Group 30

AE3200-I Design Synthesis Exercise
25 June 2024

Cargonaut

Revolutionizing Warehouse Logistics, One Cargo Item at a Time

Cargonaut: Final Report

AE3200
Design Synthesis Exercise

by

Group 30

Student Name	Student Number
Ali Alper Ataşoğlu	5449545
Ana-Maria Pelin	5484960
Barnabas Tasi	5473918
Bernardo Sousa Alves	5460565
Celina Norrman	5567580
Federica Fenoglio Gaddò	5516773
Gowri Menon	5447372
Laura Daumerie	5494192
Mihaly Fey	5476747
Trayana Ivova Georgieva	5446155

Version Control			
Version	**Date**	**Author(s)**	**Description**
1.0	25 June 2024	All	Final version

Responsible Tutor: Alessandro Bombelli
Coaches: Olaf Stroosma, Arne Wulff
Cover Image: Generated by Leonardo.ai

April - June 2024
Delft University of Technology
Faculty of Aerospace Engineering

This page is intentionally left blank.

Acknowledgements

Point clouds danced on our depth camera,
Made possible by Barni and Mihaly,
Models overfit in learning's maze.
By none other than Ali himself,
FEMs crashed loud with Celina's rage,
But our design, at last, found its end.

Coffee fueled our weary eyes,
Supplied by none other than Laura,
Monitors glowed a sleepless prize.
"We cooked, we ate," the mantra grew,
Not to quote Bernardo there,
But stronger still, our friendship crew.

Per Federica's commands,
Let the drones be controlled!
But do not let Ana and Gowri hear,
Their collision course,
Nor the propeller damage,
Trayana has to concur.

All the jokes aside,
To the minds that shared this project's quest,
Coaches, and tutors, put us to the test.
Family, friends, a patient crew,
And to our faithful computers, too,
We show our humble gratitude.

But most of all, a heartfelt thanks,
Citronella, you weathered the ranks.
With you by our side, we never strayed,
This journey's end, a triumph made.

Executive Summary

In the logistics and cargo industry, obtaining accurate volume measurements of goods is crucial for optimizing the packing process during transportation. Inefficiencies in the packing process can lead to underutilization of cargo space, increased transportation costs, and delays.

To solve this issue, there is a need for a system able to obtain the dimensions of items in a fast and reliable way: Cargonaut aims to fulfill this by providing an innovative solution satisfying the mission need statement:

Provide an autonomous system to measure the volumes of cargo items in a ground handling warehouse to improve the flexibility of loading operations.

Market Analysis

Through market segmentation, the beachhead market is defined as air cargo ground handling facilities within the Europe (EU). This decision stems from the following two factors:

- Air cargo carriers are the most limited in terms of weight and volumetric capacity when considering air, land, and sea carriers.
- The European market is the most accessible for the Cargonaut project in terms of geographical location.

With the beachhead market known, the stakeholders can be identified and split into three categories:

- Direct stakeholders: airport crew and cargo handling, responsible for palletization, loading, and unloading
- Indirect stakeholders: shippers, consignees/customers, airlines, freight carriers and forwarders, any warehouse workers who will benefit from the increased flexibility of loading, provided by Cargonaut
- Regulatory authority stakeholders: IATA, EASA, ICAO, and local airports.

It is important to note that Cargonaut's market may potentially expand to encompass several other stakeholders. It can be assumed that the product might eventually be scaled to any other warehouse industry in need of improved volumetric inventory assessment. To confirm this possibility, a sensitivity analysis will need to take place later on in the design process.

A competitor analysis is conducted to identify other businesses that provide a solution to stereometrically measure cargo objects, after which the market gap can be identified: current solutions are restricted in terms of flexibility. This means that the size of objects to be measured is limited and the accuracy varies depending on the placement of the object with respect to the measuring solution. Cargonaut aims to fill this gap by providing a flexible, autonomous, and dynamic solution. The team has decided to design a fleet of quadcopter drones: this decision offers increased efficiency, accuracy, flexibility, and high productivity.

Project Management

The design process was initiated by reflecting on and recognizing the core values of Cargonaut. To do so, an analysis of the sustainable principles used during and in the process was highlighted. These sustainable values were naturally incorporated into subsystem design. Furthermore, a reflection on the trade-off choices made during conceptual and preliminary design is considered. During the design process, various contingency margins were used.

Product Development

The design process is guided through a functional analysis and stakeholder requirements. From the requirements, the project risks were derived: the focus was placed on stakeholder risks, as they represent the main driver of the project. The risk register, presenting likelihood and severity before and after mitigation is analyzed organizationally and technically. The mitigation strategies applied are individual for each risk and are devised to lower the risk metrics, repositioning the risks in the non-critical area of the risk map. The risks that were identified as being the most critical are related to the sustainability of the system and the privacy of the information acquired. To mitigate these risks, sustainability was identified as a focus throughout the design of the subsystems, the choice of manufacturing, and value-added market strategies. As for privacy, encryption is implemented in the transmission and regular security checks are performed throughout the operational life of the system.

Operations and Logistics

To offer the flexibility Cargonaut takes pride in, an efficient fleet management and planning subsystem is required. The design of this subsystem is considered for the truck unloading zones of the warehouse, with the general warehouse layout represented in Figure 1. A multitude of logistics strategies can be used to solve the fleet management and planning problem. Finally, to offer a scalable solution, a sequential approach with centralized fleet management and distributed fleet trajectory planning is implemented.

The fleet management system allocates tasks using a single agent-single task approach, with instantaneous assignments subject to sequential and proximity constraints. The scouting mode of the drones, the Cargoscout, identifies the locations and deadlines of cargo items. Sequential and proximity constraints are managed through clustering. Task allocation is performed using an auction-based algorithm, where tasks are introduced to the market by the Ground Station (GS). Drones bid on tasks based on their utility function, and the auctioneer, which is the GS in this case, processes the bids to allocate tasks.

Complementary to the fleet management system, a planning strategy is developed. Due to the structured environments, a grid-based algorithm coupled with a heuristic function that tunes according to the nature of the problem is used. The planning problem is divided into two domains: global and local planning. The global planner is to accommodate the non-convergence movement of the agents within the warehouse: This is done through an A* algorithm implemented in a 2D space that accounts for collision avoidance between agents by implementing multiple flight levels. The local planner provides the trajectory the drone should use to explore localized areas. This trajectory is determined by an area coverage algorithm.

Furthermore, practical considerations, required for seamless deployment, are considered. This includes the positioning of the docking station(s), also known as the Cargomother(s), and the possibility for manual override. Cargomothers are made in a modular manner to accommodate flexible fleet sizes and offer customization. Manual override is included to ensure that the system remains safe for the environment and workers.

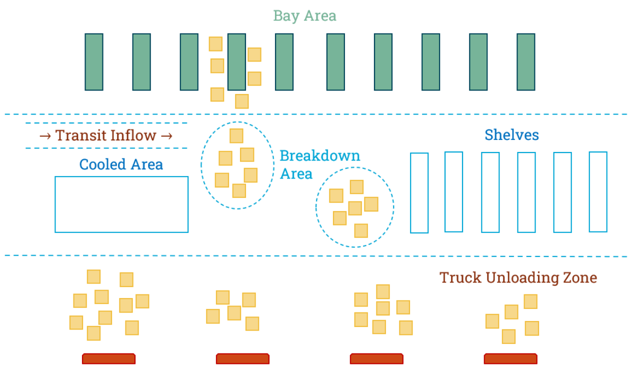


Figure 1: Reference warehouse layout

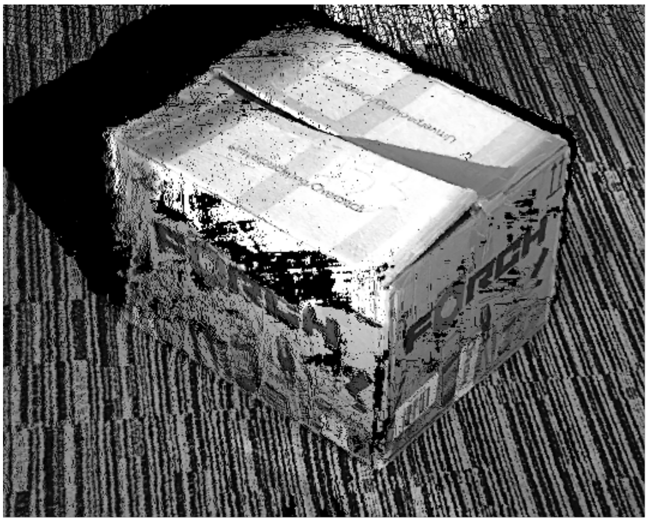


Figure 2: Example point cloud, measured in-house

Payload

Autonomous systems rely on computer vision as their primary sense, allowing them to identify, locate, and measure objects in their environment – the core function of the fleet of Cargonauts. This functionality is provided by the onboard Red Green Blue (RGB) and depth camera that every drone is equipped with, which allows every drone to visualize its environment. Using the camera, three main payload operations are conducted: object detection, label reading, and volume measurement.

Object detection is the process of identifying common objects within a warehouse to aid Cargonaut in collision avoidance and planning. This is done by training You Only Look Once (YOLO) version 8 model on 7 classes: cardboard boxes, people, wood pallets, barcodes, QR codes, drones, and forklifts, as these are the most common objects within a warehouse. This model detects objects per frame generated by the camera and is run continuously on the drones to aid the drone operations visually. The inference takes 27 ms per frame, meaning object detection is done in real time.

Label reading is conducted using the same architecture, yet focusing on identifying and reading QR codes and barcodes. The decoded values of these labels provide additional information that can be provided to the cargo item log in an attempt to further identify the cargo item in question. Given the primitive nature of this technology, all label-reading processes are optional, such that they do not hinder the measurement cycle.

Volume measurement is the last payload operation, which is done by generating point clouds (represented in Figure 3) of a target cargo item from different points of view, filtering them and registering them together. This filtered registration and segmentation of the cargo item from the environment allows for finding the dimensions of the cargo item. This cycle is conducted under 3 s per frame, which ensures that one cargo item can be measured under 30 s.

Navigation

Navigating the warehouse is a key enabler for mission performance and integrity. Localization, mapping, and collision avoidance are supported by an array of 10 micro-ranging-imaging sensors, which can map out the drone's surroundings, position it in space using the FAST-LIO2 algorithm, and detect any oncoming collisions. The odometry is supported by an additional inertial measurement unit. The absolute localization is further enhanced by QR codes or AprilTags, which serve as reference points during continuous operations. Collision avoidance is done by visual and time of flight sensors, in order to prevent interactions with static and dynamic objects within the warehouse environment.

Propulsion

The goal of the Propulsion subsystem was to support all necessary movement and enable all maneuvers, demanded by other departments for task fulfillment. Among the important considerations were generating sufficient thrust for minimum power consumption, but also executing all tasks safely and smoothly. The performed analysis concluded that the quadcopter shall be equipped with four 7x3″ propellers, consisting of two blades. Combined with the selected motors, they are able to provide a maximum Thrust-to-Weight ratio (T/W) of over 4:1, which increases agility and reinforces flexibility as a core value of Cargonaut. Nonetheless, complying with regulations for a Class 2 Unmanned Aerial Vehicle (UAV), the speed near warehouse workers shall be limited to 3 m/s.

Flight Performance

Within the department of Flight Performance, the behavior of the propulsive system was tested for various configurations, with a focus on its aerodynamic aspect. Simulating the environment involved the proper definition of boundary conditions, as well as propeller modeling and meshing. Relying on Computational Fluid Dynamics (CFD) computations and manufacturer data, the performance was examined for a single propeller in hover, vertically translating, and moving at a given bank angle, as seen in Figure 3. Interactions between multiple propellers were also analyzed, laying a solid foundation for future modeling of the full drone assembly. At the current stage, calculated values for thrust and power are 38.4% and 20.3% lower than manufacturer data, respectively. However, when changing the velocity of the incoming flow, OpenFOAM results follow the same trend as the rotor provider indicates. This provides confidence in the model, and encourages future model refinement to obtain more accurate numerical values.

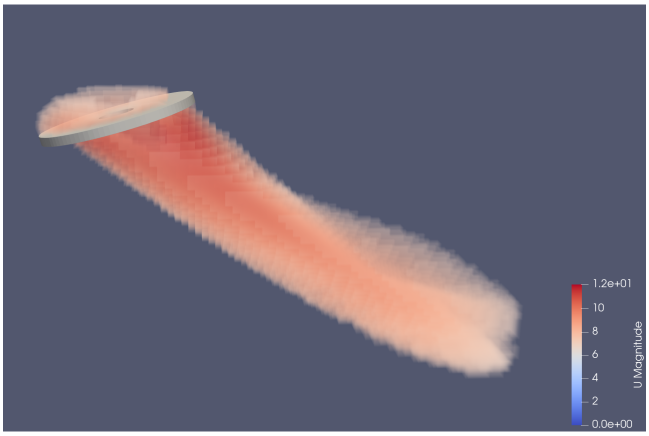


Figure 3: Analysis of propeller flow for 75° of tilt (with respect to the flow, coming horizontally from the left)

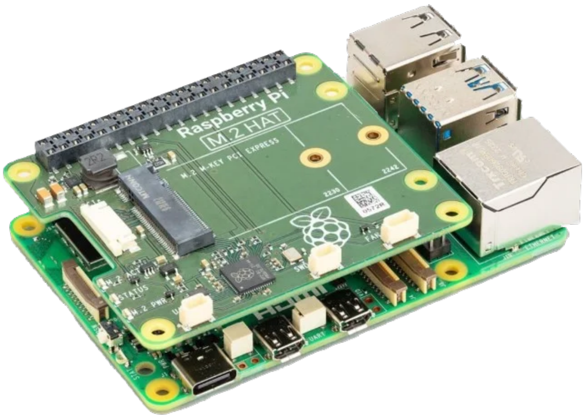


Figure 4: Raspberry Pi with M.2 Hat+

Communications and Data Handling

The goal of the Communications and Data Handling subsystem is to communicate with the GS and other drones within the fleet, and to process relevant incoming data, both from Cargonaut's sensors and from the mentioned external sources. To achieve this, WiFi has been selected as the data transmission method. An analysis of the messages exchanged between the drone and the ground station indicated a bandwidth requirement of 1,860.82 KB/s. The WiFi 6 protocol was selected, utilizing OFDMA modulation on the 2.4 GHz frequency to ensure better penetration through solid objects, typical in a warehouse environment. Channel selection is dynamic, tailored to the specific warehouse, with channels of 20 MHz bandwidth. A simple network scan software can aid in this selection process. Link budgets were calculated using the UAV Low Altitude Air to Ground (U-LAAG) model for path loss, and the chosen hardware setup includes a Raspberry Pi 5 flight computer with an integrated WiFi module and antenna, with a Hailo 8-L TPU module for advanced data handling, shown in Figure 4.

Following hardware selection, link margin calculations demonstrated that Cargonaut can maintain a connection up to 203 meters from the ground station and 130 meters from other drones. A detailed block diagram visualizes the data flow within Cargonaut, highlighting that image data, due to their large size, have the highest data rates. The verification and validation processes include sensitivity analysis on hardware data for link margin calculations, code verification, and validation through literature and Mavlab expertise. Future recommendations include exploring double-band hardware, which involves adding an extra antenna, and investigating latency to ensure efficient message delivery between components.

Electronics

The electronics subsystem is crucial, acting as the drone's powerhouse, managing battery operations and power transmissions. The main objective is to support a continuous power supply to all subsystems, ensuring sufficient energy throughout the mission. Key functions of the subsystem include connecting hardware for parallel operations, providing necessary power, redirecting electricity as needed, and ensuring the correct voltage and current are supplied. The design process prioritizes selecting components and strategies that mitigate potential risks, primarily focusing on the battery's performance and reliability.

In the design process, the selection of the battery is critical, with Lithium Polymer battery (LiPo) chosen for its high energy density and lower internal resistance. The battery capacity is carefully considered to balance between weight and power needs, ensuring a minimum hover time of 24.7 minutes for the drone. The power delivery system involves detailed planning for voltage conversions and power consumption to maintain efficiency. Lastly, sustainability and recycling considerations are also incorporated, aligning with environmental goals and ensuring responsible resource management.

Structures and Materials

The selected material for the subsystem is that of Wood-Carbon Fiber Plastic Composite (WCFC), chosen through a rigorous elimination and combination process backed by research and test data. WCFC is highly sustainable, with the potential to be up to 97% recycled, significantly exceeding the required 70%. Its excellent properties were verified through structural analysis, confirming its suitability for the application. The production process utilizes injection molding, which is also highly sustainable due to minimal waste, no labor costs, reusable molds, and a fast processing time, making it both efficient and cost-effective.

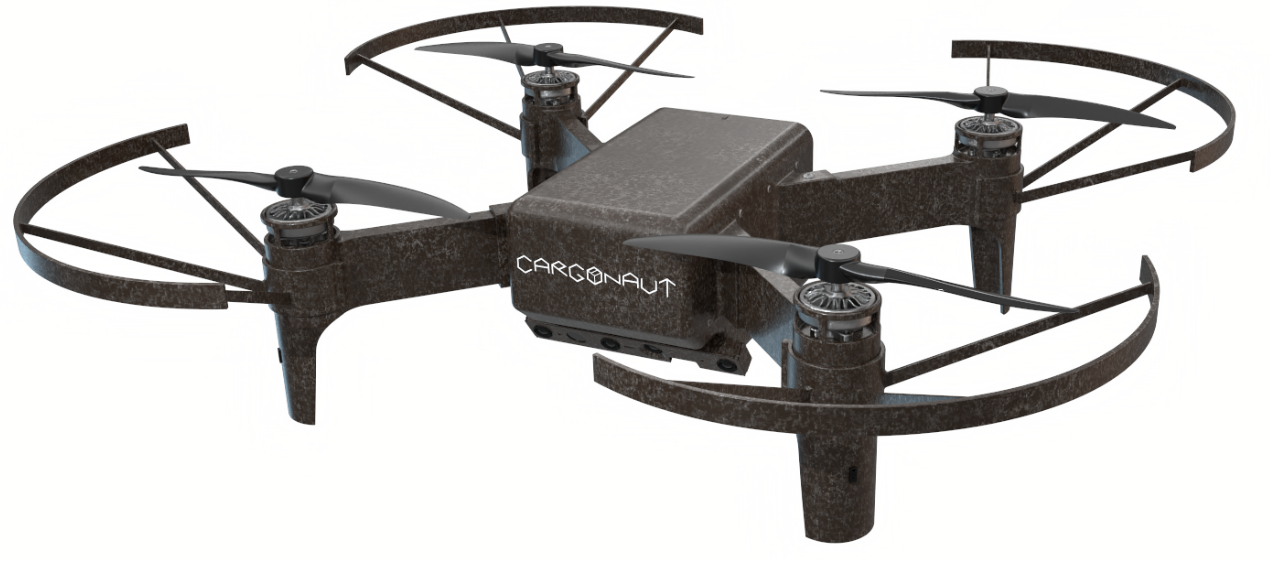


Figure 5: Render of final drone design

The structural design process involved choosing the appropriate topology and detailed design elements. The drone's frame will adopt a modular design, allowing for easy assembly and disassembly. The chosen topology is the true X configuration, which optimizes weight distribution and structural integrity. The frame will feature arms with rectangular cross-sections, selected for their higher moment of inertia in the vertical direction, which is critical for handling bending loads. This shape also allows for internal wiring, providing better protection and durability. Lastly, propeller guards are included to ensure the safety of human staff working with the drones. A render of the final drone is included in Figure 5.

The initial and modified arm geometries are analyzed under different load cases, comparing deflections and stresses to ensure all values remain within acceptable limits. Modifications are made to reduce mass while maintaining structural integrity, resulting in a finalized arm design that meets all requirements. Ansys Mechanical is used to perform Finite Element Analysis (FEA) on the arm and the whole drone frame to validate the design and check compliance with all relevant requirements.

Control

The control subsystem ensures stability and precise control across six degrees of freedom. The control architecture uses four Proportional–Integral–Derivative (PID) controllers for altitude, roll, pitch, and yaw, selected for their proven reliability. A control simulation, developed in Python, begins with setting initial and desired positions. It calculates position errors, processes these errors through the PID controllers to determine required accelerations, and computes the necessary propeller speeds.

To ensure the correct behavior of the drone, PID tuning is performed using the Ziegler-Nichols method and refined through trial and error. The tuned gains are verified through simulations of various trajectories, including waypoint-based paths to mimic real-world operations. Sensitivity analysis was performed to ensure robust performance under parameter variations. Future recommendations include testing the system under real-world conditions and implementing auto-tuning algorithms for further optimization. Continued validation will follow with a physical prototype, ensuring the UAV meets operational performance standards.

Integration

Integration was performed to unify all subsystems of the drone system into a cohesive whole. This involves merging the software components with the hardware elements. The integration process ensures that the technical budgets for mass and power are balanced and aligned with the design specifications. A final cost breakdown analysis was performed, confirming the project's financial viability. Crucially, the integration process verifies that the system meets all requirements and regulatory standards. This holistic approach ensures that the final product is both functional and market-ready, providing a scalable and efficient solution for cargo measurement.

Post-DSE

The team focused on post-Design Synthesis Exercise (DSE) activities, outlining the steps planned after the exercise to further develop, commercialize, and integrate the Cargonaut product into the market. A project logic map is introduced, ensuring alignment with the initial goals and providing a systematic approach for future improvements, iterations, and company growth. Key components include analyzing the current situation, identifying priorities, and outlining the expected short, medium, and long-term impacts. Additionally, preliminary diagrams like the workflow diagram, work breakdown structure, and Gantt chart are used to plan critical future events and establish a logical flow of operations, ensuring a smooth transition from the DSE to corporate life.

Market integration is also considered, focusing on a dual entry strategy of direct sales and partnerships, targeting high-demand areas such as airports and cargo terminals. Cargonaut aims to offer a premium, flexible, and efficient solution compared to existing systems, supported by proactive customer support and value-added services. The long-term growth strategy emphasizes innovation, market expansion, and continuous improvements, with a commitment to sustainability and community engagement. Risk analysis identifies economic, market, operational, and regulatory risks, with mitigation strategies such as market monitoring, customer feedback, quality assurance, and compliance monitoring to ensure stability and growth.

Contents

List of Symbols

Greek Symbols

α	Angle of attack	rad
α	Angular acceleration	rad/s
β	Blade angle	rad
δ_{allow}	Maximum allowable deflection	mm
ν	Poisson's ratio	-
ω	Rotational velocity	rad/s
ω	Turbulence specific dissipation rate	1/s
$\omega_1, \omega_2, \omega_3, \omega_4$	Propeller speeds	rpm
ω_{hov}	Hover propeller angular velocity	rpm
ω_{max}	Maximum propeller angular velocity	rpm
ϕ	Flow angle	rad
ϕ	Roll angle	rad
ψ	Yaw angle	rad
ρ	Density	kg/m^3
ρ_{air}	Density of air	kg/m^3
σ_{allow}	Maximum allowable stress	MPa
σ_y	Yield strength	MPa
τ_ϕ	Torque in roll	Nm
τ_ψ	Torque in yaw	Nm
τ_θ	Torque in pitch	Nm
θ	Flow angle	rad
θ	Pitch angle	rad

Latin Symbols

A	Area	m^2
A_d	Reference area for drag	m^2
b	Propeller thrust factor	N s^2
b	Width of the cross section	m
bat	Battery status	-
c	Chord	m
C_d	Drag coefficient	-
C_l	Lift coefficient	-
C_T	Thrust coefficient	-
d	Propeller torque drag factor	Nm s^2
d	Required commute	m
dQ	Sectional torque	Nm
dT	Sectional thrust	N
E	Young's modulus of elasticity	GPa
e_x, e_y, e_z	Position errors in x, y, and z directions	-
e_ϕ, e_θ, e_ψ	Errors in roll, pitch, and yaw	-
F	Point force	N
F	Tip factor	-
f_z	Axial blade sectional force	N
f_{hov}	Hover propeller frequency	Hz
f_{max}	Maximum propeller frequency	Hz
f_θ	Tangential blade sectional force	N
g	Gravitational acceleration	m/s^2
h	Height of the cross section	m
I	Current	A
I	Mass moment of inertia	kg·m^2
k	Turbulence kinetic energy	m^2/s^2
K_E	Kinetic energy	J
K_v	Motor's speed constant	rpm/V
L	Reference length	m
l	Distance from rotor to drone's center	m
M	Bending moment	Nm
m_{max}	Maximum drone mass	kg
m_{motor}	Motor mass	kg
m_{prop}	Propeller mass	kg
P	Power	W
P_E	Potential energy	J
Q	Torque	Nm
Q_{hov}	Hover propeller torque	Nm
Q_{max}	Maximum propeller torque	Nm
R	Resistance	Ω
r	Radius	m
S	Drone surface area	m^2
SF	Safety factor	-
T	Thrust	N
t	Thickness of the cross section	m
T/W	Thrust-to-weight ratio	-
T_x	Thrust component in x-axis	N
T_y	Thrust component in y-axis	N
T_{howp}	Hover thrust per propeller	N
T_{how}	Total hover thrust	N
T_{maxp}	Maximum thrust per propeller	N
$th\%$	Throttle percent	-
u'_{rms}	Perturbation velocity, root mean square value	m/s
V	Voltage	V
v	Current drone speed	m/s
v_{imp}	Impact velocity	m/s
v_{max}	Maximum drone speed	m/s
W	Inflow velocity	m/s
W	Weight	N
w	Load distributed along the length	N/m
y	Distance from the neutral axis	m

List of Abbreviations

ABS	Acrylonitrile Butadiene Styrene
ADT	Actuation Disk Theory
AI	Artificial Intelligence
BEM	Blade Element Momentum Theory
BET	Blade Element Theory
BPF	Blade Pass Frequency
CDH	Communications and Data Handling
CFD	Computational Fluid Dynamics
COCO	Common Objects in Context
COTS	Commercial Off-The-Shelf
CPU	Central Processing Unit
DSE	Design Synthesis Exercise
DoF	Degree of Freedom
EOL	End of Life
EPS	Electrical Power System
ESC	Electronic Speed Controller
ESA	European Space Agency
EU	Europe
FBD	Free-body Diagram
Fast-RCNN	Faster-Region Convolutional Neural Network
FCS	Frame Sequence Control
FEA	Finite Element Analysis
FEM	Finite Element Method
FM	Fleet Management
FP	Flight Performance
FPFH	Fast Point Feature Histogram
FTP	Fleet Trajectory Planning
GPU	Graphical Processing Unit
GS	Ground Station
ICP	Iterative Closest Point
IMU	Inertial Measurement Unit
I/O	Input/Output
IR	Infra-Red
ISM	Industrial, Scientific, and Medical
KLM	Koninklijke Luchtvaart Maatschappij N.V. (Royal Dutch Airlines)
LiDAR	Light Detection and Ranging
LDO	Low-Dropout Regulator
MAC	Media Access Control
MAPP	Maleic Anhydride-graded Polypropylene
MLESAC	Maximum Likelihood Estimation Sample Consensus
MU MIMO	Multi-User Multiple-Input Multiple-Output
NASA	National Aeronautics and Space Administration
NDT	Normal Distribution Transform
OFDMA	Orthogonal Frequency-Division Multiple Access
PBB	Plant-Based Bioplastic
PCB	Printed Circuit Board
PCD	Point Cloud Data
PCL	Point Cloud Library
PDF	Probability Density Function
PFH	Point Feature Histogram
PID	Proportional–Integral–Derivative
RANS	Reynolds-averaged Navier–Stokes equations
RANSAC	RANdom SAmple Consensus
RGBD	Red Green Blue Depth
RGB	Red Green Blue
RnD	Research and Development
RFID	Radio Frequency Identification
RPM	Revolutions per minute
RPB	Netherlands Institute for Spatial Research
RRT	Rapidly-exploring Random Trees
RSSI	Received Signal Strength Indicator
SLAM	Simultaneous Localization And Mapping
SSD	Single-Shot Detector
SWOT	Strengths, Weaknesses, Opportunities, and Threats
TBC	To be confirmed
ToF	Time of Flight
TOPS	Tera Operations Per Second
TPM	Technical Performance Measurement
TPU	Tensor Processing Unit
TRL	Technology Readiness Level
T/W	Thrust-to-Weight ratio
UAV	Unmanned Aerial Vehicle
U-LAAG	UAV Low Altitude Air to Ground
ULD	Unit Load Device
UN	United Nations
UNSDGs	United Nations Sustainable Development Goals
VLM	Vortice Lattice Method
VnV	Verification and Validation
VPM	Vortice Particle Method
WCFC	Wood-Carbon Fiber Plastic Composite
WiFi	Wireless Fidelity
WPA	WiFi Protected Access
WPA3	WiFi Protected Access 3
WPC	Wood-Plastic Composite
YOLO	You Only Look Once
LiPo	Lithium Polymer battery
EASA	European Union Aviation Safety Agency

1: Introduction

Air cargo transports over 5.5 trillion euros worth of goods each year, accounting for approximately 35% of world trade by value[1]. In 2021, 65.6 million metric tons of freight was transported by aircraft globally[2]. Despite this, the ground handling stage is still a significant bottleneck in the air logistics supply chain[3]. Errors in the volume data provided by shipping companies or the lack of calculated volumetric dimensions entirely present a significant hurdle in ground handling and cargo load planning [1]. This commonly results in delays and the suboptimal utilization of aircraft cargo space. Although the integration of cameras for faster and more precise volume measurement has been initiated[4], they often pose restrictions on the size and shape of packages that can be measured, and they are limited in the minimum distance from which they can perform the measurements. Their manned nature also presents the additional cost of manual labor, hindering the overall operational profitability.

Cargonaut aims to improve the flexibility of loading operations in ground-handling warehouses by developing an autonomous system to measure the volume of cargo items. Warehouse facilities and ground workers can rely on this technology to improve operational efficiency and thereby increase processing throughput, making use of a novel solution to a deep-rooted problem in the air cargo transportation and logistics industry. In addition to keeping an accurate inventory of the packages, the information gathered by the system can be refined further to help solve the air cargo palletization problem [1].

The purpose of this report is to build upon the previous work packages and to perform the final design phase of the project, converging to a solution that complies with the stakeholder requirements. In earlier works, the project organization and planning have been presented. Furthermore, the preliminary design has been conducted, generating design options, analyzing them, and performing trade-offs. Following is the detailed design phase, presented in this report along with a review of some of the proceedings. The process makes use of concurrent engineering to cater for seamless integration of subsystems and to refine requirements in parallel. The organization of the project closely follows system engineering guidelines and targets to reflect a top-down approach flowing from the stakeholder requirements and mission statement to lower-level requirements and subsystem design concluded by a bottom-up design verification and validation. The entirety of the process was guided by the core values of Cargonaut: sustainability, flexibility, efficiency, and safety. The final product is aimed at reflecting these values as well as making a significant step towards improving logistic operations.

To put the challenge in context and to further highlight the gaps Cargonaut can fill a market analysis opens this document in Chapter 2. Chapter 3, Chapter 4, and Chapter 5 on project management, product development, and interface modeling respectively translate these needs into engineering constraints and aim at devising an interface chart to identify potential design conflicts and to ensure proper system integration. The report focuses next on subsystem design. Here, each chapter presents the goal, the main functions, and the requirements of the subsystem. Then follows the design approach concluded by verification, validation, and future recommendations. First, Chapter 6 describes how operations ensure a flexible and efficient measurement of cargo items. Chapter 7 presents a solution to volume measuring, the main function of the system. It also includes object identification for collision avoidance and target recognition. Chapter 8 builds on the design presented in the preceding chapters to describe how the mobile vehicles perform localization within the operational area. Shifting focus onto the supporting hardware, Chapter 9 presents a propulsion system enabling the mobile nature of the system. Chapter 10 provides a knowledge framework for design optimization by analyzing the vehicle's mechanics. Next, Chapter 11 about communications and data handling ties together the commands and the data flow of the subsystems. Electronics in Chapter 12 ensures the power flow to the subsystems. Chapter 13 about structures and materials designs a rigid frame to house most subsystems, while also focusing on sustainable material selection. Control design in Chapter 14 finally concludes the subsystem design by ensuring the stability and controllability of the system. Chapter 15 focuses on the integration of the previously devised subsystems. Sustainability is incorporated in Chapter 16, while Chapter 17 describes project integration after the design phase. Finally, Chapter 18 concludes the report.

[1] Value of Air Cargo: https://www.iata.org/en/programs/cargo/sustainability/benefits/. Accessed 3 May 2024.

[2] Worldwide Air Freight Traffic from 2004 to 2021: https://www.statista.com/statistics/564668/worldwide-air-cargo-traffic/. Accessed 2 May 2024.

[3] Air Cargo Handling has become a Bottleneck: https://logfret.com/air-cargo-handling-has-become-a-bottleneck/. Accessed 19 June 2024.

[4] KLM Tech data: https://techdata.klm.com/en/our-work/cargo-volume-scanning/. Accessed 3 May 2024.

2: Market Analysis

This chapter presents a market analysis for the Cargonaut project. The market segmentation and justification for focusing on the air cargo ground handling industry is outlined in Section 2.1. The stakeholders are then identified and elaborated upon in Section 2.2. An evaluation of current market competitors is performed in Section 2.3, followed by the introduction of the solution that Cargonaut will offer. Finally, a SWOT Diagram is constructed to establish the project's market position in Section 2.4.

In the logistics and cargo industry, obtaining accurate volume measurements of goods is crucial for optimizing packing during transportation. Inefficiencies in packing can lead to underutilization of cargo space, increased transportation costs, and delays. To solve this issue, there is a need for a system able to obtain the dimensions of items in a fast and reliable way. Cargonaut aims to solve this by providing an innovative solution to the problem.

The first step for the creation of a successful and useful product is to perform an analysis of the market, to gain valuable knowledge about the key industries and players. This chapter thus focuses on market analysis and needs.

2.1. Market Segmentation

The market analysis is here performed from a functional perspective, where the system's ability to measure the volume of packaged objects is used to evaluate possible market segments. In general, the broad market is identified to be the ground storage and operations division of the logistics and transportation industry.

The logistic chain of cargo transportation consists of several segments. Shippers provide the goods, which are then brought by freight forwarders to ground handlers in the area of origin. They are then placed onto carriers to be transported near the final destination. Ground handlers then pass them on to freight forwarders, who bring them to the consignees, or the end receivers of the goods [2]. Three carrier modes can be established: **air**, **land**, and **sea**. Air carriers include all-cargo aircraft and passenger aircraft with cargo capacity [2]. Land carriers comprise trucks, trains, vans, and other smaller delivery vehicles[1]. Sea carriers consist of general cargo ships, container ships, bulk carriers, tankers, and barges[2].

Air cargo carriers are the fastest and most reliable mode of freight transportation, offering a wide range of global destinations virtually unrestricted by geographical barriers[3]. However, they are the most limited in terms of weight and volumetric capacity. Air cargo accounts for approximately 35% of world trade by value, but less than 2% by volume [2]. The rapid growth of the cargo industry has also highlighted the ground handling stage to be a bottleneck in the overall air logistics supply chain[4]. The beachhead market is therefore identified to be air cargo ground handling warehouses: this is due to the substantial transportation benefits, critical limiting factors, and potential for further optimization of air freight logistics.

The European market will be primarily considered due to its proximity to the project. To streamline the requirement compliance process, EU regulations will be followed, further narrowing the market to air cargo ground handling facilities within the EU. The Netherlands Institute for Spatial Research (Netherlands Institute for Spatial Research (RPB)) identifies 20 major cargo airports, located in Germany, France, Luxembourg, Belgium, Italy, Denmark, Spain, Austria, and the Netherlands [3] More specifically, the European air fright industry is valued at 32.3 billion $ and expected to grow by 6.6% in the forecast period 2024-2032[5], with Frankfurt Airport, Paris Charles de Gaulle Airport, and Amsterdam Schipol Airport accounting for the majority of the processing[6]. Targeting these major hubs will be the focus of Cargonaut's mission, which will be accomplished by leveraging advanced technology, providing a reliable product, and establishing strategic partnerships with key stakeholders in these airports. This approach will ensure that Cargonaut can effectively streamline cargo handling processes and meet the growing demand within the European air freight market.

[1] Land Freight Transport: https://www.fmlogistic.es/en/blog/land-freight-transport/. Accessed 2 May 2024.

[2] Five Major Ship Types of Sea Transport Used for Freight: https://omnilogistics.com/five-major-ship-types-of-sea-transport-used-for-freight/. Accessed 2 May 2024.

[3] Sea vs Air vs Land Freight: https://www.seaspace-int.com/sea-vs-air-vs-land-freight-what-is-the-best-method-of-transport-for-you/. Accessed 2 May 2024.

[4] Air Cargo Handling Has Become a Bottleneck: https://logfret.com/air-cargo-handling-has-become-a-bottleneck/. Accessed 13 May 2024.

[5] Europe Air Freight Market Overview: https://www.marketresearchfuture.com/reports/europe-air-freight-market-21406. Accessed 17 June 2024

[6] Volume of Airfreight Processed by Europe's Leading Airports in 2020: https://www.statista.com/statistics/434381/airfreight-volumes-in-europe-by-airport/. Accessed 2 May 2024.

2.2. Stakeholder Identification

A visual representation of the stakeholders is presented in Figure 2.1. Here, a distinction is made between direct stakeholders, to whom Cargonaut sells the product directly, and indirect stakeholders, who may not use the product directly but are affected by its operations. In the direct category, the main players are the airport crew and cargo handling, responsible for palletization, loading, and unloading. On the indirect side, shippers, consignees/customers, airlines, freight carriers and forwarders, and other warehouse workers will all benefit from increased flexibility of loading, which can contribute to a faster process. Additionally, the product shall abide by certain regulations and will need to be certified for successful roll-out, so regulatory authorities like the Dutch government or European Union Aviation Safety Agency (EASA) are also main stakeholders.

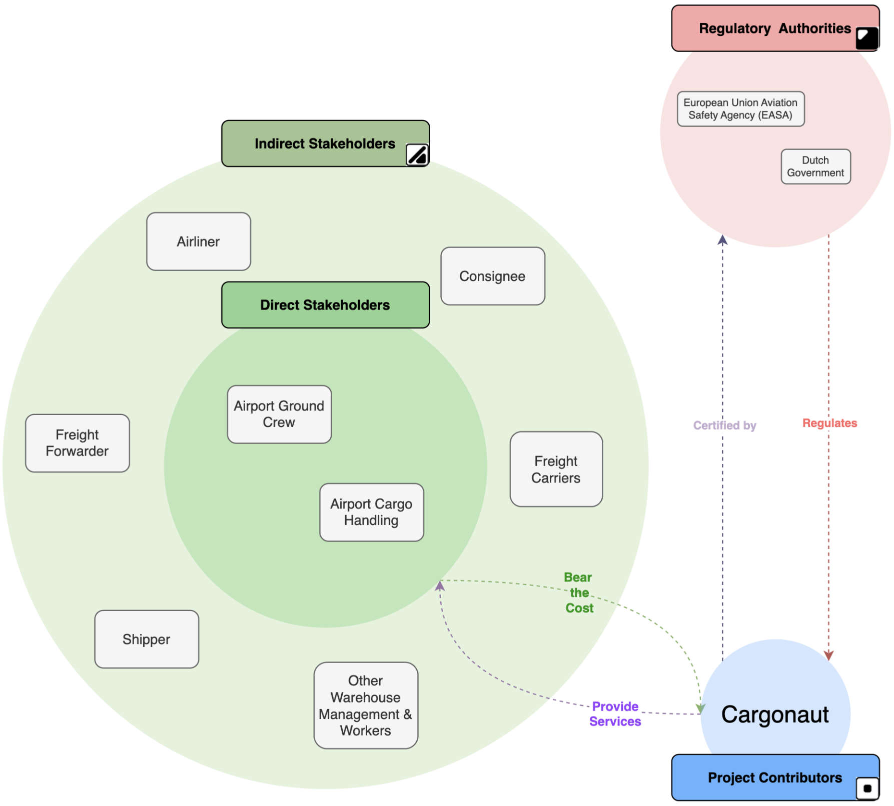


Figure 2.1: Stakeholder map

While air cargo ground operations will be the focus of Cargonaut's mission, the final product will likely be scalable to any warehouse industry in need of improved volumetric inventory assessment. Thus, "Other Warehouse Management & Workers" are identified to be indirect stakeholders. After Cargonaut's design is completed for air cargo facilities, a sensitivity analysis will be performed to investigate the extent to which different operational layouts within other industries have an effect on its configuration. Within the logistics and transportation industry, five main sectors can be identified: **postage**, **retail**, **wholesale**, **e-commerce**, and **public storage** services.

The postage sector deals with the shipment of packages and parcels sent by individual people, typically through postal services or courier companies. The retail sector pertains to selling products directly to consumers, whereas the wholesale sector relates to selling bulk goods to other businesses[7]. The e-commerce sector handles the operations behind online orders placed by customers, and the public storage sector provides independent entities with rentable storage spaces. All of these services require warehouse facilities to manage and store their goods. Several industries fall under the aforementioned categories, some of which are given in the following list. Their names are mentioned along with the various goods they would likely need to be stored in warehouses.

- **Food & beverage**: perishable and non-perishable consumables
- **Manufacturing**: components, unfinished and finished products
- **Automative**: vehicle materials, parts, and equipment
- **Aerospace**: military equipment and aerospace components
- **Healthcare**: pharmaceutical and medical products
- **Agriculture**: seeds, fertilizers, and farming tools
- **Construction**: building materials and tools
- **Chemicals**: hazardous materials
- **Fashion**: textiles, clothing, and accessories
- **Electronics**: electronic and IT equipment
- **Furniture**: furniture and home decor
- **Waste**: recyclable materials and waste products

[7] Wholesale vs Retail: https://www.shopify.com/retail/wholesale-vs-retail. Accessed 1 May 2024.

2.3. Competitor Analysis

Competitor analysis is a paramount step in assessing the market, as it allows for the identification of the market gap that ones wishes to fill. This section presents an analysis of the competition in Section 2.3.1 where the gap is identified. Section 2.3.2 shows the solution that Cargonaut will implement to position itself successfully in the market.

2.3.1. The Market Gap

Within any market analysis, it is crucial to consider competitors in the market, as these strongly influence the overall marketability of a new product. Moreover, the output of the competitor analysis can help clarify the market gap that Cargonaut aims to fill and can provide reference material for future design decisions. For clarity, throughout this section, competitors are defined as existing key players that provide a solution to the problem that Cargonaut will be solving.

From online literature research, it has become clear that the existing solutions regarding automated cargo measuring can be split up into two main groups: stationary solutions and dynamic solutions[8]. Stationary solutions are those that enable the measuring of stable objects, while the latter enables the measuring of objects that are moving e.g. on a conveyor belt. Additionally, note that both types of existing solutions are fixed, e.g. fixed scanner placed above conveyor belt.

Furthermore, an overall advantage of most of these existing solutions is that the cargo packages, to be stereometrically measured, can usually be weighed in addition, either in their stationary position as they are brought toward the fixed solution setup or as they move dynamically e.g. on the conveyor belt or forklift. Some of the existing solutions researched are the following: Cargometer (used by Koninklijke Luchtvaart Maatschappij N.V. (Royal Dutch Airlines) (KLM)), DS-C4RGO, Beevision, Metrilus, and Cargo Spectre. To get a better view of how these different solutions compare, their stereometric measurement accuracies are defined within Table 2.1[8] [9] [10] [11] [12].

Table 2.1: Stereometric measurement accuracy of Cargonaut's competitors

Competitor	Cargometer	DS-C4RGO	Beevision	Metrilus	Cargo Spectre
Accuracy	± 3 cm	± 0.8 to 2 cm	± 0.5 cm	± 1 to 2 cm	± 0.5 cm

Along with this competitor analysis, a market gap could be acknowledged: current solutions are restricted in terms of flexibility. They might not be able to measure objects of certain large sizes, as these might not fit in the fixed solution's field of view or on the potential conveyor belt (in the case of a dynamic solution). Moreover, the statement also refers to the accuracy of measurements, which is influenced by the distance and angle between the fixed solution and the object to be measured.

2.3.2. Solution

Based on the market gap presented above, it is obvious that the market is missing a solution that does not require human help, can be used to measure a wide variety of items, independently of shape and size, and is able to operate and provide measurements in the entire warehouse environment. Cargonaut's mission need statement was thus identified as:

Provide an autonomous system to stereometrically measure volumes of cargo items in a ground handling warehouse to improve the flexibility of loading operations.

With this in mind, the team focused on creating a system that would be autonomous, flexible, and dynamic: a quadcopter drone was chosen to be the most suited option to achieve such goals. The unmanned aerial vehicle (UAV) is able to fly around to measure items located everywhere in the warehouse, can achieve high accuracy by moving closer to the target and achieve different positions and orientations with respect to it, and is not limited by the size and shape of the cargo items, and can operate in full autonomy. Furthermore, the drones will be able to operate both individually and in fleet formation, allowing for higher productivity rates, efficiency, and better operational flexibility.

[8] Beevision: https://www.beevision.ai/. Accessed 1 May 2024.

[9] Cargometer "On-the-fly" Freight Dimensioning: https://www.ifoy.org/en/ifoy-2019-en/nominations-2019/cargometer-frachtvermessung-on-the-fly. Accessed 3 May 2024.

[10] DS-C4RGO: 1st Multi-Volume Scanner: https://www.c4rgo.com/en/data-scanner-3d-ds-c4rgo/. Accessed 3 May 2024.

[11] Metrilus: https://www.metrilus.de/products/l-230-235-pallet-dimensioner. Accessed 3 May 2024.

[12] Cargo Spectre: https://www.cargospectre.com/parcel-dimensioner/. Accessed 3 May 2024.

2.4. Market SWOT Diagram

A SWOT Diagram was generated to indicate the position of Cargonaut in the current market. SWOT stands for Strengths, Weaknesses, Opportunities, and Threats, where the first two refer to internal factors within the project and the latter two refer to external factors within the market. The analysis is shown here:

- Strengths:
  - Flexibility: Cargonaut's drone-based system offers unparalleled flexibility in measuring cargo items of varying sizes and shapes within ground handling warehouses.
  - Accuracy and speed: The drone's ability to maneuver close to objects ensures high measurement accuracy and speed, enhancing operational efficiency.
  - Scalability: The system is easily scalable and does not require extensive modifications to warehouses.
  - Environmental sustainability: Utilization of recycled materials underscores Cargonaut's commitment to sustainability, appealing to environmentally conscious stakeholders.
- Weaknesses:
  - Cost structure: There is a risk that the development and operational costs of the drone system may lead to a higher price point compared to existing solutions.
  - Limited expertise in drone technology: The project team lacks extensive experience in drone development and manufacturing, potentially impacting initial implementation and scalability.
  - Lack of weight measurements: The product is unable to accurately measure weight of cargo items.
- Opportunities:
  - Growing drone industry: The global investment in drone technologies is expanding rapidly, presenting opportunities for Cargonaut to leverage advancements and partnerships in the industry[13].
  - Expansion in air cargo ground handling: The air cargo ground handling services market is poised for growth, driven by increasing trade volumes and logistics efficiency demands[14].
  - Technological advancements: Automation and Artificial Intelligence are progressing rapidly, allowing for more efficient and reliable algorithms and pipelines.
- Threats:
  - Competitive market: Intense competition from established and emerging players offering similar drone-based solutions for logistics and warehouse automation poses a threat to market penetration.
  - Regulatory challenges: Evolving regulations governing drone operations, including airspace restrictions and safety protocols, could impact Cargonaut's deployment and operational flexibility[15].
  - Operators' experience: The noise and presence of drones could negatively affect the comfort and experience of operators.

A visual summary is depicted in Table 2.2.

Table 2.2: Market SWOT diagram

	Helpful	**Harmful**
Internal	*Strengths* • More flexible • Higher accuracy and speed • Scalability • Sustainability	*Weaknesses* • Higher costs • Limited expertise • Low accuracy in weight measurements
External	*Opportunities* • Growing drone industry • Expansion in air cargo ground handling • Advancements in technology	*Threats* • Competitive market • Increasing regulations on drone usage • Operators' experience and comfort

[13] Global Investments in Drone Companies from 2008 to 2020: https://www.statista.com/statistics/1117058/global-commercial-drone-investments/. Accessed 2 May 2024.

[14] Airport Ground and Cargo Handling Services Market: https://www.fortunebusinessinsights.com/airport-ground-and-cargo-handling-services-market-105327. Accessed 3 May 2024.

[15] The Evolution of Drone Laws: https://droneii.com/the-evolution-of-drone-laws. Accessed 3 May 2024.

3: Project Management

In this chapter the general organizational approach is presented along with the principles employed in completing the product development. General organizational aspects agreed upon at the beginning of the project are first introduced, while managerial design tools are later detailed. Section 3.1 opens the chapter by presenting the organizational sustainability guidelines that were followed throughout the DSE. Shortly after, in Section 3.2, design tools such as trade-offs and sensitivity analysis are presented along with their contribution to the progress of the project. Lastly, resource management and project management-related tools are discussed in Section 3.3.

3.1. Sustainability within the Project

After a general introduction to sustainability principles and their dimensions is presented, clear guidelines regarding their implementation in this project are defined. Several rules are agreed upon, which represent the team's collective effort toward organizational sustainability:

1. **Conscious use of resources:** a dynamic project of this scope demands utilizing several resources, in terms of both physical and software tools. Following the philosophy of lean manufacturing, each member will consciously try to create value while minimizing waste. Additionally, members will aim to keep the online directories organized and the progress documented.
2. **Environmental awareness:** apart from the use of sustainable products, healthy habits that benefit personal well-being and the environment are also encouraged. These include choosing green transport, either walking on foot or riding a bicycle, for commuting to and from the workspace. In addition, awareness is promoted through limited electricity consumption, which involves turning off computers and lights whenever possible.
3. **Positive work environment:** to ensure a smooth, high-quality product and an enjoyable work process, the social aspect of sustainability is highly valued. Clear communication and constructive feedback are recommended, along with occasional coffee breaks, to promote high productivity rates. To ensure healthy collaboration and bonding, weekly check-ins and team-building activities were organized, prioritizing carbon-neutral options such as picnics, and house-hosted dinners.

Alongside integrating sustainability into the practical aspect of the design process, sustainability was an intrinsic component driving the design choices. Wherever applicable within subsystem design and analysis, measures were taken to increase the feasibility of the product. This is evident through the choice to design a flexible, and robust product.

3.2. Trade-off Analysis

In the design process, trade-off analysis is essential for selecting the most suitable concepts from a set of initial options. This methodology can be applied to various subsystems to ensure a systematic and objective evaluation.

First, the remaining design concepts are revisited and clearly defined. A numerical trade-off method is then introduced, which involves identifying relevant criteria for evaluation and assigning weights to these criteria based on their importance. The importance of each criterion is rated on a scale from 1 to 10, with higher values indicating greater importance: a visualization is shown in Figure 3.1

Each concept is evaluated against these criteria, with performance scores ranging from 1 (worst) to 5 (best). To enhance readability, these evaluations are presented in tables with color coding, where red indicates lower scores and green indicates higher scores. The concept with the highest overall rating is selected for further development.

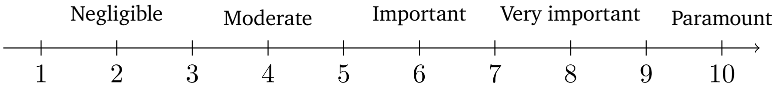


Figure 3.1: Trade-off criteria weighting scale

To validate the chosen concept, a sensitivity analysis is performed. This analysis checks the robustness of the selected solution by examining how changes in the weights of the criteria affect the outcome. The weights are adjusted based on the level of uncertainty associated with each criterion, and the trade-off analysis is recomputed for each iteration. This process helps ensure that the selected concept remains best under various scenarios and is not overly sensitive to minor changes in the criteria weights.

By applying this trade-off methodology to each subsystem, a consistent and objective approach is maintained throughout the design process. This ensures that each subsystem is evaluated thoroughly and that the overall system design is both compatible and efficient.

Reflection on Trade-Offs and Sensitivity Analysis
Trade-offs were consistently performed throughout the project, aiming to select concepts that are "the best" in their performance domain, while being integration-friendly with adjacent subsystems or components. Most of the time the trade-offs performed proved reliable and useful in the integrated design. Nevertheless, there were design choices that demanded iteration and that requested a trade-off reevaluation. Typically, to account for such flaws in the trade-off tables, sensitivity analysis was conducted at the end of the trade-off process. Additionally, sensitivity analysis was also employed in the form of evaluating the propagation of certain choices within other subsystems (i.e. the choice of the onboard processing unit impacts the amount of data that needs to be transferred, which further impacts communication channels that further reflects in the real-time collision avoidance capacity of the drone).

3.3. Project Management and Budgets

Project management tools such as workflow diagrams, Gantt charts, and technical budgets are used systematically to ensure a complete information flow between departments and track the resources available. Budgets in the design process, especially with respect to mass, power, and cost, are essential for managing resources effectively while ensuring project feasibility and performance [4]. Initially, these budgets establish targets based on project goals and constraints, incorporating larger margins to accommodate uncertainties. By the final stages, these margins are minimized, ensuring budgets align closely with actual requirements and product specifications. The margins chosen throughout the design project were selected according to [4], and are as follows: 30% for the preliminary design, 20% for the conceptual phase, and 5% for the final design.

As the project progressed, more and more design choices were made thus gradually leading to the convergence of the budgets. Budget management played an important role in the design process, ensuring that out-of-budget choices were either compensated by adjusting the neighboring architecture or mitigated by alternative choices that adhered to the established budget. The mass budget turned out to be most troublesome, mostly due to the software focus of the project. As algorithms became more complex, and with software integration close to the end, the hardware components were not strictly refined until the very end. Some iterations were performed, and thanks to contingency margins, they did not lead to mission failure.

From a scheduling point of view, budgets were updated along with the completion or extension of tasks. If delays appeared in the design process, they were addressed by reallocation of resources, adjustment of tasks, or, if needed, by extra hours to ensure timely progress. The initial budgets at the start of this design phase are introduced in Chapter 5, and further elaborated upon. After integration, the outcome is then presented in Chapter 15.

4: Product Development

This chapter presents an overview of the product development. Firstly, the functional diagrams are given in Section 4.1. The stakeholder requirements are presented in Section 4.2. Lastly, in Section 4.3 the project risks stemming from stakeholder requirements are described.

4.1. Functional Diagrams

To provide a breakdown of the functionality of Cargonaut, a Functional flow diagram and a Functional breakdown structure were generated. These diagrams combine the flow of operations and the type of operations to elucidate how Cargonaut would function within its work environment. Throughout the design process, these diagrams have also reflected the development of the design wherein changes were constantly made to ensure that the functionality that all departments provide was reflected. The most significant change was to detail the payload and logistics operations that the system would take, as these were undetermined and underdeveloped at the early stages of design and hence did not offer in-depth insight until this update. This analysis assures that all subsystems work coherently and the stakeholder needs are fulfilled from a functional perspective.

4.2. Stakeholder Requirements

Alongside the analysis of the functions the system must perform, a driving component in system design was the stakeholder requirements. During the design process, these requirements formed the foundation for developing mission and subsystem requirements. The stakeholder requirements are summarized in the table below.

Table 4.1: Stakeholder requirements, based on project guide (PG) or own analysis (OA)

Identifier	Stakeholder Requirement	Importance	Source
CRG-STK-01	The volume estimation of a single cargo item shall be performed in under 30 seconds.	Key	Customer (PG)
CRG-STK-02	The fleet of drones shall be able to process at least 1,000 cargo items per hour.	Key	Customer (PG)
CRG-STK-03	Each drone shall be able to carry out volume measurements both independently and via distributed planning during normal operations.	Driving	Customer (PG)
CRG-STK-04	Each drone should be able to autonomously switch between modes.	Key	Customer (PG)
CRG-STK-05	The measurement dimensions shall be accurate within 1 +/- % of precision.	Key	Customer (PG)
CRG-STK-06	The drones shall be able to operate in visibility conditions of 140-170 cdm^{-2}.	Key	Customer (PG)
CRG-STK-07	Each drone shall be able to recharge fully in 30 minutes.	Driving	Customer (PG)
CRG-STK-08	For each drone, the probability of collision with other drones or human operators shall be less than 1/10,000 operational hours.		Customer (PG)
CRG-STK-09	The drone system design and operations shall be non-damaging to the environment.	Driving	Regulations (OA)
CRG-STK-10	Batteries used in the system shall comply with Regulation (EU) 2023/1542 at all times.		Regulations (OA)
CRG-STK-12	The materials used to build the drone's structural frame shall be at least 70% recycled.		Sustainable Design (OA)
CRG-STK-13	A drone shall be able to carry a payload of at most 4 kg during normal operations.	Driving	Customer (PG)
CRG-STK-15	A single drone's largest dimension shall not exceed 50 cm during normal operation.	Key	Customer (PG)
CRG-STK-16	The production cost of a single drone shall not exceed 2,000 Euros.	Key	Customer (PG)
CRG-STK-17	The system shall provide an interface for human operators concerning basic functions.		Operations(OA)
CRG-STK-18	The drone system shall be maintainable.	Driving	Sustainable Design (OA)
CRG-STK-19	The drone system shall legally operate within regulations in the country of operation.	Key	Regulations (OA)
CRG-STK-20	The drone system shall be sustainable.	Key	Sustainable Design (OA)
CRG-STK-21	The drone system shall be able to operate within a warehouse.		Operations (OA)
CRG-STK-22	The data privacy of all associated parties shall be protected.	Key	Data Privacy (OA)
CRG-STK-23	During operations, the drone system shall not obstruct ongoing procedures in the warehouse.		Operations (OA)
CRG-STK-24	An operator shall be able to override autonomous decisions on command.	Key	Operations (OA)
CRG-STK-25	The drone system shall be adaptable to various warehouse configurations.	Driving	Operations (OA)
CRG-STK-26	The design of the product shall be conducted in 10 weeks.	Driving	Schedule (OA)
CRG-STK-27	A drone shall be able to achieve an operational cycle of at least 30 minutes.	Key	Customer (PG)

4.3. Risks

Project risks can be identified based on the stakeholder requirements. These encompass the main risk that the design does not meet all requirements. This overarching risk is then divided to consider the risk of not complying with individual requirements. The focus here is on stakeholder requirements which is presented in Table 4.2. To quantify the possible risk events, two metrics are used: the probability that the risk event will occur, and the severity of the event on the mission if it occurs. The metrics were quantified using numbers 1 to 5, as shown in Table 4.3. Each subsystem will assess its respective risks using the same method.

For the stakeholder requirements Table 4.4 presents the risk register: for each risk, likelihood and severity are displayed both before and after mitigation. The reasoning for the score and mitigation actions are also provided.

Based on the risk register presented in Table 4.4, a risk map was created displaying the risks, before and after mitigation. The risk map depicts the likelihood and severity of each of the risk events, and it can be seen in Figure 4.1. Likelihood is given on the x-axis and severity is plotted on the y-axis. Under such a representation, the high-risk zone is presented in the top right corner (indicated in red). Numerically these risks are quantified as having a risk score (that is the product of the risk's severity and likelihood) greater than 15.

Table 4.2: Project risks and associated requirements

ID	Risk	Associated requirement
RSK-STK-01	Volume estimation takes longer than 30 seconds	CRG-STK-01
RSK-STK-02	Less than 1,000 cargo items processed in one hour	CRG-STK-02
RSK-STK-03	Failure in volume measurements	CRG-STK-03
RSK-STK-04	Failure of autonomous mode switch algorithm	CRG-STK-04
RSK-STK-05	Failure to measure target volume within 1%	CRG-STK-05
RSK-STK-06	Failure to operate in visibility of 140-170 cdm^{-2}	CRG-STK-06
RSK-STK-07	Recharging cycle takes longer than 30 minutes	CRG-STK-07
RSK-STK-08	Two or more drones collide	CRG-STK-08
RSK-STK-09	Drone system damages the environment	CRG-STK-09
RSK-STK-10	Batteries do not comply with Regulation (EU) 2023/1542	CRG-STK-10
RSK-STK-12	Materials used are less than 70% recyclable	CRG-STK-12
RSK-STK-13	Drone exceeds payload limits	CRG-STK-13
RSK-STK-15	Drone exceeds dimension limits	CRG-STK-15
RSK-STK-16	Individual cost of drone exceeds the cost limit	CRG-STK-16
RSK-STK-17	User-interface issues impede operator effectiveness	CRG-STK-17
RSK-STK-18	Maintenance issues occur	CRG-STK-18
RSK-STK-19	Drone system fails to comply with regulations	CRG-STK-19
RSK-STK-20	Drone system is not sustainable	CRG-STK-20
RSK-STK-21	Drone system is incompatible with warehouse layout and/or infrastructures	CRG-STK-21, CRG-STK-25
RSK-STK-22	Data breaches leak private and sensitive information	CRG-STK-22
RSK-STK-23	Drone system causes operational disruptions	CRG-STK-23
RSK-STK-24	Glitches and/or communication errors prevent manual override	CRG-STK-24
RSK-STK-26	Design phase takes longer than 10 weeks	CRG-STK-26
RSK-STK-27	Battery inefficiency or power drain issues occur, operational cycle ends prematurely	CRG-STK-27

On the contrary risks with scores less than four form the low-risk zone. These risks do not require strict mitigation strategies as the relevance of the risk is low.

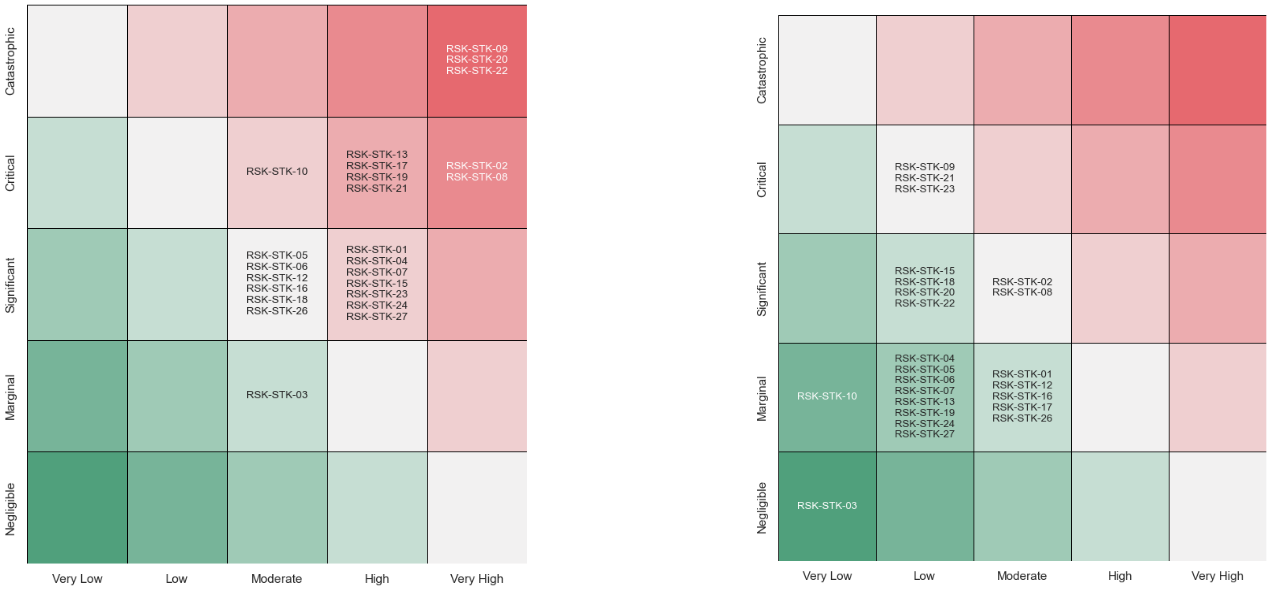
Table 4.3: Risk likelihood and severity criteria

(a) Likelihood criteria

Likelihood	Quantitative Measure
Very likely	5
Likely	4
Moderate	3
Low	2
Very low	1

(b) Severity criteria

Severity	Quantitative Measure
Catastrophic	5
Critical	4
Significant	3
Marginal	2
Negligible	1



	Very Low	Low	Moderate	High	Very High
Catastrophic					RSK-STK-09 RSK-STK-20 RSK-STK-22
Critical			RSK-STK-10	RSK-STK-13 RSK-STK-17 RSK-STK-19 RSK-STK-21	RSK-STK-02 RSK-STK-08
Significant			RSK-STK-05 RSK-STK-06 RSK-STK-12 RSK-STK-16 RSK-STK-18 RSK-STK-26	RSK-STK-01 RSK-STK-04 RSK-STK-07 RSK-STK-15 RSK-STK-23 RSK-STK-24 RSK-STK-27	
Marginal			RSK-STK-03		
Negligible					

(a) Risk map before mitigation

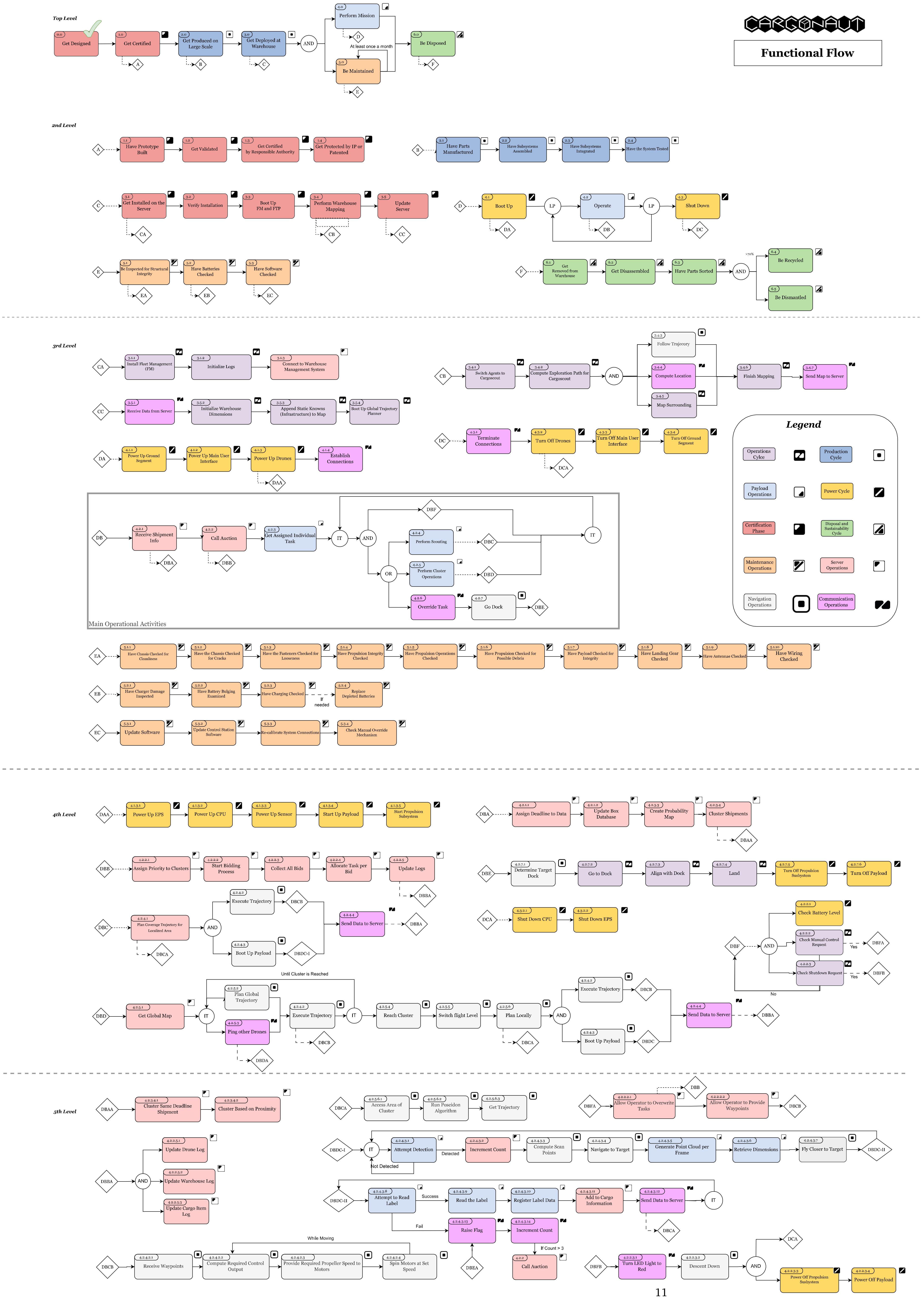
	Very Low	Low	Moderate	High	Very High
Catastrophic					
Critical		RSK-STK-09 RSK-STK-21 RSK-STK-23			
Significant		RSK-STK-15 RSK-STK-18 RSK-STK-20 RSK-STK-22	RSK-STK-02 RSK-STK-08		
Marginal	RSK-STK-10	RSK-STK-04 RSK-STK-05 RSK-STK-06 RSK-STK-07 RSK-STK-13 RSK-STK-19 RSK-STK-24 RSK-STK-27	RSK-STK-01 RSK-STK-12 RSK-STK-16 RSK-STK-17 RSK-STK-26		
Negligible	RSK-STK-03				

(b) Risk map after mitigation

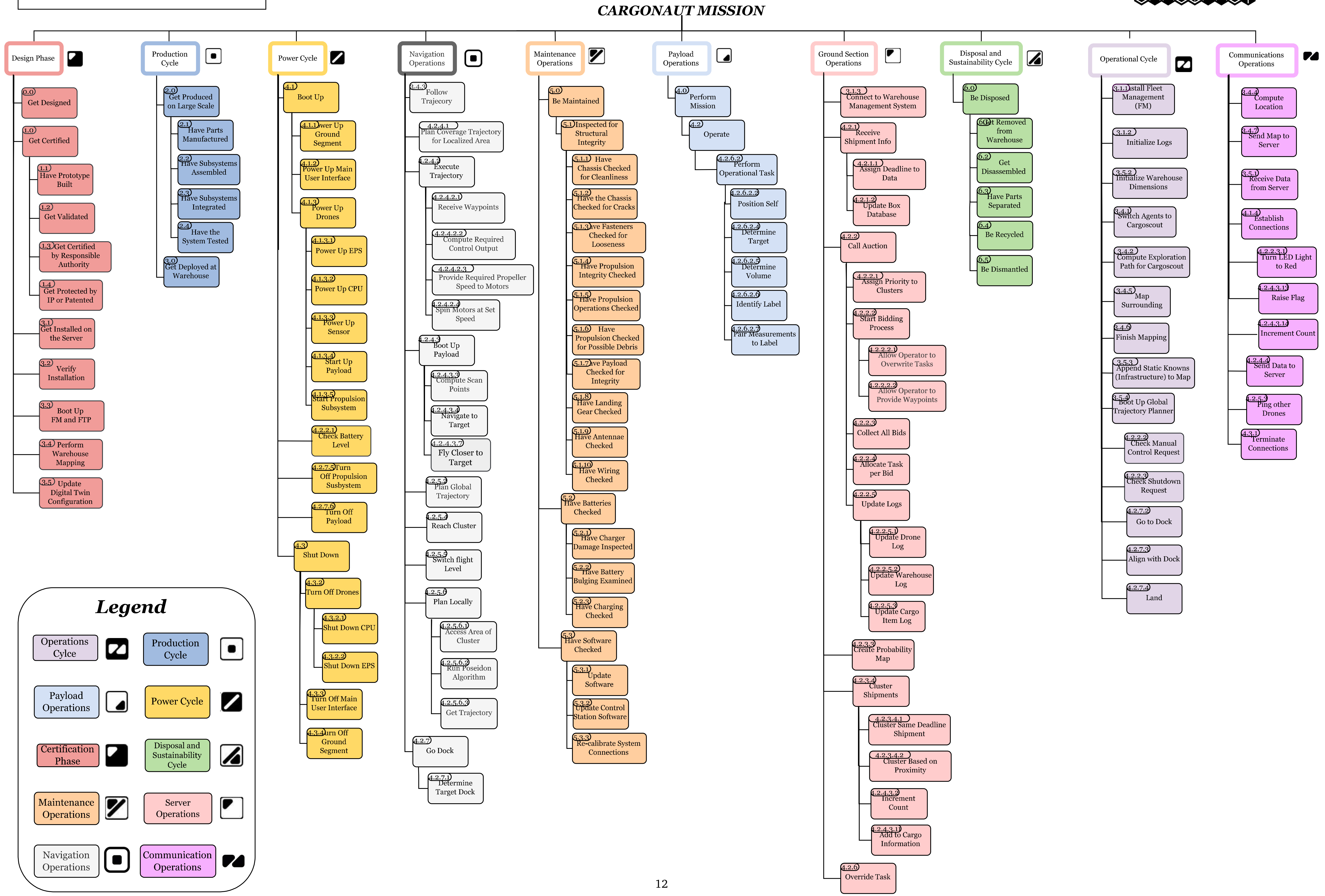
Figure 4.1: Risk maps before and after mitigation

Table 4.4: Risk register

ID	Reasoning and mitigation	Likelihood (Before)	Severity (Before)	Likelihood (After)	Severity (After)
RSK-STK-01	The risk depends on warehouse layout, cargo items' size and shape, and positioning: by optimizing and upgrading the system, the likelihood and severity are improved	4	3	3	2
RSK-STK-02	The risk depends on fleet and task management. By optimizing and streamlining the process both likelihood and severity decrease	5	4	3	3
RSK-STK-03	The risk depends on calibration and payload stability. By performing regular calibration and by optimizing stability, the likelihood and severity improve	3	2	1	1
RSK-STK-04	The risk depends on the algorithm implementation: by thorough testing and by implementing failsafe, the likelihood and severity decrease	4	3	2	2
RSK-STK-05	The risk depends on calibration, payload stability, and measurement redundancy. By performing regular calibration, optimizing stability, and ensuring correct task management, the likelihood and severity improve	3	3	2	2
RSK-STK-06	The risk depends on the sensors used. By ensuring the compliance with the requirement during the choice, the likelihood and sensitivity improve	3	3	2	2
RSK-STK-07	The risk depends on the battery choice and its performance. By investing in fast-charging batteries the risk impact is lowered	4	3	2	2
RSK-STK-08	The risk depends on the collision avoidance algorithm. By following the V&V plan and improving the communication between the drones, the risk is decreased	5	4	3	3
RSK-STK-09	The risk depends on the sustainability of the project and the material choice. By choosing materials that are environmentally-friendly and/or recyclable the likelihood and severity are improved	5	5	2	4
RSK-STK-10	The risk depends on the battery choice. By choosing suppliers that adhere with the regulation and by consistent monitoring, the risk is decreased	3	4	1	2
RSK-STK-12	The risk depends on the material choice. By using recyclable materials the risk impact is lowered	3	3	2	2
RSK-STK-13	The risk depends on the sensors and cameras used. By ensuring the use of margins, and the correct integration between structure and payload subsystem, the risk is lowered	4	4	2	3
RSK-STK-15	The risk depends on the design choices, especially propellers' dimensions. By choosing the subsystems accounting for the required size, the risk impact is improved	4	3	3	2
RSK-STK-16	The risk depends on the design choices of each subsystem. By using cost effective components, and by optimizing production, the severity is improved	3	3	3	2
RSK-STK-17	The risk depends on the design and implementation of the interface. By testing the interface and implementing user feedback, the likelihood and severity are lowered	4	4	2	3
RSK-STK-18	The risk depends on the maintenance. By ensuring a regular maintenance schedule performed by trained personnel, the likelihood and severity are lowered	3	3	2	2
RSK-STK-19	The risk depends on the design choices and system operation. By constantly monitoring the regulations and ensuring the system compliance, the risk is decreased	4	4	2	3
RSK-STK-20	The risk depends on the design choices made. By implementing a sustainable approach and ensuring a limited carbon footprint, the likelihood and severity are reduced	5	5	2	4
RSK-STK-21	The risk depends on the system operation and navigation. By generalizing the design and implementing modulation, the risk is reduced	4	4	2	3
RSK-STK-22	The risk depends on data transmission and storage. By implementing data encryption and performing regular security checks, the risk is lowered	5	5	2	4
RSK-STK-23	The risk depends on the system operation. By extensive testing in different environments and conditions, the likelihood and severity are reduced	4	3	2	2
RSK-STK-24	The risk depends on the algorithms and the manual override system. By creating multiple manual override mechanisms and manual control backups, the risk is lowered	4	3	3	2
RSK-STK-26	The risk depends on the organization and management of the project. By adhering to the project plan and allocating adequate resources to each task, the risk is reduced	3	3	2	2
RSK-STK-27	The risk depends on the battery discharge and on the power consumption of all the subsystems. By monitoring the power consumption and optimizing the battery usage, the risk is lowered	4	3	2	2


CARGONAUT
Functional Flow
Top Level
0.0 Get Designed
1.0 Get Certified
2.0 Get Produced on Large Scale
3.0 Get Deployed at Warehouse
AND
4.0 Perform Mission
At least once a month
5.0 Be Maintained
6.0 Be Disposed
A
B
C
D
E
F
2nd Level
1.1 Have Prototype Built
1.2 Get Validated
1.3 Get Certified by Responsible Authority
1.4 Get Protected by IP or Patented
2.1 Have Parts Manufactured
2.2 Have Subsystems Assembled
2.3 Have Subsystems Integrated
2.4 Have the System Tested
3.1 Get Installed on the Server
3.2 Verify Installation
3.3 Boot Up FM and FTP
3.4 Perform Warehouse Mapping
3.5 Update Server
CA
CB
CC
4.1 Boot Up
LP
4.2 Operate
LP
4.3 Shut Down
DA
DB
DC
5.1 Be Inspected for Structural Integrity
5.2 Have Batteries Checked
5.3 Have Software Checked
EA
EB
EC
6.1 Get Removed from Warehouse
6.2 Get Disassembled
6.3 Have Parts Sorted
AND
>70%
6.4 Be Recycled
6.5 Be Dismantled
3rd Level
3.1.1 Install Fleet Management (FM)
3.1.2 Initialize Logs
3.1.3 Connect to Warehouse Management System
3.4.1 Switch Agents to Cargoscout
3.4.2 Compute Exploration Path for Cargoscout
AND
3.4.3 Follow Trajecory
3.4.4 Compute Location
3.4.5 Map Surrounding
3.4.6 Finish Mapping
3.4.7 Send Map to Server
3.5.1 Receive Data from Server
3.5.2 Initialize Warehouse Dimensions
3.5.3 Append Static Knowns (Infrastructure) to Map
3.5.4 Boot Up Global Trajectory Planner
4.3.1 Terminate Connections
4.3.2 Turn Off Drones
4.3.3 Turn Off Main User Interface
4.3.4 Turn Off Ground Segment
DCA
4.1.1 Power Up Ground Segment
4.1.2 Power Up Main User Interface
4.1.3 Power Up Drones
4.1.4 Establish Connections
DAA
Legend
Operations Cylce
Production Cycle
Payload Operations
Power Cycle
Certification Phase
Disposal and Sustainability Cycle
Maintenance Operations
Server Operations
Navigation Operations
Communication Operations
DB
4.2.1 Receive Shipment Info
4.2.2 Call Auction
4.2.3 Get Assigned Individual Task
IT
AND
DBF
IT
4.2.4 Perform Scouting
DBC
OR
4.2.5 Perform Cluster Operations
DBD
4.2.6 Override Task
4.2.7 Go Dock
DBE
DBA
DBB
Main Operational Activities
5.1.1 Have Chassis Checked for Cleanliness
5.1.2 Have the Chassis Checked for Cracks
5.1.3 Have the Fasteners Checked for Looseness
5.1.4 Have Propulsion Integrity Checked
5.1.5 Have Propulsion Operations Checked
5.1.6 Have Propulsion Checked for Possible Debris
5.1.7 Have Payload Checked for Integrity
5.1.8 Have Landing Gear Checked
5.1.9 Have Antennae Checked
5.1.10 Have Wiring Checked
5.2.1 Have Charger Damage Inspected
5.2.2 Have Battery Bulging Examined
5.2.3 Have Charging Checked
If needed
5.2.4 Replace Depleted Batteries
5.3.1 Update Software
5.3.2 Update Control Station Software
5.3.3 Re-calibrate System Connections
5.3.4 Check Manual Override Mechanism
4th Level
DAA
4.1.3.1 Power Up EPS
4.1.3.2 Power Up CPU
4.1.3.3 Power Up Sensor
4.1.3.4 Start Up Payload
4.1.3.5 Start Propulsion Subsystem
DBA
4.2.1.1 Assign Deadline to Data
4.2.1.2 Update Box Database
4.2.3.3 Create Probability Map
4.2.3.4 Cluster Shipments
DBAA
DBB
4.2.2.1 Assign Priority to Clusters
4.2.2.2 Start Bidding Process
4.2.2.3 Collect All Bids
4.2.2.4 Allocate Task per Bid
4.2.2.5 Update Logs
DBBA
DBE
4.2.7.1 Determine Target Dock
4.2.7.2 Go to Dock
4.2.7.3 Align with Dock
4.2.7.4 Land
4.2.7.5 Turn Off Propulsion Susbystem
4.2.7.6 Turn Off Payload
DBC
4.2.4.1 Plan Coverage Trajectory for Localized Area
DBCA
AND
4.2.4.2 Execute Trajectory
DBCB
4.2.4.3 Boot Up Payload
DBDC-I
4.2.4.4 Send Data to Server
DBBA
DCA
4.3.2.1 Shut Down CPU
4.3.2.2 Shut Down EPS
DBF
AND
4.2.2.1 Check Battery Level
4.2.2.2 Check Manual Control Request
Yes
DBFA
4.2.2.3 Check Shutdown Request
Yes
DBFB
No
Until Cluster is Reached
DBD
4.2.5.1 Get Global Map
IT
4.2.5.2 Plan Global Trajectory
4.2.5.3 Ping other Drones
DBDA
4.2.4.2 Execute Trajectory
DBCB
IT
4.2.5.4 Reach Cluster
4.2.5.5 Switch flight Level
4.2.5.6 Plan Locally
DBCA
AND
4.2.4.2 Execute Trajectory
DBCB
4.2.4.3 Boot Up Payload
DBDC
4.2.4.4 Send Data to Server
DBBA
5th Level
DBAA
4.2.3.4.1 Cluster Same Deadline Shipment
4.2.3.4.2 Cluster Based on Proximity
DBCA
4.2.5.6.1 Access Area of Cluster
4.2.5.6.2 Run Poseidon Algorithm
4.2.5.6.3 Get Trajectory
DBFA
4.2.2.2.1 Allow Operator to Overwrite Tasks
DBB
4.2.2.2.2 Allow Operator to Provide Waypoints
DBCB
DBBA
AND
4.2.2.5.1 Update Drone Log
4.2.2.5.2 Update Warehouse Log
4.2.2.5.3 Update Cargo Item Log
DBDC-I
IT
4.2.4.3.1 Attempt Detection
Detected
Not Detected
4.2.4.3.2 Increment Count
4.2.4.3.3 Compute Scan Points
4.2.4.3.4 Navigate to Target
4.2.4.3.5 Generate Point Cloud per Frame
4.2.4.3.6 Retrieve Dimensions
4.2.4.3.7 Fly Closer to Target
DBDC-II
DBDC-II
4.2.4.3.8 Attempt to Read Label
Success
4.2.4.3.9 Read the Label
4.2.4.3.10 Register Label Data
4.2.4.3.11 Add to Cargo Information
4.2.4.3.12 Send Data to Server
IT
DBCA
Fail
4.2.4.3.13 Raise Flag
4.2.4.3.14 Increment Count
If Count > 3
4.2.2 Call Auction
DBEA
While Moving
DBCB
4.2.4.2.1 Receive Waypoints
4.2.4.2.2 Compute Required Control Output
4.2.4.2.3 Provide Required Propeller Speed to Motors
4.2.4.2.4 Spin Motors at Set Speed
DBFB
4.2.2.3.1 Turn LED Light to Red
4.2.2.3.2 Descent Down
AND
DCA
4.2.2.3.3 Power Off Propulsion Susbystem
4.2.2.3.4 Power Off Payload

Functional Breakdown



CARGONAUT MISSION

- Design Phase
  - 0.0 Get Designed
  - 1.0 Get Certified
    - 1.1 Have Prototype Built
    - 1.2 Get Validated
    - 1.3 Get Certified by Responsible Authority
    - 1.4 Get Protected by IP or Patented
  - 3.1 Get Installed on the Server
  - 3.2 Verify Installation
  - 3.3 Boot Up FM and FTP
  - 3.4 Perform Warehouse Mapping
  - 3.5 Update Digital Twin Configuration
- Production Cycle
  - 2.0 Get Produced on Large Scale
    - 2.1 Have Parts Manufactured
    - 2.2 Have Subsystems Assembled
    - 2.3 Have Subsystems Integrated
    - 2.4 Have the System Tested
  - 3.0 Get Deployed at Warehouse
- Power Cycle
  - 4.1 Boot Up
    - 4.1.1 Power Up Ground Segment
    - 4.1.2 Power Up Main User Interface
    - 4.1.3 Power Up Drones
      - 4.1.3.1 Power Up EPS
      - 4.1.3.2 Power Up CPU
      - 4.1.3.3 Power Up Sensor
      - 4.1.3.4 Start Up Payload
      - 4.1.3.5 Start Propulsion Subsystem
  - 4.2.2.1 Check Battery Level
  - 4.2.7.5 Turn Off Propulsion Susbystem
  - 4.2.7.6 Turn Off Payload
  - 4.3 Shut Down
    - 4.3.2 Turn Off Drones
      - 4.3.2.1 Shut Down CPU
      - 4.3.2.2 Shut Down EPS
    - 4.3.3 Turn Off Main User Interface
    - 4.3.4 Turn Off Ground Segment
- Navigation Operations
  - 3.4.3 Follow Trajecory
  - 4.2.4.1 Plan Coverage Trajectory for Localized Area
  - 4.2.4.2 Execute Trajectory
    - 4.2.4.2.1 Receive Waypoints
    - 4.2.4.2.2 Compute Required Control Output
    - 4.2.4.2.3 Provide Required Propeller Speed to Motors
    - 4.2.4.2.4 Spin Motors at Set Speed
  - 4.2.4.3 Boot Up Payload
    - 4.2.4.3.3 Compute Scan Points
    - 4.2.4.3.4 Navigate to Target
    - 4.2.4.3.7 Fly Closer to Target
  - 4.2.5.2 Plan Global Trajectory
  - 4.2.5.4 Reach Cluster
  - 4.2.5.5 Switch flight Level
  - 4.2.5.6 Plan Locally
    - 4.2.5.6.1 Access Area of Cluster
    - 4.2.5.6.2 Run Poseidon Algorithm
    - 4.2.5.6.3 Get Trajectory
  - 4.2.7 Go Dock
    - 4.2.7.1 Determine Target Dock
- Maintenance Operations
  - 5.0 Be Maintained
    - 5.1 Inspected for Structural Integrity
      - 5.1.1 Have Chassis Checked for Cleanliness
      - 5.1.2 Have the Chassis Checked for Cracks
      - 5.1.3 Have Fasteners Checked for Looseness
      - 5.1.4 Have Propulsion Integrity Checked
      - 5.1.5 Have Propulsion Operations Checked
      - 5.1.6 Have Propulsion Checked for Possible Debris
      - 5.1.7 Have Payload Checked for Integrity
      - 5.1.8 Have Landing Gear Checked
      - 5.1.9 Have Antennae Checked
      - 5.1.10 Have Wiring Checked
  - 5.2 Have Batteries Checked
    - 5.2.1 Have Charger Damage Inspected
    - 5.2.2 Have Battery Bulging Examined
    - 5.2.3 Have Charging Checked
  - 5.3 Have Software Checked
    - 5.3.1 Update Software
    - 5.3.2 Update Control Station Software
    - 5.3.3 Re-calibrate System Connections
- Payload Operations
  - 4.0 Perform Mission
    - 4.2 Operate
      - 4.2.6.2 Perform Operational Task
        - 4.2.6.2.2 Position Self
        - 4.2.6.2.4 Determine Target
        - 4.2.6.2.5 Determine Volume
        - 4.2.6.2.6 Identify Label
        - 4.2.6.2.7 Pair Measurements to Label
- Ground Section Operations
  - 3.1.3 Connect to Warehouse Management System
  - 4.2.1 Receive Shipment Info
    - 4.2.1.1 Assign Deadline to Data
    - 4.2.1.2 Update Box Database
  - 4.2.2 Call Auction
    - 4.2.2.1 Assign Priority to Clusters
    - 4.2.2.2 Start Bidding Process
      - 4.2.2.2.1 Allow Operator to Overwrite Tasks
      - 4.2.2.2.2 Allow Operator to Provide Waypoints
    - 4.2.2.3 Collect All Bids
    - 4.2.2.4 Allocate Task per Bid
    - 4.2.2.5 Update Logs
      - 4.2.2.5.1 Update Drone Log
      - 4.2.2.5.2 Update Warehouse Log
      - 4.2.2.5.3 Update Cargo Item Log
  - 4.2.3.3 Create Probability Map
  - 4.2.3.4 Cluster Shipments
    - 4.2.3.4.1 Cluster Same Deadline Shipment
    - 4.2.3.4.2 Cluster Based on Proximity
    - 4.2.4.3.2 Increment Count
    - 4.2.4.3.11 Add to Cargo Information
  - 4.2.6 Override Task
- Disposal and Sustainability Cycle
  - 6.0 Be Disposed
    - 6.1 Get Removed from Warehouse
    - 6.2 Get Disassembled
    - 6.3 Have Parts Separated
    - 6.4 Be Recycled
    - 6.5 Be Dismantled
- Operational Cycle
  - 3.1.1 Install Fleet Management (FM)
  - 3.1.2 Initialize Logs
  - 3.5.2 Initialize Warehouse Dimensions
  - 3.4.1 Switch Agents to Cargoscout
  - 3.4.2 Compute Exploration Path for Cargoscout
  - 3.4.5 Map Surrounding
  - 3.4.6 Finish Mapping
  - 3.5.3 Append Static Knowns (Infrastructure) to Map
  - 3.5.4 Boot Up Global Trajectory Planner
  - 4.2.2.2 Check Manual Control Request
  - 4.2.2.3 Check Shutdown Request
  - 4.2.7.2 Go to Dock
  - 4.2.7.3 Align with Dock
  - 4.2.7.4 Land
- Communications Operations
  - 3.4.4 Compute Location
  - 3.4.7 Send Map to Server
  - 3.5.1 Receive Data from Server
  - 4.1.4 Establish Connections
  - 4.2.2.3.1 Turn LED Light to Red
  - 4.2.4.3.13 Raise Flag
  - 4.2.4.3.14 Increment Count
  - 4.2.4.4 Send Data to Server
  - 4.2.5.3 Ping other Drones
  - 4.3.1 Terminate Connections

Legend

- Operations Cycle
- Production Cycle
- Payload Operations
- Power Cycle
- Certification Phase
- Disposal and Sustainability Cycle
- Maintenance Operations
- Server Operations
- Navigation Operations
- Communication Operations

5: Subsystem Interface Modeling

This chapter presents two interdisciplinary tools, used to assess interactions between subsystems. In Section 5.1, the N2 chart is introduced as a systems engineering tool to mitigate design complexity and showcase subsystem interfaces. Then, in Section 5.2, preliminary technical budgets for mass, power, and cost are presented.

5.1. Interface Identification: the N2 Chart

So far, an outline has been presented on the overhead organization of the project, as well as governing principles. Nonetheless, this definition lacks depth, as it does not truly showcase how the design of each particular system would be conducted. It is difficult to come up with a general outline since different tools and approaches are considered per department. In spite of this, a fully developed product shall showcase the integration of both software and hardware aspects. The entire design process consists of layers and encompasses continuous interactions between all subsystems. To assist in creating an overarching map with relevant interfaces, the team adapted the N2 chart as a systems engineering tool. Essentially, it is a square matrix, depicting the flow of information between interconnected elements within subsystems [5]. Whereas the systems are gathered in the main diagonal, the remaining cells present their interaction, with horizontal outputs and vertical inputs. As a visual aid, it provides a clear image of key interfaces and assists in managing them, allowing to set up an effective concurrent engineering environment. Via the chart, it can be easily seen how changing a particular parameter affects other subsystems, and why design loops might be necessary for convergence. Consider for instance the interaction between the flight performance and control subsystems: whereas the maximum rotational speed of the propellers is important for modeling the flight dynamics, the required thrust is then translated into a throttle setting by the flight controller.

For the design of Cargonaut, nine main subsystems were considered and listed in Table 5.1, along with their main points of interaction. The subsystem design is showcased in deeper detail in the next chapters, from 6 to 14, with the final integration presented in Chapter 15.

5.2. Technical Resource Budget Breakdown

Another technique that was utilized, based on Technical Performance Measurement (TPM) and adapted to the mission objective, was that of developing technical budgets. Essentially, each budget presents an upper bound on the quantity of interest - either mass, power, or cost - that a subsystem needs to comply with. This assures that the subsystems set requirements that do not overlook these budgets, but also consider these upper limits during their design. Overall, options that violate upper limits could have a detrimental effect on the product capabilities, and shall not be considered as feasible. Unless a thorough analysis of the N2 chart and explicit approval from the system engineer is provided, the budgets shall not be exceeded.

Furthermore, each budget also includes a contingency margin, based on the level of maturity of the given phase. The dedicated contingencies are decreased at each subsequent stage of the design process, based on the resource allocation and margin regulations of the American Institute of Aeronautics and Astronautics [4]. As the final design begins, it relies on initial margins of 20%, and presented in this section. After the product is complete, the budgets shall be updated with the exact values, leaving a 5% margin that accounts for non-ideal manufacturing and design tolerances.

5.2.1. Initial Mass budget

The phase of conceptual design was finalized with a mass budget, given in Table 5.2. The subsystems that are expected to mainly contribute to the overall weight are listed, along with the allocated resources. It can be seen that the mass of the power subsystem, mostly based on the battery, is roughly half of the overall UAV mass. Despite being higher than typical drones on the market, this value was deemed reasonable, taking into consideration the strict endurance requirement. Also, the navigation and control subsystems were coupled, both governed by the selection of a flight controller and an Electronic Speed Controller (ESC).

Table 5.2: Mass Budget per subsystem, including contingency

Subsystem	**Mass [g]**	**Contingency: 20%**
Propulsion	200	240
Power	600	720
Navigation and Control	25	30
Communications	30	36
Payload and Operations	90	108
Structural frame	250	300
Total:	**1,195**	**1,434**

Table 5.1: N2 chart, depicting Cargonaut's interface definitions; subsystems are indicated on the main diagonal, with horizontal outputs and vertical inputs

Flight Performance	Dynamic loading analysis			Acceleration Rotors maximum rotational velocity Wake behaviour		Pointing jitter		Chassis aerodynamic analysis
Propeller geometry Static performance Lift and drag polars	**Propulsion**	Propulsive efficiency Minimum T/W ratio Motor connection type Motor power consumption and voltage		Propulsive efficiency Motor maximum rotational velocity		Vibrations	Noise levels	Rotor and motor size Vibrations, Noise Applied forces Operational T° range
Battery mass Battery capacity	Operational voltage and current	**Electronics**					Battery level	Battery size Battery mass Operational T° range
Velocity for measurement taking	Speed requirements	Power consumption and voltage of navigation instruments	**Navigation**	Localization accuracy and frequency Planned trajectory	Measured data Data rate of navigation instruments	Drone location	Environment mapping Localization	Navigation instruments specifications
	Input signals Required thrust Target velocity	Power consumption and voltage of control unit	Actuator signal	**Control**	Current state and position		Time to stabilize drone	Control unit specs
		Power consumption and voltage of communication instruments Data handling hardware	Instrument operational frequency Target update		**Comms & Data Handling**	Transmit commands	Task status Manual commands	Communication instruments specs Required placement, orientation
Design velocity for taking measurements	Stability requirements	Power consumption and voltage of payload	Pointing accuracy requirements Payload moment of inertia	Pointing accuracy requirements	Data rate of payload Measurement data gathered	**Payload**	Measurement duration Total count of cargo items	Payload specifications Payload sensitivity Field of View
	Desired altitude (flight level)		Operational mode Target location	Trajectory waypoints	Task distribution Operational mode	Task assignment Scanning trajectory	**Ops & Logistics**	
Frame shape & size	Rotor and motor location Maximum propeller size Total mass	Component location Total mass	Sensors location Actuators location	Frame size Frame topology	Hardware location			**Structures & Materials**

5.2.2. Initial Power budget

Only a number of subsystems require any power input, as indicated in Table 5.3. The motors, which would turn spin the propellers according to the throttle setting, are allocated the highest proportion of the overall budget. The communication subsystem shall facilitate both transmitting and receiving signals, for which 10 W in total are allocated. Power requirements regarding instrumentation for the payload and the navigation and control subsystems stem from market analysis and are not significant. It is expected that a more detailed market analysis of Commercial Off-The-Shelf (COTS) components and further development of the Electronics subsystem will allow for a more comprehensive final modeling.

Table 5.3: Power Budget per subsystem, including contingency

Subsystem	**Power [W]**	**Contingency: 20%**
Propulsion	205	246
Navigation and Control	5	6
Communications	10	12
Payload	5	6
Total:	**225**	**270**

5.2.3. Initial Cost Budget

The third budget includes the expected cost per subsystem, based mostly on COTS components. It considers all departments, apart from flight performance, which does not explicitly require the acquisition of any extra components. More resources are allocated to subsystems that require specific instrumentation and directly influence the ability of the drone to perform its task and take measurements. Possible implementation of hot-swapping[1] also implies that two batteries might be needed per UAV, hence the higher budgeted cost for the power subsystem. Nonetheless, design would attempt to avoid this scenario, due to its unsustainable nature. Options for the GS, incorporating both charging and docking, have not been thoroughly examined. Thus, it has the highest proportion of the overall budget allocated to it, considering how it might need to be designed from scratch to meet the particular mission needs, with its value left as a To be confirmed (TBC).

After the design of the product is finalized, further analysis will be necessary, before a market price is established. Maintenance, operations support, manufacturing, market competitiveness, and profitability are only a few of the considerations that shall be taken into account before the market price is decided upon.

Table 5.4: Cost budget per subsystem, including contingency and VAT

Subsystem	**Cost [€] FY2024**	**Contingency: 20%**
Propulsion	120	144
Power	200	240
Navigation and Control	200	240
Communications	100	120
Payload and Operations	200	240
Structural frame	120	144
Data processing	80	96
Ground station[2]	400 TBC	480
Total:	**1,420**	**1,704**

[1] Hot-Swapping would entail exchanging the empty battery of the drone for a full one with the vehicle still operational, albeit docked at the station.

[2] Value not linearly scaling with number of drones, and not definite. Deviations are not unexpected, due to lack of research during conceptual design phase.

6: Operations and Logistics

In this chapter the framework of the operations subsystem is presented. The goals and functions of the subsystem are given in Section 6.1. This is followed by a consideration of the requirements and risks in Section 6.2. The solution approach is given in Section 6.3. Subsequently, the fleet management system and fleet trajectory planning system are presented in Section 6.4 and Section 6.5 respectively. The operational cycle is given in Section 6.6, and the required infrastructure considerations are given in Section 6.7. To conclude the subsystem framework, a fleet size analysis is performed in Section 6.8. Verification and validation procedures are elaborated in Section 6.9. A fleet motion simulation is presented in Section 6.10. Lastly, future recommendations are presented in Section 6.11.

6.1. Goals & Functions

The air cargo loading problem arises from a lack of complete and reliable information about the size of the shipments that travel and are handled in cargo warehouses. From an operational point of view, this additional information would enhance the warehouse management system, enabling more efficient packing algorithms and resulting in higher load factors for Unit Load Device (ULD)s. Accordingly, the goal of Cargonaut is to:

Provide efficient fleet management and planning for improved cargo handling warehouse operations.

With this objective in mind, several system functions are naturally identified for the system. Besides being safe, sustainable, and flexible, the drone system shall improve cargo handling operations of the warehouse management and aid human workers in processing incoming cargo. Due to extended coverage considerations, a fleet of drones is deemed superior to other options and, subsequently, designed for the mission statement. To craft an operational cycle for the drone system, it is crucial first to understand the exact context in which such a solution can be integrated and then apply a level of abstraction and assumptions to the problem domain to reduce it to a defined problem statement. Section 6.1.1 will introduce a typical warehouse environment and present the assumptions required for the optimization problem Cargonaut is trying to solve. Furthermore, technical considerations are given in Section 6.1.2.

6.1.1. Problem Domain

Warehouses worldwide largely differ in terms of surface area and interior layout. Nevertheless, a common feature of all these facilities is the flow of shipments through the warehouse. It all starts with trucks unloading their cargo at the entrance of the warehouse. There, the packages undergo an acceptance process and are relayed to the warehouse by forklifts or human operators. Once accepted into the facility, the boxes are identified and distributed to their target location, which can be: the storage, the breakdown area, or the build-up area [1]. At shelves, the shipments can be stored for several days before being loaded on a freighter, while in the build-up area, they are already getting prepared to be shipped. Additionally, in the breakdown area, the shipments are disassembled and individually registered in an inventory and further distributed to other areas of the warehouse. Since this is just the general layout of a cargo warehouse, a case study has been chosen for a more accurate definition of the problem: *the KLM Cargo Export Warehouse at Schipol Airport*. In Figure 6.1 a schematic of the KLM Export Warehouse is presented. All the information on the case study has been acquired through an external connection: Ir. Wouter van der Wal - Operations Research Consultant at KLM and is presented below.

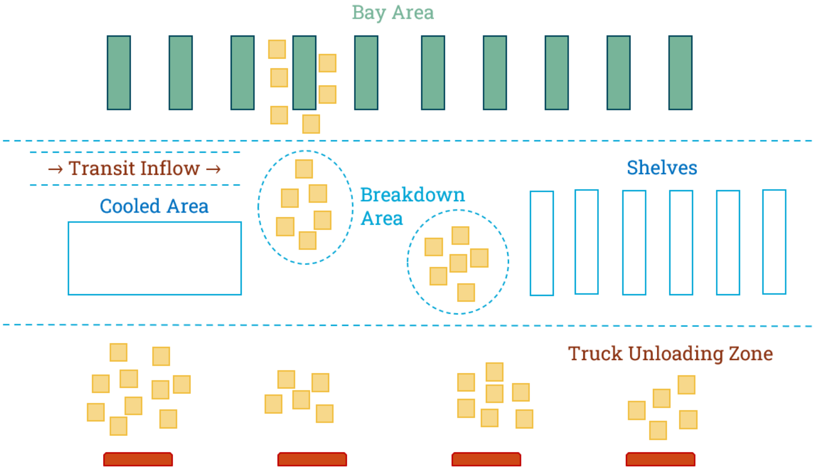


Figure 6.1: Reference warehouse layout

[1] Storage and Cargo Handling https://www.keyence.eu/ss/products/auto_id/logistics/role/function-safekeeping.jsp. Accessed 12 June 2024.

As part of the KLM Export Warehouse facility, three main areas are identified: the unloading zone, the main warehouse area, and the bay area. The Truck Unloading Zone is marked with red and represents the area where cargo items are introduced into the warehouse through the unloading doors. This area is approximately 100 x 40 m. The main warehouse area, delimited by blue dotted lines, is further divided into storage areas (shelves and cold storage room(s)), breakdown pits, and buildup areas/packaging bays). Next to that, the main warehouse accommodates for a second flow of packages - the transiting cargo items coming from the import warehouse. Finally, the Bay Area (represented in green) is the location where packed cargo items are loaded onto ULDs and configured by workers in such a way that their load factor is maximized. From this configuration, three areas of interest have been identified for the Cargonaut fleet, along with their characteristics:

- The unloading areas: localized areas that have a constant inflow of shipments from one or multiple unloading doors.
- The breakdown pits: localized areas of around 6x10m where individual boxes are constantly moved around, unpacked, and restacked.
- The bays/build-up area: localized areas that receive individual or packaged cargo items and where the ULD loading scheme is performed.

6.1.2. Technical Considerations

In the light of the discussed case study, the limitations of the problem were identified and some technical considerations were selected in order to make the optimization problem more structured.

- Two to five trucks unload shipments every 10 to 20 minutes at various doors throughout the day.
- The cargo items are on the floor after acceptance for between 10 and 20 minutes, the time in which they are processed in the warehouse management system and their next location is defined. This window of time will henceforth be referred to as the *operational window*.
- During an operational window, the location of cargo items placed on the floor does not change.
- The volume of each cargo item falls in the range $0.2\,\mathrm{m}^3$ to $1.5\,\mathrm{m}^3$.
- Each truck delivers between 10 and 120 cargo items.
- A cargo item may contain one or more boxes.
- The shipments are placed in localized areas around the doors in the unloading zone.
- Each cargo item or shipment may have multiple labels attached to them with different information. The drones will attempt to read the first identified label. If no relevant information is read, a flag is raised and the process continues.

With the warehouse layout in mind, the problem can either be analyzed as an integral logistics challenge or split into the areas of the warehouse mentioned above. For the timeframe of the project, the operations are developed mainly considering the truck-unloading area. This implies that typically the cargo items will be scanned and measured in the form the forwarders deliver them (might be a group of boxes). Nevertheless, this design space can easily be extended to the full warehouse by imposing and accounting for some new constraints such as shipment movement.

6.2. Requirements & Risks

From the global objectives of the product and based on the inter-dependencies between the subsystems, a list of subsystem requirements can be postulated for the operations subsystem. The list of all requirements relevant to the operational aspect is given in Table 6.1.

Table 6.1: Requirements for Operations subsystem

Identifier	Operations System Requirements	Associated Risk ID
CRG-OPS-01	Each drone shall keep a minimum distance of at least 100 cm from the rest of the fleet during operations.	RSK-OPS-01
CRG-OPS-02	Each drone shall be equipped with a trajectory planning algorithm.	RSK-OPS-02
CRG-OPS-04	A server shall facilitate the communication between the drone system and the human operators via a digital interface.	RSK-OPS-03
CRG-OPS-05	Each drone shall have a noise level that complies with the EASA regulations.	RSK-OPS-04
CRG-OPS-06	Each drone shall keep a minimum distance of at least 100 cm from human operators.	RSK-OPS-01
CRG-OPS-07	The operator shall be provided with a manual override button to take control of the tasks of the drone system.	RSK-OPS-03
CRG-OPS-08	The operator shall be provided with a kill switch that can turn off the drone system at any point during operations.	RSK-OPS-03
CRG-OPS-09	The operations logic shall activate an automatic stop when signals from other fleet members or the server are not received.	RSK-OPS-01
CRG-OPS-10	The operator interface shall be positioned in such a way as to be safely operated during normal operations, including fault modes.	RSK-OPS-05
CRG-OPS-11	The operator interface shall be positioned outside of danger zones except for the event where the kill switch is activated, during normal operations, including fault modes.	RSK-OPS-05
CRG-OPS-12	Operations shall ensure that in case the system is stopped, it does not drift away from the stopping position in a way that presents a hazard, during normal operations, including fault modes.	RSK-OPS-01
CRG-OPS-13	Each drone shall be able to switch between operational modes when requested by the server.	RSK-OPS-01

Coupling the requirements of the system, a list of associated risks is formulated to account for the dangers of the operational cycle. The risks that have a high coupled likelihood ×severity rating are identified and mitigation strategies are devised accordingly. These technical operational risks, along their mitigation methods, are presented in Table 6.2.

Table 6.2: Risks for Operations subsystem

Identifier	Operational Risk	Likelihood	Severity	Mitigation
RSK-OPS-01	Drones get into each other's control sphere or too close to other system elements and eventually, crash.	3	3	Collision Avoidance algorithm ensures that a distance of 1 m is maintained with respect to all dynamic objects.
RSK-OPS-02	Trajectories of drones intersect and drones collide with each other or the environment.	2	2	Onboard re-planning is conducted to ensure nonconvergent paths.
RSK-OPS-03	Environment gets damaged as operators cannot stop or intervene in the activity of the drones.	2	2	Implement automated protocols that trigger drone intervention or landing in case of unexpected environmental conditions or system malfunctions.
RSK-OPS-04	The noise levels of the drone damage the health of the human operators. Non-compliance with noise regulations could lead to legal issues.	2	2	Human operators shall have access to devices/equipment to protect their hearing.
RSK-OPS-05	Drones collide with and injure human operators.	2	2	Collision Avoidance algorithm ensures that a distance of 1 m is maintained with respect to all dynamic objects.
RSK-OPS-06	Sensor errors could lead to misjudgment of the required distance.	2	2	Establish a routine schedule for calibrating sensors to ensure their accuracy over time.
RSK-OPS-07	The manual override or kill switch fails to function when needed.	1	5	A redundant kill switch will be provided and regular maintenance will be performed.

6.3. Solution Approach

In the broadest sense, the logistics of the mission operation are characterized by an optimization problem. This problem refers to determining the optimal manner of organizing the fleet such that the mission objective is satisfied under the established constraints and the cost function is minimized. The exact constraints are a function of the warehouse geometry. Hence, the domain and constraints of the problem are investigated in the Section 6.3.1, Section 6.3.2 and Section 6.3.3.

6.3.1. Optimization Domain

This optimization problem can be split into two domains: the management of the fleet and the trajectory planning to reach the targets. These two domains are not independent, however, and can be treated as a decoupled system. As a result, for simplicity, the scope of the domains will be described separately. If they are treated as independent problems, during integration the compatibility of the two domains should be taken into account. The scope is given as:

1. **Fleet management (FM):** this refers to determining the optimal organization of the fleet to perform the mission. Therefore it deals with problems such as determining how many drones should be sent to measure a target, and which drones from the fleet should be sent to which target (the closer a drone is to a target the more likely it will be called to scan said target).
2. **Fleet trajectory planning (FTM):** this refers to the path each drone must take to reach its target location. As additional objective drone traffic could be minimized.

6.3.2. Geometry of Problem Space

The geometries of warehouses vary greatly depending on the exact warehouse. However, in general, four types of objects can be expected, namely:

- **Static Knowns:** static objects that are known from the bare structure of the warehouse, such as the shelves, and the location of external lights.
- **Static Unknowns:** objects that do not move, and the location of these objects varies under the circumstances. To exemplify, the new cargo items that arrive to the warehouse during the day are not placed on the shelves or machinery that is temporarily stationary.
- **Dynamic Knowns:** dynamic objects whose location can be inferred or are known by the system, such as the drones in the warehouse.
- **Dynamic Unknowns:** objects that can move, however, the motion of these objects is unpredictable and hence the location of these objects is unknown, such as the humans in the warehouse.

6.3.3. Logistics Strategies

Logistics implementation strategies can be loosely divided into two categories, integrated approaches and sequential approaches. Within an integrated approach, the two domains of the optimization problem are solved in parallel. On the contrary, sequential approaches tackle the problem in series, that is, the problem of fleet management is solved and used as a priori for solving the problem of fleet trajectory planning.

Furthermore, the architecture of the solution can be divided into two categories, namely centralized and distributed frameworks. Within a centralized architecture, there exists a hierarchy in the system. Drones are assigned tasks and paths by a central body that has real-time knowledge of all drones. Drones can communicate amongst themselves, however, the management of the fleet is not done by them [6]. On the contrary, in a distributed architecture the fleet of drones is homogeneous without a leader. Decisions are made through the information gained from local interactions between drones and the dynamic environment. As a result, these decisions rely on a collective consensus amongst all drones [6].

The integration strategy is independent of the architecture and hence, each combination of integration strategy is compatible with any choice of architecture. Under a sequential approach, hybrid strategies can also be explored. As a result, there are a total of six approaches, namely:

- Integrated approach with a single algorithm for centralized fleet management and planning.
- Integrated approach with a single algorithm for distributed fleet management and planning.
- Sequential approach with centralized fleet management and centralized planning.
- Sequential approach with distributed fleet management and distributed planning.
- Sequential approach with centralized fleet management and distributed planning.
- Sequential approach with distributed fleet management and centralized planning.

Table 6.3 explains the theory behind each combination.

Table 6.3: Possible architectures for operations and logistics

Architecture	**Fleet Management**	**Fleet Trajectory Planning**	**Integrated**
Centralized	Management of the tasks is conducted on a central server.	Trajectory planning for each drone is conducted on a central server.	Management and trajectory planning are integrated and conducted on a central server.
Distributed	Management of the tasks is conducted on each drone.	Trajectory planning for each drone is conducted on its platform.	Drones interact with the environment and themselves to manage tasks and determine trajectories.

Integrated & Centralized and Sequential & Distributed

Integrated centralized approaches require a centralized unit capable of making all decisions simultaneously. In such cases, even small localization errors can result in functionality failure. Due to the lack of robustness of the system, it is not further investigated. Sequential distributed approaches require significant coordination overhead and complete synchronization within the system. Such methods are extremely complex to implement in practice and hence, are not further considered.

Integrated & Distributed

In an integrated distributed approach, the fleet is responsible for determining targets, navigating towards these targets, and organizing themselves to scan targets. The path each drone must take to reach targets and the location of these targets are not known as a priori; rather, the drones distribute themselves as a collective body to solve the problem. On a high level, this type of approach makes use of the non-Markovian properties of the problem (each drone has limited knowledge of the environment and system) [6, 7].

Under such an architecture, the system is completely autonomous and requires little to no input from an external user. Such high levels of autonomy could hinder the accessibility of the system to humans, introducing challenges for monitoring the system. In addition, because a swarm is capable of processing and distributing data without human oversight, unintended exposures of data could occur. These exposures can be reduced through cyber-counter measures, however, their effectiveness is limited for highly autonomous systems capable of understanding and interacting with complex, dynamic environments [7]. Furthermore, requirement **CRG-SYS-OPS-07** also becomes difficult to comply with, unless a redundant manual system is included in the architecture. Based on these concerns, this solution is deemed impractical.

Sequential & Centralized and Sequential & Hybrid
Sequential centralized approaches split the optimization problem into two independent problems; fleet management and fleet trajectory planning. Both these problems are solved in a centralized manner such that the drones are given information on the tasks they must perform and the path they must take to perform their tasks. Such a system requires that the central body has complete knowledge of the location and status of the drones at all times.

Within sequential hybrid approaches, fleet management and fleet trajectory planning do not use the same architecture. This is beneficial as it offers more flexibility and allows the system to exploit the benefits of the different approaches. A centralized approach is useful for fleet management. Although, this is more computationally intensive with cloud computing the demands are limited [6]. Whereas, a distributed architecture is more suitable for trajectory planning with large fleet sizes [8]. Under these considerations, to offer more flexibility a sequential approach making use of centralized fleet management and distributed fleet trajectory planning is chosen.

6.4. Fleet Management

The problem of fleet management can be split into the description of the problem, the determination of task locations, the method of task allocation, and the chosen architecture. This is explored in Section 6.4.1, Section 6.4.2, and Section 6.4.3 respectively.

6.4.1. Problem Description

The fleet management system is responsible for keeping track of the drones, determining the tasks the fleet should complete, and allocating the fleet to the tasks. The path the drone must follow to reach to its task is not considered. Here, “task” refers to the scanning of an indivisible item of cargo in the warehouse. By using a centralized architecture to solve this problem, it can be assumed that the location of the drones is known as a priori for the fleet manager. The following paragraphs investigate the constraints and the applicable cost function for the optimization problem.

The main constraints stem from the nature of the problem space and the subsystem requirements. The drones in the fleet can each be characterized by a state. Three basic states can be identified namely: *ACTIVE*, *IDLE*, and *CHARGING*. With regards to the problem space, it is given that the cargo items remain at the unloading area for 10-20 minutes and that the location of these cargo items is unknown. As the location of the cargo items that are to be scanned is unknown, the fleet must be capable of determining the locations. From **CRG-STK-02** it is given that the fleet must be capable of scanning 1,000 boxes in an hour. However, this throughput might be lower than what would be required to scan all boxes within the span of an operational window. In such a case, the cost function for the optimization function can be to minimize the number of boxes that remain unscanned at the end of an operational window.

Under such considerations, the aim of the fleet management system can be phrased as follows:

To determine the organization of the fleet such that the number of cargo items that remain unscanned at the end of an operational window is minimal provided the fleet scans a minimum of 1,000 boxes per hour.

The solution to this problem is two-fold. Firstly the location of the cargo items must be determined, along with an estimate of the time they will remain in the unloading zone. Secondly, the fleet should be assigned to complete the tasks. These two components are coupled yet they feed into each other sequentially. This is because the last component cannot be considered without the first. As a result, the two components can be solved independently. The implementation of this sequential, management system, from a software perspective can be done in several ways. The possible algorithms are investigated separately for the two components.

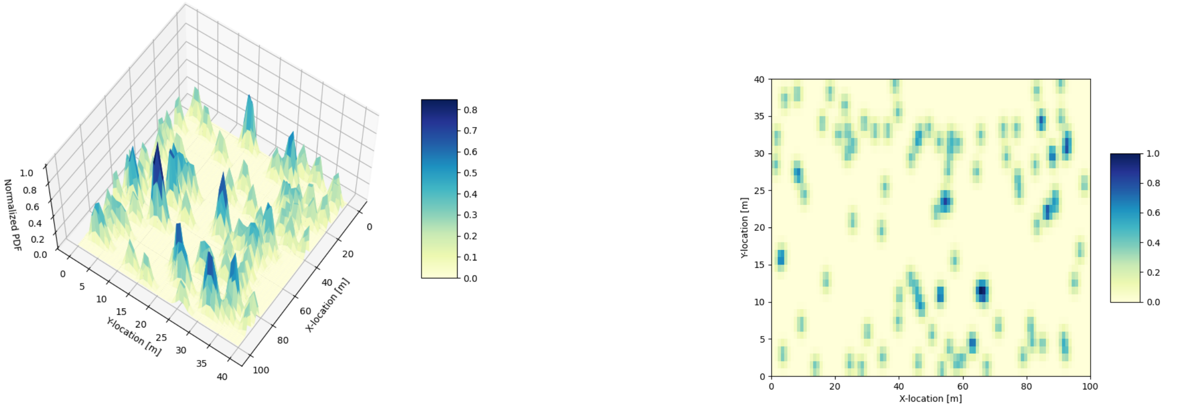
6.4.2. Determination of Task Locations

Since the exact number and size of cargo items to be measured are unknown, the system must be capable of inspecting and predicting the resources required to carry out a task. In addition, the location of these tasks in the warehouse must be determined. In every operational window, the cargo items will placed in different locations. Hence, the management system must adapt to the variable environment. This can be done by the *Cargoscout* mode of the drones, which allows to scan the environment at the instance a truck unloads. To ensure that the system remains flexible, this will be an operating mode a drone can take on. As a result, there are two operating modes (this is shown in Table 6.4), and four possible states namely, *ACTIVE*, *IDLE*, *CHARGING*, and *SCOUTING*.

Table 6.4: Operational modes

Modes	Function
Cargoscout	Performs a preliminary scout of the localized area and relays the gained information of the local surroundings to the server.
Cargonaut	Performs the mission task of determining target topology and sends the obtained measurements to the server.

While scouting, Cargoscout will be counting the static unknowns it has observed within the vicinity and noting down their approximate location using Simultaneous Localization And Mapping (SLAM) sensors on board. Each location is noted down multiple times for redundancy purposes. This allows for a probabilistic global map to be created for the operational cycle to allow for clustering and task allocation. This probabilistic map can be visualized on Figure 6.2.



(a) Probabilistic map of the location of the cargo items

(b) Heat map representation of the probabilistic map

Figure 6.2: Exemplary probabilistic map

The probabilistic map is generated taking into account the redundant measurements per box. This is done by passing a Gaussian kernel over the warehouse grid to evaluate the probability of cargo items' existence at the grid block. The kernel gives a higher probability to location data that are closer together, indicating that there is a high chance that a cargo item exists there. On the contrary, areas in the grid with no location data would be marked with 0 probability of a cargo item existing.

6.4.3. Task Allocation

The problem of task allocation is a complex optimization problem. On a high level, the problem can be classified by resource demand, the extent to which temporal planning is considered, inter-task constraints, and the possibility for specializations. These four categories are described below.

Resource Demand

Resource demand refers to whether one drone or multiple drones are assigned to the same task. To determine whether there exists a need for allocating multiple drones to one task, the following is considered. If multiple drones scan the same target, the time taken to scan one target could be reduced. However, this is not valid as then additional time is required to combine the images, as the location at which each drone scanning the target took its respective image must be assembled correctly. Based on the size of the target, **CRG-OPS-01** could be hard to comply with as the drones would have to be close to each other. In addition, it would require each drone to send the images it took to either a central drone or the server. Based on these considerations, it is concluded that for each task a single drone will be assigned. Considering multiple drones will not be allocated to scan the same cargo item, there is little need for multiple drones to be assigned to the same task.

Extent of Temporal Planning
The extent of temporal planning refers to whether horizon-based or myopic planning will be considered. Under a horizon-based approach, future tasks (tasks that will occur at time $t+1, t+2, ..., t+n$) are taken into consideration when assigning the tasks at time t. This approach can improve the efficiency of task allocation as the number of transitions and idle drones is minimized. However, algorithm complexity increases rapidly [9]. Hence, instantaneous assignment will be used.

Inter-task Constraints
Inter-task constraints include dependencies and/or requirements that affect how the tasks are allocated relative to one another. Examples of such constraints include sequential, resource, mutual exclusivity, and proximity constraint [9]. In the context of scanning boxes, it is desirable that cargo items close to each other are scanned by the same drone - this imposes proximity constraints. In addition, it is desired that boxes that have a higher chance of being moved (that is they are closer to the end of their operational window) are given priority - this imposes sequential constraints. This can done in the following manner: once an item has been identified by the cargoscout (and registered in the database) it is assigned an operational deadline. As the warehouse is aware of when it expects shipments, assigning the deadline to an identified item can be done.

Specialization Possibilities
The inclusion of specialization possibilities refers to having a heterogeneous fleet in which drones show varying levels of proficiency for different tasks. The methods by which drones complete tasks would be different in such a fleet. As the fleet is designed to be flexible and scalable, it is essential that the fleet of drones can be bought in any number rather than having a designated configuration. This minimizes the need for excessive resources, allows for more efficient deployment based on specific warehouse requirements and thereby offers a more sustainable solution. Hence, specializations are not considered.

Based on these considerations the task allocation problem is single agent - single task, instantaneous assignment with sequential and proximity constraints, and without fleet heterogeneity. Such problems can be solved using several types of algorithms. These are investigated below.

Description of Algorithms
The possible algorithms can be divided into four categories, namely: market algorithms, machine learning methods, evolutionary methods, and probabilistic methods. A short description of each type is given below:

- **Market Algorithms**: these algorithms make use of economic principles to assign tasks. In these algorithms, tasks are treated as commodities, and agents are considered to be bidders. Agents place bids on tasks based on their utility function. Tasks are then assigned by the auctioneer to agents such that the total cost is minimized. In most cases, as each agent behaves selfishly, there is no reason for the auctioneer to prioritize some agents over others [10].
- **Machine Learning Methods**: reinforcement learning is a versatile and used technique in the scope of task allocation. Agents make decisions that aim to minimize a cost function. Over time agents learn about the types of tasks they should choose and how to split the tasks efficiently amongst the fleet [10].
- **Evolutionary Methods**: inspired by the evolution of animals in nature, these types of algorithms aim to allocate tasks based on the "survival of the fittest". However, such algorithms may result in conflicts and exhibit slow responses to changes in the environment [6].
- **Probabilistic Methods**: such methods, based on game theory, assumes that the agents are players. The task allocation concept is applied to the specific strategy chosen by each player. Based on the choices each player makes they receive a payoff after the game has finished. Such approaches aim to reach a Nash equilibrium, a state where all players do not want to change their strategy as they would have no incentive to do so [10].

Comparison of Algorithms
These methods are compared against their solution uniqueness, fault tolerance, scalability, and development costs (see Table 6.5. Solution uniqueness refers to whether the algorithm returns a unique solution over multiple runs. Fault tolerance is the resistance of the solution to errors in initial conditions, (for example, if a drone communicates it is *IDLE* when it is actually *ACTIVE*). Scalability refers to the suitability of the algorithm to accommodate an increase in drones and tasks. Lastly, development costs consider the ease of implementation. A trade-off in a traditional manner is not done as quantifying these algorithms depends on the specific architecture chosen and implementation structure.

Table 6.5: Fleet management algorithms

Algorithm	Solution Uniqueness	Fault Tolerance	Scalability	Development Costs
Market algorithms	Typically unique	Resistant	Can easily handle increasing number of drones and tasks	Medium
Machine learning	Not unique	Affected by errors	Provided sufficient data, increasing numbers of drones and tasks can be accommodated	High
Evolutionary methods	Not unique	Resistant	Suitable for scaling up	High (easy to implement but hard to tune)
Probabilistic methods	Unique	Resistant	Suitable for scaling up	High (tuning is difficult)

From this table, one algorithm cannot be deemed “the best.” Hence, for practical reasons, a scalable algorithm with lower development costs will be implemented. This choice stems from sustainability considerations. By opting for a more scalable and cost-effective algorithm, the system can be deployed with fewer resources, ensuring that it can be maintained and updated efficiently over time.

Chosen Algorithm

Based on the considerations above an auction-based approach is chosen. Drones behave prioritizing their interests by bidding higher for tasks that maximize their utility. The utility function of drones aims to maximize the workload a drone can acquire under a minimal commute. If an auction is called, all drones are required to place bids, for all clusters. Bid calculations are processed on the drone. Once all bids have been placed, the auctioneer assigns the task to the drone that placed the highest bid. The time complexity and memory requirements are significantly reduced as each drone only needs to compute its bids. In general, the auction will be called under two circumstances namely: if a drone becomes task-less, or if a new shipment arrives.

Before considering the utility function in more detail it is important to note that within this market-based algorithm, the sequential and proximity constraints should be handled prior to the assignment of tasks [11]. Even though the assignment of tasks is instantaneous, coupling with the sequential constraints, it is useful to cluster tasks into groups. These groups are formed based on temporal and spatial considerations; that is, a group contains cargo items that are “close together” with the same “deadline”. It is reasonable to argue that imposing temporal constraints might result in overlapping clusters or smaller clusters. However, this is not valid, as cargo items within one operational window will be placed closer to each other rather than to items with a different deadline. A constraint imposed on the clusters is that each cluster has a maximum size of 15 items. This is because scanning 15 items will take up to 9 minutes according to **CRG-STK-01**, and within this timeframe, the auction will likely be recalled. The minimum number of clusters is the number of *IDLE* drones. Clusters are formed using a density-based method.

These considerations show that drones do not place bids for individual tasks but for a cluster. The bid placed is dependent on the utility function. This utility function of the drones is dependent on the drone’s current task status, the commute distance, the expected workload of a group, and its battery status. The relationship between the criteria and the placed bids is as follows:

- The task state of the drone is binary, this is because task assignment is instantaneous. If the drone is occupied at the time the auction is called, it has a full workload and is not able to accept more workload. The auction does not plan tasks for the future and hence a busy drone cannot accept more tasks. As a result, its current workload is zero. If the drone is not occupied, its current workload is one.
- The required commute of a drone for a cluster is defined as the shortest Euclidean distance from the drone’s current position (at the time of bid placement) and a point in that cluster that has more than 80 % probability of containing a cargo item.
- The expected workload of a cluster is defined as the product of the number of cargo items to scan and the area spanned by the cluster. This is not a binary indicator.
- The battery status provides an indicator for the time the drone has left before needing to recharge. Drones that have less battery will place higher bids for clusters that are closer to a docking station.

The bids are directly proportional to the workload of the cluster, the current workload of the drone, and the battery status. They are inversely proportional to the distance of the drone to the cluster, and the distance of the cluster center to a docking station. Once all bids have been gathered by the auctioneer, task assignment is done. Here, a priority-based framework is used by the auction. Clusters are assigned a priority level based on the time remaining till their deadline (the shorter the remaining time the higher the priority). Higher-priority clusters are assigned drones first. Note that scouting is also a task, and is assigned the highest level of priority. This entails that if a Cargoscout is required and an auction has been called, a Cargoscout will be assigned. Furthermore, this indicates that mode switching (**CRG-OPS-13**) happens via the auction.

Each drone gets assigned to a task (no two drones share a task). The priority level of a cluster could have been considered during bid placement but in such a case, a drone may place a higher bid on a cluster further from its deadline to maximize the drone's utility. By holistically considering cluster priority, there is no need for an auctioneer to consider the selfish interests of the fleet. An overview of the entire algorithm is given below.

Algorithm 1 Auction Algorithm

```
Require: Drone IDs $D = \{d_1, d_2, .., d_n\}$, and drone states $D_S = \{d_{S_1}, d_{S_2}, .., d_{S_n}\}$
Require: deadline for boxes $B = \{b_1, b_2, ..., b_m\}$
Ensure: 1,000 boxes are scanned per hour by the fleet
  Form clusters based on proximity $C = \{c_1, c_2..., c_p\}$ where each cluster contains up to 15 cargo items
  Assign priority factors $P = \{p_1, p_2..., p_p\}$ for each cluster in $C$ based on the deadline
  while auction is called do
      for each drone $d_i \in D$ do
          for each cluster $c_j \in C$ do
              Calculate bid value $V_{ij}$ based on:
                Current task status of drone $d_i$:
                  If $d_{S_i}$ is IDLE: current workload is 1 : Else: current workload is 0
                Euclidean distance from $d_i$ to closest point in $c_j$
                Expected workload of $c_j$
                Battery status of $d_i$
          end for
      end for
      Assign each cluster $c_j$ to the drone $d_i$ with the highest bid $V_{ij}$ prioritizing clusters with a higher $p_j$
      Update drone states and remaining unassigned boxes
  end while
```

Note that as an edge case, if multiple trucks unload occur during the same operational window, the auction algorithm will be called at each unloading. A cargoscout will be sent to each local area. In such a case, clusters closer to their deadline are prioritized, which could result in the newer shipments being "neglected." However, this makes sense because there is no incentive to prioritize new shipments if old shipments cannot fully be scanned. Once the operational window for the original shipments has passed, drones will be allocated to the new shipments.

6.5. Fleet Trajectory Planning

Complementary to the fleet management system, a planning strategy is required to ensure the completion of the devised tasks. The problem of fleet trajectory planning can be split into the description of the problem, the global planning strategy, and the local planning strategy.

6.5.1. Problem Description

Path planning within a warehouse environment presents significant challenges due to the dynamic and complex nature of the space. Warehouses are filled with numerous dynamic and static obstacles such as shelving units, pallets, and moving equipment, creating a relatively dangerous environment that requires precise navigation. Employing a multi-agent drone fleet in such an environment further increases the complexity, making it vital to ensure coordination and adaptability.

Considering the problem domain which entails a set of agents completing tasks simultaneously, a desirable planning strategy allows for a distributed way of accessing and updating each of these agents. With that in mind, a distributed path planning strategy was chosen to address the problem and to enable the update of the tasks and location of each agent without disrupting the entire fleet system. Each agent becomes, therefore, its entity, being able to make decisions locally and coordinate with other agents toward collision avoidance and task efficiency maximization.

In this framework, the planners are responsible for devising unique trajectories for each agent, ensuring they can reach their goals in a timely manner without overlapping. This distributed path planning strategy facilitates real-time updates and adjustments, enabling the system to handle dynamic changes effectively. Several ready-made algorithms can meet some of these requirements, thus, before diving into a selected solution or algorithm, it is important to acknowledge the exact problem space with its characteristics and specific objectives.

Path Planning Considerations

In the context of multi-agent path planning in the environment of the warehouse, a few considerations are critical in determining the most suitable path-finding algorithms for the problem. These are the problem space, the number of known and unknown objects, and the objective function. Firstly, the problem space is a structured, yet large 3D space. The most typical algorithms for such environments are grid-based due to the arrangement of the elements in the system and they involve a shortest-path search (Dijkstra), coupled with a heuristic function that tunes according to the nature of the problem. However, the drawback of such algorithms is the increased computation time that comes with increasing mesh resolution. Next, the choice of the path-finding algorithm depends on the amount of dynamic elements in the system. Considering all the human operators, drones, and forklifts, a real-time and adaptable algorithm is desired to ensure that plans remain effective in the evolving environment. Last but not least, the objective function, which involves time, power usage, and delays, drives the choice of the algorithm in terms of time and space complexity.

Following these considerations, two separate problem spaces are identified: the full warehouse environment and the localized area around a cluster. While the warehouse space demands a complete and optimal solution to ensure that the drones reach their tasks, the localized area would benefit more from a fast and extendable range algorithm due to the uncertainty of the size and location of the boxes in a cluster. Consequently, the planning problem is divided into a global and a local planner, each dedicated to addressing the specific challenges within their respective domains. The two planners complement each other in terms of performance and they seamlessly integrate into a flexible planning strategy presented in Figure 6.3. With this sequence in mind, the next sections will elaborate on the exact configuration of each of the planners.

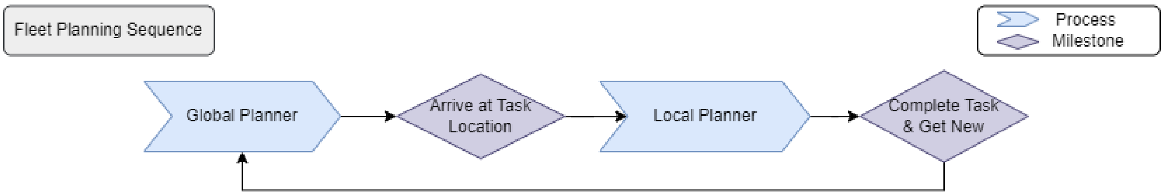


Figure 6.3: Fleet planning strategy

6.5.2. Global Trajectory Planning

On the global level, the goal of the trajectory planner is to facilitate the non-convergent movement of all the agents within the warehouse. Secondary objectives in this context are time efficiency and power preservation. With this in mind, a fast and complete solution is desired as the problem space is sparse, and timely arrival at the target location is crucial for increased throughput. Since the exact location of the shipments is not known at this point, arrival at the target location (closest point in the cluster to the drone) ensures that the agent starts exploring the cluster as soon as possible. Additionally, this target location does not need to be extremely precise, but rather reliable such that the agent arrives in the proximity of cargo items to be scanned.

Comparison of Algorithms

The global planner is responsible for providing each drone with a complete trajectory to the task location while ensuring that no collision occurs. Several distributed multi-agent path planning concepts are available for this purpose, each with advantages and disadvantages for operation in a warehouse environment. Following thorough market research, the most suitable ones are chosen and summarized in Table 6.6:

Table 6.6: Trajectory planning algorithms

Algorithm	Time Complexity	Optimality	Completeness	Development Costs	Scalability	Challenges
Traditional Search Algorithms: A*/AO*/ D*/θ*	Medium	Optimal	Complete	Low/Medium (Easy to implement)	Good	Non-simultaneous planning
Evolutionary Algorithms (Genetic Algorithms)	High	Optimal	Complete	High (Easy to implement but hard to tune)	Superior	Difficult to configure a suitable fitness function
Swarming Algorithms (Ant Colony Optimization, Particle Swarm Optimization)	Medium - High	Sub-optimal	Complete	High	Superior	Might get stuck at local optimums; Hard to integrate with local planner
Multi-Agent Reinforcement Learning (MARL)	High	Sub-optimal/Optimal	Can be complete	High (Requires complex reward function)	Superior	Requires a lot of training data

Although no algorithm emerges as the sole winner from the options considered, traditional search algorithms are more reliable for fast and sustainable product development. Swarming and reinforcement learning approaches show significant promise for high-performance scalable solutions. However, they require substantial resources for training and development and may not guarantee stable and safe outcomes. Therefore, to ensure completeness and scalability, the chosen solution is a distributed planner that uses an A* algorithm to plan each agent independently.

Chosen Algortihm

The A* algorithm is an informed graph-based search that makes use of a heuristic function (such as Euclidean distance - in our implementation) to find the shortest path to a goal. In essence, A* extends on the principles of the Dijkstra algorithm by reducing the solution space and prioritizing cost-efficient nodes, from both a distance and a heuristic factors point of view. Using such a robust and accessible algorithm, along with its variations (AO* for instance), gives the benefit and insurance of convergence and allows for more complex design choices in other parts of the operational cycle.

One way to implement the algorithm in the warehouse environment is to employ a 3D grid-based A* search. This would perhaps be the fastest solution, however, it comes at the expense of increased computational time due to the expansion of neighboring nodes from a set of 8 neighbors to a set of 26 neighbors. Analyzing the scale of the selected study case (KLM Warehouse), the team concluded that for single cluster - single drone tasks, the problem space can be reduced to a 2D A* or AO* search algorithm that accounts for collision avoidance between agents and manages traffic by implementing multiple flight levels. The flight levels are managed within the fleet management system, thus ensuring real-time tracking. This simplification comes in line with sustainable development principles as it reduces the computational time and the extra power associated with vertical maneuvers.

In the operational framework, the distributed strategy is implemented so that each drone/agent has a trajectory planner that takes on the task from the server (via communication) and computes the optimal path to the target with the search algorithm. Once this preliminary path is generated, the flight level is selected and the coordination & collision avoidance phase commences. During the plan execution, agents start sharing information and concurrently moving toward their target. At each information-sharing instance, they exchange properties such as ID, current location, next few steps, battery level, distance to goal, delays, etc. that are used to determine "warnings" for collisions that might happen in the future. In the event of a detected collision, a priority negotiation takes place between the two agents and the agent that loses the negotiation has to replan its trajectory based on the new constraints from the winner. An overview of the global planner is presented in algorithm 2.

Algorithm 2 Global Planning Algorithm

```
Require: Warehouse size $(length, width)$ with static knowns location $S_k = \{(x_{i1}, y_{i1}), (x_{i2}, y_{i2}), ...\}$
Require: Cluster $c_j$ location for $\forall d_i \in D$
 while at least one agent has a task to fulfill do
    Initialize $Drone$ Agents $D = \{d_1, d_2, .., d_n\}$ and $Dynamic\ Object$ Agents $DO = \{do_1, do_2, ..., do_m\}$
    $Drone$(dict) $\leftarrow$start and cluster location from fleet management
    $Drone$['constraints'] $\leftarrow$list of initial drone positions
    $Drone$['path'] $\leftarrow$computed path by A* algorithm
    while no update from the auction algorithm do
        for $d_i \in D$ and $do_j \in DO$ do
            if $do_j$ detected by navigation: distance $(d_i, do_j) \leq 3m$ then
                direction dynamic object $\leftarrow do_j$['direction']
                $Drone[d_i]$['constraints'] $\leftarrow$next locations of the dynamic object based on direction vector
            end if
        end for
        for $d_i, d_j \in D$, $i \neq j$ do
            if distance $(d_i, d_j) \leq 3m$ then
                $Drone[d_i]$.receive_message($Drone[d_j]$.send_message())
                $Drone[d_j]$.receive_message($Drone[d_i]$.send_message())
                Identify potential collisions in the path
                if collision detected then
                    $Drone[d_i]$.compute_priority($Drone[d_j]$) and $Drone[d_j]$.compute_priority($Drone[d_i]$)
                    The agent without priority replans
                end if
            end if
        end for
    end while
```

From a formal point of view, the drone agents are equipped with static and dynamic properties, but also with cognitive and behavioral properties. The cognitive properties allow the agent to reach decisions such as "who has priority", while the behavioral ones allow the agent to swiftly react to dynamic obstacles and adjust its trajectory in a rule-based manner. To simulate the global behavior of the designed agents, the following framework is used to test the planning and coordination between agents and between agents and dynamic objects. The algorithm assumes that the detection of dynamic obstacles is managed and relayed from the navigation algorithm to the planner. An overview of the agent properties and proximity-based interactions is presented in Figure 6.4.

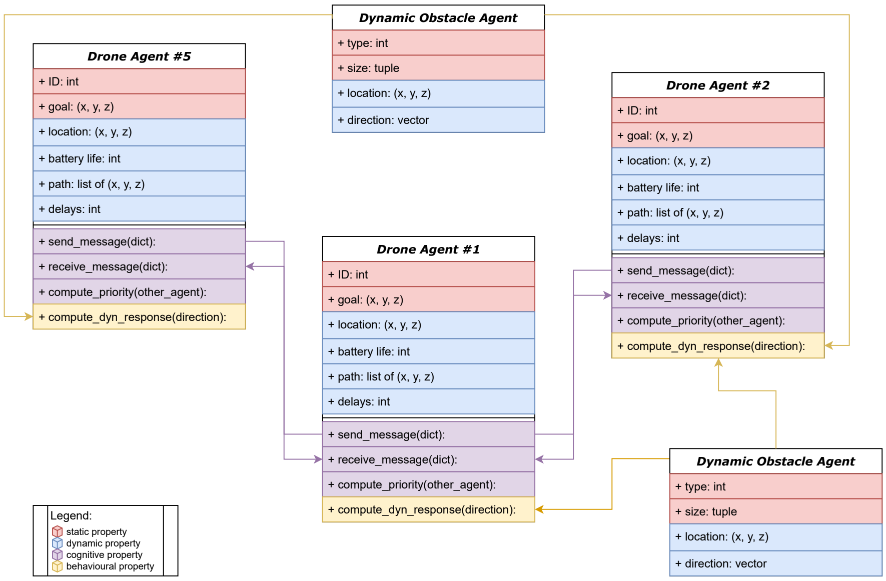


Figure 6.4: Multi-agent coordination and integration

Drone-to-drone Negotiation
Each agent is equipped with a method that allows the computation of the priority factor - a global metric that dictates the traffic at the global flight level. This negotiation can be either conducted in a rule-based manner (i.e. agent coming from the right has priority) or by employing a dynamic and adjustable metric that takes into account various aspects of the trajectory of the drone. The second option was chosen and implemented in the global planner due to the flexibility it offers in adapting to other warehouse environments that have “different” rules. It employs an indicator that accounts 50% for the estimated path to the goal (with the agent further away from the goal having priority), 30% for the current delay (how many priorities it has offered so far), and 20% for battery level (with a lower battery level having priority over a higher one). Based on this property, drones conduct negotiations with each other once they enter each other’s dangerous control zone (3m), with a drone control box of 6m ensured by navigation sensors.

In summary, the global planner ensures that movement to and between tasks is performed in a fast fashion and that the elements of the system remain interconnected and communicate with each other when situated in adjacent localized areas. Furthermore, collision avoidance mechanisms (communication and interface with navigation algorithms) are integrated into the global planner to avoid all types of dynamic and static objects during path execution.

6.5.3. Local Trajectory Planning

Once global trajectory planning is secured, the only challenge that remains is local navigation, namely the exploration of the localized areas within the assigned clusters. Seemingly trivial, the problem becomes more complicated as the uncertainty of the previous locations and estimations are considered. The problem space shifts from a large and sparse 3D space to a fine and sensitive mesh as the payload operations become part of the operational cycle.

Challenges and Requirements
For the payload operations and navigation, a high resolution is desired to meet the goal of accurately identifying and measuring cargo items on a local level. Next to this, another challenging element for navigation and planning is the height of the boxes. Since the dimensions of the shipments are not necessarily consistent along a single-truck shipment, it is very likely that neighboring packages on the skids have different heights. Additionally, to conduct payload measurements, several poses and angles are required to gather the complete information for volume measurements. These poses include 4 to 5 positions for taking volume measurements - typically describing a quarter circle arc, and a pose for measuring and reading the label information. The transition between these points is facilitated by the drone’s controller, yet the provision of the waypoints for the controller is required from the local planner.

The main objectives of this local planner are to be real-time and able to re-route in the event obstacles are encountered. An additional objective of this local planner is to minimize the number of 90 degree turns. This constraint comes from the difficulty of performing sharp maneuvers from a flight dynamics perspective. Several algorithms are suitable for such requirements, coverage algorithms being superior to traditional graph-search algorithms for this purpose. This is due to the uncertainty associated with the location and height of the targets. Without exact information on the goal location and required flight level for scanning, grid searches have limited usefulness and completeness.

Comparison of Algorithms
Four types of area-coverage algorithms can be identified, namely: Boustrophedon cellular decomposition algorithm, spiral, waveform, and rapidly exploring random trees. The methodology behind these algorithms is described below, and a depiction of their rough paths is given in Figure 6.5:

- **Boustrophedon Cellular Decomposition Algorithm**: this algorithm divided the problem space into a series of cells, such that the generated path covers each cell in a back-and-forth manner [12].
- **Spiral Algorithm**: this algorithm aims to cover the entire problem space following a spiral pattern. The spiral can be done in an outward or inward manner. This algorithm is particularly useful for round/elliptic clusters.
- **Waveform Algorithm**: similar to the Boustrophedon algorithm, however, it is capable of exploiting angled trajectories. This is particularly useful for irregular domains.
- **Rapidly-exploring Random Trees (RRT*)**: this algorithm aims to cover the entire problem space in a random manner. It builds a tree by randomly sampling points and connecting them to the nearest node in the tree, optimizing the path incrementally [13].

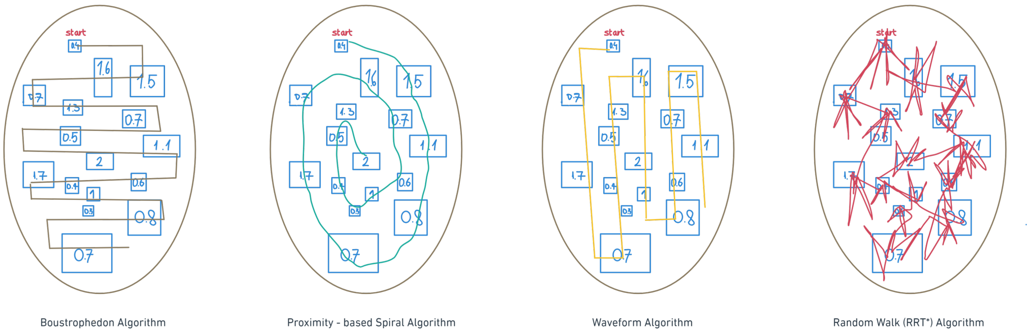
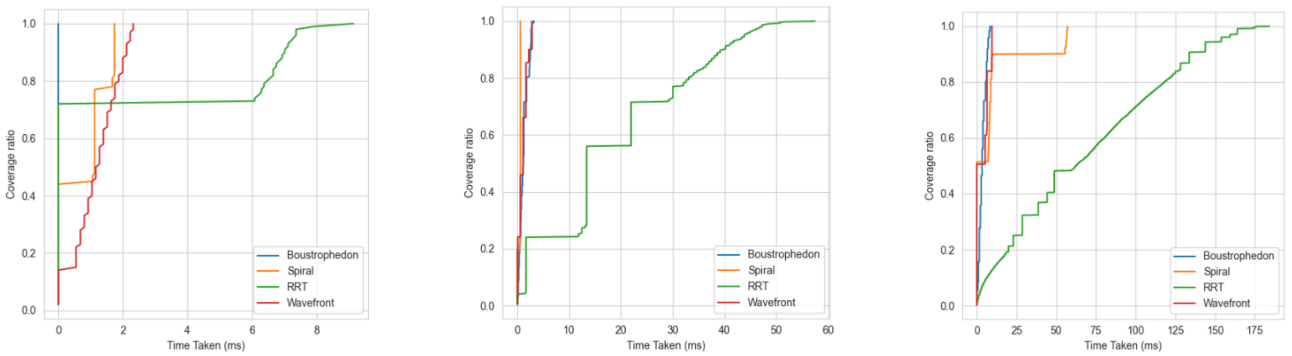

start
start
start
start
Boustrophedon Algorithm
Proximity - based Spiral Algorithm
Waveform Algorithm
Random Walk (RRT*) Algorithm

Figure 6.5: Local planning strategies

To investigate the feasibility of the four algorithms under these constraints a run-time, and control analysis was performed. The first analysis was performed by comparing the run-time of the four algorithms to establish a path for a rectangular grid with obstacles. Obstacles are not given as a priori during the initialization of the coverage algorithm, rather as the drone deploys the traditional algorithm until it encounters an obstacle. Once an obstacle is encountered the drone will reroute to avoid colliding with these obstacles. The control analysis was performed by quantifying the number of sharp turns the drone would make to transverse the given trajectory. Both of these results are plotted over the coverage ratio. The coverage ratio refers to the amount of domain the drone has traveled.

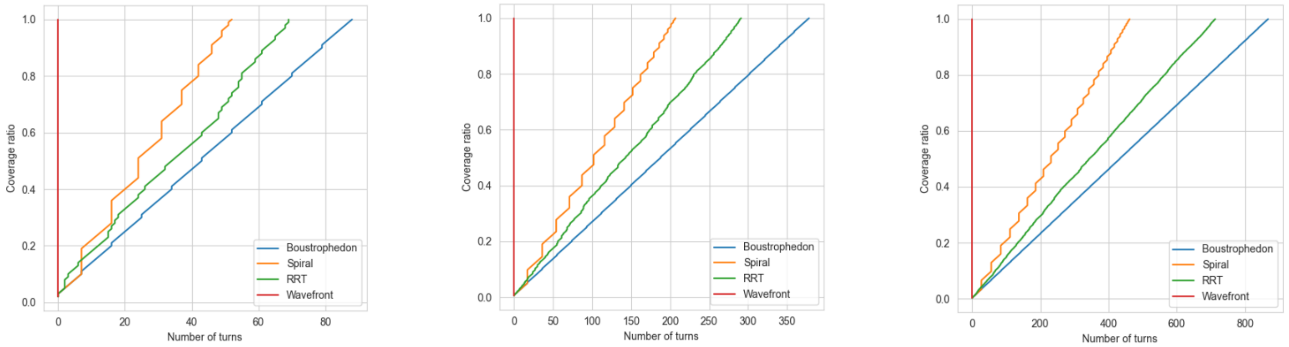
Both analyses are performed for a series of mesh refinements. Furthermore, the comparison analysis is conducted in 2D. This is because the drone will exploit the coverage algorithm when it is at a constant flight level. The results of this run-time analysis are given in Figure 6.6, and the control analysis is given in Figure 6.7.

From these figures, it appears that the Boustrophedon cellular decomposition algorithm has the lowest run-time over all three grid refinements. However, it also requires the most number of sharp turns. As the ratio of the width to height (or vice versa) of the domain increases, it becomes beneficial to use this algorithm. The wavefront algorithm has few sharp turns as the solution deploys angled turns. As a result, for quasi-square areas, the wavefront or spiral algorithm offers a more dynamic solution.



(a) Grid refinement of 10 **(b)** Grid refinement of 20 **(c)** Grid refinement of 30

Figure 6.6: Run-time for different local planning algorithms based on grid refinement



(a) Grid refinement of 10 **(b)** Grid refinement of 20 **(c)** Grid refinement of 30

Figure 6.7: Number of sharp turns executed by different local planning algorithms based on grid refinement

Other performance metrics such as the power usage of the drone to follow these trajectories could be used. This would require more in-depth knowledge on the battery discharge characteristics and remaining useful life. Hence this is left as a recommendation for future works.

Chosen Algorithm

Although the problem so far has been treated in 2D, for local planning, a 3D path planning implementation is desired to account for the various heights of boxes, payload operations, and dynamic obstacles. Therefore, the chosen solution would be extendable to a 3D space, with multiple flight levels that are about 0.5 m apart from each other. This extension is attainable for all algorithms as it entails running the same algorithm on each of the 2D layers, however with different collision avoidance constraints coming from the height of the boxes.

Considering the available algorithms for local planning, no evident consensus regarding superiority emerges. As a result, the local planner's final implementation incorporates three of the four available algorithms and allows for a flexible coverage strategy depending on the segmented cluster. Assuming that some information about the estimated height of the shipments can be drawn from the Cargoscout scan, it would be possible to build a reinforcement learning-based algorithm that analyzes the topology of the cluster and decides which of the four algorithms would be the most efficient. However, at this point of design, this piece of information is rather a speculation that might not be achievable in real life. Therefore, the local planner implementation allows for the selection of the coverage algorithm, while having the default as Boustrophedon.

6.6. Operational Cycle

Having elaborated on the architecture of the subsystem and the specific algorithms used, an overview of the operational cycle is given. This overview is split between the two possible operational modes. Cargoscout functionalities are described in Section 6.6.1, and Cargonaut functionalities are given in Section 6.6.2.

Mode Switching between these two operational cycles can be either done autonomously at the end of the task or at the order of the fleet management. In-flight mode switching allows for flexible operations, allowing a Cargoscout to switch to Cargonaut and start scanning immediately. An important note is that the Cargoscout only runs the local coverage algorithm for path planning.

6.6.1. Cargoscout

The operational cycle of the Cargoscout is given in Figure 6.8.

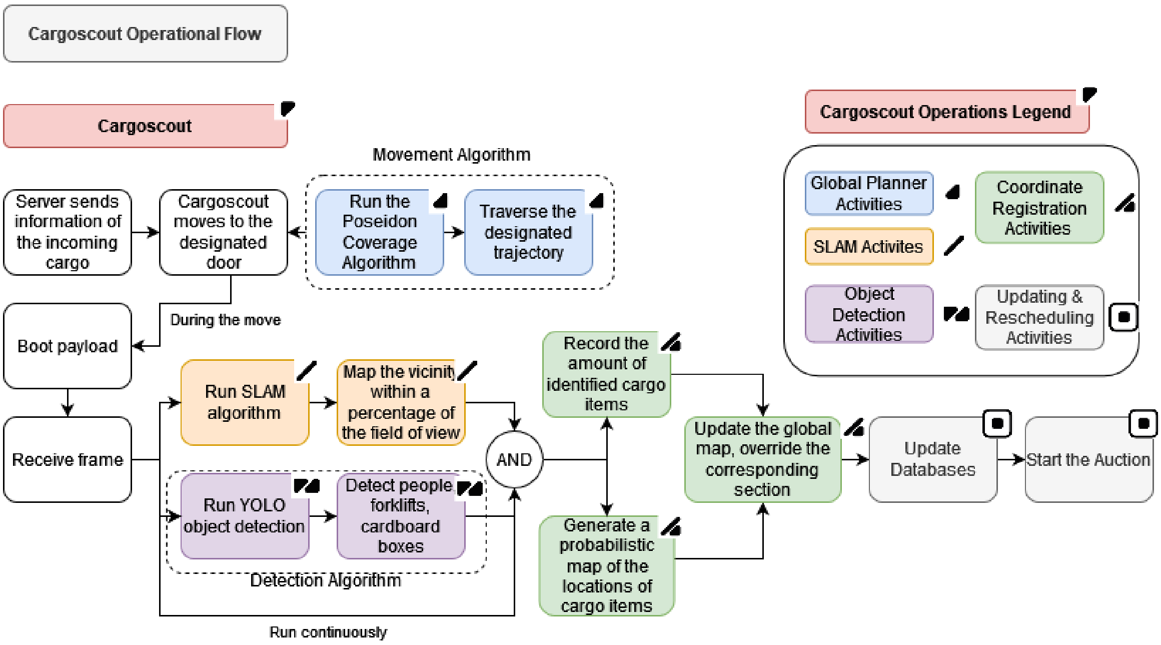

Cargoscout Operational Flow
Cargoscout
Server sends information of the incoming cargo
Cargoscout moves to the designated door
Movement Algorithm
Run the Poseidon Coverage Algorithm
Traverse the designated trajectory
During the move
Boot payload
Receive frame
Run SLAM algorithm
Map the vicinity within a percentage of the field of view
Run YOLO object detection
Detect people, forklifts, cardboard boxes
Detection Algorithm
Run continuously
AND
Record the amount of identified cargo items
Generate a probabilistic map of the locations of cargo items
Update the global map, override the corresponding section
Update Databases
Start the Auction
Cargoscout Operations Legend
Global Planner Activities
Coordinate Registration Activities
SLAM Activites
Object Detection Activities
Updating & Rescheduling Activities

Figure 6.8: Cargoscout operational cycle

6.6.2. Cargonaut

Similarly, the operational cycle of the Cargonaut is given in Figure 6.9.

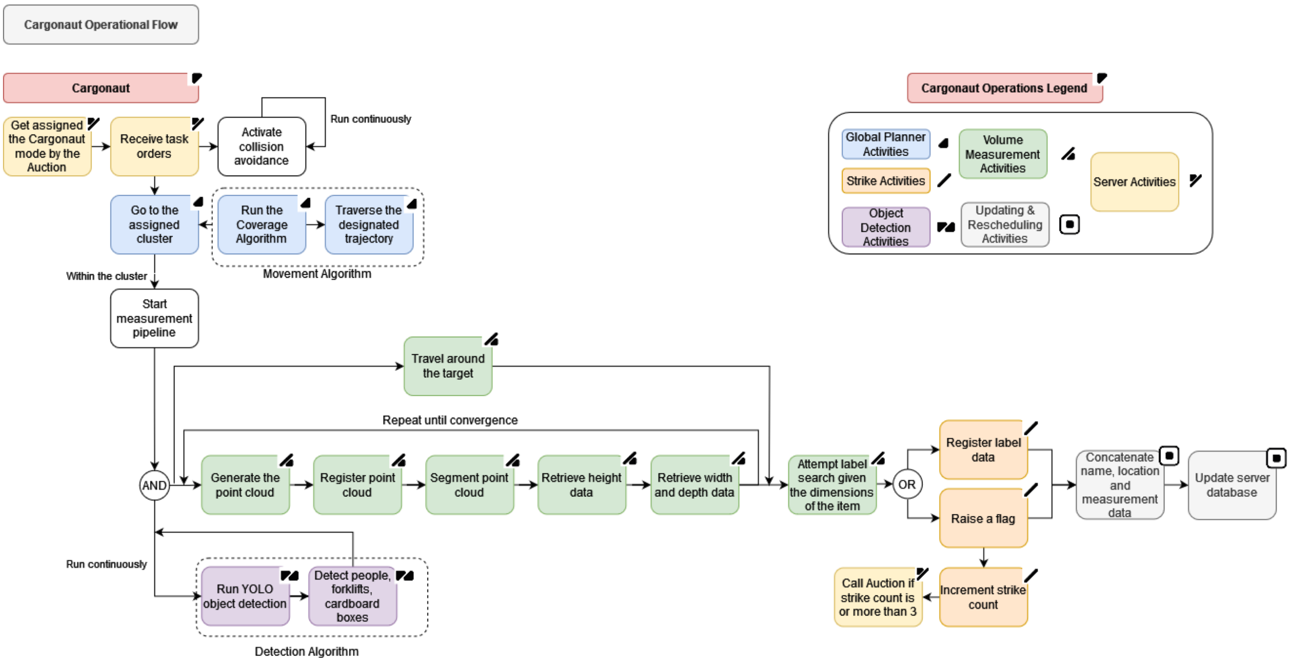

Cargonaut Operational Flow
Cargonaut
Get assigned the Cargonaut mode by the Auction
Receive task orders
Activate collision avoidance
Run continuously
Go to the assigned cluster
Run the Coverage Algorithm
Traverse the designated trajectory
Movement Algorithm
Within the cluster
Start measurement pipeline
Travel around the target
Repeat until convergence
AND
Generate the point cloud
Register point cloud
Segment point cloud
Retrieve height data
Retrieve width and depth data
Attempt label search given the dimensions of the item
OR
Register label data
Raise a flag
Increment strike count
Call Auction if strike count is or more than 3
Concatenate name, location and measurement data
Update server database
Run continuously
Run YOLO object detection
Detect people, forklifts, cardboard boxes
Detection Algorithm
Cargonaut Operations Legend
Global Planner Activities
Volume Measurement Activities
Strike Activities
Server Activities
Object Detection Activities
Updating & Rescheduling Activities

Figure 6.9: Cargonaut operational cycle

Payload Operations

The payload operations involve multiple processes as described in the Cargonaut operational cycle. To ensure compliance with the stakeholder requirement of conducting volume measurement for a single cargo item in 30 seconds, the following payload schedule breakdown has been established.

- **Target identification: 25 milliseconds** assuming this is completed with the object identification module. When identified, the approach phase initiates and the local planner is put on hold until the volume measurement is completed.

- **Navigation to target: 2 seconds** assuming that cargo items can be detected from 3 meters and that travel to an advantageous pose for measuring is conducted with 1.5-2 m/s.
- **Travel between poses: 0.5 seconds per pose** assuming an arc of 90 degrees is followed around the target to perform volume measurements from various angles. The largest distance to be spanned is around 4 meters assuming a box length of 4 meters.
- **Stabilization of drone at a pose: 0.3-1.5 seconds per pose** assuming either yaw or translation movement is conducted to move between poses.
- **Point cloud acquisition and volume measurements: 1-1.5 seconds per frame** around 10 frames are required for an accurate volume measurement.

Warehouse Integration

With the full cycle in mind, the last thing to be considered is the implementation of the operational cycle in a real house. The Cargonaut solution aims to minimize the time associated with moving the cargo around the warehouse for measurement purposes. In the ideal scenario, upon acceptance in the warehouse by authorized personnel, the shipments will be placed around the unloading area and cargoscout operations will be first carried out. Afterwards, a fleet of cargonauts is sent to process the items and upon competition, the operators are given the green light to start moving the cargo items (from the localized area) around the warehouse. In the grand picture, as long as human operators do not attempt to move the packages in this time window, the interactions and potential collisions are easily mitigated. Furthermore, by saving time from the unloading processing, the human power can be used in a different warehouse area.

6.7. Deployment

At deployment of the system in a warehouse, a multitude of practical considerations are to be made. This includes but is not limited to, the positioning of the docking stations, that is the Cargomothers, the possibility for manual override via a kill switch, and other safety and sustainability considerations.

6.7.1. Cargomother

Cargomothers are the docking and charging stations of the drone fleet. They are vital for the functionality of the system as they support and shelter the drones during recharging or hot-swapping. The Cargomothers are wardrobe-sized stations that provide shelves for the entire or part of the fleet. They bring together two critical elements in the recharging of the batteries: the shelves and the protective charging station in which the batteries are stored while charging in order to eliminate the risks of short-circuit or other hazards. The Cargomother design is presented in Figure 6.10

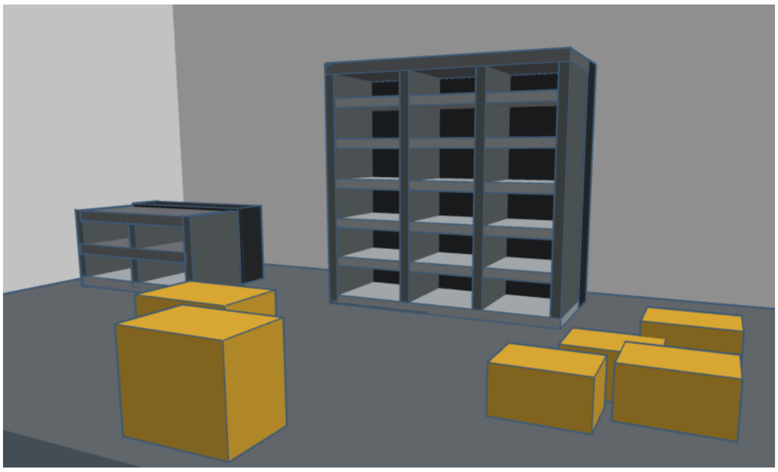


Figure 6.10: Various size Cargomothers in a warehouse

The positioning of the Cargomothers is crucial for the efficient operation of the system. Docking stations should be strategically placed to minimize travel time for drones or robots and ensure they are accessible without interfering with warehouse operations. Factors to consider include the layout of the warehouse, proximity to high-traffic areas, and availability of power sources. Proper placement will ensure quick recharge or refueling, facilitating continuous operation and minimizing downtime. With these constraints in mind, a preliminary location for the Cargomother has been chosen to be in the proximity of a wall.

To accommodate flexible fleet sizes, the Cargomothers will be made in a modular manner. This offers the possibility for customization and does not compel customers to buy Cargomothers in a configured number. In turn, this minimizes the need for over-production and results in an efficient use of resources.

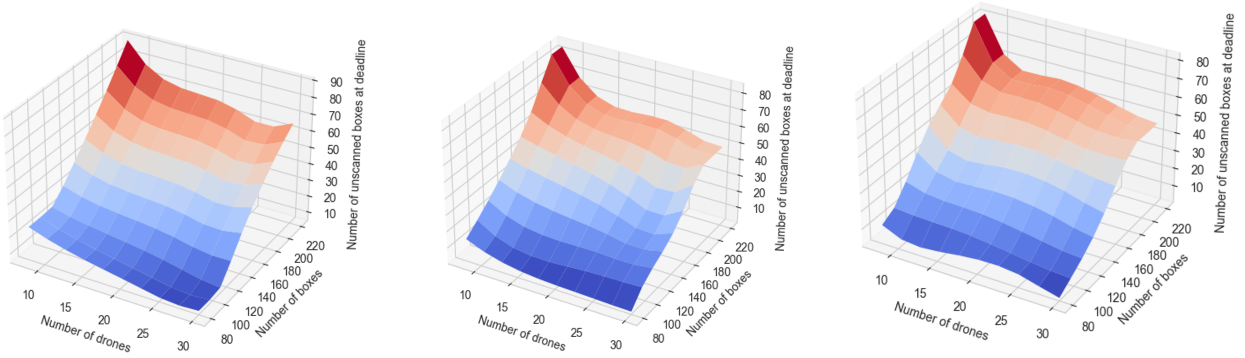
6.7.2. Safety and Manual Override

In the event that the system does not function as intended, it is paramount that the system remains safe for the environment and the workers. To provide this functionality a kill switch is embedded. This switch will be embedded into the server. In addition, every 10 meters on the wall a switch will be placed. This reduces the likelihood of **RSK-OPS-06** that operators will be injured by using the switch. If the kill switch is used the drone will descend to the ground. If the navigation sensors still function as intended there will be an attempt to avoid landing on cargo items, or other objects.
Next to the kill switch, the system also has a functionality that allows drones to be manually assigned to tasks or removed from operations. This functionality accompanies the digital environment on the server that keeps a real-time map of drone locations. Such assignment overrides any task allocations performed by the auction.

6.8. Fleet Size Sensitivity

Determining the most feasible fleet size depends on the distribution of the cargo items amongst the doors, task duration, and operational constraints. It is quintessential that **CRG-STK-02** is satisfied. However, otherwise, the size of the fleet does not have any additional constraints. Hence, three three variables will be considered in detail. Regarding the distribution of the cargo items at the door, three cases can be identified, either a convex distribution, concave or uniform distribution. As it is not known which type will prove to be critical, all three cases will be investigated. From **CRG-STK-01** it can be assumed that scanning a cargo item will be performed in at most 30 seconds. Furthermore, a nominal speed of 2 m/s will be assumed between clusters and $0.5\,\mathrm{m/s}$ within clusters.

To determine the fleet size, a peak hour operational window is modeled (a span of 15 minutes). During this window, it is assumed that two drones will need to be charged and one truck unloading will occur (hence one drone will need to be sent as a Cargoscout). The reason two drones will need to be charged can be motivated as follows. In general warehouses have peak and off-peak hours. During peak hours hotswapping will be required to mimic continuous operations. Furthermore, the warehouse has the same structure as that of KLM's The results of this simulation for the three distributions are given in Figure 6.11. Note that the total number of incoming boxes is the same regardless of the distribution. The number of unscanned boxes is given an integer because the total number of incoming boxes is the same regardless of the distribution. Hence, it does not need to be given as a ratio.



(a) Convex incoming box distribution **(b)** Concave incoming box distribution **(c)** Uniform incoming box distribution

Figure 6.11: Fleet size analysis for various cargo unloading distributions

From the figure, it is evident that the convex load distribution is critical. The increased concentration of tasks at the ends of the operational window creates a bottleneck, leading to a higher number of unscanned boxes. Furthermore, the analysis indicates that a fleet size of 10-12 drones is appropriate for a warehouse of a size similar to that of KLM's. This range is provided to account for varying operational demands and the acceptable number of boxes that can remain unscanned by the end of the operational window. Depending on the specific requirements and the criticality of having all boxes scanned, a slightly larger or smaller fleet size may be chosen to balance efficiency and coverage.

6.9. Verification and Validation

At this stage, preliminary implementation of algorithms has been performed. Prior to using the results of these algorithms, verification and validation must be performed. Once the algorithms have been verified an validated, it is of primary concern to ensure that the final design is compliant with all requirements and fulfills the mission objectives within the established constraints. Section 6.9.1 and Section 6.9.2 consider the procedures for software verification for fleet management and fleet trajectory planning respectively. Validation procedures are given in Section 6.9.3. Compliance with requirements is verified in Section 6.10.1.

6.9.1. Verification of Fleet Management

Verification of the fleet management system was done following a hierarchical procedure. Firstly unit tests were performed, followed by system tests. Once the functionality of the code was checked, extreme value tests and a robustness analysis was performed. These are elaborated below.

Unit Tests

Unit tests were performed on each indivisible function of the fleet management algorithm. These tests were performed to check the functionality of the code. Notable examples of this analysis include:

- The bid valuation function was compared to numeric values computed manually.
- The maximum size constraint of the clustering algorithm was verified via visual inspection.

System Tests

The system tests were performed to test the combined functionality of the management system. These tests verified the clusters made, and the tasks assigned. Examples of this analysis include:

- Verify that the clusters made include both temporal and spatial aspects. This was done through visual inspection by assigning each box a deadline.
- Verify that occupied drones are not given tasks. This was checked by manually inputting a test case where some drones were initialized to be active.
- Ensure that the global goal is prioritized over spatial considerations. To test this boxes with the different deadlines (one close and one farther away) were placed quasi-equidistant from the fleet.

Extreme Value Tests

Furthermore, extreme value tests were performed to check the functionality of the auction algorithm under extreme inputs. Two cases were identified namely all drones are *ACTIVE*, or all drones are *IDLE*. If all drones are *ACTIVE* there is no need to call an auction as all drones are currently occupied with their tasks. Considering only instantaneous assignment is performed by the auction, if there are no *IDLE* drones, it is expected that the auction does not reassign tasks to any of the drones. On the contrary, if all drones are *IDLE*, the auction should assign tasks for all of them such that clusters with items closer to the deadline are prioritized. This was tested and the results were in line with expectations.

Robustness Analysis

The robustness analysis evaluates the algorithm's ability to handle variations in fleet size and task load, ensuring its performance remains stable under different operational conditions. This was done by performing a runtime analysis to understand how the algorithm scales with increasing fleet size and the number of tasks. The results of such an analysis can provide insight to the potential bottlenecks in the algorithm. This run-time analysis was performed by computing the time required to perform a complete task allocation (including clustering and computing bids) for various fleet sizes under a range of incoming cargo items (with a convex distribution). A convex distribution was used as it is the critical case. The graph obtained is visualized in Figure 6.12.

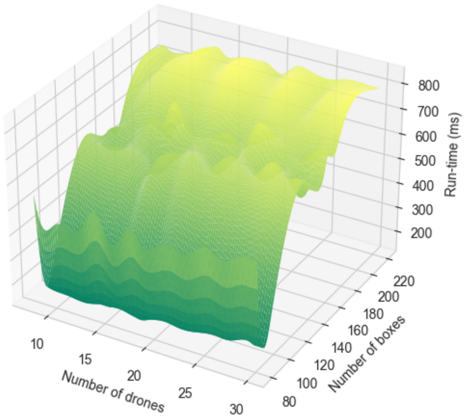


Figure 6.12: Run-time for the auction algorithm as a function for the number of drones and the number of boxes

From this graph, it is apparent that the algorithm is robust to an increase in the number of drones and incoming cargo items. It appears that an increase in boxes has a greater effect on the run-time, compared to an increase in fleet size. This suggests that while the algorithm scales efficiently with the number of drones, managing a higher volume of tasks requires more computational resources and time. However, even with an incoming load of 280 boxes, the run-time is less than a second.

6.9.2. Verification of Trajectory Planning

Similar to the software verification performed for the fleet management system, a hierarchical procedure is used for verifying the trajectory planning algorithm.

Unit Tests

Unit tests were performed on each indivisible function of the fleet trajectory algorithms. These tests were performed to check the functionality of the code. Notable examples of this analysis include:

- The variation of task locations for various fleet sizes.
- The functionality of the drone agent in a different planner.
- The priority valuation function was compared to numeric values computed manually.

System Tests

System tests were performed on the full functionality of the global and local planning algorithm. These tests were performed to check the collision avoidance and obstacle avoidance functionalities of the global planner and to ensure the coverage of the local planner. Notable examples of this analysis include:

- The variation of task locations for various fleet sizes.
- Random generation of agents and tasks in the environment.
- Generation of static obstacles that only occur for a few time stamps such as a stationary forklift.
- The maximum size constraint of the clustering algorithm was verified via visual inspection.
- The update of tasks during planning.
- The performance of the planning strategy against other planning algorithms.

Other system tests were conducted to ensure the integrability of the trajectory planning algorithms and the overall scalability of the model.

Robustness Analysis

Robustness analysis was conducted on the global planning algorithm to assess the flexibility of the architecture to various fleet sizes and task loads. This was done by performing a runtime analysis to understand how the algorithm scales with increasing fleet size. The runtime analysis has been conducted on the algorithm implemented (A*), as well as on the implementation of the AO* algorithm (that is seen as the more flexible variation of A* for highly dynamic environments) and is shown in Figure 6.13. The results of the analysis prove the scalability of the algorithm, with reasonable runtime for a fleet size of up to 20 drones. However, for fast responses (under 30 seconds distributed along the fleet), a fleet size of under 15 drones is best suited for the planning algorithm. Furthermore, the slight superiority of the AO* for the warehouse environment is observed. In the future, some extended implementations of the AO* algorithm will be investigated to explore all the benefits and drawbacks.

Cargomother Location Analysis

Modularity analysis was conducted on the positioning of the Cargomothers. A few scenarios with various task assignments were simulated under three configurations and the results of the analysis are presented in Figure 6.14:

- Centralized Cargomother (placed in the middle of the warehouse wall - at 50 meters)
- Two Cargomothers (placed at 15 and 85 meters respectively on the warehouse wall)
- Three Cargomothers (placed at 15, 50, and 85 meters along the warehouse wall)

The analysis reveals the superiority of modular multiple location Cargomother. It also shows that two Cargomothers would be more suited for the expected fleet size: 10-12 drones for the KLM warehouse operations.

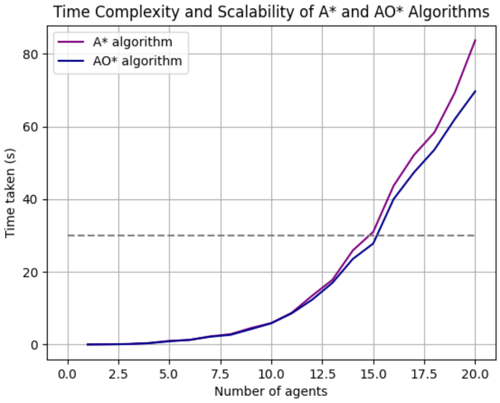


Figure 6.13: Time complexity global planner

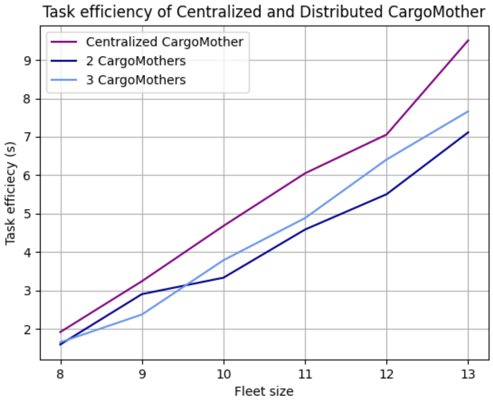


Figure 6.14: Task efficiency for various Cargomother configurations

6.9.3. Validation

Validation procedures aim to specify whether the simulated results are in line with real-life data. To perform such analyses, hardware-in-the-loop and model-in-the-loop procedures will be used. These methods have not been performed at this current stage, however after integration (with the payload and control subsystem) the following procedures will be used.

Model-in-the-loop

- Test the complete system (fleet management and fleet trajectory planning) in a simulated real-life environment using a digital twin. Through deploying the system in an external, digital, simulation environment task, and mode assignment, the given trajectories can be verified by users through visual inspection.
- Test the adherence to safety requirements such as minimum distance to other drones, obstacles, and human operators.

Hardware-in-the-loop

- Test the implementation of the fleet management and trajectory planning algorithms after the integration of the navigation subsystem, to validate the real-life performance of the subsystem. This test should be performed along with the communications subsystem.
- Test the integration of the operations and planning software with the payload software to validate the capacity of the system to complete topology measurement tasks in the postulated time frame.
- Test the manual control and system override will kill the switch to validate the complete functionality of the model.

6.10. Fleet Motion Simulation

A unifying simulation was created to assess the performance of the system when multiple departments are integrated. Hence, this chapter details the simulation of a single operational cycle: from package interception to scanning competition. Performance metrics are analyzed throughout the process and the overall speed of the implementation is determined. A plausible operations scenario will be analyzed to assess the runtime (currently on a computer CPU) and the time associated with each step of the process.

Simulation Scenario

For a good overview of the order of operations, the following scenario is analyzed: Two trucks arrive at the warehouse simultaneously at doors 1 and 5, which are a random selection. Both of the trucks go through acceptance and the cargo items are introduced in the warehouse. At this point, in reality, some of the drones might be ACTIVE and others CHARGING. However, for simplification, a scenario with 10 available (IDLE) drones is analyzed, disregarding the status of the remaining fleet. The fleet management receives the information about the arrival, initializes the items in the log with the deadline, and commences the fleet operations as follows:

1. Cargoscouts are sent to scan the area around arrival and to count the cargo items (see Figure 6.15)

   The simulated response is as follows:

   - Two drones that are in the proximity of / closest to the respective door areas are sent towards the door.
   - When each of these drones detects a cargo item, it employs the coverage algorithm and starts scanning the surroundings.

- Depending on the amount and spread of cargo items, a suitable coverage algorithm is selected. Below, two coverage options are simulated. On the left, the spiral coverage strategy is employed, while on the right, the Boustrophedon cellular decomposition algorithm is used. The choice is made with a deep learning algorithm that favors the spiral motion when the cargo items are spread apart, or else it defaults to Boustrophedon.

The computational time for this step takes on average: 0.2 - 0.8 seconds on Central Processing Unit (CPU). The operational phase takes on average: 2-3 minutes depending on the localized area size and assuming a maximum grid of 15 x 30 meters and a 2 m resolution between two subsequent turns.

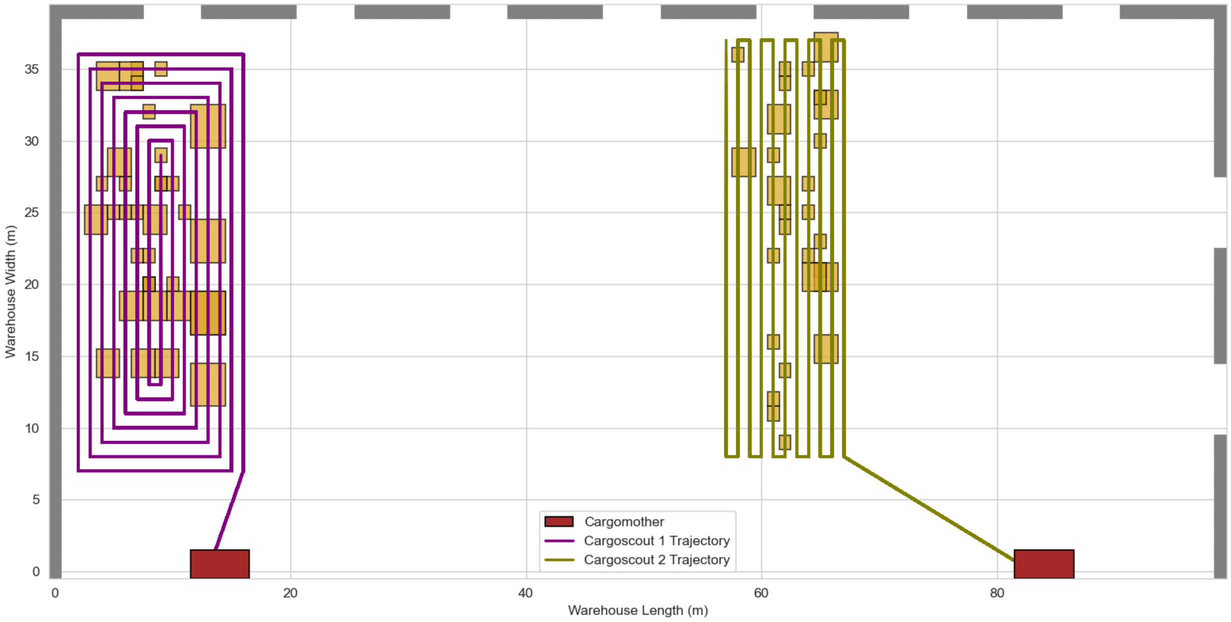


Figure 6.15: Cargoscouts scanning the incoming cargo with two different strategies

2. Server builds the probability density map based on the information from the Cargoscouts

   - Once the Cargoscouts finish scanning the localized areas, they send the information to the server.
   - On the server, the probability density map is built as discussed in Section 6.4.2 and relayed to the auction system.

   The computational time for this step takes on average: < 1 second on CPU. The operational phase takes on average: < 1 second to compute the probability density map.

3. Fleet Management performs clustering and assigns tasks (see Figure 6.16)

   - With the information received regarding the cargo items distribution, clustering is conducted and clusters are formed according to the available fleet size and deadline of the boxes.
   - Next, IDLE drones are assigned to clusters based on proximity, and the system is updated.

   It is important to note is that the clustering algorithm is not unique, thus re-running the simulation would most likely generate a different clustering pattern and therefore, different task assignments. This is because the minimum threshold for clusters is the number of taskless drones, and the location of boxes is randomly generated. However, this is fine, as drones place bids based on their utility function, and the auction is only run on demand.

   The computational time for this step takes on average: 0.2 - 0.8 seconds on CPU. The operational phase takes on average: < 1 second to compute the clusters and relay the task assignment.

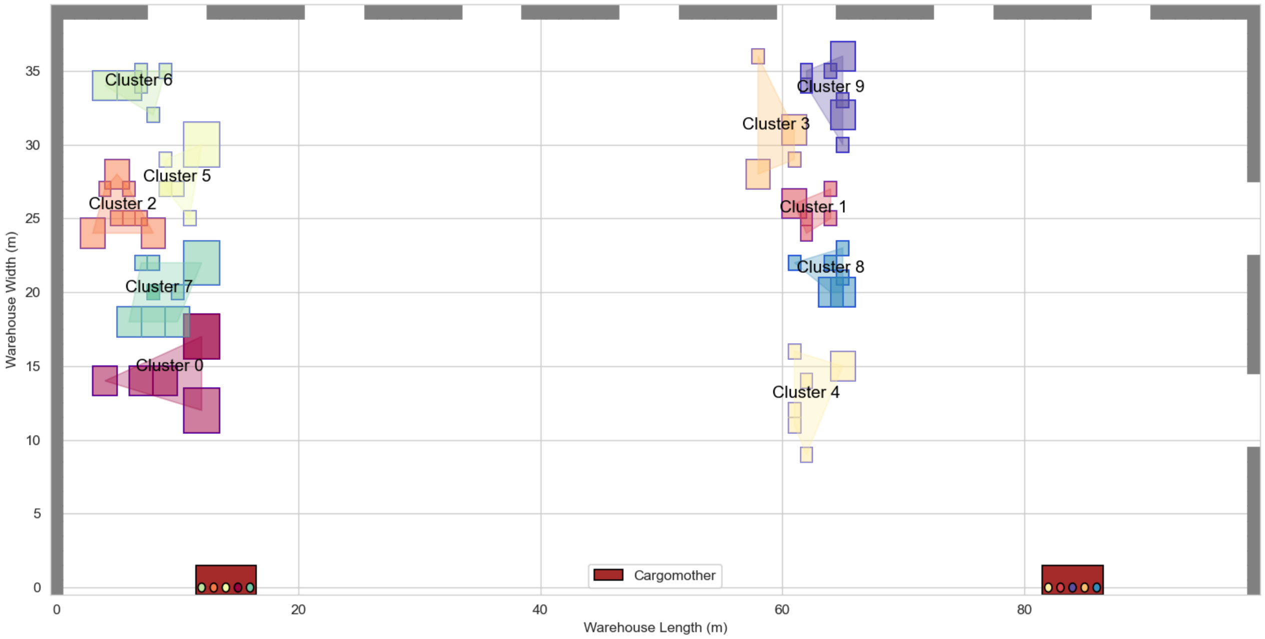


Figure 6.16: Clustering on incoming boxes

4. Global trajectory planner decides the route of each drone (see Figure 6.17)
    - Each drone plans its path to the goal with the search algorithm.
    - Dynamic objects are modeled in the environment.
    - While running on the planned path, real-time collision avoidance is performed.
    - Negotiation for priority is conducted between neighboring drones and constraints are appended to agents.

It is important to note is that the current planning strategy leads to believe that the drones are always traveling together. However, this is not valid for all scenarios. Due to the fact that all the preliminary drone trajectories steer vertically and left for the first step, they have to constantly perform collision avoidance with their neighbours. As the objective of the negotiation function involves minimizing delays and stops, constraints are constantly passed between neighboring drones.

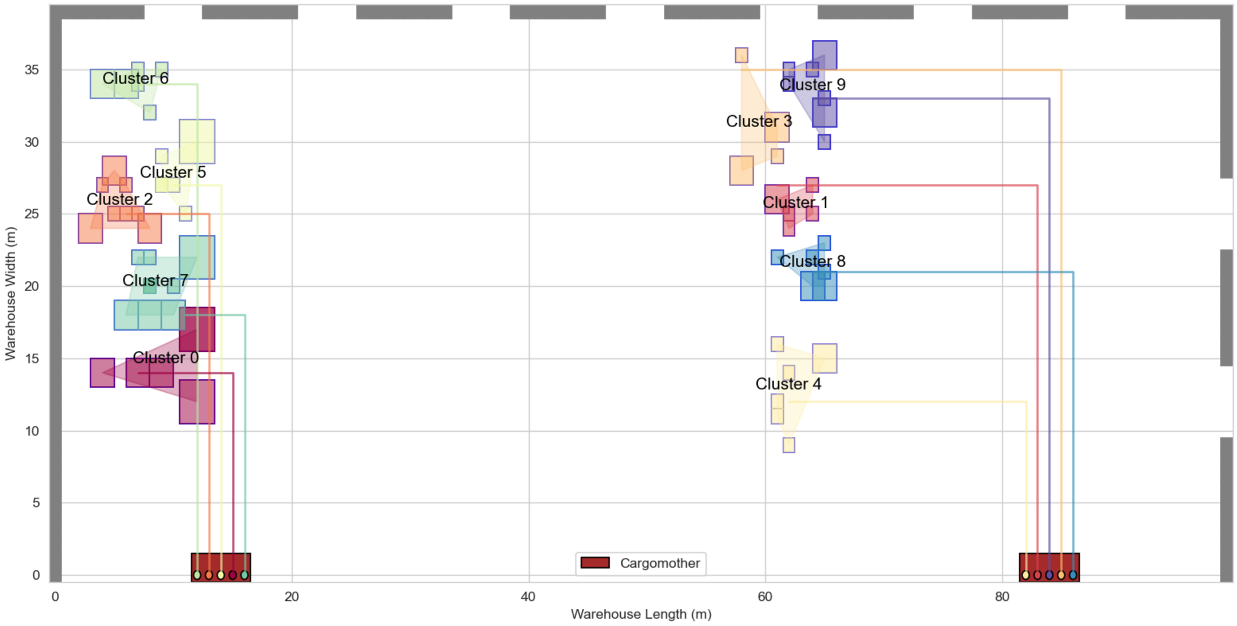


Figure 6.17: Planned trajectories for drones to their assigned clusters

The computational time for this step takes on average: 30 seconds on CPU. However, this time drastically reduces to less than 1 second when the algorithm is run for only one drone at a time (how it would be in real life). The operational phase takes on average: 0.5 - 1.5 minutes per drone assuming a speed of 2 m/s on the flight level of global planning.

5. Local trajectory is employed to navigate within a cluster and between drones (see Figure 6.18)
   - Each drone plans its path to span the entire cluster area.
   - Dynamic objects are modeled in the environment.
   - Real-time collision avoidance is employed to avoid humans and forklifts.

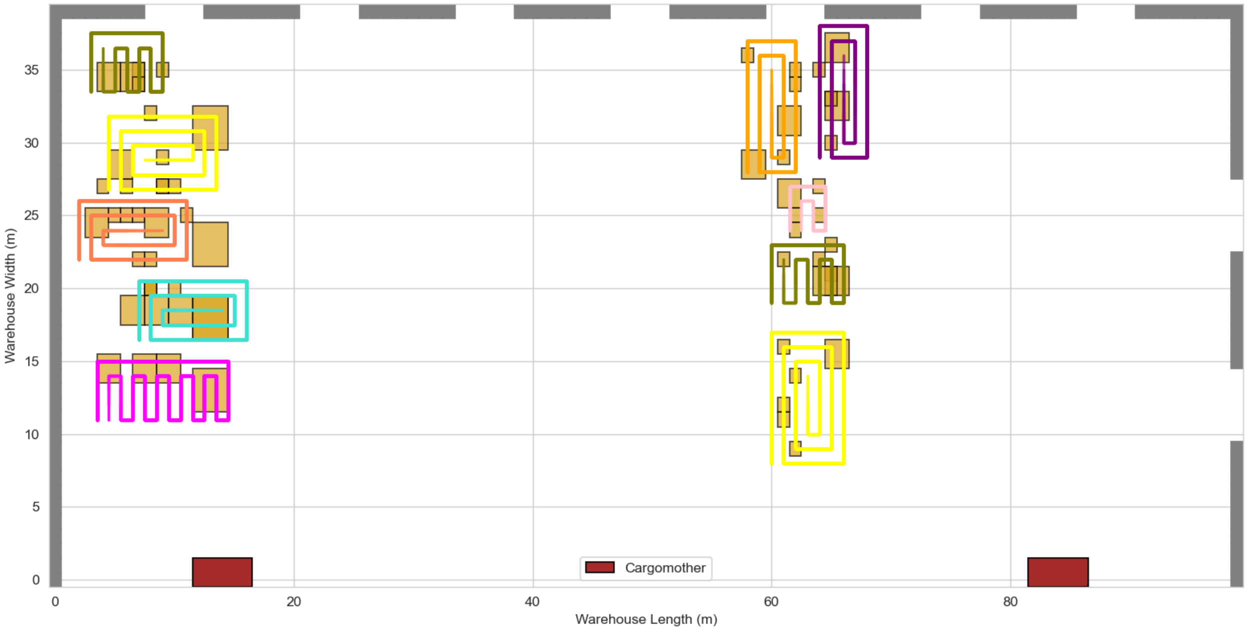


Figure 6.18: Planned trajectories for drones within their assigned clusters

Within one cluster, the trajectory used by the drone can be visualized in more detail in Figure 6.19. Here, the path is simulated including the tuned controllers.

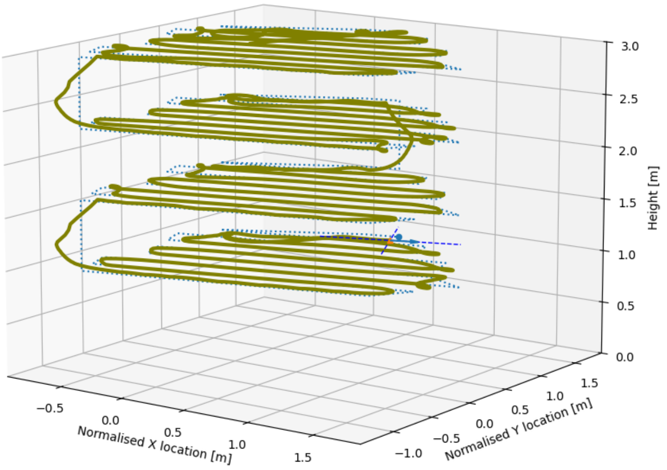


Figure 6.19: Planned trajectories for drones within their assigned clusters

The computational time for this step takes on average: 0.1 seconds on the CPU. The operational phase takes on average: 5-9 minutes depending on the cluster size (assuming 7 to 15 cargo items per cluster).

6.10.1. Compliance Analysis

Having performed software verification, the compliance of the subsystem to its requirements can be verified. The methods used to show this compliance are shown in Table 6.7.

Table 6.7: Compliance matrix for Operations subsystem

Identifier	Method	Compliance
CRG-OPS-01	Inspection with Python visualization tools; furthermore Simulation: through drone simulators such as AirSim/Unity	✓
CRG-OPS-02	Inspection to ensure the planning algorithm is correctly employed on the onboard computer.	✓
CRG-OPS-04	Simulation testing of drone-server communication in OMNeT++.	TBC
CRG-OPS-05	Analysis to compute the noise levels from the Blade Pass Frequency (BPF) and propeller topology.	TBC
CRG-OPS-06	Simulation to model random behavior of human operators in the simulation environment and assess compliance.	✓
CRG-OPS-07	Simulation to implement a simulation environment for the system and model manual control.	✓
CRG-OPS-08	Simulation to implement the kill switch functionality in code and test it in the simulation environment.	✓
CRG-OPS-09	Test the implemented the check for receiving the signal.	✓
CRG-OPS-10	Inspection to ensure that the operator room is positioned accordingly.	✓
CRG-OPS-11	Demonstration server area requirements are outlined.	✓
CRG-OPS-12	Inspection & Simulation to ensure drones return to a “safe” location on demand.	✓
CRG-OPS-13	Simulation switch between strategies.	✓

6.11. Future Recommendations

The implementation of the fleet management and trajectory planning algorithms can be extended. Utilizing reinforcement learning for auction-based task assignments can optimize task distribution among the fleet. In this approach, tasks are treated as items in an auction, and fleet units place bids based on their current state and capabilities. Over time the nuances of task assignment will be learned by the fleet.

For future trajectory planning, the incorporation of the Theta* algorithm can be considered on the global level. Theta* offers more flexible path-planning capabilities compared to traditional algorithms, allowing for moving along non-neighbouring nodes and optimized paths through continuous space. This flexibility is crucial for navigating warehouse environments efficiently and saving power. Additionally, the current local planning configuration can be further enhanced by changing the objective function from a coverage-maximization goal to a dual coverage and power use optimization.

From a scalability point of view, future development steps involve devising an extended logistics plan that takes into account the extra constraints of operating in the Bay Area and/or at the Breakdown Areas. Operating at all of these locations further increases the flexibility of the product and offers the opportunity for a complete cargo item tracking inventory. However, this comes at the expense of increased computation time and more complex collision avoidance architectures.

7: Payload

The payload subsystem allows for the recognition and measurements of cargo items. This chapter elaborates on the design procedures taken for this integration. It starts with introducing goals and the functions of the subsystem in Section 7.1, followed by the requirements and the risks in Section 7.2. The design process is further detailed in Section 7.3, emphasizing verification and validation in Section 7.4, and finalizing the chapter with recommendations in Section 7.5.

This subsystem is responsible for conducting the measurement of cargo items to determine their volume, as a final output of the entire system. This activity forms the core of the mission need statement and thus the main functionalities of the system as a whole. Forming the essence of the final product, the payload is given the utmost attention. In its final form, it equips individual drones with computer vision and allows for stereometrical measurement capabilities for topology determination. This is done via hardware and software, used to acquire and process data, respectively. This chapter further details the implementation of this as part of a flow-down procedure from the mission statement and the requirements. It then finally presents the specifications and capabilities of the designed subsystem.

7.1. Goals & Functions

To provide such functionality, a goal was set in an attempt to evaluate the success of the payload subsystem further. This ensures that the payload operations feed into this goal, and the operational cycle allows for the above-mentioned measurement to be conducted. The goal is as follows:

To implement a real-time procedure for each aerial unit that recognizes and outputs the dimensions of (rectangular) cargo items using a combination of software and hardware.

The established mission objective and subsystem goal necessitated specific functionalities for the payload subsystem, to enable its successful accomplishment. These functions are listed below, essentially identified in earlier phases of the design and here provided in more detail.

1. Identify static and dynamic unknowns and read labels attached to the static unknowns;
2. Locate and document the probable position of the cargo items within the global map;
3. Obtain three-dimensional information about the target;
4. Process the data stream from the payload hardware onboard;
5. Determine the width, length, and height dimensions of the target for volume estimation.

7.2. Requirements & Risks

This section introduces the payload-specific requirements and risks associated with not fulfilling each of the presented functions or requirements. The items presented in Table 7.1 primarily drive the design. Table 7.2 ensures the quality of the final product by adding a loop to the design, where redundancy and exception handling are taken into account, leading to a more robust result and perhaps alternate solutions to still be able to fulfil the stakeholder requirements.

7.2.1. Requirement and Risk Specification

From the identified functions, relevant requirements were generated. These requirements reflect the operations of the payload and allow for traceability of the actions that the payload takes for identification and measurement. Table 7.1 shows the subsystem requirements and the associated risk identifications.

Table 7.1: Requirements for Payload subsystem

Identifier	Payload Requirement	Associated Risk ID
CRG-PAY-01	The payload measurement device shall have a depth error of less than 1%.	RSK-PAY-01
CRG-PAY-02	The payload shall finish its operations in 30 seconds per measurement cycle.	RSK-PAY-02
CRG-PAY-03	The payload shall have a 15s data processing rate per target to meet quantity requirements as postulated in CRG-STK-02 (throughput requirement).	RSK-PAY-03
CRG-PAY-04	The payload shall be able to tell dynamic and static unknowns apart.	RSK-PAY-04
CRG-PAY-05	The payload shall correctly identify unobstructed labels of cargo items within a distance of 1 meter.	RSK-PAY-05
CRG-PAY-06	The payload shall acquire data about the requested properties of the scanned targets.	RSK-PAY-06
CRG-PAY-07	The payload shall be able to transmit all data gathered to the Flight Computer.	RSK-PAY-07
CRG-PAY-08	The payload shall not have lighting that produces dangerous stroboscopic effects on moving parts during normal operation.	RSK-PAY-08
CRG-PAY-09	The payload shall be fitted with means to signal its movements during normal operations.	RSK-PAY-09

Not fulfilling these functions poses risks to the success of the operation of the system. The associated risks are supplied with a mitigation plan in case failure occurs. These risks along with their mitigation can be found in

Table 7.2. It must be noted, however, that the overall system is less sensitive to the failure of an individual target measurement. This is because the top-level requirements only specify a throughput minimum to be maintained. The measurement of all targets is not necessitated by any parties.

Table 7.2: Risks for Payload subsystem

Identifier	Payload Risk	Likelihood	Severity	Mitigation
RSK-PAY-01	The measurement device is too inaccurate for precise measurement purposes	3	2	Intending to implement a robust registration algorithm, the pipeline accounts for parts of the target missing from some image frames and combines more frames to obtain a better statistical representation of the target
RSK-PAY-02	The payload takes longer than 30 seconds per measurement cycle.	1	3	The pipeline was tested using a test camera and a laptop. It is well within 30s, going over this would indicate more severe failure
RSK-PAY-03	The processing rate is too slow for the measurement cycle	2	5	Operations assumes that the measurements and label reading take 30s, out of which the data processing is 15s therefore this risk is dependent on RSK-PAY-02 and on operations
RSK-PAY-04	The drone fails to identify dynamic and static unknowns in its vicinity	3	4	More training data can be acquired with rigorous testing to ensure this doesn't happen
RSK-PAY-05	The drone is not able to read label information	5	2	Label reading operations are optional, this can be improved but does not impede measurements if the label cannot be read
RSK-PAY-06	The payload fails to acquire requested data about the target	2	2	As it is not severe to fail in a single case as long as throughput is achieved, it is less mitigated. However, individual tests on the modules have been performed and are being improved continuously
RSK-PAY-07	The payload fails to transmit data to the Flight Computer	1	2	In all test cases run, no such occurrence happened. The exception handling module can flag this issue and if it persists, the drone can be sent to maintenance
RSK-PAY-08	The payload lighting causes dangerous stroboscopic effects	1	3	Regulatory requirement. According to the electrical diagram, the lights are connected in a way that no oscillation in the circuit can occur. Also, the lights are diodes and there is not enough voltage to break down the diode, enabling reverse currents.
RSK-PAY-09	The payload fails to signal the drone's intended states	1	3	Regulatory requirement. Multiple simple lights are incorporated in the design for redundancy

7.2.2. Operational Environment Description and Assumptions

There are not only internal interfaces within subsystems, but also external interfaces of the full system with the environment in which it operates. Stemming from its desired functionalities, the payload is equipped with abundant environment-sensing capabilities. Therefore, the payload subsystem provides one of the primary interfaces between the system and the environment. To achieve a working solution, some assumptions have to be made about how the payload functions and how the cargo items relate to the payload.

In principle, the system is designed to operate in an indoor environment, inside cargo processing buildings located at airports. These facilities deal with the flow of cargo meant for import, export, and transit, and feature a large indoor space with storage shelves, build-up, and breakdown locations, and cargo acceptance areas [2]. All information regarding the layout and other intricacies is obtained from the previously cited resource and via personal communication with warehouse employees.

Following this consultation, the concluded assumptions are stated in the list below, together with some explanation of their impact on the system design and the relation to the stakeholder requirements. The usage of these assumptions is agreed upon with the users during a personal meeting[1].

- **Cargo items will be comprised of boxes:** cargo boxes, which eventually become the targets of the payload subsystem, are items enclosed in packing material that maintains its shape. In contrast, bulk material does not have a well-defined shape and may also deform. This definition is needed to be able to reliably identify and measure targets. It is also limiting in the sense that the system is unable to identify and measure other items, however, the majority of the cargo is of this form, therefore the proposed throughput requirement shall be still satisfied.
- **The boxes will be cuboids:** this prior is needed for a more robust implementation of the measurement pipeline. Together with the assumption above, the system could be upgraded for more general usage, as the only limitation to do so at this stage is the development schedule.
- **Targets will not touch:** for an initial robust implementation, it is assumed that the cargo items have faces that are not occluded by other planar faces, such as other targets.
- **Cargo items will be on the ground:** for similar reasons, it is assumed that the targets are on a large-level planar surface.
- **Targets will be considered as static unknowns:** it is assumed that once the Cargoscout creates the probability map, the real scene does not change, hence the operational loop remains intact.

[1] Assumptions presented during Status Meeting 5, conducted on May 31 2024.

7.2.3. Subsystem Architecture

From the requirements, the functions of the subsystem, and the assumptions, a high-level concept was devised for the payload, encompassing all functions and laying out a subsystem architecture. This is kept at an abstract level, to provide a framework for organizing and documenting the design process.

To perform all functions of the subsystem, two main physical components were deemed necessary: a measurement device and a data processing unit. The selection and integration of these are later specified, in Section 7.3. Software is also a major part of the subsystem and is considered as part of the data processing unit. The design and verification of the algorithms used are also presented later in the chapter.

A system description of the payload subsystem is provided in Figure 7.1. The elements, interfaces, and links are discussed in the following sections as part of the flow-down process from the requirements to the final results. This is done by first treating functions 1) - 2) on object detection from Section 7.1, followed by 3) - 5) about volume measurement.

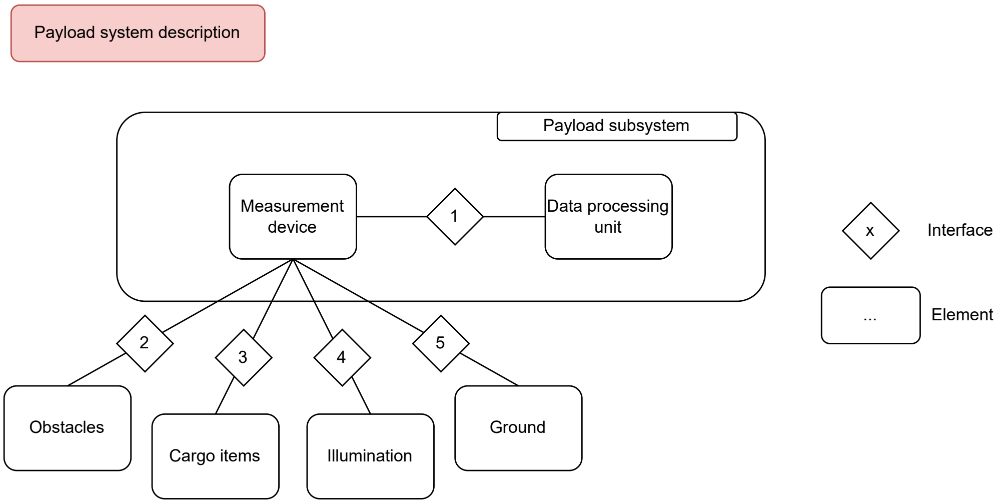


Figure 7.1: System description diagram for the payload subsystem. The block depicts the payload subsystem elements, with links and interfaces between them and environment elements highlighted

The main function is encapsulated in the third interface in Figure 7.1. The measurement device has to acquire sufficient information about the target, eventually outputting the required dimensions. The second interface covers the collision avoidance requirements. The awareness of the environment in this sense is provided partially by the payload to the collision avoidance algorithms. Furthermore, obstacles can obscure the sight of the targets. Interfaces four and five focus design efforts on the possible effects of the environment, especially illumination on the behavior of the measurement device and the stable reference that the ground can be during measurements. Finally, the first interface makes sure that the acquired data is transferred to the processing unit.

Work up to this moment already illustrated a versatile subsystem, that is equipped with environment-sensing capabilities as well as onboard computational power. This implies that in terms of engineering budgets, the payload subsystem can potentially cover parts of the budgets of other subsystems, too. For instance, this includes the computing required by navigation Chapter 8 for SLAM or the local planner in Chapter 6.

7.3. Design Process

The design process is aimed at devising a solution for the elements in Figure 7.1, which complies with the requirements taking into account the additional risks and interfaces. The process is documented following the order of the main subsystem functions, as initially presented. The organization considers object detection (Section 7.3.2), label reading (Section 7.3.3), and volume measurement (Section 7.3.4). However, the first hardware design is introduced in Section 7.3.1.

7.3.1. Hardware Design and Selection

To comply with the mission statement and **CRG-STK-01, CRG-STK-03, CRG-STK-05** user requirements, the payload shall include hardware, capable of obtaining three-dimensional information about the target and some processing unit (not limited to either hardware or software) that is able to output the desired measurements. Furthermore, for function 1) in Section 7.1, the hardware shall be capable of providing input data for object detection and label reading. In the scope of the design, only RGB imaging units (traditional cameras) are considered as an option to fulfil this function.

Multiple ways of measuring a scene's three-dimensional information in real-time exist in indoor and human presence-friendly applications. These latter constraints entail that contact measurement methods, as well as options using ionizing radiation, are eliminated. The use of (near-) visible light therefore seems obvious, however, other methods might be suitable as well, such as acoustic imaging. The available solutions are divided into active and passive methods. In active methods, the payload emits signals to assist with measurements, whereas in passive methods, the payload simply collects input without any signal emission.

Both active and passive measurements can gather depth information by exploiting lens focus/defocus information [14, 15]. In general, this is a less precise way of fulfilling the function stated above, while also being less readily available as integrated systems. Regarding active methods, it is possible to use the deformation of a projected pattern to compute the possible three-dimensional shape of objects in a scene [16]. This method also has less heritage and it poses its challenges too [17]. A similarly disadvantaged method is shape from shading. The problem in most cases is ill-posed [18], with scarce applications found [19].

Sonar methods, covering the use of sound for navigation and ranging [20], can also be used to obtain depth information from a scene. However, indoor applications of this method for robotics have only been recently proposed [21, 22]. It is not clear whether this solution would be able to fulfil the requirements with great certainty.

The final hardware choice to fulfil function 3) proposed in Section 7.1 potentially falls on triangulation methods, implemented in the form of an active sensor aiding measurements in low-light and featureless environments, too [23–25]. These sources present readily available products with design heritage and industrial applications. The method employed by these products is based on binocular vision to obtain a stereo pair of images to calculate depth [26]. It also projects a light pattern to generate perceived texture, helping in the correspondence problem described by the same source. The light pattern and nature can vary [27, 28], however, in most commercial products, it is a grid of dots emitted by an Infra-Red (IR) projector. This is a suitable choice for indoor applications, due to the lack of IR noise otherwise present in outdoor scenes. Since there are COTS solutions that have the potential to meet the requirements and fit within the engineering budgets, this method is compatible with the envisioned system.

The last method to investigate is the active Time of Flight (ToF). It measures the time it takes for the light to travel from a light source on the camera to a reflective surface and back to the camera [29]. This option is also considered a viable one, due to similar reasoning. It is virtually insensitive to lighting conditions or motion blur.

The investigation of these methods covers the interface between the measurement device and its environment (obstacles, cargo item, illumination, and ground) as per Figure 7.1. The primary function is the resolution of interface three, according to the figure. Both active triangulation and ToF methods can provide depth information of a scene with sufficient accuracies to meet **CRG-STK-05**. The second interface relates to the collision avoidance functions, treated in Section 8.3.2. In the above discussion, interface four is also treated by addressing the sensitivity to scene illumination.

Channeling these models into physical implementations requires a detailed analysis to consider the hardware integration for the payload. COTS products have been narrowed down to the options presented in Table 7.3 on the basis of engineering budgets and system integration considerations. For instance, ToF Light Detection and Ranging (LiDAR)s tend to be incompliant with the cost and mass budgets, with a number of existing depth vision systems with less design heritage too.

Table 7.3: Final Payload Options and Integration Parameters

Camera Options	**Intel RealSense D450 Depth module + Vision processor D4 board V3**	**Orbbec Astra Mini Pro**	**Onion Tau Lidar & RPI Camera module**
Size [WxDxH, mm]	124×26×29	84.90×20.0×19.92	Lidar: 90×41×20 Camera: 25×24×11
Max data rate [Mbit/s]	3262	550.5	18.43
Weight [kg]	0.055	0.055	0.104
Power consumption [W]	<3.5	2.4	1.25
Total cost [€]	250	160	180

The options presented in Table 7.3 are still somewhat modular. For instance, the Tau LiDAR is combined with an RGB camera in order to accommodate the label reading functions. Furthermore, in the rightmost entry, a custom-integrated solution is proposed with a vision system, combined with an onboard computing unit.

To be able to finalize the hardware configuration, the data processing side of the payload is considered so that the functions 4) and 5) in Section 7.1 can be fulfilled.

The goal in Section 7.1 emphasizes a real-time implementation for an aerial unit. This means that the pipeline from data acquisition to the output of the numbers describing the target dimensions has to run in an amount of time that is comparable to the measurement itself and shall not exceed 30 seconds according to **CRG-PAY-02** in Table 7.1. Furthermore, it has to apply to all the drones that are operating. This does not necessarily imply that the entirety of data processing happens on each individual drone.

In the case where the data acquired by the hardware is sent to a ground station, the drone does not have to be equipped with a rather powerful onboard computational unit. However, this would transfer the burden of data handling to the communications subsystem. The size of such data for one single timeframe is on the order of megabytes. Considering the system as a whole with multiple drones actively operating, onboard processing of depth data is chosen to limit bandwidth requirements and to take communications failure out of **RSK-PAY-02, RSK-PAY-03 and RSK-PAY-06** in Table 7.2. The implementation of an onboard computational unit is included in Table 7.3 and has the added benefit that it can potentially cover all necessary command handling and algorithm running other than the payload, facilitating other subsystems' needs as well.

Table 7.3 considers the NVIDIA Jetson Graphical Processing Unit (GPU)[2] for onboard computational purposes. Further research brought attention to the Hailo-8L Tensor Processing Unit (TPU)[3]. This is similar to a GPU, but it is a better integrable solution offering less weight and power consumption, at the cost of slightly decreased computational power. However, both this solution and the Jetson GPU have similar characteristics to an average laptop. Therefore, using the available laptops as benchmarks, the choice of the TPU can be verified for the sake of better integration. In case computational performance is lacking, the NVIDIA solution can potentially offer superior capabilities compared to the TPU. It also has to be mentioned that TPUs and GPUs are designed to perform repetitive computational tasks, such as matrix operations, and are less suitable for process-oriented applications, such as multi-threading. Fortunately, point cloud and range image-related computations inherently involve repetitive geometric calculations, implementing such hardware meaningfully.

With the added weight of the computational unit, there is no room for a heavier ToF solution in the form of the Tau LiDAR. Amongst the remaining options, the Intel camera is deemed to be more suitable, featuring a larger baseline length and more design heritage. These aspects are taken into account to help mitigate the risks in Table 7.2. The added cost that comes with the choice of this solution is still within budget. The summary of the payload hardware is included in Figure 7.2.

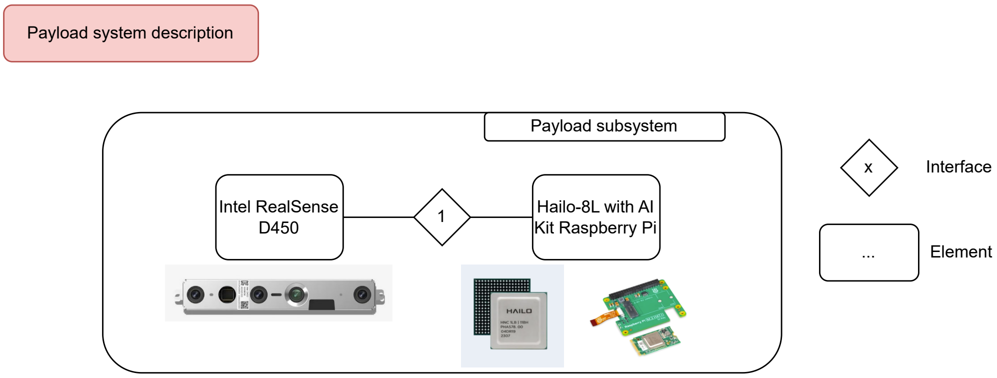


Figure 7.2: System description with added hardware. This is a supplement to Figure 7.1, presenting the implementation of the system elements

7.3.2. Object Detection

Object detection is a common problem in computer vision that involves identifying and possibly classifying objects in a frame. Depending on the scenario, object detection can focus on accuracy (how accurately the object is detected) or inference speed (how fast the object is detected)[4]. This detection relies entirely on the frames generated by the RGB camera wherein all the frames are generally input into a model for inference purposes.

[2] IMX291 and its integration with the Jetson Nano board: https://www.waveshare.com/product/imx219-83-stereo-camera.htm. Accessed 22 May 2024.

[3] Hailo-8L Entry-Level Artificial Intelligence (AI) Accelerator: https://hailo.ai/products/ai-accelerators/hailo-8l-ai-accelerator-for-ai-light-applications. Accessed 16 June 2024.

[4] Object Detection https://paperswithcode.com/task/object-detection. Accessed 5 June 2024.

These frames are dense in pixel data, in this scenario even with three channels, which is a hurdle for conventional neural networks to train on.

Selection of Architecture

For these considerations, there is a multitude of model architectures available, namely You Only Look Once (YOLO), Faster-Region Convolutional Neural Network (Fast-RCNN), and Single-Shot Detector (SSD) [5]. For Cargonaut, inference time is paramount given that the cargo scanning has to be under 30 seconds, given the requirement **CRG-PAY-02** in Table 7.1. This strict condition already eliminates Fast-RCNN structures, and the condition of detection of multiple objects in a singular frame eliminates SSD structures. On top of that, ease of training coupled with the scarcity of resources makes the YOLO architecture the better option, therefore YOLO was chosen as the computer vision architecture [6]. This is also a valid trade-off to make, given that YOLO is preferred for real-time applications due to the good balance between the model accuracy and inference speed.

Model Architecture

The YOLOv8 model uses convolutional layers and residual connections to transform the input frame machine-interpreted inputs that feed into dense and sparse prediction layers that are responsible for bounding the detected object and classifying. This is visualized on Figure 7.3.

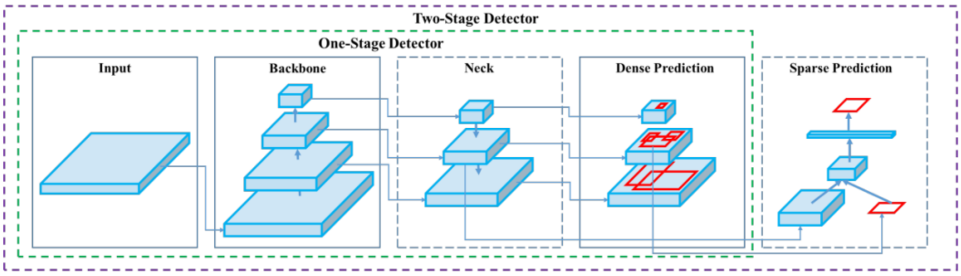


Figure 7.3: YOLOv8 simplified architecture, Viso.ai [7]

This architecture consists of a two-stage detector, which not only segments the object, but also classifies it. Therefore, the model is both a regressor and a classifier. For the operations of Cargonaut, and to be able to distinguish static and dynamic unknowns, the sparse detection head has been changed to include seven neurons that classify seven classes; namely 'cardboard box', 'drone', 'barcode', 'forklift', 'person', 'QR code', 'wood pallet'.

Given the scarcity of resources, transfer learning was applied where pre-trained layers were frozen[8]. These layers retain the weights that were the result of the architecture being trained on a much larger dataset by Ultralytics which is the Common Objects in Context (COCO) Dataset[9]. Once the model is loaded through Ultralytics, it already comes with the training loop that was verified and validated externally. This speeds up the pipeline-building procedure as the minimum effort was required to construct the training pipeline.

Data Collection and Training Loop

For the training of YOLO, a training, a test, and an optional validation directory containing images of the aforementioned classes with their bounding box coordinates are required. This was accomplished by combining three publically available datasets retrieved from Roboflow Universe[10]. These datasets are *Logistics Computer Vision Project*[11], which consists of 69,379 training images, 19,885 validation images and 9,974 testing images; *forklift det Computer Vision Project* [12] which consists of 2,311 train images, 451 validation images and 265 testing images; and *People Detection Computer Vision Project* [13] which consists of 5,070 train images, 1,431 validation images and 759 testing images.

[5] YOLOv8 vs SSD: Choosing the Right Object Detection Model https://keylabs.ai/blog/yolov8-vs-ssd-choosing-the-right-object-detection-model/. Accessed 12 June 2024.

[6] Comparative Analysis of YOLO and SSD https://medium.com/@nikitamalviya/comparative-analysis-of-yolo-and-ssd-7433e249d429. Accessed 12 June 2024.

[8] Transfer learning with frozen layers https://docs.ultralytics.com/yolov5/tutorials/transfer_learning_with_frozen_layers/. Accessed 12 June 2024.

[9] Models https://github.com/ultralytics/ultralytics. Accessed 12 June 2024.

[10] Explore the Roboflow Universe https://universe.roboflow.com/. Accessed 24 June 2024.

[11] People Detection Computer Vision Project https://universe.roboflow.com/large-benchmark-datasets/logistics-sz9jr. Accessed 15 May 2024.

[12] forklift det Computer Vision Project https://universe.roboflow.com/nithesh-sanil-iud92/forklift-det. Accessed 15 May 2024.

[13] People Detection Computer Vision Project https://universe.roboflow.com/leo-ueno/people-detection-o4rdr. Accessed 15 May 2024.

Simple pre-processing was required given that the datasets involved contained various other classes that had to be deleted before the training sequence. This pipeline also required downloading the pre-trained weights, initializing the model, and starting the training loop with both processes combined. This is visualized on Figure 7.4.

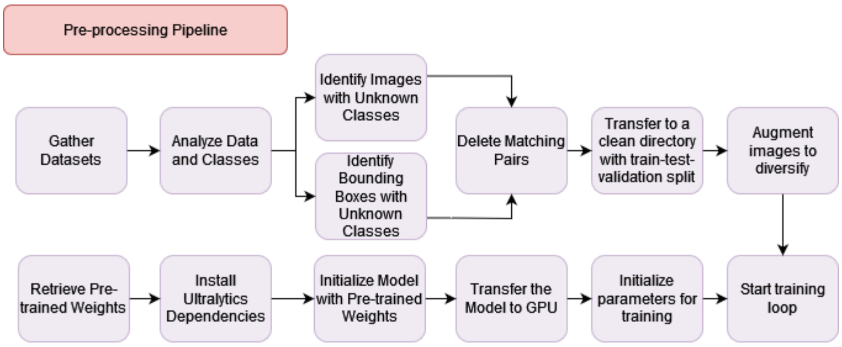

Pre-processing Pipeline
Gather Datasets
Analyze Data and Classes
Identify Images with Unknown Classes
Identify Bounding Boxes with Unknown Classes
Delete Matching Pairs
Transfer to a clean directory with train-test-validation split
Augment images to diversify
Retrieve Pre-trained Weights
Install Ultralytics Dependencies
Initialize Model with Pre-trained Weights
Transfer the Model to GPU
Initialize parameters for training
Start training loop

Figure 7.4: Pre-processing pipeline for training process

The pre-processing also includes image augmentation, which is done when images are artificially altered and added to the dataset to create variety. These alterations were vertical and horizontal flips, adding Gaussian blurs, adjusting the brightness, and randomly cropping. For every QR code and barcode image within the dataset, 30 more augmented images were generated and the alterations were also reflected in the bounding box coordinates. These augmentations in the end generated around $20{,}000$ additional images for the whole dataset.

The training loop was already provided by Ultralytics, and the loop was initialized for 25 epochs with 4 batches. The loop was run on NVIDIA Quadro T1000 with Max-Q Design with 4 workers. The data has been cached to memory, and per epoch, the weights of the best and last models have been saved so as not to lose progress during training. This has also allowed for resuming the training on another device or recording the waypoints to resume later.

Training Results and Inferences

After 25 epochs of training roughly equating to 55 hours of GPU activity, the model has been tested on the entire dataset to generate a confusion matrix, which is visualized on Figure 7.5. The model consists of 3 loss components, namely, box loss which is the loss for bounding box predictions weighted 7.5 by default, classification loss weighted 0.5 by default, and distribution focal loss which is weighted 1.5 by default. The weighted average of these losses and their variance is shown in Figure 7.6.

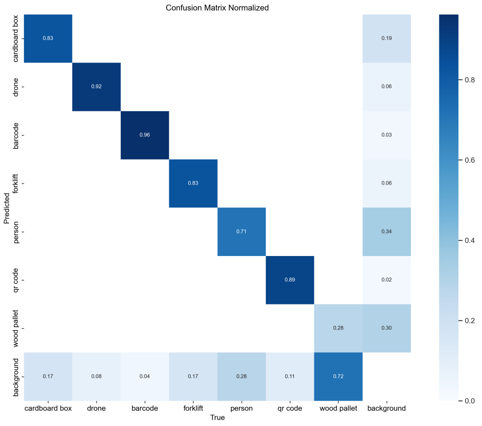


Figure 7.5: Confusion matrix after 25 epochs

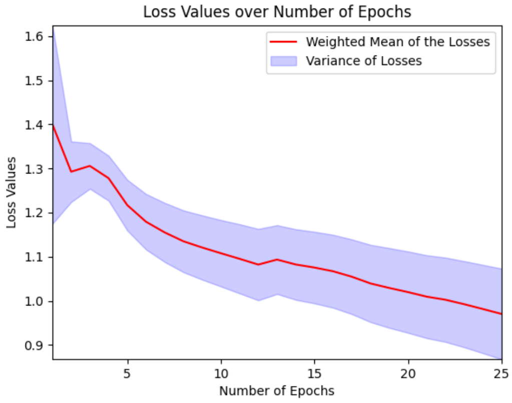


Figure 7.6: Losses after 25 epochs

From the confusion matrix, it is apparent that the model struggles with identifying wood pallets. This result is expected given the repetitive structure of a wood pallet, therefore this performance is still deemed acceptable. Given that there is a steady loss decrease, the model can be trained in the future on the same dataset for further refinement with the hopes of increasing the confidence of classifying people, as it is paramount that they are blurred from the frame to respect data privacy.

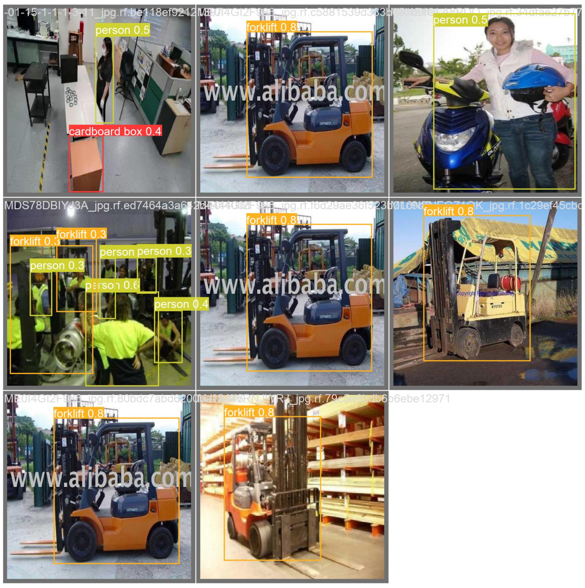


Figure 7.7: Validation batch 1 inference results

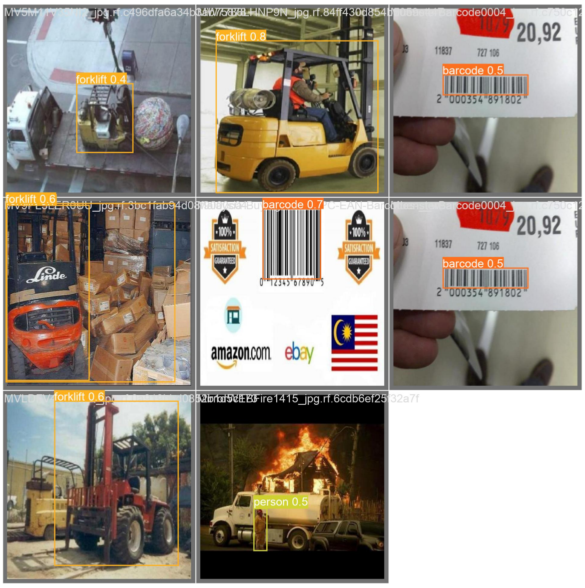


Figure 7.8: Validation batch 2 inference results

The inference results of validation batch 1 and 2 can be observed in Figure 7.7 and Figure 7.8 respectively. Most notably, the model can function when the frame is rich with various classes when the classification targets are too in-depth within the frame, and even when it has a watermark. These three elements show that the model, with further fine-tuning, is able to capture the classification elements that both operational modes would need.

7.3.3. Label Reading

Label reading consists of identifying and decoding barcodes and QR codes. This activity is to be done while a measurement cycle takes place. Reading labels is a hard activity given that the field of view has to accommodate for the smallest frame within the encoded print. This restricts the label reading to be at most 0.6 m distance from the label, which is far more stringent than other payload operations [30].

Selection of Architecture

Label reading is to be done via the onboard camera which can also act as a scanner. This is also favored in terms of mass, power, and cost budgets given that no additional hardware is required to achieve this task. The scanning activity can be achieved through the deployment of OpenCV Barcode Detector and Decoder[14] which is an advanced YOLO architecture that is trained on identifying barcodes and QR codes, but also decoding them.

Activity Timeline

Given that the label reading operations have to be done in a closer frame compared to the other payload operations, a strategy was implemented to accommodate this procedure during the measurement cycle. This is demonstrated on Figure 7.9.

The timeline involves a continuous activity of object detection and includes measurement operations that are closely linked to label reading. Once a cargo item has been measured, Cargonaut will increment the count of the cargo items it has registered, and this will be recorded to cross-check the count received by the Cargoscout. Once dimensions for the cargo item have been taken, Cargonaut will be able to encircle the cargo item and attempt a search for the label. If the label is found, this information will be then registered as additional information within the database. On the contrary, if the label reading fails, then the database will be updated such that reading registration will have a fail mark for warehouse workers to be able to track and read the label of the cargo if necessary.

In either case, Cargonaut will continue the local trajectory and continue its operations. If the count for static unknowns received by Cargoscout does not match at the end of the trajectory Cargonaut follows within a cluster, an auction will be called. This is because the local map has been hampered with by either dislocation of cargo items or other unexpected events.

[14] Barcode Recognition `https://docs.opencv.org/4.x/d6/d25/tutorial_barcode_detect_and_decode.html`. Accessed 15 May 2024.

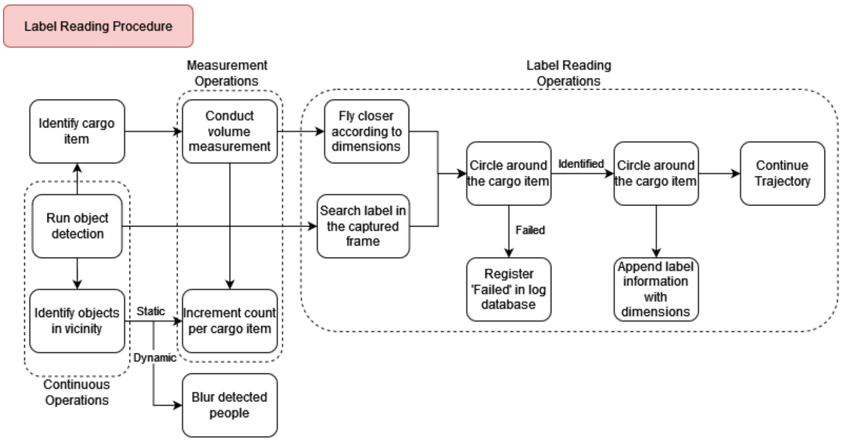

Label Reading Procedure
Measurement Operations
Label Reading Operations
Identify cargo item
Conduct volume measurement
Fly closer according to dimensions
Circle around the cargo item
Identified
Circle around the cargo item
Continue Trajectory
Run object detection
Search label in the captured frame
Failed
Register 'Failed' in log database
Append label information with dimensions
Identify objects in vicinity
Static
Increment count per cargo item
Dynamic
Continuous Operations
Blur detected people

Figure 7.9: Label reading operations

7.3.4. Volume Measurement

Previously, the hardware was put into the place of interest regarding payload. Converging first in the hardware design helps reduce the uncertainty in the engineering budgets since the software does not directly carry mass, cost, or power burdens.

The following paragraphs detail the aspects of software integration needed to fulfil functions 4) and 5) in Section 7.1. To be able to perform verification and therefore mitigate risks in Table 7.2, a less capable camera is obtained for the duration of the design for hardware-in-the-loop testing of the volume measurement pipeline. The software under development is then tested using this camera, namely the OAK-D by Luxonis [31].

In general, the output of the hardware mentioned in Figure 7.2 is either a point cloud or a range image. A point cloud is a collection of points with spatial coordinates and in certain cases other information too, such as recorded monochromatic intensity or RGB data. A range image is a pixelated image where each pixel amongst its light information contains a value on how far the content of the pixel is from the camera image plane. The goal is to *segment* the target within this data and obtain the necessary measurements. In Cargonaut's case, the chosen hardware outputs a range image which is converted into a point cloud at the beginning of the pipeline.

There exist pre-made libraries for three-dimensional scene understanding, such as a recent advancement in line mapping [32] that could be used to exploit the shape assumption as a type of prior information or a novel mapping algorithm combining depth, RGB and Inertial Measurement Unit (IMU) data[33]. There is even a complete software developed for similar purposes, called COLMAP [34]. These open-source libraries claim superior performance and novel advancements in the topic. However, as a pre-implemented solution, they do not offer full scalability and adjustability. A significant effort would be required to analyze each of them and understand the dependencies to be able to make a choice and implement it in the given schedule. Therefore, to account for possible future developments of the system in terms of flexibility and general improvements, a more basic and modular approach is chosen for software integration. Furthermore, some of these open-source libraries are implemented in Python. Other languages, such as C++, might offer more efficient runtime execution and memory handling, needed to solidify the real-time application.

Fortunately, a modular, large-scale, open project for 2D/3D image and point cloud processing exists in the form of a C++ library, called Point Cloud Library (PCL) [35]. PCL is not tailored to any specific applications. Rather, it is a collection of modules that incorporate 3D processing algorithms. To give an introduction on the topic, phraseology used, and what is possible with simple computer-vision tools, the relevant modules of the library are explained below. The order is presented in a logical flow, in the form it is usually implemented in three-dimensional scene understanding pipelines, such as the examples presented in [35].

- **Filtering:** this module is used to remove certain points that do not fit into a point cloud following a statistical analysis or to downsample a collection of points for faster processing. A statistical outlier removal is an example of the former. Based on the distribution of point-to-neighbor distances, more isolated points are filtered out. For downsampling, a voxel grid filter is implemented. This filter builds a spatial grid into the three-dimensional scene and every point inside one voxel is approximated by the centroid of the voxel. This is a similar practice to increasing the pixel size in the case of a two-dimensional image.
- **Key points:** according to the documentation of the library, key points are points within a point cloud that are relatively unique and might appear distinctively in other images taken from the same scene. Key points might belong to edges, corners, or other interest points. This can also be interpreted as a form of

downsampling practice. By identifying the key points, the rest of the pipeline can only focus on these to perform operations, reducing the computational effort and focusing only on points with more encoded information, analogous to increasing the signal-to-noise ratio as part of a transmission process.

- **Features:** in a broad sense, features are additional information about points in a three-dimensional scene. They are mostly used to compare different captures from the same scene. Simply comparing spatial information might be perceived as meaningless, as there is no guarantee that the same surfaces are visible in the point cloud. Features provide better characteristics and metrics to be able to distinguish between geometric surfaces [36]. A widely-used descriptor is proposed in [37]. This source claims informative pose-invariant local features for a point using the k-neighbor search method and faster runtime compared to previous works.
- **Registration:** according to the PCL documentation, registration involves the combination of several point clouds into a globally consistent model. It can be used to compile a scene where images are taken from multiple viewing angles, therefore capturing surfaces that are occluded from certain poses. Without features, registration is a challenging issue [37] due to optimization algorithms reaching local minimums near the global optimum. The output of this process is usually a rigid transformation matrix which is a result of a 6-Degree of Freedom (DoF) (translation + rotation) optimization problem. This document uses the term “source” to describe an incoming point cloud for which a transformation is to be estimated to fit onto a “target” cloud.
- **Segmentation:** once a sufficiently dense complete point cloud is obtained by using the methods mentioned above, segmentation can be performed to cluster together parts of the three-dimensional scene based on some common properties or model fitting. After this, clusters can be processed independently to extract desired data.
- **Sample Consensus:** segmentation mostly relies on sample consensus, such as RANdom SAmple Consensus (RANSAC) [38]. These methods search for points that fit some model, such as a geometrical shape.
- **Miscellaneous modules:** other modules include an Input/Output (I/O) and visualization implementation to read/write and visualize point clouds, respectively. The native data file for PCL has the `.pcd` extension and is referred to as Point Cloud Data (PCD) file from this point onwards.

Using the modules described above, the following presents an implementation of a cargo item measuring pipeline relying on the hardware documented earlier. The description is presented in the following order: first, the *Scene Data Acquisition* describing the interface between the camera and the processing unit is considered. Next, *Scene Data Handling* presents how the raw data is transformed into a clean, usable dataset. Finally, *Dimensions Acquisitions* describes how the final output of the system is obtained from the pipeline.

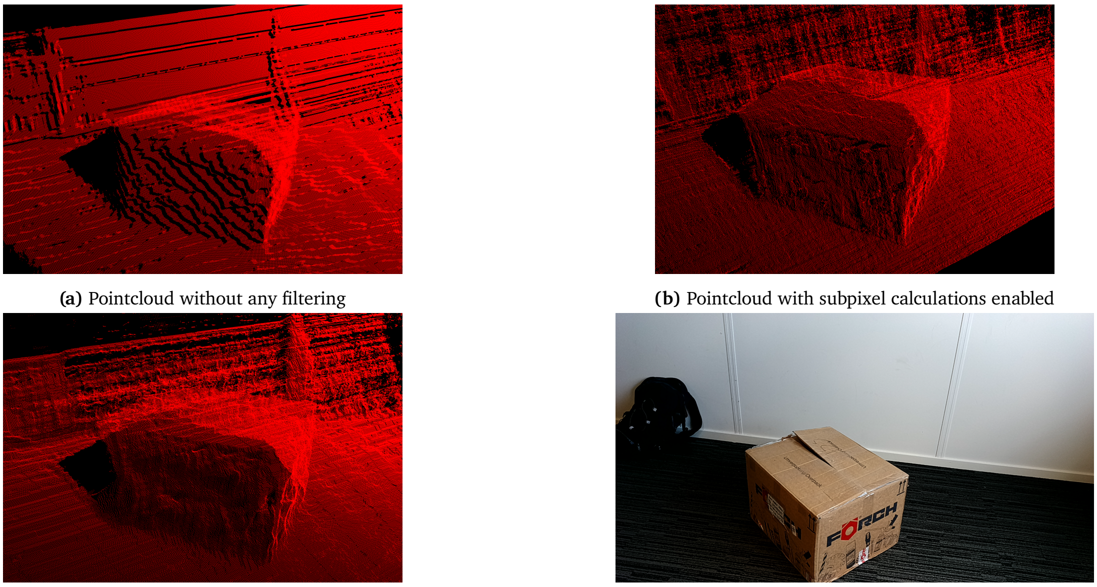
Scene Data Acquisitions

The Intel RealSense camera uses an IR projector and two color global shutter camera modules with a narrow field of view. The left and right cameras are positioned at a distance from each other, so that the scene as seen from each sensor differs slightly. Matching each point of the scene between the frames results in a pixel shift for each point in the image. From this disparity, the depth value of each pixel can be traced back, resulting in a depth image coupled with the RGB image, resulting in an Red Green Blue Depth (RGBD) image. From this image, a point cloud can be generated, which pairs an x and y position to the depth value based on the camera's physical components like focal length. Measuring this point cloud, performing segmentation and other operations is a well-researched topic and can be performed easily by the DSE team.

Generating and using point clouds and depth images is not trouble-free. For one, if the disparity is limited to an integer value, then the depth accuracy is constrained. For this reason, most cameras have built-in sub-pixel measurement modes, estimating depth to a much more accurate level. The generated cloud then tends to be usually noisy and not complete. For this, edge-preserving filters can be applied, which are usually implemented by the the camera. One such widely used option is based on [39], which filters noise and interpolates holes in the image. Due to its edge-preserving nature, the box dimensions inferred from the point cloud are not changed by applying the filter. The results of applying filters to a point cloud can be seen in Figure 7.10.

Scene Data Handling

In order to improve the data quality and to provide meaningful input to the final stages of the pipeline, the raw scene data obtained as a collection of point clouds from multiple camera poses is processed. Such depth cameras tend to produce measurement errors and noise. Furthermore, they may fail in triangulating certain parts of the scene returning no points, or parts of the target may be missing from the field of view of the sensor. Filtering the input and combining more captures could mitigate these issues. This implies the usage of registration algorithms. All of the modules above are used in the process, mostly due to the integrated nature of registration. The goal of registration is illustrated in Figure 7.11.



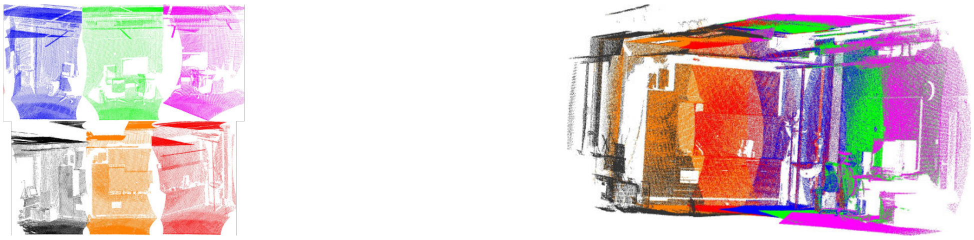
(a) Pointcloud without any filtering

(b) Pointcloud with subpixel calculations enabled

(c) Pointcloud with spatial filtering and subpixel mode

(d) Color image of the scene seen in the point clouds

Figure 7.10: Examples of point clouds with different filters of the same scene



(a) Six separate point clouds obtained from different poses in the same scene

(b) Registered images

Figure 7.11: An example of a registration process with the separate point clouds on the left transformed into one coherent point cloud on the right, taken from [35]

According to [40], registration algorithms can be categorized into five main categories. First is the traditional Iterative Closest Point (ICP) approach first described in [41] and later improved by [42]. ICP relies on an iterative way of estimating a transformation matrix with the objective of minimizing a certain error metric. The error metric is based on point correspondences, usually computed from simple Euclidean distances. In its basic implementation, ICP is rather computationally demanding and no guarantee is provided to find a global optimum.

The second set of methods involves the usage of Probability Density Function (PDF)s. One widely used method is the Normal Distribution Transform (NDT) algorithm [43]. It interprets the input data as a PDF by the normal distribution of every voxel inside the input. Therefore, no explicit correspondences have to be established. The voxel size is one of the most important parameters for this method as it directly affects the resolution, and therefore the accuracy of the method, while having an impact on the runtime too. It is a good choice however for a coarse-to-fine transformation estimation as a finishing step in a registration pipeline.

The third type of method discussed is the feature-based approach. This procedure involves the selection of a set of points for which features are generated, such as surface normal and curvature. The points selected depend on the key point identification method and the features described depend on the feature description used.

These three methods are provided as an example implementation of registration in the PCL documentation. Due to schedule constraints, only these are considered to be included in the pipeline, leaving the learning-based and other methods for future consideration.

First, the ICP has been tried. The challenges mentioned by the sources were encountered and a rather slow runtime was also observed, making it difficult to realize in the planned real-time application. The attention has been shifted to NDT, with test data obtained by the Luxonis verification device, introduced earlier.

The test scene for NDT consisted of a box on a chair. The camera has only been translated to simplify the test case. The NDT pipeline first loads the PCD files and then downsamples it using a voxel filter for faster runtime.

Once the parameters, such as the PDF voxel and the transformation step size with the convergence criteria are set, the pipeline calls the alignment function which performs the optimization. Figure 7.12 shows the scene and the results of two runs with different initial guesses. The voxel and step sizes were scaled to the dimensions of the scene but were not optimized or investigated in-depth.

(a) Initial translational guess: 150 mm. Final translation: 121 mm

(b) Initial translational guess: 200 mm. Final translation: 213 mm

Figure 7.12: A test case for the NDT registration. Source (blue), target (red), and the transformed source (green) as a result are shown. It shows a high sensitivity to the initial guess, resulting in reaching a local optimum in case an inaccurate translation is fed into the pipeline

With the goal of a robust implementation in mind, the focus is now placed on a featured-based registration method. The pipeline is similar to the ones mentioned above, with the addition of key point identification and feature description steps. A single iteration of it is illustrated in Figure 7.13. It is possible to iterate over or to conduct a collection of steps in parallel.

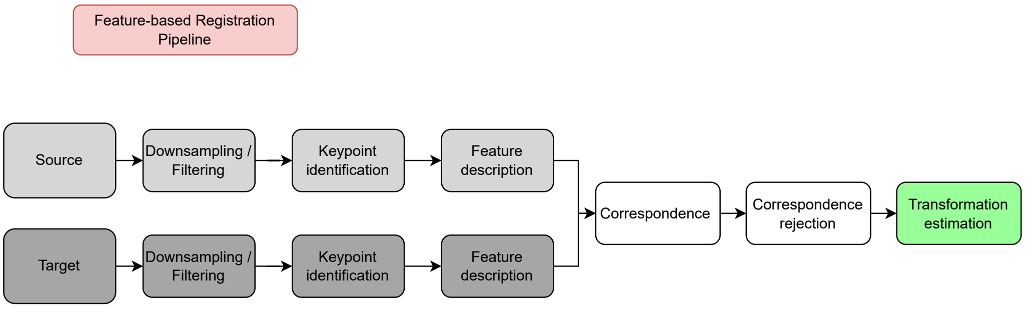


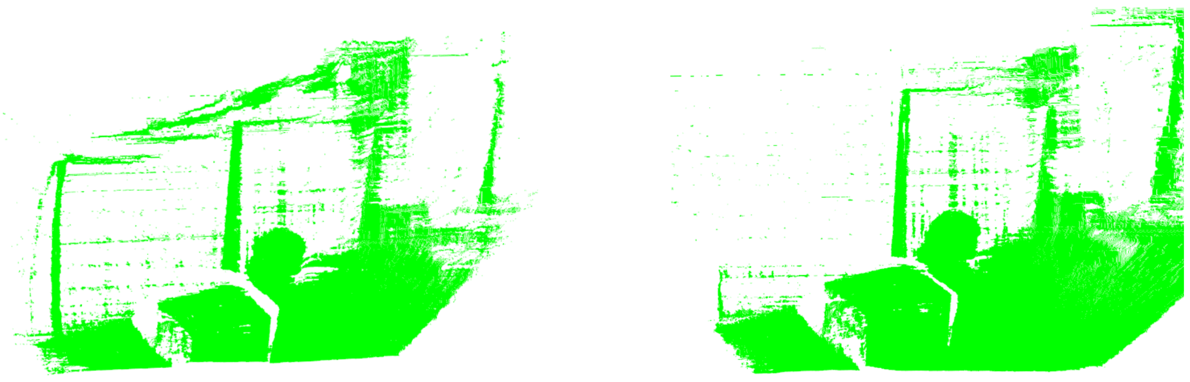
Figure 7.13: A common pipeline for a feature-based registration process; single iteration. Loops can be introduced between any points of the iteration [35]

The purpose of feature description is to introduce more certainty in correspondence matching. This is done by minimizing the search space for correspondences and by not relying on initial transformation guesses to conduct the registration. Regarding these aspects, a comparison with ICP is presented together with a proposed improvement in [44].

This time using a different scene captured by the Luxonis equipment, the two raw images are loaded from the PCD file (Figure 7.14). The scene consists of a box, filling the role of the cargo item. As per the assumptions made earlier in this chapter, it lies on a flat surface (floor). The box is placed in front of a relatively featureless vertical wall, with no other major objects present in the scene.

Several peculiarities can be observed related to this scene. Primarily, the obtained point cloud is more dense, especially the target (box, lower left corner in each scene) compared to Figure 7.12. This is due to two factors. First, improvements described earlier in this chapter as part of scene data acquisition are implemented. Second, the box itself contains more texture, aiding the disparity-matching process of the camera, resulting in more points being triangulated with a given confidence. Other specifics present in the point cloud are outliers, noise, and falsely triangulated points with large depth values. Furthermore, in the source cloud, some parts of the box are missing, making it an ideal example and test case for the registration pipeline. The goal is to have points in a point cloud that represent the edges of the box with great certainty.

Following the previous discussion and Figure 7.13, filters are implemented. It must be noted that these specificities are not limited to this scene only. By addressing all of them, a complete implementation can be achieved To remove noise and outliers, with the possibility for future improvements. First, the density of the point cloud has to be unified to some extent. Dense parts are downsampled using a voxel grid filter with cubic voxels. This also reduces the capacity represented by spatial resolution to render noise in the point cloud. Outliers are removed by a



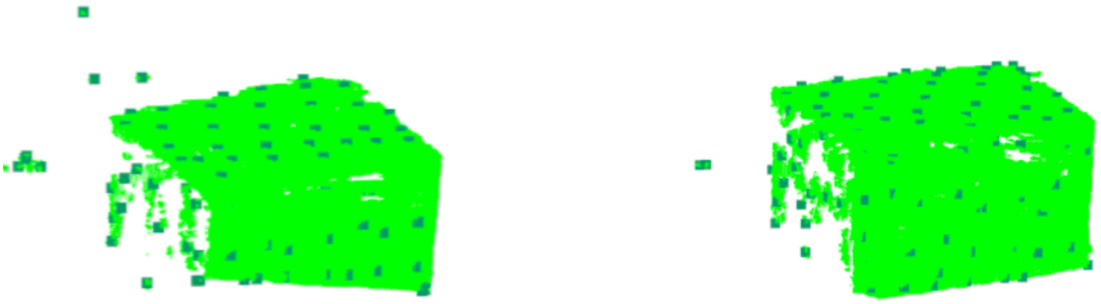
(a) Raw Source PCD

(b) Raw Target PCD

Figure 7.14: Raw captures of a test scene using the Luxonis equipment. Noise and outliers are present together with falsely triangulated depth data

distance pass-through filter limiting x, y, and z coordinates and by using the `pcl::StatisticalOutlierRemoval` algorithm to limit PCD density from the lower side.

Furthermore, to limit key point identification to the target in later stages, the largest plane (floor) is removed using a sample consensus algorithm (`pcl::SACSegmentation`). Since the pose of the floor could be beneficial to exploit priors mentioned in Section 7.2.2, the model coefficients of the segmented plane are conserved. The results of the operations so far are presented in Figure 7.15.



(a) Filtered source PCD

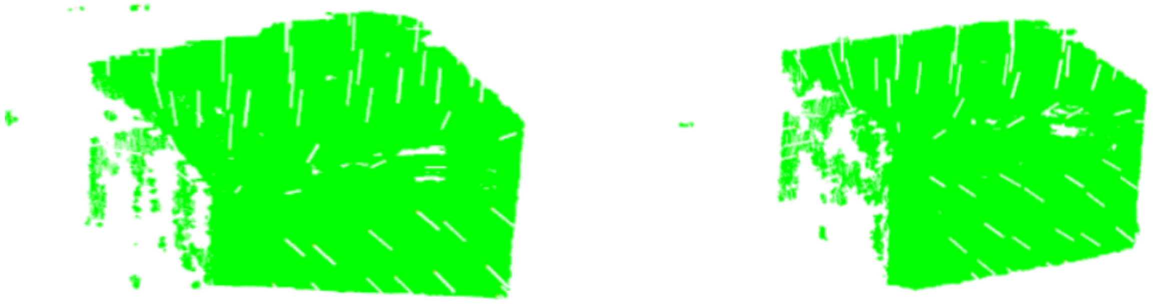
(b) Filtered target PCD

Figure 7.15: Filtered captures of a test scene using the Luxonis equipment. Noise and outliers are filtered from the raw data (light green), resulting in the filtered dataset (dark green).

The next step is the key point identification process. Key point search is a way of downsampling. However, instead of using purely statistical methods as seen before, some properties of the scene are taken into account to reduce the point cloud to points carrying special meaning. This special meaning is aimed at helping the feature descriptor in the next step. It mostly consists of a PCD conversion to a range image and searching for surface changes. A robust method is proposed in [45], which is also implemented in the pipeline.

Once key points have been identified, surface normals can be estimated. This requires an input cloud that contains the points where normals are extracted. It relies on a search cloud that provides a sufficiently dense neighbor area for these points. For the former, the key points are inserted, and for the latter, the original cloud is provided. The results are shown in Figure 7.16.

This data can now be fed into the feature descriptor. This will generate a feature description at each point, which is the basis of point matching between the source and the target during registration. This process mimics a human-in-the-loop behaviour where an observer would identify well-distinguishable points and match similar ones together to finally align the images. These unique points are called local features since they only relate to a set of neighbouring points.



(a) Filtered source PCD with Estimated Normal Vectors

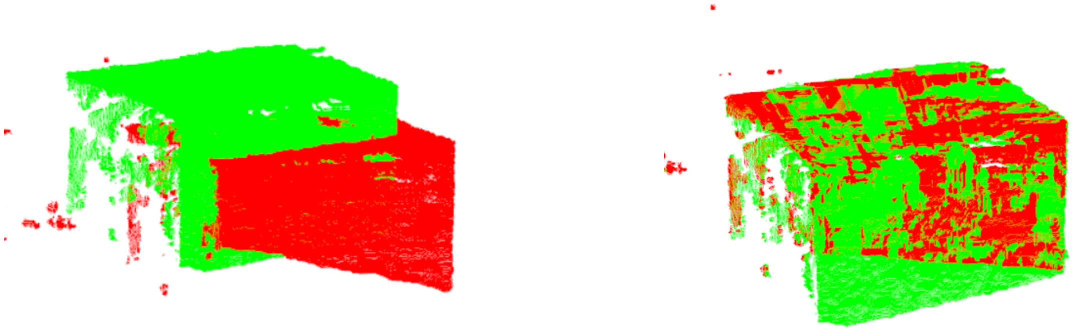
(b) Filtered target PCD with estimated normal vectors

Figure 7.16: The box in the test scene after normal estimation. The estimated normal vectors are depicted by the thin white lines and are located at previously identified key points.

In order to describe these local features with sufficient detail for the matching, that is, the local features are unique, a histogram-based descriptor is suggested by [46]. The source highlights the deficiencies of the method where local features are stored as a scalar value for the interest point, computed from the neighborhood versus the method where all points in the neighborhood are binned into a Point Feature Histogram (PFH) to represent scale and pose invariant multi-value features for local patches. This is a rather demanding process to compute, therefore [37] addressed this issue by proposing Fast Point Feature Histogram (FPFH). This is the method which is finally implemented in the pipeline.

Several sources[40, 44, 46] have recommended conducting registration in two steps. A coarse initial estimate followed by a coarse-to-fine final transformation. After obtaining feature descriptions, the coarse part can now be conducted. It uses a prerejective RANSAC algorithm[47]. It works by taking $3 \leq n$ sample points and finding their nearest neighbor correspondences in the higher dimensional space represented by the feature description. A transformation is estimated and applied based on this correspondence. Then, based on an Euclidean distance threshold applied to the transformed source and the target, inliers are identified. This process is iterated upon until the inlier ratio satisfies a threshold criteria. Prerejection comes in the form of rejecting an iteration where the polygons formed from random samples in the source and in the target have dissimilar edge length ratios. This is an exploitation of the distance conservation property of a rigid transformation and it is much cheaper than rejecting cases after performing each iteration. The level of dissimilarity used for prerejection is set using another threshold value.

The result of the process above is a coarse result, since non-perfect threshold values are to be set to account for the uncertainties of key point and feature estimations, the primary information on which this matching is based. High threshold values might result in more failed transformations and possibly longer runtimes. Finalizing the registration process, an NDT registration step is included to provide the coarse-to-fine registration. The initial point clouds together with the final registration result for the test scene are included in Figure 7.17. With these results obtained, the algorithm needs to be integrated into the pipeline described in Figure 7.20.



(a) Initial filtered source (green) and target (red) point clouds

(b) Filtered target PCD with estimated normal vectors

Figure 7.17: The box from the test scene registered from two different point clouds. As a result of the process, the source (green) is transformed into the target (red) resulting in a more complete point cloud

Dimensions Acquisitions

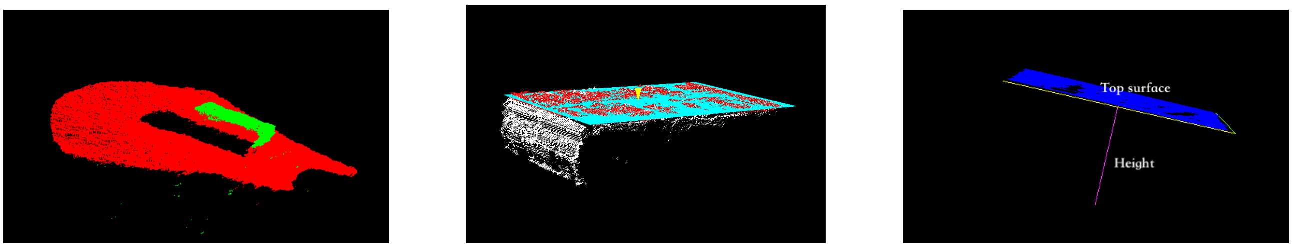
After having acquired and filtered the point cloud, the actual measurement pipeline begins. To analyze the point cloud and obtain precise measurements of the cargo item, an algorithm has been developed. Its structure is as follows:

1. The point cloud is segmented to separate the cargo item, from now on referred to as “object”, and the surrounding ground, simply called “ground”
2. The point cloud is oriented to have the ground surface parallel to the xy-plane
3. The object is further segmented to identify its **top surface**
4. The height of the cargo item is obtained from the difference between the average z-coordinates of the ground and the ones of the top surface
5. The top surface is analyzed to obtain the four vertices
6. The width and depth are obtained by calculating the distances between neighboring vertices: width is assigned to the largest measurement, and depth thus to the smallest.

The key aspects of the code are the segmentation, the vertices acquisition, and the dimensions acquisitions: these will be explained here in more detail.

- **Segmentation:** process of partitioning data into meaningful regions. Segmentation is executed twice, both using the `SACSegmentation` function of PCL [35]: once to individuate the ground and another time to obtain the top surface of the item. The ground surface is obtained using Random Sample Consensus (RANSAC): it is a method first developed by Fischler and Bolles [38] used to detect and fit a plane model to a subset of points, even in the presence of outliers. This method is applied to all the points, and it is assumed to find the largest flat surface. This is based on the assumption that the scanning targets are lying on the ground and are not touching other scanning targets. As for the top surface, this is identified using Maximum Likelihood Estimation Sample Consensus (MLESAC) which aims to find planes perpendicular to a specific axis, in this case, the vertical (z-axis) one.
  To improve the segmentation accuracy, normals are calculated before segmentation of the cloud is applied. These are obtained using the `NormalEstimation` function of the PCL.
- **Vertices acquisition:** aims at identifying key boundary points of a surface, in this case, the previously segmented top surface. The process is performed by determining the maximum and minimum points of the plane on each axis. A buffer has been included to avoid using outlier points, which could have been included in the surface segmentation. The buffer ensures that the points taken as vertices lie within the main cluster of points and are not sparsely located around the point cloud.
- **Dimensions acquisition:** the object’s measurements are obtained in two steps. Firstly the height is computed after all the segmentation and the orientation procedures are concluded; as mentioned above, it is calculated as the difference in average z-coordinates of the ground and top surface points. Finally, width and depth are acquired after the vertices have been identified. The vertices are enumerated going in a clockwise manner from the point with the highest x-coordinate, and the distances between neighboring vertices are computed using Euclidean distances.

For ease of understanding, a visualization has been included in Figure 7.18. On the left, the ground-object segmentation is shown by highlighting in red the ground points, and in green the object ones: the visual accuracy with which the box outlines are identified is quite impressive. In the middle, the segmentation of the top surface is shown, while on the right the box has been simplified in its bare components.



(a) Ground-object segmentation

(b) Top surface segmentation

(c) Cargo item components

Figure 7.18: Algorithm pipeline visualization

Numerous point clouds were acquired throughout the DSE project. These were used to assess the performance of the algorithm and fine-tune it if needed. Since many variables are present, many iterations have been performed in order to obtain accurate results in measurements. The parameters analyzed were the `DistanceThreshold` during the two segmentation processes, which represents the distance from which outliers are identified as such; and the buffer used when obtaining the vertices. Before varying these values, accuracy in measurements was 70-80%, once the iteration was concluded, the testing frames used had an accuracy of 95-99%. In order to assess whether the parameters were over-fitted for the "training data set", new frames were used to ensure the validity of the parameters: two new boxes were measured and the accuracy for the dimensions was, once again, between 95 and 99%, ensuring the soundness of the values used. This measurement precision, even though not fully compliant with **CRG-PAY-01**, is at this stage deemed reasonable. Further improvements will be performed on the software and on the pipeline at the next design stage, in order to fully comply with the requirement.

Furthermore, coarse, inaccurate and incomplete point clouds were also fed into the pipeline. The algorithm provided once again measurements in the range of 70-90% accuracy. An example of a coarse point cloud is presented in Figure 7.19: here the top surface is again blue, the box is green, and the ground red. The yellow sphere represents the identified vertices of the surface. In this frame they are identified with a quite good precision. This coarse of a point cloud is not expected to be registered during the operation, it is only an edge case, and should normally result in a measurement fail.

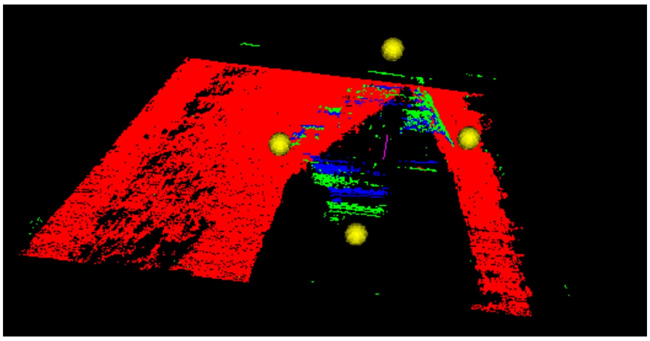


Figure 7.19: Coarse point cloud

Finally, to summarize the three aspects of the software, the volume measurement pipeline is illustrated in Figure 7.20. The three modules are highlighted, together with intermediate results. The first two run in parallel until the scene is fully registered. An initial criteria for this is set to be 4 consecutive successful registration processes. Segmentation and dimension acquisition only come after this to provide the final output. Width and length are obtained by measuring the top face, while height data is extracted by measuring the distance between the top face and the floor.

7.4. Verification & Validation

This section presents the verification and validation efforts concerning the payload subsystem. This is part of the bottom-up process in the system engineering framework where compliance with top-level requirements is verified and validated. These top-level requirements in this case flow down to the core functions of the system as a whole, such as cargo item volume measurement.

7.4.1. Object detection Verification & validation

There have been no developmental tests conducted on the object detection and label reading components of the pipeline given that these architectures were provided by widely renown and trusted third party Ultralytics [15] that have already verified and validated their setup. The only verification that was conducted was to ensure that the results were saved accordingly, and inference on random validation batches as demonstrated in Figure 7.7 and Figure 7.8. However, the algorithms were tested in real-time to assess their suitability for applications where the inference time was the parameter to measure the suitability. Using an HP HD Camera with a frame rate of 60 Hz [48], the setup was tested for 5 minutes for each frame, and on average a 23 ms inference time has been recorded, which is suitable for real-time applications. Table 7.1 shows the result of the verification and validation methods that were followed, and the compliance status of the requirements.

[15] Ultralytics https://www.ultralytics.com/. Accessed 16 June 2024.

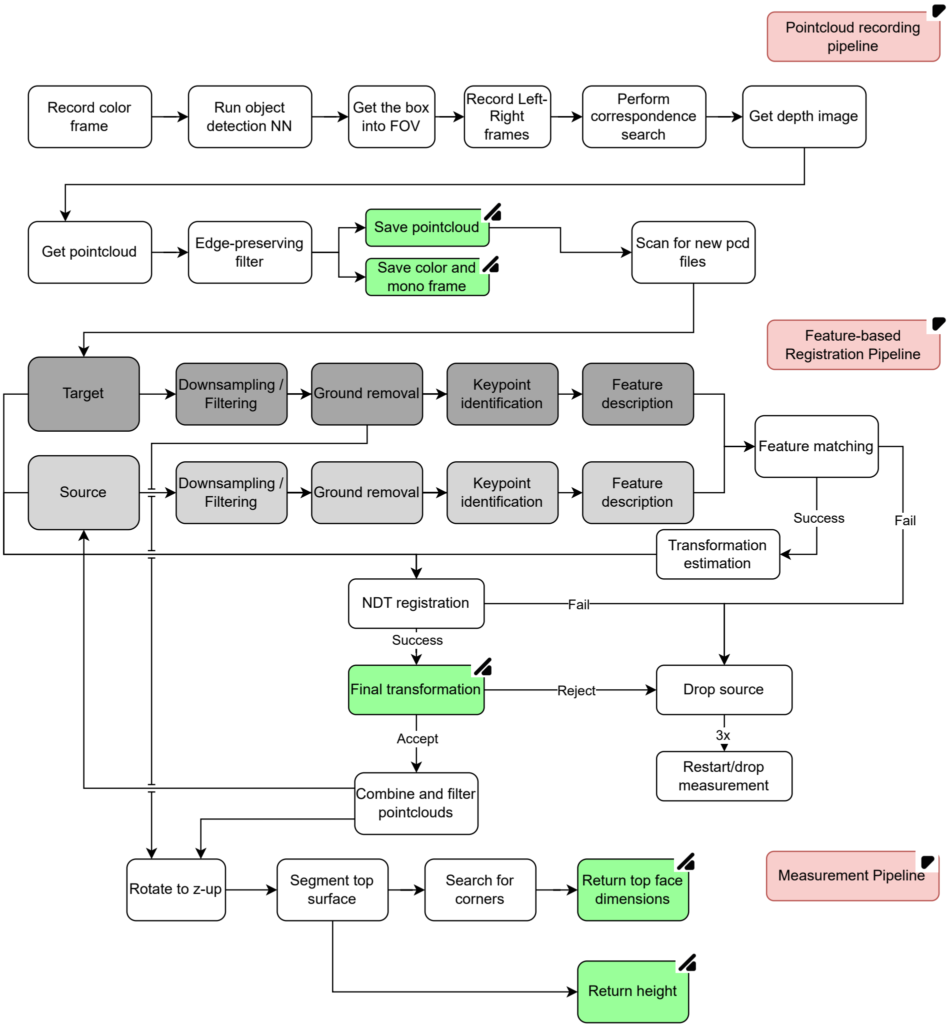


Figure 7.20: Volume measurement pipeline. The three main stages (data acquisition, data handling, and dimension acquisition) are visible. Main intermediate results/milestones are highlighted in green. Those serve as an input/output between the main stages. The pairwise nature of data handling is highlighted in the respective main stage by the darker colors indicating the two point clouds handled at the same time.

7.4.2. Volume Measurement Verification and Validation

To verify the feasibility of the volume measurement method and pipeline, it was implemented end-to-end and tested with cardboard boxes. Ground truth was obtained using a tape measure. For the camera, a used Luxonis OAK-D was obtained through individual contributions. The hardware specifications can be found in Table 7.4 [16]. Some performance differences are expected since the OAK-D is a different camera, with no projector. The rest of the pipeline was implemented as specified in Section 7.3.4. The measurements with the camera were relatively insensitive to the light level in the room, as the camera includes variable exposure parameters like sensor reading time and gain. The image quality was acceptable even when triangulating the surfaces of boxes with luminance levels as low as $10\,\mathrm{cd\,m^{-1}}$ and as high as $150\,\mathrm{cd\,m^{-1}}$. This is very low luminance level[17]. The small camera of the sensor on the other hand has very low dynamic range and dim shadows or direct sunlight can introduce holes and inaccuracies in the depth image. The camera also struggles with featureless boxes, which are most cargo boxes, as the correspondence search fails in these regions, also introducing holes in the depth image. This is the main pitfall of the passive depth camera, which the active camera would not struggle with. This validates the design choice of the structure light camera. With these in mind under ideal conditions the total pipeline runs for less than 3 seconds per frame, and requires usually 4 frames minimum for one measurement, which validates the 30 seconds measurement time requirement.

[16] OAK-D data sheet https://docs-old.luxonis.com/projects/hardware/en/latest/pages/BW1098OAK/ Accessed 16 June 2024

[17] Lightning in Warehouses: https://teamster.org/wp-content/uploads/2018/12/illuminationwarehouse.pdf. Accessed 24 June 2024.

Table 7.4: OAK-D camera parameters. As a comparison, it also includes the parameters of the camera chosen for the actual design

Camera Option	Size WxDxH [mm]	Max data rate [Mbit/s]	Weight [kg]	Power consumption [W]	Total cost [€]
Luxonis OAK-D	110x54.5x33.0	10000	0.11	3.5	250
Intel RealSense 455	124×29×26	3262	0.055	< 3.5	250

Table 7.5: Verification for Payload subsystem

Identifier	Verification Method	Compliance
CRG-PAY-01	The payload is within specs according to the datasheet. The pipeline however is not yet reliable enough to meet the accuracy requirements (Test)	To be further improved with software
CRG-PAY-02	Registration takes roughly 1.5s per frame (10 frames in total) and size estimation takes around 1s (Test/Inspection)	✓
CRG-PAY-03	Above is satisfied, rest depends on operations (Analysis)	✓
CRG-PAY-04	Demonstration of this functionality was done to confirm the requirement.	✓
CRG-PAY-05	Demonstration of this functionality was done to confirm the requirement.	Requires improvement.
CRG-PAY-06	Pipeline has been tested, it has the ability to estimate size. It can also identify labels and boxes.	✓
CRG-PAY-07	The measurement device sends all data to the processing pipeline (Test)	✓
CRG-PAY-08	Yet to be demonstrated on the fully integrated system (Demonstration)	Pending tests on fully integrated system
CRG-PAY-09	A set of LEDs are incorporated in the design.	✓

7.5. Future Recommendations

Further refinements would make the payload pipeline more secure in identifying and measuring the dimensions of cargo items. These refinements can be done individually for constituents of the payload pipeline given their independence in calculations which would increase the overall accuracy.

The speed of the process can also be improved by minimizing function calls, implementing more efficient memory-handling practices, and further customizing the pipeline. Upon experimenting with the system in practice, the number of captures needed could be reduced or the drone flight speed could potentially be increased for a faster process. Furthermore, the current assumptions used to develop the payload subsystem are somewhat limiting. The hardware is not dependent on the shape prior of the targets (rectangular box) and the system is potentially capable of providing a more detailed and general input for the optimization of the packing of the cargo items.

For object detection and label reading, using more energy-efficient models like spiking neural networks [49] would be a better suit when these models get advanced enough for real-time object classification. At present, these still have very low technology readiness levels, and any associated hardware is several orders of magnitude more expensive than the currently designed hardware. Any improvements would also tie in with the sustainable development policy, which would be a fine step to make the product more sustainable for each iteration of the design.

The volume measurement pipeline contains numerous steps that are sensitive to some parameters. These parameters could be tuned better to improve the registration and other algorithms for either a more robust result or faster runtime. Perhaps either the feature-based or NDT registration could be removed given a robust implementation of the other. Refinements of the standard algorithms included in the PCL could also be considered. Learning-based and other registration methods not analyzed here could also be investigated and perhaps implemented.

[17] A Guide to YOLOv8 in 2024https://viso.ai/deep-learning/yolov8-guide/. Accessed 12 June 2024.

8: Navigation

This chapter presents the navigation subsystem of the drone. Section 8.1 shows the goals and functions of the subsystem, followed by a brief explanation of requirements and risks in Section 8.2. The design process is then detailed in Section 8.3: here the navigation, mapping, and collision avoidance are presented. Finally, the verification and validation process and the future recommendations are shown in Section 8.4 and Section 8.5 respectively.

8.1. Goals and Functions

Navigation is necessary to ensure the correct execution of tasks within the warehouse environment. To ensure this design, a goal has been set for the subsystem to trace back all the design choices it has made to ensure the navigation of the drone. This goal is as follows

Ensure safe and efficient task execution within the warehouse by integrating collision avoidance and localization capabilities and providing navigation protocols simultaneously.

To reach this goal, a functional analysis has been conducted for the subsystem. These functions, enumerated below, reflect all necessary functions the subsystem has to provide for the drone to achieve the goal in mind.

1. Provide the precise global location of the drone with SLAM sensors.
2. Integrate collision avoidance throughout every operation phase.
3. Register local vicinity location data.
4. Provide the precise global location of static unknowns with SLAM sensors.

8.2. Requirements and Risks

Requirements were postulated based on the functional analysis; compliance with them ensures the correct behavior of the drone system regarding its navigation functions. The accuracy requirements were established based on research conducted on self-driving cars [], due to the high level of similarity in the safety-compliance of the system. Their formulation is presented in Table 8.1: here the requirements are described and a risk, stemming from non-compliance, is associated with each of them.

Table 8.1: Requirements for Navigation subsystem

Identifier	System Requirements	Associated Risk ID
CRG-SYS-NAV-01	The navigation system of the drones shall be adjustable to different warehouse configurations.	RSK-NAV-01
CRG-SYS-NAV-02	The navigation system shall be able to determine the distance between a drone and a human in the area of operations with 99% during normal operation.	RSK-NAV-02
CRG-SYS-NAV-03	The navigation system shall be able to determine the distance between a drone and a static object in the area of operations with 99% accuracy during normal operation.	RSK-NAV-03
CRG-SYS-NAV-04	Each drone's navigation system shall be able to determine the distance to neighbouring drones with 97% accuracy during normal operations.	RSK-NAV-04
CRG-SYS-NAV-05	The navigation system shall be capable of real-time localization and mapping to ensure accurate and up-to-date positioning and environmental understanding.	RSK-NAV-05
CRG-SYS-NAV-06	The drone shall be able to detect objects approaching the drone with a speed less than 10m/s.	RSK-NAV-06
CRG-SYS-NAV-07	The drone shall be able to plan an avoidance trajectory that moves the drone off from the collision-course.	RSK-NAV-07
CRG-SYS-NAV-08	The avoidance trajectory shall be implementable in real time with a maximum delay of 200ms.	RSK-NAV-08
CRG-SYS-NAV-09	The navigation system shall have a minimum range of 150 cm under ambient light.	RSK-NAV-09
CRG-SYS-NAV-10	The navigation system shall have a minimum range of 300 cm in dark lighting condition.	RSK-NAV-10

The risks are evaluated based on likelihood and severity, as shown in Table 8.2, and a mitigation plan has been implemented in an attempt to ensure that the flow of operations remains unhindered.

8.3. Design Process

Navigating inside a warehouse environment poses significant challenges due to the complex layout, dynamic conditions, and obstacles. The presence of moving forklifts, workers, and other machinery introduces unpredictable elements that necessitate real-time adaptation. Ensuring safety and efficiency in such an environment requires advanced sensors, robust mapping, and localization algorithms, and seamless integration with the Operations and the Control subsystems to efficiently plan trajectories and avoid collisions.
Furthermore, an efficient navigation strategy minimizes unnecessary movement, reducing environmental impact and enhancing overall operational sustainability.

With this in mind, the design process has been approached by focusing on navigation, and collision avoidance. The algorithm for navigation and mapping is presented in Section 8.3.1, whereas Section 8.3.2 presents the collision avoidance process.

Table 8.2: Risks for Navigation subsystem

Identifier	Payload Risk	Likelihood	Severity	Mitigation
RSK-NAV-01	The navigation system is warehouse-specific	1	4	Test the navigation algorithms and sensors for a wide range of warehouse configurations.
RSK-NAV-02	The navigation system cannot determine the distance between drone and human	3	5	Test the sensing algorithm for a wide variety of cases and in different environmental conditions.
RSK-NAV-03	The navigation system cannot determine the distance between drone and static object	2	4	Test the sensing algorithm for a wide variety of cases and in different environmental conditions.
RSK-NAV-04	The navigation system cannot determine the distance between drones	2	5	Test the sensing algorithm and the localization accuracy for a wide variety of cases and in different environmental conditions.
RSK-NAV-05	The navigation system fails at localizing and mapping such that it is unable to provide up-to-date positioning and understanding	1	5	Ensure the correct behaviour of all the navigation sensors in all possible operation situations.
RSK-NAV-06	The drone fails to detect approaching objects	2	5	Ensure the correct behaviour of all the navigation sensors in all possible operation situations
RSK-NAV-07	The drone fails to plan an avoidance trajectory	4	5	Test the trajectory planning and collision avoidance algorithms extensively for a wide variety of cases.
RSK-NAV-08	The avoidance trajectory is not implemented within the required time	2	4	Test the trajectory planning and collision avoidance algorithms extensively for a wide variety of cases.
RSK-NAV-09	The navigation system has a range of less than 150 cm under ambient light	1	5	Perform routine maintenance and calibration on the navigation sensors.
RSK-NAV-10	The navigation system has a range of less than 300 cm in dark lighting conditions	1	5	Perform routine maintenance and calibration on the navigation sensors.

8.3.1. Simultaneous Localization and Mapping

Localization of the drone and mapping of the environment, done with the same sensors, is often referred to as Simultaneous Localization And Mapping (SLAM). Additionally, the SLAM algorithm has to account for the presence of dynamic obstacles and objects, as the drones are operating in the presence of people, forklifts, and other machinery.

Numerous SLAM algorithms are currently available, each with its strengths and weaknesses. A detailed analysis was conducted, looking for an algorithm that was:

- **Computationally efficient:** the drone payload and communication structure offer limited computational power, making efficiency paramount.
- **Accurate:** the drone's navigation has to be accurate to ensure the correct identification of items and their correct measurement.
- **Able to deal with a dynamic environment:** the drone operates in the presence of people and machines, which could behave unexpectedly.
- **Robust:** the drone moves fast and can obtain high rotation rates to ensure pointing accuracy for its payload.
- **Easy to implement:** availability of resources is paramount in ensuring the implementation of the algorithm on the drone.

Based on the aforementioned criteria, it was decided to choose FAST-LIO 2 [50] as the LiDAR-inertial odometry framework. The algorithm is extremely computationally efficient due to the use of an incremental k-d tree structure, it is very accurate and robust, it can deal with dynamic environments as it builds dense point maps in real-time, and is available on GitHub.

No further development on the SLAM algorithm will be performed, as it will be directly used as provided by the source. Research was however conducted on the effectiveness of the payload sensors for navigation use, and it was found that an additional sensor is needed to ensure full field-of-view coverage for depth sensing. The VL53L9 sensor[1] was chosen to fulfill this task due to its robustness, small size, and minimal mass.

To enhance the correct localization, the warehouse will be equipped with secondary localization aids, acting as navigation reference points. Since their position in the warehouse is known, they can be used by the drones to adjust their location and re-calibrate their position within the global map. Examples of these include QR codes [51], AprilTags[2], and general visual landmarks.

[1] Navigation sensor VL53L9: https://eu.mouser.com/new/stmicroelectronics/stmicroelectronics-vl53lp-tof-modules/?_gl=1*1uoktib*_gcl_aw*R0NMLjE3MTgwMDU1MTUuQ2owS0NRanc5dnF5QmhDS0FSSXNBSUljTE1Fb0szQmZwVXIyN3otcWp5N2p2cHVTejgtTlFsWDZ3cVJTSmw0cGJYNm9tOWZ2TUxZY0ZTSWFBbGZlRUFMd193Y0I.*_gcl_au*MTI4MTQyODQ2Mi4xNzE4MDA1NTEw*_ga*MTA3MDYyMzA1OS4xNzE4MDA1NTEw*_ga_15W4STQT4T*MTcxODU1OTk3NS4yLjAuMTcxODU1OTk3Ni41OS4wLjA. Accessed 16 June 2024.

[2] What are AprilTags?: https://docs.wpilib.org/en/stable/docs/software/vision-processing/apriltag/apriltag-intro.html. Accessed 17 June 2024.

8.3.2. Collision Avoidance

Collision avoidance is a critical component for autonomous tasks, ensuring no harm to the environment due to poorly planned trajectories. This ensures that the drone avoids static and dynamic knowns and unknowns. There are three possible ways that drones avoid collisions during their operations, which are as follows:

1. Using time of flight sensors (SLAM sensors) to input information to both the global and the local planner in an attempt to register and avoid dynamic and static unknowns such as people and cargo items, but also static knowns such as warehouse environment. Furthermore, a possible reassuring measure could be the use of Radio Frequency Identification (RFID) tags[3], to be wore by the workers, in order to make their position known to the system.
2. Ensuring communication between the drones assures that no in-the-vicinity drone-to-drone collision will take place, this is representative of avoiding dynamic knowns,
3. Continuous pinging of the drones allows them to localize each other from afar, which reinforces dynamic known collision avoidance.

This is a continuous action that is used in operations while trajectories are planned, but also with controls when the planned trajectory is traversed. Given the dynamic environment, the proximate environment must be accurately sensed for both subsystems to update their operations in real time.

8.4. Verification & Validation

Since the navigation software has been fully obtained from existing material and is yet to be fully integrated into the drone system, verification and validation have not been performed at this design stage. A plan is here presented for completeness: it includes the tests that will be conducted to ensure compliance with each requirement. Table 8.3 presents the type of verification test to be executed for each requirement at the next step of Cargonaut's design. Furthermore, Table 8.4 shows a summary of the types of tests to be conducted for the validation process of the subsystem.

Table 8.3: Summary of verification methods for Navigation subsystem

Software tests	Hardware tests
• Test the individual functions of the algorithms to check for eventual errors. • Test the individual algorithms with a variety of cases, trajectories, and environments. • Test the integration of the algorithms with the rest of the payload and operations software.	• Inspect the sensors to check the positioning and integration. • Analyze and inspect the field of view of the sensors to ensure the correct coverage.

Table 8.4: Summary of validation methods for Navigation subsystem

Model-in-the-loop	Hardware-in-the-loop
Test the topology measurement and label identification software in a simulation environment such as AirSim/Gazebo. Test the navigation software in a simulation environment.	Test the functionality of topology measurement and label identification in real life either on an off-the-shelf drone or an alternative system.

8.5. Future Recommendations

The next steps in the navigation subsystem design will entail a comprehensive process of implementation and integration. This will involve not only the merging of the individual algorithms with each other but also their cohesive integration with the rest of the essential subsystems. These include critical subsystems such as Control, Operations & Logistics, and Payload.

From a concept perspective, further developments in the exact SLAM configuration will be conducted. Since the warehouse is a well-structured environment, a navigation system entailing full localization methods and less extensive mapping capabilities might fit the purpose of our system and reduce the costs and resources associated with running the SLAM algorithm at all times. However, such a solution would need to be evaluated on a warehouse basis and tested extensively before implementation due to the reduced reliability it offers.

Furthermore, extensive testing will be performed to ensure that the subsystem meets all specified requirements. This testing phase will be crucial for the validation of the subsystem, moving beyond theoretical calculations to practical, real-world application, as the performance of the navigation algorithms can currently only be demonstrated as theoretically viable for Cargonaut's specific operation scenarios.

[3] What are RFID Tags?: https://www.camcode.com/blog/what-are-rfid-tags/. Accessed 24 June 2024.

9: Propulsion

This chapter presents the development of the propulsive subsystem. After establishing the subsystem goals and follow-up functions in Section 9.1, the requirements and associated risks are listed in Section 9.2. Section 9.3 entails the design process, including component selection and assessment via an online tool. Verification of both product features and requirements is executed in Section 9.4, followed by validation, part of which can be further traced to Chapter 10. Then, concluding remarks and future recommendations are presented in Section 9.5.

9.1. Goals and Functions

Cargonaut prides itself on offering a flexible alternative to pre-existing static solutions. However, this functionality would not be possible, unless the quadcopter included a carefully designed propulsion system. Thus, acknowledging this need translates into the following goal:

Support all necessary movement, ensure precise and safe navigation in a dynamic setting and allow for timely response to environmental changes.

A number of functions can be directly identified from this setting. Closely linked with subsequent analysis from Chapter 10, it is marked that partial overlap can be observed.

1. Generate sufficient thrust to propel the drone;
2. Provide an efficient solution that minimizes power consumption;
3. Enable precise and smooth movement;
4. Incorporate safety mechanisms for environmental shielding;
5. Opt for minimal noise generation.

Naturally, these are only the main functions that the propulsive subsystem shall perform. Looking into more detail, they shall be reassessed in a more tangible way that allows the design team to track the process more easily. This is done by formulating constraints on what the system shall do, in the form of a list of requirements.

9.2. Requirements and Risks

Table 9.1 enlists the requirements, associated with the propulsive unit of each UAV. The maximum speed is dictated by EU regulations for Class 2 Speed Limit, which states that operations within 5 m distance from people shall have a low-speed mode with a maximum speed of 3 m/s[1].

Table 9.1: Requirements for Propulsion subsystem

Identifier	Propulsion Requirement	Associated Risk ID
CRG-PROP-02	Each drone shall be able to translate horizontally with a maximum speed of 3 m/s	RSK-PROP-02-01, RSK-PROP-02-02
CRG-PROP-03	Each drone shall be able to translate vertically with a maximum speed of 3 m/s.	RSK-PROP-03
CRG-PROP-04	Each drone shall translate horizontally with a minimum speed of 0.3 m/s to allow for desired scanning rate.	RSK-PROP-04
CRG-PROP-06	The propulsion system shall enable the drone to hover upon command.	RSK-PROP-06
CRG-PROP-07	The propulsion system shall be able to provide a T/W ratio of at least 2:1 during operations.	RSK-PROP-07
CRG-PROP-08	The propulsion system shall adhere to size limitations of 50 cm for maximum dimension.	RSK-PROP-08
CRG-PROP-09	The rotors shall have a physical barrier to protect the environment in case of collision.	RSK-PROP-09
CRG-PROP-10	The propulsion system shall adhere to noise regulations for the set environment.	RSK-PROP-10

Compliance would mean that the movement of the drone allows for the task to be executed nominally, and does not override any design principles. Nonetheless, overseeing a requirement could lead to potential mission failure. Thus, Table 9.2 was devised to assess this probability and what suitable actions follow. The mitigation plans must be carried out to reduce possible threats, especially concerning the critical risks, presented with bold.

9.3. Design Process

The first steps in the design process concern determining all the components, and conceptualizing how each of them works, along with the flow of information in between. Initially, for preliminary design, the Power and Propulsion subsystems were considered together. Due to the significant number of interfaces, as identified in Table 5.1, four main components were identified: propellers, motors, batteries, and an ESC. With the design maturity increasing, a further division between the two subsystems was made. As a result, in this chapter, only the rotors and motors are described in detail. Information on the power source and its transmission are provided in Chapter 12: Electronics. To convey the design approach better, the initial conceptualization is given in Figure 9.1, where key characteristics are emphasized by bold lettering. The ESC has no parameters in bold since it is usually coupled with the flight controller and is highly versatile, making its design not a first order priority.

[1] Maximum Speed Limit for Each Drone Class: `https://eudroneport.com/blog/maximum-speed-drone/`. Accessed 17 June 2024.

Table 9.2: Risks for Propulsion subsystem

Identifier	Propulsion Risk	Likelihood	Severity	Mitigation Plan
RSK-PROP-01	high-speed horizontal translation leads to lag, bad controllability	2	3	limit operational velocity based on controlling capacity
RSK-PROP-02	high-speed horizontal translation challenges warehouse operations	1	4	limit operational velocity based on safety concerns
RSK-PROP-03	drone cannot reach target altitude	2	4	analyze and test rigorously capabilities at various throttle settings
RSK-PROP-04	noisy/oscillatory behaviour of drone hinders payload operations	2	4	provide sufficient clearance between instruments, analysis of simulated environment
RSK-PROP-06	unable to hold hover position	1	5	extensive testing of hover capabilities and stability
RSK-PROP-07	not enough force to execute maneuver	1	4	determine limiting cases, design with redundancy
RSK-PROP-08	rotor tip exceeds design boundary	1	3	not needed; design and manufacturing done in compliance with size constraints
RSK-PROP-09	colliding with fast spinning rotor damages drone and environment	3	5	design propeller guards, limit rotational velocity range
RSK-PROP-10	emitted noise and vibrations cause harm to environment and workers	2	3	analyze noise emissions, ensure compliance with regulations

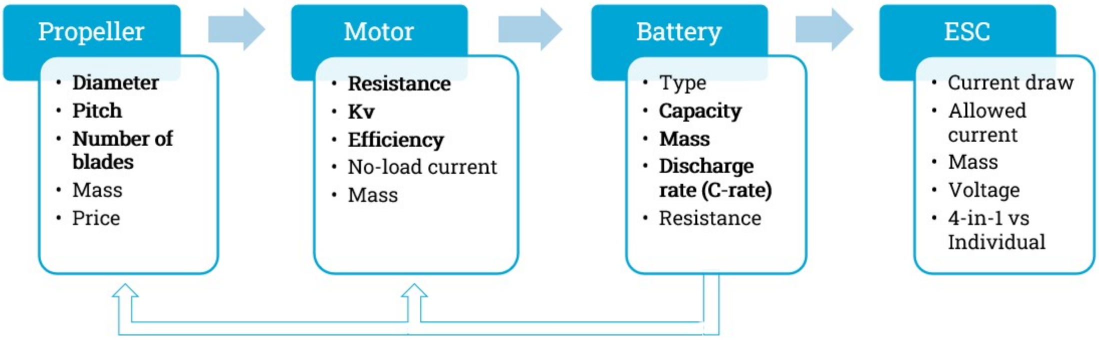


Figure 9.1: Relevant components and important parameters, driving ones marked in bold

9.3.1. Propeller Analysis

After establishing a quadrotor configuration, the team focused on narrowing down the choice of propellers. It was decided that due to the low price, wide range of available options, ease of accessibility, and convenience offered by market suppliers, a COTS component would suffice.

For a propeller, the main characteristic of size is the length; a larger propeller has a higher T/W ratio, making it more energy efficient; however, it also makes the overall product bigger and reduces the flexibility of the drone. Referring to **CRG-PROP-08**, a spatial constraint limits the design space. Ideally, the approach would be to aim for a maximum propeller size, while adhering to size limitations, taking into consideration necessary propeller spacing. At this stage, consultation with the Structures department led to iterations and a conclusion that 7″ propellers would present the most suitable option, given the viable frame topologies.

A direct follow-up was to choose the rotor pitch. Its value represents the displacement that the propeller would make if it were to complete a full revolution in a solid surface. A lower pitch translates to less air resistance, thus faster spinning and a lower current draw. Conversely, a higher value would require more power, but generate more thrust in return[2]. Essentially, the final choice is a balance between desired capabilities. Observations based on market analysis indicate that reasonable values range from 2″ to 5″. Since preserving power is of higher priority, a lower value was chosen, with the first conceptualized configuration being that for 7x3″ propellers. At the given moment, it was decided that if the analysis indicated inadequate performance, the choice would be adjusted accordingly.

Lastly, the number of blades was considered. Typical options either include two or three, and each offers different benefits. While the three-blade rotor produces less noise and is capable of reaching higher thrust values, it is more turbulent and harder to balance. Due to the high efficiency of dual-blade propellers, they typically deliver longer flight times. Thus, deciding on two blades as a first-order approach completes the initial propeller configuration.

To make sure the analysis is as accurate as possible, the preliminary component choice fell on a product by APC Propellers[3]. The manufacturer is well-renowned in the industry, providing trustworthy data on technical

[2] How to choose motor and propeller for quadcopter: https://www.droneassemble.com/how-to-choose-motor-and-propeller-for-quadcopter/. Accessed 16 June 2024.

[3] APC Propellers - Quality Propellers that are Competition Proven, 7x3:https://www.apcprop.com/product/7x3/. Accessed 16 June 2024.

performance, and most importantly, specific information on blade sections. This would allow for modeling the geometry and exporting it for analysis, this is documented in detail in Chapter 10: Flight Performance. The visual aid in Figure 9.2 depicts the generated propeller. Note that due to the lack of information provided in close proximity to the center point, extrapolation is performed near the hub.

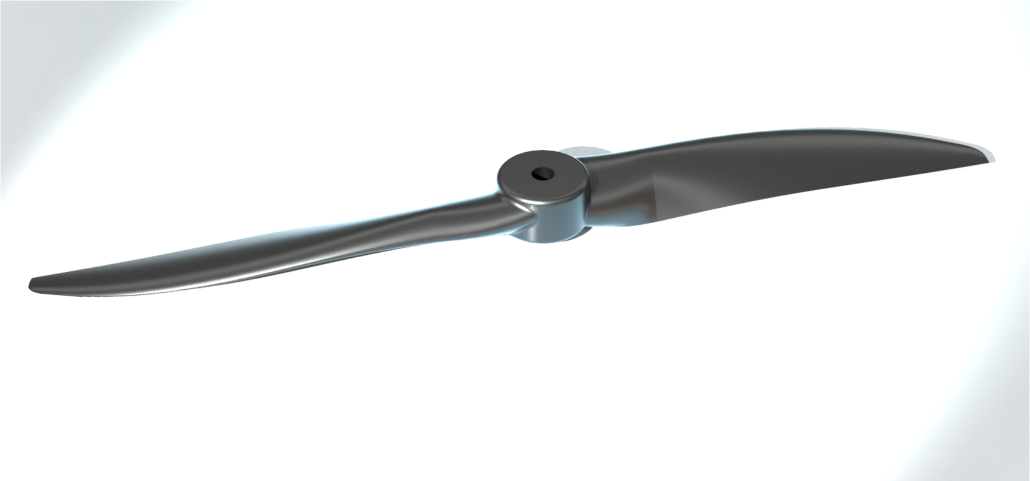


Figure 9.2: Selected propeller, modelled in 3D Experience

9.3.2. Motor Analysis

After a concept for a suitable propeller has been created, it can be paired with a given motor. The selection starts with a decision on whether it shall be brushed - have its coils rotate around a static permanent magnet, or brushless, with the magnet rotating around the coils instead. Undoubtedly, the choice falls upon a brushless motor, due to its many advantages - higher lifetime and efficiency, as well as lower noise[4] Next to that, it is important to realize what the naming convention for a given model represents. Typically, the identifier consists of 4 digits, indicative of the motor's size: the first two correspond to stator width, and the second two - to its height. A higher value for either of them leads to an increase in stator "volume", making it capable of delivering higher torque, albeit at the expense of a higher current draw[5]. In addition, a crucial parameter is the K_v rating. Representing the motor's speed constant, it is defined as the number of Revolutions per minute (RPM) the motor will turn per volt of applied voltage under no load conditions. It dictates the speed at which the propellers will rotate. In the design procedure, it is crucial to reach a balance in terms of RPM and torque. Spinning the propellers faster might generate more thrust, but require additional power input and produce more heat. Therefore, it is important to determine a suitable match for the selected propeller, under the design constraint to maximize flight time. However, the configuration would include too many variables, and a simple market search and compatibility analysis would not provide sufficient insight. As a result, the team adapted an approach based on the website eCalc and its xcopterCalc[6], specifically designed for multicopter design.

9.3.3. Configuration assessment via eCalc

Selecting eCalc as a calculation tool was based on its popularity in the field of UAV design, and implied accuracy [52, 53]. Based on a selected number of inputs, split into five categories - General, Battery Cell, Controller, Motor, and Propeller - it provides information on expected performance. Among the outputs are expected load, hover flight time, power and temperature, reached T/W ratio, and specific thrust. At this stage, refinement of the design space was carried out, based on battery limitations. The full process can be seen in Chapter 12: Electronics, and concluded with the selection of a battery with $8{,}400$ mAh capacity. Designing for endurance, iterations were then performed for different configurations, only varying the choice of motor. The database of eCalc was explored, opting for both low internal resistance and low no-load current. Eventually, the study was finalized, with the following selection:

Propeller type: APC Propellers, 7x3[7] **Motor type:** Mamba Toka 2207.5 - 1700KV[8]

[4] Brushless Vs Brushed DC Motors: When and Why to Choose One Over the Other: https://www.monolithicpower.com/en/learning/resources/brushless-vs-brushed-dc-motors. Accessed 16 June 2024.

[5] How to Choose FPV Drone Motors – Considerations and Best Motor Recommendations: https://oscarliang.com/motors/. Accessed 16 June 2024.

[6] xcopterCalc - Multicopter Calculator: https://www.ecalc.ch/xcoptercalc.php. Accessed 16 June 2024.

[7] APC Propellers - Quality Propellers that are Competition Proven, 7x3:https://www.apcprop.com/product/7x3/. Accessed 16 June 2024.

[8] Diatone Mamba Toka 2207.5 1700KV/1800KV/2450KV/2650KV Motor: https://www.getfpv.com/diatone-mamba-toka-2207-5-1700kv-1800kv-2450kv-2650kv-motor.html. Accessed 16 June 2024.

Together with the predetermined power components, this yields the following numbers for metrics of interest:

- Flight time (hover): $30.0\,\text{min}$, at $\omega = 10{,}000\,\text{RPM}$
- T/W ratio: 3.8
- Power required: $199\,\text{W}$

At first sight, it appears that no requirement has been violated, and the appointed components would perform accordingly. Nonetheless, while the desired minimum T/W ratio is easily achieved, it appears that no margin is present in terms of endurance. Therefore, to check compliance and gain a better grasp of the calculations involved, the expected behavior was modeled and then compared with outcomes from eCalc.

9.4. Verification and Validation

The process of verification starts with a focus on the behavior of the motors, and Equations 7.1 to 7.6:

$$I_{\text{stall}} = \frac{\varepsilon - R_{\text{battery}} I_{\text{payload}}}{4R_{\text{battery}} + R_{\text{ESC}} + R_{\text{motor}}} \qquad n_{\text{max,0}} = K_{\text{v}}\varepsilon \tag{7.1, 7.2}$$

$$V_{\text{m}} = \varepsilon - (4R_{\text{battery}} + R_{\text{ESC}})I \qquad P_{\text{m}} = V_{\text{m}}I - R_{\text{motor}}I^2 - \left(\frac{I}{V}\right)_0 V_{\text{m}}^2\, th_{\%} \tag{7.3, 7.4}$$

$$th_{\%} = \frac{n}{n_{\text{max,motor}}(I)} \qquad n = 60\sqrt[3]{\frac{P_{\text{m}}}{C_{\text{p,0}}\rho D^5}} \tag{7.5, 7.6}$$

Equation 7.1, which calculates the stall current, I_{stall}, is obtained by balancing the current of the circuit, assuming components are connected in parallel with the battery, with R denoting the resistance. Then, **Equation 7.2** gives the maximum rotational speed at zero load, $n_{\text{max,0}}$. Assuming a motor has a linear rotational speed curve, interpolating the points given by **Equations 7.1 and 7.2** allows calculating the maximum RPM curve. In addition, **Equation 7.3** describes the voltage of the motor, V_{m}, as a function of current. **Equation 7.4** provides an estimate for the mechanical power of the motor, P_{m}. It accounts for the electrical power of the motor ($P_{\text{e}} = V_{\text{m}}I$) and expected losses[9]. Corrections are done based on the motor voltage V_{m}^2 and throttle percentage $th_{\%}$. This throttle percentage is given by **Equation 7.5** as the ratio between the actual RPM and the maximum RPM for a given current. Finally, **Equation 7.6** models the rotational speed of a propeller in RPM[10].

In order to use these equations to validate the information given by eCalc, the computational flow shown in Figure 9.3 was devised. Most steps of the calculation were already described and are marked by their formulas; the only novel calculation worth mentioning is that the maximum current, I_{max}, is found by intersecting curves $n_{\text{motor}}(I)$ and $n_{\text{prop}}(I)$. It shall be noted that the three parameters to verify are marked in a darker color in the figure.

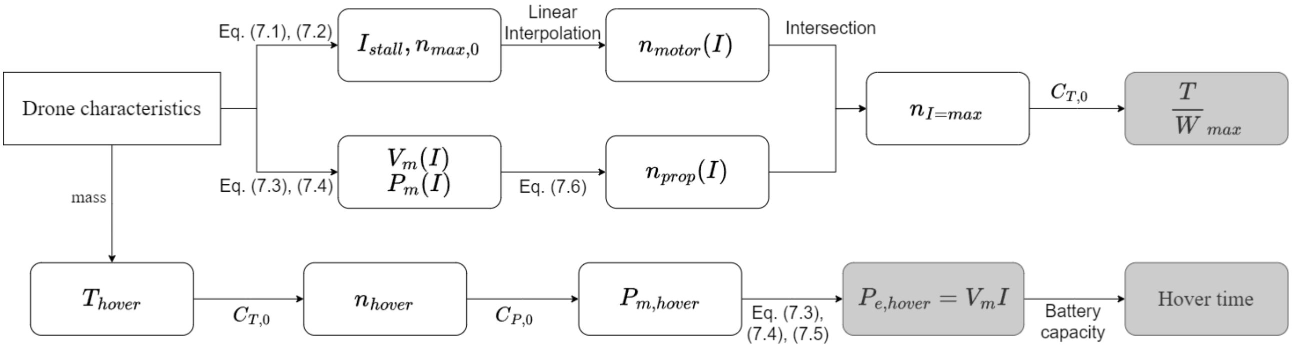


Figure 9.3: Flow of calculations for the verification of eCalc data

In Table 9.3, it can be seen that the performance values found manually differ in the order of 10-20% compared to the eCalc values. However, this does not disregard the design configuration, since the analysis indicates that eCalc provides an overall conservative estimation. It underestimates both the hover time and T/W ratio and overestimates the power consumption.

[9] Bavaria Direct - Motor Constants: https://www.bavaria-direct.co.za/constants/. Accessed 16 June 2024.
[10] UIUC Propeller Data Site: https://m-selig.ae.illinois.edu/props/propDB.html. Accessed 16 June 2024.

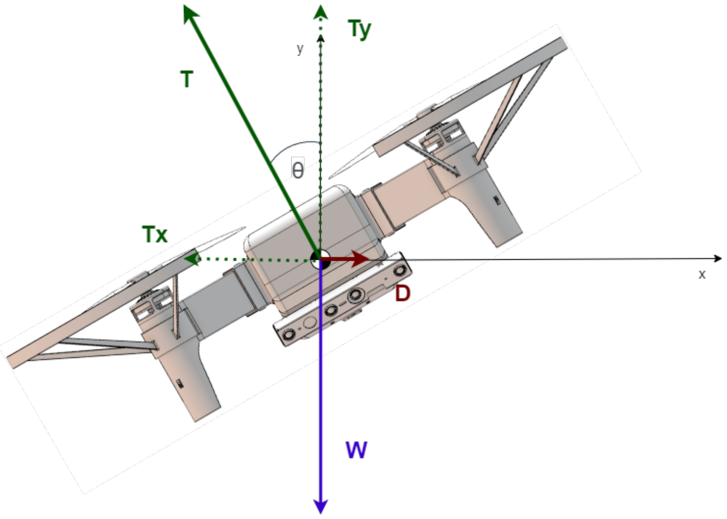
Table 9.3: Verification via comparison of own calculated values and outputs from eCalc for relevant performance parameters

Hover Time [min]		Thrust-to-Weight ratio (T/W) Ratio		Power Consumption [W]	
eCalc	**Verification**	**eCalc**	**Verification**	**eCalc**	**Verification**
30.0	35.7	3.8	4.5	199	167

Furthermore, eCalc does not provide explicit formulas, which renders unit testing much harder. However, it does provide intermediate values, like current and power for hover and maximum throttle conditions, which can be checked against manual calculations. Following the step-by-step approach in Figure 9.3, it is established that for the hovering conditions, the motor calculations mostly remain within a 1-2% range of the eCalc. However, for a given thrust at hover, the calculated RPM value is slightly higher, and apart from lower mechanical power, electrical calculations are consistent. This promotes confidence in the accuracy of modeled equations and indicates successful unit testing. However, a suspicion arose that eCalc sources its static performance data plots ($C_{\mathrm{p},0}$ and $C_{\mathrm{t},0}$) differently. This might be due to an extra contingency added by the calculator, or the fact that it sources data from a less efficient propeller. For calculations, modeling was based on data from University of Illinois Urbana-Champaign[11], which provides analysis on rotors by the manufacturer of interest, APC Propellers.
Conclusive remarks on eCalc verification yield confidence in eCalc as a used tool. It underestimates performance, but within a 15% confidence interval, which does not pose a threat of system over-design, but can be utilized as a safety margin.

Next, verification of requirements considering the velocity range was addressed. This was done by simplifying the forces, acting on the quadcopter, and presenting a model in two dimensions, in a Free-body Diagram (FBD). The diagram can be seen in Figure 9.4, with supporting **Equations 7.7 - 7.10**. Since the horizontal translation is limiting, and because the thrust needs to overcome both drag and weight in two orthogonal directions, only a tilt configuration is analyzed. Compliance with it would be in accordance with requirements regarding vertical movement.

To assess the maximum capabilities, the full-throttle setting will be analyzed. For $22{,}000\,$RPM, static tests by APC Propellers indicate that each propeller is capable of delivering $14.36\,$N of thrust. The weight that it shall counteract is taken from the mass budget in Table 5.2. The mass value of $1.434\,$kg, with 20% contingency margin, leads to a weight $W \approx 14.1\,$N. Then, the excess thrust can be used to model the bank angle. This is the case when the drone reaches vertical equilibrium at a given angle, with the vertical component of the thrust T_y equalling the weight W. Then, comparing the horizontal component T_x with an estimated drag value D provides insight into the horizontal behavior.



$$T = \sqrt{T_x^2 + T_y^2} \tag{7.7}$$

$$T_x = -T\sin\theta \tag{7.8}$$

$$T_y = T\cos\theta \tag{7.9}$$

$$D = \frac{1}{2}C_D\rho V^2 S \tag{7.10}$$

Figure 9.4: FBD of drone in $x - y$ coordinate system, for a given bank angle θ

Firstly, it is assumed that four rotors are cumulatively capable of producing $57.44\,$N of thrust, acting through the centre of gravity of the body. If the vertical component of thrust equals the weight, the maximum bank angle that can be achieved is $\theta = 14.21°$. Decomposing the thrust force then translates into a T_x of $55.68\,$N. The last step is to consider the modeling of the drag force. An empirically obtained formula, based on quadcopter mass, was established from [54], which yields a value for the drag coefficient $C_D = 0.23$, whereas analysis in [55] concludes with $C_D = 0.32$. To be conservative, the team proceeded with a value of 0.5 and approximated the surface area S to be $0.25\,\mathrm{m}^2$, based on a full square with sides being of the maximum allowable dimension. For nominal sea level

[11] UIUC Propeller Data Site: https://m-selig.ae.illinois.edu/props/propDB.html. Accessed 17 June 2024.

conditions, equaling the horizontal component of thrust and the drag leads to Equation 7.11 for computation of the maximum velocity:

$$V = \sqrt{\frac{2T_x}{\rho S C_D}} \approx 27\,\mathrm{m/s} \tag{7.11}$$

Despite seemingly high, this velocity shall never be reached due to safety concerns and will be limited by the department of Control and Simulation. This was done to comply with the warehouse in-flight speed regulations, and this is ensured by the control and simulation department, as mentioned in Section 14.3.1. In addition, in normal flight, “the rotor drag is about an order of magnitude lower than the rotor thrust”[56], thus promoting confidence in the final solution.

In terms of validation, two branches can be distinguished. The first considers a close-up look into the reliability of the data, included in eCalc. For example, selecting a pre-existing motor automatically provided exact values regarding its current draw, resistance, as well as weight. The accuracy of the information was checked via online searches for technical data. Specifications of the selected TOKA motor were compared with information from an online reseller. Values were mostly identical, with deviations never exceeding 3%[12]. On the other hand, validation of the actual drone’s capabilities constitutes of more in-depth analysis. It was established that available analytical solutions might poorly reflect the dynamic loading scenarios, especially in terms of propeller interference, tip vortices, or wake behavior. Therefore, a model setup was devised and processed via CFD. Due to its complexity and involvement, it was decided that it would be documented separately, and can be observed in the subsequent chapter 10: Flight Performance.

Requirement compliance

With Verification and Validation (VnV) concluded for the given stage, requirements shall be revisited. With design procedures being finalized, and risk mitigations taken into consideration, it is critical to readdress Table 9.1. Accountability for selected requirements is shown via a compliance matrix, in Table 9.4. At the time of submission of this report, **CRG-PROP-10** is not verified, since noise estimations have not yielded conclusive results yet. Therefore, compliance is marked as To be confirmed (TBC).

Table 9.4: Verification for Propulsion subsystem

Identifier	Verification Method	Compliance
CRG-PROP-02	Analysis, to be confirmed with tests	✓
CRG-PROP-03	Analysis, to be confirmed with tests	✓
CRG-PROP-04	Analysis, following from maximum speed requirement; minimum speed is not restricted by propulsive design	✓
CRG-PROP-06	Analysis, shows thrust equals weight for $\approx$ 25% throttle setting, enabling hover	✓
CRG-PROP-07	Analysis, tests of selected propeller, given by manufacturer, confirm a T/W value of 4:1 can be reached with selected propeller	✓
CRG-PROP-08	Inspection, design with accordance to Structures subsystem confirms no boundary is exceeded	✓
CRG-PROP-09	Inspection, propeller guards designed and compliance confirmed by visual inspection of overall assembly	✓
CRG-PROP-10	Not applicable: noise calculations are still to be performed	TBC

9.5. Future Recommendations

Concluding the design stage for the propulsive subsystem yields satisfactory results, with selected COTS components that meet postulated requirements. To extend the analysis beyond the limiting cases, the behavior of the UAV shall be additionally modeled in Chapter 10. This would allow for the finalization of the drone’s performance in terms of aerodynamics. However, at this stage, no acoustic modeling has been carried out. In fact, noise calculations are crucial, especially for operations in an enclosed setting. Firstly, EU regulations demand that the emitted sound remains below a certain threshold. Secondly, from a sustainability perspective, it is crucial to avoid noise pollution and contribute to an environment that is pleasurable to work in. As a result, it is a first-order priority for future work to entail analysis of the drone’s noise levels.

[12] 1/2/4PCS NEW MAMBA TOKA MB2207.5 2207.5 1700/2450KV 4-6S Motor 5inch-6inch Props High Speed for RC FPV Racing Drone Quadcopter: https://www.aliexpress.us/item/3256801321498673.html?gatewayAdapt=glo2usa4itemAdapt Accessed 17 June 2024.

10: Flight Performance

In this chapter flight performance analysis is presented. Firstly, the goals and functions are given in Section 10.1. This is followed by a consideration of the requirements and risks in Section 10.2. Section 10.3 describes the design process of the flight performance analysis. The analyses performed for a single propeller and multiple propellers are discussed in Section 10.4, and Section 10.5 respectively. Requirement compliance is verified in Section 10.6. Lastly, in Section 10.7 future recommendations are presented.

10.1. Goals and Functions

Unlike conventional subsystems, Flight Performance (FP) does not necessarily involve making new design choices or selecting components. In fact, it evaluates the selected configuration's ability to execute the assigned tasks successfully and proposes better solutions. Overall, the main goal of FP can be postulated as follows:

To analyze Cargonaut's flight modes for optimization of aerodynamic efficiency, safety, and assessment of operational capabilities.

Meeting this goal is governed by a number of functions that shall be performed. Most of them are linked to the performance of the propellers, as they will enable the drone's movement and maneuvering.

1. Enable drone movement in all 6 directions: forward, backward, upward, downward, left, and right.
2. Analyze the thrust-to-weight ratio for efficient takeoff and landing.
3. Prove the drone's stability in chosen flight conditions
4. Confirm the drone's ability to execute given maneuvers and hold a stable position in hover.
5. Establish the generated forces and power consumption of the propellers.
6. Provide sufficient spacing such that rotor-to-rotor, rotor-to-frame and drone-to-drone interactions do not hinder performance.

A successful design process would ensure that all these functionalities are accommodated, and the provided solution meets the set goal with certainty. To allow for traceability and accountability, the functions are used as a foundation for formulating requirements; the associated risks that these requirements would not be met are evaluated based on expected likelihood and severity.

10.2. Requirements and Risks

Requirements were postulated in a SMART (specific, measurable, attainable, realistic, timely) manner; compliance with them would ensure no functionality loss is present. Their formulation is presented in Table 10.1 and linked to risks in Table 10.2.

Table 10.1: Requirements for Flight Performance subsystem

Identifier	Flight Performance Requirement	Associated Risk ID
CRG-FPA-01	During operations, all loads shall be maintained within the flight envelope.	RSK-FPA-01
CRG-FPA-02	Each drone shall be able to withstand the wake from other drones.	RSK-FPA-02
CRG-FPA-03	The drone shall be able to withstand design gust loads.	RSK-FPA-03
CRG-FPA-04	The drone shall be able to withstand design maneuvering loads.	RSK-FPA-04
CRG-FPA-05	The drone shall be able to withstand perturbations in the form of wake from other machinery in the warehouse.	RSK-FPA-05
CRG-FPA-06	The drone propellers shall have a minimum spacing of 1/4 of the propeller diameter.	RSK-FPA-06

Strategies to prevent the occurrence of these risks are devised in the form of mitigation plans, which must be executed throughout the design stage. In addition, critical risks are marked with bold and should be carefully considered.

Table 10.2: Risks for Flight Performance subsystem

Identifier	Flight Performance Risk	Likelihood	Severity	Mitigation plan
RSK-FPA-01	Unforeseen loads reached, damaging the structure and hindering controllability	2	4	Implement safety factors, be conservative in design
RSK-FPA-02	Drones coming in close proximity critically destabilize each other	2	4	Drones operate at different height (flight levels), collision avoidance algorithms
RSK-FPA-03	Drone becomes inoperative due to gust loads	1	3	Not needed; still, typical gust loads will be modeled
RSK-FPA-04	Drone becomes unstable in a given maneuver	3	3	Model typical maneuvers and analyze controllability; change throttle setting gradually
RSK-FPA-05	Other machinery severely disturbs drone operations	1	4	Not needed; highly improbable
RSK-FPA-06	Positioning propellers too close causes interference and lowers performance	1	4	Not needed; requirement met by design, check via visual inspection

10.3. Design Process

In terms of aerodynamic performance, the component of the highest interest is the propeller. The sizing procedure and final choice were introduced in Chapter 9. In summary, four 7x3″ rotors propel the drone, obtained from APC Propellers[1]. The manufacturer provides performance data, obtained with their own software and based on Vortex theory. Results encompass blade loading for different settings, but only for static configurations. However, given that flexibility is one of Cargonaut's main values, analysis in dynamic settings is crucial for ensuring that the product behaves as expected. To simulate behavior and get insight into maneuvering, a decision was made to attempt to model the propeller with suitable software.

10.3.1. Theoretical background

Initially, literature study was conducted to assess the possible approaches, ranging from fully analytical to numerical and computationally intensive methods. Basic building blocks start with Actuation Disk Theory (ADT). Also known as Momentum theory, ADT's foundation lies in modeling the rotor as an infinitely thin disc with a constant rotational velocity. As a result, no propeller geometry is incorporated, but its overall effect on the flow can be analyzed. For a more detailed mathematical background and derivation, as well as governing assumptions, the reader is advised to refer to the book "General Momentum Theory for Horizontal Axis Wind Turbines" [57]. Whereas this theory can be used for preliminary estimations, it does not incorporate rotational velocities and lacks any blade geometry considerations or tip corrections. Thus, to obtain a more accurate model that takes into account the shape of the selected propeller, the rotor can be discretized, as depicted in Figure 10.1. This is the foundation for Blade Element Theory (BET), with which differential analysis is carried out on each individual section and is used for geometrical optimization. Blade Element Momentum Theory (BEM) combines both theories and essentially, considers the integrated effects across the entire propeller disk. In comparison, both Vortice Lattice Method (VLM) and Vortice Particle Method (VPM) assume inviscid, incompressible flow and model it with vorticity concentrated in sheets or particles, respectively. Panel methods rely on similar flow field simplifications, but can encapsulate viscous effects and blade geometry better [58].

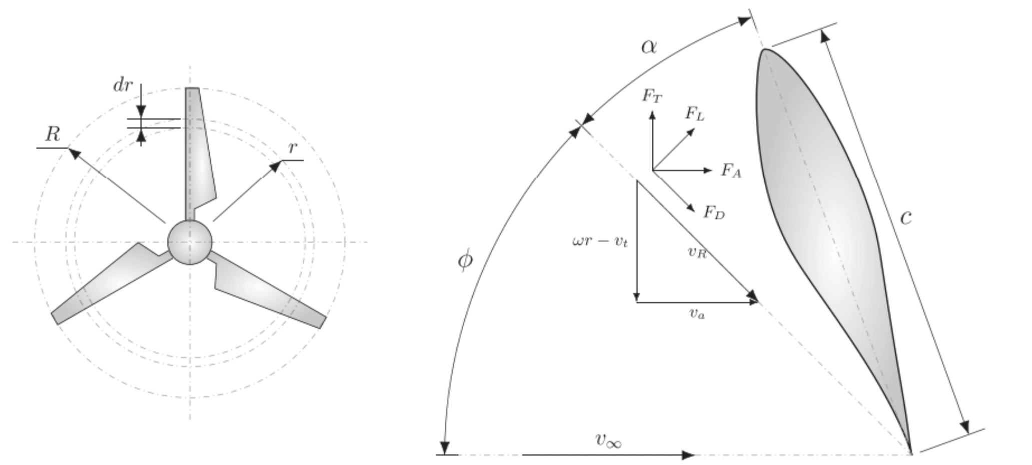


Figure 10.1: Discretization of a propeller, with a close-up view and parameter definition for a given blade section [59]

Naturally, all this theory can be incorporated with different CFD software, and at various levels of complexity. Nonetheless, given the limited resources, mostly regarding time availability, not all options would be feasible. For instance, research indicated that, whereas a full CFD solution would be most accurate, it cannot be executed within the project duration. As a result, the team decided to embark on a conservative approach and use simplified BET methods to obtain results. The foundation is based on a pre-existing class in OpenFOAM, namely `rotorDisk` [60], extended to match the case of interest. The appointed method "simulates the time-averaged flow over a propeller by adding the blade forces as source terms in the momentum equations", more accurately than usual Blade Element methods and faster than full 3D geometry CFD. This belief was solidified by a meeting with Flavio Martins, a PhD candidate at TU Delft. He met with the team, shared his expertise, and provided guidance when necessary. Martins confirmed that modeling the full blade geometry would not be suitable, since the main focus is on the total performance values of the propeller, and not the propeller design per se, as it is set in stone by the manufacturer. In addition, deciding to proceed with Capitao's tutorial [60] for adjusting the `rotorDisk` class assisted in narrowing down the choice for software to use from many commercially available programs down to a

[1] APC Propellers - Quality Propellers that are Competition Proven, 7x3:https://www.apcprop.com/product/7x3/. Accessed 12 June 2024.

single one - OpenFOAM[2]. Being open source, it offers a wide range of functionalities and is also used in academic settings.

10.3.2. Modelling approach

The first step in the analysis was to get familiarized with the propeller, based on manufacturer data: technical, performance, as well as geometry[3]. The files provided information, based on a blade, divided into 39 separate sections and profiles. They were used to generate a physical model, shown in Chapter 9. Since no information regarding the geometry in close proximity to the hub was provided, the blades were extrapolated to intersect with the central area. While this might lead to some deviations, it is not expected to strongly influence results, since these inner sections are expected to produce marginal amounts of force and moment arms. Furthermore, to assess the lift and drag characteristics of each profile, analysis was carried out with xflr5[4] to produce relevant polars.

The rotorDiskSource class - main takeaways

As previously mentioned, the general approach was adapted from Capitao's "Description and validation of the rotorDiskSource class for propeller performance estimation" [60]. In this section, some core theory will be provided, followed by an explanation of how it is implemented in OpenFOAM and its limitations. For more details, please refer to the tutorial.

Governing equations: According to Blade Element Theory (BET), the blade is discretized in 2D sections, and parameters of interest are determined for each one of them. These include blade sectional thrust dT and torque dQ for an inflow velocity of W. Geometry properties are also of interest, with the flow angle $\phi = \arctan(W_z/W_\theta)$ and blade angle β determining the angle of attack ($\alpha = \beta - \phi$). Also, for a given α and a specified chord c, tabular data from polars is gathered for lift and drag coefficients: $c_l = f(\alpha),\ \ c_d = f(\alpha)$.
Decomposing the forces in the axial and tangential direction, the blade sectional forces are described by Equation 10.1 and Equation 10.2, respectively. F denotes the tip factor, considered due to the presence of tip vortices.

$$f_z = \frac{1}{2}\rho W^2 c(Fc_l \cos\phi - c_d \sin\phi) \qquad (10.1)$$

$$f_\theta = \frac{1}{2}\rho W^2 c(Fc_l \sin\phi + c_d \cos\phi) \qquad (10.2)$$

To simulate the entire propeller, the results are integrated along the full circumference, depicted in Equations 10.3 and 10.4.

$$dT = f_z dr \rightarrow T = \int_{r_{hub}}^{r_{tip}} f_z dr \qquad (10.3)$$

$$dQ = f_\theta r dr \rightarrow Q = \int_{r_{hub}}^{r_{tip}} f_\theta r dr \qquad (10.4)$$

Nonetheless, the obtained thrust and torque are per blade section: still, they shall be translated into volume forces. For the rotor of interest, consisting of two blades, this is done via Equations 10.5 and 10.6, where the points of interest are the predefined cell centers r_{cell}. After the summation, the power P is easily obtained through multiplying the torque by the rotational velocity, via $P = Q\omega$.

$$T = \frac{1}{\pi}\sum_{k=1}^{nCells} \frac{A_{cell,k}}{r_{cell,k}} f_z(r_{cell,k}) \qquad (10.5)$$

$$Q = \frac{1}{\pi}\sum_{k=1}^{nCells} \frac{A_{cell,k}}{r_{cell,k}} f_\theta(r_{cell,k}) r_{cell,k} \qquad (10.6)$$

Implementation: The `rotorDiskSource` class consists of three files, which are called by the program if a `fvOptions` file with a `rotorDisk` type entry is found. A short description follows:

1. `rotorDiskSource.H` - this is the main class declaration file, where the high-level rotor geometry is declared, such as number of blades. Also, main functions are declared, including reading the data, constructing the coordinate systems, and blade divisions. Source terms are added, and the setup for subsequent force calculations is made.
2. `rotorDiskSource.C` - In short, this file is used to call relevant constructors and define the functions initialized in `rotorDiskSource.H`, that can read parameters from the `fvOptions` file.
3. `rotorDiskSourceTemplates.C` - For this implementation, this file is crucial, since it contains the force calculations and practically all modifications. Most of the code permits the application of the theory to the particular case of interest.

In addition, three folders, representing classes, can also be found. Two of them are of interest - `bladeModel` and `profileModel`, which allow for specifying the geometry.

[2] OpenFOAM: https://www.openfoam.com/. Accessed June 13 2024.

[3] APC Propellers - Quality Propellers that are Competition Proven - Technical Information | Downloads: `https://www.apcprop.com/technical-information/file-downloads/`. Accessed 13 June 2024.

[4] xflr5: `https://www.xflr5.tech/xflr5.htm`. Accessed 13 June 2024.

As an official OpenFOAM tutorial, used in academic settings, there is already some confidence in the model that it has a suitable degree of accuracy. In addition, the author performs some VnV and achieves similar magnitudes for thrust and power, roughly 15% lower from true values [60]. Suggestions for improvement involve varying boundary conditions and turbulence models. While the team modified the boundary conditions, to make them representative of the expected conditions during warehouse operations, further experiments with turbulence models were not carried out. This was done due to time limitations, lack of sufficient prerequisite knowledge, and introducing complexity that would only further complicate the design process. As a result, for turbulence modeling, the team opts for a solver of the Reynolds-averaged Navier–Stokes equations (RANS) equations. The default tutorial model `kOmegaSST`[5] was kept as fixed, deemed suitable in the case of the rotors because of accurate modeling of flow separation and complex turbulent flows. Elaboration upon suitable initial and boundary conditions is integrated within the explanation of the simulation setup.

Simulation parameters: The validity of the results of a simulation is heavily governed by the set-up of the physical parameters fed into it. Choosing these conditions well ensures that the simulated flow approaches physical conditions as accurately as possible. These parameters can be divided into multiple classes:

- **Turbulence models:** RANS simulations model the time-averaged flow on a larger scale and use a set of equations to model turbulence and its transport at a small scale. As mentioned before, the chosen turbulence model was `kOmegaSST`, which requires a set of initial parameters for the turbulence kinetic energy k and turbulence-specific dissipation rate ω. These can be initialized in the inlet according to Equations 10.7 and 10.8[6]:

$$k = \frac{3}{2} u'_{rms} \tag{10.7}$$

$$\omega = \frac{k^{0.5}}{C_\mu^{0.25} L} \tag{10.8}$$

  Here, u'_{rms} is the root mean squared of the perturbation velocity, L is a reference length, and $C_\mu = 0.09$ is a model constant. Exact values for u'_{rms} are not known, but assuming a low value of $(u'_{rms})^2 = 0.1\,\mathrm{m\,s^{-1}}$ considering the indoor environment of the warehouse, and the reference length equal to the diameter of the propeller ($L = 0.1778\,\mathrm{m}$), the values of $k = 0.015\,\mathrm{m^2\,s^{-2}}$ and $\omega = 1.258\,\mathrm{s^{-1}}$ are obtained. These will be applied as boundary conditions on the inlet(s).
- **Boundary conditions:** the conditions for the flow have to be specified at the boundaries of the domain and of the bodies in contact with the flow so that the solver can compute them on the rest of the domain. These should be consistent with the physical conditions of the problem, and also ensure stability of the solution.

  The five fields being calculated are velocity U and pressure p, as well as k, ω, and the turbulent kinematic viscosity ν_t for the `kOmegaSST` model. ν_t is only used as a post-processing variable and doesn't factor into the calculations, so it can be initialized as zero. For the other four variables, the boundary conditions are summarized in Table 10.3.

Table 10.3: Boundary conditions for initialization of solver

Surface	U	p	k	ω
Inlet	fixedValue Uinlet	zeroGradient	fixedValue kInlet	fixedValue omegaInlet
Outlet	pressureInletOutletVelocity	fixedValue 0	inletOutlet	inletOutlet
Surface (only used for drone)	noSlip	zeroGradient	fixedValue 1e-12	omegaWallFunction

For the inlet, the flow is assumed to be undisturbed. This assumption will hold if the boundary domain is large enough, which shall be ensured in the domain generation. Since the flow is undisturbed, the gradient of pressure is zero, and the other fields take fixed free-stream values (already calculated and fixed for k and ω, vary by the case for U).

For the outlet, the `inletOutlet` class of boundary conditions is used. These functions assign a zero-gradient boundary condition for outflow and a fixed value (in this case, the same as the inlet) for calculated inflow. This is used to ensure that, when backflow happens, the flow is consistent with freestream values. For pressure, however, a fixed value of zero is used. The boundary conditions for pressure cannot be zero gradient on the entire boundary, so they're set as zero on the outlet. Pressure is normalized in OpenFOAM so this value only sets it at an average pressure (for free flow conditions).

Surfaces, like the frame of the drone, are modeled with no-slip conditions. This entails zero velocity and zero pressure gradient at the walls. For k, a value of zero is used (in practice, a very low value of 1×10^{-12} is used to avoid floating point errors); since the velocity is zero, the kinetic energy will also be zero. Finally, for ω, a wall function is used to ensure that this quantity is properly dissipated even for a coarser mesh.

[5] SST k-omega model: https://www.cfd-online.com/Wiki/SST_k-omega_model. Accessed 14 June 2024.

[6] k-omega Shear Stress Transport (SST): https://www.openfoam.com/documentation/guides/latest/doc/guide-turbulence-ras-k-omega-sst.html. Accessed 14 June 2024.

These formulations also comply with the convention for Ansys simulations [7], providing more confidence in the model.

- **Meshing schemes:** The choice of meshing also has a great impact on the quality of the analysis, since the RANS equations are discretized for finite volume integration. A finer mesh will provide less numerical error and converge faster, but will also consume considerably more computational resources. Given limited resources and time, it is very important to tackle this trade-off between mesh refinement and computational time.

  The `snappyHexMesh` tool in OpenFOAM provides a good balance between these by constructing a castellated mesh that "snaps" to the desired geometry, therefore achieving higher refinement levels where gradients are expected to be higher (for example, next to boundaries and walls) and a lower refinement for undisturbed flows. This tool also allows for the creation of inflated boundary layers, which are necessary to capture the flow around complex geometries. Examples are in Figure 10.2 and Figure 10.3.

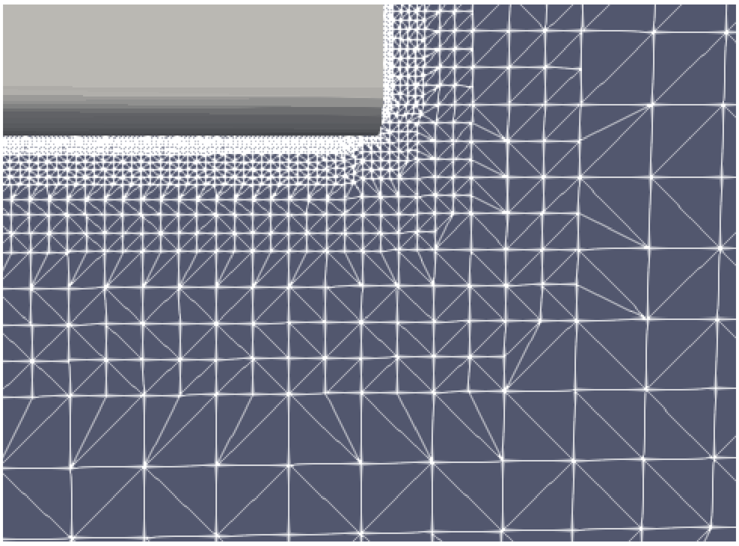


Figure 10.2: Castellated drone mesh, with different refinement levels present

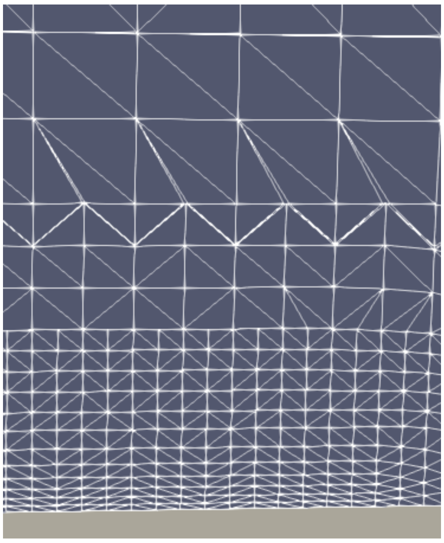


Figure 10.3: Boundary drone mesh, with inflation layers present

  Additionally, mesh evaluation tools can provide an evaluation of the quality of the mesh for complex geometries, by evaluating the volume and shape of cells and the velocity gradients that they experience. These will not be used for the rotor cases, however, since the geometry for them is quite simple.

- **Solvers:** Multiple solvers can be used to converge to a solution. Given a well-posed problem, all solvers will converge to a similar solution, albeit with different rates of convergence.

  For the rotor simulations, `simpleFoam` was used, given its simplicity and the fact that it achieved convergence fairly quickly for the analyzed cases. On the computational side, multiple different schemes were used for computing differential operators, such as Gauss linear (for Laplacian and gradient) and Gauss upwind (for divergence). These are unmodified from the original tutorial and can be found on its GitHub page [8].

10.4. Analysis of a single propeller

The test cases modeled vary in complexity, but all rely on one building block: a single propeller. Obtaining accurate results for a rotor in hover would provide confidence in the model and allow for more complex scenarios to be analyzed, such as different maneuvers and possible wake interference.

10.4.1. Propeller in hover

The first step of analyzing the rotor performance is seeing how its thrust and power behave statically, that is, with no incoming flow. For this purpose, the domain bounds and meshing need to be analyzed, which will be done in this subsection. Finally, these results will be presented, verified, and validated once a suitable domain is chosen.

Case setup: for this case, the inlet velocity conditions were set at a fixed value of zero, as to simulate a static test. As for the domain of the simulation, it was modeled in 4 areas, with different functions and mesh requirements. These are:

[7] S. Wahono, Development of Virtual Blade Model for Modelling Helicopter Rotor Downwash in OpenFOAM: https://apps.dtic.mil/sti/pdfs/ADA603409.pdf. Accessed 19 June 2024.

[8] rotorDisk fvSchemes: https://github.com/OpenFOAM/OpenFOAM-8/blob/master/tutorials/incompressible/simpleFoam/rotorDisk/system/fvSchemes. Accessed 14 June 2024.

- **The general domain:** this is where the simulation occurs. It is given as a cylinder with equal length and diameter, such that its cross-section is a square. It has a level of refinement of 1 on its inside and of 2 at its boundary.
- **The rotor:** a cylindrical disk with 0.01 m of thickness and the same diameter as the propeller (0.1778 m). The hub of the disk (the part of the propeller up to a radial distance of 0.0203 m) has been removed, as is common in the industry. This is done given the lack of polar data on the hub, which should not affect the analysis since the hub generates negligible amounts of force compared to the tip. The rotor has a level of refinement of 4.
- **The near wake:** a cylinder with radius of 0.25 m, aligned with the rotor, spanning from 0.25 m upstream to 0.5 m downstream. This zone also has a level of refinement of 4.
- **The extended wake:** a cylinder with radius of 0.3 m, also aligned with the rotor, spanning from 0.3 m upstream to the lower bound of the domain. This zone has a level of refinement of 3.

Here, the level of refinement indicates, as the name suggests, how much the local mesh is refined compared to the base one. A level of refinement of n indicates that the cells are 2^n finer; for example, for a base cell density of 8 cells per meter, the rotor zone would have $8 * 2^4 = 128$ cells per meter. Figure 10.4 illustrates how the mesh will look like, given an example domain size of 3 m and base grid of 8 cells/m.

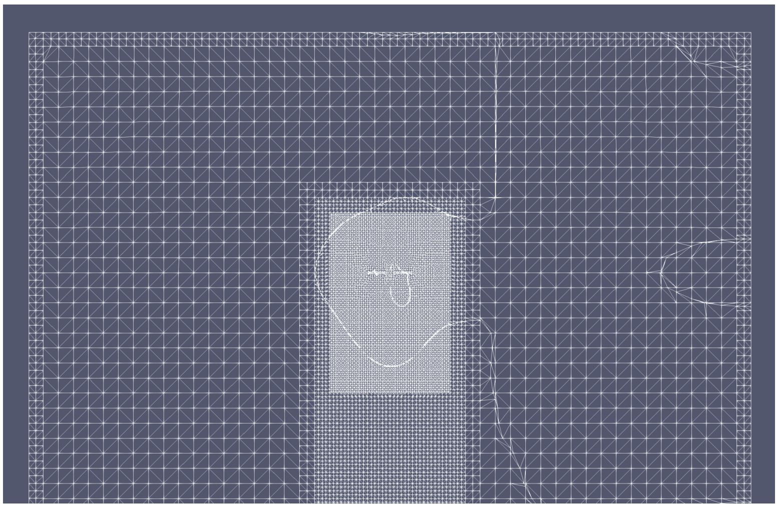


Figure 10.4: Cross-section of mesh for a rotor domain

The meshing was done this way as to assign a higher cell density where the biggest changes of velocity occur, that is, close to the rotor. This way, flow phenomena can be correctly captured on coarser base meshes, reducing the computational time without much loss of accuracy (compared to a mesh that is refined in its entire domain).

Domain sizing: Since the boundary condition for velocity for the inlet is set as zero, the inlet fundamentally acts as a non-slip wall. At first glance, to approach the 'true' static thrust of the propeller, these walls should stay as far away from the propeller as possible to not interfere with the circulation of air to the propeller; however, if the domain size is too big, it might prove too computationally expensive and not suitable for this analysis. Therefore, to analyze the effect of domain sizing, multiple simulations were run varying this parameter and keeping the mesh density constant (at 8 cells/m). Table 10.4 showcases how the runtime, convergence power, and thrust change based on domain size.

Table 10.4: Test data for multiple domain sizes

Domain diameter/length [m]	Computation time [s]	Thrust [N]	Power [W]
1	333	1.744	24.46
1.5	398	1.730	24.38
2	544	1.735	24.45
2.5	710	1.735	24.45
3	970	1.733	24.39
4	1,652	1.742	24.47
5	2,852	1.747	24.49

The domain size was limited to a range of 1 m to 5 m. Any lower value was deemed physically unsuitable, while bigger domain sizes required too many resources for this application. It can be seen in this table that, contrary to the initial considerations, the domain size does not significantly affect results. Defining the percentage of spread as the normalized difference between the maximum and minimum values, the spread of values for thrust and power are $0.99\,\%$ and $0.48\,\%$ respectively. The computational time, however, increases considerably, given the higher number of cells to be computed.

Mesh refinement: Similar to the domain size, the base mesh density can be changed and will also be analyzed. Refinement increases the number of cells and computational time, so ideally, it would be kept as low as possible, while reaching adequate levels of accuracy. A mesh that is too coarse will not capture some smaller-scale phenomena and might introduce discretization errors.

Multiple simulations were run varying the base mesh and keeping the domain size constant (at $3\,\mathrm{m}$ of diameter and length), whose results are showcased in Table 10.5.

Table 10.5: Test data for multiple base mesh densities

Base mesh [cells/m]	Computation time [s]	Thrust [N]	Power [W]
6	297	1.606	22.69
7	542	1.735	24.37
8	970	1.733	24.39
9	1,615	1.732	24.24
10	6,038	1.735	24.23

For this analysis, the base cell density ranged from $6\,\mathrm{cells/m}$ to $10\,\mathrm{cells/m}$. Lower densities did not properly snap to the rotor, while higher densities took too much time to compute. Again, it can be seen that (excluding the first test for $6\,\mathrm{cells/m}$) the spread of the values are $0.17\,\%$ and $0.64\,\%$, for the thrust and power respectively.

Final test results: Given the two previous analyses done on the test set-up parameters, it can be concluded that the domain size and base cell density do not play a big role in the behavior of the simulation. This could be due to the fact that the cells around the rotor are already refined to a high level, which is the location where the main features of the flow occur, so increasing the size of the simulation will not change the behavior considerably.

Therefore, a domain size of $3\,\mathrm{m}$ with a base mesh of $8\,\mathrm{cells/m}$ will be used as a final configuration. These are median values, which lead to accurate results, and will prospectively allow the addition of other features. The obtained thrust and power values for this case are $1.733\,\mathrm{N}$ and $24.386\,\mathrm{W}$ respectively. The resulting velocity field is also displayed in Figure 10.5.

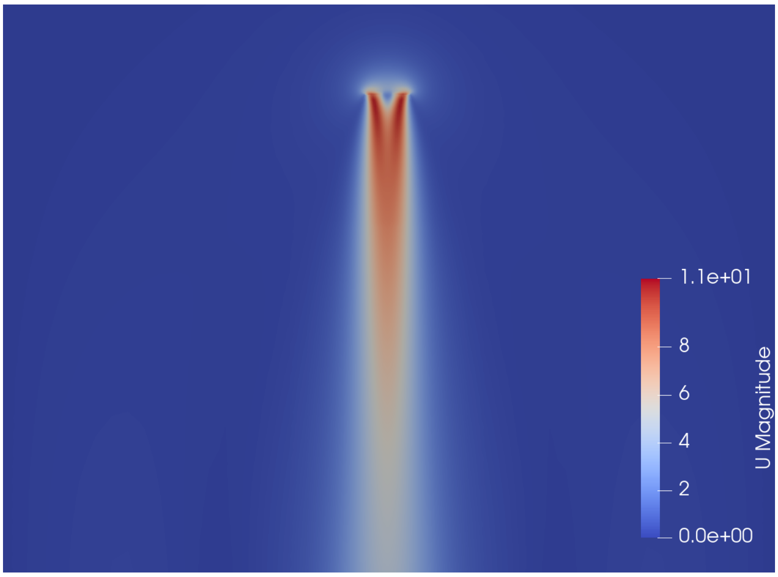


Figure 10.5: Cross-section of the velocity field for the final domain size

Verification and Validation

Model verification: After the first analysis was completed, a verification code also using BEM was implemented in Python, by calculating the forces at every blade section and integrating numerically. It operates under slightly further simplified assumptions; assuming no rotation of the flow in the blade's direction and a uniform induced velocity on the rotor. Under these simplified assumptions, the script yielded a thrust value of $2.144\,\mathrm{N}$. This value is slightly higher than the one obtained by the OpenFOAM simulation, but it lies within $23.7\,\%$ of it, which is a reasonable deviation considering the assumptions in the Python implementation are heavily simplified. This only provides more confidence that the model is indeed functioning as it was programmed to.

Validation with manufacturer data: to provide validity to the model, a comparison with test data was made. APC, the manufacturer, provides test data[9] for their propeller for different rotation velocities and advance ratios at operation. The test data itself is also computed through simulation, but it has been verified with wind-tunnel data [61] (although this verification was done for other APC propellers, and not the one currently being analyzed), and is also compliant with the thrust values provided by the eCalc calculator.

For the APC test data, at the operating conditions of $10{,}000$ rpm and an advance ratio of 0, the acquired thrust values are 2.815 N and 30.579 W respectively. This indicates that, for the computational model being analyzed, the thrust and power values are undershot by 38.4 % and 20.3 %. After consulting with Flavio, it was solidified that this margin of error is expected from simplified models such as this one; however, this implies that the values obtained by this analysis cannot be used directly to compute performance parameters for the final product. Nonetheless, this analysis tool will still be used as a comparative aid, to provide a quick analysis as to how different operating conditions change the performance of the rotors.

As an additional note, the possibility of using an optimization algorithm to scale the aerodynamic polars being fed into `rotorDiskSource` to match the validation data was considered. Even though it achieved suitable results, matching the test data much closer, it was speculated that since the polars are stripped of their physical meaning through this scaling procedure, the results of the analysis could not be extrapolated as well to other operating conditions. Therefore, this idea was discarded.

10.4.2. Propeller in vertical translation

In this subsection, the behavior of the rotor when in the presence of flow is analyzed. This comes as an extension of the previous case, where velocity was taken as zero.

Case setup: For this test case, the velocity boundary conditions on the inlet were updated to account for the incoming flow. This flow is taken to be aligned with the axis of rotation of the rotor, therefore also being parallel to the direction of the walls. The meshing of the domain and the rotor stays unaltered from the previous test case. Since the rotor is still aligned with the domain walls, the previous wake meshing still remains valid. Additionally, to be consistent with previous results, the domain from the final configuration (3 m with a base mesh of 8 cells/m) from the previous section will be maintained.

Velocity performance: the analysis was done for velocities in steps of $2.175\,\mathrm{m\,s^{-1}}$, up to $10.875\,\mathrm{m\,s^{-1}}$. These were chosen to match the velocities used for APC test data. In Table 10.6, the acquired thrust and power are displayed, alongside the static test data for completion.

Table 10.6: Test data for multiple incoming flow velocities

Vertical velocity [m/s]	Simulation		APC data	
	Thrust [N]	Power [W]	Thrust [N]	Power [W]
0	1.733	24.39	2.815	30.58
2.175	1.621	24.09	2.556	30.64
4.35	1.411	23.03	2.25	30.08
6.525	1.141	21.11	1.897	28.70
8.7	0.7970	17.93	1.502	26.36
10.875	0.3773	13.21	1.070	22.92

It can be seen that the values of thrust decrease for increasing flow velocities, which is to be expected for a rotor. As the incoming flow velocity is increased, the local angle of attack experienced by the propeller blades lowers, decreasing the blade's lift/thrust (and slightly decreasing drag/power).

Verification and Validation

Validation with manufacturer data: Table 10.6 also includes the test data provided by APC. It can be seen that, as per the static case, the values do not match up exactly; however, some conclusions can be drawn by seeing how the performance values change with the advance ratio in comparison to their static counterparts. For this purpose, the thrust and power normalized with respect to their static values are plotted in Figure 10.6.

From these figures, it can be seen that after the normalization of the performance parameters, both trust and power display quite similar behavior in both the computed and the test data. However, for the low-velocity cases, the thrust is overestimated, while the power is underestimated, considering the corresponding static values. Again, this could stem from the simplifications of the model, or be caused by the discrepancies in the aerodynamic polars, as per the previous test scenario.

[9] APC Propeller Performance Data: https://www.apcprop.com/technical-information/performance-data/. Accessed 19 June 2024.

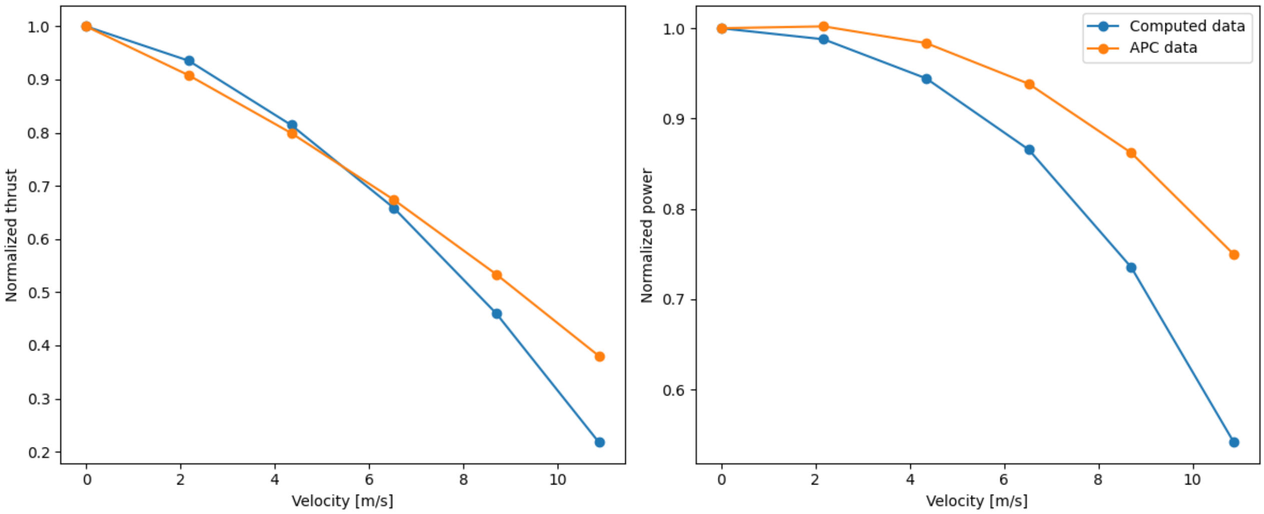


Figure 10.6: Normalized values of thrust and power plotted over velocity

10.4.3. Propeller in tilt

Another important scenario to analyze the rotors in is in the case of non-axial flow, that is, when the rotor displays an incidence angle in relation to the flow. This is additionally relevant considering that test data for non-axial flow is not available for APC propellers. Therefore, the performance of rotors in tilt will be analyzed in this section, along with extensive verification and validation with literature.

Case setup: for this test case, the flow velocity was kept at $4.35\,\mathrm{m\,s^{-1}}$ while the rotor was rotated around the X-axis for various different angles.
As for the domain, the overall dimensions were kept at $3\,\mathrm{m}$ and $8\,\mathrm{cells/m}$; however, the extended wake's radius was augmented to $0.5\,\mathrm{m}$. Since the rotor is now at an angle with the flow, its wake is expected to deviate from the direction parallel to the flow, and the grid refinement should be enlarged to capture that feature.

Tilt performance: the analysis was done for velocities in steps of $15°$, up to $90°$. These were chosen to give a full range of the directions of flow that the propeller might experience in flight. In Table 10.7, the acquired forces and moments along every axis are displayed, alongside the $0°$ case from the last section for completion. Here, the Z direction is aligned with the rotation axis of the propeller; the X direction is aligned out-of-plane, perpendicular to both the flow direction and the propeller axis of rotation; and the Y axis completes a Cartesian coordinate system.

Table 10.7: Test data for multiple tilt angles

Tilt angle [deg]	Fx [N]	Fy [N]	Fz [N]	Mx [Nm]	My [Nm]	Mz [Nm]	Power [W]
0	0	0	1.411	0	0	0.02199	23.03
15	0.001627	-0.007856	1.412	0.002922	-0.001818	0.02197	23.01
30	0.001836	-0.01795	1.428	0.004648	-0.003613	0.02176	22.78
45	0.003824	-0.02525	1.504	0.006635	-0.005202	0.02203	23.07
60	0.02246	-0.03352	1.532	0.01392	-0.006928	0.02129	22.29
75	0.004340	-0.04009	1.738	0.01226	-0.008661	0.02308	24.17
90	0.006322	-0.04294	1.868	0.01547	-0.009514	0.02338	24.49

It can be seen that, apart from some outliers, most forces increase in magnitude monotonically for increasing angles of tilt. The moment in the Z direction (and the power, since it is derived from it) stands out, however; it seems to initially decrease slightly, and then, follow an increase. Further analysis needs to be done in order to deduce if this corresponds to a calculation error or a real physical phenomenon.

Verification and Validation

Model verification: the first step in verifying the model is verifying the source of the out-of-axis forces and moments that the rotor experiences. In X. Fei's PhD dissertation [62], he quotes the source of these to be due to two phenomena: the advancing/retracting effect of the blade and the effect of the vorticity of the skewed wake.

The first is the easiest to explain. Since the blades of the propeller are not symmetrical to the flow in a tilt condition, different sections experience different flow velocities. In this case, the retracting blade, aligned with the negative X direction, rotates with the flow, therefore experiencing less velocity and less lift. On the opposite side, the advancing blade, going against the flow, rotates against the flow, experiencing higher velocity and lift force.

This uneven distribution of force should cause a moment in the negative Y direction (following the right-hand rule, since there's an excess of force in the positive X section), which is indeed observed.

The second phenomenon is caused by the tip vortices of the wake. In this case, since the wake is not aligned with the axis of the rotor anymore and follows the direction of the flow, the tip vortices from the upstream direction of the disk are positioned closer to the rotor than the downstream vortices. Tip vortices are oriented from the bottom to the top of the blade where they originated from, so these upstream vortices will inflict a moment in the positive X direction, exactly as observed with the simulation data. This phenomenon is shown in figure Figure 10.7.

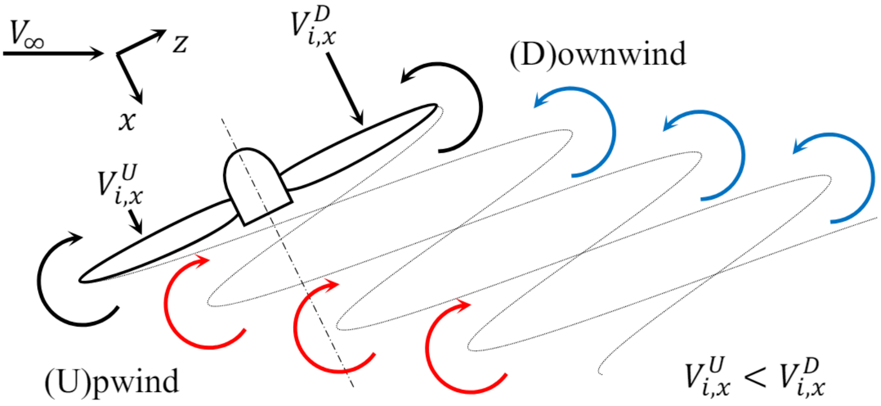


Figure 10.7: Tip vortex interaction in a rotor[62]

10.5. Analysis of multiple propellers

After the dynamics of a single propeller were analyzed, the next step taken was to assess rotor-to-rotor interactions. This would allow for a better idea of the expected behavior of propellers when they come nearby, as would be the case for parallel propellers in a quadcopter, or the influence of a rotor wake in another rotor.

10.5.1. Two propellers side-by-side

Since a quadcopter has four propellers acting nearby, it is important to assess how the distance between these will affect their thrust and power consumption. If too close interaction proves to be fatal to the performance of the drone, and the propellers need to be spaced further away at a later stage of the design, the size requirement might not be fulfilled.

Case setup: for this test case, two flow conditions were analyzed; a static test, and an incoming flow of $4.35\,\mathrm{m\,s^{-1}}$. For each flow scenario, the rotor spacing was varied.

For the domain, the outer dimensions were kept again at $3\,\mathrm{m}$ and $8\,\mathrm{cells/m}$, and the near wake (now one for each propeller) was also kept at the same dimensions. The extended wake radius was increased so it would encompass both of the propellers; ranging from $0.45\,\mathrm{m}$ to $0.375\,\mathrm{m}$ depending on the inter-rotor distance.

Performance data: for each of the two flow conditions, three rotor distances were tested: $1D$, $0.5D$, and $0.25D$, with $D = 0.1778\,\mathrm{m}$ being the diameter of one rotor. The results of the analysis are presented in Table 10.8 below, with the performance of an isolated rotor also added for comparison.

Table 10.8: Test data for multiple inter-rotor densities

Vertical velocity [m/s]	Propeller distance	Thrust [N]	Power [W]
0	Isolated	1.733	24.39
	1D	1.736	24.30
	0.5D	1.737	24.41
	0.25D	1.731	24.30
4.35	Isolated	1.411	23.03
	1D	1.392	22.82
	0.5D	1.392	22.90
	0.25D	1.389	22.81

The variation in these results is very low. For the static case, the spread of thrust and power are only $0.35\,\%$ and $0.46\,\%$ respectively. Both values seem to increase slightly until a distance of $0.5D$, and then drop slightly. The trend for the $4.35\,\mathrm{m\,s^{-1}}$ case is clearer: both thrust and power decrease as the inter-rotor distance gets smaller, with a spread of $1.62\,\%$ and $0.94\,\%$ respectively. Again, further verification will be done to discover the nature of these changes.

Verification and Validation

Validation with literature: the conclusions reached in the previous section agree with the literature. [63] analyzes the interference of three rotors in parallel, with a tip clearance of only $4\,\%$ of radius, and reports a performance loss of $1.5\,\%$ for zero angle of attack, and $3\,\%$ for an angle of attack of $5°$. [64] tested two rotors in parallel, at a distance of $0.05D$, and also reported "negligible effects" of this interaction, with no significant drop of aerodynamic parameters.

10.5.2. Interference for a propeller in the wake

Modeling the behavior of a rotor in the wake of another is also of great importance. This could happen when a drone is flying close to another, either on top or in the horizontal vicinity, and their wakes interact. Assuring the performance of the rotor, in this case, will assure stability for drone-to-drone interference, as is required by **CRG-FPA-02**.

Case setup: for this test case, the same two flow conditions as the previous test case were analyzed. For each of these conditions, there are two possible rotor arrangements: in both of them, the downstream rotor is placed one meter behind the upstream one, but in one case, both rotors are axially aligned, while on the other, the downstream rotor deviates $0.5D$ from the normal. This second case intends to simulate the effect of having an uneven velocity distribution, which might cause a detrimental moment. The diameter and the near wake were kept the same, with the extended wake radius extended to $0.35\,\mathrm{m}$.

Performance data: The results of the analysis are presented in Table 10.9 below, with the performance of an isolated rotor also added for comparison.

Table 10.9: Test data for multiple vertical wake configurations

		Upstream propeller		Downstream	
Flow Speed [m/s]	**Alignment**	**Thrust [N]**	**Power [W]**	**Thrust [N]**	**Power [W]**
0	Isolated	1.733	24.39	-	-
	Aligned	1.735	24.39	1.313	22.07
	Off-centre	1.741	24.41	1.482	23.05
4.35	Isolated	1.411	23.03	-	-
	Aligned	1.391	22.90	0.9719	19.43
	Off-centre	1.391	22.90	1.141	20.86

Verification and Validation

Verification with other test cases: the results of this analysis match with the behavior observed for other test cases. The rotor in the wake suffers from a loss of thrust since the upstream rotor speeds up the flow past it, effectively making the other rotor operate at an advanced ratio. In fact, for static aligned case, the wake velocity at the downstream rotor tends to $5\,\mathrm{m\,s^{-1}}$ to $6\,\mathrm{m\,s^{-1}}$, and calculating the isolated thrust for a propeller for this incoming speed also yields results on the order of $1.2\,\mathrm{N}$ to $1.3\,\mathrm{N}$.

Another observation to be made is that the effect of the wake is quite limited. When the downstream rotor is off-center by just half a diameter, the experienced thrust loss is almost halved. This is a reasonable result, since even just by visual evaluation, it can be seen that the wake does not expand to a considerable size with length. Therefore, it can be speculated that performance will not be greatly affected by drone-to-drone interference, since the rotors need to be quite aligned for a strong effect to be felt. It is important to note that once the propellers get integrated into the frame, the flow might alter in direction and shape, but these conclusions stand as a good starting point for the full analysis.

10.6. Requirement compliance

After the analysis of the previous test cases, a fuller picture of the behaviour of the drone can be captured. Therefore, an update on the requirement compliance, along with a short description for each, is provided in Table 10.10.

Table 10.10: Verification of Flight Performance subsystem

Identifier	Verification Method	Compliance
CRG-FPA-01	Analysis, the maximum forces experienced during flight were communicated to Structures to ensure the chassis is designed to withstand these loads.	✓
CRG-FPA-02	Analysis, the analysis of propellers in a wake indicates that the drones should be able to withstand being near each other. Further simulation and testing can be made.	✓
CRG-FPA-03	Analysis, the drone's T/W was verified to provide high velocities and forces in Chapter 9.	✓
CRG-FPA-04	Analysis, the drone's T/W was verified to provide high velocities and forces in Chapter 9.	✓
CRG-FPA-05	Not definitive yet, can be verified by further testing of the CFD model and real-life trial flights.	TBC
CRG-FPA-06	Inspection, accounted for in the structural design of the drone.	✓

Additionally, for the computation of the drone's hover time requirement, a more accurate prediction of the thrust and drag on the rotors when integrated within the drone shall be made. As a final test case, the full drone geometry, including the propeller guards and rotor disks was meshed and ran for a static case, as represented in Figure 10.8. This analysis yielded a thrust and power of 1.458 N and 22.82 W respectively, corresponding to 93.56 % and 84.13 % of the isolated static values. Given the validation of the model performed so far, these percentages will be considered a good representative figure and will be extrapolated to real propeller and drone data.

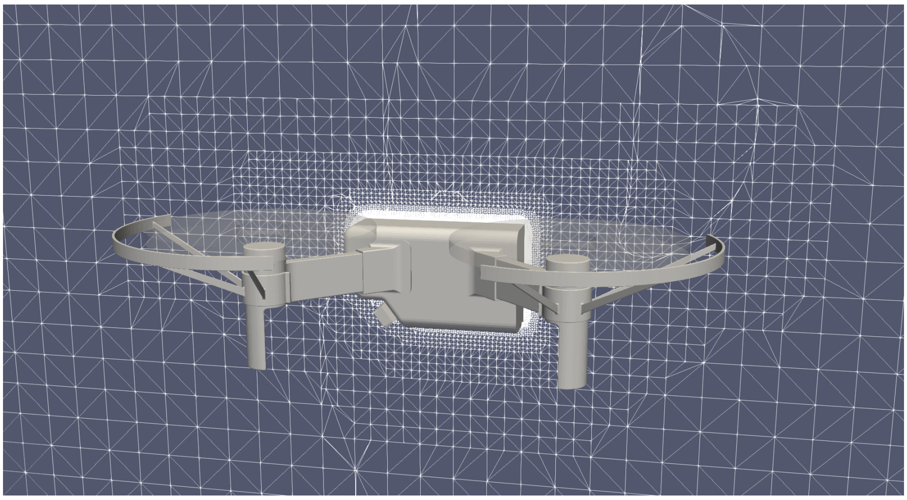


Figure 10.8: Heading direction cross-section of the mesh for the full drone

Considering the current estimate for the total mass of the drone of 1.337 kg, obtained from communication with the Structures department after updates have been made on the physical configuration, the mechanical power consumption of each rotor in the drone comes to 45.348 W. This was calculated by finding the required thrust by force balance, checking the required power for said thrust with APC test data, and applying the aforementioned percentage adjustments. It can be seen that the current value is bigger than the one calculated for the midterm, but that is to be expected given the increase in mass. In Chapter 12, the effects of the power consumption of the rotors on the runtime of the drone will be analyzed.

10.7. Future Recommendations

Before future suggestions are shared, it is crucial to reflect on the working process and posed limitations. Naturally, the simplified models used do not manage to fully encapsulate the drone's behavior. However, while we argue that the generated results fall within a reasonable confidence interval of 40 %, it is of utmost importance to point out the limitations that were encountered. Firstly, work was done on the basis of a steady flow, with a model not fully resolved on a blade scale. BET disregards any velocity distribution in the radial direction, which fails to fully describe the local sweep for a propeller. Whereas the manufacturer provided some profile definitions, the geometry close to the hub was not specified and simply extrapolated. Altogether, these factors introduce some inaccuracies, overall marginal. However, if they are considered all together, amplifications can occur and lead to higher discrepancies than initially expected. Nonetheless, acknowledging these limitations promotes confidence in the work that is carried out. This is due to the fact that deviations from expected results can be attributed to foreseen imperfections. Given the same order of magnitude in expected and obtained results and acknowledging the simplifications made, the team gained confidence that no modeling error was present. It is believed that resources are utilized, and for the provided setup, the best possible results are gathered.

With this in mind, for future iterations, it is desirable to channel efforts toward performing analysis on the full drone. Already, some preliminary efforts have been deployed in that direction, with a provisional mesh depicted in Figure 10.8. In addition, it is advisable to focus on refining the current model: tests could be carried out for unsteady flows, or blade-resolved problems. Also, boundary layer effects could be considered, along with attempts at incorporating tip vortices better. Finally, the choice of the turbulent model could be varied, starting with `k-epsilon`, which is reportedly performing well in the analysis of rotary behavior [65]. Experiments with boundary conditions, mesh refinement, and grid generation are also encouraged.

Nevertheless, the approach could be fundamentally changed, too. Simultaneously with working with OpenFOAM, some time was dedicated to exploring the opportunity to analyze the propeller through VPM. After consulting Kiril Boychev, a PhD graduate in Aeronautical Engineering from the University of Glasgow, he provided the team with access to his software Aviumtechnologies. In its essence, the Aviumtechnologies Panel Method (APM) is “a low-order, unsteady, unstructured potential flow solver for arbitrary geometries suitable for quick what-if design studies”[10]. After the first meshing of the CATIA-modelled propeller, without any further refinement, total values for thrust were approximately 2.4 N. Thus, it is expected that this analysis carries great potential to yield high accuracy results, without requiring a substantial amount of additional effort. Again, further verification shall be done to determine the nature of these changes, if either physical or due to discretization errors.

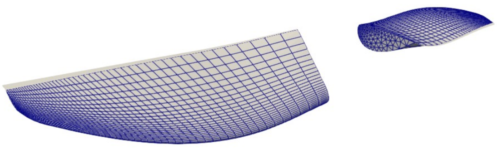


Figure 10.9: Meshed propeller with 5264 cells, hub omitted

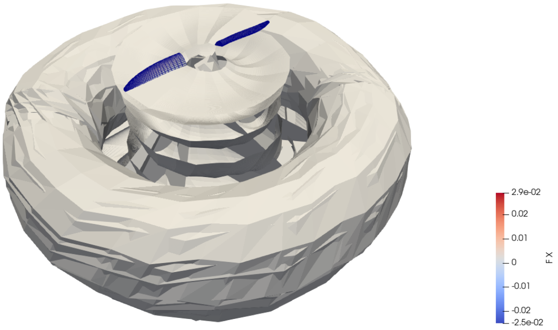


Figure 10.10: Rotor with associated wake behaviour after 185 timesteps of length 0.0003 s

To conclude, despite the steep learning curve, dedication and perseverance have paid off. Results are satisfactory and substantial progress has been made: building upon preexisting engineering knowledge, new skills have been acquired and applied in the design process. Integration with other subsystems was regularly checked upon, with most collaboration performed with the Control and Simulation department. Overall, collective efforts aim to provide a final product that not only matches set expectations but actually exceeds them.

[10] Introduction to APM: `https://docs.aviumtechnologies.com/`. Accessed 14 June 2024.

11: Communications and Data Handling

Within this chapter, the design of Cargonaut's Communications and Data Handling (CDH) subsystem is considered. First, the goals and functions of the CDH are clarified within Section 11.1. Next, the requirements and associated risks are set up within Section 11.2. This is followed by a complete description of the CDH design process in Section 11.3. After that, the verification and validation process is discussed within Section 11.4. Lastly, future recommendations are stated within Section 11.5.

11.1. Goals and Functions

The subsystem of Communications and Data Handling is hardware-based and crucial for the relay of information from the Cargonaut drone to the Ground Station (GS) or another drone and vice versa. Moreover, this subsystem ensures that all electrical components within Cargonaut can communicate with one another through targeted signals. To summarize what this subsystem aims to accomplish, the main goal can be stated as follows:

To communicate with the GS and other drones within the fleet, and to process relevant incoming data, both from Cargonaut's sensors and from the mentioned external sources.

To meet this goal, several functions to be performed by this subsystem can be set up. These are eventually crucial to the successful design of the Cargonaut drone.

1. Send and receive messages to and from the GS and other drones within the fleet
2. Process incoming data from the payload, other sensors, and the GS
3. Translate the necessary signals (e.g. location) into the correct messages to be sent to the GS or other Cargonaut drones
4. Regulate what signals are traveling through the Cargonaut drone system along with their data rates.

The design of this subsystem will be performed with a focus on these described functions. In order for the subsystem to comply with these, requirements are set up to enable the quantification such that the functions can be verified appropriately later on.

11.2. Requirements and Risks

A set of requirements is established for the subsystem of Communications and Data Handling. Many of these requirements stem from the Operations & Logistics department, as they set up the assumed warehouse layout and the expected signals to be communicated between drones. Additionally, requirements such as necessary link budgets in Cargonaut's case flow from the literature study. The complete list of requirements can be found in Table 11.1 and was linked to their respective risks.

Table 11.1: Requirements for Communications and Data Handling subsystem

ID	CDH Requirement	Risk ID
CRG-DTH-01	Cargonaut shall be able to receive and process acquired incoming data at a data rate up to 10 MB/s throughout operational time.	RSK-DTH-01
CRG-DTH-02	Communication between Cargonaut's system elements shall be ensured continuously during operations.	RSK-DTH-02
CRG-DTH-03	Cargonaut's battery life shall be logged into log 1 at least once per operational cycle.	RSK-DTH-03
CRG-DTH-04	Operations and decisions shall be logged every 1 s into log 2 for the use-case of data analysis.	RSK-DTH-04
CRG-DTH-06	The location, label, and size of any analyzed target shall be logged together into log 1, at most 1 s after determination of all.	RSK-DTH-06
CRG-DTH-07	The drone shall log the properties of at least the last 25 scanned targets into log 1.	RSK-DTH-07
CRG-DTH-08	The drone shall log the received pings from other drones into log 1.	RSK-DTH-08
CRG-COMS-01	Each drone shall be able to receive commands with a link margin of 10 dB or higher from operators or other drones when up to a distance of at least 70 m from them.	RSK-COMS-01
CRG-COMS-03	Each drone shall be able to send information to operators or other drones with a link margin of 10 dB or higher when up to a distance of at least 70 m from them.	RSK-COMS-03
CRG-COMS-04	The collected 'measurement data' of any analyzed target shall be transmitted to the ground element within 2 seconds after their determination.	RSK-COMS-04
CRG-COMS-07	The drone swarm shall be operated from a central GS transceiver.	RSK-DTH-01, RSK-COMS-01
CRG-COMS-12	Each drone shall send a ping signal to all other operational drones at a distance up to a 70 m or more every 100 ms.	RSK-COMS-12
CRG-COMS-13	Log 2 with a size up to 15 MB or more shall be communicated with GS upon demand.	RSK-COMS-13
CRG-COMS-16	Log 1 with a size up to 15 MB or more of each drone shall be sent to the server at the end of an operational cycle.	RSK-COMS-16
CRG-COMS-17	The GS shall be located in a safe area.	RSK-COMS-17
CRG-COMS-23	The communication between drones and GS shall be secure.	RSK-COMS-23
CRG-COMS-24	The location of a drone shall be communicated to the GS every 100 ms.	RSK-COMS-24
CRG-COMS-25	The GS operator shall be provided with a manual override button to take control of the drone system at any time.	RSK-COMS-25

As can be seen in the requirements table, two different logs are mentioned: “log 1” and “log 2”. The term “logging” refers to storing data in a memory log and thus the two different logs are two different files within Cargonaut’s memory that are being appended by “logging” more data into them. This ensures that different types of data are linked and can be communicated at once, by simply sending through one log. The reason for the differentiation between the two mentioned logs is that these data files need to be communicated to the receiver at different times.

Another clarification that should be made is that the term “processing” refers to the correct transferring of the incoming data from the antenna, to the WiFi module, and to the flight computer, which in turn processes the data e.g. by ensuring incoming commands are implemented. Moreover, “measurement data” will be further defined within Section 11.3.2.

The risks, represented by their identifiers in the requirements table, are shown in Table 11.2. Within this table, not only the risk, but also its likelihood, severity, and mitigation plan are presented. Furthermore, the critical risks have their identifier marked in bold; These should be mitigated throughout the design process with high care and precision.

Table 11.2: Risks for Communications and Data Handling subsystem

Identifier	**CDH Risk**	**Likelihood**	**Severity**	**Mitigation**
RSK-DTH-01	Drone fails to process all acquired incoming data at a data rate of up to 10 MB/s or more.	1	5	Design CDH hardware that can accommodate sufficient bandwidth.
RSK-DTH-02	Communication between drone’s system elements fails at any time during operations.	1	4	Implement internal communication safety systems to limit severity.
RSK-DTH-03	Drone fails to log its battery life into log 1 at least once per cycle.	1	4	Implement battery life log per drone into GS and safety stop of drone to limit severity.
RSK-DTH-04	Drone fails to log its operations and/or decisions every 1 s into log 2.	2	2	No need for mitigation.
RSK-DTH-06	Drone fails to log its label, location, and size of an analyzed target together in log 1 at most 1 s after determination of them.	2	3	Implement safety that detects fault when it occurs such that drone returns to Cargomother and can be inspected to limit severity.
RSK-DTH-07	Drone fails to log properties of at least the last 25 scanned targets into log 1.	2	2	No need for mitigation.
RSK-DTH-08	Drone fails to log all received pings from other drones every into log 1.	2	2	No need for mitigation.
RSK-COMS-01	Drone fails to receive commands without disturbances from the GS or other drones when at a distance up to 70 m or more from them.	2	4	Include redundancy in design through extra antenna.
RSK-COMS-03	Drone fails to send information to the GS or other drones without disturbances when at a distance up to 70 m or more from them.	2	4	Include redundancy in design through extra antenna.
RSK-COMS-04	Drone fails to transmit collected “measurement data” to the GS within 2 seconds after determination.	2	5	Include redundancy in design through extra antenna + safety system that detects an issue and communicates it with the GS.
RSK-COMS-12	Drone fails to send a ping signal every 100 ms to all other operational drones at a distance up to 70 m or more.	2	4	Include redundancy in design through extra antenna, safety system that detects issue and communicates it to the GS.
RSK-COMS-13	Drone fails to communicate log 2 of up to 15 MB to the GS upon demand.	1	3	Implement detection of this issue and an error message sent to GS.
RSK-COMS-16	Drone fails to communicate log 1 of up to 15 MB to the GS at the end of an operational cycle.	1	5	Implement detection of this issue and an error message sent to GS.
RSK-COMS-17	GS transceiver might fail due to external circumstances (e.g. high temperature, being hit by a moving object or person,..).	1	3	analyze the safety of its placement location before placing
RSK-COMS-23	A security breach appears in the communication between the drones and the GS.	2	5	Ensure data privacy safety is implemented within the data protocol.
RSK-COMS-24	The GS does not know the location of the drone with an accuracy of 0.3 m or more at all times.	2	5	Implement error message to GS and safe landing command.
RSK-COMS-25	The override button is not able to take control of the drone at all times.	2	5	Ensure that drone’s safety measure is taken to limit severity.

11.3. Design Process

This section discusses the complete design process regarding the Communications and Data Handling (CDH) subsystem. First, Section 11.3.1 considers the data transmission method that will be used. Secondly, Section 11.3.2 determines what data messages are going from and to Cargonaut, and the appropriate bandwidth is selected. Next, Section 11.3.3 contains an explanation of the WiFi protocol, modulation, and channel selection. After that, the link budget calculation is set up within Section 11.3.4. Lastly within Section 11.3.5, the hardware components are then selected, and the final link budget calculation is performed.

11.3.1. Data Transmission Selection

To select the data transmission method for Cargonaut’s communication system, the following characteristics are considered: cost, bandwidth, data security, range, and sustainable aspects. When examining the options of 5G, Bluetooth (IEEE 802.15.1), and WiFi (IEEE 802.11), WiFi comes out as the winner. Generally, WiFi offers a fast, reliable, and secure connection between the elements of the system while it does not require any additional infrastructure costs except for, potentially, some signal routers. It must be noted that WiFi is designed to provide wireless coverage within the immediate vicinity. The warehouse operational area for Cargonaut, as stated in Chapter 6 Section 6.1.1, is about $400\,\text{m}^2$. This operational area is considered in the requirements for the complete CDH subsystem and is considered in the next subsections as well. Under the requirement that the GS is located in a central location within this area, it can be found that drones would maximally be removed from the GS by $64\,\text{m}$. Through rounding-up, a distance of a maximum of 70 m was specified within **CRG-COMS-031**, **CRG-COMS-03**, and **CRG-COMS-12**.

11.3.2. Bandwidth Requirement

To select an appropriate frequency for the drone's communication subsystem, the required bandwidth should be considered. To determine what this required bandwidth is, an analysis is performed on all signals going from and to the drone. There are two links through which Cargonaut sends and receives messages: to and from other drones, and to and from the GS. Table 11.3 represents all types of messages the drone would send and receive throughout its operational time. For each message, its size is specified in KB, along with how often it is sent per second. Moreover, "Streams" refers to how many sources are sending out a certain message at the same continuous time: e.g. if 10 drones are sending out pings continuously at the same time, there are 10 streams ongoing [66]. Note that a drone fleet of ten drones is assumed as the maximum fleet size and that the Cargoscout communication message is also added for clarity. To clarify, a maximum of 5 drones might be in cargoscout mode at the same time, and therefore 5 streams are considered there, providing high redundancy.

It is important to understand that all messages have headers (also called "overheads"), which are used such that different devices on the same frequency channel understand whom messages are meant for. Multiple headers might be used, creating a layered structure within the message, to be deciphered on different levels in the communication system. The top header of a message for Cargonaut should include the following: source and destination Media Access Control (MAC) addresses , error checking, Frame Check Frequence (FCS) frame control, stating what type of data is being transmitted, and the duration which indicates how long the channel will be reserved for the data transaction. This layer would be considered in the WiFi module and then discarded before sending the other layers through to the UAV flight computer. A next header layer for e.g. a ping message might include the following data: transmitter ID, location, and timestamp. To better represent what a potential message would look like, Figure 11.1 is presented. The flight computer can use this header to consider the relevance of the message: e.g. an incoming ping message might only be relevant to consider for the receiver if the current location of the sender is within a certain range of the receiver. Therefore, these overheads are used to ensure the correct accommodation of data and to filter out the main payload for the respective receiving computer to consider.

Table 11.3: Messages for Communications and Data Handling subsystem

Message	Content	Transmitter	Receiver	Size [KB]	Amount [/s]	Streams
Ping	Cargonaut ID, coordinates of current and next 3 waypoint locations, timestamp	Cargonaut	Cargonaut	0.15	10	10
Measurement Data	Cargonaut ID, target ID, target location coordinates, target label (if possible), target volume	Cargonaut	GS	0.098	0.02	10
Location + Life	Cargonaut ID, current Cargonaut coordinates, current Cargonaut battery life	Cargonaut	GS	0.078	10	10
Log 1	Cargonaut ID, Measurement data of last 25 scanned targets, logged received pings of last operational cycle, most recent battery life	Cargonaut	GS	14,582.504	0.001	10
Auction Response	Bid containing 8 integers	Cargonaut	GS	0.032	0.002	10
Boxes Location	Cargoscout ID, cluster mapping: edge coordinates, target ID's and locations	Cargoscout	GS	0.562	0.01	5
Task	Cargonaut receiver ID, Area in 4 coordinates, mode code, amount of targets to scan	GS	Cargonaut	1.47	0.001	1
Safety Stop	Cargonaut receiver ID, Possible message: G = all is good, L = safety landing required, M = return to cargomother	GS	Cargonaut	0.47	10	1
Auction	location of cluster, number of targets	GS	Cargonaut	1	0.002	1

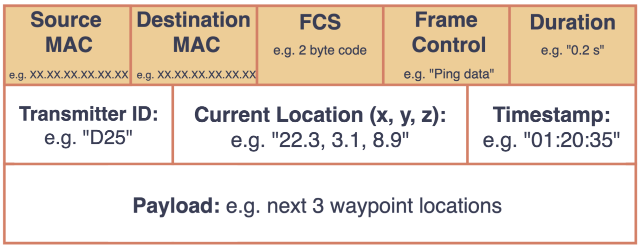

Source MAC
e.g. XX.XX.XX.XX.XX.XX
Destination MAC
e.g. XX.XX.XX.XX.XX.XX
FCS
e.g. 2 byte code
Frame Control
e.g. "Ping data"
Duration
e.g. "0.2 s"
Transmitter ID: e.g. "D25"
Current Location (x, y, z): e.g. "22.3, 3.1, 8.9"
Timestamp: e.g. "01:20:35"
Payload: e.g. next 3 waypoint locations

Figure 11.1: Example ping message format

To compute the required bandwidth, the data rates of all messages are calculated by multiplying the “Size”, “Amount”, and “Streams” data for each message. In addition, these rates are multiplied by 1.25 for a 25% margin. This margin comes from literature, which states that it accounts for message overheads and resends of lost messages [66]. Next, all these data rates can be added up to acquire the final total maximum continuous data rate of 1,860.82 KB/s. This bandwidth requirement will play a role in the choice of WiFi protocol in Section 11.3.3.

11.3.3. WiFi Protocol, Modulation, and Channel Selection

Firstly, a frequency should be selected. The following frequencies are the main options considered, as they are commonly used and recommended within the literature: 900 MHz, 2.4 GHz, and 5 GHz [66]. While higher frequencies offer larger bandwidths and data transmission speeds, they are more limited in range and object penetration than the lower frequencies. Because of this, a trade-off was made and 2.4 GHz is selected as the best frequency for Cargonaut’s communication subsystem[1]. This frequency allows for a data transmission speed of up to 600 MB/s[2,3,4], based on the hardware choices, and better penetration of solid objects, which is crucial in Cargonaut’s warehouse environment.

To choose the most appropriate WiFi protocol, the required bandwidth is considered. To start, the following 802.11 protocols, available for 2.4 GHz frequency in the ISM band are noted: 802.11 ax, 802.11 n, 802.11 g, 802.11 b[5], in order from most recent to oldest. Considering multiple criteria, the 802.11 ax WiFi 6, which was established in 2021[6], is chosen as the best protocol for Cargonaut’s case. This protocol has the highest maximum theoretical data rate of 2.4 GB/s and accommodates Multi-User Multiple-Input Multiple-Output (MU MIMO), which ensures higher signal quality and reliability. With its high theoretical data rate, this protocol complies with the required bandwidth. This protocol also allows for Orthogonal Frequency-Division Multiple Access (OFDMA)[7] modulation. This modulation accommodates multiple users at the same time and can allocate bandwidth more efficiently, thus solving parts of the congestion problem. Both this protocol and this modulation type will need to be considered when selecting the necessary hardware components.

To mitigate **RSK-COMS-01** and **RSK-COMS-03**, interference from other devices on the same frequency is taken into account. To maximally avoid interference, the available channels in the 2.4 GHz frequency range need to be understood. There are a total of 14 channels, of which three non-overlapping EU-legal ones, namely 1,6, and 11[8,9]. To select the best channel for Cargonaut’s communication system, not only this fact but also the network at the respective warehouse should be considered. For example: If 2.4 GHz channels 1 and 6 are being used the most, then Cargonaut should choose to communicate through channel 11. In case more channels than these 3 non-overlapping ones are in use, a more detailed analysis should be performed at the warehouse for Cargonaut to choose the least congested one. To find the most appropriate channel, simple software can be used, e.g. for Windows: ‘Nirsoft’s WiFi Information View’ or for IOS: ‘Wireless Diagnostics’. Figure 11.2 demonstrates an example using ‘Wireless Diagnostics’ at the Fellowship building on TU Delft’s campus. Within the “scan” window, shown in the figure, each channel detected in the network scan is shown, along with certain properties. Most important to note is that the preferred channel would have the highest Received Signal Strength Indicator (RSSI) value and lowest Noise value. As can be observed in Figure 11.2, the best 2.4 GHz channel at this location is given to be 1. This decision is based on the detected high use of channels 6 and 11. It is important to note, that the “best” channel differs depending on when the network scan is performed. Thus, to appropriately analyze what the best channel in a warehouse would be, more continuous network scans should be performed, along with an analysis of how many other routers there are in the area, and what channels are already used by other warehouse facilities. The communication hardware should be able to accommodate this dynamic design.

[1] 2.4 GHz vs. 5 GHz vs. 6 GHz: What’s the Difference? https://www.intel.com/content/www/us/en/products/docs/wireless/2-4-vs-5ghz.html. Accessed 19 June 2024.

[2] When to Use 20mhz vs 40mhz vs 80mhz https://www.cbtnuggets.com/blog/certifications/cisco/when-to-use-20mhz-vs-40mhz-vs-80mhz. Accessed 19 June 2024.

[3] The difference between 2.4 GHz and 5 GHz WiFi https://www.centurylink.com/home/help/internet/wireless/which-frequency-should-you-use.html. Accessed 19 June 2024.

[4] What’s the Difference Between 2.4 and 5 GHz WiFi? https://socialwifi.com/knowledge-base/wifi-technology/difference-between-24-and-5-ghz-wifi/. Accessed 19 June 2024.

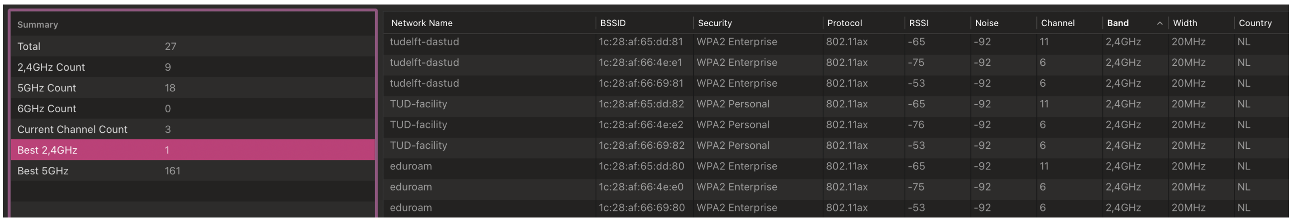
[5] Different Wi-Fi Protocols and Data Rates https://www.intel.com/content/www/us/en/support/articles/000005725/wireless/legacy-intel-wireless-products.html]. Accessed 19 June 2024.

[6] Wi-Fi 6 (802.11ax) Technical Guide https://documentation.meraki.com/MR/Wi-Fi_Basics_and_Best_Practices/Wi-Fi_6_(802.11ax)_Technical_Guide. Accessed 19 June 2024.

[7] What Is OFDMA? https://www.cisco.com/c/en/us/products/wireless/what-is-ofdma.html. Accessed 19 June 2024.

[8] Guide to Configure Wi-Fi Channels and Channel Widths to Improve Network Connection https://www.intel.com/content/www/us/en/support/articles/000058989/wireless/intel-killer-wi-fi-products.html. Accessed 19 June 2024.

[9] How to Find the Best Wi-Fi Channel for Your Router on Any Operating System https://www.howtogeek.com/197268/how-to-find-the-best-wi-fi-channel-for-your-router-on-any-operating-system/. Accessed 19 June 2024.



Summary	
Total	27
2,4GHz Count	9
5GHz Count	18
6GHz Count	0
Current Channel Count	3
Best 2,4GHz	1
Best 5GHz	161

Network Name	BSSID	Security	Protocol	RSSI	Noise	Channel	Band ^	Width	Country
tudelft-dastud	1c:28:af:65:dd:81	WPA2 Enterprise	802.11ax	-65	-92	11	2,4GHz	20MHz	NL
tudelft-dastud	1c:28:af:66:4e:e1	WPA2 Enterprise	802.11ax	-75	-92	6	2,4GHz	20MHz	NL
tudelft-dastud	1c:28:af:66:69:81	WPA2 Enterprise	802.11ax	-53	-92	6	2,4GHz	20MHz	NL
TUD-facility	1c:28:af:65:dd:82	WPA2 Personal	802.11ax	-65	-92	11	2,4GHz	20MHz	NL
TUD-facility	1c:28:af:66:4e:e2	WPA2 Personal	802.11ax	-76	-92	6	2,4GHz	20MHz	NL
TUD-facility	1c:28:af:66:69:82	WPA2 Personal	802.11ax	-53	-92	6	2,4GHz	20MHz	NL
eduroam	1c:28:af:65:dd:80	WPA2 Enterprise	802.11ax	-65	-92	11	2,4GHz	20MHz	NL
eduroam	1c:28:af:66:4e:e0	WPA2 Enterprise	802.11ax	-75	-92	6	2,4GHz	20MHz	NL
eduroam	1c:28:af:66:69:80	WPA2 Enterprise	802.11ax	-53	-92	6	2,4GHz	20MHz	NL

Figure 11.2: Network scan of the environment through 'Wireless Diagnostics'

Finally, to mitigate the risk **RSK-COMS-23**, an appropriate network security system is implemented. WiFi Protected Access 3 (WPA3) is the newest generation of WiFi Protected Access (WPA), designed to deliver simpler configuration and even stronger encryption and security than any of its predecessors (WPA and WPA2). Moreover, it is also meant to work across the latest WiFi 6 networks[10] and is thus applicable to Cargonaut's communication system. This security encryption method will thus be implemented within the design and hardware selection.

11.3.4. Link Budget Calculation Set-Up

The link budget is set up around an expanded Friis equation, presented in Equation 11.1[66]:

$$\text{Link Margin} = P_{tx} + G_{tx} - L - \text{sensitivity}_{rx} - \text{NF} - \text{SNR} \tag{11.1}$$

where P_{tx} represents the power transmitted from the sender in dBm, and G_{tx} the gain of the transmit antenna(s) in dBi.L stands for the path loss in dB, and $\text{sensitivity}_{\text{rx}}$ for the sensitivity of the receiver in dBm. Lastly, NF represents the noise figure, and SNR is the signal-to-noise ratio; both are unitless.

As will become clear: all parameters involved in the link margin calculations, postulated in Equation 11.1, can be found on the respective component data sheets of the transmitter and receiver, except for the path loss, L. The path loss can be calculated by a variety of different equations, which mainly differ based on environmental factors. For Cargonaut's communication case, it was decided to compute the path loss through the U-LAAG model. This model was built based on collected sample data from a UAV flying indoors at low altitudes in different types of environments, and it uses the following base Equation 11.2 [67]:

$$\text{L}_{\text{U}-\text{LAAG}} = 20 \cdot \text{A} \cdot log_{10}(f) + \text{B} \cdot log_{10}(d) + \text{C} + \text{X}_{\sigma} \tag{11.2}$$

Here, A and B are coefficients, which indicate the effects of frequency and distance on the path loss. C represents an offset, and X_σ is the zero-mean Gaussian distributed random variable with standard deviation σ. These coefficients depend on the environmental factors and will be estimated based on literature [67]. Moreover, f indicates the frequency in MHz at which the communication happens, and d is the distance in meters between the transmitter and receiver.

From literature, A, B, C, and X_σ, are estimated as follows: A = 0.11; B = 34.6; C = -1.5; X_σ = 1. In this estimation, the following environmental assumptions were accounted for:

- Lots of metal: e.g. from storage shelves
- Machinery with high density
- Concrete, brick, or metal walls: an average of 2 walls between the Cargonaut and ground station is considered.

As it becomes clear, the path loss varies linearly with distance and frequency. As frequency has been set to 2.4 GHz, distance is the only dynamic variable left in the equation.
Ideally, the final calculated link margin is higher than 10 dB [66]. Therefore, when calculating the link budget based on the distance between the transmitter and receiver, this wanted minimum is leading for the design and the maximum distance at which drones and the ground station will be able to communicate undisturbed with each other.

11.3.5. Component Selection and Link Budget Calculation

Now that the network frame has been decided upon, it is integral to select the appropriate hardware components for the CDH subsystem. To start this selection procedure, some new requirements can be set up, based on choices made earlier in this chapter. The necessary hardware selection requirements can be found in the bullet point list that follows. Note that these requirements do not come with new risks. The consequence of not meeting one of the requirements would merely lead to a reiteration of the decisions on which the requirement was based.

[10] What Is Wi-Fi Security? https://www.cisco.com/c/en/us/products/wireless/what-is-wi-fi-security.html. Accessed 19 June 2024.

- The hardware needs to be able to communicate on the 2.4 GHz frequency band.
- The hardware needs to be able to communicate through the 802.11 ax WiFi protocol.
- The hardware needs to allow for WPA3 security encryption.
- The hardware needs to allow for dynamic adjustment of the channel.
- The hardware needs to contain a memory of at least 35 MB.

Moreover, the stated budgets for the communication subsystem and data handling subsystem are leading in this selection: the mass budget is 87 g, cost budget 180 €, and power budget of 11.5 W, as provided in Section 15.3. Note that for the purpose of data handling, the budgets for data processing and communications were summed up. Additionally, part of the payload budget is used for the flight computer.

To start, a flight computer is chosen. This component will act as the brain of the drone, having to process all incoming data, run all software, and communicate with all other components internally. As a flight computer, the Raspberry Pi 5 was selected. This component is a lightweight computer with a memory SD card of 32 GB and an integrated WiFi module [68]. This WiFi module, the Infineon CYW43455 chip, can run on 2.4 GHz frequency and with the 802.11 ax protocol. Moreover, it can accommodate for the WPA3 security.

Since a high amount of software will be running on Cargonaut at all times (e.g. AI detection or path planning), an extra component will be selected to be joined with the Raspberry Pi 5 in order to increase Cargonaut's computational power. The Hailo-8L AI accelerator TPU is selected for this task and will be integrated into the Raspberry Pi, using a Raspberry Pi M.2 hat PCB[11] The main characteristics of this Hailo are that it provides up to 13 Tera Operations Per Second (TOPS) and it allows for Advanced AI video analytics.

Now, as the Raspberry Pi contains a WiFi module and an antenna, the communication hardware is technically complete. However, in light of mitigating risks **RSK-COMS-01** and **RSK-COMS-03**, an extra antenna is chosen for redundancy and signal quality increase through adding gain. This antenna is the 1608 2.4G Chip Antenna: it is exceptionally small in size and lightweight [69]. As such, it is mounted on a small Printed Circuit Board (PCB) which will be placed on the bottom side of Cargonaut to maximize its communicational range. Some characteristics of this antenna are its gain of 2 dBi and its low power usage of maximum 1 W. Moreover, the antenna is omnidirectional and allows for a bandwidth up to 150 MHz [69].

Lastly, it is important to consider the hardware integrated into the Ground Station (GS), the WiFi router. This is mainly relevant in the scope of this subsystem design for link margin calculation purposes. The chosen router is the AX6000 Dual-Band Gigabit WiFi 6 Router [70], which contains eight high-gain antennas and can accommodate the 802.11 ax WiFi protocol, 2.4 GHz frequency, and WPA3 security[12]. This router has an added gain of 18 dBi, which is calculated from the eight antennas, which have a gain between 5-9 dBi[12].

The mass, cost, and power consumption of each drone system component selected in this section is shown in Table 11.5. As can be seen, all of these parameters are within the budgets as defined earlier in this subsection.

Table 11.4: Hardware data

Component	Mass [g]	Cost [€]	Power consumption[W]
Raspberry Pi 5	48	72.44	11.6
Hailo + M.2 Hat	34	79.80	3.3
Antenna	1	0.33	1

Now that the communication hardware has been determined, the link budget calculations, set up in Section 11.3.4, can be performed. First, the necessary data needs to be acquired from the WiFi module, Cargonaut antenna, and GS receiver. This data was found in the literature.

Table 11.5: Communications hardware data

Parameter	Unit	WiFi Module	Cargonaut Antenna	Ground Station
P	dBm	20	-	20
G	dBi	3.5	2	18
Sensitivity	dBm	-85	-	-89
NF	-	10	-	10
SNR Margin	-	25	-	30

[11] Official Raspberry Pi AI Kit with Hailo 8L TPU for Pi 5 https://www.welectron.com/Official-Raspberry-Pi-AI-Kit-with-Hailo-8L-TPU-for-Pi-5. Accessed 19 June 2024.

[12] TP-LINK Next-Gen Wi-Fi 6 Router AX6000 (Archer AX6000) https://www.wifi-stock.com/details/tp-link-ax6000-next-gen-wi-fi-6-router-archer-ax6000.html. Accessed 19 June 2024.

With these parameters known, Equation 11.2 and Equation 11.1 can be used to calculate the link margin in terms of the distance between the Cargonaut and its transceiver e.g. another drone or the GS. These calculations were performed through a short Python script and a graph was generated, as can be seen in Figure 11.3. Note that the ideal minimum link margin of 10 dB was also graphed as a red line, thus indicating up to what distance Cargonaut would have quality communication with its respective transceiver.

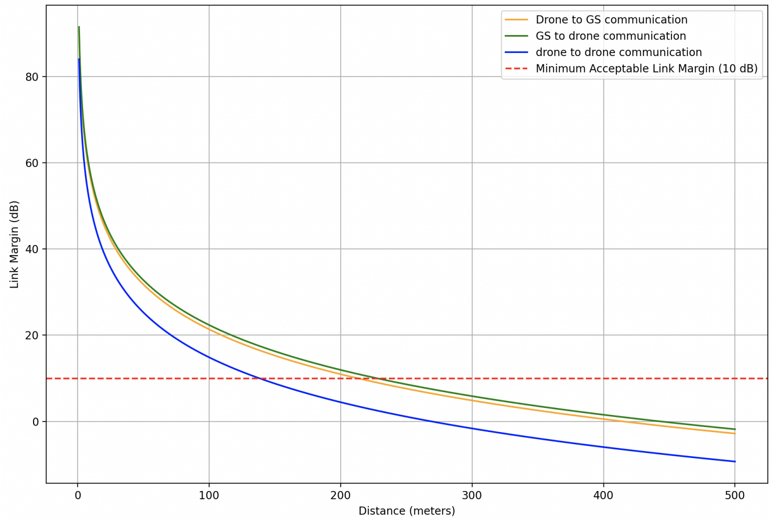


Figure 11.3: Link margin vs. distance for three communication line types

From the graph, it can be seen that drone-to-drone communication will happen without disturbances up till the drones are located at 130 m from each other. Regarding the Cargonaut - GS communication link, a distance up to 203 m is possible, limited by the Cargonaut to GS link. These numbers thus ensure the compliance of Cargonaut's communication system with **CRG-COMS-01** and **CRG-COMS-03**.

With all these components selected, the data handling block diagram is set up. This diagram can be found in Figure 11.4. All PCBs contained within Cargonaut are listed and connected according to what data they are sending where and at what data rate. An interesting observation to make is that the sending of images or frames requires the highest data rates. This is due to the large size of the images and the high frequency at which they are being sent between Cargonaut's components. Another thing to note is that the antenna acts as the direct communication link between Cargonaut's flight computer and another drone or the GS. Therefore, all information going in and out of Cargonaut passes by both the flight computer and the antenna.

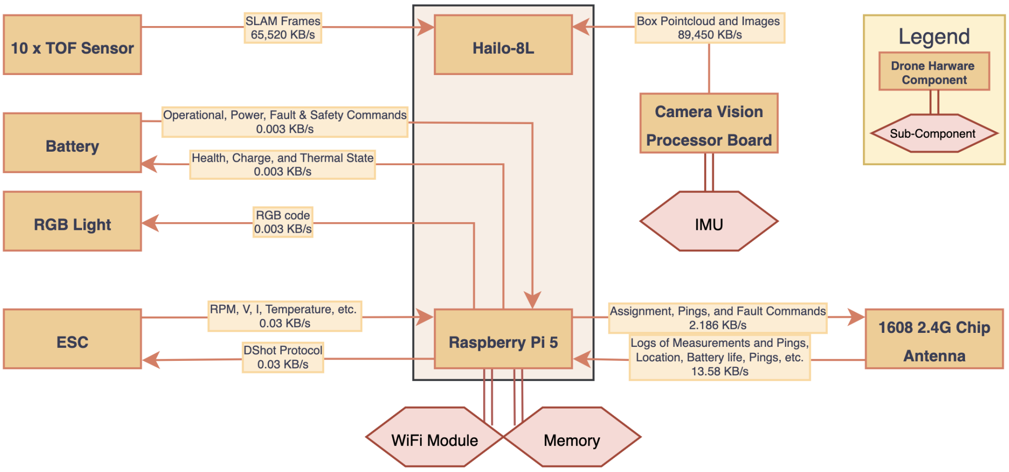


Figure 11.4: Data handling

11.4. Verification and Validation

To verify and validate the communication and data handling subsystem, multiple methods can be employed. Both for the used code and for the hardware component choices, verification and validation can be performed.

To start, the choice of frequency was validated through literature research and in-person communication with TU Delft's Mavlab experts. Both stated that 2.4 GHz indeed is preferred over 5 GHz for certain drone applications [67]. Next to that, the chosen WiFi and security protocol can also be validated through a literature study on past drone designs [71]. Following that, the choice of hardware components was also validated using both mentioned sources: Raspberry Pi's are often used as flight computers on drones, even so in Mavlab's applications [72]. Moreover, the choice of an additional computational TPU was validated, and its integration within the Raspberry Pi 5 can be verified through research that shows the components are compatible, using an additional M.2 Hat to physically couple them[13]. The use of an extra antenna is also a common practice within Mavlab. Lastly, the other hardware components in the data handling diagram and their data rates were obtained through communication with the other departments on the team, and cross-checked by the CDH department for verification purposes.

Following that, the code that was created to build the link margin graphs in Figure 11.3, consisting of 83 lines. The inputs and formulas used were verified. The inputs stemming from data sheets and estimation were used correctly, and the formulas for Friis' equation and the U-LAAG path loss function were correctly implemented. For validation, the input values from the components were compared to other existing values from similar components. Moreover, the resulting range, which was decided on based on the resulting graphs from the Python script, was validated, both through literature research and from communication with drone experts at TU Delft's Mavlab [67]. It became clear that generally, drone-to-GS communication usually allows for less distance than GS-to-drone communication. This validates what can also be seen in Figure 11.3: that drone-to-GS communication fails first.

Lastly, Table 11.6 represents the compliance matrix of the CDH requirements. It shows how the requirements were verified and whether the final subsystem design complies with them. Note that all these requirements should and will still be tested once an initial prototype of Cargonaut exists. Moreover, it would be preferred to also create a test set-up within a warehouse to properly test all functions and compliance with requirements

Table 11.6: Compliance matrix for Communications and Data Handling subsystem

ID	Verification Method	Compliance
CRG-DTH-01	Analysis to ensure communication components support high data rate from 208.11 ax WiFi protocol.	✓
CRG-DTH-02	Analysis	✓
CRG-DTH-03	Analysis to ensure the necessary data rate for logging is supported.	✓
CRG-DTH-04	Analysis to ensure the necessary data rate for logging is supported.	✓
CRG-DTH-06	Analysis to ensure the necessary data rate for logging is supported.	✓
CRG-DTH-07	Analysis to ensure the memory unit size is large enough to store the necessary data.	✓
CRG-DTH-08	Analysis to ensure the necessary data for logging is supported.	✓
CRG-COMS-01	Analysis through link margin calculation: GS or drone to drone allows for communication up to at least 100 m.	✓
CRG-COMS-03	Analysis through link margin calculation: drone to drone or GS allows for communication up to at least 100 m.	✓
CRG-COMS-04	Analysis to ensure that data rate over WiFi protocol 208.11 ax supports this.	✓
CRG-COMS-07	Central GS will send tasks and auction options to drones.	✓
CRG-COMS-08	Analysis	✓
CRG-COMS-09	Analysis	✓
CRG-COMS-12	Analysis to ensure that bandwidth is able to support necessary data rates for pings.	✓
CRG-COMS-13	Analysis to ensure that data over WiFi protocol 208.11 ax supports this.	✓
CRG-COMS-16	Analysis to ensure that data over WiFi protocol 208.11 ax supports this.	✓
CRG-COMS-17	When installing the drone fleet within a warehouse, careful analysis will be performed to choose a safe location for GS.	TBC
CRG-COMS-23	Analysis to ensure the necessary security protocol WPA3 is implemented.	✓
CRG-COMS-24	Analysis to ensure that the data over WiFi protocol 208.11 ax supports this.	✓
CRG-COMS-25	Analysis to ensure that GS sends a constant binary value indicating whether manual override is needed.	✓

11.5. Future Recommendations

As this CDH subsystem design process had a limited timeline, some recommendations should be made for how the design process could be more complete or efficient. Firstly, it would be a plus if double-band hardware was considered, as this would allow for the drone communication to seamlessly switch to 5 GHz frequency communication in case the 2.4 GHz frequency was to become congested. Currently, the Raspberry Pi 5 allows for this double-band communication system, but the newly added antenna does not. Secondly, an in-depth analysis could be performed on latency rates of the communication system: e.g. how long it takes for a signal/data to go from the transmitter to the receiver. This would be a nice addition to the design as it could help estimate network channel congestion at times of sending large data. Lastly, when integrating the drone fleet into a warehouse, the network performance graphs from e.g. 'Wireless Diagnostics' on an IOS should be used to verify the data rates and signal quality at the selected channel. The information from these graphs could then be used in iterative design by adjusting Cargonaut's network selection as needed.

[13] Official Raspberry Pi AI Kit with Hailo 8L TPU for Pi 5 https://www.welectron.com/Official-Raspberry-Pi-AI-Kit-with-Hailo-8L-TPU-for-Pi-5. Accessed 19 June 2024.

12: Electronics

The electronics of the drone are discussed in this chapter. The chapter begins by presenting the goals and functions in Section 12.1. This is followed by a description of the requirements and risks in Section 12.2. The design process is elaborated on in Section 12.3. Section 12.4 gives the verification and validation procedures. Lastly, Section 12.5 proposes future recommendations.

12.1. Goals and Functions

Electronics is the powerhouse of the drone, similar to the mitochondria of the cell. Not only does it involves operations about the battery, but it also devises electronic connections for power transmissions. To achieve this, a goal for the subsystem has been formulated, which is:

Support the uninterrupted supply of power to the subsystems of the drone and provide sufficient energy for safe operation throughout the mission lifetime.

Meeting this goal is governed by several functions that shall be performed. Most of them are linked to the performance of the battery, as this is the component that will supply power for the operations of the drone.

1. Connect and integrate hardware for parallel operations
2. Provide power for the operations of the drone
3. Redirect electricity from the battery to wherever it is needed
4. Supply correct voltage and current according to specifications.

Ideally, the design team would devise a guarantee that all functionalities are properly addressed and covered. The result is implied confidence that the set configuration achieves the established objectives. Next, the functions form a basis for generating a list of requirements, that encompasses all stakeholder needs. Naturally, there exists a possibility that some requirements would not be met, formulated as a risk. Based on the likeliness of its occurrence and associated severity, mitigation strategies are devised.

12.2. Requirements and Risks

Requirements for the electronics subsystem are listed in Table 12.1, alongside associated risks, elaborated upon in the subsequent Table 12.2. In terms of formulated risks, mitigation plans address them and shall be implemented throughout the entire working process. It can be noted that some of them are repetitive, but this is due to the similar nature of the identified risks. For instance, an unsuitable amount of power delivered to a given component, no matter what it is, would severely hinder performance. This could be alleviated by strict analysis, constant monitoring, and adherence to power budget allocations.

Table 12.1: Requirements for Electronics subsystem

Identifier	Power (Electronics) System Requirement	Associated Risk ID
CRG-POW-01	The drone shall support nominal operations for at least 30 minutes at the end of life.	RSK-ELEC-01
CRG-POW-03	The drone shall fly to the charging station when battery levels are under 20%.	RSK-ELEC-03
CRG-POW-04	The drone shall be able to recharge its battery in at most 30 minutes.	RSK-ELEC-04
CRG-POW-05	The drone shall be equipped with a power distribution system to connect all subsystems.	RSK-ELEC-05
CRG-POW-06	The EPS shall be able to provide an average of 5 W to the camera at all times.	RSK-ELEC-06
CRG-POW-07	The EPS shall provide a peak power of 5 W to operate the instruments.	RSK-ELEC-07
CRG-POW-09	The EPS shall provide 215 W to actuate the propulsion system.	RSK-ELEC-09
CRG-POW-10	The EPS shall be able to supply a minimum of 15 W to the flight controller at all times of the mission.	RSK-ELEC-10
CRG-POW-11	The drone shall be equipped with a light to indicate the operational status (on/off).	RSK-ELEC-11
CRG-POW-12	Loss of power shall trigger the landing configuration.	RSK-ELEC-12
CRG-POW-13	The battery shall be disposed of sustainably.	RSK-ELEC-13
CRG-POW-14	The EPS shall provide interfaces to ensure compatible voltage requirements of all parts.	RSK-ELEC-14
CRG-POW-15	The EPS shall provide interfaces to ensure compatible current requirements of all parts.	RSK-ELEC-15
CRG-POW-16	The EPS shall provide compatible connector types for all parts.	RSK-ELEC-16
CRG-POW-17	The battery shall have a maximum depth of discharge (DOD) of 80%.	RSK-ELEC-17
CRG-POW-18	The battery shall be removable to enable swapping at the docking station.	RSK-ELEC-18

12.3. Design Process

The department of Electronics is in charge of not only specifying a power source, but also ensuring that it is capable of delivering a sufficient amount of energy to meet imposed requirements. Therefore, the first step is to design the battery, followed by connectivity analysis to establish the correct power supply according to wiring.

Table 12.2: Risks for Electronics subsystem

Identifier	Electronics System Risk	Likelihood	Severity	Mitigation Plan
RSK-ELEC-01	battery capacity degrades faster than expected, higher power draw than expected leads to violation of endurance requirements.	3	4	Regular monitoring of battery health and maintenance, design for power optimization
RSK-ELEC-03	battery dropping below 20% does not alert the drone or initialize return to base mode	2	3	Implement a reliable low battery alert system, rigorously test it, and ensure charging stations are strategically placed
RSK-ELEC-04	battery cannot recharge within half an hour	2	3	use fast-charging, choose a suitable battery C-rate
RSK-ELEC-05	power distribution system fails to connect all subsystems.	1	4	Not necessary, additionally could perform physical system checks
RSK-ELEC-06	camera malfunctioning	2	5	Adherence to power budget, strict monitoring of power distribution
RSK-ELEC-07	instrumentation malfunction	1	4	Adherence to power budget, strict monitoring of power distribution
RSK-ELEC-09	Propulsive system malfunction	2	5	Adherence to power budget, proper motor selection, strict monitoring of power distribution
RSK-ELEC-10	EPS fails to supply a minimum of 15 W to the flight controller, leading to loss of control.	2	5	Adherence to power budget, strict monitoring of power distribution
RSK-ELEC-11	light fails: no visual means of battery life identification and uncertainty about drone status	2	2	Not necessary
RSK-ELEC-12	landing configuration does not trigger, resulting in a crash and possibly critical damage	2	5	rigorous analysis and fail-safe system tests, shall be physically tested in an appointed facility
RSK-ELEC-13	unsustainable battery disposal leads to environmental harm	4	2	follow a strict battery disposal method, devise an End of Life (EOL) strategy.
RSK-ELEC-14	inadequate voltage supply leads to malfunctioning and possible overheating	1	4	check components compatibility based on technical data
RSK-ELEC-15	inadequate current supply leads to malfunctioning and possible overheating	1	4	check components compatibility based on technical data
RSK-ELEC-16	connectivity issues	1	3	Not necessary, risk avoided by standardizing connector types and inspecting connections on physical assembly
RSK-ELEC-17	battery lifespan degrades at a faster pace	3	3	monitor battery level, design operational mode for low battery life
RSK-ELEC-18	non-removable battery complicates swapping at the ground station	3	3	design the drone with a modular structure, have a safe environment for battery storage

12.3.1. Power Source

The choice of battery is of utmost importance. Not only does it influence the lifetime of the whole assembly, but it also supports the correct functioning of each component. In addition, considering sustainability principles, the power source is a key factor. Batteries are hard to recycle due to the hazardous materials they consist of, which demand complex processes to dispose of in an appropriate manner. As a result, poor management of resources could severely harm the environment. Nonetheless, this is not a valid reason not to adopt recyclability as a target goal. In fact, the materials should be sourced, processed, and disposed of responsibly. According to goals, set by the EU Green Deal, material recovery targets for lithium will be 50% by 2027 and 80% by 2031[1]. These considerations are guiding throughout the full design process, documented in this subsection.

Earlier, in Figure 9.1, relevant battery specifications were first introduced. Battery types vary based on a number of parameters, with the most driving being type of material, capacity, and C-rate (or rate of discharge), with a higher C-rate translating into lower internal resistance. Naturally, these specifications influence the corresponding mass and price as well.

Analysis of requirements determined that the most important consideration is the endurance requirement, being that the drone shall be able to operate in the air for 30 min before its energy is depleted. Therefore, the design was conducted in a manner that opts for maximum battery capacity. Considering typical instrumentation power requirements, and referring to the budget in Table 5.3; at the conclusion of the preliminary design phase, it was established that the power source should be capable of delivering 270 W, including contingencies. A driving constraint, this already poses severe restrictions on viable options. Preliminary market research indicated that a high battery capacity is often associated with increased mass, which allowed the team to specify the battery's *specific energy* as the parameter of highest interest.

Selection of battery type:via research, it was identified that two types of batteries would be most suitable. These are Lithium polymer (LiPo) and Lithium-ion (Li-ion). Whereas Li-ion provides high energy density, which allows for mass reduction, it does in fact have a very low C-rate, typically reaching 5-10 C at maximum. Commercially available options could never suffice in terms of expected current load, mostly dictated by the maximum throttle setting and expected to reach 12-15 C with the selected propulsive system. Additionally, the internal resistance of these batteries is much higher than alternative options, and degrades greatly with time, providing another source of power losses and heating to the system. Therefore, it was decided that the battery should be LiPo, gaining confidence by the fact it is indeed the most widely used option for electrical UAVs.

[1] Green Deal: EU agrees new law on more sustainable and circular batteries to support EU's energy transition and competitive industry : https://ec.europa.eu/commission/presscorner/detail/en/ip_22_7588. Accessed 18 June 2024.

Battery capacity: next, the capacity of the battery was considered. The market search indicated that typical values in terms of drone batteries fall within the range of 1,000 mAh to 5,000 mAh, with some options reaching values up to 12,000 mAh. However, selecting an option with higher capacity also leads to a significant weight increase and shall be carefully evaluated. Therefore, initially, an upper bound of 10,000 mAh was agreed upon. It was important to negate the possible "snowball effect" that could be caused by choosing a bigger battery: a higher battery mass would lead to higher power consumption, which iteratively would demand supplying more energy.

Preliminary analysis indicated that a higher capacity quickly scales the mass, while not providing a significant increase of flight time. After searching the market for suitable batteries, a few provisional options were compiled. In terms of the number of cells, typical configurations involve connections both in series (S), as well as in parallel (P). Linking 4 cells in series (4S) was deemed as the most suitable option for this mission's needs. It provides sufficiently high voltage to reduce power losses and allows keeping the weight to a reasonably low figure. In conclusion, five potential contenders ranging from 4,200 mAh to 10,000 mAh in capacity were gathered for future references, given in Table 12.3. For consistency, a single manufacturer - DXF[2] - was used for battery selection, chosen due to the high specific energy and overall performance of listed products when compared to competitors.

Table 12.3: DXF battery specifications

Battery Capacity [mAh]	**Mass** [g]	**Specific Energy** [Wh/kg]	**Price** [€]
10,000	740	200.00	63
8,400	**592**	**210.00**	**89**
7,000	547	189.40	52
5,200	488	157.70	44
4,200	403	154.20	39

An interesting remark can be made when comparing the first two batteries listed. The battery with lower capacity has a higher cost, suggesting better performance. This expectation is reinforced by the higher specific energy and will be put to the test as design proceeds.

Viable configurations were assessed via the online tool eCalc and its xcopterCalc[3], specifically designed for multicopter design. With Verification and Validation (VnV) provided in Chapter 9, the reliability of the obtained results is promoted. In summary, the option that was indicated as best is the one with 8,400 mAh capacity[4], highlighted in bold in Table 12.3. It supports 5C fast charging, directly meeting **CRG-POW-04**. Furthermore, conservative analysis concluded that the 30 min hover time could be met, but degradation throughout the mission lifetime shall be modeled, too.

EOL Battery Strategy

An important aspect of the power unit on the drone is its mission lifetime, along with the end-of-life (EOL) strategy. Although quite advanced from a technological standpoint, LiPo batteries have a limited cycle life - typically maintaining optimal discharge rates for less than 500 cycles. Therefore, a robust operational cycle and end-of-life solution must be devised to ensure that the drone aligns with the sustainability and reliability goals. To extend the lifespan of the battery, it is critical to follow operational guidelines postulated by the manufacturer. A key component of this is the integration of a balanced charging mode in the Cargomother architecture. Unlike traditional charging methods that apply a fixed potential difference until fully charged, balanced charging incrementally increases the potential difference throughout the process. This balanced approach minimizes damage to the batteries, significantly enhancing their lifespan and ensuring consistent performance. Additionally, at the end of the lifecycle, or every 500 cycles (which would entail operations for around a month), the batteries are removed and sent to a dedicated recycling station [5] that employs a clean recovery process to obtain metals or other raw material. This battery supply and recycling strategy will be mediated by Cargonaut as an independent service, thus supporting environmental responsibility and operational efficiency.

12.3.2. EPS Architecture

The EPS system, as the name suggests, is responsible for distributing power and other electrical systems through the components of the drone, ensuring its correct functioning as a whole. Therefore, in this section, the components of the drone will be assessed and the connections for a functioning power system shall be established.

[2] Welcome DXF lipo battery official store, Softcase 4S: https://dxf-hobby.store/collections/softcase-4s. Accessed 18 June 2024.

[3] xcopterCalc - Multicopter Calculator: https://www.ecalc.ch/xcoptercalc.php. Accessed 16 June 2024.

[4] (CN)DXF 4S Lipo Battery 14.8V 120C 8400mAh Soft Case Battery: https://dxf-hobby.store/collections/softcase-4s/products/cndxf-6s-lipo-battery-22-2v-120c-8400mah-soft-case-battery-with-ec5-xt90-connector-for-car-truck-tank-rc-buggy-truggy-racing-hobby-1. Accessed 18 June 2024.

[5] Lithium-Ion Battery Recycling Service: https://www.genoxtech.com/en/news_i_guides-to-lithium-ion-battery-recycling-machines-and-systems.html. Accessed 18 June 2024.

At this stage of the design, most of the components from the subsystems of the drone have been chosen and finalized. However, there are still some choices remaining to be made. One of these is the ESC; this component was discussed as a part of the propulsion subsystem in Chapter 9, but no final choice was made on it, given the versatility of the component. Therefore, at this stage, an ESC is chosen: the Racerstar Air50 4-in-1[6]. This component supports a maximum continuous current of 50 A, well above the expected consumption of the motors, which proves it suitable for the current design. Additionally, a 4-in-1 ESC comes with the advantage of less wiring and mass compared to its one-ESC-per-motor counterpart, further solidifying this choice.

Additionally, the possibility of having an LED indicator on the outer surface of the drone was posed as a general idea. It could provide information on the current operational mode, system/shutdown, and battery status to the workers of the warehouse, making it more user-friendly. It would also be useful as an outside indicator for certain conditions during testing. For this reason, the Speedybee 2812 Light Strip [7] was added to the list of components.

After the addition of both of these, the list of essential components was finalized. Each components and its interfaces can be visualized in Table 12.4. This data was the starting point of the analysis of the electronic integration of the system.

Table 12.4: Different interfaces and voltages of each component

Component	Interface	Operating voltage [V]
Battery	Power supply	14.8
ESC	Power supply	14.8
	Input data (from RPi)	5
	Output data (to motor)	Variable
Motors	Input data	Variable
Raspberry Pi 5	Power supply	5
	Voltage pins	5 / 3.3
	GPIO pins	3.3
ToF sensors	Power supply 1	3.3
	Power supply 2	1.8
	Input/Output data	1.8
LED	Power supply	5
	Input data (from RPi)	5
Camera	Power supply	5
	Input/Output data	N/A, USB cable to RPi
AI module	Main interface	N/A, Ribbon cable to RPi
Antenna	Main interface	N/A, Soldered to RPi

In this table, the wide range of voltages for the multiple interfaces of the drone can be seen. Some connections have already been taken care of: the camera, AI module, and antenna showcase simple connections to the Raspberry Pi. Additionally, the ESC can run at various protocols that are compatible with the motors, so their hardware integration is quite simple. However, for all other interfaces, proper conversion of voltage for data and power buses needs to be analyzed.

For the data buses, level translators need to be employed. These are quite simple in architecture and are available with multiple channels of bidirectional communication so that general input/output signals can be correctly transmitted. In this case, two-level translators needed to be employed: one 1.8 V to 3.3 V translator, to ensure communication between the ToF sensors and the Raspberry Pi, and one 3.3 V to 5 V translator, for communication of LED and motor inputs. For both of these applications, the ADG3308 8-channel translator[8] was deemed suitable. It is compatible with both voltage switch ranges, displays a low current consumption, and provides enough channels for both the ToF sensors (whose communication could be done in 2 or 4 shared data buses, depending on the communication protocol) and the LED and motors (taking one data bus each), even allowing for the possibility of adding more components at a later stage.

For the power voltage conversions, multiple options are available. For DC-DC conversion at this scale, which is the case, two options are mostly used: Low-Dropout Regulator (LDO) and buck regulators. LDO changes their internal resistance and drops off the extra voltage as heat. They are very simple and cheap, but at the cost of a low efficiency; since they merely dissipate the excess voltage as heat, power losses can occur for big voltage drops. In fact, the efficiency of such a regulator is proportional to the ratio of the output to the input voltage. For

[6] Racerstar Air50 4in1: https://usa.banggood.com/30_5+30_5mm-Racerstar-Air50-3-6S-50A-4In1-ESC-Built-in-Current-Sensor-BLheli_S-DShot600-Compatibled-with-AirF7-Lite-p-1772372.html?cur_warehouse=CN. Accessed 18 June 2024.

[7] Speedybee 2812 Light Strip: https://nl.aliexpress.com/item/1005005974813952.html?src=google. Accessed 18 June 2024.

[8] ADG3308: https://eu.mouser.com/datasheet/2/609/ADG3308_3308_1-3120235.pdf. Accessed 18 June 2024.

example, in the case of a 5 V to 1.8 V regulator, an efficiency of only 36 % is achieved. Buck regulators, on the other hand, achieve higher efficiencies (on the order of 85 % to 95 %), at the cost of added complexity. These operate by storing energy in an inductor, so they do not dissipate large amounts of energy even for high voltage drops. With these considerations in mind, the LM64460CPPQRYFRQ1[9] buck converter was selected for the conversion from 14.8 V to 5 V; since this is quite a high voltage drop, and most of the components are powered at 5 V, a LDO would not be suitable for this application. For the voltage drop from 5 V to 3.3 V and 1.8 V, however, two TPS78518QWDRBRQ1 LDO[10] were selected. Since the 3.3 V and 1.8 V voltage bands only power the ToF sensors, and these have a fairly low power consumption, the added complexity of two extra buck converters was not justified.

After finalizing the selection of all the electronic components, the Electronics Block Diagram could be devised, as seen in Figure 12.1. In this figure, the multiple different components of the drone can be visualized, alongside the necessary electrical connections and their voltages. It is important to note that although there is only one ToF sensor represented in the diagram for simplicity, there are ten present in the drone.

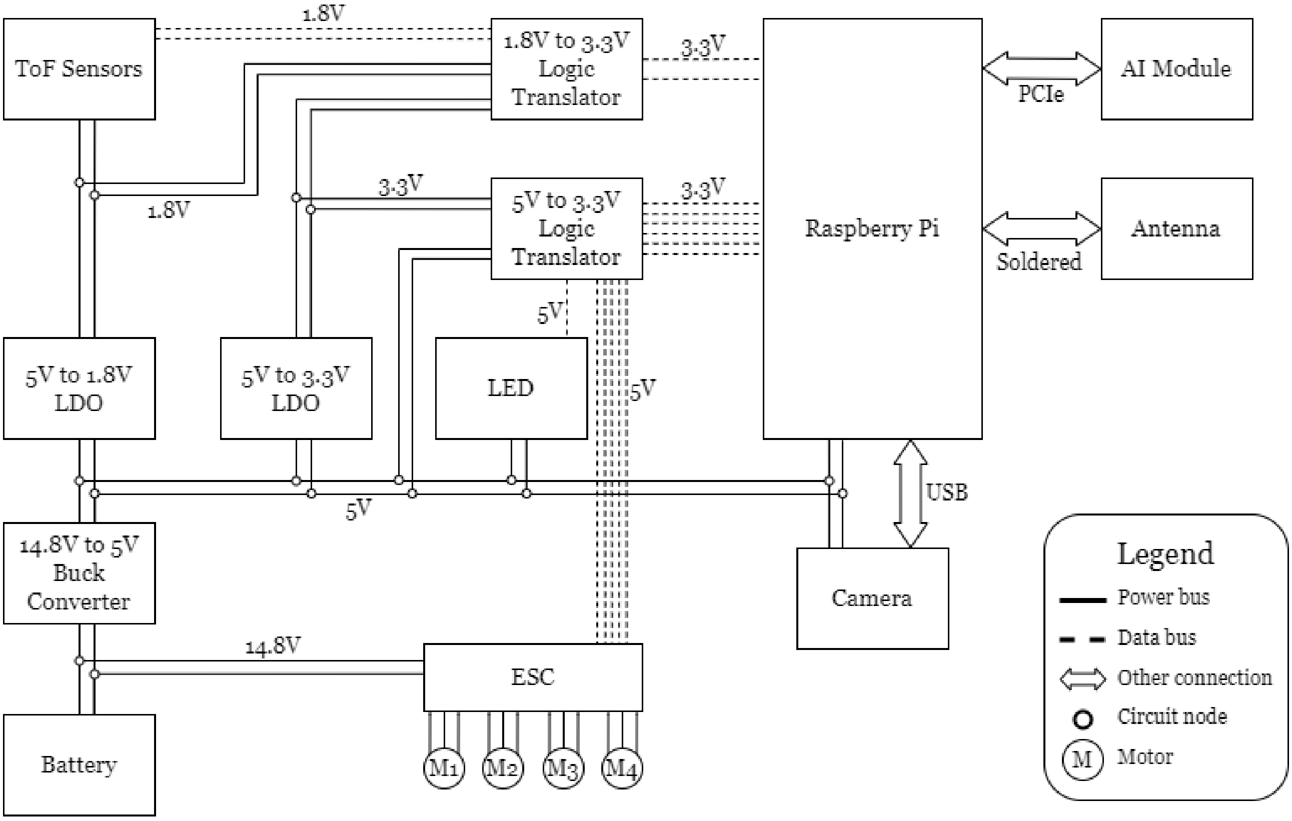


Figure 12.1: Electrical Block Diagram

12.3.3. Power Delivery and Consumption

Given the established architecture of the electrical subsystem, it is important to assess how the power consumption stated at earlier stages of the design has changed. For that purpose, a thorough analysis of every component and cable, with their power loss, is done. Table 12.5 displays the power consumption of each component as indicated in the manufacturer datasheet, taken as the start of the analysis. From manufacturer data sheets, it was also known that the current draw of the LDO and the level translators were very low (to the order of µA), so these are not accounted for, and that the efficiency of the buck converter is 92.5 %. With this in mind, the power computation for the entire system was done as follows:

- For both the 1.8 V and the 3.3 V grids, the power consumption was calculated simply by adding cable losses, based on component current and cable type.
- These power values are then converted to the 5 V grid, according to the efficiency of a LDO, and then added to the components and their cable losses, to arrive at the consumption at this grid.

[9] LM64460CPPQRYFRQ1: https://www.ti.com/lit/ds/symlink/lm64460-q1.pdf?ts=1718688696889&ref_url=https%253A%252F%252Fwww.ti.com%252Fproduct%252FLM64460-Q1. Accessed 18 June 2024.
[10] TPS78518QWDRBRQ1: https://eu.mouser.com/ProductDetail/Texas-Instruments/TPS78518QWDRBRQ1?qs=iLbezkQI%252BshL0QUXVboi%2Fg%3D%3D. Accessed 18 June 2024.

- Finally, this can be done a step further down at 14.8 V, using the efficiency of the buck converter, to arrive at the total consumption of the entire system.

Additionally, Table 12.7 provides additional information as to the types of cables used in the EPS and the losses they incur.

Table 12.5: Average power consumption for each hardware component

Component	Average power [W]	Operating voltage [V]	Number
Camera	3.461	3.3	1
ToF Sensors	0.1419	3.3	10
	0.09	1.8	
Raspberry Pi	11.6	5	1
AI module	3.3	N/A, connected to RPi	1
Antenna	1	N/A, connected to RPi	1
LED	0.06	5	1
ESC/Motors	213.4	14.8	1

Table 12.6: Power consumption at voltage level

Voltage [V]	Type	Average power [W]
1.8	Components	0.9000
	Cables	0.06381
	Total	0.9638
3.3	Components	1.419
	Cables	0.04719
	Total	1.466
5	Components (including 1.8 and 3.3V)	24.32
	Cables	0.6066
	Total	24.93
14.8	Components (including 5V)	240.35
	Cables	1.05
	Total	**241.40**

Table 12.7: Power losses for each main cable connector

Cable	Wire gauge	Resistivity [Ohm/m]	Current [A]	Length [m]	Number	Total power loss [mW]
1.8V - TOTAL						
ToF power and ground	28AWG	0.2127	0.05	0.3	20	3.191
3.3V - TOTAL						
ToF power and ground	28AWG	0.2127	0.043	0.3	20	23.60
5V - TOTAL						
1.8-5V LDO to buck	26AWG	0.1338	0.9638	0.15	2	37.29
3.3-5V LDO to buck	26AWG	0.1338	1.466	0.15	2	86.29
LED power and ground	28AWG	0.2127	0.012	0.15	2	0.009189
RPi power and ground	22AWG	0.0529	3.18	0.15	2	160.5
Camera Module power and ground	26AWG	0.1338	0.6922	0.15	2	19.23
14.8V - TOTAL						
Buck power and ground	14AWG	0.00828	1.821	0.15	2	8.235
ESC power and ground	14AWG	0.00828	14.42	0.15	2	516.4

Cable gauges were selected based on the current passing through them. 28AWG is only able to support up to 1 A, while lower gauge numbers can hold increasingly high currents. Additionally, the cables used for data buses are not represented here, but they were all set at a 28AWG gauge since they carry a minimal amount of current. Cable length itself was decided by the location of components; ToF sensors can be located at the arms of the drone so a longer cable length was accounted for.

Given the computed power consumption and the characteristics of the battery chosen in Section 12.3.1, the drone's hover time can be estimated to be 24.7 min. It is important to note that this factors in the depth of discharge of the battery of 80 %, which reduces its usable capacity. As seen before, this shows a decrease from the value computed in the midterm, which could be due to the overall mass increase, increased modeled interference between the drone and the propellers, or both. Therefore, if the operational time of 30 min from **CRG-POW-01** is still to be met with the current configuration, the possibility of hot-swapping the battery will have to be explored further.

12.4. Verification and Validation

In this chapter, the hardware components from other subsystems were integrated into a complete power system. Since most of these components have already been used and discussed in previous sections, and all of them contain manufacturer test data, which has already been validated, no major verification procedures remain to be done in the assumptions and models used. Nonetheless, the verification of the requirements can be done, and these are presented in Table 12.8 alongside a short description of the verification method conducted.

Table 12.8: Verification of Electronics subsystems

Identifier	Verification Method	Compliance
CRG-POW-01	A single battery cannot achieve an operational time of 30 min, however, the possibility of hot-swapping can still be explored.	TBC
CRG-POW-03	Analysis, the battery is equipped with a Management System, so the drone is able to head back to charging once low battery is detected.	✓
CRG-POW-04	Analysis, the battery is certified to charge at 5C, so it can charge in 12 minutes.	✓
CRG-POW-05	Analysis, said power distribution system has been designed in this section.	✓
CRG-POW-06	Analysis, the cables and components used are certified to withstand these power requirements.	✓
CRG-POW-07	Analysis, the cables and components used are certified to withstand these power requirements.	✓
CRG-POW-09	Analysis, the cables and components used are certified to withstand these power requirements.	✓
CRG-POW-10	Analysis, the cables and components used are certified to withstand these power requirements.	✓
CRG-POW-11	Inspection, an LED has been included in the design.	✓
CRG-POW-12	The plan of contingency for the loss of power has not yet been devised, but it has been considered in the design.	TBC
CRG-POW-13	Analysis, an EOL plan for the battery has been devised and will be followed.	✓
CRG-POW-14	Analysis, voltage converters have been implemented to support interfacing of the components.	✓
CRG-POW-15	Analysis, the cables and converters used are certified to withstand current requirements of the components.	✓
CRG-POW-16	Analysis, general pins were used as often as possible to ensure the compatibility of the components. All components fit mechanically.	✓
CRG-POW-17	Analysis, the battery is equipped with a Management System, so the drone is able to measure its charge and stay within the depth of discharge.	✓
CRG-POW-18	Inspection, the battery is not hardwired into the circuit in any way at this stage, elaborated upon in 13.	✓

12.5. Future Recommendations

The department of Electronics tried to adhere to sustainability principles as closely as possible. Nonetheless, given the use of a battery as a power source, and recyclability options not being widely adapted due to high costs, potential environmental threats and logistic challenges could arise in the immediate future. Further work shall focus on deeper analysis into the battery, to ensure that the most viable option is selected. In addition, it was discovered at a later stage that some extra components would be necessary, initially unforeseen. A recommendation for future iterations would be to closely examine electrical connections and instrument compatibility, to minimize the necessity of intermediate components such as voltage converters. It should be noted that the first iteration of the drone hardware has focused on primary functions and not supporting elements like cooling systems. These systems have to be reiterated and implemented properly to ensure that the hardware does not overheat, especially the ESCs.

13: Structures and Materials

This chapter presents the development of the drone structural frame. After establishing the subsystem goals and follow-up functions in Section 13.1, the requirements and associated risks are listed in Section 13.2. Section 13.3 then describes the design process of the drone frame. Verification of the drone arms is executed in Section 13.4.1, followed by validation of both the arms and the whole drone in Section 13.4.3. Then, concluding remarks and future recommendations are presented in Section 13.5.

13.1. Goals and Functions

One of the main components of a quadcopter drone is the structural frame that houses the payload and connects it to the four propeller motors. The frame needs to shelter all hardware components while remaining rigid under flight loads. Additionally, the structure must align with the sustainability goals of the project and seamlessly connect the subsystems into a single autonomous vehicle. Therefore, the main goal of the structures and materials department is to:

Develop a lightweight, recycled structural frame that supports all subsystems and maintains structural integrity under operational stresses.

Meeting this goal is governed by several functions that shall be performed. Most of them are linked to the conditions and loads the drone must withstand while stationary, in flight, and in case of impact. With this in mind, the structural design process entails building upon a chosen topology and developing a complete drone frame architecture, while keeping in mind the desired functionalities. The main functions of the structural frame are as follows:

1. Provide connection points for hardware components
2. Withstand operational loads
3. Protect the payload from the warehouse environment

These functions shall be reassessed in a more tangible manner that allows the team to track the design process in more detail. This is done by formulating constraints on what the system shall do in the form of a list of requirements.

13.2. Requirements and Risks

From the global objectives of the product and based on interactions with other subsystems, a list of subsystem requirements can be postulated for the Structures and Materials subsystem. This is given in Table 13.1.

Table 13.1: Requirements for Structures and Materials subsystem

ID	Structural Requirement	Risk ID
CRG-STR-03	The frame shall maintain its structural integrity within the temperature range 13-27 degrees Celsius.	RSK-STR-01
CRG-STR-04	The frame shall maintain its structural integrity within the humidity range 30-50% RH.	RSK-STR-02
CRG-STR-06	The frame shall protect the payload from water exposure at all times during the mission.	RSK-STR-02
CRG-STR-07	The frame shall provide the opportunity to facilitate battery hot swapping.	RSK-STR-03
CRG-STR-08	The frame shall accommodate software and electrical connections.	RSK-STR-04
CRG-STR-09	The frame shall not fail under design loads with an applied safety factor of 1.5.	RSK-STR-05
CRG-STR-10	The frame shall be manufactured from at least 70% recycled materials by volume.	RSK-MAT-01
CRG-STR-11	The frame shall have a mass of at most 300 g.	RSK-STR-11
CRG-STR-13	The frame shall facilitate a configuration that allows for all subsystem components to function within normal operating conditions.	RSK-STR-07
CRG-STR-14	The frame shall be designed for ease of assembly and disassembly.	RSK-STR-08
CRG-STR-15	The frame shall be designed such that damaged components can be replaced.	RSK-STR-09
CRG-STR-16	The frame shall ensure all components of the drone remain within horizontal dimensions of 50x50 cm.	RSK-STR-10
CRG-STR-17	The frame shall withstand vibrations such that the payload can operate with an accuracy specified in REQ-MIS-TECH-01.	RSK-STR-11
CRG-STR-18	The frame shall withstand vibrations such that resonance does not occur with the motor frequencies.	RSK-STR-18
CRG-STR-20	The frame shall accommodate the selected battery.	RSK-STR-08
CRG-STR-22	The frame shall maintain its structural integrity within the operational temperature range of the selected battery.	RSK-STR-01
CRG-STR-24	The frame shall accommodate the selected navigation system.	RSK-STR-13
CRG-STR-25	The frame shall accommodate the selected controller system.	RSK-STR-13
CRG-STR-26	The frame shall accommodate the selected communications system.	RSK-STR-14
CRG-STR-27	The frame shall accommodate the selected volume measurement system.	RSK-STR-13
CRG-STR-28	The frame shall allow for a maximum main arm deflection of 1 mm.	RSK-STR-05
CRG-STR-29	The frame shall allow for a maximum von Mises stress of the selected material's yield strength with an applied safety factor of 1.5.	RSK-STR-05
CRG-STR-30	The frame shall facilitate the protection of warehouse workers from the drone.	RSK-STR-15
CRG-STR-31	The frame shall facilitate the protection of the warehouse environment from the drone.	RSK-STR-15

Coupled with the requirements of the subsystem, a list of associated risks is formulated to account for the dangers of the structural design of the drone. The risks that have a both high likelihood and high severity ratings are identified and mitigation strategies are devised accordingly. These risks, along with their mitigation methods, are presented in Table 6.2.

Table 13.2: Risks for Structures and Materials Subsystem Requirements

Risk ID	Structural Risk	Likelihood	Severity	Mitigation
RSK-STR-01	Thermal integrity failure	2	4	Implement thermal protection materials and monitor operating temperatures. Monitor battery temperature and implement cooling systems if needed.
RSK-STR-02	Humidity impact and water damage	3	3	Seal frame joints and use moisture-resistant materials, regularly test for leaks.
RSK-STR-03	Battery hot swap failure	2	4	Design intuitive battery swapping mechanisms and provide user training.
RSK-STR-04	Connection issues or subsystem malfunctions	3	4	Use reliable connectors and perform regular maintenance checks.
RSK-STR-05	Structural failure under load or excessive stresses/deflections	2	5	Apply rigorous testing and use high-quality materials. Use stiff materials and conduct structural analysis.
RSK-STR-06	Excessive weight	3	4	Optimize design and material usage to reduce weight.
RSK-STR-07	Subsystem malfunction	3	4	Ensure proper integration and compatibility of all subsystems.
RSK-STR-08	Assembly/disassembly difficulties	3	3	Design for ease of access and provide clear assembly instructions.
RSK-STR-09	Component replacement issues	3	3	Use modular components and maintain an inventory of spare parts.
RSK-STR-10	Dimension exceedance	2	4	Conduct precise measurements and stay within tolerances.
RSK-STR-11	Vibrational effects on payload	3	4	Use vibration dampening materials and design for vibration control.
RSK-STR-12	Resonance with motor frequencies	3	4	Conduct frequency analysis and use dampers or isolators.
RSK-STR-13	Hardware system integration	3	3	Ensure proper connections and conduct system tests.
RSK-STR-14	Communications system failure	3	4	Use reliable communication components and perform regular tests.
RSK-STR-15	The drone produces damage to the environment	2	4	Provide worker training, ensure environmental safety protocols and include snappable elements.
RSK-MAT-01	Insufficient use of recycled materials	2	3	Verify recycled material content and work with certified suppliers.

13.3. Design Process

The drone frame design begins with analysis of provisional materials and corresponding manufacturing processes, explained in Section 13.3.1 and Section 13.3.2 respectively. Next, the structural design process is elaborated upon in Section 13.3.3.

13.3.1. Material Selection

To select a material appropriate for Cargonaut's structural design, a literature study is first performed. Within this study, potential drone frame materials are researched. Moreover, high emphasis is placed on gaining knowledge regarding the sustainable aspects of the provisional materials. This emphasis stems from the following structural requirement that flowed down from a main stakeholder requirement:

CRG-STR-10: *The frame shall be manufactured from at least 70% recycled materials by volume.*

In the selection of a material for Cargonaut's frame, the following material characteristics are determined to be the main criteria:

- **Density,** ρ: The density of a material, to an extent, determines the size of components within a drone frame. Their sizes should be minimized to optimize the use of the given topological sizing contraints. For example, Material 1 and Material 2 have the same specific stiffness and tensile strength. However, Material 1 allows for an arm frame with a 1 mm thickness, while Material 2 requires an arm frame thickness of 1 cm. The first material would then be preferred. While both arms would be equally heavy and strong, Material 1 would ensure the frame remains more limited in size.
- **Specific stiffness**: This is the material's Young's Modulus (E) divided by its density (ρ). This characteristic is a leading criterion, as it will give a clear indication of how stiff a material is relative to its weight. Stiffness is necessary for the structural design, as Cargonaut should be able to withstand certain forces that will be further specified later on in this section. While a higher stiffness is better to resist bending, a lower density is preferred to minimize mass and ensure efficient flight. Lightweight materials are also considered more sustainable, as they enable more efficient power consumption. Thus, the specific stiffness should be maximized.
- **Specific tensile strength**: This is defined as the material's tensile yield strength (σ_y) divided by its density (ρ). Tensile yield strength is important as it indicates at what stress a material will deform plastically. Low

yield strength might mean, for example, that a material is brittle, i.e. it does not experience much elastic deformation before yielding. Additionally, this parameter is linked to the robustness of the frame, which is another sustainable element that will enhance the frame's life span as it ensures higher durability. To conclude, since the highest possible strength is preferred, this parameter should be maximized.

- **Sustainability**: This criterion is of high priority in the material selection process. As per **CRG-STR-10**, 70% or more of the used material should be recycled. Additionally, the EOL frame recyclability and the sustainable impact of the available manufacturing processes for the material are also important factors to consider in the choice of material. To comply with **CRG-STR-10**, it is key to consider the feasibility of obtaining each material in a recycled form.
- **Cost**: This material characteristic will heavily depend on the source of the recycled material. Nonetheless, the frame structure should remain within the budgeted development cost of €120, as stated in Section 5.2.3.

From literature research, it was found that the following materials are most commonly used for building drones[1]: carbon fiber, fiber glass, Kevlar, aluminum, wood, and Acrylonitrile Butadiene Styrene (ABS). Plant-Based Bioplastic (PBB) and Wood-Plastic Composite (WPC) are added to this list of potential frame materials as literature suggests their great recyclability characteristics [73–75]. Specifically, their respective variations of "PLA 3D870" and "rHDPE57W40CA" are considered. The aforementioned material selection criteria can then be used to eliminate options accordingly. For the "sustainability" criterion, the characteristic of "recyclability" is used for simplicity, both from the perspective of obtaining the material in a recycled form and being able to discard it through a recycling process at EOL. Cost is not yet considered at this stage, as this would require further research into all the different options for obtaining recycled forms of the stated materials.

Fiber glass and aluminum are considered inferior options due to their high densities and worse structural properties than those of e.g. carbon fiber, with no significant difference in recyclability. Moreover, balsa wood is eliminated due to its bad recyclability and sensitivity to moisture, making it less durable than the other material options. After these eliminations, the final material options, shown in Table 13.3 remain. This table also considers potential sources from which the material may be obtained in recycled form.

Table 13.3: Final material options for Cargonaut frame

Material Type	**Density** [g/cm^3]	**Specific Yield Strength** [m^2/s^2]	**Specific Stiffness** [m^2/s^2]	**Recycled Form Source**
Carbon fiber	1.70	2.08	117.65 - 294.12	Waste scraps from bigger projects could be used. Next to that, commercial suppliers of recycled carbon fiber can be considered[2].
Kevlar	1.44	1.91	48.96 - 70.50	Can be recycled from 100% Kevlar products[3].
PBB	1.22	0.03	1.72	Organic: Made from natural plant sugars (mainly from field corn), that are converted into polymers[4].
WPC	0.93	0.03	2.58	97% recycled: Wood particles and polymer are recycled. Only the binding agent is most likely not recycled [73, 74].
ABS	1.10	0.05	2.00	Can be recycled from 100% ABS products[5].

The material characteristics and recyclability of the WPC material stand out, as the latter criterion indicates great compliance with the sustainability goals of Cargonaut. WPC also has better specific strength and stiffness characteristics compared to ABS and PBB, which are the other materials that are graded highly in terms of recyclability. Moreover, further research demonstrates that WPC would be easily obtainable. The only initial concern is that the specific yield strength and specific stiffness of WPC are considerably lower than those of carbon fiber and Kevlar.

The best material would be one that combines the strength and stiffness of carbon fiber with the density and recyclability of WPC. Thus, a new material option arises: WCFC. As literature explains, different composition ratios of this material have been created and tested. Furthermore, the composition that would be most appropriate for the drone frame is the following: 40% Poplar wood sawdust, 9% short carbon fibers, 48% polypropylene thermoplastic, and 3% Maleic Anhydride-graded Polypropylene (MAPP) binding agent [76, 77]. This material has a specific stiffness of 3.63×10^{-6} m^2/s^2, a specific tensile strength of 0.047×10^{-6} m^2/s^2, and a density of 1.1 g/cm^3. These properties will be verified and validated through structural analysis in Section 13.4.1 to ensure

[1] Material Selection for Unmanned Aerial Vehicles (UAVs) Wings Using Ashby Indices Integrated with Grey Relation Analysis Approach Based on Weighted Entropy for Ranking https://www.joe.uobaghdad.edu.iq/index.php/main/article/view/1855/1575. Accessed 14 May 2024.

[2] The state of recycled carbon fiber: https://www.compositesworld.com/articles/the-state-of-recycled-carbon-fiber. Accessed May 17 2024.

[3] Recycling of aramid fiber: https://nortefios.com.br/recycling-of-aramid-fiber/. Accessed May 17 2024.

[4] ISCC 100% commitment to sustainable biopolymers: https://www.iscc-system.org/news/100-commitment-to-sustainable-biopolymers/. Accessed May 17 2024.

[5] ABS:https://www.vandenrecycling.com/en/what-we-do/buy-and-sell-plastic/abs/. Accessed May 17 2024.

they are satisfactory. Furthermore, WCFC can be obtained by purchasing the different composition elements separately to then be combined in the frame production process. Overall, the material can be up to 97% recycled. The short carbon fibers can be bought off-the-shelf in recycled form or obtained from production scraps. These recycled scraps maintain comparable properties to virgin carbon fiber [78]. The Poplar wood sawdust can likewise be obtained sustainably, as there is an extreme abundance of Poplar trees in places such as Canada. There, due to how quickly the tree species grows, it is very often used for production purposes, thereby creating reusable waste scraps [79, 80]. Lastly, polypropylene is a fully recyclable material and can therefore be obtained off-the-shelf in recycled form[6]. WCFC is thus selected as the final material for the drone frame.

13.3.2. Manufacturing Process

To produce Cargonaut's frame using the selected WCFC material, injection molding would be the most appropriate process [81]. Injection molding is liquid phase processing method that uses a mold and a press. Based on the design of the Cargonaut frame that will be explained in the next few sections, seven molds are determined to be needed. These molds are reusable, thus inferring a one-time cost and minimizing waste. Additionally, labor hours are low, reducing the production cost per part. The polypropylene within the WCFC material is melted down to the appropriate temperature, after which the Poplar wood fibers and short carbon fibers are mixed in, along with the MAPP binding agent. Thee material is then injected into the mold, which is usually made of two parts held together with a press. Finally, the part is cooled and released from the mold. Overall, the injection molding process is determined to be the most sustainable option, as it produces minimal material waste. Furthermore, it is a fast process and does not consume much energy, unlike e.g. 3D printing [82].

13.3.3. Structural Design Process

Before starting the drone frame structural design process, a distinction must be made between two related concepts. In this report, **topology** is defined to be the leading 2D geometry of the frame, while **shape** is interpreted as the specific 3D characteristics of the frame within the same topology. The different possible topology configurations can be seen in Figure 13.1, which relates typical payload housing sizes to drone arm topological layouts. The preliminary design phase narrowed these options down to the square H, true X, and X box frames based on requirements from other subsystems. Beyond choosing a final topology configuration, three other explicit design choices are left to be made in the detailed design phase: unibody versus modular frame, propeller guards versus no propeller guards, and arm cross-sectional shape. The final topology configuration is chosen using justifications from these other design choices.

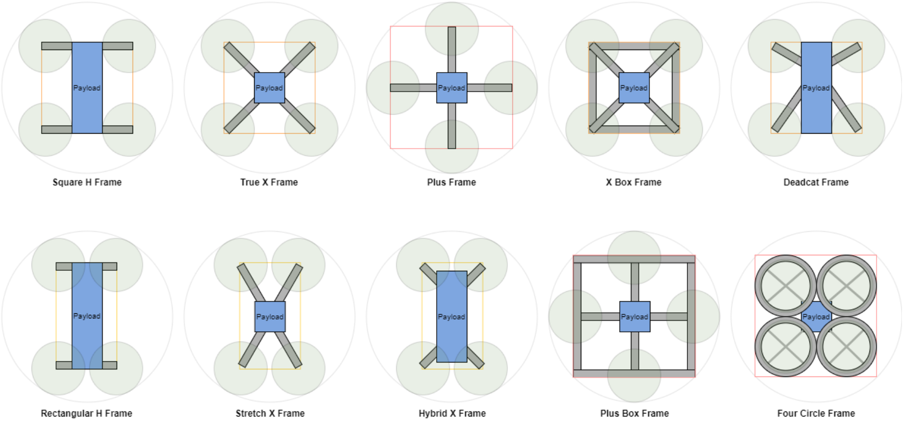


Figure 13.1: Conventional drone frame topologies

Detailed Design Options

First, the frame will either adopt a unibody or a modular design. The former refers to a frame manufactured entirely as a single unit, while the latter entails a frame with detachable arms. In order to comply with **CRG-STR-14**, a modular frame design is selected, such that the frame can be easily assembled and disassembled. Next, it must be noted that Cargonaut will operate mostly within a ground-handling warehouse, which is a closed, indoor environment. Human staff will also likely be working alongside the drones, and their safety must be prioritized. Thus, to comply with **REQ-STR-30**, propeller guards will be used to ensure the workers, cargo boxes, and warehouse environment are not harmed by the propeller blades. This inherently means the propeller blades themselves will be protected from the environment as well, which will prolong their operational use lifetime, mitigating the likelihood of **RSK-STR-09**.

[6] Recycling Polypropylene: How To Recycle PP The Benefits https://www.palmetto-industries.com/recycling-polypropylene/. Accessed 18 June 2024.

The last of the three additional design options is the cross-sectional shape of the drone arms, which is one of the more critical decisions in terms of ensuring structural integrity and compliance with **CRG-STR-28**. The following options are considered: square, rectangular, circular, and I-beam cross-sections. To determine the optimal shape, the loads that the drone will undergo must be identified. There are three basic types of forces a structure may experience: axial, shear, and bending. Since axial stresses depend entirely on the cross-sectional area of the structure, and because the trade-off between the cross-sectional shapes will assume a relatively similar cross-sectional area between them, axial loads are neglected for now. Additionally, if the arm is modeled as a cantilever beam, its deflection due to shear is assumed to be negligible compared to that of bending. Like with axial stresses, shear stresses also depend on cross-sectional area. Therefore, bending loads are prioritized for the trade-off. In particular, vertical bending due to propeller thrust and drone weight will likely be the critical load case, whereas lateral bending due to aerodynamic forces and propeller reaction bending moments will likely have much lower magnitudes. Thus, a high moment of inertia in the vertical direction is vital, which is provided by the rectangular and I-beam cross-sections.

Aside from bending forces, the arm's ability to withstand vibrational loads is also necessary due to vibrations that will emanate from the motors during flight. High moments of inertia are preferred for this as well. Torsional forces due to in-flight maneuvering are deemed negligible compared to the magnitude of the expected bending loads, meaning a circular cross-section is not necessarily required, although this should be verified in the future. Furthermore, the two most important factors for the feasibility of the drone structure are ease of manufacturing and ease of assembly. As stated in Section 13.3.1, the chosen material calls for the drone frame to be injection molded. Given the similar levels of complexity between the cross-sectional shapes, assuming the hollow options have open ends such that injection molding can indeed take place, the ease of manufacturing is relatively similar for each option. Consequently, the critical feasibility factor becomes the ease of assembly. The circular cross-section requires additional, more complex structural components to transition between the curved surface of the arm to the flat surfaces of the motor mount and connection to the payload housing. The already-flat surfaces of the other three cross-sectional shapes are thus preferred in this regard. With these factors in mind, the main contenders are limited to the square, rectangular, and I-beam cross-sections. Although lateral loads will likely have much smaller magnitudes compared to vertical ones, meaning a square cross-section is not necessarily required, they are not negligible. A rectangular arm will more comfortably withstand lateral loads due to its generally higher lateral moment of inertia compared to an I-beam arm. In addition to this, the hollow nature of the rectangular cross-section would allow for wiring to run through the inside of the arm, which is a safer and more protective option compared to having the wires wrap around the outside of an I-beam arm. A rectangular cross-sectional shape is therefore chosen for the design of the drone frame arms.

Initial Frame Sizing

To provide a starting point for the structural frame design, an initial 3D sizing is performed using TinkerCAD[7]. The outer dimensions of the hardware components that belong in the payload housing area are represented by rectangular prisms, and each one is placed relative to one another with some margin in accordance with their individual positioning requirements. A preliminary payload housing geometry, which has dimensions of 15x11x11 cm, is then created to encompass these components. This is then placed in the center of a 50x50 cm grid to represent the maximum allowable dimensions as specified in **CRG-STR-15**. The 7 inch propellers just fit into the grid while still maintaining a quarter diameter spacing to comply with the aforementioned requirement. If a straight line is drawn between the front two propellers, the payload housing does not reach it by a significant distance, meaning extra material is necessary to allow for the possibility of using the square H topological configuration. This would increase the mass and cost of the drone frame, thereby justifying the elimination of this option. The remaining topologies are the true X and X box frames. However, the X box topology option is eliminated, since its main benefit is its ability to protect the payload. This becomes redundant with the propeller guards, and the extra weight is rendered unnecessary. Thus, the true X topology is chosen for the drone frame.

The propellers are then connected to the payload housing in an X configuration using arms with rectangular cross-sections. With the propeller positioning and payload housing dimensions currently fixed, the length of the arms turns out to be around 10 cm, while their width and height are sized with respect to the payload housing and estimated to be 1.3 and 3.3 cm respectively. Due to uncertainties in the payload components at this stage of the design process, the simplest option for the landing gear is to place them beneath the propellers and motor mounts, thereby integrating them into the drone arms. This also provides higher stability and lower risk of tipping over given the large distance to the drone's center of gravity. The integrated landing gear is used by several other drones of similar size, including the DJI Mini series and the Fimi X8 Mini drone[8]. Preliminary dimensioning yields landing gear of height 8 cm.

[7] TinkerCAD: https://www.tinkercad.com/. Accessed 27 May 2024.

[8] Attack of the Mini Drones: https://www.premiumbeat.com/blog/mini-drones-more-value/. Accessed 18 June 2024.

Detailed Design Process

The detailed design process will be performed using an integrated approach, whereby verification and validation guide the design and are performed throughout. The main driving requirements for the structural frame are **CRG-STR-09, CRG-STR-28, CRG-STR-29**, which enforce a safety factor of 1.5 on the operational loads, a maximum arm deflection of 1 mm, and another safety factor of 1.5 on the operational stresses. The design process is then split into two parts. To begin with, the drone arm will be designed, verified, and validated using the chosen material. Once the requirements are met for the individual arm, four of them will be attached to a more detailed payload housing to ensure global compliance with the structural requirements. Seven load cases are identified that will be used to test the design, which are described in Table 13.4. The first five load cases for both the arm and the whole drone are the ones for which the structure must comply with the requirements, so the 1.5 safety factor must be applied to each of them. The final two load cases will be used to determine the failure points and modes, so no safety factor is needed for those. The hover and maximum thrust load cases do not account for bending relief provided by the weight of the landing gear and other components that may be placed at the ends of the arms, therefore providing a relatively conservative estimate. The values that will be used throughout the design process are given in Table 13.5, which are taken from various other subsystems. Hover entails a thrust-to-weight ratio of $T/W = 1$.

Table 13.4: Load cases used to design drone structural frame

Part	Identifier	Title	Description
Arm	**LC-ARM-01**	Stationary	Quarter of full drone weight applied at arm base
	LC-ARM-02	Hover	Quarter of hover thrust subtracted by propeller weight and motor weight applied at motor mount
	LC-ARM-03	Lateral forces	Maximum aerodynamic drag applied at vertical arm face and maximum reaction torque due to propeller rotation causing bending in same direction applied at motor mount
	LC-ARM-04	Maximum thrust	Quarter of maximum thrust subtracted by propeller weight and motor weight applied at motor mount
	LC-ARM-05	Landing	Quarter of landing impact force due to freefall from 20 cm approximated as static load applied at bottom of landing gear
	LC-ARM-06	Hard landing	Quarter of landing impact force due to freefall from 100 cm approximated as static load applied at bottom of landing gear
	LC-ARM-07	Front impact	Half of front impact force at maximum velocity approximated as static load applied at propeller guard
Whole Drone	**LC-WD-01**	Stationary	Full drone weight applied at center of payload housing
	LC-WD-02	Hover	Hover thrust subtracted by propeller weight and motor weight applied at four motor mounts
	LC-WD-03	Lateral forces	Maximum aerodynamic drag applied at four vertical arm faces and maximum reaction torque due to propeller rotation applied at four motor mounts
	LC-WD-04	Maximum thrust	Maximum thrust subtracted by propeller weight and motor weight applied at four motor mounts
	LC-WD-05	Landing	Landing impact force due to freefall from 20 cm approximated as static load applied at bottom of four landing gears
	LC-WD-06	Hard landing	Landing impact force due to freefall from 100 cm approximated as static load at bottom of four landing gears
	LC-WD-07	Front impact	Front impact force at maximum velocity approximated as static load applied at front two propeller guards

Table 13.5: Values used for structural frame design

Symbol	Description	Value	Unit	Symbol	Description	Value	Unit
ρ	WCFC density	1100	kg/m^3	T_{hov_p}	Hover thrust per propeller	3.434	N
E	WCFC Young's modulus	4	GPa	$(T/W)_{\text{max}}$	Maximum T/W	4	-
σ_y	WCFC yield strength	52	MPa	T_{max}	Total maximum thrust	54.936	N
ν	WCFC Poisson's ratio	0.29	-	T_{max_p}	Maximum thrust per propeller	13.734	N
SF	Required safety factor	1.5	-	m_{prop}	Propeller mass	0.011	kg
δ_{allow}	Maximum allowable deflection	1	mm	m_{motor}	Motor mass	0.035	kg
σ_{allow}	Maximum allowable stress	35	MPa	Q_{hov}	Hover propeller torque	0.029	Nm
g	Acceleration due to gravity	9.81	m/s^2	Q_{max}	Maximum propeller torque	0.135	Nm
ρ_{air}	Air density	1.225	kg/m^3	ω_{hov}	Hover propeller angular velocity	9600	rpm
v_{max}	Maximum drone speed	10	m/s	ω_{max}	Maximum propeller angular velocity	22000	rpm
m_{drone}	Total drone mass	1.4	kg	f_{hov}	Hover propeller frequency	160	Hz
T_{hov}	Total hover thrust	13.734	N	f_{max}	Maximum propeller frequency	366.667	Hz

13.4. Verification and Validation

The drone arm will first be sized, iterated upon, and verified using simplified analytical solutions and various basic physics formulas. These solutions are described in Section 13.4.1. The design will then be modeled in CATIA 3DEXPERIENCE, after which the Ansys Mechanical program will be used to perform Finite Element Analysis (FEA) on the arm. Frequency analysis will also be performed using the CATIA Structural Scenario Creation app, whereby a frequency step will be applied and the structure analyzed for the first six eigenfrequency modes. This validation process is detailed in Section 13.4.2. Finally, FEA will be performed on the whole drone to ensure maximum deflections and stresses are not exceeded, which is completed in Section 13.4.3.

13.4.1. Arm Verification

In order to verify the drone arm design, the arm is initially approximated as a cantilever beam. Three types of basic loadings can be applied on a cantilever beam for which standard solutions exist. These are depicted in Figure 13.2, where w is a load distributed along the length of the beam L, F is a point force applied at the end of the beam, and M is bending moment applied at the end of the beam. The equations for end deflection δ due to each of these load types are given in Equation 13.1, Equation 13.2, and Equation 13.3 respectively, where E is the Young's modulus of the material and I is the cross-sectional vertical moment of inertia [83].

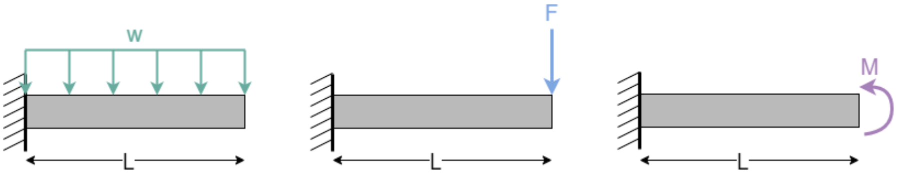


Figure 13.2: Cantilever beam with applied distributed load, point force, and moment

$$\delta = \frac{wL^4}{8EI} \quad (13.1) \qquad \delta = \frac{FL^3}{3EI} \quad (13.2) \qquad \delta = \frac{ML^2}{2EI} \quad (13.3)$$

The moment of inertia around the x-axis I_{xx} of a hollow rectangular cross-section is calculated using

$$I_{xx} = \frac{1}{12}\left(bh^3 - (b-2t)(h-2t)^3\right) \tag{13.4}$$

where b is the width of the cross-section, h is its height, and t is its thickness. In addition to finding the deflection of a cantilever beam due to an applied load, the maximum compressive or tensile stress due to a bending moment can also be calculated to ensure the requirement related to this parameter is met. The equation for this is

$$\sigma_{\text{max}} = \frac{My_{\text{max}}}{I_{xx}} \tag{13.5}$$

where y_{max} is the maximum distance to the neutral axis, which is the axis at which $\sigma = 0$. For a rectangular cross-section, $y_{\text{max}} = \frac{h}{2}$. The first step in proceeding with the structural design is to verify the chosen material is able to withstand operational loads while meeting the maximum deformation and stress requirements. To do so, Equation 13.2 will be rearranged to solve for E, where δ_{allow} is given by **REQ-STR-28** to be 1 mm and F follows from **LC-ARM-04**, which is chosen due to the likelihood of its more critical nature. This E_{req} is thus the minimum Young's modulus necessary to allow for a maximum deflection of 1 mm for this specific load case. Rearranging yields

$$E_{\text{req}} = \frac{FL^3}{3\delta_{\text{allow}}I}$$

For **LC-ARM-04**, $F = (T_{\text{max}_\text{p}} - (m_{\text{prop}} + m_{\text{motor}}) \times g) \times SF = 19.924$ N. L was previously found to be 100 mm. I_{xx} is calculated using the preliminary dimensions found earlier, where $b = 13$ mm, $h = 33$ mm, and $t = 1$ mm. This yields $I_{xx} = 11623.333$ mm^4. Solving for E_{req} leads to $E_{\text{req}} = 0.571$ GPa, which is less than a quarter of the Young's modulus of WCFC. Additionally, compliance with the maximum stress requirement will be evaluated by solving Equation 13.5, where for the same load case, $M = F \times L$ which yields $M = 1992.4$ Nmm. It follows from the current geometry that $y_{\text{max}} = \frac{h}{2} = 16.5$ mm. Thus, $\sigma_{\text{max}} = 2.828$ MPa, which is less than ten times the yield strength of WCFC. As a result of these first two verification steps, it is likely that the material chosen will be able to withstand all operational loads while remaining below the required maximum deflection and stress, thereby verifying its initial use as a structural material.

To continue with arm verification, the rest of the loads that will be applied to the structure must be quantified. Using parameters given in Table 13.5 and the aforementioned geometric characteristics, the load cases specified in Table 13.4 can be assigned numerical values using basic calculations. For the lateral load cases, **LC-ARM-03** and **LC-ARM-07**, I_{yy} is used instead of I_{xx}, and x_{max} instead of y_{max}. Using the same geometry, $I_{yy} = 2603.333$ mm^4 and $x_{\text{max}} = \frac{b}{2} = 6.5$ mm. To model aerodynamic drag, the following formula is used:

$$F_d = \frac{1}{2}\rho_{\text{air}} v^2 C_d A_d \tag{13.6}$$

where C_d is the drag coefficient and A_d is the reference area[9]. The reference area of the rectangular part of the arm that will experience drag is $A_d = L \times h = 3300$ mm^2, and it has an aspect ratio of $\frac{L}{h} = 3.03$, which will be rounded up to 5 to correspond with an approximate C_d of 1.20 [84]. Finally, the landing impact forces are calculated by assuming freefall from 20 cm for nominal landing and 100 cm for hard landing. To do this, the following basic physics equations are used:

$$PE = mgh_{\text{fall}} \tag{13.7}$$

$$KE = \frac{1}{2}mv_{\text{imp}}^2 \tag{13.8}$$

$$v_{\text{imp}} = \sqrt{2gh_{\text{fall}}} \tag{13.9}$$

where PE is potential energy, KE is kinetic energy, h_{fall} is the height from which the drone freefalls, and v_{imp} is the velocity just before it impacts the ground. Assuming all potential energy is converted into kinetic energy, Equation 13.7 and Equation 13.8 can be combined to obtain Equation 13.9. The nominal landing impact velocity v_{land} is thus calculated to be 1.981 m/s, and the hard landing impact velocity v_{hardland} 4.429 m/s. These impact velocities are converted into forces using

$$F_{\text{imp}} = \frac{m\Delta v_{\text{imp}}}{\Delta t_{\text{imp}}} \tag{13.10}$$

where impact time t_{imp} is assumed to be 0.02 seconds for **LC-ARM-05** and **LC-ARM-06**. For **LC-ARM-07**, v_{imp} is taken to be the maximum operational drone speed v_{max}, while its impact time is assumed to be longer at 0.5 seconds due to the more elastic nature of the propeller guards. The results of these calculations are quantified in Table 13.6. It should be noted that for **LC-ARM-01**, the arm is assumed to be clamped at the landing gear end instead of at the end where it is connected to the payload housing, the latter of which is the case for the rest of the loading scenarios. Also, the values for **LC-ARM-03** should be added up.

Table 13.6: Verification for initial arm geometry

Identifier	Force Type	Force Equation	Magnitude	Unit	Deflection [mm]	Max Stress [MPa]
LC-ARM-01	Point force	$F = m_{\text{drone}} \times g/4 \times SF$	5.150	N	0.037	0.731
LC-ARM-02	Point force	$F = (T_{\text{hov}_\text{p}} - (m_{\text{prop}} + m_{\text{motor}}) \times g) \times SF$	4.473	N	0.032	0.250
LC-ARM-03	Distributed load	$w = \frac{1}{2}\rho_{\text{air}} v_{\text{max}}^2 C_d A_d \times SF$	3.638	N/m	0.004	0.045
	Moment	$M = Q_{\text{max}} \times SF$	0.203	Nm	0.097	0.507
LC-ARM-04	Point force	$F = (T_{\text{max}_\text{p}} - (m_{\text{prop}} + m_{\text{motor}}) \times g) \times SF$	19.924	N	0.143	2.828
LC-ARM-05	Point force	$F = m_{\text{drone}} \times \Delta v_{\text{land}}/(4\Delta t_{\text{land}}) \times SF$	34.666	N	0.249	4.921
LC-ARM-06	Point force	$F = m_{\text{drone}} \times \Delta v_{\text{hardland}}/(4\Delta t_{\text{hardland}})$	77.515	N	0.556	11.004
LC-ARM-07	Point force	$F = m_{\text{drone}} \times \Delta v_{\text{imp}}/(2\Delta t_{\text{imp}})$	14.000	N	0.448	3.496

It can be seen that no deflection exceeds the maximum allowable deflection of 1 mm, nor does any stress exceed the maximum allowable stress of 35 MPa. Thus, the initial sizing of the arm is verified to comply with the requirements. That being said, these calculations are highly simplified representations of the model, meaning this verification step is to be treated only as a sanity check and preliminary justification for the arm sizing. Further validation will provide a more accurate portrayal of the seven load cases.

Detailed Arm Shape Design

The main length of the arm is now initially sized and verified. However, the motor mount, connection to the payload housing, and propeller guards still need to be designed. A 3D model of it is then started within CATIA 3DEXPERIENCE to create the detailed shape of the arm. The size of the motor is given by the Propulsion subsystem, around which a circular platform is created extending from the main arm length. The outer ring of this platform is then raised slightly to provide motor protection. As mentioned earlier, the landing gear will be integrated with the arms. Thus, an arc of thickness 3 mm is taken from the motor mount and extended downwards by 8 cm.

A propeller guard design is created such that it simply clips into the arm beside the motor mount, making use of two grooves to hold the guard in place vertically. It roughly follows the design of the DJI drone propeller guards, with three supporting connections from the clipping point to an arc-shaped piece of WCFC acting as the main protective component[10]. Furthermore, it is chosen to have a buffer component, known as the arm insert, between the main arm and the payload housing that is designed to fail before other structural components. This makes it such that only the arm insert needs to be replaced in case operational loads are exceeded significantly, compared to needing to replace the entire payload housing or the entire arm. Again, inspiration is vaguely taken from the DJI Mini drone, whereby the arm fits into a slot in the arm insert and is fastened using one M3 screw at the top and bottom[11]. It also features a hole at its center for wires to run through. The arm insert is attached to the

[9] NASA The Drag Equation: https://www.grc.nasa.gov/www/k-12/VirtualAero/BottleRocket/airplane/drageq.html. Accessed 18 June 2024.

[10] DJI Propeller Guards: https://www.dronekenner.nl/en/drone-accessories/propellers-and-guards/propeller-guards-en/50cal-dji-fpv-propeller-guards-en/. Accessed 17 June 2024.

[11] DJI Mini 2 SE Drone Teardown: https://www.fictiv.com/articles/dji-mini-2-se-drone-teardown. Accessed 19 June 2024.

payload housing using 4 M2 screws at its corners. The drone arm assembly and arm insert details are shown in Figure 13.3.

(a) Arm assembly

(b) Arm insert

Figure 13.3: Drone arm components

The material for the screws is chosen to be Aluminum 7075 T6 due to its strength being 90% that of steel while having a third of its weight[12]. WCFC is applied to the arm, integrated landing gear, and the arm insert. Fillets are used at most sharp corners to reduce the possibility of detrimental stress concentrations. However, the mass of the arm is already dangerously high at 75 grams, which multiplied by four leads to a mass of 300 grams for the arms alone, already reaching the mass budget without having any payload housing. To reduce this, the main length of the arm is reduced in width from 13 mm to 10 mm while still maintaining its height to keep a relatively high vertical moment of inertia. Smaller changes are made to reduce the amount of material used in the arm insert as well. Verification must then be performed on the new arm geometry to ensure the requirements are still complied with, especially for lateral forces given the reduced lateral moment of inertia. The results are compared with the old deflections and stresses in Table 13.7.

Table 13.7: Verification for modified arm geometry

Identifier	Magnitude	Unit	Old Deflection [mm]	New Deflection [mm]	Old Max Stress [MPa]	New Max Stress [MPa]
LC-ARM-01	5.150	N	0.037	0.043	0.731	0.842
LC-ARM-02	4.473	N	0.032	0.037	0.250	0.288
LC-ARM-03	3.638	N/m	0.004	0.008	0.045	0.064
	0.203	Nm	0.097	0.178	0.507	0.711
LC-ARM-04	19.924	N	0.143	0.165	2.828	3.259
LC-ARM-05	34.666	N	0.249	0.286	4.921	5.671
LC-ARM-06	77.515	N	0.556	0.640	11.004	12.680
LC-ARM-07	14.000	N	0.448	0.817	3.496	4.904

Once again, although the deflections and stresses have increased slightly due to the reduced moment of inertia, it can be seen that the deflections and stresses still comply with the requirements, justifying the modification. With the main arm components initialized in CATIA, the various components that will be placed in the payload housing area are imported as .stp files from online. They are then positioned relative to one another, after which it can be seen that with the battery placed on an upper level within the payload housing, there is space to position the depth camera such that it does not protrude from the bottom of the drone at all. As a result, the landing gear is reduced from 8 cm in height to 5.4 cm. These changes all result in a total drone arm mass of 60 grams.

13.4.2. Arm Validation

In order to validate the drone arm design, FEA will be conducted using the Ansys Mechanical program. Similar to the CFD program mentioned in Chapter 10, Ansys makes use of Finite Element Methods (FEM). This is the process by which a structure is divided into small elements connected by nodes, which form a mesh. Once boundary conditions are defined and loads are applied, Ansys formulates equations based on physical laws to find solutions at the mesh nodes[13]. Additionally, to ensure resonance does not occur within the drone arm, the frequency analysis tool within CATIA will be utilized to obtain the first six eigenfrequency modes of the arm.

Finite Element Analysis

The boundary conditions, along with the force types, magnitudes, and directions for each load case are specified in Table 13.8. The horizontal screw holes refer to the holes for the screws used to connect the arm insert to the payload housing. The model assumes these holes a fixed for most of the load cases, while the screws attaching the arm insert to the arm itself are given contact constraints to analyze their relative behavior. The propeller guard

[12] All About 7075 Aluminum Alloy: https://www.xometry.com/resources/materials/7075-aluminum-alloy/. Accessed 18 June 2024.

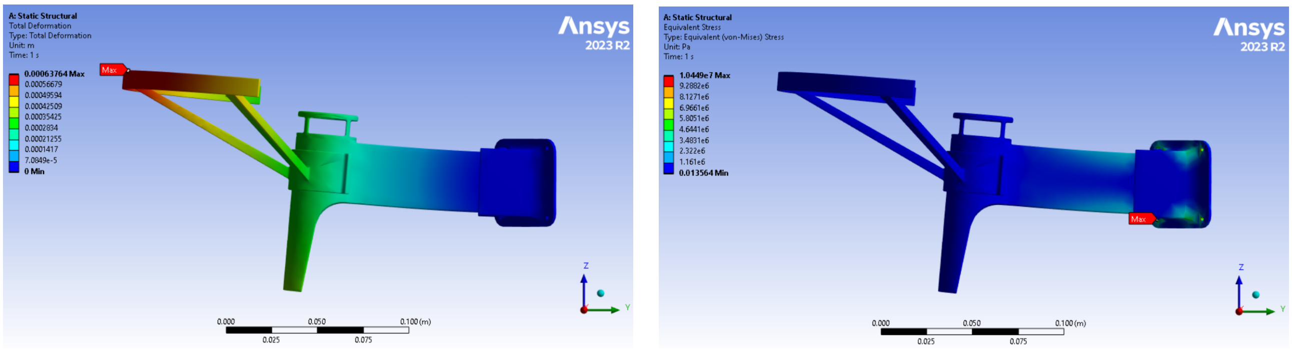
[13] Ansys: https://www.ansys.com/products/structures/ansys-mechanical. Accessed 18 June 2024.

is also constrained to the face of the arm it is attached to. The force magnitudes are rounded up to either the nearest whole number, tenth, or hundredth depending on the initial magnitude to provide more margin for error.

Table 13.8: Finite Element Analysis arm validation load case specifications

Identifier	Fixed Boundary	Force Type	Force Location	Force Magnitude	Force Magnitude Rounded	Force Direction (x, y, z)
LC-ARM-01	Bottom of landing gear	Point force	4 horizontal screw holes	5.150 N	5.2 N	(0, 0, -F)
LC-ARM-02	4 horizontal screw holes	Point force	Motor mount	4.473 N	5 N	(0, 0, F)
LC-ARM-03	4 horizontal screw holes	Distributed load	45° to front arm face	3.638 N/m	4 N/m	$(\frac{\sqrt{2}}{2}$ w, $\frac{\sqrt{2}}{2}$ w, 0)
		Moment	Motor mount	0.203 Nm	0.21 Nm	(0, 0, M)
LC-ARM-04	4 horizontal screw holes	Point force	Motor mount	19.924 N	20 N	(0, 0, F)
LC-ARM-05	4 horizontal screw holes	Point force	Bottom of landing gear	34.666 N	35 N	(0, 0, F)
LC-ARM-06	4 horizontal screw holes	Point force	Bottom of landing gear	77.515 N	80 N	(0, 0, F)
LC-ARM-07	4 horizontal screw holes	Point force	45° to propeller guard face	14.000 N	14 N	$(\frac{\sqrt{2}}{2}$ F, $\frac{\sqrt{2}}{2}$ F, 0)

FEA is performed on version 1 of the drone arm for the seven load cases, and results are depicted in Table 13.9. The total deformation and von Mises (equivalent) stress are analyzed to ensure compliance with **REQ-STR-29**. The location of maximum deformation is at the top of the propeller guard for all but the first load case, for which the maximum deformation is at the inner face of the arm insert. The location of maximum stress is at the inner corner of the bottom left horizontal screw hole for the second, fourth, fifth, and sixth load cases, the front bottom corner of the landing gear for the first, the inner front corner at the intersection between the arm and motor mount for the third, and the base of the middle propeller guard support for the seventh. For the last two load cases, this indicates that failure will occur either at the arm insert or at the propeller guards, both of which are replaceable components. An example of the FEA solutions for total deformation and von Mises stress is depicted in Figure 13.4.



(a) Total deformation

(b) Von Mises stress

Figure 13.4: Finite element solution of drone arm with 20 N thrust applied to motor mount (**LC-ARM-04**)

However, **LC-ARM-01** violates the 1 mm maximum deflection, meaning reinforcements must be added to the connection between the landing gear and the main arms length to reduce the deformation due to drone weight. This is first done by creating a smoother transition between the components to distribute stress more evenly. Iterations are then performed on the landing gear arc angle until a solution is converged that complies with both the maximum deflection and maximum stress requirements. The convergence process is visualized using red, orange, and yellow highlights in Table 13.9 to indicate a solution closer and closer to complying with the requirements. Based on these iterations, a final version is completed with additional small changes to the inner geometry of the arm to maintain a total arm mass of 60 grams. Although **LC-ARM-05** indicates a maximum deformation of 1.34 mm, this occurs at the top of the propeller guards, while **REQ-STR-28** enforces the 1 mm requirement on the arm itself. A probe can be used within Ansys to demonstrate that the maximum deflection of the arm itself is 7.9234E-04, which is less than 1 mm. The first and second versions of the drone arm are visualized in Figure 13.5.

Table 13.9: Finite Element Analysis arm validation results

Version	Identifier	Total Deformation		Von Mises Stress		
		Maximum [m]	Average [m]	Minimum [Pa]	Maximum [Pa]	Average [Pa]
V1	**LC-ARM-01**	2.47E-03	1.29E-03	0.29981	1.71E+07	2.45E+05
	LC-ARM-02	1.49E-04	7.47E-05	1.06E-02	2.97E+06	43967
	LC-ARM-03	9.51E-06	5.32E-06	2.69E-05	92375	1538.9
	LC-ARM-04	5.94E-04	2.99E-04	4.24E-02	1.19E+07	1.76E+05
	LC-ARM-05	1.24E-03	6.15E-04	0.19326	2.31E+07	3.84E+05
	LC-ARM-06	2.83E-03	1.40E-03	0.44187	5.27E+07	8.78E+05
	LC-ARM-07	9.85E-02	3.49E-02	369.57	4.48E+08	6.31E+06
V1.1	**LC-ARM-01**	2.46E-03	1.39E-03	0.11435	3.17E+07	2.07E+05
V1.2	**LC-ARM-01**	1.71E-03	9.52E-04	2.53E-02	5.16E+06	36928
V1.3	**LC-ARM-01**	1.32E-03	7.38E-04	1.35E-01	2.28E+07	1.68E+05
V1.4	**LC-ARM-01**	1.09E-03	5.98E-04	1.41E-01	1.57E+07	1.54E+05
V1.5	**LC-ARM-01**	9.54E-04	5.16E-04	1.33E-01	7.49E+07	1.80E+05
V1.6	**LC-ARM-01**	1.01E-03	5.50E-04	0.1448	1.47E+07	1.52E+05
V1.7	**LC-ARM-01**	9.78E-04	5.31E-04	0.1415	1.15E+07	1.51E+05
V2	**LC-ARM-01**	9.99E-04	5.54E-04	6.61E-02	1.29E+07	1.47E+05
	LC-ARM-02	1.59E-04	8.44E-05	3.39E-03	2.61E+06	49362
	LC-ARM-03	5.39E-04	2.91E-04	1.36E-02	1.74E+06	72772
	LC-ARM-04	6.38E-04	3.38E-04	1.36E-02	1.04E+07	1.97E+05
	LC-ARM-05	1.34E-03	6.96E-04	6.80E-02	2.04E+07	4.24E+05
	LC-ARM-06	3.05E-03	1.59E-03	0.1554	4.66E+07	9.69E+05
	LC-ARM-07	1.12E-01	4.66E-02	910.37	2.33E+08	7.36E+06

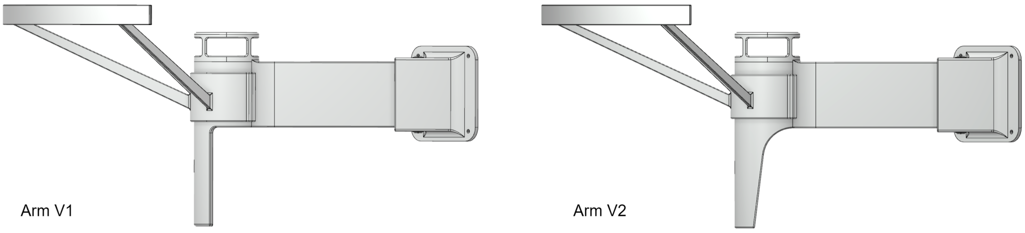


Figure 13.5: First and second versions of the drone arm

Frequency Analysis

Before finalizing the drone arm design, frequency analysis must be conducted to ensure resonance does not occur due to vibrations emanating from the motor and propeller. The propeller hover frequency is 160 Hz, while its maximum frequency is 366.667 Hz. With the fixed supports again being the four horizontal screw holes, the first six eigenfrequencies of the arm are listed in Table 13.10. Beyond the sixth mode, resonance is unlikely to be detrimental to the arm's structural integrity[14]. It can be seen that the arm eigenfrequencies are significantly lower than both the hover and maximum propeller frequencies, thereby significantly reducing the likelihood of resonance and ensuring compliance with **REQ-STR-18**. With this, the arm design is finalized.

Table 13.10: Eigenfrequencies of final drone arm design

Mode	Frequency [Hz]
1	16.592
2	39.335
3	51.300
4	98.489
5	115.591
6	118.461

13.4.3. Whole Drone Validation

Finally, validation must be performed on the entire drone structure to ensure the payload housing does not reveal any failure points under operational loads. The following components are finalized and added to the main payload housing within the drone frame: Raspberry Pi PCB stack (AI module on top of Raspberry Pi 5), Vision Processor D4 Board, ESC, D450 Depth Camera, battery, RGB PCB, and antenna PCB. The camera is required to point downward at roughly a 45° angle. The battery should have the opportunity to be hot-swapped, for which the decision is made to use a snap-fit and hinge connection on the rear cover of the payload housing. This will be split into an upper and lower cover, the former to access the battery and the latter to access the other PCBs. The detailed hot-swapping mechanisms are beyond the scope of the DSE. The battery casing should also leave some margin in case the battery expands during operations. The weight distribution of the components within the payload housing should be relatively symmetrical as well to ensure all the motors work roughly the same amount, further justifying the placement of the battery in the center of the housing above the other PCBs. The

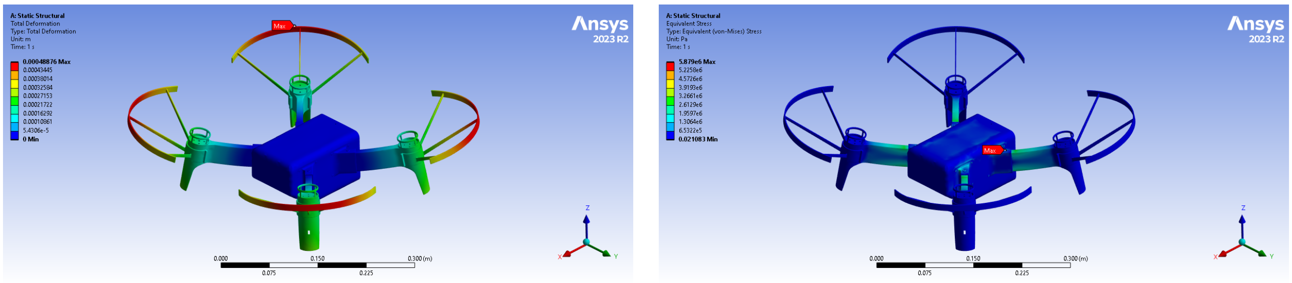
[14] Eigenfrequency Analysis: https://www.comsol.com/multiphysics/eigenfrequency-analysis. Accessed 18 June 2024.

PCBs themselves are fixed by creating bolt holes in the bottom of the structure and attaching stand-offs to them, atop which the PCBs are fastened. Since the battery is the longest component in the housing, a sloped face can be made below it such that the PCBs still fit inside, while the depth camera can be placed flush against the slope to eliminate the need for a bottom extrusion, as mentioned earlier. The entire payload housing is preliminarily enclosed in a structure of 1 mm thickness made of WCFC to protect the internal components from water damage. Additional internal cylinders with screw holes are created to attach the arm inserts to the payload housing, which are connected and attached to the walls beside the battery for more efficient load transfer.

Similar to the FEA performed for the arm, the FEA load case specifications for the whole drone simulation are given in Table 13.11. The forces are specified as total magnitudes, meaning they are distributed over the different force locations. The total deformation and von Mises stress results are visualized in Figure 13.6 and quantified in Table 13.12. The maximum deformation occurs at the propeller guards for all but the first load case, for which the maximum deformation occurs at the weight application point. The maximum stress occurs at the back left top right horizontal screw for the first load case, the back left top vertical screw for the second, fourth, and fifth load cases, the back right intersection between the arm and motor mount for the third load case, the back left right bottom corner of the propeller guard for the sixth, and the front right base of the middle propeller guard for the seventh. For the last two load cases, there is a large discrepancy between the stresses at the propeller guards and the stresses in the rest of the drone body, the latter of which experiences a maximum of 28 and 32.065 MPa for each load case respectively. To reiterate, the maximum allowable stress is set at 35 MPa. This again indicates that for the extreme load cases, the propeller guards will fail first, meaning they can be replaced and the drone can continue operations as usual. That being said, **LC-WD-05** once again has a maximum total deformation of 1.05 mm, which slightly exceeds the 1 mm requirement. However, as with the final version of the drone arm, this deformation occurs at the propeller guards, while the actual maximum arm deformation is 6.1415E-04 mm, which is again less than 1 mm. In general, it can be seen that arms within the whole drone deflect slightly less and experience less stress than the arms individually. This is likely a result of the additional support provided by the arm insert area in the payload housing structure. The final drone frame is thus depicted Figure 13.7, and it results in a mass of 381 grams. While this is above 300 g requirement specified in **CRG-STR-11**, further steps can be taken to reduce this mass, which will be described in Section 13.5. The requirements compliance matrix is shown Table 13.13. The final Cargonaut drone with all of its components is shown in Figure 13.8, and a labeled exploded view is depicted in Figure 13.9.

Table 13.11: Finite Element Analysis whole drone validation load case specifications

Identifier	Fixed Boundary	Force Type	Force Location	Force Magnitude	Force Magnitude Rounded	Force Direction (x, y, z)
LC-WD-01	Bottom of 4 landing gear	Point force	Drone center plate	20.601 N	21 N	(0, 0, -F)
LC-WD-02	Drone center plate	4 point forces	4 motor mounts	17.893 N	20 N	(0, 0, F)
LC-WD-03	Drone center plate	4 distributed loads	45° to 4 front arm faces	14.553 N/m	20 N/m	(-F, 0, 0)
		4 moments	4 motor mounts	0.810 Nm	0.84 Nm	(0, 0, M)
LC-WD-04	Drone center plate	4 point forces	4 motor mounts	79.696 N	80 N	(0, 0, F)
LC-WD-05	Drone center plate	4 point forces	Bottom of 4 landing gear	138.664 N	140 N	(0, 0, F)
LC-WD-06	Drone center plate	4 point forces	Bottom of 4 landing gear	310.061 N	311 N	(0, 0, F)
LC-WD-07	Drone center plate	2 point forces	45° to 2 propeller guard faces	28.000 N	30 N	(-F, 0, 0)



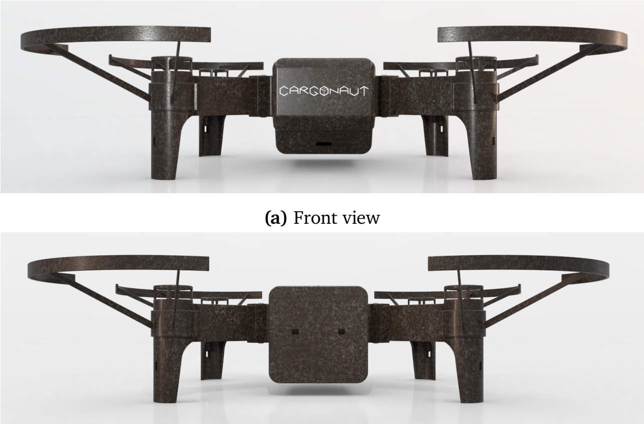
(a) Total deformation

(b) Von Mises stress

Figure 13.6: Finite element solution of whole drone with 80 N thrust distributed across four motor mounts (**LC-WD-04**)

Table 13.12: Finite Element Analysis whole drone validation results

Identifier	Total Deformation		Von Mises Stress		
	Maximum [m]	Average [m]	Minimum [Pa]	Maximum [Pa]	Average [Pa]
LC-WD-01	5.91E-05	2.19E-05	2.87E-02	1.43E+06	4.24E+04
LC-WD-02	1.22E-04	6.57E-05	5.27E-03	1.47E+06	3.62E+04
LC-WD-03	5.09E-04	2.82E-04	1.06E-02	1.79E+06	54392
LC-WD-04	4.89E-04	2.63E-04	2.11E-02	5.88E+06	1.45E+05
LC-WD-05	1.05E-03	5.52E-04	9.41E-02	1.15E+07	3.13E+05
LC-WD-06	2.34E-03	1.23E-03	0.20901	2.52E+08	7.17E+05
LC-WD-07	1.19E-01	2.83E-02	2.03E-05	2.89E+08	3.23E+06



(a) Front view

(b) Back view

Figure 13.7: Final drone frame

Table 13.13: Compliance matrix for Structures and Materials subsystem

ID	Verification Method	Compliance
CRG-STR-03	Analysis of material properties.	✓
CRG-STR-04	Analysis of material properties.	✓
CRG-STR-06	Analysis of possible leakage points within structure.	✓
CRG-STR-07	Analysis of ease of battery replacement.	✓
CRG-STR-08	Analysis of electronics requirements.	✓
CRG-STR-09	Analysis of frame under operational loads.	✓
CRG-STR-10	Analysis of literature provided by material suppliers.	✓
CRG-STR-11	Inspection of final drone frame mass.	TBC
CRG-STR-13	Analysis of subsystem component requirement compliance.	✓
CRG-STR-14	Analysis of frame design ease of assembly.	✓
CRG-STR-15	Analysis of frame modularity.	✓
CRG-STR-16	Inspection of final drone dimensions.	✓
CRG-STR-17	Analysis of payload performance under motor frequency vibrations.	✓
CRG-STR-18	Analysis of frame eigenfrequencies.	✓
CRG-STR-20	Analysis of battery requirements.	✓
CRG-STR-22	Analysis of material properties.	✓
CRG-STR-24	Analysis of navigation requirements.	✓
CRG-STR-25	Analysis of controls requirements.	✓
CRG-STR-26	Analysis of communications requirements.	✓
CRG-STR-27	Analysis of payload requirements.	✓
CRG-STR-28	Analysis of arm deflections under operational loads.	✓
CRG-STR-29	Analysis of the frame stresses under operational loads.	✓
CRG-STR-30	Inspection and analysis of propeller guard usage.	✓
CRG-STR-31	Inspection and analysis of propeller guard usage.	✓

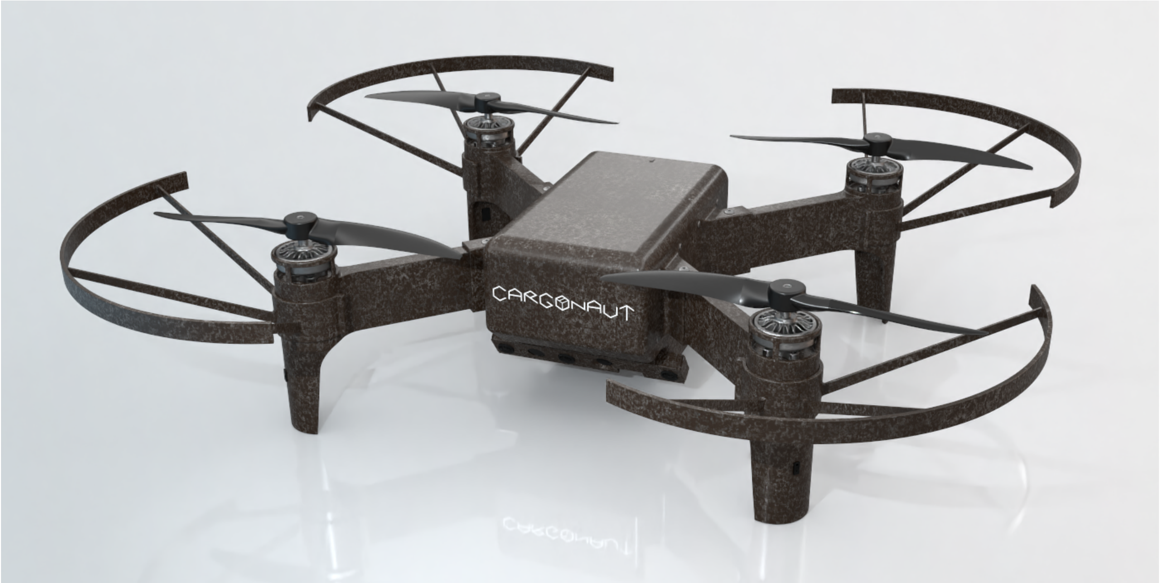


Figure 13.8: Final Cargonaut drone assembly

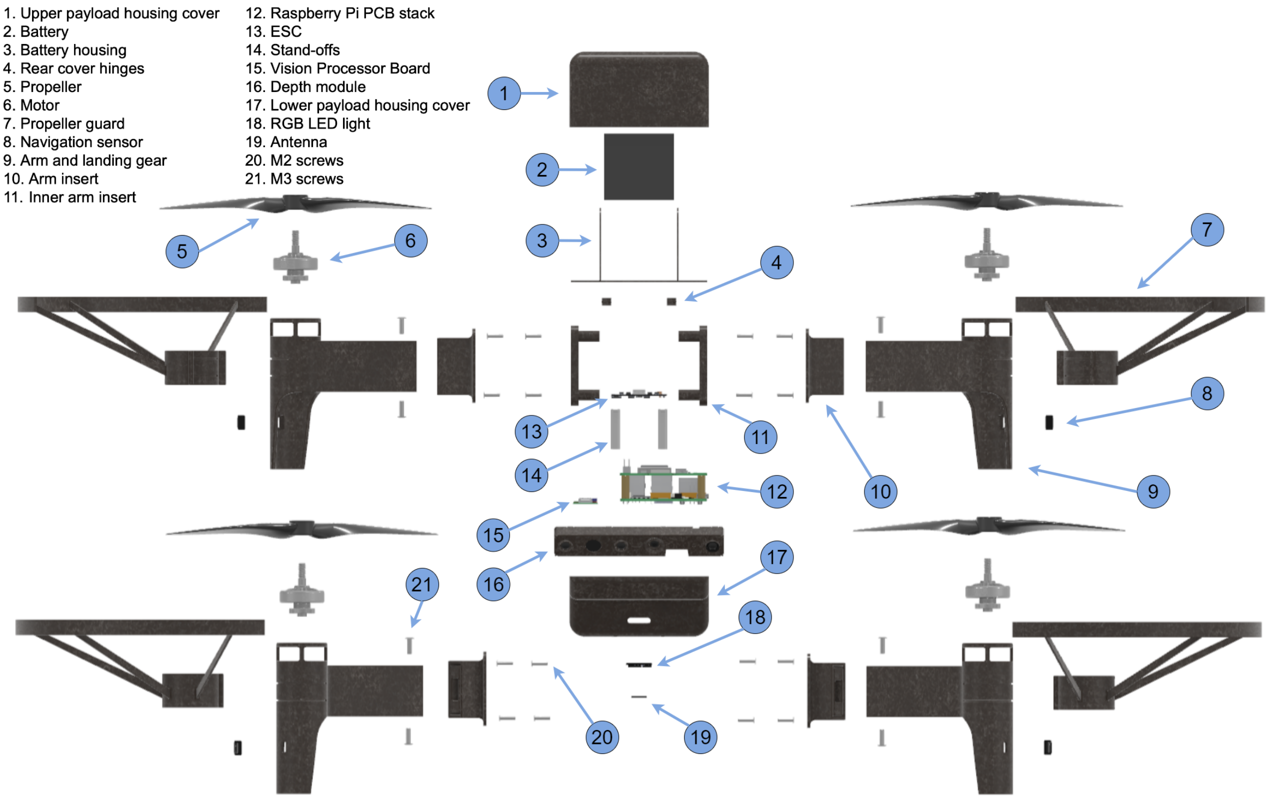


Figure 13.9: Cargonaut drone exploded view

13.5. Further Recommendations

The structural design of the drone frame was conducted based on simplified analytical solutions for verification, FEA conducted using Ansys, and frequency analysis performed in CATIA for validation. However, the mass of the drone frame exceeded the given requirement. To reduce this value, shape optimization may be performed within CATIA to remove material wherever it is deemed unnecessary. Furthermore, the risk of water damage to the components may be reassessed to possibly reconsider the payload housing structure, i.e. an open housing configuration may be worth the saved weight. To further verify the material selection, fatigue analysis may be conducted to evaluate the drone's performance with respect to repetitive loadings over long periods of time. Additional verification can also be performed by setting up a simplified analytical model of the whole drone, specifically to analyze the effects of shear and torsion, which were neglected during this design process.

Within Ansys, more detailed validation can be performed by varying the mesh sizes for each load case and observing whether or not the solutions converge. Dynamic analysis would be useful to more accurately model the impact load cases, as opposed to simplifying them as stationary forces. Iterations may also be carried out to find the exact loads at which the maximum allowable deflection and stresses are reached, along with the maximum height from which the drone may freefall and not fail. Moreover, more detailed frequency analysis can be performed on the entire drone to ensure compliance with the vibrational requirement, specifically to analyze whether or not resonance is reached as the motors power up to hover angular velocity. Finally, a more in-depth evaluation of various handling loads the drone may experience would allow for a more holistic overview of the structural behavior, and more sophisticated mechanisms may be used for ease of operation.

14: Control

This chapter covers the control and simulation design. The control subsystem design process is analyzed in detail by first presenting the goals and functions in Section 14.1, followed by the requirements and risks, presented in Section 14.2. The design process itself is shown in Section 14.3: here the control architecture and the control simulation are described, followed by the verification and validation steps, displayed in Section 14.4. Finally, the future recommendations are presented in Section 14.5.

14.1. Goals and Functions

The control subsystem is paramount to ensuring correct control, stability and attitude of the UAV. Accordingly, the goal of the subsystem is to

To implement a control architecture that will control and stabilize the drone around 6 degrees of freedom during all stages of the operations.

To achieve the goal, the following functions have to be fulfilled by the drone's control subsystem:

1. Stabilize the drone around 6 degrees of freedom.
2. Orient the drone such that the payload is able to carry on the required operations.
3. Provide accurate and reliable movements in all axes.

14.2. Requirements and Risks

Table 14.1 shows the subsystem requirements that were identified, stemming from the stakeholder and system requirements. Each requirement has been tailored based on the desired performance of the system. Safety as a value was instilled throughout the design process of the control and stability of the drones, ensuring that requirements relating to overshooting, and emergency stopping are taken into consideration.

Table 14.1: Requirements for Controls subsystem

Identifier	**Control Requirement**	**Associated Risk ID**
CRG-SNC-01	The drone shall be able to dampen perturbations	RSK-SNC-01
CRG-SNC-02	Each drone shall support the alignment of the payload with the target.	RSK-SNC-02
CRG-SNC-03	Each drone shall be controllable by authorized personnel.	RSK-SNC-03
CRG-SNC-04	The control system of the drone shall not be hazardous in case of a software or hardware failure in the control system.	RSK-SNC-04
CRG-SNC-05	Each drone shall be able to hover with a translation movement error of 1%.	RSK-SNC-05
CRG-SNC-06	The drone system shall not overshoot when yawing by more than 5 degrees of the requested attitude.	RSK-SNC-06
CRG-SNC-07	The drone system shall not overshoot in the x, y, and z direction by more than 5 cm the requested displacement.	RSK-SNC-07
CRG-SNC-08	The drone system shall have a settling time of no more than 3 seconds in each direction.	RSK-SNC-08
CRG-SNC-09	The drone system shall have a settling time of no more than 1 seconds in yaw attitude.	RSK-SNC-09
CRG-SNC-10	The drone system shall include a flight controller that is able to control individual propeller's rotational speed.	RSK-SNC-10
CRG-SNC-11	The drone system shall be able to perform trajectories based on waypoints.	RSK-SNC-11

Analyzing the risks associated with the requirements is critical to ensuring the success and reliability of the subsystem. For this reason, one risk was identified and analyzed for each requirement in terms of likelihood and severity, as shown in Table 14.2. Furthermore, a mitigation plan was created for each risk, to minimize their chance of occurrence.

14.3. Design process

Designing the control subsystem involves integrating sensors, actuators, and algorithms to ensure stable flight and precise maneuverability. Following the hardware choice performed during the preliminary design, the control architecture is presented in Section 14.3.1. The control simulation, created to address the performance and the tuning of the controllers, is presented in Section 14.3.2.

14.3.1. Control Architecture

Establishing the control architecture is the first step in providing the required controllability of the drone. It was decided to proceed with four PID controllers, namely the altitude, roll, pitch, and yaw PID, due to ease of implementation and reliable results shown in past implementations [85–88]. The general control logic is shown in Figure 14.1. The outputs of the controllers are used to obtain the required propeller speeds, as well as producing the new acceleration, which is used to update the velocity and position of the drone.

14.3.2. Control Simulation

A simulation was created from scratch in Python to test the correct implementation and behavior of the controllers. To create the simulation, the quadcopter dynamics and kinematics were implemented following [85, 86].

Table 14.2: Risks for Control subsystem

Identifier	Control Risk	Likelihood	Severity	Mitigation
RSK-SNC-01	Perturbations are not dampened by the control subsystem.	2	2	Ensure correct behavior of controllers for a wide range of scenarios and limit cases.
RSK-SNC-02	The payload is not aligned with the target due to the wrong attitude of the drone	1	5	Ensure correct behavior of controllers for a wide range of scenarios and limit cases.
RSK-SNC-03	The drone is not controllable by personnel	2	5	Perform routine maintenance checks to ensure the correct behavior of the control subsystem and of the emergency pipeline.
RSK-SNC-04	The drone behaves erratically due to hardware or software failure	3	5	Perform routine maintenance checks to ensure the state of the control subsystem and of the emergency pipeline.
RSK-SNC-05	The drone hovers with a translation error that hinders the payload performance	1	5	Ensure correct behavior of controllers for a wide range of scenarios and limit cases.
RSK-SNC-06	The drone system overshoots when yawing by more than 5 degrees	3	3	Ensure correct behavior of controllers for a wide range of scenarios and limit cases.
RSK-SNC-07	The drone system overshoots in the displacement directions by more than 5 cm	3	3	Ensure correct behavior of controllers for a wide range of scenarios and limit cases.
RSK-SNC-08	The drone system has a settling time in the displacement directions that hinders the payload performance timeline	2	4	Ensure correct behaviour of controllers for a wide range of scenarios and limit cases.
RSK-SNC-09	The drone system has a settling time in the yaw attitude that hinders the payload performance timeline	2	4	Ensure correct behavior of controllers for a wide range of scenarios and limit cases.
RSK-SNC-10	The flight controller fails to control individual propeller's speed	2	5	Perform routine maintenance checks to ensure the correct integration of flight controller, ESC, and motors.
RSK-SNC-11	The drone fails to follow trajectories based on waypoints	1	4	Ensure correct integration of the control algorithm with the Operations & Logistics, and Navigation subsystems for a wide range of scenarios.

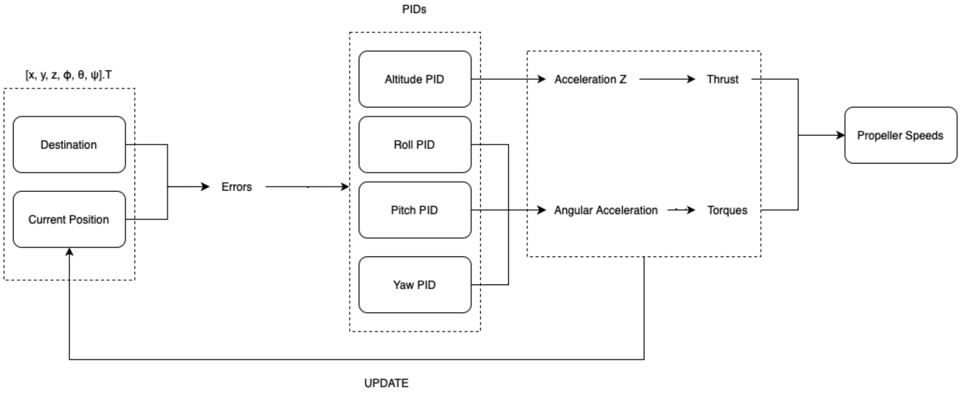


Figure 14.1: Control logic

To start the loop, an initial position and a desired position are required in the shape of array as $[x, y, z, \phi, \theta, \psi]$, and a total running time (in seconds) is also requested. Starting from the current position, the position error is obtained by vector subtraction with the desired position (also identified as the destination). The errors obtained in roll and pitch are further updated by also including the required rotation for thrust vectoring. This is done by including the additional angles required to obtain the error vector $[e_x, e_y, e_z]^T$. The error vector $[e_z, e_\phi, e_\theta, e_\psi]^T$ is then fed to the PIDs: each PID has as input its respective error and produces as output the required acceleration (in body frame), leading to a total output $[a_z, \alpha_\phi, \alpha_\theta, \alpha_\psi]^T$. This is then transformed into an inertial frame, from which thrust $[T_x, T_y, T_Z]^T$ is obtained. The propeller speeds $[\omega_1, \omega_2, \omega_3, \omega_4]^T$ are then computed from the thrust magnitude and torques $[\tau_\phi, \tau_\theta, \tau_\theta]^T$ (obtained from angular acceleration) following Equation 14.1: these values will be re-directed to the ESC[1] as input for the motors.

$$\begin{bmatrix} \omega_1^2 \\ \omega_2^2 \\ \omega_3^2 \\ \omega_4^2 \end{bmatrix} = \begin{bmatrix} 1/4b & 0 & -1/2bl & -1/4d \\ 1/4b & -1/2bl & 0 & 1/4d \\ 1/4b & 0 & 1/2bl & -1/4d \\ 1/4b & 1/2bl & 0 & 1/4d \end{bmatrix} \begin{bmatrix} T \\ \tau_\phi \\ \tau_\theta \\ \tau_\psi \end{bmatrix} \tag{14.1}$$

In the above-mentioned equation, b represents the propeller thrust factor, d the propeller torque drag factor, and l the distance from the rotor to the center of the drone. Furthermore, ω represents the propeller speed, with each number corresponding to a respective motor, as shown in Figure 14.2. Finally, T represents the total thrust magnitude needed, and each τ the torque with respect to each angle. At last, the update is performed: here, based on the acceleration, velocity, and position are updated, closing the simulation loop.

[1] Racerstar Air 50: https://www.123materialen.com/products/30-5-30-5-mm-racerstar-air50-3-6s-50a-4in1-esc-blheli_s-dshot600-compatibel-met-airf7-lite_1772372?gad_source=1&gclid=CjwKCAjwg8qzBhAoEiwAWagLrDBuEz35BXs5c6KR_gFL81GD7D5fkZ-goaFp620_rsMgRMxCmsQ1DhoCwvcQAvD_BwE. Accessed 19 June 2024.

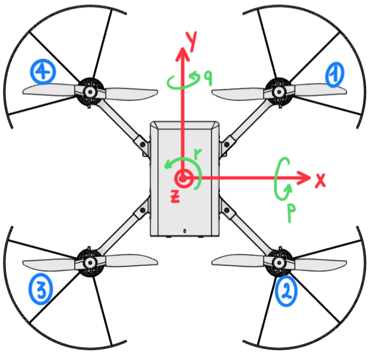


Figure 14.2: Rotors enumeration

Boundary values, such as maximum thrust, propeller speed, and maximum speed (during movements and during measurements) are also accounted for via checkpoints. Here if the requested thrust and/or propeller speed exceed the maximum allowed, they are manually set those values. For the maximum speed, if measurements are taken, this is set to $0.3\,\mathrm{m\,s^{-1}}$ as per **CRG-PROP-04**; if the drone is moving within the warehouse, the maximum speed becomes $3\,\mathrm{m\,s^{-1}}$ as per **CRG-PROP-02** and **CRG-PROP-03**. This is achieved by an if statement, initiated by object detection and localization, that switches the maximum velocity from the operational one to the measurement one. Overall, the specifications used are summarized in Table 14.3, coming from the Structures and Flight Performance subsystems.

Table 14.3: Drone specifications

Parameter	Symbol	Value	Unit
Mass	m	1.337	kg
Moment of inertia for X	I_{xx}	$7e-3$	kg m^2
Moment of inertia for Y	I_{yy}	$7e-3$	kg m^2
Moment of inertia for Z	I_{zz}	0.013	kg m^2
Rotor moment of inertia	I_r	$1.2e-5$	kg m^2
Distance from rotor to drone's center	l	0.169	m
Maximum propeller speed	ω_M	22,000	rpm
Propeller thrust factor	b	$1.01e-4$	N s^2
Propeller torque drag factor	d	$1.05e-6$	N m s^2
Maximum thrust over weight	$(T/W)_M$	4.0	-
Maximum speed measurements	v_M^M	0.3	m/s
Maximum speed	v_M	3.0	m/s

Furthermore, in case of emergency, when the kill switch is pressed or manual override is performed, the control system ensures that the rotors steadily stop spinning, causing the drones to slowly land on the ground, complying fully with **CRG-SNC-03** and **CRG-SNC-04**.

An additional risk was individuated during the design of the subsystem, due to the capping of signals during the control loop. Table 14.4 presents the risk and its mitigation plan.

Table 14.4: Windup Risk

Identifier	Control Risk	Likelihood	Severity	Mitigation
RSK-SNC-12	The controllers experience integrator windup.	4	4	Ensure integral term clamping when actuator is saturated [89]

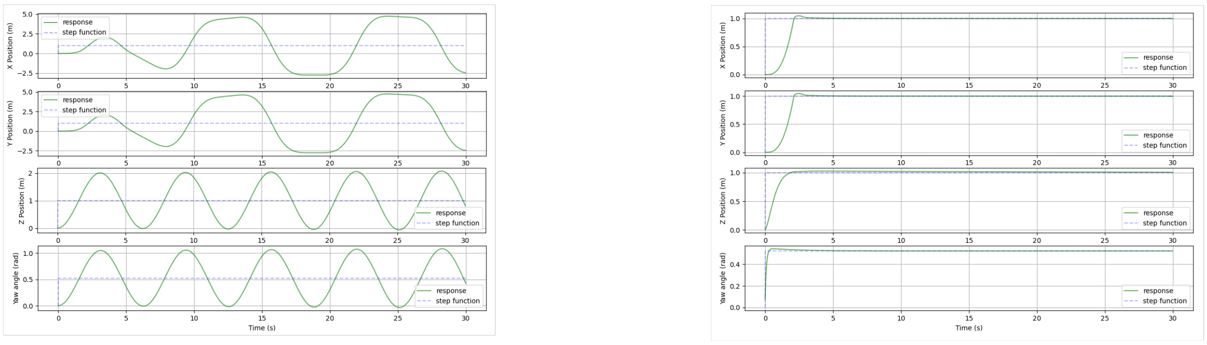
PID Tuning

Correctly tuning the controllers is essential in ensuring a correct response of the system to inputs. Many tuning methods exist, both rule-based and model-based[2] and are widely used for different applications. It was decided to proceed with the Ziegler-Nichol's method due to its simple and systematic approach, and ability to deliver robust performance in various dynamic conditions [90]. The results were then further improved via trial and error, leading to the behaviors shown in Figure 14.3. On the left, (Figure 14.3a) the responses before tuning are shown, whereas Figure 14.3b presents the tuned response to the same step input.

Sensitivity analysis was also performed by varying the drone parameters used by 10%. This was done to assess the behavior of the drone with the current PID tuning in case of slight changes in the specifications. The drone

[2] Explore the 3 PID tuning methods https://www.incatools.com/pid-tuning/pid-tuning-methods/. Accessed 12 June 2024.

behavior was considered acceptable, as the requirements were not met in only two cases. These cases were when the majority of the parameters were altered. These cases were: maximum overshoot of 8.2 cm (in the X and Y direction) once all the parameters, except the propellers factors, were altered; and a maximum settling time of 3.35 seconds (in the Z direction) when the moments on inertia and the maximum thrust were changed. The final tuned gains are presented in Table 14.5.



(a) Untuned PIDs

(b) Tuned PIDs

Figure 14.3: Quadcopter step response

Table 14.5: PID gains

PID	K_p	K_i	K_d
Altitude	5.70	0.25	4.30
Roll	0.90	0.15	0.90
Pitch	0.90	0.15	0.90
Yaw	4.70	0.20	10.00

After tuning, multiple random trajectories were generated to observe the drone's behavior: a circular one is shown in Figure 14.4 for completeness.

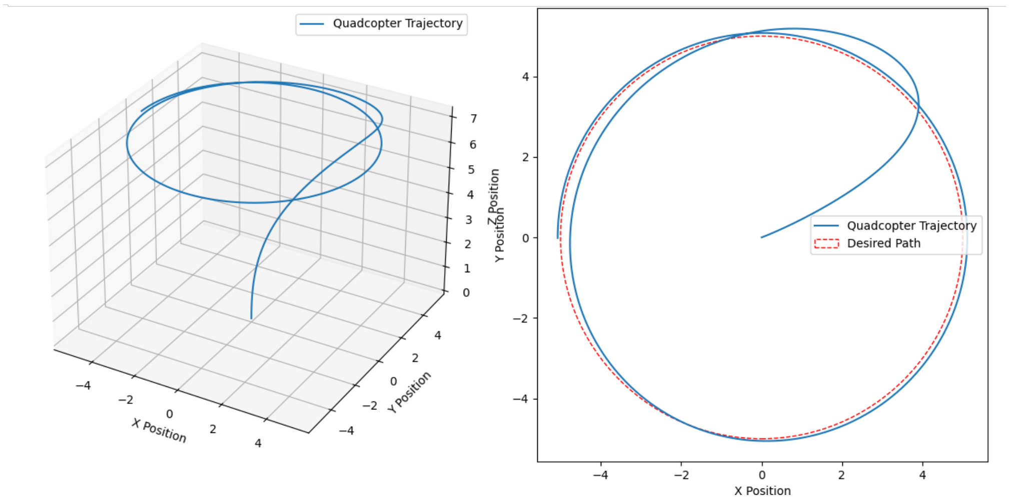
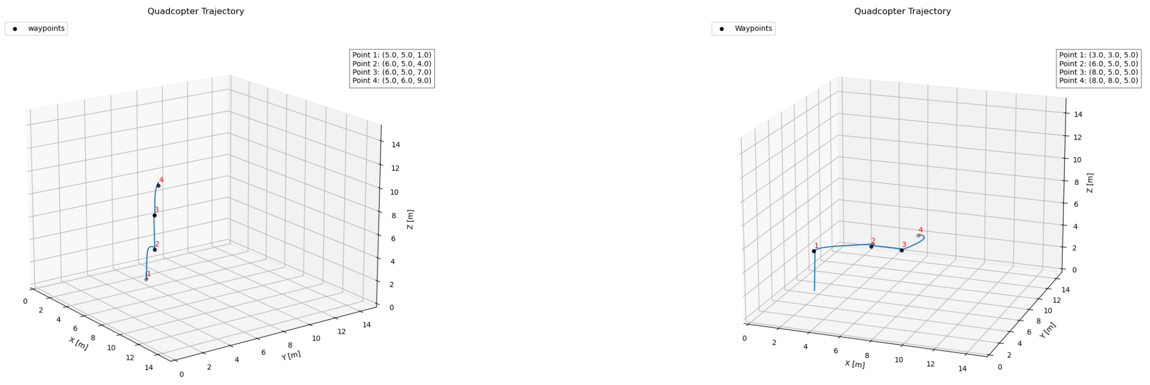


Figure 14.4: Circular trajectory

Waypoint Implementation

The simulation described in Section 14.3.2, was further augmented by including waypoints. This was done to observe the drone behavior during fast and sharp changes of direction, similar to what could be expected during cargo items inspection: here each waypoint simulates one target measurement. Two examples of waypoint trajectories, that could happen in warehouses, are presented in Figure 14.5. Figure 14.5a presents a trajectory where the drone mainly moves in the x-y plane, simulating a possible inspection of packages on the warehouse floor. Figure 14.5b shows an example of possible shelf inspection, where the movements are mostly vertical.



(a) Shelf inspection simulation

(b) Ground inspection simulation

Figure 14.5: Trajectories using waypoints

14.4. Verification and Validation

Verification and validation are paramount to ensure the performance and behavior of the UAVs. Verification is used to assess if the product is working properly [91]. Verification for the stability and control subsystem has been primarily performed using the simulation program. In the validation process one aims at matching the simulated data with real-life data [91]. Table 14.6 presents the verification and validation executed as well as the tests that will be performed in the future, once a working prototype has been obtained. The table shows the verification and validation tests, a brief description of what they entail, and finally how they have been carried out.

Table 14.6: Verification and validation for Controls subsystem

Verification	Unit tests	• Perform unit tests on individual functions of the simulation code to assess the performance with a variety of inputs.	• Each function was tested individually with a range of inputs. The outputs were than compared to manual calculation results.
	Module tests	• Test the quadcopter dynamic to ensure it accurately represents real-world physics, including mass, inertia, and aerodynamic forces. • Verify that the controllers are correctly implemented and integrated into the simulation.	• A variety of trajectories and cases were tested and compared to manually computed dynamics. • A variety of cases were performed for different PID settings, to ensure their correct influence in the code.
	System tests	• Test the fully integrated simulation system to verify all components work together seamlessly. This is done by simulating hovering, static yawing, and nominal-condition movements. • Test the simulation under extreme conditions, such as rapid movements, to assess the boundary performance and robustness of the code.	• Hovering, static yawing, and a range of trajectories were simulated and judged to ensure the correct movements. • Fast movements, rapid changes of directions, and circular trajectories were simulated and assessed.
	Controller tests	• Verify the step response performance to ensure the compliance with requirements.	• Step responses were assessed for all directions and for yaw rotations. Settling time and overshoot were compared to requirements to ensure their compliance.
Validation	Cross validation	• Other simulation programs are used to validate the performance of Cargonaut's stability and control system.	• The simulation was compared to other available simulations found via literature study.
	Hardware-in-the-loop tests	• Test the integration of the simulation with the control hardware to validate the real-life performance of the controllers.	• To be performed once prototype is acquired.
	Flight tests	• Test the integration and performance of the subsystem via real-life testing of the drone system.	• To be performed once prototype is acquired.

During the process, requirement compliance was assessed. Particular attention was posed to the controllers' performance, mainly referring to **CRG-SNC-05**, **CRG-SNC-06**, **CRG-SNC-07**, **CRG-SNC-08**, and **CRG-SNC-09**. Compliance with these requirements is shown in Table 14.7. The complete compliance matrix for all requirements is shown in Table 14.8.

Table 14.7: Compliance matrix for PID requirements

Identifier	Requirement	Result
CRG-SNC-05	Each drone shall be able to hover with a translation movement error of 1%.	TBC
CRG-SNC-06	The drone system shall not overshoot when yawing by more than 5 degrees of the requested attitude.	3.6 degrees
CRG-SNC-07	The drone system shall not overshoot in the x, y, and z direction by more than 5 cm of the requested displacement.	X and Y: 4.8 cm Z: 2.8 cm
CRG-SNC-08	The drone system shall have a settling time of no more than 3 seconds in each direction.	X and Y: 2.15 seconds Z: 2.53 seconds
CRG-SNC-09	The drone system shall have a settling time of no more than 1 seconds in yaw attitude.	0.33 seconds

Table 14.8: Compliance matrix for controls subsystem

Identifier	Verification Method	Compliance
CRG-SNC-01	Via simulation and future flight tests, the drone will be tested in the presence of gusts, winds, and other perturbations.	TBC
CRG-SNC-02	Via simulation, the drone is able to provide the required attitude.	✓
CRG-SNC-03	Implementations for control safety have been added to the control loop.	✓
CRG-SNC-04	Implementations for control safety have been added to the control loop.	✓
CRG-SNC-05	Via the simplified simulation, the drone fulfills the requirement. Finale compliance will be proven with a flight test.	TBC
CRG-SNC-06	Via simulation, the drone is able to provide the required attitude within the margin.	✓
CRG-SNC-07	Via simulation, the drone is able to provide the required position within the margin.	✓
CRG-SNC-08	Via simulation, the settling time complies with the requirement.	✓
CRG-SNC-09	Via simulation, the settling time complies with the requirement.	✓
CRG-SNC-10	The PID architecture has been chosen to provide individual propeller speeds.	✓
CRG-SNC-11	Waypoint trajectories were tested using the control simulation.	✓

14.5. Future Recommendations

The control subsystem design has been completed for the basic scenarios that can occur during Cargonaut's operation inside warehouses. It is necessary to further integrate and assess the performance of the controllers in the presence of turbulence and gusts caused by open doors and rapid movements of large items. Furthermore, attention has to be posed on the stability and controllability of the drone system in case of failure of one or more propellers. Improvements can be applied to the tuning of the controllers: an auto-tuning algorithm shall be implemented to optimize the gains and decrease the tuning time.

Further validation is also required. This is scheduled to happen once the hardware is acquired and the first prototype is built. These tests will provide valuable information regarding the integration of the flight controller, ESC, and motors, as well as provide feedback on the performance of the drone with the PID tuning implemented.

15: System Integration

This chapter presents the integration of all the subsystems and software modules, in order to offer a total overview of the drone system. Section 15.1 starts by detailing the overarching software configuration and the interfaces between the various modules. Subsequently, Section 15.2 elaborates on the hardware integration via the use of a hardware diagram. Then, Section 15.3 presents the technical budgets of mass and power, while Section 15.4 exhibits the cost breakdown structure of the project as a whole. Furthermore, the compliance with the stakeholder requirements is shown in Section 15.5. Finally, the market solution is displayed in Section 15.6: here the product offered is described from a marketing perspective.

15.1. Software Architecture

Having so many software modules raises concerns about computational power distribution and data transfer. The main benefit of having an on-board TPU with 14 cores is the availability of running multiple threads in parallel. The main processes to be handled by the onboard computer are the payload, operations, navigation, and control. An overview of the modules of each subsystem is included in Figure 15.1 along the interactions between elements.

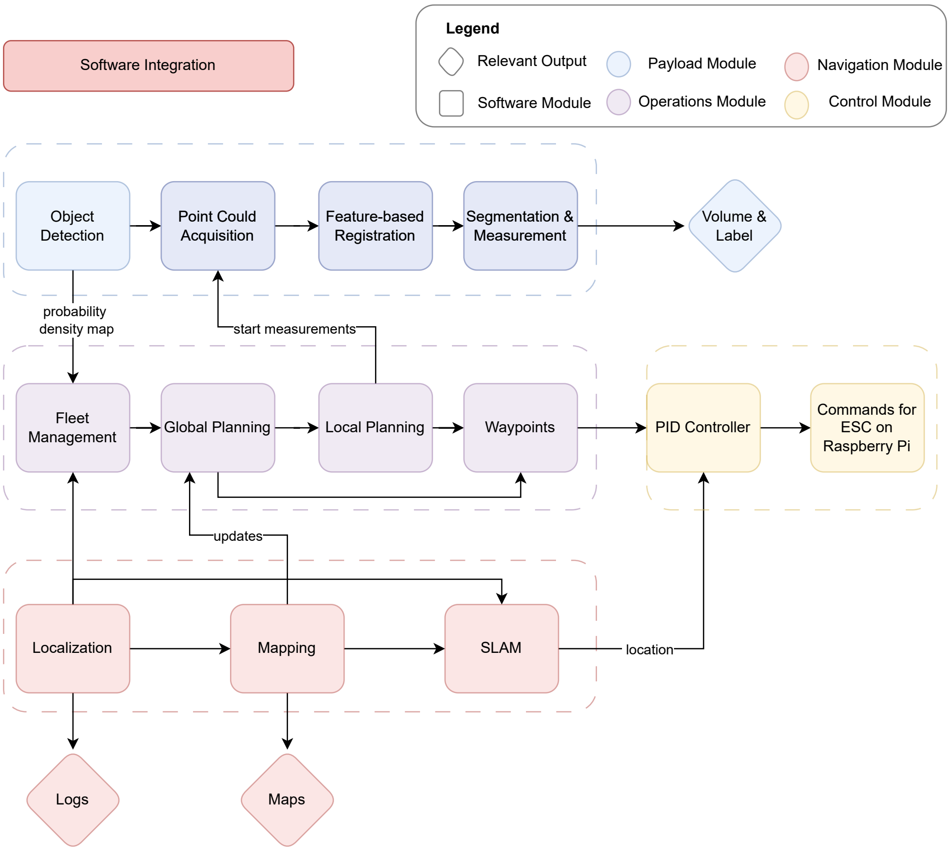


Figure 15.1: Software integration

Interactions between navigation and path-planning algorithms are typically handled through various packages of the Isaac ROS framework. Additionally, the interactions between the controller and SLAM as well as the local planner and the payload are facilitated by the Raspberry Pi.

Management of computational power will be conducted by running certain algorithms on the CPU and others on the TPU, depending on their requirements. Since payload data processing requires extensive resources, it will be run on the TPU. Object recognition and point cloud acquisition are intended to be run on the TPU along with the SLAM processes, while fleet management, trajectory planning, and state-space models can be run solely on the CPU of the Raspberry Pi to avoid overload. On the TPU itself, the management of computational power can be conducted by dedicating certain cores to certain processes. However, such a procedure is very complex and does not guarantee the completion of the computation in a fast manner. Therefore, at this point of design, it is assumed that the computational resources: TPU and CPU are sufficient to support all the software modules of Cargonaut.

15.2. Hardware Integration

Having performed the software integration in the previous chapter, this section brings together the software modules with the components of other subsystems. From a hardware point of view, the software is represented by the Raspberry Pi and AI Module. Figure 15.2 displays the connections between various hardware components from both a power and data interface standpoint.

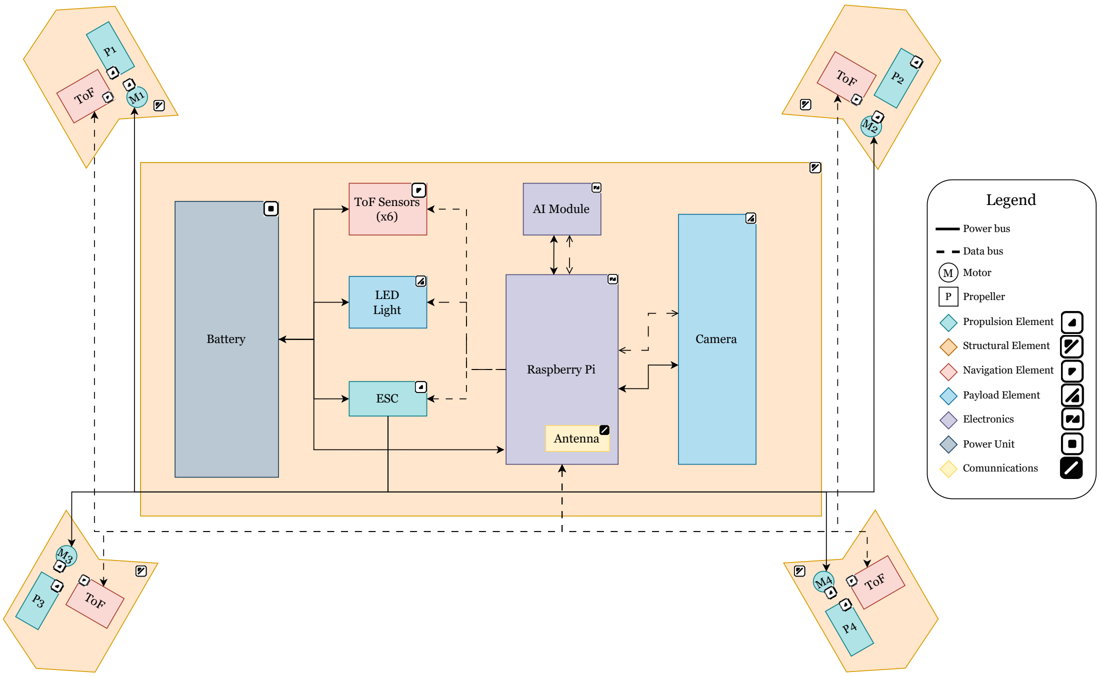


Figure 15.2: Hardware Block Diagram

15.3. Budgets

This section introduces the final technical budgets: the power and mass budgets. Section 15.3.1 presents the mass budget, while Section 15.3.2 presents the power one. Both budgets follow the guidelines presented in Chapter 3 regarding the use of contingency margins. These budgets were compiled depending on the design choices each subsystem department makes, which are bounded by relevant requirements sets to comply with stakeholder requirements.

15.3.1. Mass Budget

The final mass budget is presented in Table 15.1. The values have been obtained from the specifications of the components used for each subsystem, and, in the case of Structures & Materials, they stem from the **CAD!** (**CAD!**) design. Furthermore, the Electronics subsystem includes not only the battery but also the necessary converters and cables needed for the integration and functioning of the system.

The budget is divided by subsystems and presents, in the final row, a conservative margin of 5%. This includes the possible non-availability of some components, causing the use of similar ones with different mass specifications, the accuracy of manufacturing, and the cabling needed to connect the components.

Table 15.1: Mass budget

Subsystem	**Mass [g]**	**Percentage [%]**
Payload	124	9.2
Navigation	2	0.1
Control	13	1.0
Propulsion	184	13.8
Communication	1	0.1
Electronics	632	47.3
Structures & Materials	381	28.5
Total	**1,337**	**100**
Total with contingency	**1,404**	

15.3.2. Power Budget

The final power budget is presented in Table 15.2. These values have been obtained from datasheets of the components used for each subsystem, and, in the case of Propulsion, they stem from the internal analysis done in Chapter 10. Additionally, the Electronics subsystem includes the power losses of cables, transmitters, and other interfaces.

Similarly to the mass budget, it is also divided by subsystems and presents, in the final row, a conservative margin of 5%. This encompasses the possibility of certain power losses being miscalculated, needing to add other supplementary components to the system, and other shortcomings present in the design.

Table 15.2: Power budget by subsystem

Subsystem	Power [W]	Percentage [%]
Payload	18.42	7.63
Navigation	2.319	0.961
Control	0	0
Propulsion	213.4	88.4
Communication	1.000	0.414
Electronics	6.260	2.59
Structures & Materials	0	0
Total	**241.40**	**100**
Total with contingency	**253.47**	

15.4. Cost Breakdown Structure

The project can be further assessed in terms of costs. The cost breakdown structure focuses on the following components:

- **Research & Development:** representing the phases of the design, from the initial assessment to the concluding tests of verification and validation for each subsystem and the system as a whole. Since the main part of the design was done during the 10 weeks of DSE, the current costs for R&D amount to only the price of the depth camera that was acquired for testing. For the remaining part of the design, costs are expected to rise steeply, since one/multiple prototypes will have to be built and tested: the costs are expected to be around 10,000 euros[1].
- **Hardware:** the costs concerning the acquisition of all the necessary components. The costs have been obtained using prices of required components available in stores and on the Internet. Table 15.3 presents the cost budget per drone divided by components, while Table 15.4 presents the costs per subsystem. It is important to note that the last row presents the final hardware cost with a +5% margin applied to account for the chance of having to buy similar components due to non-availability.

Table 15.3: Cost budget by component

Item	Cost [€] FY2024	Subsystem
Time-of-flight sensors	85.31	Navigation
ESC	37.99	Control
Intel 455	227.49	Payload
RGB light	0.93	Payload
AI accelerator	79.80	Payload
Raspberry Pi 5	72.44	Payload
Motors	64.41	Propulsion
Propellers	10.34	Propulsion
Antenna	0.33	Communication
LDOs	2.02	Electronics
Buck converter	2.05	Electronics
Logic translators	3.68	Electronics
Cables	11.67	Electronics
Battery	90.44	Electronics
Wood Carbon Fiber Composite	4.98	Structures & Materials
Total	**693.66**	

[1] How much do Prototypes cost?: https://design1st.com/how-much-do-hw-prototypes-cost/. Accessed 23 June 2024.

Table 15.4: Cost budget by subsystem

Subsystem	Cost [€] FY2024	Percentage [%]
Payload	380.66	54.9
Navigation	85.31	12.3
Control	37.99	5.5
Propulsion	74.53	10.7
Communication	0.33	0.1
Electronics	109.86	15.8
Structures & Materials	4.98	0.7
Total	**693.66**	**100**
Total with contingency	**728.34**	

- **Software:** regarding the costs related to the implementation and maintenance of the algorithm used. Since all the software is being developed in-house or using open-source resources, the costs are currently none.
- **Manufacturing:** representing the costs related to the manufacturing and assembly of the drones. As the manufacturing process consists of injection molding, the category can be further broken down into mold costs, production, assembly, and quality control. Molds are expected to be a one-time cost: as seven molds are required, this will amount to around 140,000 euros[2]. Expecting a successful operation, this can be spread on each drone, accounting for 3 euros for each. Production includes the actual process, labor costs, and machine operation; assuming a medium-scale production, the cost for each drone will be around 3 euros[1]. Assembly consists of integrating the structure with the hardware, based on labor rates and expected complexity, this cost can be assumed to be 50 euros per drone[3]. Finally, to ensure quality standards and reliability, quality control has to be performed: it is expected to cost around 5 euros per drone.
- **Certifications & Compliance:** concerning the required certifications, patents, and audits required to comply with EU (and more generally international) regulations and ensure the correct behavior of the system. These costs are estimated to be around 20,000 euros due to the operation of the drone systems in the presence of people and in indoor environments. Furthermore, in this category, costs related to liability insurance, necessary consultations, and legal fees are also accounted for. These are expected to be in the 5,000-10,000 euros range[4].
- **Deployment:** representing the installation and integration of the system in the warehouse. The cost is estimated to be 2,000 euros for each warehouse installation, due to the flexibility of the product and the minimally invasive installation process.
- **Training & Support:** the costs required to train the operator(s) responsible for the system once deployed. The total amount is estimated to be 2,000 euros per customer.
- **End of Life Solution:** Cargonaut makes itself responsible for taking care of the end-of-life procedures of the system, in order to ensure its correct and sustainable disposal. Based on the material used, and the hazardous components, such as the battery, these costs are expected to be quite high, ranging in the 1,000-3,000 euros.
- **Overhead:** representing the general costs expected within any institution. Considering currently the members are working for free and are given free spaces to work in, the costs have amounted to none. However, after the end of the 10 weeks of DSE, the costs are expected to rise significantly, due to the need for a working space and other amenities. By keeping the personnel costs to zero, meaning that each member will volunteer until revenues start coming in, the costs are expected to amount to 2,000 euros per month, based on workspace prices in Delft, The Netherlands, where the project wishes to be located.

Table 15.5 presents a summary of the expected costs. The table is divided into three categories, namely total costs, costs per order, and yearly expenses. More specifically, total costs include the research & development, software, and certifications & compliance costs; the costs per order account for hardware, manufacturing, and end-of-life for each drone in the requested fleet (for ease of calculations assumed to be 10), as well as the deployment and training required; finally, the yearly expenses represent the overhead costs. In case a range was previously presented, here the mean value is shown. The amount has to be considered as a ballpark figure since not a lot of information can be found regarding similar companies and start-ups due to the novelty of the solution that Cargonaut is offering.

[2] How to Estimate Injection Molding Costs?: https://formlabs.com/eu/blog/injection-molding-cost/. Accessed 18 June 2024.

[3] Xometry: https://www.xometry.com. Accessed 18 June 2024.

[4] What does ISO Certification costs?: https://reciprocity.com/resources/what-does-iso-certification-cost/. Accessed 23 June 2024.

Table 15.5: Cost breakdown structure

Item	Cost [€] FY2024	Category
R&D	10,000	Total
Software	0	Total
Certifications	27,500	Total
Hardware	7,000	Per order
Manufacturing	610	Per order
Deployment	2,000	Per order
Training	2,000	Per order
End-of-life	2,000	Per order
Overhead	24,000	Yearly

Based on the breakdown shown above, **CRG-STK-16** has been complied with, since the manufacturing and hardware costs per drone amount to 754.66 euros. If the research & development and certification costs are also included, assuming a first production size of 100 drones - circa ten orders - the amount of the total costs is 1,129.66 euros per drone. Since, as mentioned above, these are to be considered approximations, a margin of 10% is to be considered: with this conservative approach, the cost per drone will be 1,242.63 euros.

In the cost breakdown structure, the focus was posed on the drone itself; the costs related to Cargomother have not been accounted for. It is expected that a lot of resources will have to be focused on the development of these charging stations, as well as analyzing components and manufacturing costs. Based on the initial design of these docks, the costs per unit are expected to be around 700 euros. With these additional costs, the team is still expecting to comply with requirement **CRG-STK-16**.

Based on companies in the drone manufacturing industry[5] and in the logistics market[6], the average profit margin is around 30 to 45%. Since the initial costs are just an estimate, a margin of 45% is here considered. Based on this, Cargonaut is planning to sell the drone-software unit at a price of 2,000 euros, with the possibility of discount for orders larger than 8 drones to 1,900 euros per unit. Cargomother is expected to be sold at 1,200 euors, with stations larger than 8 docks being discounted to 1,100 per dock unit. Accounting for this, assuming an average order of 10 drones, the revenues per fleet will amount to 30,000 euros. The profits are thus expected to be 13,500 euros per order. The return on investment on the molds, needed to start the manufacturing process, is thus planned for the 11th order, once the break-even point is passed. This further emphasizes the production size of around 100 drones mentioned above.

15.5. Requirement Compliance

After integration, the team is at a position to check whether the designed solution complies with the stakeholder requirements. Note that compliance with subsystem requirements has been treated within subsystems. The complete compliance matrix for the stakeholder requirements is given in Table 15.6. For further reference recall the stakeholder requirements as presented in Table 4.1.

[5] Drone Manufacturer Earnings, a Comprehensive Guide: https://finmodelslab.com/blogs/how-much-makes/how-much-business-owner-makes-drone-manufacturing. Accessed 24 June 2024.

[6] Are logistics companies profitable?: https://medium.com/@eximlogisticspvtltd/are-logistics-companies-profitable-d7c47d56d89c. Accessed 24 June 2024.

Table 15.6: Verifying compliance with stakeholder requirements

Identifier	Verification Method	Compliance
CRG-STK-01	Analysis: compute the cumulative time required to perform a task.	✓
CRG-STK-02	Simulation: measure the throughput of the system.	✓
CRG-STK-03	Simulation: switch between strategies.	✓
CRG-STK-04	Simulation: switch between strategies.	✓
CRG-STK-05	Analysis: compare design specifications against required precision. Test: verify actual precision with measurements.	90-99% accuracy at present, further development required.
CRG-STK-06	Simulation: test that the drone can operate in these conditions	✓
CRG-STK-07	Test: confirm that the drone's battery can be fully recharged.	TBC
CRG-STK-08	Simulation: model and predict collision probabilities in various operational scenarios.	✓
CRG-STK-09	Analysis: the damages by the drone system are none (in terms of collision).	✓
CRG-STK-10	Inspection: check that batteries meet regulatory standards.	✓
CRG-STK-12	Inspection: the materials used are least 70% recycled.	✓
CRG-STK-13	Test: ensure that the payload of the drone is less than 4 kg.	✓
CRG-STK-15	Inspection: measure the largest diameter of the drone.	✓
CRG-STK-16	Analysis: the cumulative sum of components and the drone production do not exceed 2,000 Euros.	✓
CRG-STK-17	Inspection: a human interface is provided.	✓
CRG-STK-18	Analysis: maintainability is considered within the design. Test: the drone system can be maintained.	✓
CRG-STK-19	Analysis: ensure all design choices are compatible with regulations.	✓
CRG-STK-20	Analysis: review the sustainability of the design.	✓
CRG-STK-21	Test: deploy system in a warehouse.	TBC
CRG-STK-22	The data privacy of all associated parties shall be protected.	TBC
CRG-STK-23	Test: deploy system in a warehouse.	TBC
CRG-STK-24	Sminulation: check the functionality of the kill switch.	✓
CRG-STK-25	Simulation: simulate the system in different warehouse configurations. Test: deploy the drone system in various warehouse configuration.	✓
CRG-STK-26	Inspection: review the completeness of the design after ten weeks.	✓
CRG-STK-27	Test: deploy drone in warehouse	TBC

15.6. Market Solution

Cargonaut offers an autonomous, flexible, and dynamic solution for the measurement of cargo items. The product consists of a docking station, the drone(s), and the software, which has to be integrated into a server. The customer can fully customize the number of drones needed: from one to multiple working in swarm formation. Due to the high flexibility of the product, models can always be added, as they can be easily booted in the existing system. The cost of each drone will include the software plug-in. The Cargomother docking station is available in multiple configurations (1, 4, 9, or 16 docks), with each docking slot priced according to the specific configuration selected. This modular approach not only enhances operational efficiency but also allows for cost-effective scalability.

The solution represents a significant leap forward in air freight cargo logistics, offering a state-of-the-art solution for dimensioning cargo with great accuracy and speed. Traditional cargo measurement methods are often time-consuming and prone to error, leading to inefficiencies and potential revenue losses. Cargonaut's technology enables rapid and reliable assessment, significantly reducing measurement times by up to 50%[7], processing up to 1,000 items per hour. This efficiency gain translates into higher throughput and productivity rates. Additionally, the enhanced accuracy of cargo measurements ensures better utilization of cargo space, which can lead to better utilization of cargo space. Cargonaut's innovative approach, leveraging autonomous drone technology and scalable infrastructure, positions it as a pivotal tool for the future of air freight logistics. By integrating this advanced system, companies can achieve higher operational efficiency, reduce costs, and maximize revenue, all while maintaining the flexibility to adapt to evolving logistical demands.

[7] CARGOMETER, On-the-fly Freight Dimensioning Category: Intralogistics Software: https://logistik-heute.de/sites/default/files/public/data-event/EN_IFOY2019_Test_Report_CARGOMETER__on-the-fly-feight-dimensioning.pdf. Accessed 17 June 2024.

CARGONAUT

FACT SHEET

EFFICIENCY TAKES FLIGHT

Dimensions	50 x 50 x 12 cm
Mass	1.3 kg
Market Cost per Unit	€2,000
Max Flight Time	30 min
Max Speed	3 m/s
Max Operative Distance	200 m
Propeller Diameter	7 inches
Max RPM	20,000 RPM
Object Detection	Real-time
Camera	Intel® RealSense™ Depth Camera
Frame Recyclability	>70%
Frame Material	Wood Carbon Fiber Composite

Battery Type	Lithium-Polymer
Battery Capacity	8,400 mAh
Battery Voltage	14.8 V
Battery Depth of Discharge	80%
Average Charge Time	17 mins
Fleet Management	Auction-based
Operational Modes	Cargonaut & Cargoscout
Trajectory Planning	A* with Collision Avoidance
Processing Unit	Raspberry Pi 14 with Core TPU
Navigation Sensors	VL53L9 Micro TOF Camera
Telemetry Link	2.4GHz WiFi

CARGONAUT
CARGONAUT

16: Sustainable Development Strategy

Sustainability and sustainable practices are crucial in a design to ensure that a better future can be preserved. Therefore, significant effort has been made in the design and operations of Cargonaut in terms of sustainability which is introduced in this chapter. This chapter includes an overview of sustainability in Section 16.1, ethical considerations in Section 16.2, sustainability in hardware and design choices in Section 16.3, sustainability in terms of data in Section 16.4 and a life cycle analysis in Section 16.5. Section 16.6 introduces the mitigation plan taken and Section 16.7 wraps the chapter up with posing recommendations for future reference.

Undermining sustainability is detrimental to any project, both in the idealization phase and the operations phase. Given the decline of resources and the Overshoot Day approaching ever-so-closely to New Year[1], sustainability becomes a critical component of any system. Holistically, sustainability, in this case, is not only limited to ecological impact reduction but a multitude of actors influencing economic, social, and environmental aspects altogether. To accommodate the encompassing definition of sustainability, sustainability is defined according to the 1987 United Nations (UN) Brundtland Commission, which is "meeting the needs of the present without compromising the ability of future generations to meet their own needs" [2].

16.1. Sustainability Overview

This section outlines what sustainability entails and what the necessary capacities a system has to accommodate sustainable practices. Also, the distinction between sustainability and sustainable development is introduced, and how the project goals align accordingly.

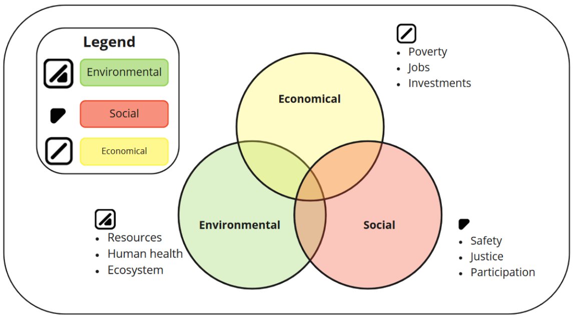


Figure 16.1: Sustainability pillars

Sustainability is based on three pillars: environmental, social, and economic [92]. Environmental sustainability deals with the balance between nature's replenishment and humanity's consumption levels, social sustainability is to what extent and in what manner society upholds human rights, and economic sustainability addresses whether secure sources of livelihood are available to everyone. These pillars are interconnected, and they influence each other with varying severity. This interaction is visualized on Figure 16.1.

Within the scope of this project, the decision to incorporate ethics as an extension to all of these pillars is taken. This decision was made to address the potential ethical implications of the product, but also to incorporate values in the design process by using Value Sensitive Design approaches. This is also reflected in the sustainable development policy, given that this consideration is part of a responsible innovation approach.

Sustainable development consists of many processes and pathways to achieve sustainability, it is interconnected to the general idea, yet is action-oriented[3]. Abiding by these ideas, the goal of this chapter is also to introduce how the project aligns itself with sustainable development and what practices adapted within it assure a prolonged development.

16.2. Ethical Perspective

"Designing is (always) about making choices and setting a future belief that people (users) must eventually embrace." [93]. To incorporate an ethical approach to the project, systematic steps are taken to involve Value Sensitive Design principles. This section will detail the moral aspect of the project from a socio-technical perspective and provide a breakdown of values that the project upholds immediately during the project design phase.

[1] Earth Overshoot Day: https://www.genevaenvironmentnetwork.org/resources/updates/earth-overshoot-day/. Accessed 2 May 2024.

[2] Sustainability: https://www.un.org/en/academic-impact/sustainability. Accessed 2 May 2024.

[3] What is Sustainable Development?: https://www.un.org/sustainabledevelopment/blog/2023/08/what-is-sustainable-development/. Accessed 3 May 2024.

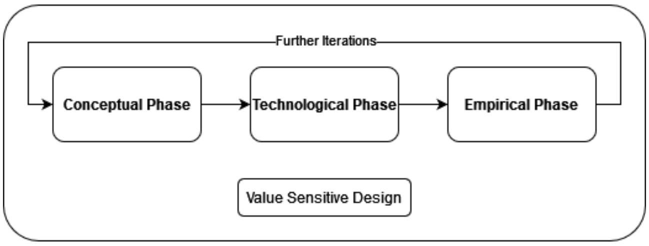


Figure 16.2: Value sensitive design phases [94]

Value-sensitive design is a framework that allows for making value-based choices in design. It is not a guideline, nor a rule book, which would automatically make a design ethical. It is merely an iterative tool that allows for a conscious check of the current design and its implications on ethics in the hopes of incorporating the values held by the stakeholders, such as sustainability or privacy, within design decisions. Therefore, within Cargonaut, value-sensitive design helps conceptualize values into technical elements.

This iterative nature of value-sensitive design, depicted in Figure 16.2, is crucial for ensuring that all stakeholders' needs are considered throughout the design process. The conceptual phase also contextualizes non-technical values, which encapsulate sustainability, privacy, safety, and accessibility. The technological case is where user-centered prototypes are generated that attempt to reflect these values in the design. The empirical phase is where the designs are weighed with various sociometric methods to propose an iteration that has the most adaptability and suitability concerning the proposed values [94].

Cargonaut has determined various cases in the warehouse environment that will be subject to change. Firstly, the widespread adoption of drones will cause a shift in the type of work that is available, meaning the inclusion of Cargonaut in a workspace can require management to consider sourcing different skills and talents for the same operational job. This can manifest in either the threat of being laid off or having to give a path to the drone and halt worker activity for a while when the drone and the workers are cooperating. The latter would cause discomfort, yet this would only be applicable if the drone is flying low, which is unlikely unless the cargo items are placed low. Given that jobs are not only a monetary supply but also a social entity, it is evident that the mental well-being of personnel will be affected [95]. This can manifest in anxiety over being laid off and diminished dignity with the thought that a machine or personal devaluation could replace them given that jobs are a major part of self-identity. Even with these harsh implications, Cargonaout has opted for alleviating accessibility to combat possible unintentional setbacks of automation. By making the product as accessible and having the interface as simple as possible, it is envisioned (with an optimistic outlook) that minimal training will be required to reorient and diversify the skills of the workers. This dynamic within the work environment can be observed in Figure 16.3.

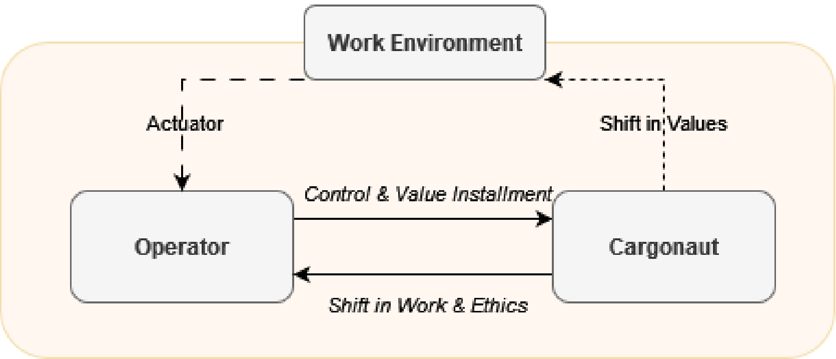


Figure 16.3: Cargonaut interaction with the work environment

A setback that is difficult to account for is the increase in surveillance within a workspace [93]. This increment would promote the unintentional idea of strengthening the overseeing aspect of the authority figure and might promote a "chill effect" for the workers. The fact that there would be little to no communication simply due to the management that employs Cargonaut's services thinking that it is not necessary to determine what data is processed and how it is processed could only worsen this effect, in case of loss of transparency. This would unintentionally cause workers to modify their behaviors over the fear of being watched, and promote an unhealthy workspace. Designing optical indicators for when recording takes place making them visible to workers in the warehouse and blurring people from the payload frames to uphold their privacy was implemented in the design to counteract these difficulties.

Finally, Cargonaut recognizes the importance of technological readiness, as such it envisions tracking the project's progress from a technological readiness point of view as it is being conceptualized, designed, and progressed through operations. This will be done by following National Aeronautics and Space Administration (NASA)'s technological readiness cycle[4] and the guidelines set forth by the European Space Agency (ESA)[5]. Specifically, the design will consider Technology Readiness Level (TRL) 6 and above as much as possible, and specific requirements for critical subsystems such as the payload will be imposed relating to their minimum TRL rating. This gives more feasibility in the design, but also reduces part-verification given high TRL subsystems are likely to be widely used in the market, thus trusted way more.

16.3. Sustainability in Hardware

During the design process, hardware choices were made with sustainability and engineering budgets in mind. These choices vary from choosing the most power-efficient components to aligning with companies such as Intel that have implemented plans to source their power from only renewable energy sources [6], especially relevant for the payload camera selection.

Regarding the material choice of Cargonaut's frame, WCFC was selected. Up to 97% of this material can be obtained in recycled form. Moreover, this material is robust and when selecting it life span was prioritized, making it most durable. For the structural frame design overall, a modular design was created, keeping ease of part replacement in mind for sustainability purposes as discussed in Section 13.3.1. In the design choices of the Communications and Data Handling subsystem, sustainable aspects were likewise considered. WiFi was selected as a data transmission method mainly due to its high throughput, resulting in high power usage efficiency, which is also beneficial in reducing the power needed to communicate between systems. Furthermore, this high efficiency extends the functional lifespan of networking devices, reducing the need for replacement of the devices and thus electronic waste. When navigating and controlling the drone, sustainability was kept in mind: the decision to proceed with FAST-LIO2 was mainly based on computational efficiency, ensuring minimal use of battery and electrical resources. The controller architecture and tuning performance were also chosen to minimize overshoot and non-vital components, resulting in an efficient design. In terms of the propulsive subsystem, motors were selected to minimize internal losses and waste of energy.

Nonetheless, in terms of environmental sustainability, there is still analysis that must be performed on noise calculations. To ensure a pleasant working environment, acoustic emissions shall be held to a minimum, and comply with EU requirements. It is acknowledged that human health is a priority, and noise shall be closely monitored. On a side note, giving precedence to workers' health is an additional reason why the rotors are surrounded by propeller guards, to minimize any damage in case of collisions.

16.4. Sustainability in Data Practices

In an attempt to track developmental and estimate operational carbon emissions, an external wrapper called `codecarbon`[7] was utilized. The wrapper tracks per line of specified code how much carbon emissions it takes for the code to compile, expressed in kW h of electricity used by components on the device. This has been implemented to track the carbon emissions during the development of AI algorithms, OpenFOAM usage for flight performance calculations, operational sequences; point cloud generation, filtering, registration, and volume estimation. The indicators used to track sustainability in the process were energy consumed for all GPUs, energy consumed for all CPUs, and electricity used since the beginning in terms of kW h.

These values were then categorized into two categories, namely developmental and operational. In the developmental category, AI model training presented in Section 7.3.2 and OpenFOAM activities presented inSection 10.3 are included. Due to the software being a wrapper for code, all activities related to CAD design were neglected as they were untraceable. From the operational side, all activities that a singular drone will conduct on the software side are included. This includes inference of the AI model onboard, point cloud operations, and autonomous decisions from the operational side. For training the AI algorithm for 25 epochs in 50 hours, 1.238150 kW h of GPU, 1.468475 kW h of CPU and 3.092375 kW h of total electrical power is reported. For all other software that is to be run on the drones continuously, refer to Table 16.1. It should be noted that these are representative numbers as these values change depending on the processing units utilized.

[4] Technology Readiness Levels: https://www.nasa.gov/directorates/somd/space-communications-navigation-program/technology-readiness-levels/. Accessed 3 May 2024.

[5] Technology Readiness Levels (TRL): https://www.esa.int/Enabling_Support/Space_Engineering_Technology/Shaping_the_Future/Technology_Readiness_Levels_TRL. Accessed 3 May 2024.

[6] More Sustainable Computing for a More Sustainable Future https://www.intel.com/content/www/us/en/environment/sustainability.html. Accessed 18 June 2024.

[7] Track and reduce CO2 emissions from your computing: https://codecarbon.io/. Accessed 14 June 2024.

Table 16.1: Estimated energy consumption per operational cycle

Code Structure	GPU Usage (kWh)	CPU Usage (kWh)	Total Power Usage (kWh)
Auction Algorithm	0	0.000024	0.000034
Local Trajectory Planning	0	0.012014	0.018037
Volume Measurement Pipeline	0.016024	0.000024	0.019307
30 min. YOLO inference	0.005100	0.017010	0.026100
Total	0.021124	0.029072	0.063478

16.5. Life Cycle Analysis

Life cycle analysis is a tool to identify and quantify environmental impact through all stages of a product cycle [8] Especially for drone operations, this analysis is lacking from an operational implementation perspective due to their recent widespread implementation [96]. This detailed analysis consists of cycles from raw material extraction to end-of-life solution, but also developmental and manufacturing steps. To address this gap, a preliminary analysis of the life cycle for Cargonaut is shown on Figure 16.4. For the power breakdown of all subsystems, refer to Table 15.2.

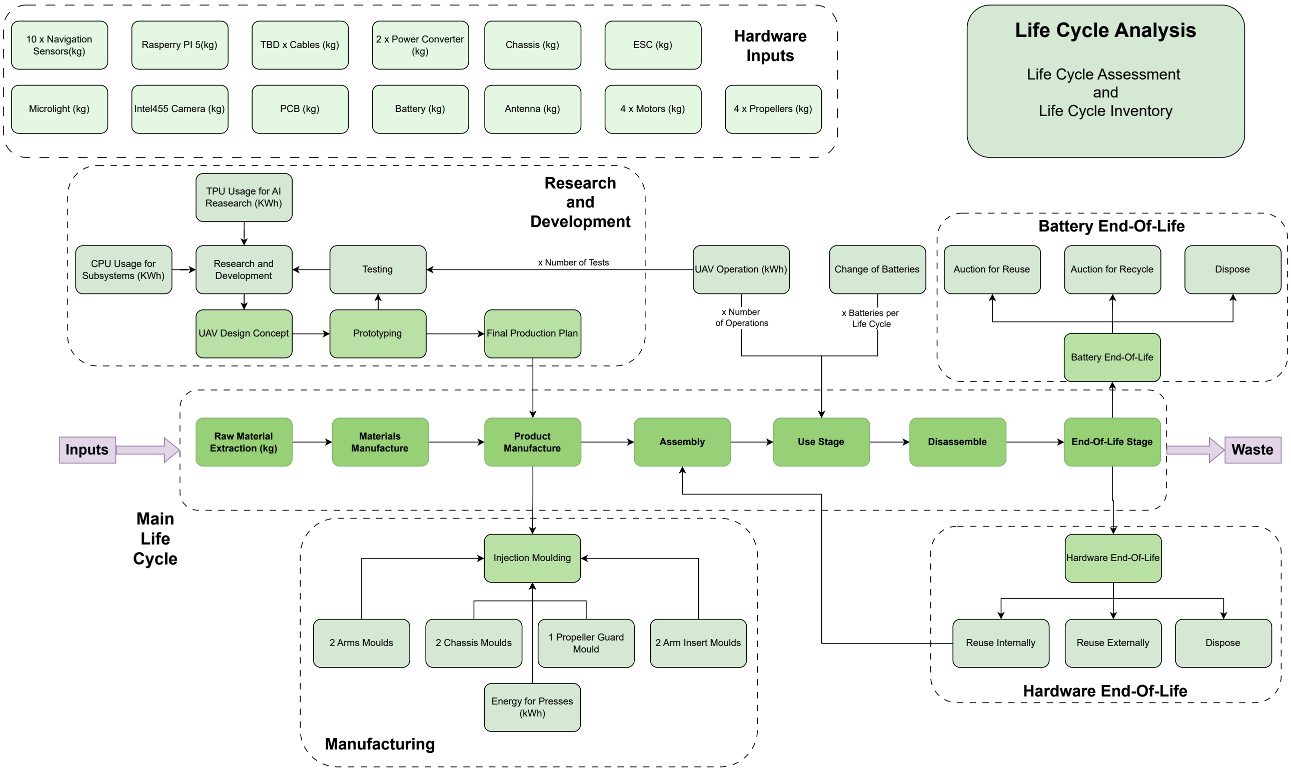


Figure 16.4: Life cycle analysis of Cargonaut

The most wasteful side of all life cycles are raw material extractions and part manufacturing [97]. Selectively choosing companies that aspire for sustainability for the supply of raw materials is envisioned to combat this challenge. Moreover, choosing additive manufacturing during part production to reduce waste produced is a conscious choice in terms of reducing waste. For the operational cycle of the drone, all operational algorithms and hardware have been designed optimally to reduce power consumption.

The bottleneck of the main life cycle is the end-of-life for the battery, given that the battery is the most toxic element within the component breakdown. Three alternatives have been proposed to counter this disadvantage, namely, reuse of the battery by other companies, recycling the batteries, or pure disposal which is the most polluting out of the three. The same idea was also implemented for hardware end-of-life, yet it is evident that except for the batteries other hardware will last more than the envisioned lifetime of the drone.

Given that the design is still primitive, it is not feasible to give a complete numerical overview of the life cycle except for determining cycles that feed into it. To quantify the life cycle, a similar drone [98] was taken into consideration to illuminate a rough estimate for Cargonaut. It is reported that the operations take up 35% of the life cycle pollution whereas 65% belongs to parts manufacturing. From the hardware point of view, 92% of the pollution is derived from batteries which is expected given that this is the most toxic part of a product's life cycle.

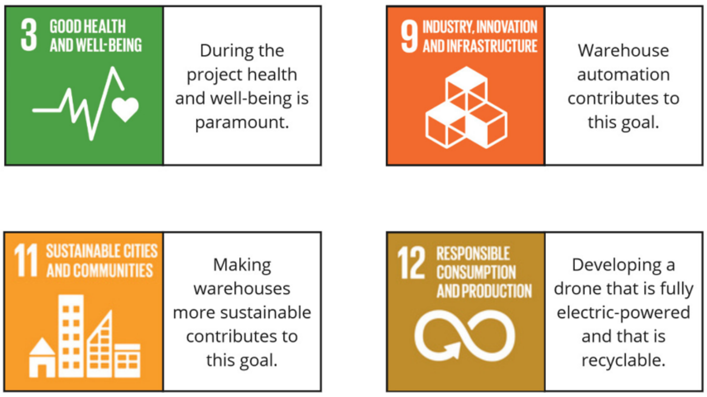
[8] What is LCA? https://www.rivm.nl/en/life-cycle-assessment-lca/what-is-lca. Accessed 18 June 2024.

Per Figure 7 on F. Pollet [98], one hallmark of this analysis was that the operations cycle uses the most non-renewable energy out of every other component whereas the battery was the most carcinogenic component that directly impacts human health. Both of these elements were found to be equally damaging to soil quality. The most critical point made was that the drawbacks of the battery usage steadily increase per 500 cycles which is critical to the operations of Cargonaut as battery replacements per month, as discussed in Section 12.3.1 are envisioned to assure the demanding throughput that is requested.

16.6. Mitigation Plan

To incorporate all of these in action points, Cargonaut will be implementing a mitigation plan. This will be a parallel, in-the-loop document that is intended to serve as a checklist for compliance with the promises that Cargonaut makes. This plan includes referring back to the operationalization [9]. of values that are being held and tracing whether the design choice can be monitored using the indicators. Checking whether approaches surrounding data handling respect national and international regulations and internal checkpoints to see whether the sustainable development policy is respected is paramount, thus progress up until that point will be recorded, and identifying what went wrong is also an additional non-confirmatory report is a key action that will be taken.

To align with the values that the mission entails, the management of Cargonaut believes that upholding the United Nations Sustainable Development Goals (UNSDGs) is a must. This approach not only increases trust in the generated sociotechnical system for the consideration of sustainable development but also provides a systematic way of analyzing which constitutive elements Cargonaut has to look out for while in the design phase. The immediate UNSDGs that align with can be visualized on Figure 16.5. These goals are ensured to be linked to indicators in the design phase to operationalize sustainability in general. More goal orientation can be made possible should the Cargonaut management decide to expand, for example by promoting awareness campaigns, stimulating sustainable innovation by parenting, and guiding start-ups should it become a considerable market player.


3 GOOD HEALTH AND WELL-BEING
9 INDUSTRY, INNOVATION AND INFRASTRUCTURE
11 SUSTAINABLE CITIES AND COMMUNITIES
12 RESPONSIBLE CONSUMPTION AND PRODUCTION

During the project health and well-being is paramount.

Warehouse automation contributes to this goal.

Making warehouses more sustainable contributes to this goal.

Developing a drone that is fully electric-powered and that is recyclable.

Figure 16.5: Sustainable development goals and project relevance

16.7. Recommendations

Throughout the process sustainable practices needed to be upheld yet this was not always the case given the complexity and scarcity of resources. Firstly, not all development emissions were tracked due to infeasible applications, such as tracking CAD development - in future interactions, a way can be devised to achieve this as well. Consequently, documentation on sustainability practices could be streamlined and standardized to incorporate them in the development so as not to hinder progress and allow for sustainable habits to develop.

From the hardware side, choosing a more power-efficient battery and propellers would be the key in ensuring sustainability. The use of SLAM is very resource intensive, as it requires simultaneous computations which are expensive to compute on board; different localization methods with provisional maps could be a more sustainable implementation. Operations-wise, if scanning was possible without being stationary, then less time could be spent per cluster and the need for drones per warehouse would decrease, increasing the overall ecological footprint.

From the software side, an AI model such as Spiking Neural Networks with their significantly higher energy efficiency compared to traditional models [49] should be considered. This is also applicable for the label reading, given that additional hardware might be more sustainable in the long run given the computationally costly training sequences.

[9] Operationalism: https://plato.stanford.edu/entries/operationalism/. Accessed 21 May 2024.

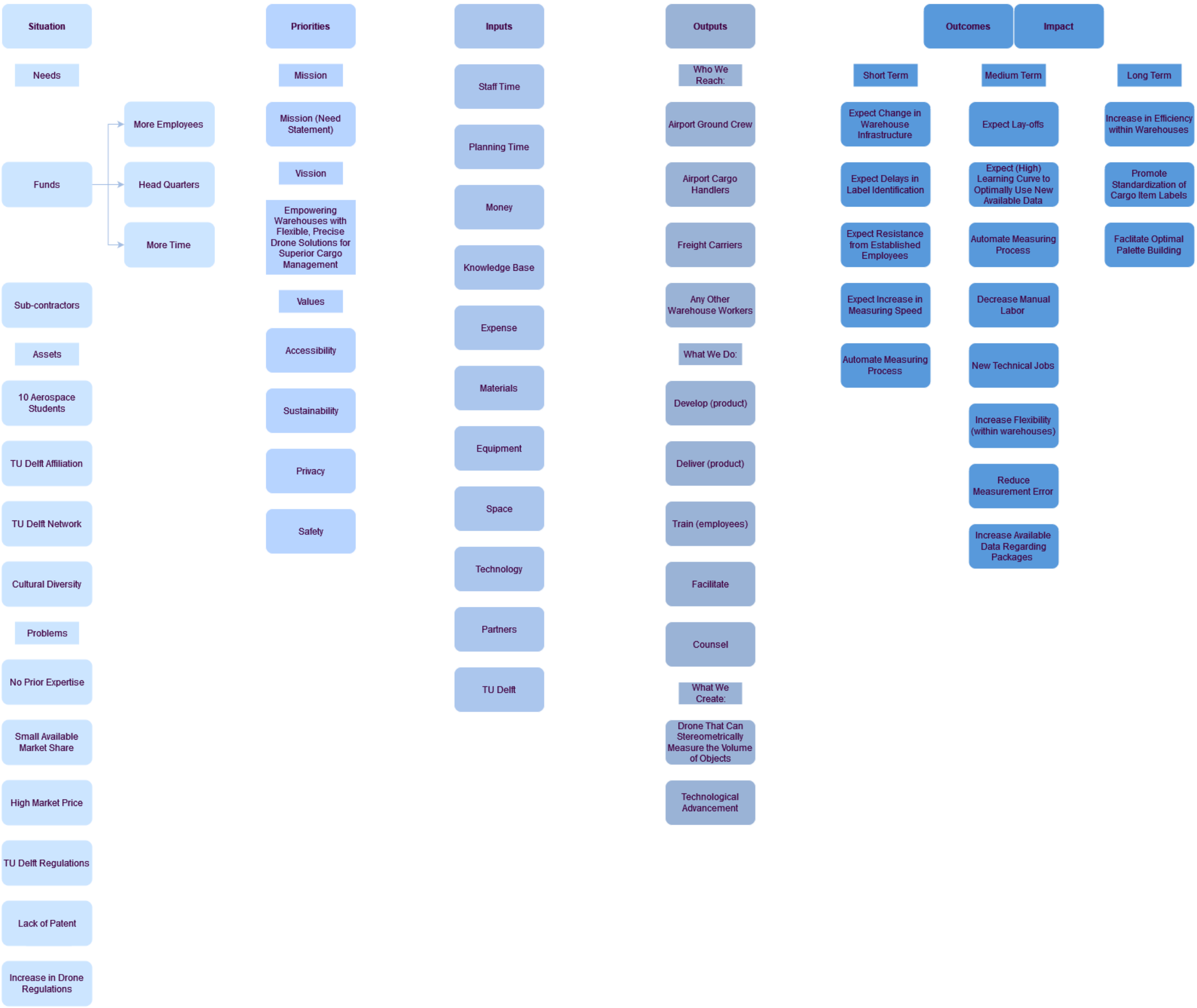
17: Post-DSE Activities

This chapter includes information about activities that are planned after the Design Synthesis Exercise is concluded. The chapter introduces the project logic in Section 17.1 and illustrates steps taken to commercialize and improve the solution in Section 17.2. The breakdown structure, workflow diagram, and Gantt chart are then presented afterward.

17.1. Project Logic Map

“Project logic provides the basis for planning and implementing, monitoring, and evaluating projects.” [99]. Even though the design process is finalized, there is further work to be done in terms of improving the product, generating new iterations, integrating it into the market, and building up the future company. Therefore, the project logic baseplate has been generated for post-DSE activities to align the activities that have been conducted during the DSE with the planned activities after the exercise.

The approach that was followed was to analyze the situation, identify priorities, generate a logical flow of the impacts, provide outputs that the product is envisioned to generate, and provide outcomes and their generated impact in the short, medium, and long term. These can be visualized on Figure 17.1



Situation
Needs
Funds
More Employees
Head Quarters
More Time
Sub-contractors
Assets
10 Aerospace Students
TU Delft Affiliation
TU Delft Network
Cultural Diversity
Problems
No Prior Expertise
Small Available Market Share
High Market Price
TU Delft Regulations
Lack of Patent
Increase in Drone Regulations
Priorities
Mission
Mission (Need Statement)
Vission
Empowering Warehouses with Flexible, Precise Drone Solutions for Superior Cargo Management
Values
Accessibility
Sustainability
Privacy
Safety
Inputs
Staff Time
Planning Time
Money
Knowledge Base
Expense
Materials
Equipment
Space
Technology
Partners
TU Delft
Outputs
Who We Reach:
Airport Ground Crew
Airport Cargo Handlers
Freight Carriers
Any Other Warehouse Workers
What We Do:
Develop (product)
Deliver (product)
Train (employees)
Facilitate
Counsel
What We Create:
Drone That Can Stereometrically Measure the Volume of Objects
Technological Advancement
Outcomes
Impact
Short Term
Expect Change in Warehouse Infrastructure
Expect Delays in Label Identification
Expect Resistance from Established Employees
Expect Increase in Measuring Speed
Automate Measuring Process
Medium Term
Expect Lay-offs
Expect (High) Learning Curve to Optimally Use New Available Data
Automate Measuring Process
Decrease Manual Labor
New Technical Jobs
Increase Flexibility (within warehouses)
Reduce Measurement Error
Increase Available Data Regarding Packages
Long Term
Increase in Efficiency within Warehouses
Promote Standardization of Cargo Item Labels
Faclitate Optimal Palette Building

Figure 17.1: Project logic map

This map provides a rationale for the actions that are planned for the future and assures that these align with the goal that was established at the start of DSE. Adhering to this systematic approach would also allow for changes to be made in the structure of the organization, define renewed goals, and refine sustainability approaches.

Following this, a work flow diagram and work breakdown structure have been generated for reference. These preliminary diagrams help determine critical events to plan for in the future, and establish a logical flow of operations for Cargonaut. These diagrams provide a sound base for further developmental and organizational activities and ensure a smooth transition from the DSE to corporate life.

The Gantt chart shows the schedule for activities scheduled post DSE, limited only until January 1 2027 for better visualization. As it can be seen, some actions are left not concluded and are, in general, not positioned with certainty in the future. The reasons for these are presented here.

- **Provide Customer Consultation:** consultations will happen throughout, as Cargonaut values customer support before, during, and after the purchase of the product.
- **Deploy:** deployment occurs post-manufacturing, following personnel training and the issuance of maintenance plans and guides. Deployment steps are scheduled sequentially with a testing period lasting one week. Deployments coincide with new product orders.
- **Maintain:** scheduled maintenance occurs regularly, with swift repairs for any issues. End-of-life solutions are provided upon product being non-fixable, customer upgrade requests, or when no longer needed.
- **Expand in Other Geographies and/or Markets:** planned to happen from early 2026 onwards, depending on stable high revenues. The timeline ranges from three months to two years, involving consultation with legal, financial, and consulting firms.
- **Expand Product Line:** Also planned for 2026, dependent on product planning and regulatory compliance. Currently scheduled for half a year, execution timelines vary from three months to several years depending on product complexity.
- **Go Public:** possibly beginning in 2027, it requires a strong industry presence and sound financial strategy. If decided, this process may take up to a year, involving legal, financial, and consulting firms.
- **Engage with Community:** ideally, a global reach and stable revenue would support strong links and community engagement programs. Cargonaut's commitment to sustainability and ethical practices translates to ongoing community interactions without a specified end date.

17.2. Market Integration

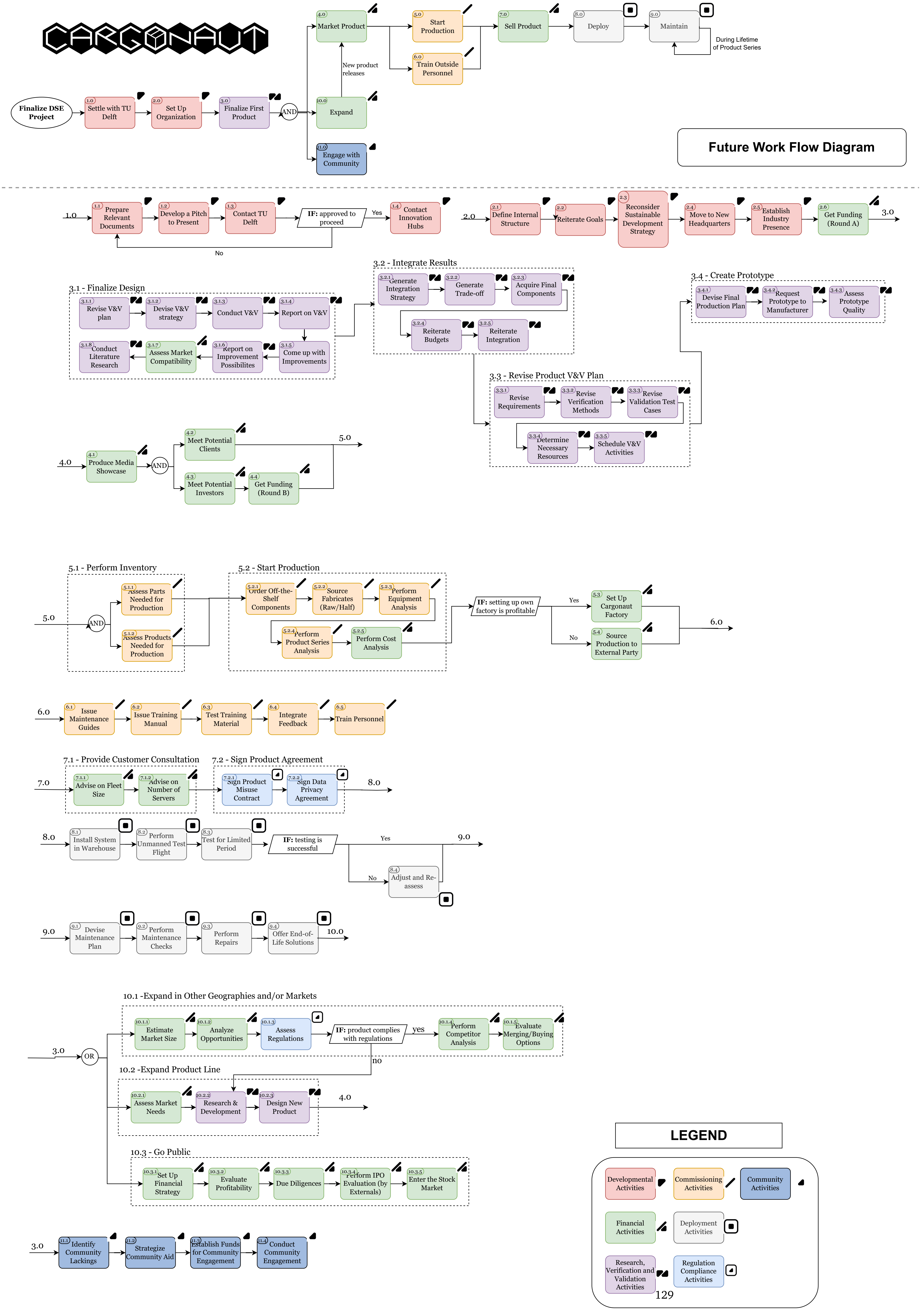
A paramount part of Project Logic is market integration, as it provides a competitive advantage and chances to maximize market potentials. The integration can be further divided into two main components: entry strategy, and growth strategy.

Cargonaut enters the market as a premium solution, offering superior flexibility, accuracy, and efficiency compared to existing measurement systems. The initial focus is on airports and cargo terminals with high throughput and demand for enhanced operational efficiency: an example could be the KLM Cargo warehouses at Schiphol Airport. To maximize market penetration, a dual strategy will be employed: direct sales, and partnerships. The former entails establishing direct relationships with airports, freight carriers, and logistics providers to showcase improved efficiency and reduced operational costs. To ensure customer retention, Cargonaut plans on providing proactive customer support via timely assistance and training. Furthermore, via value-added services, such as providing end-of-life solutions, the team plans enhancing customer satisfaction and long-term relationships.

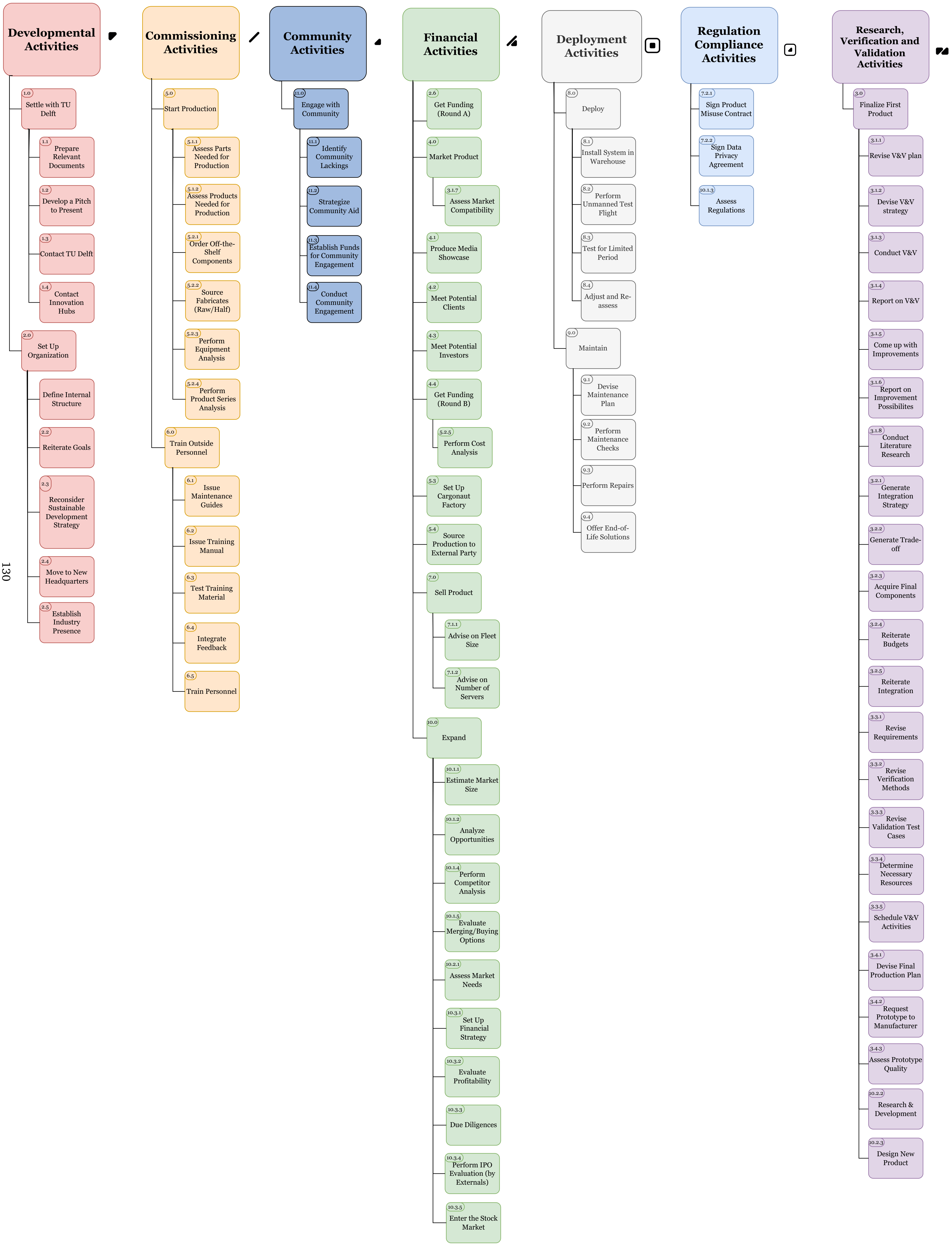
Cargonaut's long-term growth strategy is anchored in innovation, market expansion, and continuous improvements. Cargonaut aims to capitalize on emerging trends, expand its market presence, and solidify its position as a leader in the logistics and aviation sectors. The team hopes to obtain global reach by expanding in different geographies via product adaptation and alignment with different regulations and preferences. Furthermore, multiple sectors can be conquered thanks to the versatility of the product and of the technology used.

Risk analysis and mitigation will also be assessed, as markets and industries are inherently volatile. Four categories of risks have been identified at this time. These are here presented, together with mitigation strategies:

- **Economical and financial risks:** fluctuations in economic conditions, currency exchange rates, and availability of funding can impact financial stability and growth projections.
  Mitigation: monitoring market trends, and diversification of revenue streams.
- **Market acceptance and competition:** competitive pressures and customer preferences may affect market penetration and product adoption rates.
  Mitigation: market research, customer feedback, branding strategy.
- **Operational risks:** potential disruptions in the supply chain, technology failures, or operational inefficiencies could impact service delivery and customer satisfaction.
  Mitigation: quality assurance, a diversified supplier base.
- **Regulatory compliance:** non-compliance with ever-changing local, national, or international regulations could result in fines, legal issues, or barriers to market entry.
  Mitigation: compliance monitoring, legal counsel, and advisory.

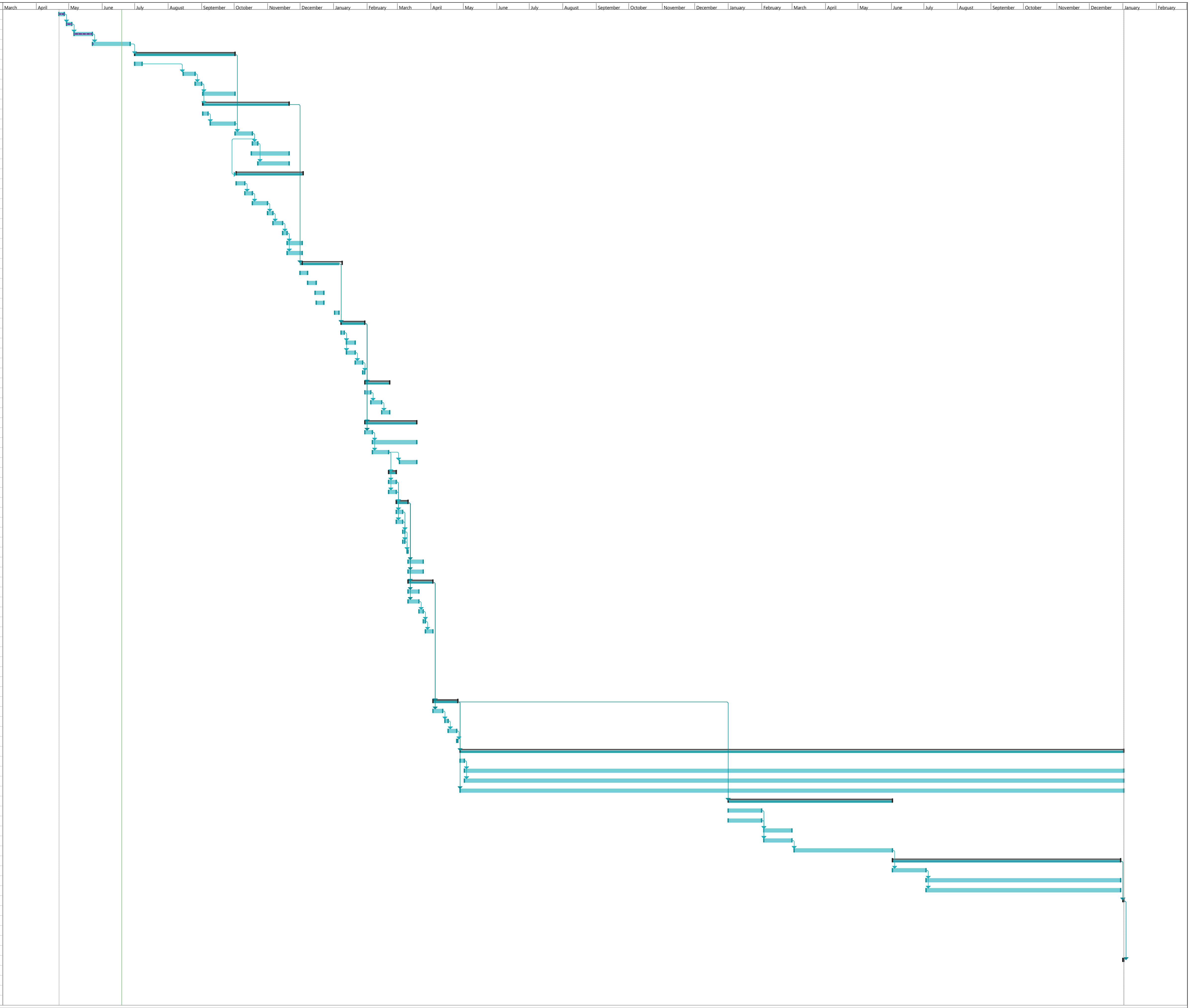

CARGONAUT
Future Work Flow Diagram
Finalize DSE Project
1.0 Settle with TU Delft
2.0 Set Up Organization
3.0 Finalize First Product
AND
4.0 Market Product
5.0 Start Production
6.0 Train Outside Personnel
7.0 Sell Product
8.0 Deploy
9.0 Maintain
During Lifetime of Product Series
New product releases
10.0 Expand
11.0 Engage with Community
1.0
1.1 Prepare Relevant Documents
1.2 Develop a Pitch to Present
1.3 Contact TU Delft
IF: approved to proceed
Yes
No
1.4 Contact Innovation Hubs
2.0
2.1 Define Internal Structure
2.2 Reiterate Goals
2.3 Reconsider Sustainable Development Strategy
2.4 Move to New Headquarters
2.5 Establish Industry Presence
2.6 Get Funding (Round A)
3.0
3.1 - Finalize Design
3.1.1 Revise V&V plan
3.1.2 Devise V&V strategy
3.1.3 Conduct V&V
3.1.4 Report on V&V
3.1.5 Come up with Improvements
3.1.6 Report on Improvement Possibilites
3.1.7 Assess Market Compatibility
3.1.8 Conduct Literature Research
3.2 - Integrate Results
3.2.1 Generate Integration Strategy
3.2.2 Generate Trade-off
3.2.3 Acquire Final Components
3.2.4 Reiterate Budgets
3.2.5 Reiterate Integration
3.3 - Revise Product V&V Plan
3.3.1 Revise Requirements
3.3.2 Revise Verification Methods
3.3.3 Revise Validation Test Cases
3.3.4 Determine Necessary Resources
3.3.5 Schedule V&V Activities
3.4 - Create Prototype
3.4.1 Devise Final Production Plan
3.4.2 Request Prototype to Manufacturer
3.4.3 Assess Prototype Quality
4.0
4.1 Produce Media Showcase
AND
4.2 Meet Potential Clients
4.3 Meet Potential Investors
4.4 Get Funding (Round B)
5.0
5.0
5.1 - Perform Inventory
AND
5.1.1 Assess Parts Needed for Production
5.1.2 Assess Products Needed for Production
5.2 - Start Production
5.2.1 Order Off-the-Shelf Components
5.2.2 Source Fabricates (Raw/Half)
5.2.3 Perform Equipment Analysis
5.2.4 Perform Product Series Analysis
5.2.5 Perform Cost Analysis
IF: setting up own factory is profitable
Yes
No
5.3 Set Up Cargonaut Factory
5.4 Source Production to External Party
6.0
6.0
6.1 Issue Maintenance Guides
6.2 Issue Training Manual
6.3 Test Training Material
6.4 Integrate Feedback
6.5 Train Personnel
7.1 - Provide Customer Consultation
7.0
7.1.1 Advise on Fleet Size
7.1.2 Advise on Number of Servers
7.2 - Sign Product Agreement
7.2.1 Sign Product Misuse Contract
7.2.2 Sign Data Privacy Agreement
8.0
8.0
8.1 Install System in Warehouse
8.2 Perform Unmanned Test Flight
8.3 Test for Limited Period
IF: testing is successful
Yes
9.0
No
8.4 Adjust and Re-assess
9.0
9.1 Devise Maintenance Plan
9.2 Perform Maintenance Checks
9.3 Perform Repairs
9.4 Offer End-of-Life Solutions
10.0
10.1 -Expand in Other Geographies and/or Markets
3.0
OR
10.1.1 Estimate Market Size
10.1.2 Analyze Opportunities
10.1.3 Assess Regulations
IF: product complies with regulations
yes
no
10.1.4 Perform Competitor Analysis
10.1.5 Evaluate Merging/Buying Options
10.2 -Expand Product Line
10.2.1 Assess Market Needs
10.2.2 Research & Development
10.2.3 Design New Product
4.0
10.3 - Go Public
10.3.1 Set Up Financial Strategy
10.3.2 Evaluate Profitability
10.3.3 Due Diligences
10.3.4 Perform IPO Evaluation (by Externals)
10.3.5 Enter the Stock Market
3.0
11.1 Identify Community Lackings
11.2 Strategize Community Aid
11.3 Establish Funds for Community Engagement
11.4 Conduct Community Engagement
LEGEND
Developmental Activities
Commissioning Activities
Community Activities
Financial Activities
Deployment Activities
Research, Verification and Validation Activities
Regulation Compliance Activities

Future Work Breakdown Structure



- **Developmental Activities**
  - 1.0 Settle with TU Delft
    - 1.1 Prepare Relevant Documents
    - 1.2 Develop a Pitch to Present
    - 1.3 Contact TU Delft
    - 1.4 Contact Innovation Hubs
  - 2.0 Set Up Organization
    - Define Internal Structure
    - 2.2 Reiterate Goals
    - 2.3 Reconsider Sustainable Development Strategy
    - 2.4 Move to New Headquarters
    - 2.5 Establish Industry Presence
- **Commissioning Activities**
  - 5.0 Start Production
    - 5.1.1 Assess Parts Needed for Production
    - 5.1.2 Assess Products Needed for Production
    - 5.2.1 Order Off-the-Shelf Components
    - 5.2.2 Source Fabricates (Raw/Half)
    - 5.2.3 Perform Equipment Analysis
    - 5.2.4 Perform Product Series Analysis
  - 6.0 Train Outside Personnel
    - 6.1 Issue Maintenance Guides
    - 6.2 Issue Training Manual
    - 6.3 Test Training Material
    - 6.4 Integrate Feedback
    - 6.5 Train Personnel
- **Community Activities**
  - 11.0 Engage with Community
    - 11.1 Identify Community Lackings
    - 11.2 Strategize Community Aid
    - 11.3 Establish Funds for Community Engagement
    - 11.4 Conduct Community Engagement
- **Financial Activities**
  - 2.6 Get Funding (Round A)
  - 4.0 Market Product
    - 3.1.7 Assess Market Compatibility
  - 4.1 Produce Media Showcase
  - 4.2 Meet Potential Clients
  - 4.3 Meet Potential Investors
  - 4.4 Get Funding (Round B)
    - 5.2.5 Perform Cost Analysis
  - 5.3 Set Up Cargonaut Factory
  - 5.4 Source Production to External Party
  - 7.0 Sell Product
    - 7.1.1 Advise on Fleet Size
    - 7.1.2 Advise on Number of Servers
  - 10.0 Expand
    - 10.1.1 Estimate Market Size
    - 10.1.2 Analyze Opportunities
    - 10.1.4 Perform Competitor Analysis
    - 10.1.5 Evaluate Merging/Buying Options
    - 10.2.1 Assess Market Needs
    - 10.3.1 Set Up Financial Strategy
    - 10.3.2 Evaluate Profitability
    - 10.3.3 Due Diligences
    - 10.3.4 Perform IPO Evaluation (by Externals)
    - 10.3.5 Enter the Stock Market
- **Deployment Activities**
  - 8.0 Deploy
    - 8.1 Install System in Warehouse
    - 8.2 Perform Unmanned Test Flight
    - 8.3 Test for Limited Period
    - 8.4 Adjust and Re-assess
  - 9.0 Maintain
    - 9.1 Devise Maintenance Plan
    - 9.2 Perform Maintenance Checks
    - 9.3 Perform Repairs
    - 9.4 Offer End-of-Life Solutions
- **Regulation Compliance Activities**
  - 7.2.1 Sign Product Misuse Contract
  - 7.2.2 Sign Data Privacy Agreement
  - 10.1.3 Assess Regulations
- **Research, Verification and Validation Activities**
  - 3.0 Finalize First Product
    - 3.1.1 Revise V&V plan
    - 3.1.2 Devise V&V strategy
    - 3.1.3 Conduct V&V
    - 3.1.4 Report on V&V
    - 3.1.5 Come up with Improvements
    - 3.1.6 Report on Improvement Possibilities
    - 3.1.8 Conduct Literature Research
    - 3.2.1 Generate Integration Strategy
    - 3.2.2 Generate Trade-off
    - 3.2.3 Acquire Final Components
    - 3.2.4 Reiterate Budgets
    - 3.2.5 Reiterate Integration
    - 3.3.1 Revise Requirements
    - 3.3.2 Revise Verification Methods
    - 3.3.3 Revise Validation Test Cases
    - 3.3.4 Determine Necessary Resources
    - 3.3.5 Schedule V&V Activities
    - 3.4.1 Devise Final Production Plan
    - 3.4.2 Request Prototype to Manufacturer
    - 3.4.3 Assess Prototype Quality
    - 10.2.2 Research & Development
    - 10.2.3 Design New Product

ID	Task Name	Duration	Start	Finish
1	**Phase 1: Project Planning**	**40 hrs**	**Mon 22-4-24**	**Fri 26-4-24**
2	**Phase 2: Conceptual Design**	**40 hrs**	**Mon 29-4-24**	**Fri 3-5-24**
3	**Phase 3: Preliminary Design**	**88 hrs**	**Mon 6-5-24**	**Wed 22-5-24**
4	**Phase 4: Detailed Design**	**200 hrs**	**Thu 23-5-24**	**Wed 26-6-24**
5	**1. Settle with TUDelft**	**536 hrs**	**Mon 1-7-24**	**Tue 1-10-24**
6	1.1 Prepare Relevant Documents	48 hrs	Mon 1-7-24	Sun 7-7-24
7	1.2 Develop a Pitch to Present	64 hrs	Thu 15-8-24	Sun 25-8-24
8	1.3 Contact TUDelft	48 hrs	Mon 26-8-24	Sat 31-8-24
9	1.4 Contact Innovation Hubs	176 hrs	Mon 2-9-24	Tue 1-10-24
10	**2. Set Up Organization**	**464 hrs**	**Mon 2-9-24**	**Wed 20-11-24**
11	2.1 Define Internal Structure	40 hrs	Mon 2-9-24	Fri 6-9-24
12	2.2 Reiterate Goals	136 hrs	Mon 9-9-24	Tue 1-10-24
13	2.3 Reconsider Sustainable Development Strategy	96 hrs	Wed 2-10-24	Thu 17-10-24
14	2.4 Move to New Headquarters	24 hrs	Fri 18-10-24	Tue 22-10-24
15	2.5 Establish Industry Presence	200 hrs	Thu 17-10-24	Wed 20-11-24
16	2.6 Get Funding (Round A)	168 hrs	Wed 23-10-24	Wed 20-11-24
17	**3. 1 Finalize Design**	**352 hrs**	**Thu 3-10-24**	**Tue 3-12-24**
18	3.1.1 Revise V&V Plan	48 hrs	Thu 3-10-24	Thu 10-10-24
19	3.1.2 Devise V&V Strategy	40 hrs	Fri 11-10-24	Thu 17-10-24
20	3.1.3 Conduct V&V	80 hrs	Fri 18-10-24	Thu 31-10-24
21	3.1.4 Report on V&V	24 hrs	Fri 1-11-24	Tue 5-11-24
22	3.1.5 Come up with Improvements	56 hrs	Wed 6-11-24	Thu 14-11-24
23	3.1.6 Report on Impovement Possibilities	16 hrs	Fri 15-11-24	Mon 18-11-24
24	3.1.7 Assess Market Compatibility	80 hrs	Tue 19-11-24	Mon 2-12-24
25	3.1.8 Conduct Literature Research	80 hrs	Tue 19-11-24	Mon 2-12-24
26	**3.2 Integrate Results**	**216 hrs**	**Tue 3-12-24**	**Wed 8-1-25**
27	3.2.1 Generate Integration Strategy	56 hrs	Sun 1-12-24	Sat 7-12-24
28	3.2.2 Generate Trade-off	56 hrs	Sun 8-12-24	Sun 15-12-24
29	3.2.3 Acquire Final Components	56 hrs	Sun 15-12-24	Sun 22-12-24
30	3.2.4 Reiterate Budgets	48 hrs	Mon 16-12-24	Sun 22-12-24
31	3.2.5 Reiterate Integration	24 hrs	Thu 2-1-25	Sun 5-1-25
32	**3.3 Revise Product V&V Plan**	**128 hrs**	**Wed 8-1-25**	**Wed 29-1-25**
33	3.3.1 Revise Requirements	24 hrs	Wed 8-1-25	Fri 10-1-25
34	3.3.2 Revise Verification Methods	48 hrs	Mon 13-1-25	Mon 20-1-25
35	3.3.3 Revise Validation Test Cases	48 hrs	Mon 13-1-25	Mon 20-1-25
36	3.3.4 Determine Necessary Resources	40 hrs	Tue 21-1-25	Mon 27-1-25
37	3.3.5 Schedule V&V Activities	16 hrs	Tue 28-1-25	Wed 29-1-25
38	**3.4 Create Prototype**	**136 hrs**	**Thu 30-1-25**	**Fri 21-2-25**
39	3.4.1 Devise Final Production Plan	28 hrs	Thu 30-1-25	Tue 4-2-25
40	3.4.2 Request Prototype to Manufacturer	64 hrs	Tue 4-2-25	Fri 14-2-25
41	3.4.3 Assess Prototype Quality	44 hrs	Fri 14-2-25	Fri 21-2-25
42	**4. Market Product**	**272 hrs**	**Thu 30-1-25**	**Tue 18-3-25**
43	4.1 Produce Media Showcase	40 hrs	Thu 30-1-25	Wed 5-2-25
44	4.2 Meet Potential Clients	232 hrs	Thu 6-2-25	Tue 18-3-25
45	4.3 Meet Potential Investors	88 hrs	Thu 6-2-25	Thu 20-2-25
46	4.4 Get Funding (Round B)	96 hrs	Mon 3-3-25	Tue 18-3-25
47	**5.1 Perform Inventory**	**40 hrs**	**Fri 21-2-25**	**Thu 27-2-25**
48	5.1.1 Assess Parts Needed for Production	40 hrs	Fri 21-2-25	Thu 27-2-25
49	5.1.2 Assess Products Needed for Production	40 hrs	Fri 21-2-25	Thu 27-2-25
50	**5.2 Start Production**	**56 hrs**	**Fri 28-2-25**	**Mon 10-3-25**
51	5.2.1 Order Off-the-Shelf Components	32 hrs	Fri 28-2-25	Wed 5-3-25
52	5.2.2 Source Fabricates (Raw/Half)	32 hrs	Fri 28-2-25	Wed 5-3-25
53	5.2.3 Perform Equipment Analysis	16 hrs	Thu 6-3-25	Fri 7-3-25
54	5.2.4 Perform Product Series Analysis	16 hrs	Thu 6-3-25	Fri 7-3-25
55	5.2.5 Perform Cost Analysis	8 hrs	Mon 10-3-25	Mon 10-3-25
56	**5.3 Set Up Cargonaut Factory**	**80 hrs**	**Tue 11-3-25**	**Mon 24-3-25**
57	**5.4 Source Production to External Party**	**80 hrs**	**Tue 11-3-25**	**Mon 24-3-25**
58	**6. Train Outside Personnel**	**136 hrs**	**Tue 11-3-25**	**Wed 2-4-25**
59	6.1 Issue Maintenance Guides	64 hrs	Tue 11-3-25	Thu 20-3-25
60	6.2 Issue Training Manual	64 hrs	Tue 11-3-25	Thu 20-3-25
61	6.3 Test Training Manual	16 hrs	Fri 21-3-25	Mon 24-3-25
62	6.4 Integrate Feedback	16 hrs	Tue 25-3-25	Wed 26-3-25
63	6.5 Train Personnel	40 hrs	Thu 27-3-25	Wed 2-4-25
64	**7.1 Provide Customer Consultation**	**16 hrs**	**TBD**	**TBD**
65	7.1.1 Advise on Fleet Size		TBD	TBD
66	7.1.2 Advise on Number of Servers		TBD	TBD
67	**7.2 Sign Product Agreement**	**8 hrs**	**TBD**	**TBD**
68	7.2.1 Sign Product Misuse Contract		TBD	TBD
69	7.2.2 Sign Data Privacy Agreement		TBD	TBD
70	**8. Deploy**	**136 hrs**	**Thu 3-4-25**	**Fri 25-4-25**
71	8.1 Install System in Warehouse	56 hrs	Thu 3-4-25	Fri 11-4-25
72	8.2 Perform Unmanned Test Flight	24 hrs	Mon 14-4-25	Wed 16-4-25
73	8.3 Test for Limited Period	48 hrs	Thu 17-4-25	Thu 24-4-25
74	8.4 Adjust and Reassess	8 hrs	Fri 25-4-25	Fri 25-4-25
75	**9. Maintain**		**Mon 28-4-25**	**Fri 1-1-27**
76	9.1 Devise Maintenance Plan	32 hrs	Mon 28-4-25	Thu 1-5-25
77	9.2 Perform Maintenance Checks		Fri 2-5-25	Fri 1-1-27
78	9.3 Perform Repairs		Fri 2-5-25	Fri 1-1-27
79	**9.4 Offer End-of-Life Solutions**		**Mon 28-4-25**	**Fri 1-1-27**
80	**10.1 Expand in Other Geographies and/or Markets**	**864 hrs**	**Thu 1-1-26**	**Mon 1-6-26**
81	10.1.1 Estimate Market Size	184 hrs	Thu 1-1-26	Sat 31-1-26
82	10.1.2 Analyze Opportunities	184 hrs	Thu 1-1-26	Sat 31-1-26
83	10.1.3 Assess Regulations	160 hrs	Tue 3-2-26	Sat 28-2-26
84	10.1.4 Perform Competitor Analysis	160 hrs	Tue 3-2-26	Sat 28-2-26
85	10.1.5 Evaluate Merging/Buying Opportunities	520 hrs	Tue 3-3-26	Mon 1-6-26
86	**10.2 Expand Product Line**	**1208 hrs**	**Tue 2-6-26**	**Tue 29-12-26**
87	10.2.1 Assess Market Needs	184 hrs	Tue 2-6-26	Thu 2-7-26
88	10.2.2 Research & Development	1024 hrs	Fri 3-7-26	Tue 29-12-26
89	10.2.3 Design New Product	1024 hrs	Fri 3-7-26	Tue 29-12-26
90	**10.3 Go Public**		**Fri 1-1-27**	**Fri 1-1-27**
91	10.3.1 Set up Financial Strategy			
92	10.3.2 Evaluate Profitability			
93	10.3.3 Due Diligences			
94	10.3.4 Perform IPo Evaluation (by Externals)			
95	10.3.5 Enter the Stock Market			
96	**11. Engage with Community**		**Fri 1-1-27**	**Fri 1-1-27**
97	11.1 Identify Community Lackings			
98	11.2 Strategize Community Aid			
99	11.3 Establish Funds for Community Engagement			
100	11.4 Conduct Community Engagement			



Project: DSE CARGONAUT
Date: Wed 19-6-24

Task	🟧	Milestone	◆	Project Summary	⊓	Inactive Milestone	◇	Manual Task	▬	Manual Summary Rollup	▬	Start-only	[	External Tasks	▬	Deadline	⬇
Split	⋯	Summary	⊓	Inactive Task	▭	Inactive Summary	⊓	Duration-only	▬	Manual Summary	⊓	Finish-only	]	External Milestone	◇	Progress	▬

Manual Progress ▬

18: Conclusion

The completion of this detailed design stage also marks the end of Cargonaut within the Design Synthesis Exercise (DSE). The development of such a complex system as an autonomous drone is never easy, especially under the strict time constraint of ten weeks. Nonetheless, bringing ten passionate students together proved to form a nurturing environment in which creative ideas were born, shaped, and put into action. The fundamental core of aerospace was examined through the prism of project management and systems engineering, representative of true work within the industry. To ensure that everyone who contributed to Cargonaut gets the most out of this great learning opportunity, it is necessary to look back and reflect on the work done.

Naturally, the first step in developing a competitive product is to conduct market analysis. Despite the lack of experience in this field, the team successfully managed to identify issues in cargo loading operations, discover a market gap, and exploit this opportunity. To gain a better perspective of the domain of the work and the operational environment, it was crucial to examine the setting of a typical warehouse. The KLM Schiphol Cargo Hub was the main source of information, allowing the team to create a model that correctly represents reality. However, this information came in increments and at different times, based on meetings with experts and online resources. Ideally, this would have been unified and conducted at an earlier stage, to have a starting point with a clear vision in mind. Nonetheless, despite some conceptual stumbling blocks on the way, the team is satisfied with the final product. Flexible, autonomous, and dynamic, it has the potential to tackle the air cargo load planning problem at its core.

To reach this goal, a crucial task was to devise all the functions that a drone needs to perform. These then flowed into the conceptualization of requirements on what Cargonaut shall do and risks that these requirements would not be met. This process had to be extensive, and entailed a lot of bookkeeping, met with some initial pessimism. Nonetheless, as the design process gained maturity, the importance of these tools stood out. They allowed for traceability and accountability and facilitated the smooth flow of information within the team, not just among individual team members, but also with stakeholders.

Initial design involved trade-offs of preliminary concepts, with obtained solutions reinforced through sensitivity analysis. After they were finalized, the preliminary design phase moved the focus from bigger overhead tasks into separate departments. However, subsystems rely on each other's input and are in constant interaction, with interfaces modeled via an N2 chart. Subsystem divisions were also reflected in technical budgets on mass, power, and cost, including contingency margins. As design became more detailed, unforeseen developments made the team more appreciative of this contingency - despite putting an upper limit on a certain quantity, the fact that it was not strictly fixed allowed for flexibility and broader design space.

This project prioritized software development, crucial for meeting the goal of identifying and stereometrically measuring cargo items. In the department of Operations and Logistics, this translated into an extensive amount of strategizing, both on a local and global scale. Implementations of planning algorithms provide a viable solution, and future analysis could expand the algorithmic domain and introduce higher complexity in warehouse modeling. Regarding the Payload subsystem, this also included obtaining a representative camera, which allowed to gather empirical data and insight into what information each image carries. After developing an AI algorithm for object detection, cardboard boxes were successfully identified and measured. Further developments could consist of research into models that bring more sustainability benefits, such as the energy-efficient spiking neural networks. Navigation mainly provided an overview of Simultaneous Localization And Mapping (SLAM) as a crucial part of collision avoidance and localization. Following a set trajectory could not be possible without the input of Control, which relied on Proportional–Integral–Derivative (PID) controllers to ensure stable flight and precise maneuverability. Then, the software integration and simulation were made a first-order priority. The aforementioned subsystems shared a lot of common instruments and interfaces. This demanded that they would be successfully merged and operate simultaneously, without adversely affecting each other. This was proven via simulation, visualized in Figure 18.1, which yielded satisfactory results for a number of configurations, with benchmarks and data on computational time. In accordance with stakeholder requirements, there is confidence in the statement that Cargonaut will manage to perform a measurement of a cargo item within $30\,\mathrm{s}$, and a fleet will always be capable of reaching a throughput of at least $1{,}000$ items per hour.

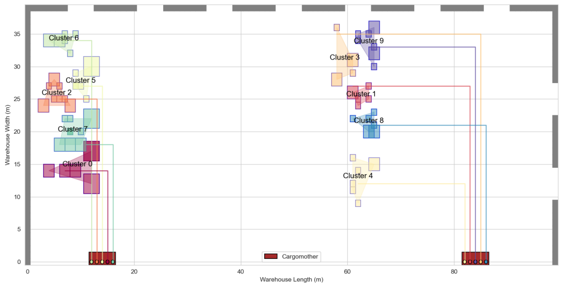


Figure 18.1: Planned trajectories for drones to their assigned clusters, conceptualized after software integration

Despite allocating significant resources to software, the hardware departments were not underdeveloped and also came up with detailed solutions. Confident with the propeller choice, Flight Performance proceeded with CFD for analysis of the airflow. In spite of the steep learning curve, the obtained model yields satisfactory results. Force magnitude is underestimated, but provides a reasonable confidence interval for test cases of interest. With regards to Communications and Data Handling (CDH), it shall be noted that the expected workload was quite underestimated. In reality, it ended up being quite involved, with a focus on bandwidth, protocols, as well as modulation. Future works include the possible implementation of double-band hardware for congestion avoidance. To ensure that all components receive sufficient power, the design of the Electronics subsystem included selecting a battery and accompanying hardware for current transmission. Last-minute revisions led to the discovery that a number of voltage converters would be necessary due to the varying requirements of selected instruments. This consideration is important, leading to a key future takeaway to try and design for the same operational voltage. The physical placement of each component and general overview of the drone were a task dedicated to the Structures and Materials department. The analysis of sustainable materials concluded with the selection of Wood-Carbon Fiber Plastic Composite (WCFC), with the potential to reach 97% recyclability. Detailed design options include 3D modeling and Finite Element Method (FEM) analysis of the applied loads, both on an individual arm and the entire drone assembly. After positive conclusive remarks, the hardware integration was also marked as complete, with the drone visualized in Figure 18.2.

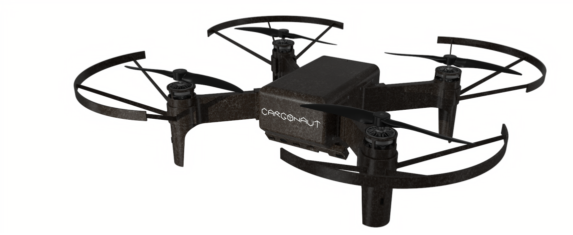


Figure 18.2: Render of Cargonaut drone, with all relevant components present

The purpose of the report was to build upon the previous work packages and converge to a solution that complies with the stakeholder needs. Whereas there are still some requirements that shall be verified, it is important to highlight that compliance has been achieved for most of them. Mitigating risks at a subsystem level and executing Verification and Validation (VnV) has been done in accordance with stakeholder needs, directly meeting them. Incorporating core values within these final results completes the development of Cargonaut within the scope of DSE. However, that does not mean that interest in the project is exhausted. The team members believe that, despite the shared passion, there is no bias in stating that this drone is a viable product that could have a bright future. As a result, strategies have been devised on how to proceed, involving the continuation of Research and Development (RnD) and an updated project logic, including market integration.

To reiterate once again, going back to the start of this project, designing Cargonaut has been a true adventure. The Unmanned Aerial Vehicle (UAV) was made not simply for basic maneuvering or navigation, but encompasses a real purpose and tackles real-life problems. Sometimes battling with a lack of literature, the team had to be resourceful and creative and definitely come up with outside-of-the-box solutions. Challenged and supported by tutor, coaches, and external experts, the team was able to gain confidence in every step of the design and avoid tunnel vision.

Hard work helped Cargonaut turn from a blurry vision into reality. Grateful for all the guidance and dedication, it is now time to prove that this drone is more than capable of revolutionizing cargo handling operations.

Bibliography

[1] Brandt, F., and Nickel, S., “The air cargo load planning problem - a consolidated problem definition and literature review on related problems,” *European Journal of Operational Research*, Vol. 275, 2018. doi: 10.1016/j.ejor.2018.07.013.

[2] Bombelli, A., “AE4446: Airport and Cargo Operations - Air Cargo Operations Lecture Notes,” , 2024.

[3] “ATLAS of Cargo Airports in Europe,” , 2007.

[4] American Institute of Aeronautics and Astronautics, “Mass Properties Control for Space Systems,” , 2015.

[5] Burge, S., “The Systems Engineering Tool Box,” , 2011.

[6] rokia Nathan, A., John, R., Kurmi, I., and Bimber, O., “Drone swarm strategy for the detection and tracking of occluded targets in complex environments,” *Communications Engineering*, Vol. 2, 2023. doi: 10.1038/s44172-023-00104-0.

[7] Reda, M., Onsy, A., Haikal, A. Y., and Ghanbari, A., “Path planning algorithms in the autonomous driving system: A comprehensive review,” *Robotics and Autonomous Systems*, Vol. 174, 2024, p. 104630. doi: https://doi.org/10.1016/j.robot.2024.104630.

[8] Belkadi, A., Abaunza, H., Ciarletta, L., Castillo, P., and Theilliol, D., “Design and Implementation of Distributed Path Planning Algorithm for a Fleet of UAVs,” *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 55, No. 6, 2019, pp. 2647–2657. doi:10.1109/TAES.2019.2906437.

[9] Zlot, R., Stentz, A., Dias, C., Veloso, M., and Balch, T., “An Auction-Based Approach to Complex Task Allocation for Multirobot Teams Thesis Committee,” Ph.D. thesis, 12 2006.

[10] Skaltsis, G. M., “A Review of Task Allocation Methods for UAVs,” *Journal of Intelligent Robotic Systems*, 2023.

[11] Heap, B., and Pagnucco, M., “Sequential Single-Cluster Auctions for Robot Task Allocation,” 2011, pp. 412–421. doi:10.1007/978-3-642-25832-9_42.

[12] Choset, H., and Pignon, P., “Coverage Path Planning: The Boustrophedon Cellular Decomposition,” *Field and Service Robotics*, edited by A. Zelinsky, Springer London, London, 1998, pp. 203–209.

[13] Tan, C., Mohd-Mokhtar, R., and Arshad, M. R., “A Comprehensive Review of Coverage Path Planning in Robotics Using Classical and Heuristic Algorithms,” *IEEE Access*, Vol. 9, 2021, pp. 119310–119342. doi: 10.1109/ACCESS.2021.3108177.

[14] Levin, A., Fergus, R., Durand, F., and Freeman, W. T., “Image and Depth from a Conventional Camera with a Coded Aperture,” 2007.

[15] Kim, T. H., and Lee, K. M., *Shape from Focus and Defocus*, Springer International Publishing, 2021, pp. 1–3. doi:10.1007/978-3-030-03243-2_839-1.

[16] Peng, T., and Gupta, S. K., “Model and algorithms for point cloud construction using digital projection patterns,” *Journal of Computing and Information Science in Engineering*, Vol. 7, 2007, pp. 372–381. doi: 10.1115/1.2798115.

[17] Geng, J., “Structured-light 3D surface imaging: a tutorial,” *Advances in Optics and Photonics*, Vol. 3, 2011, p. 128. doi:10.1364/aop.3.000128.

[18] Prados, E., Faugeras, O., Prados, E., and Faugeras, O., “Shape from Shading: a well-posed problem,” 2005, pp. 870–877. doi:10.1109/CVPR.2005.319ï.

[19] Prados, E., and Faugeras, O., *Shape From Shading*, Springer US, Boston, MA, 2006, pp. 375–388. doi:10.1007/0-387-28831-7_23.

[20] Britannica, T. Editors of Encyclopaedia, “sonar,” , Apr. 2024. Encyclopedia Britannica.

[21] “Broadband 3-D sonar system using a sparse array for indoor navigation,” *IEEE Transactions on Robotics*, Vol. 29, 2013, pp. 161–171. doi: 10.1109/TRO.2012.2221313.

[22] Kerstens, R., Laurijssen, D., and Steckel, J., “eRTIS: A Fully Embedded Real Time 3D Imaging Sonar Sensor for Robotic Applications,” *2019 International Conference on Robotics and Automation (ICRA)*, 2019, pp. 1438–1443. doi:10.1109/ICRA.2019.8794419.

[23] YLIDAR, *YDLIDAROS30A DATA SHEET*, Shenzhen EAI Technology Co.,Ltd, 2022.

[24] Astra, *Astra Mini Pro Datasheet v1.0*, Orbbec Inc., Troy, MI, USA, 2024.

[25] Intel, *Intel® RealSense Product Family D400 Series*, Intel Inc., Santa Clara, California, 2024.

[26] Davies, E., *The three-dimensional world*, Elsevier, 2018, pp. 497–531. doi:10.1016/b978-0-12-809284-2.00016-2.

[27] Fechteler, P., Eisert, P., and Rurainsky, J., “Fast and High Resolution Face Scanning,” , 2007.

[28] Scharstein, D., and Szeliski, R., “High-Accuracy Stereo Depth Maps Using Structured Light,” , 2003.

[29] O’sullivan, P., and Dortz, N. L., “Time of Flight System Design-Part 1: System Overview,” , 2021.

[30] Kalinov, I., Petrovsky, A., Ilin, V., Pristanskiy, E., Kurenkov, M., Ramzhaev, V., Idrisov, I., and Tsetserukou, D., “WareVision: CNN Barcode Detection-Based UAV Trajectory Optimization for Autonomous Warehouse Stocktaking,” *IEEE Robotics and Automation Letters*, Vol. PP, 2020, pp. 1–1. doi:10.1109/LRA.2020.3010733.

[31] luxonis, “OAK-D: Stereo camera with Edge AI,” , 2020. Stereo Camera with Edge AI capabilities from Luxonis and OpenCV.

[32] Liu, S., Yu, Y., Pautrat, R., Pollefeys, M., and Larsson, V., “3D Line Mapping Revisited,” 2023.

[33] Campos, C., Elvira, R., Rodriguez, J. J., Montiel, J. M., and Tardos, J. D., “ORB-SLAM3: An Accurate Open-Source Library for Visual, Visual-Inertial, and Multimap SLAM,” *IEEE Transactions on Robotics*, Vol. 37, 2021, pp. 1874–1890. doi:10.1109/TRO.2021.3075644.

[34] Schönberger, J. L., Zheng, E., Pollefeys, M., and Frahm, J.-M., “Pixelwise View Selection for Unstructured Multi-View Stereo,” *European Conference on Computer Vision (ECCV)*, 2016.

[35] Rusu, R. B., and Cousins, S., “3D is here: Point Cloud Library (PCL),” *IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, Shanghai, China, 2011.

[36] Rusu, R. B., “Semantic 3D Object Maps for Everyday Manipulation in Human Living Environments,” Ph.D. thesis, Computer Science department, Technische Universitaet Muenchen, Germany, October 2009.

[37] Rusu, R. B., Blodow, N., and Beetz, M., “Fast Point Feature Histograms (FPFH) for 3D registration,” *2009 IEEE International Conference on Robotics and Automation*, 2009. doi:10.1109/robot.2009.5152473.

[38] Fischler, M. A., and Bolles, R. C., “Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography,” *Commun. ACM*, Vol. 24, No. 6, 1981, p. 381–395. doi: 10.1145/358669.358692.

[39] Gastal, E. S. L., and Oliveira, M. M., “Domain transform for edge-aware image and video processing,” *ACM Trans. Graph.*, Vol. 30, No. 4, 2011. doi:10.1145/2010324.1964964.

[40] Li, L., Wang, R., and Zhang, X., “A Tutorial Review on Point Cloud Registrations: Principle, Classification, Comparison, and Technology Challenges,” *Mathematical Problems in Engineering*, Vol. 2021, 2021. doi: 10.1155/2021/9953910.

[41] Arun, K. S., Huang, T. S., and Blostein, S. D., “Least-Squares Fitting of Two 3-D Point Sets,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. PAMI-9, No. 5, 1987, pp. 698–700. doi: 10.1109/TPAMI.1987.4767965.

[42] Besl, P., and McKay, N. D., “A method for registration of 3-D shapes,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 14, No. 2, 1992, pp. 239–256. doi:10.1109/34.121791.

[43] Biber, P., and Strasser, W., “The normal distributions transform: a new approach to laser scan matching,” *Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003) (Cat. No.03CH37453)*, Vol. 3, 2003, pp. 2743–2748 vol.3. doi:10.1109/IROS.2003.1249285.

[44] He, Y., Liang, B., Yang, J., Li, S., and He, J., “An Iterative Closest Points Algorithm for Registration of 3D Laser Scanner Point Clouds with Geometric Features,” *Sensors*, Vol. 17, No. 8, 2017. doi:10.3390/s17081862.

[45] Steder, B., Rusu, R., Konolige, K., and Burgard, W., “Point feature extraction on 3D range scans taking into account object boundaries,” 2011, pp. 2601–2608. doi:10.1109/ICRA.2011.5980187.

[46] Rusu, R. B., Blodow, N., Marton, Z. C., and Beetz, M., “Aligning point cloud views using persistent feature histograms,” *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2008, pp. 3384–3391. doi:10.1109/IROS.2008.4650967.

[47] Buch, A., Kraft, D., Kämäräinen, J.-K., Petersen, H., and Kruger, N., "Pose Estimation using Local Structure-Specific Shape and Appearance Context," *2013 IEEE International conference on Robotics and Automation (ICRA), May 6-10, 2013, Karlsruhe, Germany*, 2013. doi:10.1109/ICRA.2013.6630856, contribution: organisation=sgn,FACT1=1<br/>Portfolio EDEND: 2013-07-29<br/>Publisher name: Institute of Electrical and Electronics Engineers IEEE.

[48] HP, "HP ZBook Power G7 Mobile Workstation Specifications," , 2021.

[49] Wu, D., Yi, X., and Huang, X., "A Little Energy Goes a Long Way: Build an Energy-Efficient, Accurate Spiking Neural Network From Convolutional Neural Network," *Frontiers in Neuroscience*, Vol. 16, 2022. doi: 10.3389/fnins.2022.759900.

[50] Xu, W., Cai, Y., He, D., Lin, J., and Fu, Z., "FAST-LIO2: Fast direct lidar-inertial odometry," *IEEE Transactions on Robotics*, Vol. 38, No. 4, 2022, pp. 2053–2073.

[51] Bach, S., Khoi, P., and Yi, S., "Application of QR Code for Localization and Navigation of Indoor Mobile Robot," *IEEE Access*, Vol. 11, January 2023.

[52] Okulski, M., and Ławryńczuk, M., "A Small UAV Optimized for Efficient Long-Range and VTOL Missions: An Experimental Tandem-Wing Quadplane Drone," *Applied Sciences*, Vol. 12, 2022, p. 7059. doi: 10.3390/app12147059.

[53] Vogeltanz, T., "A Survey of Free Software for the Design, Analysis, Modelling, and Simulation of an Unmanned Aerial Vehicle," *Archives of Computational Methods in Engineering*, Vol. 23, 2016, pp. 449–514. doi: 10.1007/s11831-015-9147-y.

[54] Hattenberger, G., Bronz, M., and Condomines, J.-P., "Evaluation of drag coefficient for a quadrotor model," *International Journal of Micro Air Vehicles*, Vol. 15, 2023, p. 175682932211483. doi:10.1177/17568293221148378.

[55] Kang, Y., and hong, S., "Improved Dynamic Modeling and Velocity Loop Augmentation for Quadcopters with estimated Rotor-Drag Coefficient," *International Journal of Control and Automation*, Vol. 11, 2018, pp. 105–116. doi:10.14257/ijca.2018.11.4.10.

[56] Gill, R., and D'Andrea, R., "Propeller thrust and drag in forward flight," , 2017. doi:10.1109/CCTA.2017.8062443.

[57] Sørensen, J. N., "General Momentum Theory for Horizontal Axis Wind Turbines," Vol. 4, 2016. doi:10.1007/978-3-319-22114-4.

[58] Sanderse, B., Pijl, van der, S., and Koren, B., "Review of computational fluid dynamics for wind turbine wake aerodynamics," *Wind Energy*, Vol. 14, No. 7, 2011, pp. 799–819. doi:10.1002/we.458.

[59] Edmunds, M., Williams, A. J., Masters, I., Banerjee, A., and VanZwieten, J. H., "A spatially nonlinear generalised actuator disk model for the simulation of horizontal axis wind and tidal turbines," *Energy*, Vol. 194, No. C, 2020. doi:10.1016/j.energy.2019.116.

[60] Capitao Patrao, A., "Description and validation of the rotorDiskSource class for propeller performance estimation," *Proceedings of CFD with Open-Source Software*, 2017.

[61] Dantsker, O. D., Caccamo, M., Deters, R. W., and Selig, M., "Performance testing of APC electric fixed-blade UAV propellers," *AIAA AVIATION 2022 Forum*, 2022. doi:10.2514/6.2022-4020.

[62] Fei, X., "The Causes of Propeller Pitching Moment and the Conditions for its Significance," Phd thesis, Georgia Institute of Technology, April 2021.

[63] de Vries, R., van Arnhem, N., Sinnige, T., Vos, R., and Veldhuis, L. L., "Aerodynamic interaction between propellers of a distributed-propulsion system in forward flight," *Aerospace Science and Technology*, Vol. 118, 2021, p. 107009. doi:https://doi.org/10.1016/j.ast.2021.107009.

[64] Zanotti, A., "Experimental Study of the Aerodynamic Interaction between Side-by-Side Propellers in eVTOL Airplane Mode through Stereoscopic Particle Image Velocimetry," *Aerospace 2021, Vol. 8, Page 239*, Vol. 8, 2021, p. 239. doi:10.3390/AEROSPACE8090239.

[65] Stergiannis, N., Beeck, J., and Runacres, M., "Full HAWT rotor CFD simulations using different RANS turbulence models compared with actuator disk and experimental measurements," *Wind Energy Science*, 2017. doi:10.5194/wes-2017-6.

[66] Valavanis, K. P., and Vachtsevanos, G. J., *Handbook of Unmanned Aerial Vehicles*, Springer, 2015.

[67] Gulia, R., "Path Loss Model for 2.4GHZ Indoor Wireless Networks with-Path Loss Model for 2.4GHZ Indoor Wireless Networks with Application to DronesApplication to Drones," *RIT Digital Institutional RepositoryRIT*, 2020.

[68] HDMI Licensing Administrator, I., "Raspberry Pi 5," *Raspberry Pi Ltd*, 2024.

[69] Pulse, "1608 2.4G Chip Antenna," *PART NUMBER: ANT1608LL14R2400A*, 2021.

[70] Nighthawk, "Nighthawk® AX12 12-Stream AX6000 WiFi Router Datasheet," *Data Sheet | RAX120*, 2021.

[71] et al., H. L., "Drone Presence Detection by the Drone's RF Communication," , 2021.

[72] Vale, G. L. S., "Raspberry Drone: Unmanned Aerial Vehicle," , 2015.

[73] Adhikary, K. B., Pang, S., and Staiger, M. P., "Dimensional stability and mechanical behaviour of wood–plastic composites based on recycled and virgin high-density polyethylene (HDPE)," *Composites Part B: Engineering*, 2007.

[74] Bhaskar, K., Jayabalakrishnan, D., Kumar, M. V., Sendilvelan, S., and Prabhahar, M., "Analysis on mechanical properties of wood plastic composite," *aterials Today: Proceedings*, 2021.

[75] Muhammad Hilmi Senan, M. M.-N. M. . S. I. A. K., M.A Shaharuzaman, "Recycled Wood Dust Polypropylene Composite (r-WoPPC) Material Selection for Drone Frame Structures by using TOPSIS," *Journal of Advanced Research in Applied Mechanics*, 2023.

[76] Guo, G., and Kethineni, C., "Direct injection molding of hybrid polypropylene/wood-fiber composites reinforced with glass fiber and carbon fiber," *The International Journal of Advanced Manufacturing Technology*, 2020.

[77] Kada, J., Migneault, M., Tabak, G., and Koubaa, A., "Physical and Mechanical Properties of Polypropylene- Wood-Carbon Fiber Hybrid Composites," *BioResources*, 2016.

[78] G., W., and B., S., "Recycling carbon fiber-reinforced polymers by pyrolysis and reused to prepare short-cut fiber C/C composite," Ph.D. thesis, January 2019.

[79] Nilakantan, G., and Nutt, S., "Reuse and upcycling of aerospace prepreg scrap and waste," *Reinforced Plastics*, 2015.

[80] M.-S. Wu, T. C., and Nutt, S. R., "Compression molding of reused in-process waste – effects of material and process factors," *Advanced Manufacturing: Polymer & Composites Science*, Vol. 4, No. 1, 2018, pp. 1–12. doi:10.1080/20550340.2017.1411873.

[81] Guo, G., Chen, J. C., and Gong, G., "Injection molding of polypropylene hybrid composites reinforced with carbon fiber and wood fiber," *ANTEC 2016*, 2017.

[82] David Kazmer, D. M.-A. R. C. J. K., Amy M. Peterson, "Strategic cost and sustainability analyses of injection molding and material extrusion additive manufacturing," , 2023.

[83] de Freitas, S. T., and Peeters, D., "AE1108-II Aerospace Mechanics of Materials," , 2022.

[84] Gero Mohammed, B. D. M.-K. M. T., Arianne Messerman, "Theory and Practice of the Hydrodynamic Redesign of Artificial Hellbender Habitat," *Herpetological Review*, 2016.

[85] Hernan Ceron, M., Caputo, R., and Ariza, P., "Qaudrotor Modeling and a PID Control Approach," , April 2020.

[86] Safarov, T., "Matlab simulation of quadcopter dynamics and PID attitude controller," *Technium*, Vol. 18, 2023, pp. 82–91.

[87] Sulficar, A., Suresh, H., Varma, A., and Radhakrishnan, A., "Modeling, Simulation and complete control of a quadcopter," , 2017.

[88] Bouabdallah, S., and Siegwart, R., "Full Control of a Quadrotor," *Proceedings of the 2007 IEEE/RSJ International*, 2007.

[89] Astrom, K., and Rundqwist, L., "Integrator Windup and How to Avoid it?" 1989.

[90] Astrom, K., and Hagglund, T., *PID Controllers: Theory, Design, and Tuning*, 2nd ed., ISA - The Instrumentation, Systems and Automation Society, 1988.

[91] van der Wal, W., Mooij, E., Papp, Z., and De Teixeira Encarnacao, J., "Simulation, Verification and Valida- tion. Lecture notes," , February 2023.

[92] McGill Office of Sustainability, "What is Sustainability?" , 2024.

[93] Jeroen van den Hoven, I. v. d. P., Pieter E. Vermaas, *Handbook of Ethics, Values, and Technological Design: Sources, Theory, Values and Application Domains*, Springer Reference, 2015.

[94] John Basl, C. W., Ron Sandler, "INTRO TO VALUE SENSITIVE DESIGN," , 2019.