 12.5 shows four possible Path Planning methods: Dijkstra's algorithm, the A-star (A*) algorithm, which is an extension of Dijkstra's, the Rapidly-Exploring Random Tree (RRT) algorithm, which stochastically approaches the target, and learning methods, which optimise Path Planning over time.

Table 12.5: Comparison of Path Planning algorithms

Algorithm	Grid-Based	Runtime	Optimality	Notes
Dijkstra	Yes	Medium	Optimal	No heuristics
A*	Yes	Fast	Optimal	Heuristic speeds up search
RRT	Not ideal	Fast	Suboptimal	Good in high dimensions
Learning methods	Yes	Fast	Suboptimal	Needs lots of training

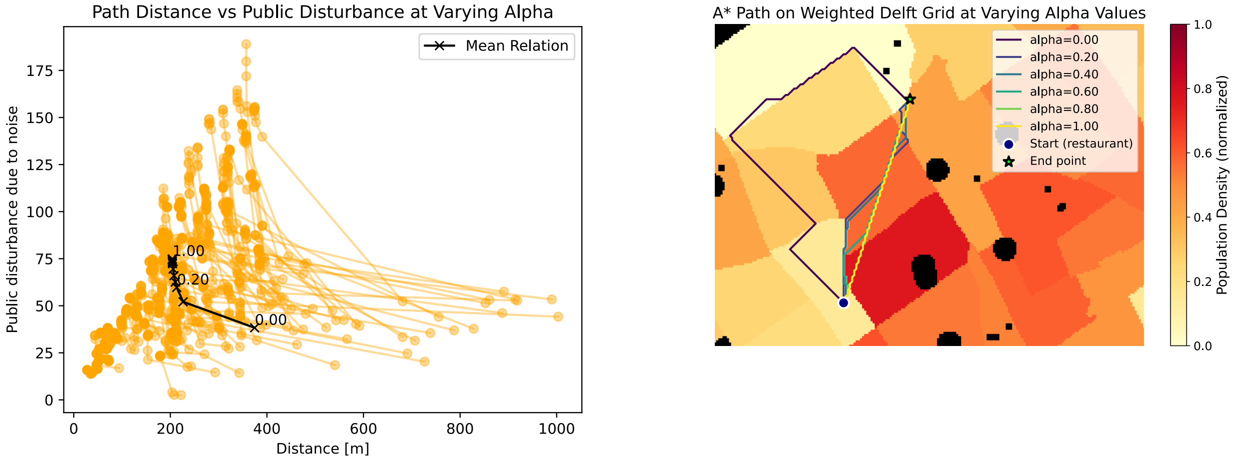
Looking at Table 12.5, it can be seen that only the top two methods guarantee optimality, while RRT and learning methods cannot be trusted to always provide a consistently good and robust solution. Furthermore, the A* algorithm has a much faster average runtime than Dijkstra's while being almost as easy to implement, making it the obvious choice for Path Planning. While extensions of A*, such as Theta*, which allows for any angle Path Planning, were also considered, it was deemed that A* was better suited due to faster run times and more robustness.

A* Implementation

The implementation of the A* algorithm in Path Planning follows that of Zammit and Kampen [32]. The algorithm can be extended, however, by including a term related to public disturbance in the objective function, greatly improving the social sustainability and public acceptance of DronebezORgd. This can be done by assuming that public disturbance per distance flown is proportional to population density over that distance, as stated in AS-OP-12. This is summarised by Equation 12.20 where α dictates the relative weight of distance cost versus disturbance cost. The third cost, deviation cost, was added to make the algorithm prefer a straight route to the goal by penalising deviations from a straight line from the current node to the end node, with an additional weighting factor, β, which is treated as a hyperparameter.

$$c_{\text{total}} = \alpha \cdot c_{\text{distance}} + (1 - \alpha) \cdot c_{\text{disturbance}} + \alpha \cdot \beta \cdot c_{\text{deviation}} \tag{12.20}$$

The effects of varying α can be seen in Figure 12.2b. At high values of α, the paths are very direct to the target. As α is reduced, the paths can be seen reducing the distance they travel in densely populated areas, instead taking longer routes through less populated areas. By simulating 100 orders from random restaurants to random customer locations at varying α, the distance and public disturbance due to noise can be plotted as seen in Figure 12.2a. It can be seen that lowering α initially greatly reduces public disturbance while barely increasing path distance. A natural elbow can be seen at an α of 0.2, below which further reductions start to greatly increase the distance travelled. Therefore, to minimise noise pollution and increase social sustainability while barely affecting route distances, an α value of 0.2 is chosen.



(a) Average distance and public disturbance for varying α over 100 runs

(b) A-star paths for varying α

Figure 12.2: A-star implementations at varying α

Path Smoothing

Once a path is found with the A* algorithm, it must somehow be smoothed to comply with the drone's flight dynamics. Specifically, since the full path is assumed to be done during cruise at constant speed, where take-off, landing, and transition are handled separately, this implies that the path needs to comply with a minimum turn radius requirement of 55 m, including safety factors. This converts the grid-based, discrete path to a near-continuous path which the drone would take through the city. It is important, however, that after smoothing, the path still complies with no-fly zones and does not hit any obstacles.

Table 12.6 presents the four considered smoothing methods, namely: a corner rounding method, which simply replaces all sharp turns with circular arcs of the set minimum radius, an iterative mass spring system simulation, where points want to straighten out yet are repelled from obstacles, a basis splining method which connects the centre points of each grid node with a basis spline, and Bezier curves which using control points, create efficient smooth curves to construct the path.

Table 12.6: Comparison of path smoothing techniques

Method	Description	Runtime	Smoothness	Notes
Corner rounding	Replace corners with circular arcs using turning radius	Fast	Medium	Easy to implement
Mass-spring	Treat path as particles connected by springs; simulate until equilibrium	Medium	Medium	Robust obstacle avoidance
B-spline	Fit spline curves through path points	Fast	High	Easy to implement
Bezier curves	Use control points to construct continuous smooth curves	Fast	High	Requires preprocessing

While the corner rounding method is easy to implement and robust, it was eliminated as an option due to its unrealistic path, which, while complying with the minimum radius constraints, would often make the drone oscillate unnecessarily. The remaining three methods were tested on simple cases to evaluate their performance. The mass-spring system had very robust obstacle avoidance as repulsion from obstacles was modelled inversely to distance, meaning paths would never approach too close. The paths, however, could not be made to always satisfy the minimum radius requirement, as this was not directly implemented in the model. Furthermore, the runtime was found to be much worse than other methods, as many iterations would be required to reach equilibrium. The Bezier curves were also found to be suboptimal as they required pre-processing to determine which nodes would be used as control points to create the curves, and the implementation was found to be challenging and not very robust. The basis splines were found to be the best solution due to their fast run times, high levels of smoothness and realistic pathing, and ease of implementation.

To implement the B-spline path smoothing, the *splprep* and *splev* functions were used from the *Scipy* Python package[3]. While these functions do not directly incorporate a minimum radius, the smoothing factor in the function can be iteratively increased until a suitable path is calculated. This slightly increases run time, although the smoothing still runs very fast, as the order of iterations is only in the order of 10. Furthermore, to ensure obstacle avoidance, the Path Planning is computed on a grid with a 55 m length per grid box. This means not only that each no fly zone is additionally padded with an additional buffer area, but any circular arc connecting the centre of two nodes will have a radius of 55 m, meaning feasible paths will not collide with obstacles no matter how sharp a turn is made.

In conclusion, the Path Planning module, given a start and end point, computes an optimal flight path on a discrete grid using the A* algorithm. This path is then iteratively interpolated using basis splines of increasing smoothness until it meets the drone's flight dynamics requirements. The resolution of the grid is set such that obstacle avoidance is inherently present in the smoothed path. Finally, the near-continuous calculated path is returned.

12.3.2. Verification & Validation

This section presents the verification of the Path Planning algorithms. A bottom-up approach has been used, where first unit tests were performed, followed by system tests and finally, robustness tests have been performed.

Grid Verification

To ensure that the Path Planning algorithms correctly identify tall buildings and no-fly zones, a unit test was performed on the five tallest locations in Delft[4] (e.g., Nieuwe Kerk) and the Delft police station [5]. This test verified that each location is assigned an infinite weight on the grid, ensuring these areas are effectively avoided in route planning. Similar tests were performed on areas of known population density to ensure the values were correct before normalisation. Afterwards, a unit test was performed on all non-infinity grid nodes to verify that no node had a value greater than 1.0 or less than 0.0. To ensure that each height point, originally measured at 0.5 m accuracy, is correctly mapped to the grid with a cell size of 100 m^2, the height data was clustered using the DBSCAN algorithm[6]. A unit test was performed to verify that each clustered point is assigned to the correct grid cell and that all height data is represented without loss or duplication. This verification confirms the correct implementation of the clustering and mapping process.

Correct Path Finding

To ensure that each path avoids both obstacles and no-fly zones, a unit test was performed in which the sum of the weights of all traversed nodes was calculated and verified to be finite.

Grid Resolution Robustness

To verify the robustness of the A* algorithm across different spatial resolutions, tests were performed using both finer and coarser grid settings. The planner was evaluated to ensure it consistently produced valid and feasible paths, maintained obstacle and no-fly zone avoidance, and respected kinematic constraints, regardless of the chosen grid resolution.

[3] `https://docs.scipy.org/doc/scipy/reference/interpolate.html` [cited 15 June 2025]
[4] `https://nl.wikipedia.org/wiki/Lijst_van_hoogste_gebouwen_van_Delft` [Accessed 16 June 2025]
[5] `https://www.politie.nl/mijn-buurt/politiebureaus/06/bureau-delft.html` [Cited 16 June 2025]
[6] `https://scikit-learn.org/stable/modules/generated/sklearn.cluster.DBSCAN.html` [Cited 16 June 2025]

Handling Orders in Restricted Areas

As a system test, the planner was evaluated by simulating orders placed at inaccessible locations, such as inside no-fly zones or tall buildings, to verify that the system consistently directs users to the nearest accessible pick-up point and ensures successful order delivery in all cases.

Flight Dynamics

To ensure that smoothed paths adhere to flight kinematics, unit tests verified that the minimum turning radius along each spline path remained above the BezORger's constraint by estimating local curvature, and confirmed that no smoothed path intersected obstacles or no-fly zones by representing paths as geometric lines and zones as polygons, then checking for intersections using standard geometric operations. These tests guarantee that all generated flight paths are both physically feasible and safe.

Maximum Obstacle Density

As an extreme value test, the Path Planning system was evaluated under conditions of maximum obstacle density (Papsouwselaan). The system was verified to successfully generate paths that remain feasible while minimizing the number of required manoeuvres, demonstrating robust performance even in highly constrained environments.

12.4. Simulation and Results

To allow for the verification of the realistic performance of Mission Planning and Path Planning, and to establish the operational capacity, which would be most profitable while meeting all requirements, a simulation of a city with real-time orders, integrated with the Mission Planning and Path Planning modules, must be created. This section details the creation of this simulation as well as the operations design found to best maximise the return on investment. Subsection 12.4.1 presents the analysis done to arrive at a working simulation as well as the results concluded from it, and Subsection 12.4.2 presents the verification & validation conducted to provide confidence in those results.

12.4.1. Analysis

This subsection discusses the analysis related to the creation and working of the simulation used for the design and verification of the operations of DronebezORgd. This is followed by the refined estimates of profit and return on investment found using the simulation.

Simulation Structure

The simulation is created to design and verify the operations of DronebezORgd. It does this by creating a digitised version of a Dutch city, as will be discussed in the next section, simulating orders within that city in real time, and integrating Mission Planning and Path Planning to assign the drones to take each order, their route, and their pathing. In the case of the greedy algorithm, Mission Planning is calculated every time a drone is ready to leave a node, whereas in the case of the DARP formulation, it is calculated every 2 min if there are drones ready at a depot. In both cases, Path Planning is computed when the drone leaves for its next node.

To ensure coherence between the two layers, a validation step compares the actual cost associated with the assumptions made in the mission plan. If the deviation exceeds a certain threshold, replanning may be triggered to preserve reliability and efficiency. However, if not, it is assumed that the mission plan affects the path plan but not the other way round, to minimise the computations required for an iterative operational planning.

Once the three modules are integrated, an outer loop is constructed, allowing the simulation to be run multiple times with varying variables such as the number of drones, the number of depots, and depot locations. The outcomes of these simulations will be described in the results section.

Delft-Like City Digitisation

While the focus of DronebezORgd and its profit estimates are specific to the Dutch city of Delft, the simulation was created to allow for the digitisation of any Dutch city. The creation of this digitisation is described in this section, including the methods by which the necessary data was acquired.

The data used for the locations of each neighbourhood, their respective population densities, and the border of the city, was taken from the Centraal Bureau voor de Statistiek[7]. To get actual restaurant locations and no-fly zones, the Overpass API, specifically the *Overpy* python package[8] was used. The Overpass API allows for data extraction from the OpenStreetMap (OSM)[9], an open-source world map. This allows the simulation to use all locations tagged as pizza places for the restaurant locations, which should be almost exactly true to reality. No-fly zones are also defined as places which drones should avoid, including hospitals, police stations, fire stations, and schools.

Using these two sources, a digitised map of any Dutch city can be created and used to design the operations of DronebezORgd. This map will be used in the following sections to simulate orders and estate profits. This map, with all the elements of the simulation can be seen in Figure 12.3. In this figure, the Mission Planning elements - the orders, restaurants, depots and drones - are overlayed on the Path Planning elements. These are the no fly zones such as tall buildings, schools and the outer perimeter of Delft, indicated in black, the population density indicated on a yellow to red scale, and the paths indicated by pink lines.

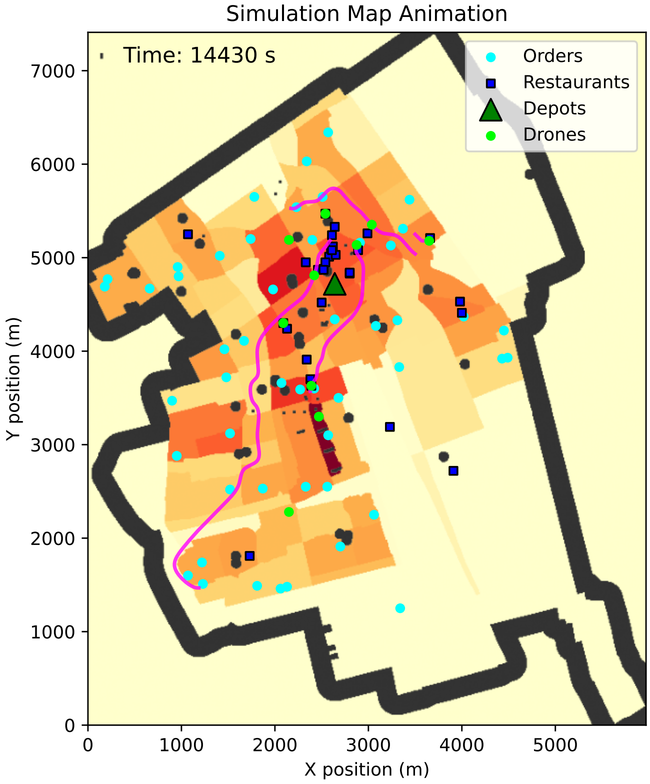


Figure 12.3: Map of Delft showing drones, restaurants, depots, orders, no fly zones, population density and drone paths

[7] https://www.cbs.nl/nl-nl/cijfers/detail/86003NED?q=wijken%20en%20buurten%202024%20bevolkingsdichtheid [cited 16 June 2025]

[8] https://python-overpy.readthedocs.io/en/latest/ [cited 16 June 2025]

[9] https://www.openstreetmap.org/#map=7/52.154/5.295 [cited 16 June 2025]

Mission Planning Algorithm Choice

Out of the three algorithms that were, only the clustered MILP and the greedy heuristic could realistically be compared, as the full MILP took too long to produce results. The two remaining algorithms were compared by profit after 5 years and runtime for the simulation of a single day. The results can be seen in Table 12.7.

Table 12.7: Comparison of Mission Planning algorithm results

	runtime (s)	profit (€)
Clustered MILP	87	1,450,000
Greedy algorithm	13	1,916,000

From the table, it can be seen that the greedy algorithm outperforms the clustered MILP both in runtime and profit. However, a factor that is not accounted for in the table is the time at which it is decided whether an order will be accepted or rejected. The greedy algorithm only decides this between 7.5 minutes in advance and 2.5 minutes after the scheduled delivery time, which is not acceptable according to Section 3.2. On the other hand, the clustered MILP decides this about 45 minutes in advance. For this reason, it was decided that the greedy algorithm, while useful as a benchmark, would not be used in the final operations and clustered MILP was chosen instead.

Mission Planning Objective Function

As mentioned in Section 12.2, the objective function consists of a weighted sum of two values: the number of orders delivered, and the time of return at the depot for each drone. The first term is evidently necessary to incentivize the drones to deliver orders, while the second term pushes them to finish their mission as soon as possible so that they can start a new one fast. Since there are two terms, their relative weights $\frac{w_1}{w_2}$ have a significant effect on the performance of the optimisation. Therefore, a study on the effect of the weight of the second term of Equation 12.1 on the profit was performed, averaged over five seeds, and the results can be seen in Figure 12.4. For the graph, w_1 was kept constant at $w_1 = 500$ and w_2 was changed.

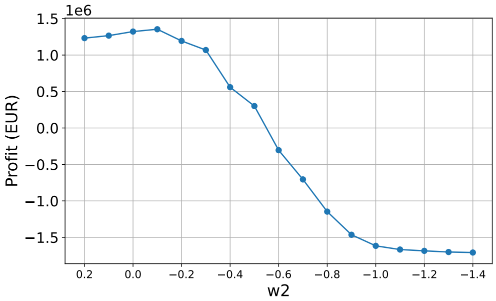


Figure 12.4: Profit vs return to depot time weight

A negative weight means the algorithm tries to get the drones to return as soon as possible. As expected, lowering the weight from zero causes an increase in profit. However, as the weight keeps getting lower, the profit starts decreasing. This is because the weight of returning to the depot late is outweighing the weight of delivering pizzas, so the drones start rejecting orders when they should not. In fact, at the lowest weights, the line flattens because the drones are not accepting any orders at all. From this graph, it can be seen that the optimal weight of the objective term is -0.1. However, the increase in profit compared to not penalising finish time by setting w_2 to 0 is small.

Max Payload Size

The requirement on the number of pizzas that the BezORger must be able to carry is of 5 pizzas. However, due to the design chosen having two payload bays, accommodating a sixth pizza would not be too inconvenient. Therefore, a study was performed on the difference in profit caused by increasing the payload size. The total profit over 5 years was calculated to be €1,143,000 for 5 pizzas and €1,391,000 for 6 pizzas. Based on these results, it was decided that designing for 6 pizzas was worth the increase in profit. However, some limitation of these results to consider are that it was not taken into account that separate deliveries should be in separate compartments to avoid theft and that the cost of the drones was assumed to be the same in both cases.

Profit and Return on Investment

After evaluating the effect of the relative objective weights to find the weight for maximum profit, the number of drones and the position of the depot were analysed. The objective in this case was not only to maximise profit, but also to maximise return on investment (ROI) after five years. To do this, the simulation was run several times for different numbers of drones and depot locations, and the profits in each scenario were plotted. The results can be seen in Figure 12.5 and Figure 12.6a.

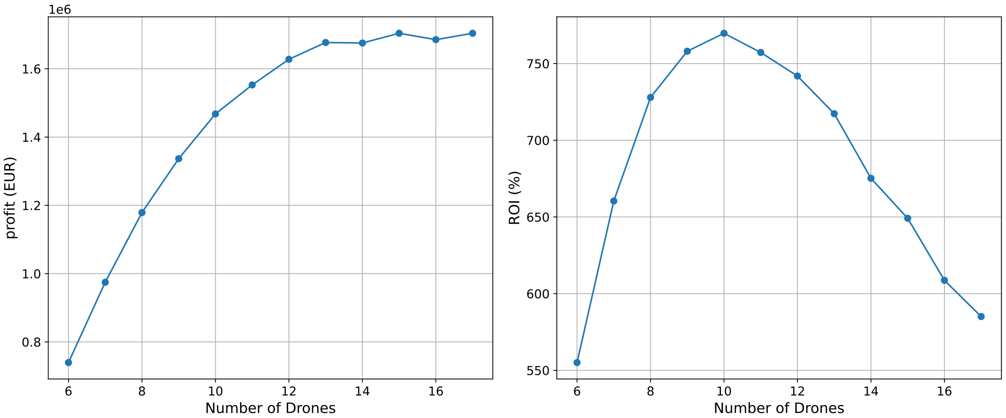
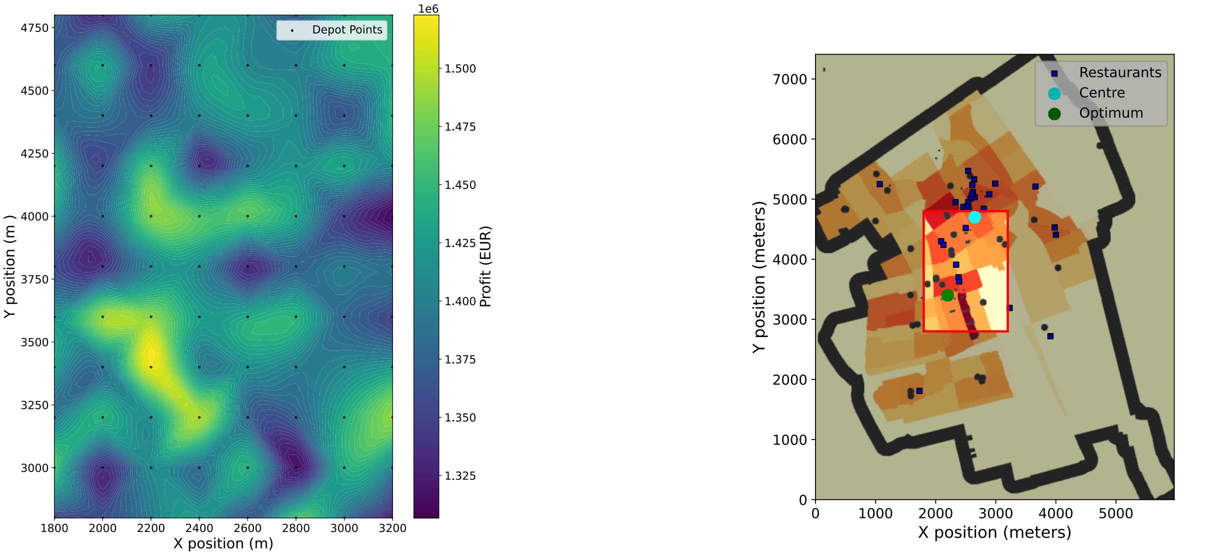


Figure 12.5: Profit and ROI vs number of drones

As expected, initially the profit increases greatly with number of drones as many more orders can be accepted. However as the number of drones increases, the marginal profit decreases, since the drones already capture more demand so the increase is smaller until it plateaus at around 13 drones. However, the peak in ROI happens with slightly less drones, at 10, because the initial investment is smaller. Therefore, it was decided to size the fleet at 10 drones.



(a) Heatmap of profit vs depot location

(b) Delft map areas covered by the heatmap

Figure 12.6: Heatmap of profit and map for reference

In the case of the depot location, there are two factors affecting the profit. From Chapter 3, it is assumed that the depot rent will decrease linearly with distance from the centre, which was decided to be at coordinates (2650,4700) from visual inspection of the map. This point can be seen in Figure 12.6b. However, the closer the city is to the centre, the less distance it needs to cover on average to deliver pizzas. Initially, a coarse grid was made that covered the whole map, and it was found that the optimum was in the region shown in Figure 12.6b, and a second heatmap with greater resolution was made which can be seen in Figure 12.6a. The ROI graph would look the same as the profit graph, as the initial costs are the same. From the graph, it can be seen that the optimal position is around (2200,3400).

Results Summary

Table 12.8 shows a summary of all the decisions made in this chapter. Table 12.9 shows a breakdown of the costs and revenues over the first five years, together with other relevant financial calculations.

Table 12.8: Summary of decisions made

MP algorithm	MP frequency	w_2	Max number of pizzas	Number of drones	Depot location
clustered MILP	120 s	-0.1	6	10	(2200,3400)

Table 12.9: Summary of financial values

Profit	Initial costs	Total costs	Revenue	Break even point	ROI
€1,391,000	€191,000	€1,774,000	€3,165,000	220 days	729%

12.4.2. Verification & Validation

Verification of the overall simulation involved many different aspects. Most of these were performed using the animation. Firstly, verification of the correct implementation of the city data was required. This was done by looking at the animation and comparing the position of the restaurants and population densities to *Google Maps*.

Next, the interface between the simulation and the Mission Planning had to be verified. This was again done using the animation, by printing the outcome of the mission plan adding labels to the orders in the animation. Then the animation was run to check that the drones correctly moved towards their targets.

Like Mission Planning, the implementation of the Path Planning into the simulation also required verification. This was done by checking in the animation, firstly, that the start and end nodes were correct. Then it was checked that the paths correctly avoided obstacles and densely populated areas, and finally that the drones were correctly following the paths at the right speed and arrived on time. Furthermore, to ensure that orders placed within no-fly zones or on top of tall buildings are handled appropriately, a unit test was performed. The test verified that if an order is placed at an inaccessible node, such as within a no-fly zone or a high building, the system identifies the nearest accessible grid node and moves the order to that place.

Finally, the implementation of the financial model into the simulation was verified. This was done by means of a sensitivity study, which can partially be seen in Subsection 12.4.1, where many different parameters were varied and their effects on the costs and revenue were observed.

12.5. Recommendations

Despite the extensive work completed for operations, numerous opportunities for further study still exist. This subsection highlights some of the main areas where more research could be done. Subsection 12.5.1 presents potential future research in Mission Planning, specifically ways to improve the current method and a proposal of other methods to be tried. Subsection 12.5.2 focuses on Path Planning and the ways in which it could be improved by taking additional factors into account.

12.5.1. Mission Planning

While a sensitivity study was performed on the number of drones, depot location and objective function weights, there are many other variables that could be changed to investigate their effect on profit. These include the time in advance an order will be considered, the optimisation time limit, and many drone parameters such as speed or capacity. Furthermore, the simulation model could also be refined to take into account uncertainties in, for example, the pickup and delivery times. Testing the simulation on other cities to calculate profit could also provide useful insight.

Additionally, more work can still be done in improving the optimisation of the clustered DARP method. First, other clustering techniques should be investigated. One possible improvement in this area is to come up with a clustering method that groups orders together in space but apart in time, so that the drones have enough time to deliver all orders without delays. Furthermore, a clustering algorithm with a fixed cluster size would be preferred as the clusters should be limited in size by what the optimisation algorithm can handle efficiently. Finally, the option of updating the mission plan of the drones while they are still in the air instead of only when they are ready in a depot should be explored.

Another area where further research could be done is in investigating completely different optimisation methods. Due to time constraints and the unavailability of data, genetic algorithms and learning algorithms were not explored, but could have the potential to produce efficient solutions with reasonable computation times. The same is true for other heuristic algorithms that are more complex than the greedy algorithm, such as the ones implemented in *VeRyPy*. [10]

12.5.2. Path Planning

A sensitivity analysis on taking more direct paths has been performed as displayed previously in Figure 12.2. However, since the core of DronebezORgd's service relies on reliable and efficient path planning, this component must be continuously improved. Therefore, the following improvements are recommended:

- **Implement 3D Path Planning**: Accurately model the VTOL's operational profile, including the transitional phases from Figure 11.2, to improve flight time estimates.
- **Incorporate Wind Effects**: Integrate live wind data into the path planner to optimise routes for efficiency and safety, making use of available meteorological sources like Windy. [11]
- **Landing Site Verification**: Include real-time feasibility checks of landing sites within the planning loop.
- **Dynamic Collision Avoidance**: Use real-time sensing or external data to detect and avoid unexpected obstacles not included in the static map.

[10] URL:`https://github.com/yorak/VeRyPy/blob/master/README.md#implemented-heuristics-with-references` [cited 16 June 2025]

[11] URL: `https://www.windy.com/` [Cited 17 june, 2025]

13 Design Integration

The design integration represents a key step in the design process after subsystem design. The key function of design integration is to combine the subsystems and iterate the design towards a converged, feasible design that meets the requirements. Section 13.1 presents the design iteration, followed by the internal and external configuration in Section 13.2 and the data handling and communication architecture in Section 13.3.

13.1. Design Iteration

To achieve the final design, careful integration of each subsystem is crucial. This ensures the absence of conflicts and the effective management of subsystem interfaces, which is key to a successful design and compliance with subsystem requirements. The design iteration framework is presented in Subsection 13.1.1, followed by a summary of the design iteration in Subsection 13.1.2. A sensitivity analysis will then be performed to get more insight about the final design, while also serving as a preliminary system-level verification in Subsection 13.1.3.

13.1.1. Iteration Framework

A design iteration framework is set up for the integration of the design. The goal of this framework is to start with the class I weight estimate of the drone from the early design stage and perform Class II weight estimates, while incorporating the different subsystems of the drone following the N^2 chart presented earlier in Figure 4.1. An overview of the design iteration loop has been shown in Figure 13.1.

The input for the iteration process is the Class I weight estimate, derived from a statistical database based on similar drones in the Midterm Report [2]. A regression relating payload mass to the drone MTOW is used for this purpose. The payload mass of the drone was taken as 3 kg, accounting for six pizzas.

The design loop is comprised of four main modules, including structures, stability & control, performance, and aerodynamics. The aerodynamics module performs an internal iteration to converge to a wing planform needed for the given MTOW. This is necessary because the required wing area calculated from the MTOW changes the wing planform, which in turn affects the lift produced, thereby changing the required wing area. Using the new wing planform, the structures module computes the load distribution on the wing fuselage and tail structures. A structural minimisation loop is conducted as explained for each structural element to prevent overdesign, resulting in the final structural mass. With the structural mass and loads acting on the drone with their respective locations, the stability & control module can update the tail sizing and wing positioning according to Subsection 10.3.1. The performance module can then update the battery sizing based on the mission energy estimation presented in Section 11.3.

After these steps are performed, the Class II weight estimate of the drone is revised, taking into account the various components of the drone. The Maximum Take-Off Weight (MTOW) is the summation of the individual weights of all components, following Equation 13.1 where W_i represents the weight of the $i-$th component, n is the total number of components. The following components are considered:

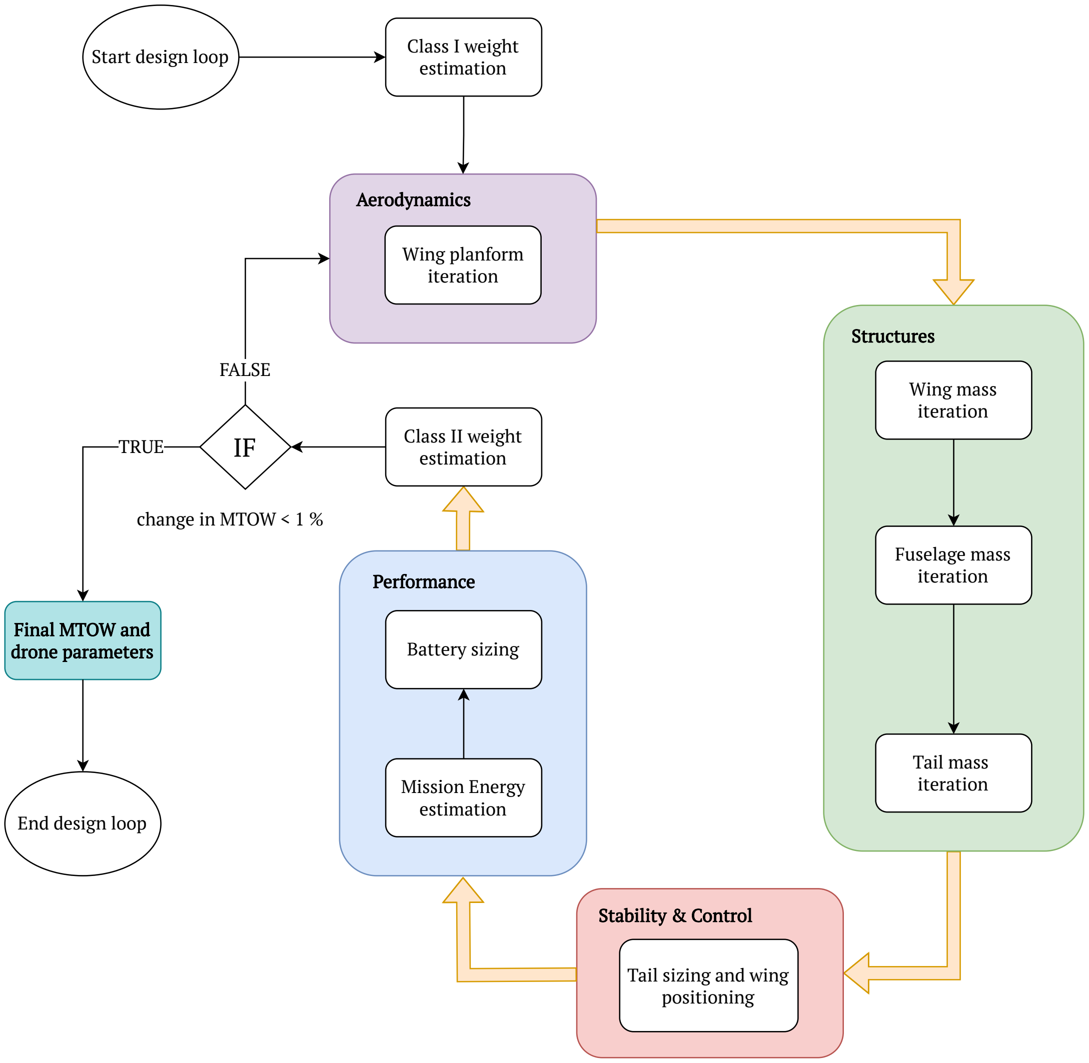


Figure 13.1: Overview of design iteration framework

$$MTOW = \sum_{i=1}^{n} W_i \tag{13.1}$$

- W_{motor}: Motor weight
- W_{prop}: Propeller weight
- W_{batt}: Battery weight
- W_{wing}: Wing weight
- W_{fuselage}: Fuselage weight
- W_{tail}: Tail weight
- $W_{\text{P/L}}$: Payload weight
- W_{fold}: Folding mechanism weight
- $W_{\text{P/L delivery}}$: Payload delivery mechanism weight
- W_{avionics}: Avionics and perception sensor weight

This iteration process is repeated until convergence of the drone MTOW within 0.1%.

13.1.2. Iteration Results

Following the iteration framework, the final iterated MTOW of the BezORger can be obtained, presented in Figure 13.2.

The iteration was first run with a simplified transition model. This simplification implies that the drone operates in vertical mode until cruise altitude and starts cruising, neglecting the transition phase. The result of the convergence is represented by the green line.

Since transition was later explicitly modelled as a flight phase for the mission energy estimation in Subsection 11.3.1, this model was included in the second and final iteration. The final MTOW obtained after convergence is represented by the blue curve differentiated by the label ”With TM”, which gives a **final MTOW** of **13.06 kg**. The inclusion of the transition model results in a slightly lower mass of the drone, which makes sense as the lift generated by the wing now contributes to gaining altitude, requiring less power from the propellers. The converged MTOWs for both cases are remarkably close, with the effect of transition modelling affecting the drone mass by around 0.2 kg.

The payload mass to obtain the initial Class I weight estimate, which is the starting point of the iteration, is taken as the weight of the pizzas, which is 3 kg for six pizzas. However, the effect on the converged MTOW of changing this payload mass to include the delivery mechanism in the regression for Class I weight estimation was investigated. While the delivery mechanism is not part of the payload, it represents additional mass unique to this mission. In Figure 13.2, the addition of the delivery mechanism in the payload mass of regression for Class I weight is represented by ”With DM”. In both cases, the converged MTOW is the same, which indicates that the iterative framework is robust and gives reliable results. However, this is only partial verification, as extensive testing would be needed to perform complete system-level verification of the design iterations.

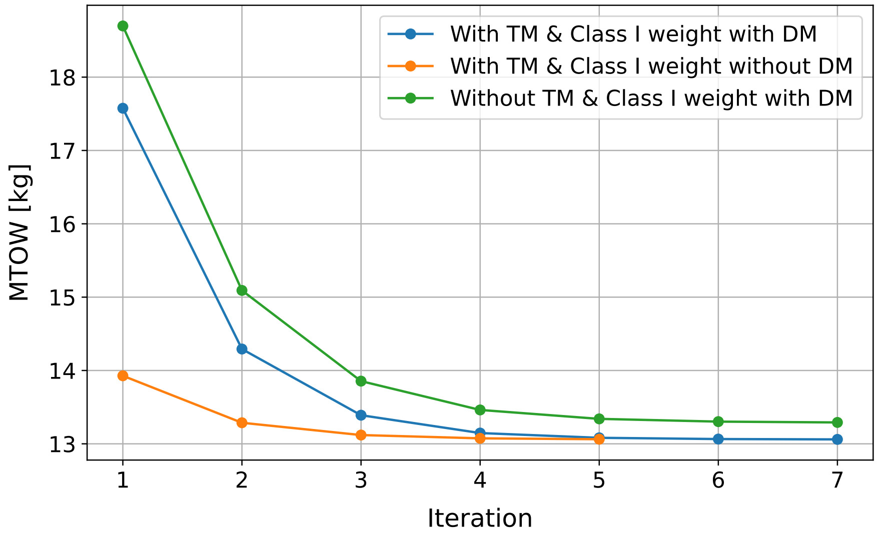


Figure 13.2: Results of the design iterations for different cases. TM refers to the transition model presented in Subsection 11.3.1, replacing a simplified take-off to cruise mode. DM refers to whether or not the delivery mechanism weight is included in the payload weight for Class I weight estimation.

13.1.3. Sensitivity Analysis

A preliminary sensitivity analysis for partial system-level verification is proposed in Subsection 13.1.3. For this analysis, the baseline is defined as the following, corresponding to the blue curve in Figure 13.2:

- **Delivery Mechanism:** Included in payload regression for Class I weight estimation
- **Wing Span:** 3 m
- **Maximum Payload Capacity:** 6 pizzas
- **Cruise Altitude:** 60 m
- **Mission Sequence:** ”DRCCRCD”

The results of changing the wing span to 2.5 m and 3.5 m have been shown in Figure 13.3. Increasing the wing span was shown to result in a higher wing mass, each option varying by around 0.45 kg for an increase of the wing span by 0.5 m. This supports the choice of 3.0 m for the span balances the tradeoff between a lighter and smaller drone versus a larger aspect ratio, which improves cruise efficiency. More analysis is recommended to refine the selected span based on its mass penalty.

Figure 13.4 was used to analyse the effect of the choice of a maximum payload capacity of five or six pizzas. While the MTOW of the drone increases by around 0.6 kg due to the addition of a pizza, as explained in Subsection 12.4.1, an increase in profit of 27.7% was observed. Though this analysis carried limitations by not accounting for limiting a single order to each pizza pile, the modest mass increase supports the decision of choosing a **six-pizza configuration**.

While one might expect a significant mass increase due to the additional battery mass of a sixth pizza, the mass breakdown revealed that the tail mass reduced in the six-pizza configuration compared to five. This reduction was a result of a lower centre of gravity range achieved by a more symmetric pizza arrangement.

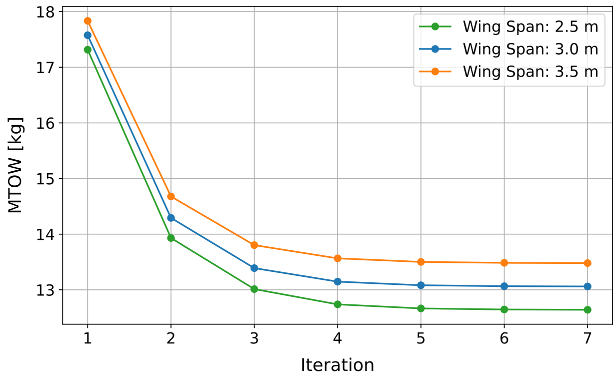


Figure 13.3: Sensitivity of wing span

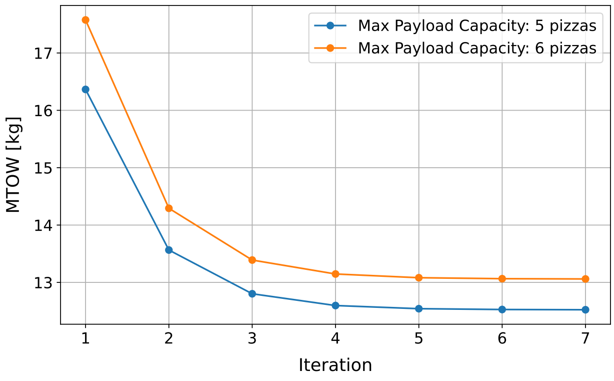


Figure 13.4: Sensitivity of maximum payload capacity

Next, the effect of changing the mission sequence was analysed in Figure 13.5. As expected, a longer mission results in an additional customer leg increases the mass by approximately 0.6 kg. Finally, Figure 13.6 reveals that a slight reduction in the cruise altitude does not affect the MTOW of the drone. This is a strange result and shows the limitations of the transition modelling, as the transition model assumes a constant time of transition. This means that the energy estimate does not adjust based on a lower cruise altitude, revealing a key limitation in the model.

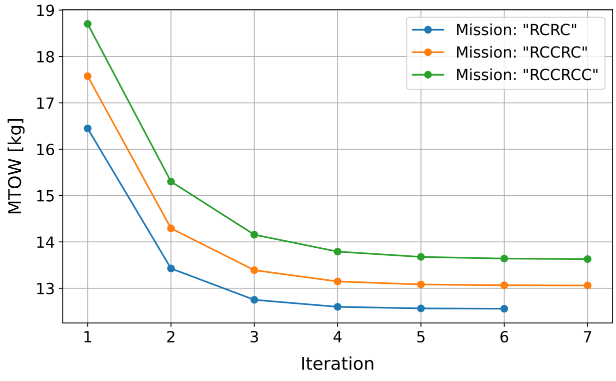


Figure 13.5: Sensitivity of mission sequence

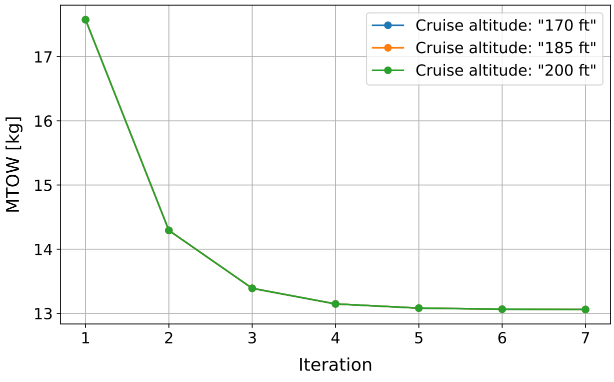


Figure 13.6: Sensitivity of cruise altitude

The sensitivity analysis reveals that the design iteration framework exhibits behaviour as expected in most cases, serving as a preliminary system-level verification. However, more extensive verification and subsequent validation are needed to identify limitations of the model.

13.2. Configuration & Architecture

After the integration of the design, the finalised configuration and architecture of the BezORger can be presented. This includes the external configuration, presented in Subsection 13.2.1, the internal layout presented in Subsection 13.2.2, and the electrical architecture displayed in Subsection 13.2.3.

13.2.1. External Configuration

In this subsection, the 3D model of the BezORger is presented in order to showcase the external configuration of the components. Figure 13.7 displays the general configuration of the drone: four vertical propellers placed at a 3° angle relative to the drone (level with the ground), and a fifth horizontal propeller positioned in the rear of the fuselage. All propellers are surrounded by propeller guards for safety.

The fuselage contains two separate payload bays, with the rear one open for visualisation purposes. Each one can hold up to three pizza boxes. At the nose of the drone lies the main navigation camera, the Ultraleap IR-stereo camera. On top of the fuselage, close to the nose, the communication antenna is located.

In Figure 13.8, the emphasis is placed on the rear of the fuselage. The V-tail is placed on top of the tapering body, at the end of the fuselage, ending where the rear propeller is placed. On each side of the drone, one of the several Waveshare stereo cameras is placed for perception.

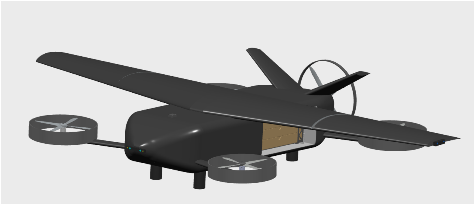


Figure 13.7: External layout configuration

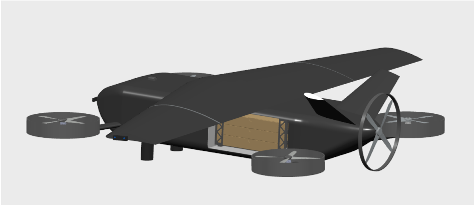


Figure 13.8: External layout configuration focusing on the aft of the fuselage

Figure 13.9 showcases the drone with its wings folded. They fold at an angle of 95° with respect to the fuselage, ensuring that the footprint of the drone is reduced during landing. In this model, the wing folding mechanism is not yet present, but it will be finalised in the complete detailed design of the BezORger.

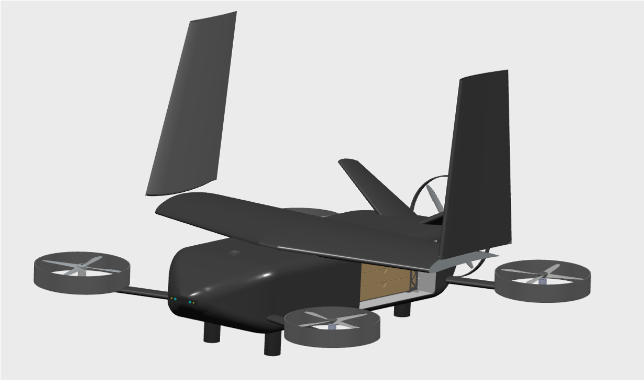


Figure 13.9: External layout showing folded wing

13.2.2. Internal Layout

This subsection describes the internal layout of the drone, displaying all the internal components in more detail. In Figure 13.10, a lateral perspective shows the cross-section of the drone. The bulk of the fuselage volume is taken up by both payload bays. Each pile of pizzas is supported by four rail supports on each corner, which connect to the insulation material (light grey). The nose and tail sections are reserved for the electronic components and batteries.

On the front, the small (blue) rectangle is the Time of Flight sensor (Grove VL53L0X). On its left lies the Raspberry Pi (green) with the NAVIO2 shield (grey) on top of it. In orange are the batteries, two of them placed on the front and the remaining two in the back.

Finally, Figure 13.11 displays a top view of the cut drone. It is possible to view the full size of the batteries and payload, as well as where the propellers connect to the fuselage.

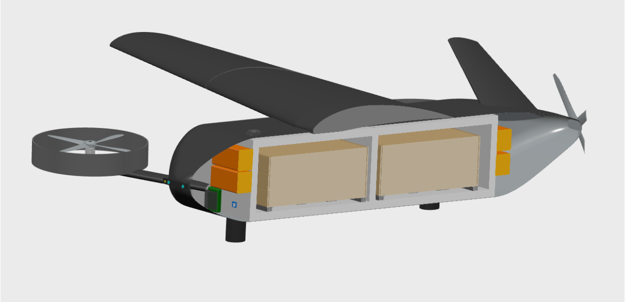


Figure 13.10: Internal layout of the front at an angle

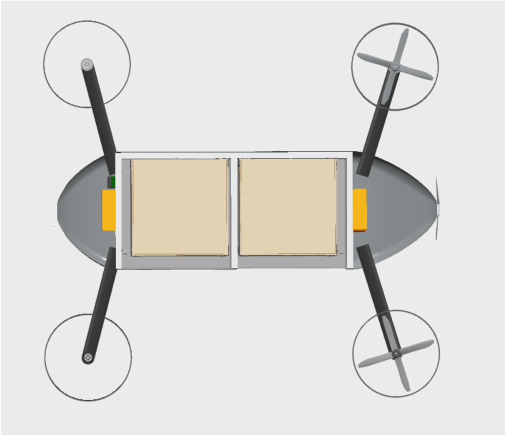


Figure 13.11: Internal layout from a top perspective

13.2.3. Electrical Architecture

The electrical components of BezORger were divided into high-voltage (HV) or low-voltage (LV) depending on their operating voltage (27 V and 5 V respectively). Based on function, the components were divided into four categories:

- **Power (HV):** Battery pack, battery management system, and power distribution board
- **Propulsion (HV):** Electronic Speed Controllers (ESCs) and motors
- **Controllers (LV):** Navio2 shield on Raspberry Pi
- **Sensors & antennae (LV):** All sensors and perception equipment along with the antenna(e)

Figure 13.12 shows the layout of each block along with an overview of how clusters of components interact with each other. The voltage regulator separates the HV and the LV sides of the electrical architecture. The dotted lines show transfer of information through electrical signals but not transmission of electrical power. For the sake of clarity, closed circuits have not been shown; the focus was to portray the flow of power from source to sink. In addition, every component shares a common electrical ground, provided through the wiring harness and Power Distribution Board (PDB). The drone frame is not used as a ground path.

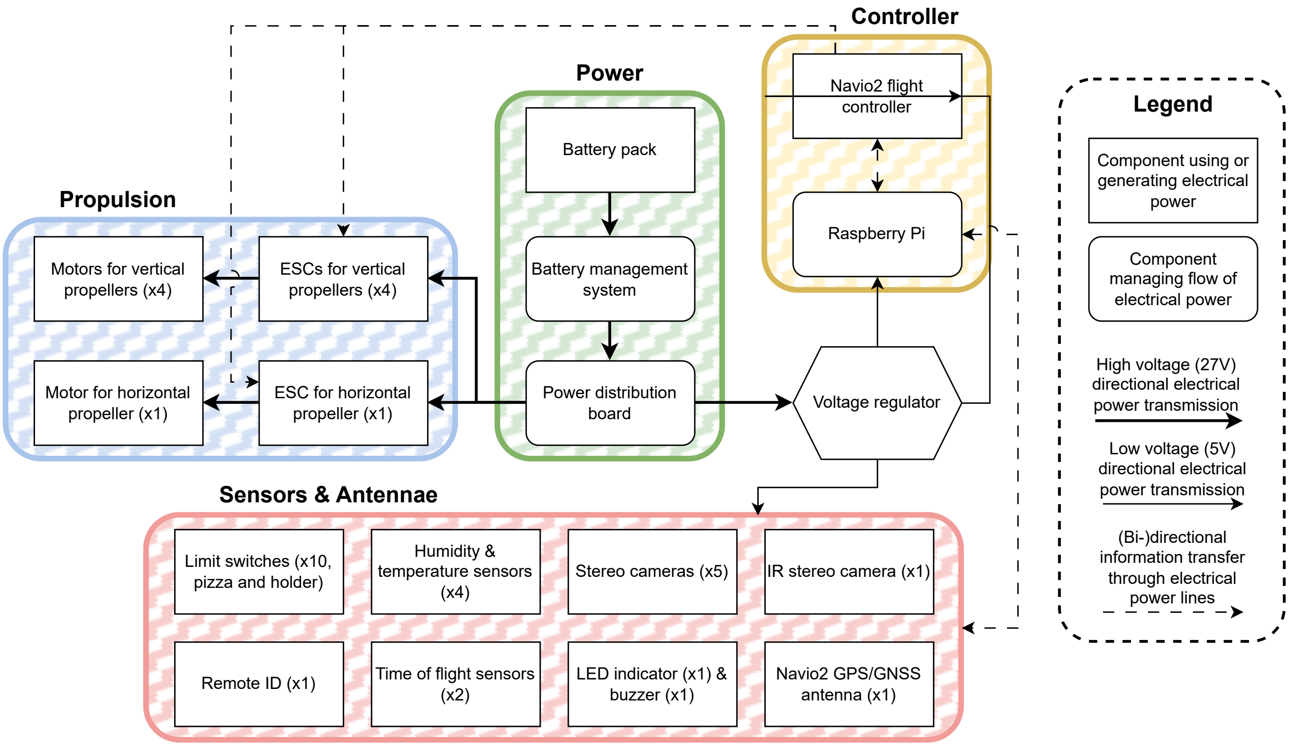

Controller
Navio2 flight controller
Raspberry Pi
Power
Battery pack
Battery management system
Power distribution board
Voltage regulator
Propulsion
Motors for vertical propellers (x4)
ESCs for vertical propellers (x4)
Motor for horizontal propeller (x1)
ESC for horizontal propeller (x1)
Sensors & Antennae
Limit switches (x10, pizza and holder)
Humidity & temperature sensors (x4)
Stereo cameras (x5)
IR stereo camera (x1)
Remote ID (x1)
Time of flight sensors (x2)
LED indicator (x1) & buzzer (x1)
Navio2 GPS/GNSS antenna (x1)
Legend
Component using or generating electrical power
Component managing flow of electrical power
High voltage (27V) directional electrical power transmission
Low voltage (5V) directional electrical power transmission
(Bi-)directional information transfer through electrical power lines

Figure 13.12: Electrical block diagram (grounding & neutral lines not shown)

13.3. Communication & Data Handling

The following section describes the way in which the DronebezORgd handles communication and data handling. Specifically, Subsection 13.3.1 describes the data handling of DronebezORgd and Subsection 13.3.2 depicts the interfaces between all hardware and software components on the BezORger and the ground station, and how they communicate.

13.3.1. Data Handling

During delivery, the drone will have to maintain data communication between internal and external sources in order to perform properly. This communication and data flow is described in Figure 13.13. First, the customers and restaurants communicate with the ground station through the business's app, which manages orders and restaurant updates, while the ground station sends updates on the status of the delivery. Next, the ground station sends to the main data receiver inside the flight controller of the drone the delivery data, such as the ideal path and mission planning, and if necessary, manual control instructions in case of emergencies. The drone returns to ground control with its position and telemetry data. From the external environment, a number of sensors measure data for 3D mapping, obstacle avoidance, and ground tracking, which are needed by the flight controller to determine the real flight path. The drone also communicates with the thermal sensors for the payload box in order to update the customer about the freshness of the pizza. Internally, the flight controller uses the obtained data, sends it to the computer interface for processing, then sends instructions to the propulsion system for the required RPM of each propeller for the path planning, and also sends instructions to activate the LED strips and Buzzer in case of a failure emergency, in order to alert the people around the drone.

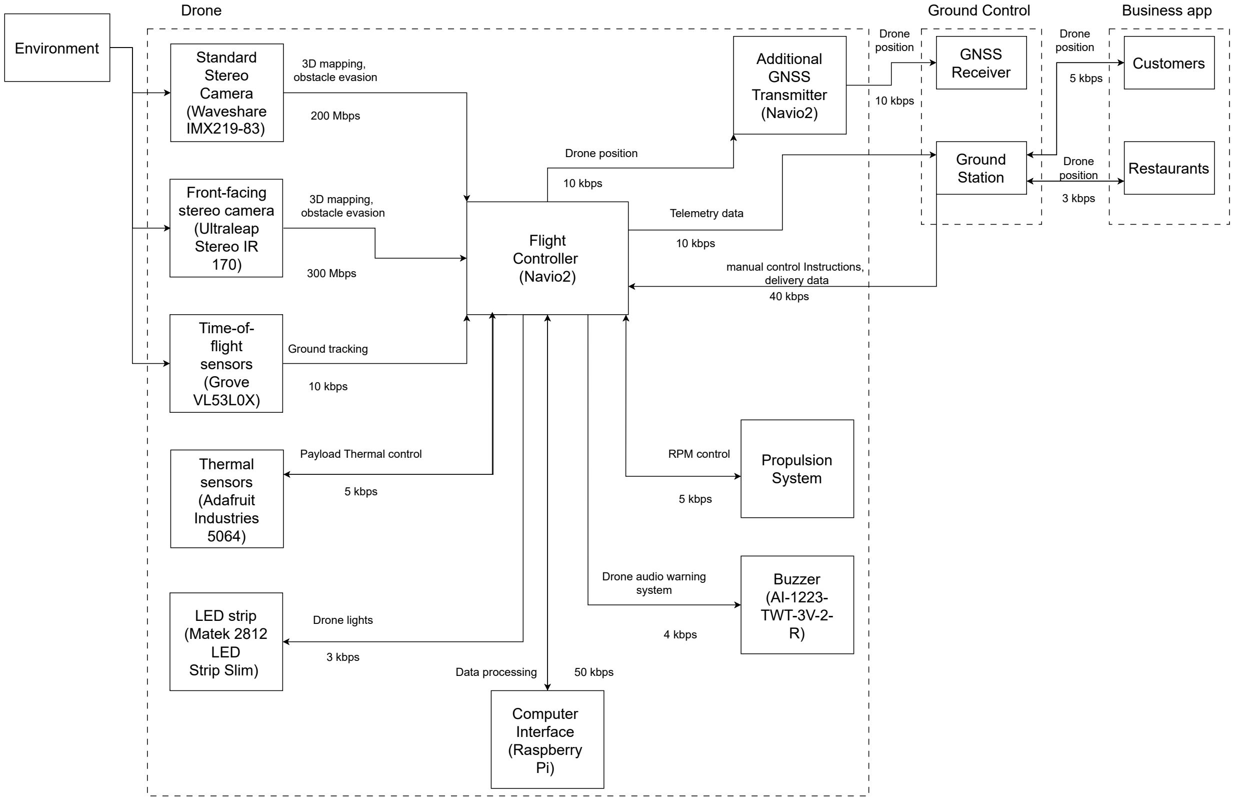

Environment
Drone
Ground Control
Business app
Standard Stereo Camera (Waveshare IMX219-83)
3D mapping, obstacle evasion
200 Mbps
Front-facing stereo camera (Ultraleap Stereo IR 170)
3D mapping, obstacle evasion
300 Mbps
Time-of-flight sensors (Grove VL53L0X)
Ground tracking
10 kbps
Thermal sensors (Adafruit Industries 5064)
Payload Thermal control
5 kbps
LED strip (Matek 2812 LED Strip Slim)
Drone lights
3 kbps
Flight Controller (Navio2)
Drone position
10 kbps
Additional GNSS Transmitter (Navio2)
Drone position
10 kbps
GNSS Receiver
Ground Station
Telemetry data
10 kbps
manual control Instructions, delivery data
40 kbps
Drone position
5 kbps
Customers
Drone position
3 kbps
Restaurants
RPM control
5 kbps
Propulsion System
Drone audio warning system
4 kbps
Buzzer (AI-1223-TWT-3V-2-R)
Data processing
50 kbps
Computer Interface (Raspberry Pi)

Figure 13.13: Data handling block diagram

13.3.2. Hardware-Software Interface

To get a better overview of the interaction between hardware and software, both onboard BezORger as well as with the ground station, a hardware-software block diagram was made, presented in Figure 13.14. The diagram helps visualise the interdependencies between different hardware and software components and presents an overall flow of communication both within the drone as well as with the ground control station (the depot).

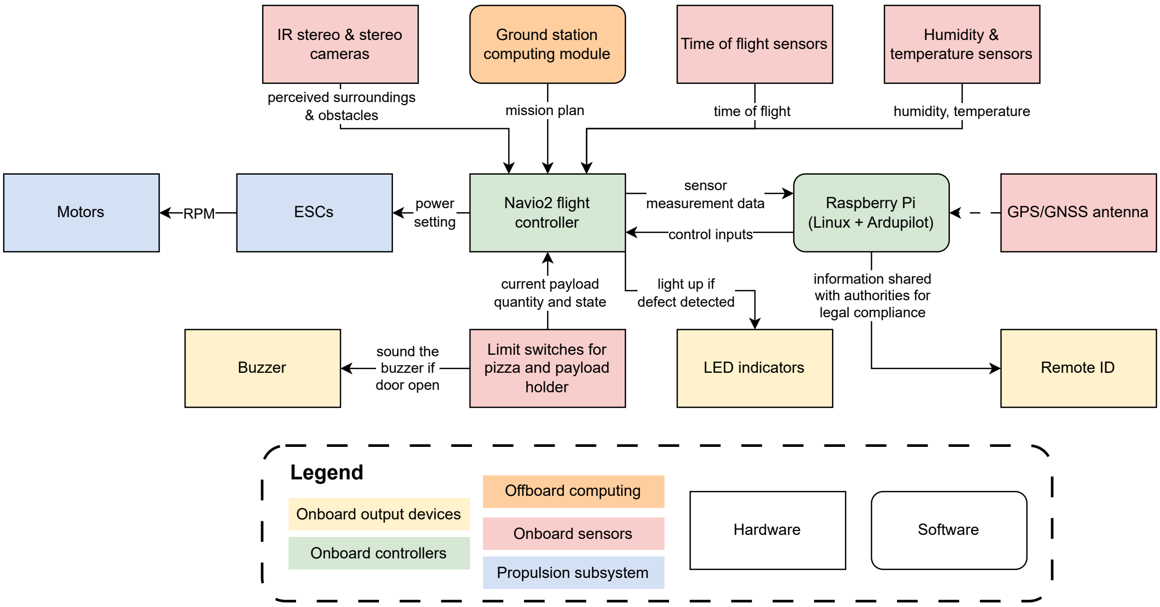

IR stereo & stereo cameras
Ground station computing module
Time of flight sensors
Humidity & temperature sensors
perceived surroundings & obstacles
mission plan
time of flight
humidity, temperature
Motors
RPM
ESCs
power setting
Navio2 flight controller
sensor measurement data
control inputs
Raspberry Pi (Linux + Ardupilot)
GPS/GNSS antenna
current payload quantity and state
light up if defect detected
information shared with authorities for legal compliance
Buzzer
sound the buzzer if door open
Limit switches for pizza and payload holder
LED indicators
Remote ID
Legend
Onboard output devices
Onboard controllers
Offboard computing
Onboard sensors
Propulsion subsystem
Hardware
Software

Figure 13.14: Hardware-software block diagram

14 Design Overview

The overview of the design of both the BezORger and the operations of DronebezORgd is presented in this chapter. A summary of the most important design parameters, as well as the mass, cost, and power breakdown, is presented in Section 14.1. Then, in Section 14.2, an overview of the compliance with requirements is shown, with justifications of all unverified requirements. Finally, Section 14.3 details the efforts made to make the design and operations sustainable.

14.1. Design Summary

A summary of the final design is presented in this section. Subsection 14.1.1 showcases the final design parameters collected from all subsystems. Then, Subsection 14.1.2 presents a mass breakdown and Subsection 14.1.3 presents a cost breakdown of all components of the drone. Finally, Subsection 14.1.4 presents a breakdown of the power consumption of the drone.

14.1.1. Final Design Parameters

After the design iteration, the parameters from all subsystems can be compiled and summarised. This compilation can be seen in Table 14.1

Table 14.1: Summary of aircraft parameters and components

	Parameter	**Values**	**Units**		**Parameter**	**Values**	**Units**
Main Parameters	MTOW	13.29	kg	Wing	Airfoil	NACA 6412	-
	OEW	10.06	kg		Root chord	0.63	m
	Max load factor	3.68	-		Tip chord	0.25	m
	Structural mass	5.16	kg		Dihedral	1.00	°
	Payload mass	3.00	kg		Sweep c/4	0.00	°
	Fuselage length	1.80	m		Wing area	1.33	m^2
	Avionics (sensor) mass	0.88	kg		Span	3.00	m
	Number of pizzas carried	3x2	-		MAC	0.47	m
	Cruise speed	14	m/s		X_{LEMAC}	0.92	m
V-Tail	Airfoil	NACA0012	-	Powertrain	Number of vertical propellers	4	-
	Root chord	0.27	m		Number of horizontal propellers	1	-
	Tip chord	0.14	m		Number of blades per propeller	4	-
	Dihedral	0.00	°		Blade airfoil	Clark Y	-

	V-tail angle	31.21	°		Vertical propeller diameter	0.40	m
	Tail area	0.17	m^2		Horizontal propeller diameter	0.44	m
	Span	0.407	m		Vertical duct diameter	0.43	m
	MAC	0.2035	m		Horizontal duct diameter	0.46	m
	X_{LEMAC}	1.45	m		Max revolution speed	628.32	rad/s
	Y_{LEMAC}	0.28	m		Max vertical thrust	56.4	N
Operations	Optimal depot latitude and longitude	[51.996446, 4.352530]	°		Max horizontal thrust	56.4	N
	Number of drones for max profit	15	-		Battery total capacity	0.47	kWh
	Number of drones for max ROI	10	-		Battery max voltage	30.8	V

14.1.2. Mass Breakdown

In the mass breakdown, the mass of each component and assembly of the drone is presented in Table 14.2.

Table 14.2: Mass breakdown

Component	Mass [kg]	Component	Mass [kg]
Battery	2.4	Motor	1.95
Propeller	0.25	Delivery mechanism	2.35
Folding mechanism	0.38	Landing gear	0.07
Sensors	0.30	Wing structure	1.80
Fuselage structure	0.56	Tail structure	0.39
Payload	3.0		

14.1.3. Cost breakdown

In the cost breakdown, each component of the drone has been attributed a monetary value. The components have been ordered into the 4 main departments they come from. The price showcased is the cost of the component per drone, while also adding how much manufacturing costs add up on average for manufacturing the drone. Both the components and assembly costs are going to be high at the beginning of production, but are expected to go down as production scale increases and the business starts purchasing components in bulk.

Table 14.3: Cost breakdown

	Component	Cost [€]		Component	Cost [€]
Structures	Wing	200	Propulsion	Batteries	824
	Fuselage	150		Motors	250
	Landing gear	50		Propellers	100
	Tail	50			
Avionics	Flight controller	312	Payload	Wingspan reduction mechanism	78

	Perception sensors	600		Delivery mechanism	126
Misc.	Manufacturing costs	2500		Thermal insulation structure	70

14.1.4. Power breakdown

The power breakdown explores the power consumption of the drone during its mission. Through analysis, it has been observed that the propulsion system uses more than 90% of the full available power, while the rest is distributed between the internal data handling and communication systems inside the drone.

14.2. Compliance

Once the final design is established, the compliance of each requirement can be assessed to ensure DronebezORgd's adherence to design goals. The compliance of each system requirement can be seen in Table 14.4. The requirements which have not yet been tested or implemented, as well as requirements which are found to be non-compliant, will be explained below.

Table 14.4: Requirement Compliance Overview (Compliance: compliance level, ✓: compliant, ×: not compliant, □: not yet tested, ⊘: not yet implemented)

Requirement	Compliance	Requirement	Compliance	Requirement	Compliance
RQ-01-01-01	✓	RQ-01-01-02	✓	RQ-01-01-03	□
RQ-01-02-01	✓	RQ-01-02-02	✓	RQ-01-02-03	✓
RQ-01-02-04	✓	RQ-01-02-05	✓	RQ-01-03-01	✓
RQ-01-03-02	✓	RQ-01-03-03	✓	RQ-01-03-04	✓
RQ-01-04-02	✓	RQ-01-04-03	✓	RQ-01-04-06	✓
RQ-01-04-07	✓	RQ-01-04-08	✓	RQ-01-04-09	✓
RQ-01-05-01	✓	RQ-01-05-02	✓	RQ-01-05-03	✓
RQ-01-05-04	✓	RQ-01-05-06	✓	RQ-01-05-08	✓
RQ-01-05-10	✓	RQ-01-05-11	✓	RQ-01-05-12	✓
RQ-01-05-13	✓	RQ-01-05-15	✓	RQ-02-01-01	⊘
RQ-02-01-03	✓	RQ-02-01-04	✓	RQ-02-02-01	✓
RQ-02-02-02	✓	RQ-02-02-03	✓	RQ-02-03-02	✓
RQ-02-03-03	✓	RQ-02-03-04	✓	RQ-03-01-01	✓
RQ-03-01-02	✓	RQ-03-02-01	✓	RQ-03-02-02	✓
RQ-03-03-01	✓	RQ-03-03-02	□	RQ-03-04-01	□
RQ-03-04-02	□	RQ-04-01-01	✓	RQ-04-01-02	✓
RQ-04-02-01	✓	RQ-06-01-01	✓	RQ-06-01-02	✓
RQ-06-02-01	✓	RQ-06-02-02	✓	RQ-07-01-02	✓
RQ-08-01-01	⊘	RQ-08-01-02	⊘	RQ-08-02-01	⊘
RQ-09-01-01	✓	RQ-09-01-03	✓	RQ-09-02-01	✓
RQ-09-02-02	✓	RQ-09-02-03	✓	RQ-09-02-04	⊘
RQ-09-03-01	⊘	RQ-09-03-02	✓	RQ-10-01-01	✓
RQ-10-02-02	✓	RQ-10-03-01	✓	RQ-10-03-02	✓
RQ-12-03-01	✓	RQ-13-01-01	✓	RQ-13-01-02	✓
RQ-13-01-03	✓	RQ-13-02-01	✓	RQ-13-02-02	✓
RQ-13-02-03	✓	RQ-14-01-01	✓	RQ-14-01-02	✓
RQ-14-02-01	✓	RQ-14-02-02	✓	RQ-14-03-01	✓
RQ-14-03-02	✓	RQ-14-04-01	✓	RQ-14-04-02	✓
RQ-14-04-03	✓				

Untested Requirements

- **Every requirement is verified:** RQ-01-01-03
  This cannot be compliant until all other requirements are implemented and verified.
- **Accident rates:** RQ-03-04-01, RQ-03-04-02
  Long-term accident rates cannot be verified with the limited verification time and budget.
- **Delay rates:** RQ-03-03-02
  Long-term delay rates can also not be verified until operations begin and sufficient statistics can be collected.

Unimplemented Requirements

- **Path planning updates with weather changes:** RQ-02-01-01
  Due to time constraints, real-time path planning updates have not been implemented.
- **User interface requirements:** RQ-08-01-01, RQ-08-01-02, RQ-08-02-01, RQ-09-02-04
  The user interface has not yet been designed and therefore cannot be verified.
- **No drone grouping:** RQ-09-03-01
  This requirement was originally intended to increase the perceived safety of the system. The obstacle avoidance systems, however, are deemed to be sufficient to allow for safe operations, so this was not explicitly implemented.

Non-compliant Requirements

While there remain requirements which are not verified or implemented, there are currently no non-compliant requirements. DronebezORgd is therefore seen as designed to specifications.

14.3. Sustainable Development Strategy

For a startup like *DronebezORgd*, introducing a novel form of last-mile food delivery with the BezORger drone, sustainability is not just a societal expectation but also a business-critical concern. Three main aspects of sustainability were considered throughout the project:

- **Environmental** – focusing on energy efficiency, material circularity, and minimisation of waste;
- **Social** – emphasising noise impact, public perception, and operational safety;
- **Financial** – ensuring the design is cost-effective, reaches breakeven, and supports long-term economic viability.

These sustainability dimensions are relevant not only for the technical design of the drone, but also for the overall strategy and business model of *DronebezORgd*. Therefore, sustainability has been embedded across the entire lifecycle of the BezORger system. This includes all six lifecycle phases defined earlier in the functional analysis: **Design, Manufacturing, Assembly, Integration and Testing (AIT), Operations, Maintenance and Repair (MRO), and End-of-Life (EOL)**.

Sustainability was a key factor even during early concept selection, where it contributed significantly to the decision to adopt a fixed-wing eVTOL configuration (Section 4.1). Concrete design decisions within each subsystem were made considering the sustainability over the six lifecycle phases of the drone BezORger and the DronebezORgd business. Several requirements were also phrased regarding the three aspects of sustainability in Section 2.3, including, for example, noise limits, material recyclability, and financial break-even point.

Table 14.5 summarises the sustainability contributions of key design decisions across different subsystems, elaborated in their respective chapters. Table 14.6 provides a complementary overview of sustainability considerations structured by lifecycle phase, illustrating how sustainability was considered comprehensively from Design to End-of-Life.

Table 14.5: Subsystem-level sustainability considerations

Department	Key design decisions	Sustainability impact
Aerodynamics	Airframe designed for high lift-to-drag ratio	Reduced cruise power requirement lowers energy usage
Control & Stability	Use of tails for passive stability; statically stable layout; safe landing possible with one engine; vertical thrusters only used for maneuvering (no control surfaces)	Less energy-intensive control; increased operational safety and service life; lower material usage; simpler assembly; reduced cruise-phase energy waste
Propulsion & Powertrain	Quasi-optimised propeller design; circular ducts around propellers; off-the-shelf motor and battery selection	Reduced power consumption by propulsion subsystem; reduced noise emissions; increased safety, especially during landing and take-off in public spaces; increased propulsive efficiency due to ducts; economically sustainable manufacturing due to off-the-shelf components
Performance & Design Integration	Wing-powered cruise, transition phase energy modelling; Minimize mass in design iteration	Reduce energy consumption; Avoid over-designed drone
Payload Handling & Sensors	Passive insulation instead of heaters; separate compartments; off-the-shelf actuators; folding wings; 3D-printed ABS structure; use of commercial sensors	Lower energy use; improved trust and user experience; recyclable and easily manufactured parts; minimal waste; avoids custom production chains; improves reparability and sourcing
Structure & Materials	Modular design; symmetric wing folding parts; no custom moulds required	Simplified repairability; lower spare inventory; extended drone lifespan; scalable to future payload variants
Operations	Mission/path planning minimizes flight time and hover duration; avoids sensitive/noisy areas; optimised drone and depot use	Lower energy and noise footprint; minimal unused capacity; improved social acceptance and operational cost-efficiency
Business & HR	Union-compliant wages; rejection of low-efficiency orders	Ensures fair labor practices; supports long-term economic sustainability while reducing energy and material waste

Table 14.6: Sustainability across the system lifecycle

Phase	Environmental	Social	Financial
Design	High lift-to-drag ratio; modularity; off-the-shelf components; energy-saving tail configuration	Fail-safe layout improves public trust; intuitive payload interface; reduced material use via design	Design-for-reuse; simplified parts; minimized actuator count; off-the-shelf availability
Manufacturing	Additive manufacturing (FDM) minimizes waste; recyclable ABS; no special tooling	Wing folding reduces physical footprint in shared spaces	Low upfront cost due to no moulds; identical L/R parts reduce inventory
Assembly, Integration & Testing	Efficient, minimal material use; no redundant electronics needed due to passive mechanisms	Manual payload system is familiar and trusted; wing locks improve safety perception	Quick integration reduces labour/testing time; modular layout simplifies QA
Operations	Optimised routing minimises energy; minimal hover time; no active payload heating	Noise-sensitive flight paths; avoids schools, hospitals, dense areas; reduces road traffic	Lower energy = lower cost per delivery; drone allocation avoids overcapacity; optimised depot location

Maintenance & Repair (MRO)	Recyclable and replaceable parts; no toxic substances; minimal spares needed	Fewer failure points = improved safety; transparent parts reduce tampering risk	Local 3D-printing of spares; shared parts across drones; reduced logistics chain
End-of-Life	Recyclable plastics; component reusability; no hazardous waste	Safe decommissioning; no public nuisance or risk on disposal	Extended lifespan by modular upgrades; no special disposal infrastructure required

15 Manufacturing & Maintenance

This chapter outlines the manufacturing strategy and operational reliability framework for the BezORger delivery drone. Section 15.1 describes the complete production workflow, from component manufacturing to assembly processes, while also addressing scalable manufacturing approaches for potential future expansion due to the successful outcome of the project. The analysis extends to quality control measures and production optimisation to ensure commercial viability.

On the other hand, Section 15.2 describes how RAMS (Reliability, Availability, Maintainability, and Safety) principles are incorporated throughout the product lifecycle to meet operational requirements. This systematic approach verifies critical performance parameters, from structural integrity to maintenance protocols, creating a framework that supports the drone's transition from prototype testing to certified mass production while ensuring operational safety and reliability.

15.1. Production Plan

This section outlines the manufacturing strategy for the BezORger delivery drone, explaining the processes required from sketching and modelling the drone to scaling production. The plan establishes a structured workflow that identifies both sequential and parallel production stages, optimising efficiency while maintaining quality standards.

The production plan is separated into three main parts: firstly, Subsection 15.1.1 highlights the process of obtaining all subcomponents of the drone, including make-or-buy decisions for critical components. Furthermore, Subsection 15.1.2 describes the assembling processes for each of the structural subgroups, as well as the integration of all subgroups to arrive at the final drone as the product. Finally, Subsection 15.1.3 explains the possible ways of scaling the current single drone production plan to fulfil future requirements, given a successful outcome of the project. This structured approach provides manufacturing flexibility while ensuring consistent output quality across all production stages. Figure 15.1 shows the high-level flowchart of production for a single drone.

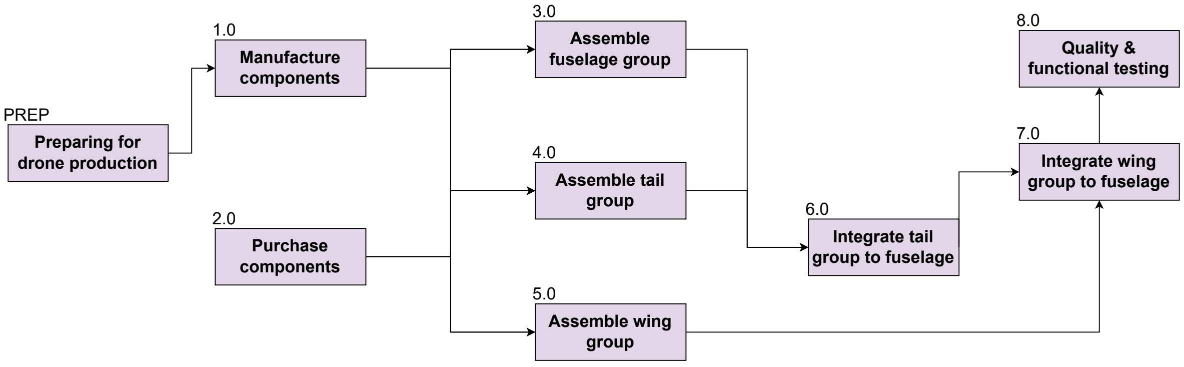


Figure 15.1: Overall production flowchart

15.1.1. Component Manufacturing & Purchasing

The production process begins with preparation activities to decrease the time needed to recheck items in the following phases, as seen in Figure 15.2. This phase involves making CAD models and technical drawings, verifying the complete Bill of Materials, and calibrating all necessary production equipment. Quality control protocols are established while material suppliers are selected after comparison. The phase concludes with ordering the needed material for production, creating the foundation for subsequent manufacturing stages.

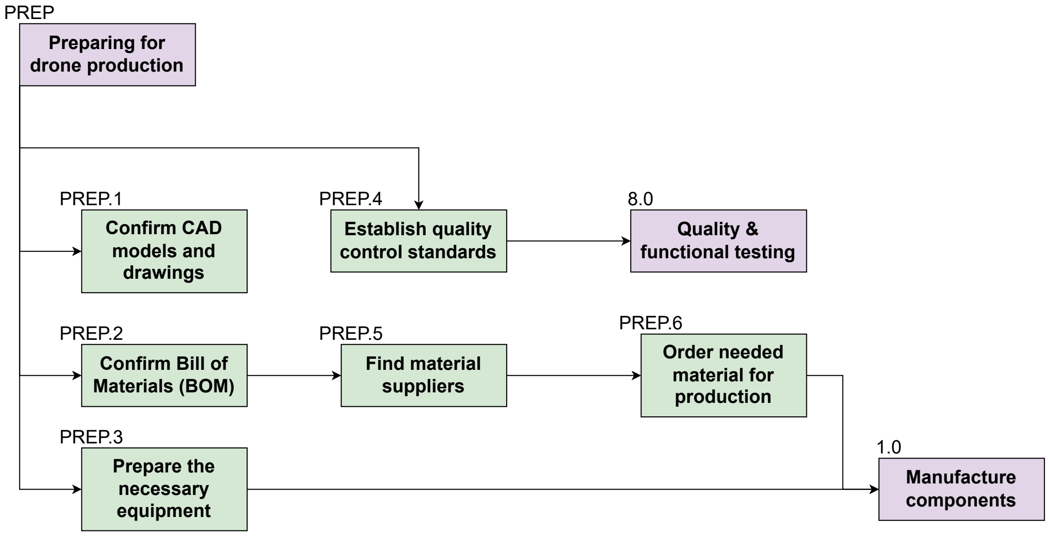

PREP
Preparing for drone production
PREP.1
Confirm CAD models and drawings
PREP.2
Confirm Bill of Materials (BOM)
PREP.3
Prepare the necessary equipment
PREP.4
Establish quality control standards
PREP.5
Find material suppliers
PREP.6
Order needed material for production
8.0
Quality & functional testing
1.0
Manufacture components

Figure 15.2: Preparation phase flowchart

The component manufacturing phase covers the fabrication of all structural airframe components through different production lines, as seen in Figure 15.3. The fuselage, wings, and tail sections are manufactured as primary load-bearing structures separately, while flight-critical elements like propeller blades and landing gear shall undergo precision machining. Simultaneously, the payload delivery mechanism is fabricated to meet the dimensional requirements since it has to fit inside the fuselage. All components progress to subgroup assembly after passing individual quality inspections.

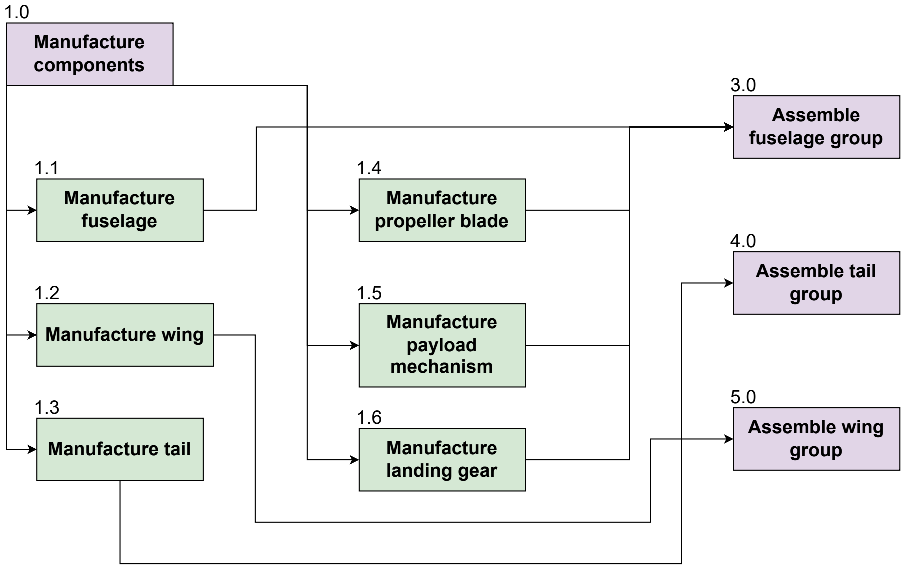


Figure 15.3: Manufacturing component flowchart

The component purchasing phase is presented in Figure 15.4. Some avionics and propulsion subcomponents are sourced from qualified suppliers to provide performance consistency. The purchasing list includes navigation components, power systems, and control hardware. Lighting systems for operational compliance are also purchased items. All components undergo incoming quality assurance before being released for fuselage, tail, and wing subgroup integration.

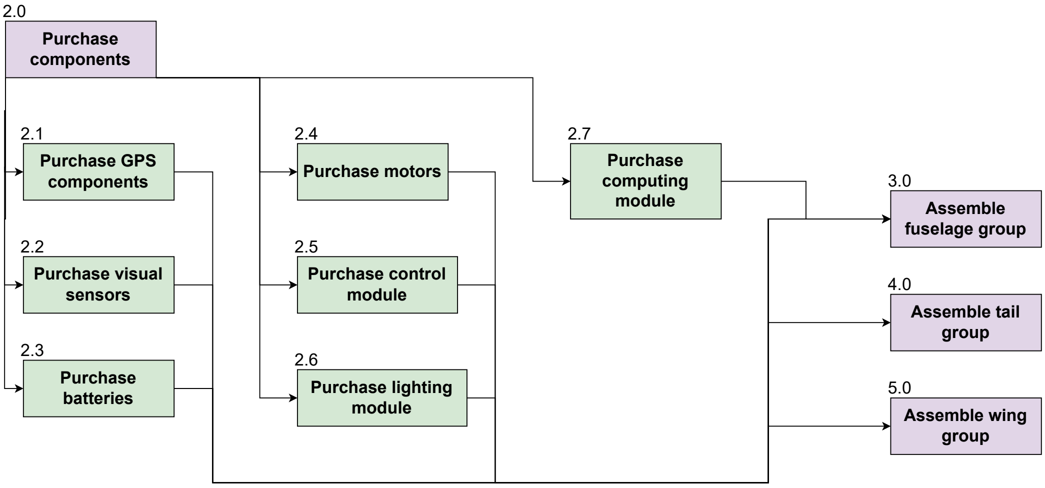


Figure 15.4: Purchasing component flowchart

15.1.2. Assembly Process

The fuselage assembly presented in Figure 15.5 begins with primary structure integration before installing propulsion systems, which consist of vertical and horizontal propellers, followed by the payload delivery mechanism. Concurrently, the tail group shown in Figure 15.6 was constructed through sequential assembly of root/tip components, aluminium skin panels, and structural members, culminating in stability verification. Simultaneously, wing assembly in Figure 15.7 progresses through rib assembly, skin assembly, and spar integration, with particular attention to the folding mechanism's reliability. All subgroups undergo structural validation before final integration.

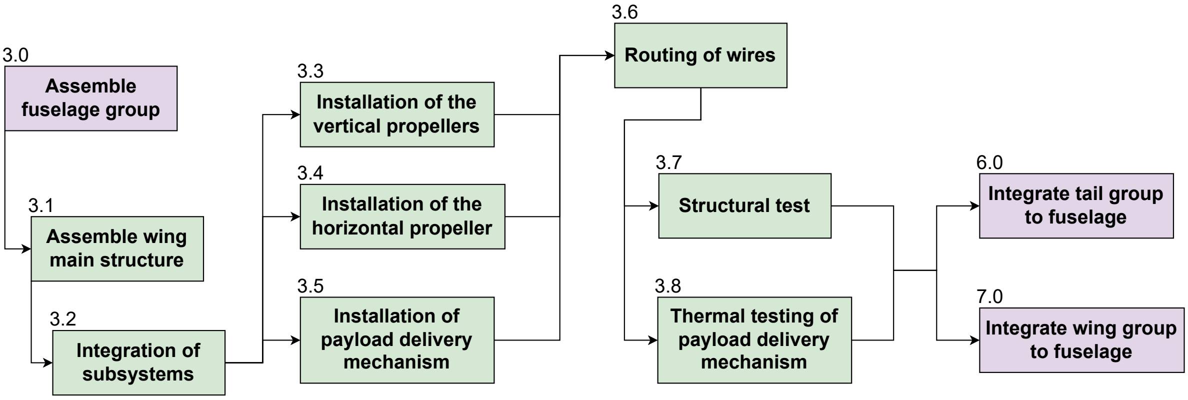

3.0
Assemble fuselage group
3.1
Assemble wing main structure
3.2
Integration of subsystems
3.3
Installation of the vertical propellers
3.4
Installation of the horizontal propeller
3.5
Installation of payload delivery mechanism
3.6
Routing of wires
3.7
Structural test
3.8
Thermal testing of payload delivery mechanism
6.0
Integrate tail group to fuselage
7.0
Integrate wing group to fuselage

Figure 15.5: Fuselage assembly flowchart

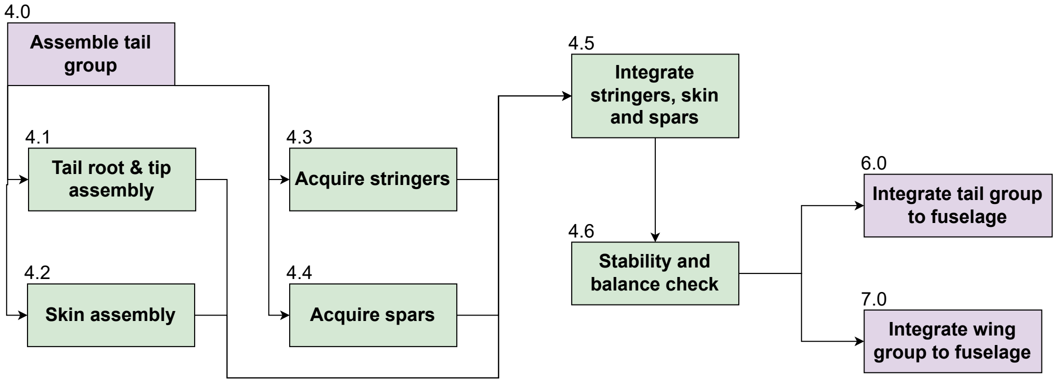

4.0
Assemble tail group
4.1
Tail root & tip assembly
4.2
Skin assembly
4.3
Acquire stringers
4.4
Acquire spars
4.5
Integrate stringers, skin and spars
4.6
Stability and balance check
6.0
Integrate tail group to fuselage
7.0
Integrate wing group to fuselage

Figure 15.6: Tail assembly flowchart

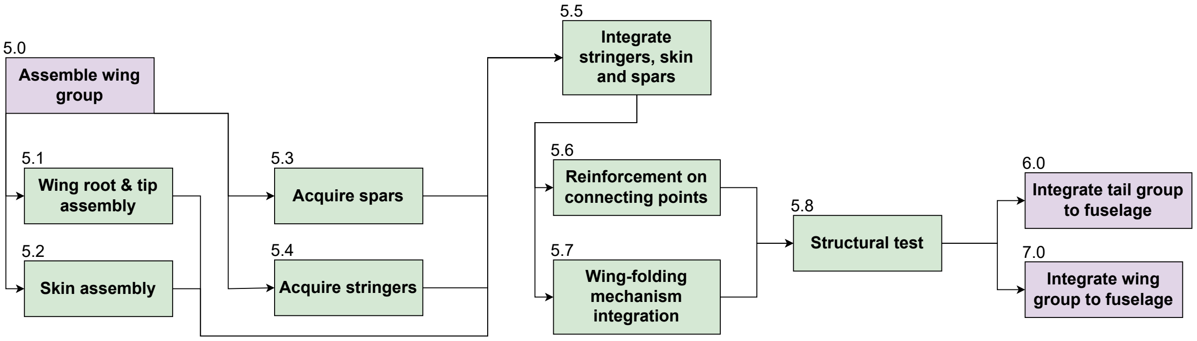

5.0
Assemble wing group
5.1
Wing root & tip assembly
5.2
Skin assembly
5.3
Acquire spars
5.4
Acquire stringers
5.5
Integrate stringers, skin and spars
5.6
Reinforcement on connecting points
5.7
Wing-folding mechanism integration
5.8
Structural test
6.0
Integrate tail group to fuselage
7.0
Integrate wing group to fuselage

Figure 15.7: Wing assembly flowchart

After the parallel assembly of the three groups, the tail group is connected with the fuselage, as demonstrated in Figure 15.8, starting by aligning the two structural components and attaching the fasteners, followed by dynamic vibration and static load testing. Next, wing integration, as depicted in Figure 15.9, employs a similar procedure, with an extra step of checking the wiring logistics due to the presence of subsystems, symmetry tests, folding mechanism tests, and structural stability checks are needed to make sure no failure is present. Both integration phases feed into comprehensive quality testing, confirming all systems meet operational specifications before final certification. The parallel integration approach reduces assembly time while maintaining certain tolerances.

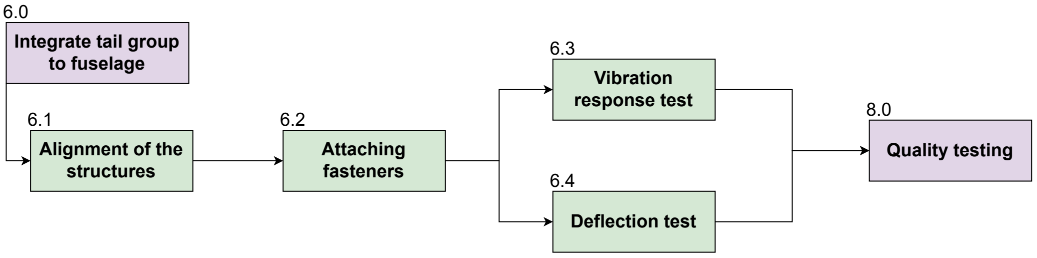


Figure 15.8: Tail-fuselage integration flowchart

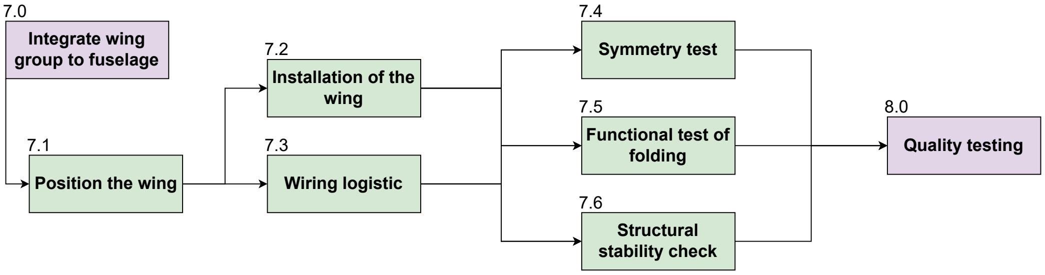


Figure 15.9: Wing-fuselage integration flowchart

Finally, quality control, as depicted in Figure 15.10, assesses the assembled drone from operational simulations. Vibration testing replicates flight loads, while subsystem checks verify avionics response times. An automated mission test executes takeoff/landing cycles with payload drops. Final sign-off requires no critical failures on all electrical and functional criteria before the drone is deployed for the mission.

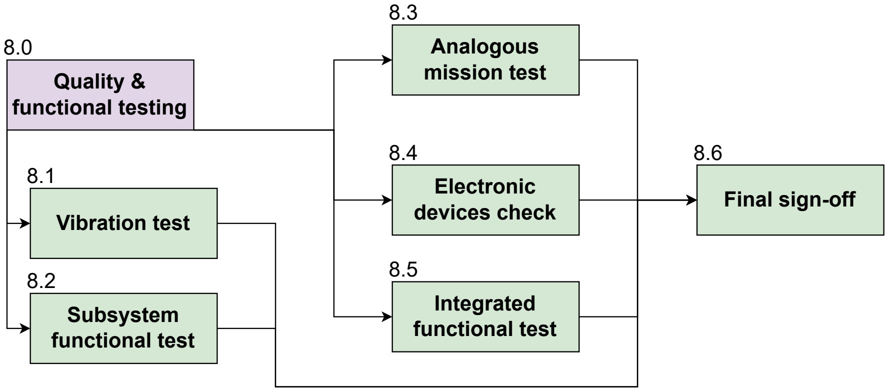


Figure 15.10: Quality control flowchart

15.1.3. Scaling Production

The BezORger's current design enables efficient scaling through parallel production lines and standardised components. Key subsystems will transition to dedicated workstations, supported by a continuous, steady supply chain of primary material and automated quality checks to maintain consistency as volume grows.

Regional depots will stock critical spare parts for rapid maintenance, while phased deployment prioritises urban hubs first. Workforce training and logistics networks will scale along with the increase in production, guaranteeing reliability standards are maintained during expansion. Feedback from early operations will guide iterative refinements to both manufacturing and delivery processes.

15.2. RAMS Assessment

The RAMS framework, consisting of reliability, availability, maintainability, and safety, has been applied to evaluate the BezORger drone's operational robustness and readiness for real-world deployment. This section outlines the relevant design features and strategies associated with each RAMS component.

Firstly, the reliability features of the drone shall be discussed in detail in Subsection 15.2.1, followed by the description of the availability of the drone under the given mission profile operating environment in Subsection 15.2.2. Moreover, the frequency, as well as the procedure for maintaining the drone before and after daily operation, shall be discussed in Subsection 15.2.3, eventually arriving at the explanation of the significance and importance of safety features in Subsection 15.2.4.

15.2.1. Reliability

The reliability feature in RAMS assessment refers to the ability to perform a specific function, focusing on maximising the undisturbed operational duration. A clear usage of this philosophy in this drone design is the design for redundancy for all subsystems. For example, the propulsion system is fault-tolerant; in the event of a motor failure, remaining units can sustain controlled flight and return to the depot. Similarly, the electronic equipment onboard has one back-up sensor from each functional group present so that operational continuity is maintained, preventing single-point failures from compromising mission integrity.

Beyond redundancy, the drone demonstrates structural durability. The structural design employs proven aerospace materials and geometries optimised for operational loads. Critical electronics are housed in sealed, vibration-isolated compartments, reducing risk from moisture, impacts, and mechanical shock. Self-check routines monitor overall system health, providing early warnings for fault detection.

This comprehensive approach to reliability guarantees the drone can consistently complete its delivery missions without critical performance degradation.

15.2.2. Availability

The availability of the BezORger drone is ensured through quick-swap components and operational planning. Modular battery and payload systems reduce turnaround times. Depot-based storage of spare parts and standardised components supports efficient parallel maintenance. The drone is weather-resistant for light precipitation, allowing extended operability. Predictive maintenance is planned based on operational cycles and system diagnostics, with technician training covering routine inspections and fault response to maximise uptime.

15.2.3. Maintainability

Maintainability has guided material selection, manufacturing, and assembly. The design supports rapid part replacement using interchangeable components and depot-stocked spares. Maintenance protocols prioritise minimal downtime: damaged components are replaced immediately and repaired off-line. Standardisation simplifies logistics and enables consistent quality control, facilitating high-frequency operations.

15.2.4. Safety

Lastly, safety is incorporated through redundant systems and predefined fail-safe responses. Critical subsystems, including the wing-folding and payload mechanisms, have safety clearances and emergency protocols. These include return-to-base functions, supervised descent, and mission abort logic. Structural features, such as shielding around vertical rotors, reduce user risk. Obstacle detection via sensors and cameras, combined with conservative routing to avoid residential areas, enhances operational and environmental safety. Risk management procedures, supervisory oversight, and system-level testing ensure compliance with safety requirements across all mission phases.

16 Future Outlook

DronebezORgd envisions the conceptual subsystem design performed as part of the Design Synthesis Exercise becoming a viable business model in the near future. The potential for improvement in last-mile delivery in urban environments is vast, and DronebezORgd aims to demonstrate that, one BezORger at a time. As such, this chapter outlines a detailed and structured plan to make this project a reality, covering an overview of the project development in Section 16.1 and a detailed breakdown and schedule in Section 16.2.

16.1. Project Development Overview

The project development plan covers the first four of the six identified phases of the project, as the last two correspond to a single drone and not to the overall business plan.

- **Design:** The design phase starts with the DSE and includes a detailed subsystem design as the next step, including rigorous V&V procedures. It concludes when a prototype has been designed and tested successfully.
- **Manufacturing:** Manufacturing the drone fleet involves contacting potential facilities, procuring materials and equipment, hiring and training personnel, producing the parts and sub-assemblies, and performing documentation and quality control.
- **Assembly, Integration, & Testing (AIT):** The AIT phase involves integrating the components and sub-assemblies produced and performing full-system tests, demonstrating a feasible and viable design.
- **Operations:** The operations of the business entail acquiring capital, setting up a workspace, ensuring compliance with legal regulations, branding and public outreach, recruitment of personnel, and ensuring everything runs smoothly. In the far future, scaling up the business operations can be explored.
- **Maintenance, Repair, & Overhaul (MRO):** This phase corresponds to the MRO of a single drone; as such, it is included in the Operations phase for the purposes of planning the project development.
- **End-of-Life (EOL):** EOL, again, corresponds to the proper disposal and recycling of a single drone; the business operations continue as usual.

To better structure the plan for building up a viable business based on DronebezORgd, the first four phases were then divided into sub-phases as shown in Figure 16.1. This breakdown provides a clearer overview of the sequential and parallel activities required to develop the system and the business around it.

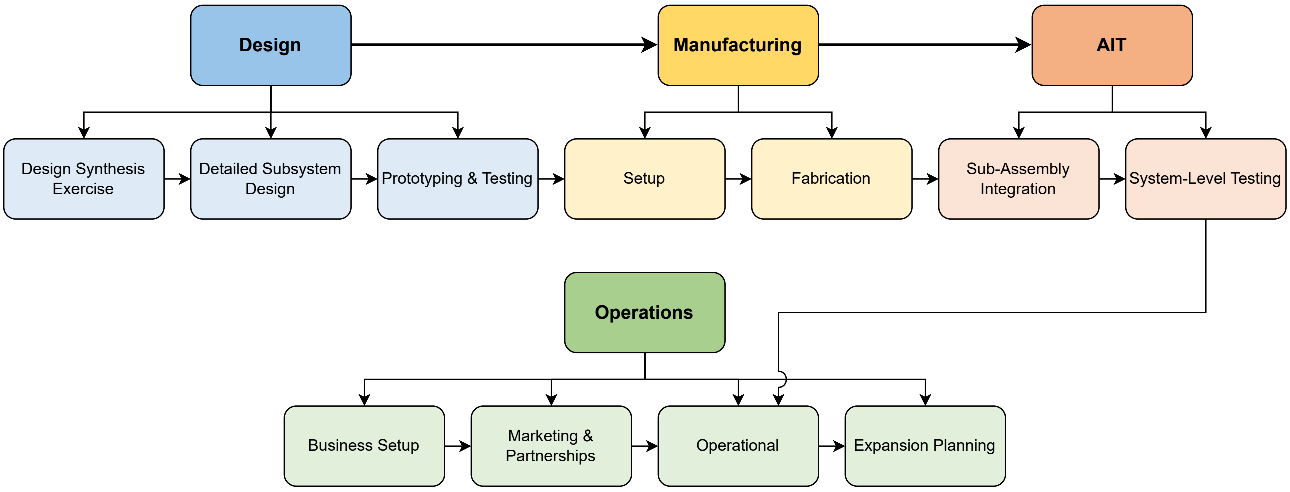


Figure 16.1: Project development sub-phases

It should be noted that while the Design, Manufacturing, and AIT phases are quite sequential, the tasks comprising the Operations phase will run in parallel to these phases. While the former three set up the technical side of things, the latter sets up the organisational aspect.

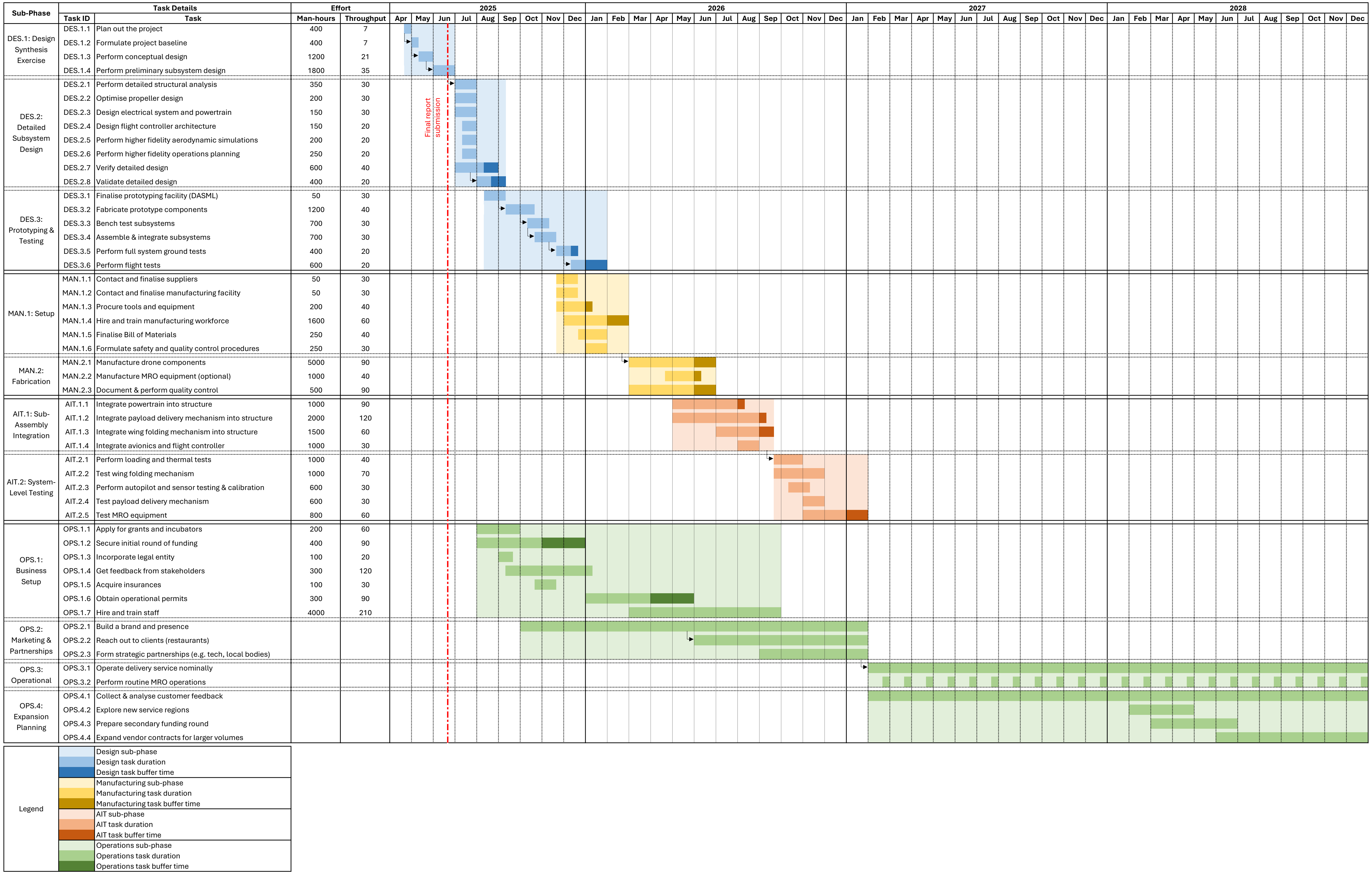
16.2. Project Development Schedule

With a structured division of phases and sub-phases of the project, sub-tasks were identified for each sub-phase. The effort needed in man-hours as well as the throughput time in calendar hours were estimated for each, based on which, a Gantt chart was made to draft a big-picture timeline for the project development. The Gantt chart, shown in Figure 16.2, also shows dependencies between tasks.

For the sake of practicality, the granularity of the identified tasks has been kept quite large (in the order of months). Furthermore, estimates of the effort and throughput time have not been made for phases that involve continuous and potentially never-ending tasks. Furthermore, buffer time has been included for tasks that may involve significant delays and/or require a partial redesign.

Some notable milestones that can be seen in the Gantt chart are as follows:

- **Oct 2025:** Acquire initial funding
- **Dec 2025:** Working and tested BezORger prototype
- **Mar 2026:** Start manufacturing components & sub-assemblies
- **Jun 2026:** Finish manufacturing components & sub-assemblies, start assembly & integration
- **Aug 2026:** Finish assembly & integration, start full system-testing
- **Jan 2027:** Finish full system-testing, ready for initial business operations



Sub-Phase	Task Details		Effort	
	Task ID	Task	Man-hours	Throughput
DES.1: Design Synthesis Exercise	DES.1.1	Plan out the project	400	7
	DES.1.2	Formulate project baseline	400	7
	DES.1.3	Perform conceptual design	1200	21
	DES.1.4	Perform preliminary subsystem design	1800	35
DES.2: Detailed Subsystem Design	DES.2.1	Perform detailed structural analysis	350	30
	DES.2.2	Optimise propeller design	200	30
	DES.2.3	Design electrical system and powertrain	150	30
	DES.2.4	Design flight controller architecture	150	20
	DES.2.5	Perform higher fidelity aerodynamic simulations	200	20
	DES.2.6	Perform higher fidelity operations planning	250	20
	DES.2.7	Verify detailed design	600	40
	DES.2.8	Validate detailed design	400	20
DES.3: Prototyping & Testing	DES.3.1	Finalise prototyping facility (DASML)	50	30
	DES.3.2	Fabricate prototype components	1200	40
	DES.3.3	Bench test subsystems	700	30
	DES.3.4	Assemble & integrate subsystems	700	30
	DES.3.5	Perform full system ground tests	400	20
	DES.3.6	Perform flight tests	600	20
MAN.1: Setup	MAN.1.1	Contact and finalise suppliers	50	30
	MAN.1.2	Contact and finalise manufacturing facility	50	30
	MAN.1.3	Procure tools and equipment	200	40
	MAN.1.4	Hire and train manufacturing workforce	1600	60
	MAN.1.5	Finalise Bill of Materials	250	40
	MAN.1.6	Formulate safety and quality control procedures	250	30
MAN.2: Fabrication	MAN.2.1	Manufacture drone components	5000	90
	MAN.2.2	Manufacture MRO equipment (optional)	1000	40
	MAN.2.3	Document & perform quality control	500	90
AIT.1: Sub-Assembly Integration	AIT.1.1	Integrate powertrain into structure	1000	90
	AIT.1.2	Integrate payload delivery mechanism into structure	2000	120
	AIT.1.3	Integrate wing folding mechanism into structure	1500	60
	AIT.1.4	Integrate avionics and flight controller	1000	30
AIT.2: System-Level Testing	AIT.2.1	Perform loading and thermal tests	1000	40
	AIT.2.2	Test wing folding mechanism	1000	70
	AIT.2.3	Perform autopilot and sensor testing & calibration	600	30
	AIT.2.4	Test payload delivery mechanism	600	30
	AIT.2.5	Test MRO equipment	800	60
OPS.1: Business Setup	OPS.1.1	Apply for grants and incubators	200	60
	OPS.1.2	Secure initial round of funding	400	90
	OPS.1.3	Incorporate legal entity	100	20
	OPS.1.4	Get feedback from stakeholders	300	120
	OPS.1.5	Acquire insurances	100	30
	OPS.1.6	Obtain operational permits	300	90
	OPS.1.7	Hire and train staff	4000	210
OPS.2: Marketing & Partnerships	OPS.2.1	Build a brand and presence		
	OPS.2.2	Reach out to clients (restaurants)		
	OPS.2.3	Form strategic partnerships (e.g. tech, local bodies)		
OPS.3: Operational	OPS.3.1	Operate delivery service nominally		
	OPS.3.2	Perform routine MRO operations		
OPS.4: Expansion Planning	OPS.4.1	Collect & analyse customer feedback		
	OPS.4.2	Explore new service regions		
	OPS.4.3	Prepare secondary funding round		
	OPS.4.4	Expand vendor contracts for larger volumes		

Legend	🟦	Design sub-phase
	🟦	Design task duration
	🟦	Design task buffer time
	🟨	Manufacturing sub-phase
	🟨	Manufacturing task duration
	🟨	Manufacturing task buffer time
	🟧	AIT sub-phase
	🟧	AIT task duration
	🟧	AIT task buffer time
	🟩	Operations sub-phase
	🟩	Operations task duration
	🟩	Operations task buffer time

Figure 15.2: Project development Gantt chart (throughput time in working hours)

17 Conclusion

The Design Synthesis Exercise (DSE) tasked Team DronebezORgd with the conceptual design of a drone-based pizza delivery system, developed from first principles over a 10-week period. The result of this effort is the **BezORger** — a fully electric, fixed-wing VTOL drone featuring a wing-folding mechanism, modular payload delivery system, and a tailored operations architecture. The project aimed to demonstrate the feasibility of delivering 200 pizzas in a single evening within a Delft-like urban environment, with minimal environmental and operational impact.

The project was structured in four main phases: project planning, systems engineering foundation (baseline), conceptual design selection, and preliminary subsystem design and integration. Each phase built on the previous, culminating in a detailed and coherent design supported by simulation, analysis, and iterative integration.

The final drone design incorporates several innovative features. The folding wing reduces the wingspan from 3.0 m to 1.6 m, enabling operation in constrained urban landing zones without the need for additional infrastructure. The payload handling system uses dual vertical stacks of independently accessible compartments, optimised for thermal retention, hygiene, and secure transport. A hybrid propulsion system, including vertical and horizontal rotors, supports both efficient cruise and stable VTOL performance.

Subsystem-level development was guided by rigorous requirements. Computational Fluid Dynamics (CFD) simulations were employed to optimise the wing planform and airfoil selection for lift efficiency. Structural design was informed by both idealised modelling and finite element analysis (FEA), ensuring strength under peak operational loads while minimising mass. A time-stepping energy simulation for all flight phases, including transition dynamics, provided a refined estimate of mission energy demand. Acoustic modelling confirmed that noise emissions remained within urban limits, supporting social sustainability goals.

The operation of the fleet was a key point of focus. A mission planning and path planning framework was implemented and tested on a simulated map of Delft, allowing evaluation of fleet size, depot location, and profitability. The use of mixed-integer linear programming (MILP) and A* algorithms enabled order assignment and route optimisation within defined constraints. This analysis demonstrated that a 10-drone fleet with six-pizza payload capacity offered the optimal balance between profitability and demand.

While challenges such as late-stage design inputs and subsystem integration complexity did arise, they were mitigated through adaptable planning, effective communication, and robust framework development. These experiences underscored the importance of systems thinking and interdisciplinary collaboration in a complex engineering project.

DronebezORgd extended beyond the scope of a conventional aerospace design exercise. Treated as a startup venture, the project included market analysis, financial modelling, and stakeholder identification. Cost constraints influenced design decisions, and the financial model demonstrated the potential for achieving break-even within five years under favourable regulatory and operational conditions.

Several future directions have been identified:

- **Enhancement of Operational Algorithms:** Further refinement of mission and path planning algorithms is necessary to enable scalability, real-time adaptability, and robustness across diverse urban environments.

- **Regulatory Pathways:** At present, commercial deployment is not feasible under current regulatory conditions. As such, both airworthiness and airspace regulations need to be complied with, with a potential need for lobbying for policy changes to allow for drone-based businesses.
- **Prototype Development and Testing:** Building and testing a prototype will be essential to validate design assumptions, verify safety and performance targets, and refine subsystem integration.
- **System Scalability and User Interface:** A fully operational version will require a robust control architecture, real-time communication systems, and a user-friendly interface for customers and operators, alongside considerations of data privacy and cybersecurity.

Although theoretical, the DronebezORgd system presents a design that is technically grounded, operationally feasible, and potentially scalable. The project demonstrates how an autonomous drone delivery system can be structured to meet functional, environmental, and economic constraints in a realistic urban context.

The DSE project also served as a comprehensive learning experience in engineering design, systems integration, and project execution. In addition to deepening understanding of core aerospace disciplines, the project introduced aspects of entrepreneurship, mathematical optimisation, and real-world constraints, offering valuable preparation for future work in both industry and academia.

References

[1] Andreas Swaan et al. *Baseline Report DronebezORgd: Designing a Fleet of Drones for Last-Mile Food Delivery AE3200: Design Synthesis Exercise.* Tech. rep. Delft University of Technology, 2025.

[2] Andreas Swaan et al. *Midterm Report DronebezORgd: Designing a Fleet of Drones for Last-Mile Food Delivery AE3200: Design Synthesis Exercise.* Tech. rep. 2025.

[3] Lucas Harms and Maarten Kansen. *Cycling Facts Netherlands Institute for Transport Policy Analysis | KiM.* Tech. rep. Ministry of Infrastructure and Water Management, 2018.

[4] Radimir Y Yanev and Ingo Staack. "FRAMEWORK DEVELOPMENT FOR CONCEPTUAL DESIGN AND CONFIGURATION ANALYSIS OF EVTOL AIRCRAFT". In: International Council of the Aeronautical Sciences. Florence, Sept. 2024.

[5] T H G Megson. *Aircraft Structures for engineering students.* Amsterdam Elsevier, 2017. ISBN: 978-0-08-100914-7.

[6] V. B. Jadhao, R. B. Charde, and S. M. Dhengle. "INVESTIGATION OF STRESSES IN CANTILEVER BEAM BY FEM AND ITS EXPERIMENTAL VERIFICATION". In: *International Journal of Technical Research and Applications* 3.1 (Jan. 2015), pp. 141–144.

[7] Ahmed Hasan and Ibrahim Dincer. "Investigation of new insulation materials for environmentally-benign food delivery bags". In: *Green Energy and Technology.* Springer Science and Business Media Deutschland GmbH, 2020, pp. 751–777. ISBN: 9783030206369. DOI: 10.1007/978-3-030-20637-6{_}37.

[8] European Commission. "Commission Regulation (EU) No 10/2011 of 14 January 2011 on plastic materials and articles intended to come into contact with food Text with EEA relevance". In: *Official Journal of the European Union* 12.1 (2011), pp. 1–89.

[9] Extrudr. *TECHNICAL DATASHEET GREENTEC PRO.* Tech. rep. Lustenau: FD3D GmbH, 2024.

[10] Extrudr. *REGULATORY INFORMATION SHEET GREENTEC PRO.* Tech. rep. Lustenau: FD3D GmbH, 2024.

[11] colorFabb. *Technical Datasheet colorFabb_HT.* Tech. rep. Belfeld: colorFabb, 2023.

[12] Godfrey Onwubolu. *Mechatronics Principles and Applications.* 1st ed. Oxford: Elsevier Butterworth-Heinemann, 2005, pp. 279–398. ISBN: 0 7506 6379 0.

[13] M. V. Yarygina and Yu I. Popov. "Development of the weight formula for a folding wing". In: *Russian Aeronautics* 55.2 (Apr. 2012), pp. 120–126. ISSN: 10687998. DOI: 10.3103/S106879981202002X.

[14] Josh Mills and Rafic Ajaj. "Flight dynamics and control using folding wingtips: An experimental study". In: *Aerospace* 4.19 (June 2017), pp. 2–24. ISSN: 22264310. DOI: 10.3390/aerospace4020019.

[15] Jatinder Goyal. *Blade Element Momentum Theory for Positive and Negative Thrust Propellers.* Feb. 2024. URL: https://gitlab.tudelft.nl/jatindergoyal/Bem_github2.

[16] Jatinder Goyal et al. "Benchmarking of Aerodynamic Models for Isolated Propellers Operating at Positive and Negative Thrust". In: *AIAA Journal* 62.10 (Oct. 2024), pp. 3758–3775. ISSN: 0001-1452. DOI: 10.2514/1.J064093.

[17] Hugo F.M. Bento, Reynard de Vries, and Leo L.M. Veldhuis. “Aerodynamic performance and interaction effects of circular and square ducted propellers”. In: *AIAA Scitech 2020 Forum*. American Institute of Aeronautics and Astronautics Inc, AIAA, 2020, pp. 1–21. ISBN: 9781624105951. DOI: `10.2514/6.2020-1029`.

[18] Donald B. Hanson. “Helicoidal surface theory for harmonic noise of propellers in the far field”. In: *AIAA Journal* 18.10 (Oct. 1980), pp. 1213–1220. ISSN: 00011452. DOI: `10.2514/3.50873;WEBSITE:WEBSITE:AIAA-SITE;WGROUP:STRING:AIAA`. URL: `/doi/pdf/10.2514/3.50873`.

[19] Jatinder Goyal. *Hanson’s Model in Frequency Domain - Tonal Noise of Rotors in Uniform Inflow.* Delft, 2024. DOI: `10.4121/7da5aa45-e44b-4fa3-9407-8bf61e835d99.v1`. URL: `https://data.4tu.nl/datasets/7da5aa45-e44b-4fa3-9407-8bf61e835d99`.

[20] Deepak C. Akiwate et al. “On the balance between the tonal and broadband noise of isolated propellers”. In: *International Journal of Aeroacoustics* 23.1-2 (Mar. 2024), pp. 122–153. ISSN: 20484003. DOI: `10.1177/1475472X231225631`.

[21] International Electrotechnical Commission. *NORME INTERNATIONALE: Electroacoustics – Sound level meters – Part 1: Specifications*. Geneva, 2013.

[22] American National Standard. *American National Standard Specification for Sound Level Meters*. 2006.

[23] Jatinder Goyal, Francesco Avallone, and Tomas Sinnige. “Isolated propeller aeroacoustics at positive and negative thrust”. In: *Aerospace Science and Technology* 147 (Apr. 2024). ISSN: 12709638. DOI: `10.1016/J.AST.2024.109021`.

[24] Christopher S. Thurman, Nikolas S. Zawodny, and James D. Baeder. *Computational Prediction of Broadband Noise from a Representative Unmanned Aerial System Rotor*. 2020.

[25] Mohammad H. Sadraey. “Design of unmanned aerial systems”. In: *Design of Unmanned Aerial Systems* (Feb. 2020), pp. 1–623. DOI: `10.1002/9781119508618;WGROUP:STRING:PUBLICATION`. URL: `https://onlinelibrary.wiley.com/doi/book/10.1002/9781119508618`.

[26] J.J Ruijgrok. *Elements of airplane performance*. 2nd ed. Delft: VSSD, 2009.

[27] Brenda Acevedo-Sánchez et al. “Friction coefficient of cardboard packing samples”. In: *Proceedings of the Institution of Mechanical Engineers, Part J: Journal of Engineering Tribology* (May 2025). ISSN: 1350-6501. DOI: `10.1177/13506501251341037`. URL: `/doi/pdf/10.1177/13506501251341037?download=true`.

[28] Dieter Jungnickel. *Algorithms and Computation in Mathematics • Volume 5*. Tech. rep. DOI: `https://doi.org/10.1007/978-3-642-32278-5{\_}5`. URL: `http://www.springer.com/series/3339`.

[29] Choong Yeun Liong et al. “Vehicle routing problem: Models and solutions”. In: *Journal of Quality Measurement and Analysis JQMA* 4.1 (2008), pp. 205–218. URL: `https://www.researchgate.net/publication/313005083`.

[30] Jean François Cordeau and Gilbert Laporte. “The dial-a-ride problem: Models and algorithms”. In: *Annals of Operations Research* 153.1 (Sept. 2007), pp. 29–46. ISSN: 02545330. DOI: `10.1007/s10479-007-0170-8`.

[31] K. Gkiotsalitis and A. Nikolopoulou. “The multi-vehicle dial-a-ride problem with interchange and perceived passenger travel times”. In: *Transportation Research Part C: Emerging Technologies* 156 (Nov. 2023). ISSN: 0968090X. DOI: `10.1016/j.trc.2023.104353`.

[32] Christian Zammit and Erik-Jan van Kampen. “Comparison between A* and RRT algorithms for UAV path planning”. In: *AIAA Guidance, Navigation, and Control Conference, 2018*. American Institute of Aeronautics and Astronautics Inc, AIAA, Jan. 2018. ISBN: 9781624105265. DOI: `10.2514/6.2018-1846`.

A
External Resources

A.1. GitHub

The code repository is hosted on GitHub and can be accessed here (URL:`https://github.com/andreasswaan/DSE-Group-18`). All code for the project was saved there.

A.2. AI Appendix

AI was used with careful consideration in this project. It was used for generating the cover page, grammar checks, and for LaTeX formatting and code debugging. Ethical use was kept in mind.