TurbEye

Autonomous Drone for Wind Turbine Inspection

Anton Hendrickx
Enes Berkay Koca
Jip van Sommeren
Louis van Tienhoven
Michael van Breukelen

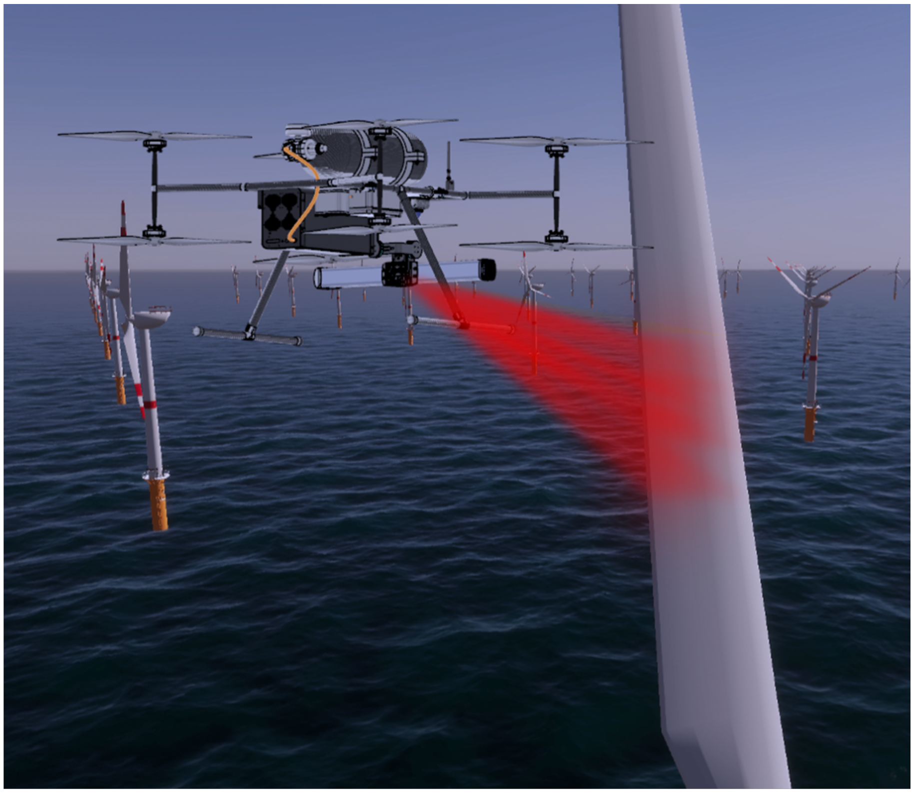
Nachiket Dighe
Tomás Machado
Radu C. Serseniuc
Mike Timmerman
Muhammad Arham Elahi

TU Delft

Delft University of Technology

AE3200

Designing an Autonomous Drone for Wind Turbine Maintenance Inspection: Final Report



Team Members	
Anton Hendrickx (A.H.)	4803671
Enes Berkay Koca (E.K.)	5256690
Jip van Sommeren (J.S)	4545559
Louis van Tienhoven (L.T.)	5022118
Michael van Breukelen (M.B.)	5097460
Mike Timmerman (M.T.)	5216796
Muhammad Arham Elahi (M.E.)	5213320
Nachiket Dighe (N.D.)	5289793
Radu C. Serşeniuc (R.S.)	5245524
Tomás Machado (T.M.)	5301777

June 27, 2023

Executive Overview

Author: Louis

This chapter offers a brief account of the final design of the TurbEye project, focusing on the creation of an autonomous drone for the inspection of wind turbine farms. After thorough research, the team has decided to focus on the offshore wind energy market, as the onshore market is already very saturated and leaves little to no room for improvements from an engineering perspective. However, the offshore market can benefit significantly from technical improvements, such as improved flight time of drones, in combination with take-off from the wind farm substations if necessary. By placing the focus here the team can make a lasting impact on the sector and create an innovative product which is competitive in its field.

The main goal of this executive summary chapter is to outline the core points covered in the report in a clear and comprehensible manner, while also emphasizing crucial findings and recommendations that will be further discussed in later sections.

Project Objective

The mission need statement for the TurbEye project has been phrased as follows:

Develop an autonomous drone for inspection of offshore wind turbines to replace manual methods that are currently in use.

Moreover, the project objective statement is defined as:

Design a drone capable of autonomously performing maintenance checks on off-shore wind turbines through various Non-Destructive Testing (NDT) methods by 10 students in 10 weeks' time.

Market Analysis

An essential part of the project is to conduct an extensive market analysis to assess the current state of technology and identify opportunities within the wind turbine inspection industry. Based on this analysis, projections can be made regarding the demand for the product and its potential for growth. Furthermore, a comprehensive understanding of the market is crucial to determine and fulfil stakeholder requirements.

It was concluded that the wind turbine market is a continuously growing sector with an increasing focus on sustainable and renewable energy. Wind energy is the dominant source of non-hydro renewable energy, accounting for nearly as much power generation as all other sources combined. Another significant finding is the projected growth of offshore wind energy, which is expected to have a substantial share of the overall wind energy capacity in the near future. In 2021 22% of the total capacity increase was due to offshore whereas of all wind turbines worldwide only 7% is currently offshore. Additionally, there is a trend of larger wind turbine sizes, which highlights the need for automation in inspection processes. As turbines and blades increase in size, manual inspections become more challenging and time-consuming for human inspectors to thoroughly examine for defects or damage.

Moreover, based on an extensive customer and competition analysis, the team identified a market gap for a fully autonomous drone propelled by hydrogen, specifically designed for performing complete autonomous offshore wind turbine inspections. The main strengths of this solution are its increased range, which allows for the inspection of entire wind farms, and its full autonomy, which eliminates the need for human intervention. The current practice of human interference, involving the transportation of personnel to the wind farm, is a major cost driver for existing drone operators. TurbEye believes that by offering more cost-effective and efficient maintenance inspections, it can significantly reduce wind turbine downtime.

Based on the outcome of the market analysis, stakeholders could be identified. Afterwards, stakeholder and system requirements could be set up. These requirements were the main driver during the design phase and ultimately led to the final design of the drone as presented in this report.

System Overview

As mentioned in the project objective, the team has developed a detailed design of the drone capable of performing automated wind turbine inspections in 10 weeks' time. The various subsystems that can be distinguished are structures, control, propulsion, and the inspection system. In Figure 0.1 and Table 0.2, an exploded view of the drone is provided, highlighting the different subsystems.

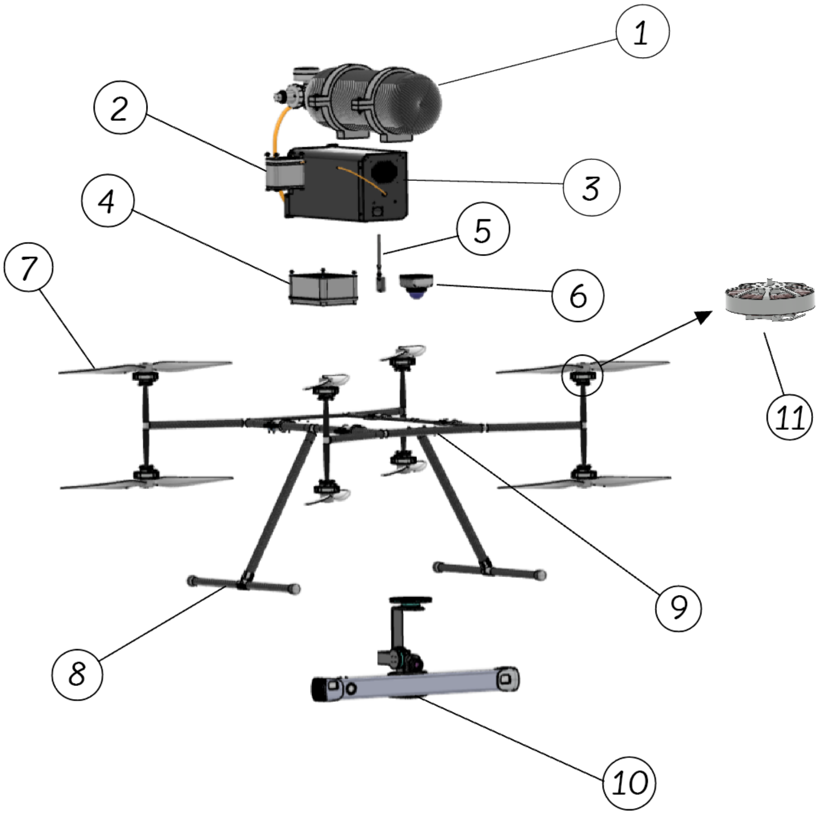


Figure 0.1: Overview Complete Drone

Table 0.2: Component Description

Number	System	Description
1	Propulsion	Hydrogen tank
2	Propulsion	Back-up Battery
3	Propulsion	Fuel Cell
4	Control	Control Box
5	Control	Transmitter/Receiver
6	Control	Lidar
7	Propulsion	Propeller
8	Structure	Landing Gear
9	Structure	Chassis
10	Inspection	Cameras
11	Propulsion	Motor

Top-level specifications

This section shows the most important top-level specifications of the drone, which are highlighted in Table 0.3. The team expects to be able to deliver a total range of 280 [km] on one full hydrogen tank and up to 3.5 hours of flying time while the drone's total weight is under 12 [kg]. With these specifications, the team aims to perform inspections on up to 6 wind turbines per trip when taking off from the wind turbine substation. Further details and elaboration on the specifications of each subsystem will be provided in the subsequent sections of the overview.

Table 0.3: Top-Level Specifications

Characteristic	Value
Range	280 km
Endurance	3.5 h
Mass	11.8 kg
No. of inspection per trip	5 inspections [1]
Hydrogen Mass	0.12 kg
Fuel Cell Capacity	2200 W
Thrust per engine	6.5 kg

Propulsion system

The propulsion system of the drone comprises six main components: the propeller, motor, Electronic Speed Controller (ESC), fuel cell, hydrogen tank and backup battery. The propeller generates thrust, the motor drives the propeller, and the ESC controls the power and voltage supplied to the motor. In terms of power delivery, the system includes a fuel cell and a backup battery for power generation. To store the power source, a hydrogen tank and pressure regulator are utilized. Given the wide range of options available in the market, commercially available off-the-shelf (COTS) components were selected for the propulsion system. Another important argument for using COTS components is the number of safety regulations that apply when working with hydrogen.

Structures

The structures subsystem is responsible for providing physical support to all other subsystems, ensuring optimal operation, robustness, and integration. After trading off various configurations, a coaxial octocopter was determined to be the most suitable design, serving as a foundation for the overall chassis design. The chassis is primarily constructed using Carbon Fibre Epoxy Resin with Graphene Nano-Platelets, which offers excellent strength and lightweight properties. Connections between the chassis tubes are made using 3D-printed PETG connections. Additionally, a comprehensive Finite Element Analysis was conducted to validate the stiffness and rigidity of the structure and to prevent any resonance issues with the propellers.

---

[1] Average number of inspections per trip tested on Hornsea 2

Guidance, Navigation and Control

Guidance, Navigation, and Control (GNC) can be considered the brains behind every UAV including TurbEye. It provides the demanded intelligence and control capabilities that the drone shall require to perform its operations effectively. Additionally, the autonomous nature of TurbEye makes such a system even more crucial, enabling it to autonomously navigate through environments, adapt to dynamic situations, and execute tasks with precision. Furthermore, guidance is responsible for planning the drone's path, helping it avoid obstacles, and finding the best route to follow to get from point A to B. Navigation details how the selected sensors (GPS, IMU, and Barometer) determine the drone's position, velocity, attitude, and angular velocity. Finally, Control allows for the management of the drone's actuators in order to provide the necessary forces and moments to control the drone.

Inspection system

Regarding the inspection system, after an extensive trade-off, it has been concluded that a combination of the (NDT) inspection methods as described below provides an as complete as possible inspection of the wind turbine.

- **Visual**
  A regular high-quality camera is used to identify any surface damage visible from the outside such as cracks, paint peeling or scrape
- **Passive Thermography**
  Through passive thermography, a sub-surface level inspection should be provided, visualizing damage such as skin/adhesive debonding or delamination.
- **Geometry Inspection**
  In order to detect erosion, which is not visible with a regular stereo camera but heavily reduces aerodynamic performance, a depth sensor is added.

Operations and Logistics

The operational procedures consist of five stages that the drone follows during a typical mission. Prior to takeoff, a pre-flight checklist is performed to ensure all systems are functioning properly. This includes sensor calibration, checking the battery and fuel tank, and verifying communication, among other tasks. During the mission, the drone deploys from the station and ascends to a safe flying altitude according to certification standards. It then approaches the first wind turbine for inspection, which typically takes around 30 minutes. After completing the inspection, the drone follows its pre-programmed route to the next wind turbine. If no further inspections are required or if the drone needs to be refuelled, it returns to the substation. After the landing, the data collected during the flight is retrieved from the SD card for post-processing, and the hydrogen tank is refuelled or replaced. At the end of the operation, a decision is made regarding any noticeable damage that may have occurred. If no damage is reported, the cycle begins again with the pre-flight checklist.

Wind Farm Traversal Problem

A major pre-flight procedure involves planning the trips required for the drone to efficiently cover all the wind turbines in a wind farm. It is necessary to follow a strategy that allows the drone to visit all turbines while ensuring each trip has sufficient fuel. An optimized traversal path reduces the number of trips and minimizes downtime and costs. Therefore, the project implemented both a VRP (Vehicle Routing Problem) and an open-source tool developed by Google. The VRP algorithms were tested on the Hornsea 2 wind farm, the world's largest, and the results are presented in Table 0.4. The outcome is that the entire wind farm can be inspected with only 35 trips and less than 3.7 [kg] of hydrogen.

Table 0.4: Hornsea 2 wind farm inspection

Parameter	**Original**	**Updated**	**OR-Tools**
Trips [-]	39	38	35
H_2 used [kg]	3.802	3.755	3.647
Total flight time [hrs]	101.45	100.36	98.08

AI Damage Detection

Moreover, the team implemented two AI algorithms for wind turbine damage detection. The first algorithm, developed by TurbEye engineers, focused on a binary classification task to detect dirt or damage in the images. It exhibited good accuracy, demonstrating its feasibility for post-processing operations. The second algorithm aimed to differentiate between dirt and damage while specifying the location of the dirt. However, it was solely implemented and did not perform well due to low accuracy, recall, and precision. This can be attributed to the limited training data available, as the dataset used consisted of only 3,000 images compared to the typical 50,000-image datasets used for similar models.

In order to train these algorithms, the team utilized a publicly available database of drone images specifically captured for wind turbine inspection purposes. This open source dataset also consisted of YOLO labels, a special image labelling for class detection.

Overall, these AI algorithms contribute to the understanding of wind turbine conditions but highlight the importance of adequately sized datasets for training reliable models in complex tasks such as dirt detection and localization.

Financial Analysis

Cost of the drone

The production cost of a single drone is estimated to be nearly 60,000 euros. However, the cost is primarily driven by three components. The fuel cell is estimated to cost around 25,000 euros, the 3D scanner costs 13,000 euros, and the visual and thermal camera costs 12,000 euros. Together, these components account for more than 80% of the total drone cost.

Financial Projections

Based on the final design of the drone and the market analysis, preliminary financial projections were made. These projections include the market volume for the product, achievable market share, production costs, operational costs, and the costs associated with necessary loans to secure sufficient financial capital. Insurance costs were also considered to cover the risk in case of accidents. Taking all these factors into account, a total return on investment of 30.8% is expected. The total revenue is projected to be 9.77 million euros, while the total costs amount to 6.76 million euros, as shown in the financial summary in Table 0.5.

Table 0.5: Financial Summary

Year	**Revenue (€)**	**Costs (€)**	**Profit (€)**	**RoI (%)**
Year 1	1,600,000	1,324,741.31	275,258.69	17.20
Year 2	1,760,000	1,301,025.05	458,974.95	26.08
Year 3	1,936,000	1,338,129.11	597,870.89	30.88
Year 4	2,129,600	1,376,755.03	752,844.97	35.35
Year 5	2,342,560	1,416,271.99	926,288.01	39.54
Total	9,768,160	6,756,922.49	3,011,237.51	30.83

Sustainable Development Strategy

For the design of a wind turbine inspection drone, a holistic approach was followed, considering environmental, societal, and economic impacts. The goal is to minimize the negative impact on the environment while maximizing benefits to society and the economy. To achieve this, design goals and choices were derived from the Engineering for Sustainable Development framework developed by the United Nations. These choices include the utilization of an advanced VRP algorithm, the use of low-carbon transportation options for drone transport, and the incorporation of recycled materials in component production. Furthermore, the aim is to utilize "green" hydrogen, which is the cleanest and least emissive form of hydrogen available.

Future Steps

Due to the limited time available, further improvements are still possible in the design of the drone. On the control side, these improvements include developing a non-linear control model, enhancing waypoint tracking, and incorporating emergency control procedures. Additionally, the reliability and accuracy of the AI model need to be improved, by training on a bigger dataset. In terms of CAD, more details can be added, such as mounts, wires, and connections, to enhance the overall design. Furthermore, the FEM model can be further refined to optimize the chassis. Concerning the propulsion system, simulating dynamic conditions, such as varying weight during flight and changing environmental conditions, would be a key improvement.

Having the complete drone designed would pave the way for the production phase, during which all individual components can be purchased or manufactured, followed by assembly. The next step is to conduct an extensive testing phase to validate the design and make any necessary design changes. This testing phase involves testing all subsystems and the complete system to ensure their correctness and functionality. Once the testing phase is completed, TurbEye can begin attracting customers and performing their first wind turbine inspections.

Contents

List of Symbols

Abbreviations

ABS	Acrylonitrile Butadiene Styrene
AC	Alternating Current
AI	Artificial Intelligence
ARS	Automatic Recovery Sequence
CAD	Computer Aided Design
CCF	Cut Carbon Fiber
CEO	Chief Executive Officer
CF	Carbon Fiber
CFRP	Carbon Fibre Reinforced Polymer
CNC	Computer Numerical Control
CNN	Convolutional Neural Network
COTS	Commercial Off-The-Shelf
DC	Direct Current
DCM	Direction Cosine Matrix
DOT	Design Option Tree
DSE	Design Synthesis Exercise
ECEF	Earth-Centered Earth-Fixed
EKF	Extended Kalman Filter
ER	Epoxy Resin
ESC	Electronic Speed Controller
ESD	Engineering for Sustainable Development
FC	Flight Computer
FCHS	Fuel Cell Hybrid System
FEA	Finite Element Analysis
FP/FN	False Positive/False Negative
FPS	Frames Per Second
FTS	Flight Termination System
GFRP	Glass Fiber Reinforced Polymer
GLS	Guided Local Search
GNC	Guidance Navigation Control
GNC	Guidance, Navigation, and Control
GNP	Graphene Nano-Platelets
GNSS	Global Navigation Satellite System
GPS	Global Positioning System
IMU	Inertial Measurement Unit
IOP	Integral O&M Provider
IPCC	Intergovernmental Panel on Climate Change
ISA	International Standard Atmosphere
JSON	JavaScript Object Notation
Li-Ion	Lithium-Ion
Li-Po	Lithium-Polymer
LiDAR	Light Detection And Ranging
LLA	Latitude, Longitude and Altitude
LQR	Linear Quadratic Regulator
MHI	Mitsubishi Heavy Industries
MIL	Model-In-the-Loop
MTVRP	Multi-Trip Vehicle Routing Problem
NDT	Non Destructive Testing
NED	North, East, Downs
NX NASTRAN	NeXt (Unigraphics) NASA STRucture ANalysis
NZE	Net Zero Emissions
O&M	Operations & Maintenance
OEM	Original Equipment Manufacturer
OOM	Order Of Magnitude
OSS	Offshore Sub Station
PC	Polycarbonate
PETG	PolyEthylene Terephthalate Glycol
PID	Proportional Integral Derivative
PIL	Propeller-In-the-Loop
PLA	Polylactic Acid
PoE	Power over Ethernet

RAMS	Reliability, Availability, Maintainability and Safety
RMSE	Root Mean Square Error
ROI	Return on Investment
RPM	Rotations Per Minute
RRT	Rapidly-exploring Random Tree
RTK	Real-Time Kinematics
SCF	Short Carbon Fiber
SD	Secure Digital
SDS	Sustainable Development Strategy
SIL	Software-In-the-Loop
SLA	Service-Level Agreement
SLAM	Simultaneous Localization and Mapping
SWOT	Strengthes, Weaknesses, Opportunities and Threats
TBC	To Be Confirmed
TBD	To Be Determined
TBI	To Be Implemented
TBT	To Be Tested
TNO	Nederlandse Organisatie voor Toegepast Natuurwetenschappelijk Onderzoek (Netherlands Organization for Applied Scientific Research)
TP/TN	True Positive/True Negative
TR	Technical Risk
TRL	Technology Readiness Level
TSP	Travelling Salesman Problem
TUD	Delft University of Technology
UAS	Unmanned Aerial System
UAV	Unmanned Aerial Vehicle
UN	United Nations
USD	United States Dollars
V&V	Verification and Validation
VRP	Vehicle Routing Problem
YOLO	You Only Look Once

Greek Symbols

η	Attitude	$[rad]$
μ	Mean	$[-]$
σ	Standard Deviation	$[-]$
ω	Angular velocity	$[rad/s]$
ϕ	Roll Angle	$[rad]$
ψ	Yaw Angle	$[rad]$
ρ	Density	$[kg/m^3]$
σ_y	Yield strength	$[N/m^2]$
τ_m	Time constant of motor response	$[1/s]$
θ	Pitch Angle	$[rad]$

Latin Symbols

$\mathbf{u}(\mathbf{t})$	Control Vector at time t	$[-]$
$\mathbf{x}(\mathbf{t})$	State Vector at time t	$[-]$
$\mathbf{y}$	Sensor Measurement	$[-]$
A	State Space Matrix	$[-]$
B	Input Matrix	$[-]$
C	Output Matrix	$[-]$
D	Feed-through Matrix	$[-]$
$H(s)$	Transfer Function	$[-]$
I	Moment of Inertia	$[-]$
K	Gain Matrix	$[-]$
Q	Weight Matrix of State	$[-]$
R	Weight Matrix of Input	$[-]$
T	Time-step Matrix	$[-]$
v	Measurement noise	$[-]$
w	Process noise	$[-]$
A	Measured IMU Acceleration	$[m/s^2]$
B_p	Number of Blades	$[-]$
C_D	Drag Coefficient	$[-]$
C_M	Torque Coefficient	$[-]$
C_T	Thrust Coefficient	$[-]$
D_p	Blade Diameter	$[m]$
d_p	Vertical distance between Propellers	$[m]$

E	Modulus of Elasticity	$[MPa]$
G	Shear Modulus	$[MPa]$
g	Acceleration due to gravity	$[m/s^2]$
h	Altitude	$[m]$
H_p	Propeller Pitch	$[m]$
I_m	Motor Current	$[A]$
k_C	Aerodynamic Efficiency of Lower Propeller	$[-]$
k_F	Propeller Thrust Coefficient	$[m^4/rad^2]$
k_M	Propeller Moment Coefficient	$[m^5/rad^2]$
K_V	Motor velocity constant	$[1/V]$
M	Moment	$[Nm]$
m	Mass	$[N]$
N	Normal Force	$[N]$
N_m	Number of Motors	$[-]$
p	Roll Rate	$[rad/s]$
q	Pitch Rate	$[rad/s]$
r	Yaw Rate	$[rad/s]$
R_e	Motor Internal Resistance	$[\Omega]$
R_m	Motor Resistance	$[\Omega]$
S	Surface Area	$[m^2]$
T	Thrust	$[N]$
t	t	$[s]$
T/W	Thrust-Weight Ratio	$[-]$
U_m	Motor Voltage	$[V]$
v	velocity	$[m/s]$
W	Weight	$[N]$
x	x-position	$[m]$
y	y-position	$[m]$
z	z-position	$[m]$
R	Radius of propellant tank	$[m]$

1: Market Analysis

Author: Anton

A thorough market analysis is vital before the design and development of a product. This chapter aims to analyze the current state of the supply & demand in the market, competitors, market dynamics, and potential growth and provide takeaways from companies as well as experts in the field. The chapter starts off with analyzing the current market in section 1.1, predicted trends within the market in section 1.2, and the competitors' similar products in section 1.3. Potential customers for TurbEye are explained in section 1.4, and finally, a technical SWOT analysis is performed in section 1.6.

1.1 Current Market

Author: Anton

The wind energy market is on a continuous growth trajectory with more resources allocated towards sustainable and renewable energy. The Global Wind Energy Market Size was valued at 79.7 Billion USD in 2021 and is expected to reach 151.47 Billion USD by 2030 [1]. Wind Energy is by far the leading source of non-hydro renewable energy, totalling almost as much power generation as all the others combined. In 2021, 1870 TWh of power were generated by wind energy, an increase of 273 TWh (17%) over 2020. Furthermore, wind energy aims to increase its capacity even further to get on track with the net zero emissions goal (NZE) in 2050, targeting 7900 TWh of wind energy generation by 2030 [2].

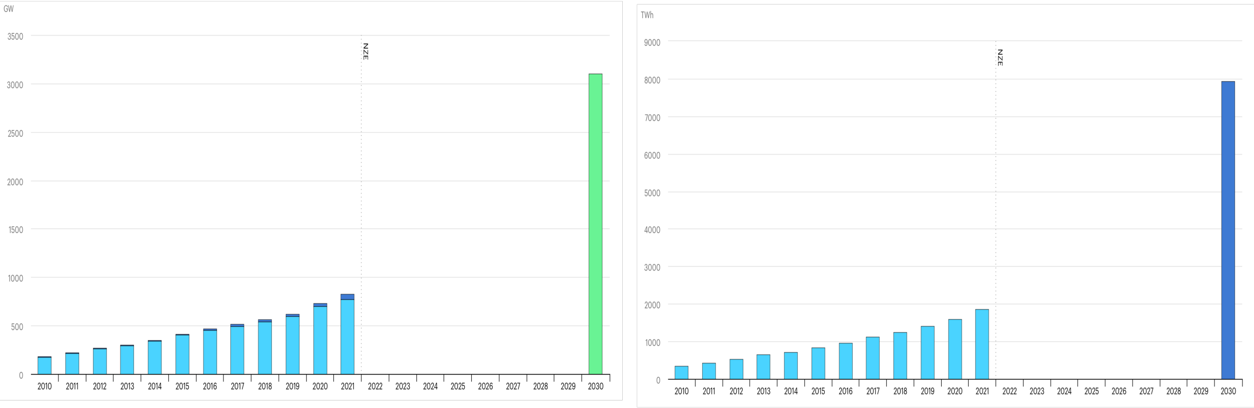


Figure 1.1: Onshore and offshore wind energy capacity and wind power generation from 2010 to 2030 [2]

While onshore wind energy currently accounts for the majority of the produced wind energy, there is a steady growth in the offshore wind energy market. It is projected that offshore wind energy will soon constitute a significant portion of the overall wind energy capacity. 93% of all wind farms are onshore while 7% are offshore, with only 19 countries having active offshore wind farms. About 22% of the total capacity increase in 2021 of 94 GW was due to offshore wind farms [2].

1.2 Future Trends

Author: Anton

As the energy market continues to evolve, the demand for renewable energy increases. Wind energy accounts for 7.54% of global energy production, generated by over 341,000 wind turbines worldwide. As the market and number of wind turbines continue to increase, their maintenance systems are also required to be more thorough, accurate, and efficient. The trends for inspection drones are shifting towards having completely autonomous drones with strong stability characteristics. In addition, contact methods like ultrasound and eddy current testing can be implemented alongside visual inspection and thermography to get more reliable and accurate results.

Extremely precise and accurate take-off and landing systems are also extremely useful for operation in adverse conditions.

1.2.1 Turbine size

Author: Anton

Another interesting trend within the wind turbine industry is the growing size of wind turbines. TNO recently published a report stating the expected blade size is expected to grow to 145 meters to account for the growing need of green energy without having to build more wind turbines and reduce cost per wind turbine [3].
With the increasing size of wind turbines, it becomes even more necessary to automate inspection processes. The larger the turbine and its blades become, the more difficult and time-consuming it becomes for human inspectors to thoroughly examine them for defects and damage.

1.3 Competitors

Author: Anton, Tomás

Competitors can be distinguished into two categories: companies that provide inspections with drones (Figure 1.2) and companies that provide inspections with manual labour (1.3). Although the goal of this project is to design an autonomous drone for wind turbine maintenance inspection, and the companies that perform drone inspections are the main competitors, 'rope-based inspection' companies can also be investigated to some extent in order to look at some of their limitations.

Figure 1.2: Drone Based Inspection

Figure 1.3: Rope Based Inspection

Figure 1.4: Comparison of Inspection Methods

1.3.1 Manual labor inspections

When technicians perform rope-based inspections, they must physically climb the wind turbine using ropes and harnesses. This method is not only time-consuming and labour-intensive but also increases the potential for accidents. However, rope-based inspections still happen in practice due to their hands-on approach, enabling technicians the ability to carefully examine wind turbine elements and, if required, immediately address maintenance/repair needs.

1.3.2 Drone inspection companies

In order to deal with the main limitations of rope-based inspections, a great alternative could be the use of drones. Drones can be equipped with technology so that they become capable of detecting failures. There are already quite a few players in this market, which are listed below in Table 1.1. Among the various competitors, certain companies distinguish themselves in terms of their advancements in development and design. These companies are listed as such.

Table 1.1: Market competitors

SkySpecs	Aerodyne Group
Sterblue	Percepto
Clobotics	Sulzer Schmid
DroneBase	Cyberhawk
PrecisionHawk	Perceptual robotics
Bladeinsight	Voliro

1. **Voliro:** Voliro [4] is a Swiss company known for its innovative drone technology. Their drones offer unique capabilities, such as the ability to hover in any orientation, which can be particularly advantageous for inspecting wind turbines. Their high degree of manoeuvrability and precision allows for close-up inspections while reducing the risk of collisions.

   Key strengths:

   - Innovative drone design with advanced manoeuvrability
   - Customizable solutions for various industries
   - Strong research and development capabilities

2. **SkySpecs:** SkySpecs [5] is a US-based company that provides automated drone inspections for the wind energy industry. Their drones are equipped with advanced computer vision and machine learning technology, enabling them to autonomously inspect wind turbines and generate detailed, actionable reports.

   Key Strengths:

   - Proven track record within the wind energy industry
   - Advanced computer vision and machine learning capabilities
   - Quick inspection times and high-quality data analysis
   - Strong partnerships with industry leaders

3. **Sulzer Schmid:** Sulzer Schmid [6] is a Swiss company that offers advanced drone inspection solutions for wind turbines. They use their branded 3DXTM inspection technology which uses high-resolution cameras to capture images which are processed and analyzed to identify any potential defects.

   Key strengths:

   - Highly accurate 3DXTM inspection technology
   - Heavy focus on data analysis and making reports from it
   - Already operational

4. **BladeInsight:** BladeInsight [7] is a Portuguese company specializing in drone-based inspection solutions for wind turbines. They integrate the utilization of drones with advanced software to automate the inspection process and generate an inspection report. By automating the inspection process the downtime of the wind turbines is reduced.

   Key Strengths:

   - Automation and efficiency improvements
   - Comprehensive inspection and data analysis
   - Wind turbine downtime of only 30 minutes (fast)

1.4 Customers

Author: Anton

Integral operations and maintenance of different wind farms can be provided by different types of companies: the wind farm operator, the original equipment manufacturer (OEM), or a specialized operation and maintenance (O&M) firm. In the Netherlands, Belgium, and Germany, the majority of wind farms are serviced by OEM's and wind farm operators themselves. OEM's are companies such as *Siemens Gamesa* and *Vestas* while operators include *Ørsted*, *Vattenfall*, and *Eneco*. The organization providing the integral O&M is further referenced as Integral O&M Provider (IOP). The IOP is responsible for the whole chain of operations: inspection, maintenance, repair, and replacements. When the wind farm operator does not do the integral O&M themselves, they mostly hire an independent IOP on a service level agreement (SLA) basis which guarantees a minimum uptime of the turbines. Even though the IOP is responsible for the whole chain of operations, they use subcontractors to provide certain activities. For example, when an off-shore turbine blade has to be replaced, they hire a jack-up vessel with a crew from a company such as *Van Oord* or *Fred. Olsen Windcarrier*. This is also done for inspection; the IOP hires external companies for blade inspection with ground-based telephoto lens inspection, helicopter inspection, or drone inspection. Thus, TurbEye's clients are the IOPs responsible for the uptime of the turbines. They encompass a limited group of billion-dollar revenue companies. A list of the most important companies (by size) operating in the Netherlands and Belgium can be found in Table 1.2 and 1.3 with their respective number of installed turbines and total installed power.

Integral O&M Provider	**No. turbines**	**Installed power**
Eneco Wind Offshore Operations	103	249 MW
Ørsted	94	752 MW
Shell	36	108 MW
Siemens Gamesa (NL)	335	1270.7 MW
Enercon	72	387 MW
Vattenfall	118	423 MW
Ventolines	137	526.6 MW
CrossWind HKN	69	759 MW
Blauwwind	77	731.5 MW

Table 1.2: IOP's active in The Netherlands

Integral O&M Provider	**No. turbines**	**Installed power**
MHI Vestas Offshore Wind	201	771 MW
Siemens Gamesa (BE)	154	1121 MW
Norther NV	44	370 MW

Table 1.3: IOP's active in Belgium

1.5 Interviews

Authors: Anton, Tomás

While conducting the market analysis, the team proactively reached out to many companies and experts (if deemed valuable) to aid the team in understanding the industry better by utilising the knowledge and expertise they possess. If the efforts to contact the companies/experts were successful, a meeting or a call was arranged. The companies/experts that were successfully contacted are listed below as well as a short description of the interview:

1. **Sulzer & Schmid:** As previously mentioned, Sulzer & Schmid is a Swiss-based inspection company which specializes in advanced drone inspections for wind turbines. The team had the opportunity to meet with one of its founders, Christof Schmid.
   Key takeaways
   - An inspection of 1 turbine is typically carried out in approximately 30 minutes.
   - Fully automating the process is possible but is usually slower than having a pilot.

   - Most inspections performed in a day is 20 turbines.
   - Off-shore inspections are drastically more expensive due to the requirement of having a boat present at all times to carry out the inspections.
   - Passive thermography is feasible in Nordic countries due to the de-icing system the turbines are equipped with.

2. **Smulders group:** Smulders group [8] is a Belgian-based company specialising in the construction of the foundation of offshore wind turbines. Although they do not perform maintenance/inspections of wind turbines, it was still deemed useful to have a brief phone call with them due to the nature of the relationship that one of the team members had with an employee of the Smulders group.
   Key takeaways

   - The connection will forward the team to a wind turbine operator.
   - The market trend (growing) was confirmed
   - A possible field trip was discussed

3. **Dr. D. (Donatella) Zappalá:** A meeting was held at the faculty with Dr. D. Zappalá, who specializes in wind energy and thus was a very useful contact for the team.
   Key takeaways

   - There is a growing demand for automatised systems to inspect wind turbines and quite a bit of research is done into this field
   - Current methods have difficulties capturing all types of failure, i.e. both outside and inside failure
   - Inspections are currently done 2 or 3 times a year per wind turbine. Rope-based inspection is still a very commonly used approach
   - Current drone systems are not 'state of the art' due to lack of automation and lack of ability to capture all failure modes
   - There is still a lot of potential in the offshore wind farm market

4. **Ir. João Sandro:** Ir. Sandro is a TU Delft alumni who is specialized in the testing of wind turbines, having even showcased the difficulty of his job in a Youtube video with nearly 2 million views [9].

   Key takeaways

   - Since wind turbines are mainly made out of carbon fibre, corrosion of the turbine's surface is not an issue.
   - Mr. Sandro will arrange internal communication with Blade Insight, as he was a source of inspiration for the project.
   - Not using an autonomous system nowadays is illogical for two reasons: the technology is already advanced enough to continue with manual inspection, and it significantly reduces manpower in a severely dangerous and repetitive task.
   - An autonomous system can guarantee that the distance between the drone and the wind turbine is constant throughout the entire testing, as well as it aids the camera maintain its image focus.
   - One aspect which may influence the camera selection is the fact of the turbines being completely white, making it more difficult for cameras to focus on/ find defects.
   - The reason why one cannot seem to find drones for wind turbine testing in offshore situations is that there seems not to exist a big difference between onshore and offshore drone inspection. Thus, companies which are able to provide the service on land are also able to provide on water.
   - In a best-case scenario, a minimum distance of 800m is required between wind turbines.
   - A great idea would be for the software to be universal, i.e. capable of being applied to any turbine model.
   - Post-processing of the data is done after landing through a cloud system.
   - BladeInsight and similar companies store the collected data in an SD card.
   - Thermal inspection of wind turbines is apparently highly unfeasible. As a matter of fact, it was a very popular concept five years ago, but never proved its viability and has been discarded since.

5. **H3 Dynamics:** H3 Dynamics is a technology company that specializes in hydrogen-propelled aircraft, making long-endurance flights possible. [10]

   Key takeaways

   - A hydrogen fuel cell has great potential for drone operations.
   - A flight time of 3 hours is feasible.
   - Although the rules are strict about hydrogen-propelled vehicles, no issues are foreseen.
   - An external battery is required to provide an emergency power supply and help with takeoff and landing.
   - Retrofitting of off-the-shelf drones with a hydrogen fuel cell is possible.

6. **BladeInsight:** Once again, the team had the chance to have a meeting with BladeInsight's founder, André Croft de Moura.

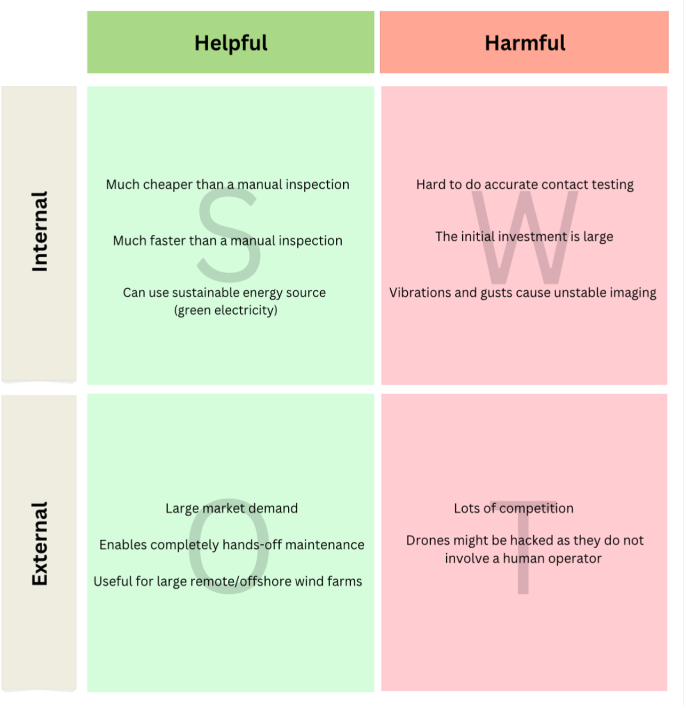
   key takeaways

   - All data obtained during the inspection (visual and thermal footage) is stored in an SD card.
   - Post-processing of the drone's data is done inland.
   - BladeInsight does not build its own drone, it merely implements its payload and respective software into a *DJI Matrice 300* drone.
   - RTK sensors are not used in offshore inspections, since there is no proper location to place the fixed base station.

1.6 SWOT Analysis

Author: Anton

Performing a SWOT (strength, weakness, opportunity and threat) analysis can provide valuable insights and help in understanding the market dynamics of a given market. Additionally identifying opportunities, and developing effective strategies. The SWOT analysis performed can be found below.



	Helpful	Harmful
Internal	Much cheaper than a manual inspection Much faster than a manual inspection Can use sustainable energy source (green electricity)	Hard to do accurate contact testing The initial investment is large Vibrations and gusts cause unstable imaging
External	Large market demand Enables completely hands-off maintenance Useful for large remote/offshore wind farms	Lots of competition Drones might be hacked as they do not involve a human operator

Figure 1.5: Technical SWOT Diagram

1.7 Conclusion

Author: Anton

After doing an extensive market analysis and having conducted interviews with multiple companies and experts in the field, the team has decided to focus on the offshore wind energy market. This is because the onshore market is already very saturated and leaves little to no room for improvements from an engineering perspective. However, the offshore market can benefit significantly from technical improvements, such as improving the flight time of drones. By placing the focus here, the team can make a lasting impact on the sector and create an innovative product which is competitive in its field.

2: Functional Analysis

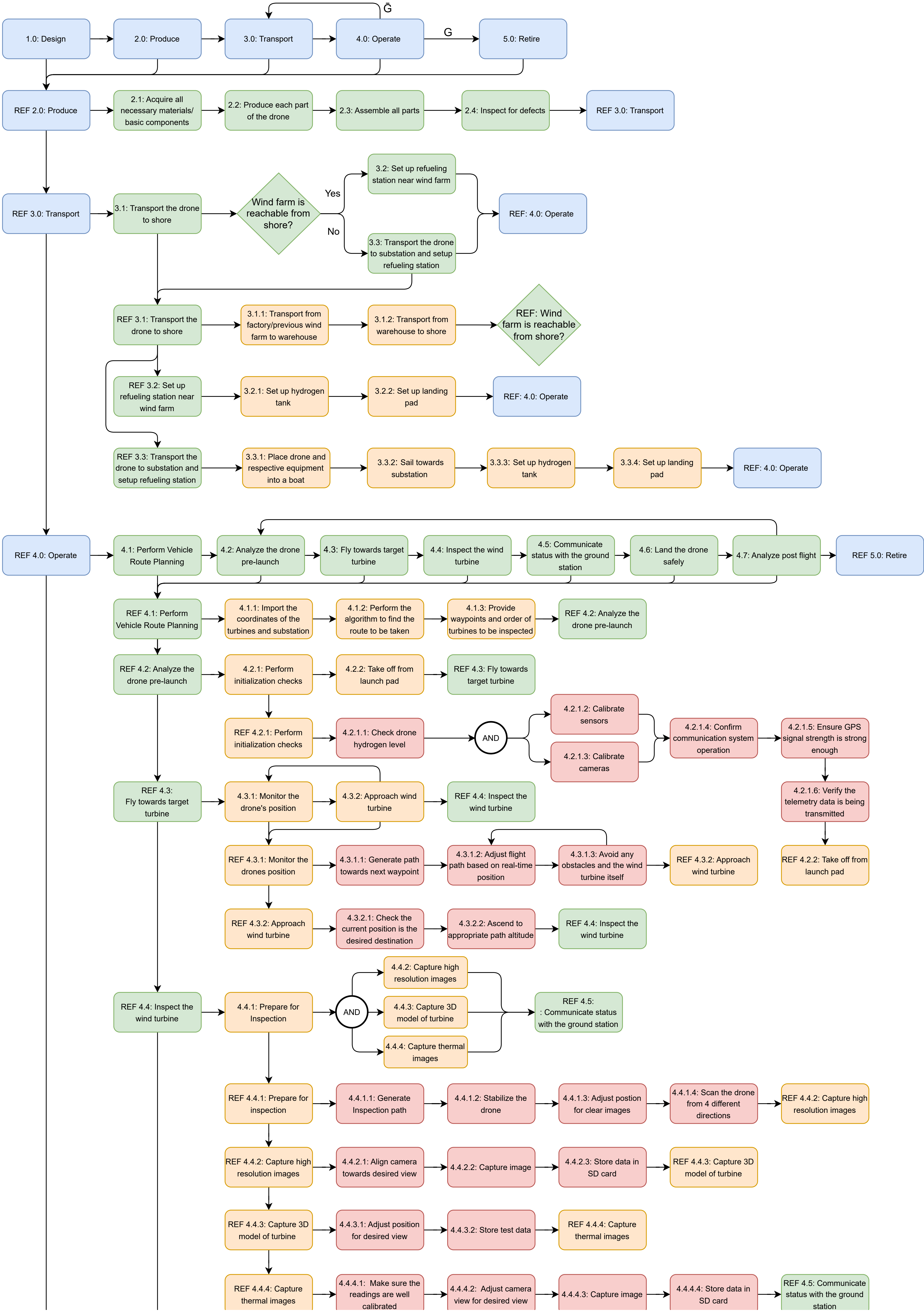
Author: Arham, Tomás

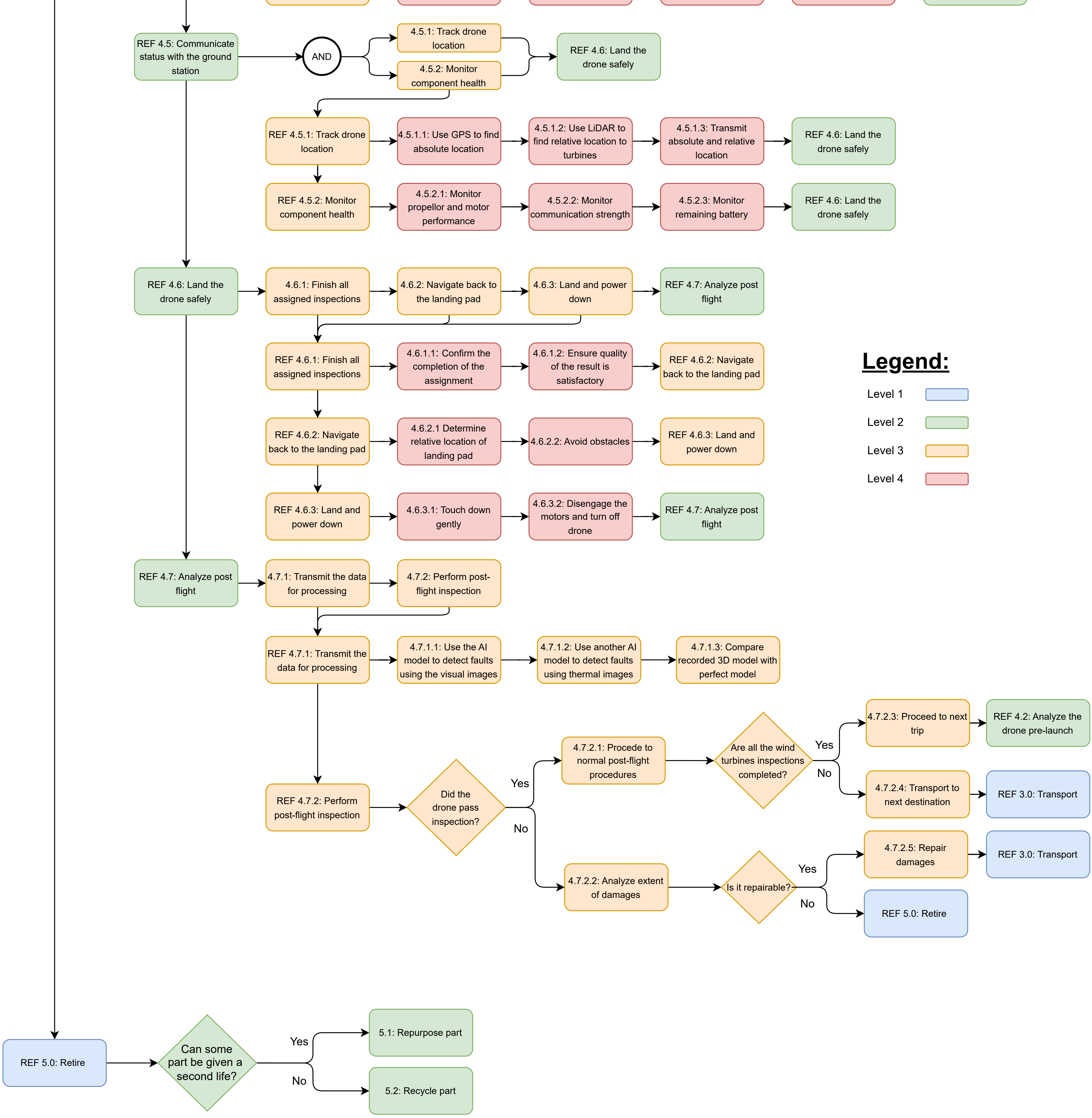
2.1 Functional Flow Diagram

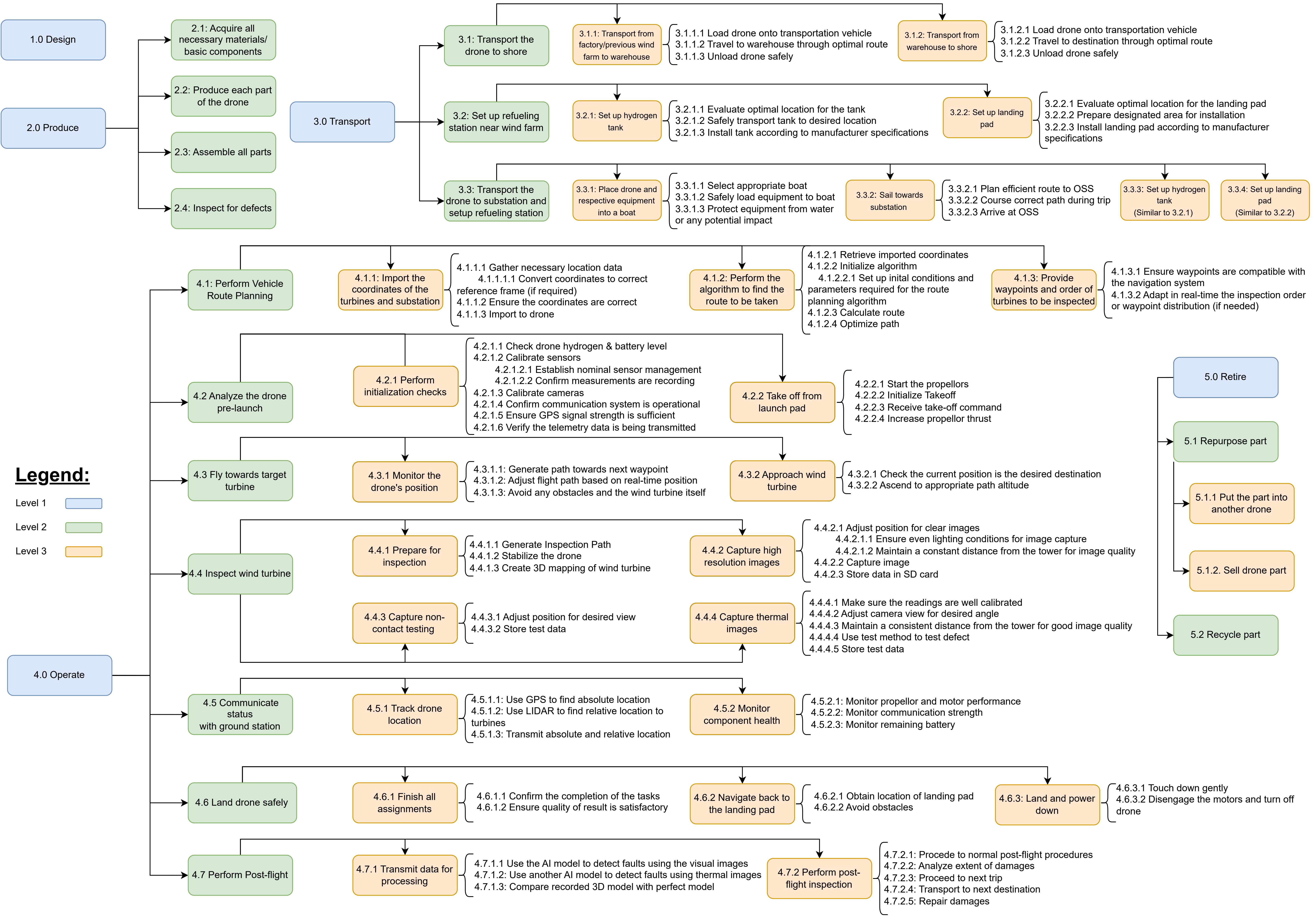
The functional flow diagram provides a visual representation of the sequential steps and interactions involved in the operation of the drone. It outlines the logical progression of tasks and interdependencies between components, enabling the optimization of operational efficiency and workflow. By analyzing the functional flow diagram, insights can be gained into the drone's operation, identifying areas to design and improve, and enhancing the effectiveness of wind turbine maintenance inspections. The diagram is given in the next page. Regarding the "Retire" block, a further explanation shall be done in the post-DSE Gantt chart (section 10.2).

2.2 Functional Breakdown Structure

The functional breakdown structure (FBS) is a hierarchical diagram that organizes the functions and sub-functions of an autonomous drone system for wind turbine maintenance inspections. It provides a structured representation of tasks and their relationships, enabling efficient coordination and optimization of system capabilities. Analyzing the FBS allows for clear task definition, resource allocation, and meeting functional requirements for successful inspections. It is given two pages later.


1.0: Design
2.0: Produce
3.0: Transport
4.0: Operate
5.0: Retire
Ḡ
G
REF 2.0: Produce
2.1: Acquire all necessary materials/ basic components
2.2: Produce each part of the drone
2.3: Assemble all parts
2.4: Inspect for defects
REF 3.0: Transport
REF 3.0: Transport
3.1: Transport the drone to shore
Wind farm is reachable from shore?
Yes
No
3.2: Set up refueling station near wind farm
3.3: Transport the drone to substation and setup refueling station
REF: 4.0: Operate
REF 3.1: Transport the drone to shore
3.1.1: Transport from factory/previous wind farm to warehouse
3.1.2: Transport from warehouse to shore
REF: Wind farm is reachable from shore?
REF 3.2: Set up refueling station near wind farm
3.2.1: Set up hydrogen tank
3.2.2: Set up landing pad
REF: 4.0: Operate
REF 3.3: Transport the drone to substation and setup refueling station
3.3.1: Place drone and respective equipment into a boat
3.3.2: Sail towards substation
3.3.3: Set up hydrogen tank
3.3.4: Set up landing pad
REF: 4.0: Operate
REF 4.0: Operate
4.1: Perform Vehicle Route Planning
4.2: Analyze the drone pre-launch
4.3: Fly towards target turbine
4.4: Inspect the wind turbine
4.5: Communicate status with the ground station
4.6: Land the drone safely
4.7: Analyze post flight
REF 5.0: Retire
REF 4.1: Perform Vehicle Route Planning
4.1.1: Import the coordinates of the turbines and substation
4.1.2: Perform the algorithm to find the route to be taken
4.1.3: Provide waypoints and order of turbines to be inspected
REF 4.2: Analyze the drone pre-launch
REF 4.2: Analyze the drone pre-launch
4.2.1: Perform initialization checks
4.2.2: Take off from launch pad
REF 4.3: Fly towards target turbine
REF 4.2.1: Perform initialization checks
4.2.1.1: Check drone hydrogen level
AND
4.2.1.2: Calibrate sensors
4.2.1.3: Calibrate cameras
4.2.1.4: Confirm communication system operation
4.2.1.5: Ensure GPS signal strength is strong enough
4.2.1.6: Verify the telemetry data is being transmitted
REF 4.2.2: Take off from launch pad
REF 4.3: Fly towards target turbine
4.3.1: Monitor the drone's position
4.3.2: Approach wind turbine
REF 4.4: Inspect the wind turbine
REF 4.3.1: Monitor the drones position
4.3.1.1: Generate path towards next waypoint
4.3.1.2: Adjust flight path based on real-time position
4.3.1.3: Avoid any obstacles and the wind turbine itself
REF 4.3.2: Approach wind turbine
REF 4.3.2: Approach wind turbine
4.3.2.1: Check the current position is the desired destination
4.3.2.2: Ascend to appropriate path altitude
REF 4.4: Inspect the wind turbine
REF 4.4: Inspect the wind turbine
4.4.1: Prepare for Inspection
AND
4.4.2: Capture high resolution images
4.4.3: Capture 3D model of turbine
4.4.4: Capture thermal images
REF 4.5: : Communicate status with the ground station
REF 4.4.1: Prepare for inspection
4.4.1.1: Generate Inspection path
4.4.1.2: Stabilize the drone
4.4.1.3: Adjust postion for clear images
4.4.1.4: Scan the drone from 4 different directions
REF 4.4.2: Capture high resolution images
REF 4.4.2: Capture high resolution images
4.4.2.1: Align camera towards desired view
4.4.2.2: Capture image
4.4.2.3: Store data in SD card
REF 4.4.3: Capture 3D model of turbine
REF 4.4.3: Capture 3D model of turbine
4.4.3.1: Adjust position for desired view
4.4.3.2: Store test data
REF 4.4.4: Capture thermal images
REF 4.4.4: Capture thermal images
4.4.4.1: Make sure the readings are well calibrated
4.4.4.2: Adjust camera view for desired view
4.4.4.3: Capture image
4.4.4.4: Store data in SD card
REF 4.5: Communicate status with the ground station


REF 4.5: Communicate status with the ground station
AND
4.5.1: Track drone location
4.5.2: Monitor component health
REF 4.6: Land the drone safely
REF 4.5.1: Track drone location
4.5.1.1: Use GPS to find absolute location
4.5.1.2: Use LiDAR to find relative location to turbines
4.5.1.3: Transmit absolute and relative location
REF 4.6: Land the drone safely
REF 4.5.2: Monitor component health
4.5.2.1: Monitor propellor and motor performance
4.5.2.2: Monitor communication strength
4.5.2.3: Monitor remaining battery
REF 4.6: Land the drone safely
REF 4.6: Land the drone safely
4.6.1: Finish all assigned inspections
4.6.2: Navigate back to the landing pad
4.6.3: Land and power down
REF 4.7: Analyze post flight
REF 4.6.1: Finish all assigned inspections
4.6.1.1: Confirm the completion of the assignment
4.6.1.2: Ensure quality of the result is satisfactory
REF 4.6.2: Navigate back to the landing pad
REF 4.6.2: Navigate back to the landing pad
4.6.2.1 Determine relative location of landing pad
4.6.2.2: Avoid obstacles
REF 4.6.3: Land and power down
REF 4.6.3: Land and power down
4.6.3.1: Touch down gently
4.6.3.2: Disengage the motors and turn off drone
REF 4.7: Analyze post flight
REF 4.7: Analyze post flight
4.7.1: Transmit the data for processing
4.7.2: Perform post-flight inspection
REF 4.7.1: Transmit the data for processing
4.7.1.1: Use the AI model to detect faults using the visual images
4.7.1.2: Use another AI model to detect faults using thermal images
4.7.1.3: Compare recorded 3D model with perfect model
REF 4.7.2: Perform post-flight inspection
Did the drone pass inspection?
Yes
No
4.7.2.1: Procede to normal post-flight procedures
Are all the wind turbines inspections completed?
Yes
No
4.7.2.3: Proceed to next trip
REF 4.2: Analyze the drone pre-launch
4.7.2.4: Transport to next destination
REF 3.0: Transport
4.7.2.2: Analyze extent of damages
Is it repairable?
Yes
No
4.7.2.5: Repair damages
REF 3.0: Transport
REF 5.0: Retire
Legend:
Level 1 🟦
Level 2 🟩
Level 3 🟧
Level 4 🟥
REF 5.0: Retire
Can some part be given a second life?
Yes
No
5.1: Repurpose part
5.2: Recycle part


1.0 Design
2.0 Produce
2.1: Acquire all necessary materials/ basic components
2.2: Produce each part of the drone
2.3: Assemble all parts
2.4: Inspect for defects
3.0 Transport
3.1: Transport the drone to shore
3.1.1: Transport from factory/previous wind farm to warehouse
3.1.1.1 Load drone onto transportation vehicle
3.1.1.2 Travel to warehouse through optimal route
3.1.1.3 Unload drone safely
3.1.2: Transport from warehouse to shore
3.1.2.1 Load drone onto transportation vehicle
3.1.2.2 Travel to destination through optimal route
3.1.2.3 Unload drone safely
3.2: Set up refueling station near wind farm
3.2.1: Set up hydrogen tank
3.2.1.1 Evaluate optimal location for the tank
3.2.1.2 Safely transport tank to desired location
3.2.1.3 Install tank according to manufacturer specifications
3.2.2: Set up landing pad
3.2.2.1 Evaluate optimal location for the landing pad
3.2.2.2 Prepare designated area for installation
3.2.2.3 Install landing pad according to manufacturer specifications
3.3: Transport the drone to substation and setup refueling station
3.3.1: Place drone and respective equipment into a boat
3.3.1.1 Select appropriate boat
3.3.1.2 Safely load equipment to boat
3.3.1.3 Protect equipment from water or any potential impact
3.3.2: Sail towards substation
3.3.2.1 Plan efficient route to OSS
3.3.2.2 Course correct path during trip
3.3.2.3 Arrive at OSS
3.3.3: Set up hydrogen tank (Similar to 3.2.1)
3.3.4: Set up landing pad (Similar to 3.2.2)
4.0 Operate
4.1: Perform Vehicle Route Planning
4.1.1: Import the coordinates of the turbines and substation
4.1.1.1 Gather necessary location data
4.1.1.1.1 Convert coordinates to correct reference frame (if required)
4.1.1.2 Ensure the coordinates are correct
4.1.1.3 Import to drone
4.1.2: Perform the algorithm to find the route to be taken
4.1.2.1 Retrieve imported coordinates
4.1.2.2 Initialize algorithm
4.1.2.2.1 Set up inital conditions and parameters required for the route planning algorithm
4.1.2.3 Calculate route
4.1.2.4 Optimize path
4.1.3: Provide waypoints and order of turbines to be inspected
4.1.3.1 Ensure waypoints are compatible with the navigation system
4.1.3.2 Adapt in real-time the inspection order or waypoint distribution (if needed)
4.2 Analyze the drone pre-launch
4.2.1 Perform initialization checks
4.2.1.1 Check drone hydrogen & battery level
4.2.1.2 Calibrate sensors
4.2.1.2.1 Establish nominal sensor management
4.2.1.2.2 Confirm measurements are recording
4.2.1.3 Calibrate cameras
4.2.1.4 Confirm communication system is operational
4.2.1.5 Ensure GPS signal strength is sufficient
4.2.1.6 Verify the telemetry data is being transmitted
4.2.2 Take off from launch pad
4.2.2.1 Start the propellors
4.2.2.2 Initialize Takeoff
4.2.2.3 Receive take-off command
4.2.2.4 Increase propellor thrust
4.3 Fly towards target turbine
4.3.1 Monitor the drone's position
4.3.1.1: Generate path towards next waypoint
4.3.1.2: Adjust flight path based on real-time position
4.3.1.3: Avoid any obstacles and the wind turbine itself
4.3.2 Approach wind turbine
4.3.2.1 Check the current position is the desired destination
4.3.2.2 Ascend to appropriate path altitude
4.4 Inspect wind turbine
4.4.1 Prepare for inspection
4.4.1.1 Generate Inspection Path
4.4.1.2 Stabilize the drone
4.4.1.3 Create 3D mapping of wind turbine
4.4.2 Capture high resolution images
4.4.2.1 Adjust position for clear images
4.4.2.1.1 Ensure even lighting conditions for image capture
4.4.2.1.2 Maintain a constant distance from the tower for image quality
4.4.2.2 Capture image
4.4.2.3 Store data in SD card
4.4.3 Capture non-contact testing
4.4.3.1 Adjust position for desired view
4.4.3.2 Store test data
4.4.4 Capture thermal images
4.4.4.1 Make sure the readings are well calibrated
4.4.4.2 Adjust camera view for desired angle
4.4.4.3 Maintain a consistent distance from the tower for good image quality
4.4.4.4 Use test method to test defect
4.4.4.5 Store test data
4.5 Communicate status with ground station
4.5.1 Track drone location
4.5.1.1: Use GPS to find absolute location
4.5.1.2: Use LIDAR to find relative location to turbines
4.5.1.3: Transmit absolute and relative location
4.5.2 Monitor component health
4.5.2.1: Monitor propellor and motor performance
4.5.2.2: Monitor communication strength
4.5.2.3: Monitor remaining battery
4.6 Land drone safely
4.6.1 Finish all assignments
4.6.1.1 Confirm the completion of the tasks
4.6.1.2 Ensure quality of result is satisfactory
4.6.2 Navigate back to the landing pad
4.6.2.1 Obtain location of landing pad
4.6.2.2 Avoid obstacles
4.6.3: Land and power down
4.6.3.1 Touch down gently
4.6.3.2 Disengage the motors and turn off drone
4.7 Perform Post-flight
4.7.1 Transmit data for processing
4.7.1.1 Use the AI model to detect faults using the visual images
4.7.1.2: Use another AI model to detect faults using thermal images
4.7.1.3: Compare recorded 3D model with perfect model
4.7.2 Perform post-flight inspection
4.7.2.1: Procede to normal post-flight procedures
4.7.2.2: Analyze extent of damages
4.7.2.3: Proceed to next trip
4.7.2.4: Transport to next destination
4.7.2.5: Repair damages
5.0 Retire
5.1 Repurpose part
5.1.1 Put the part into another drone
5.1.2. Sell drone part
5.2 Recycle part
Legend:
Level 1 🟦
Level 2 🟩
Level 3 🟧

3: Product Description

This chapter provides a high-level overview of the final design of the drone. Many specifications of the drone are also presented along with the overall architecture of the software and hardware, as presented in section 3.1. This is followed by mass, cost, and power budgets in section 3.2. The chapter further includes a discussion on the technical risk assessment in section 3.3. Finally, a compliance matrix is presented in section 3.4.

3.1 Specifications and diagrams

Author: Arham, Enes, Michael

3.1.1 Specifications

The following Table 3.1 details the main specifications of the final design of the drone. The reasoning behind these values shall be further explained throughout the report.

Table 3.1: Table detailing the main specifications of the final design of the drone.

Type	Parameter	Value
Main Specifications	Mass $[kg]$	11.84
	Power $[W]$	2400
	Cost [€]	59932
Surface Areas	S_x $[m^2]$	0.085
	S_y $[m^2]$	0.747
	S_z $[m^2]$	0.2095
Moments of Inertia	I_{xx} $[kgm^2]$	1.128
	I_{yy} $[kgm^2]$	0.615
	I_{zz} $[kgm^2]$	1.512
Flight Performance	Endurance $[h]$	3.5
	Range $[km]$	280
	Max. Speed $[m/s]$	35
Dimensions	Length $[m]$	0.247
	Width $[m]$	0.247
	Height $[m]$	0.615

3.1.2 Hardware block diagram

The hardware block diagram, shown in Figure 3.1, provides a visual representation of the drone's hardware architecture for wind turbine maintenance inspections. It includes the flight control system, sensors, communication modules, power and propulsion system, and any additional auxiliary hardware required for specific tasks. Understanding this diagram allows for the analysis of the connections and interactions between these components, enabling optimization of the drone's performance and reliability during operation.

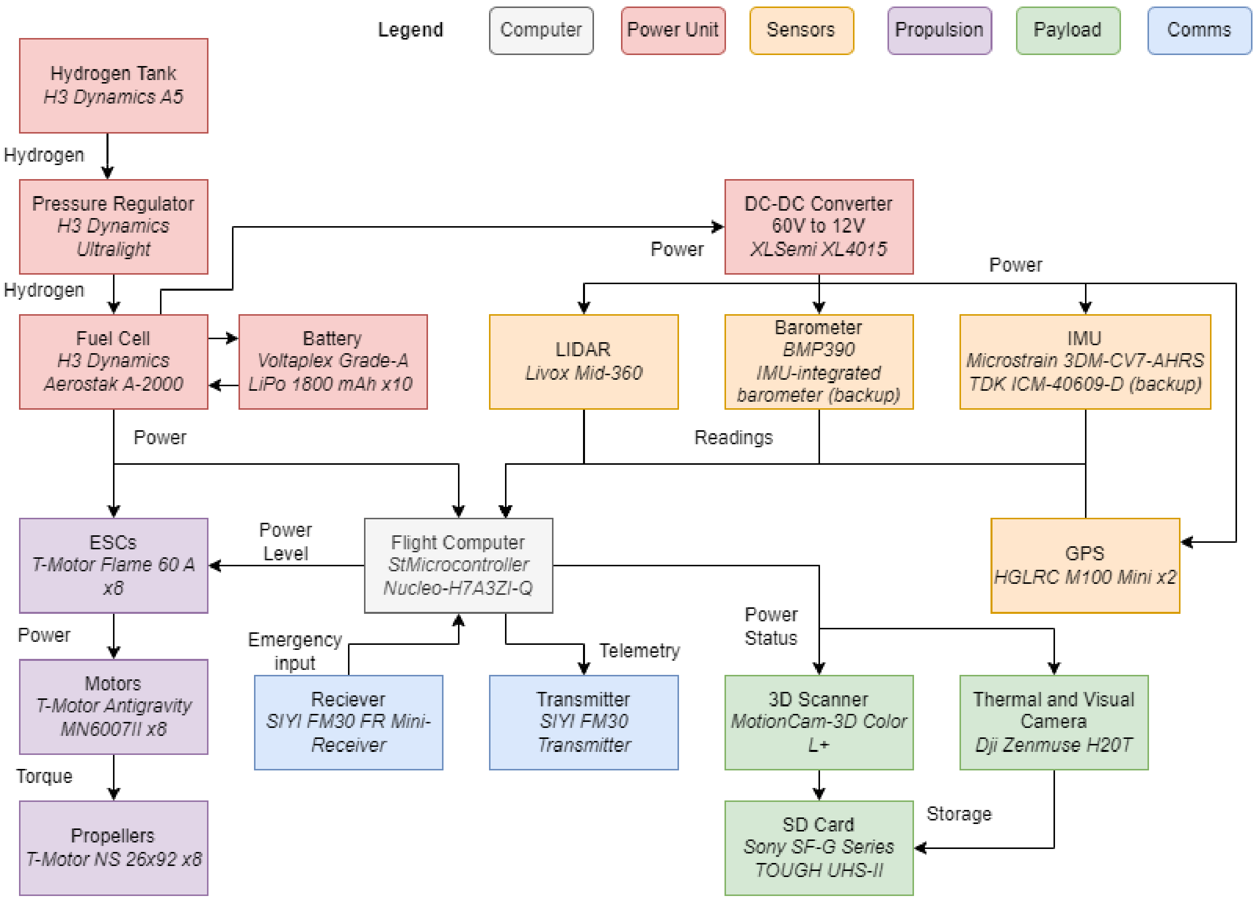


Figure 3.1: Hardware Block Diagram

3.1.3 Software block diagram

The software block diagram is given in Figure 3.2. The main systems that comprise the software logic are the guidance, navigation and control modules. The navigation module is responsible for estimating the position and orientation of the drone through multiple sensors. An estimated state is then passed onto the guidance module, which together with the mission information, generates the trajectory the drone must follow throughout the flight to reach its end goal. The guidance module will output a reference position and yaw which become inputs to the control module. The control module also takes the estimated position and yaw as input from the navigation module. Using the reference and estimated state, the control module is then responsible for calculating and allocating the control commands for the actuators. Each arrow is also labelled with the number of inputs or outputs in that signal, as well as its type, where for example 8f, represents 8 floating point values.

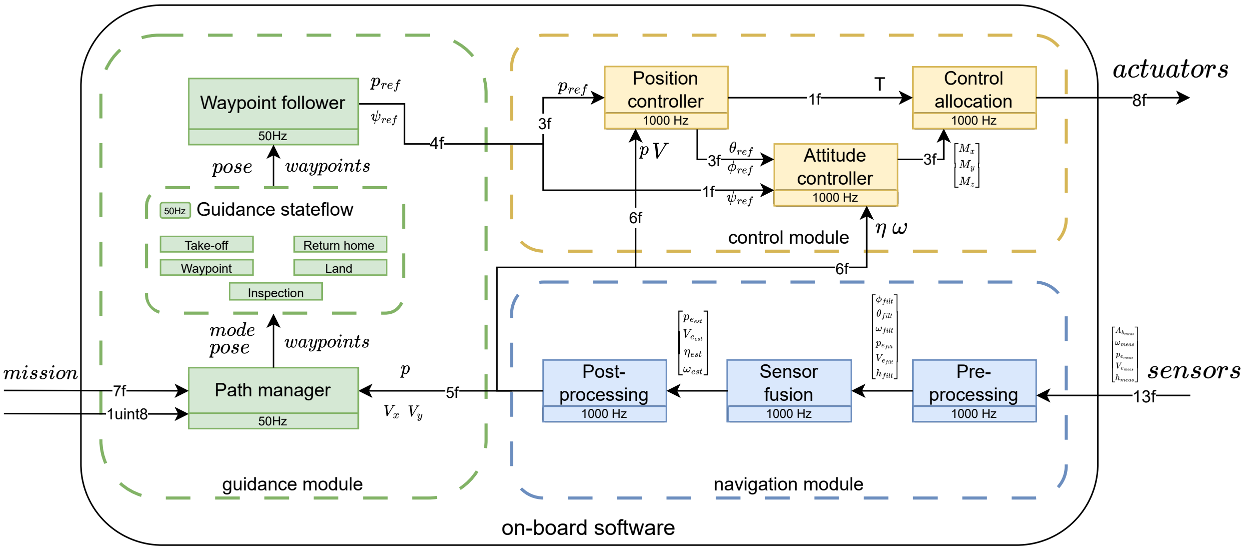


Figure 3.2: Software block diagram of the drone

3.1.4 Electrical block diagram

The electrical block diagram, given in Figure 3.3, provides a schematic representation of the drone's electrical architecture for wind turbine maintenance inspections. It illustrates the interconnections and functionalities of various electrical components and subsystems within the drone system. It allows for the design of the power distribution system and enables the optimization of electrical efficiency. The power source consists of a fuel cell and a battery, the battery gets charged from the excess energy produced by the fuel cell and in times of peak load the battery supplies energy to the circuit increasing the maximum load that can be sustained for a short period.

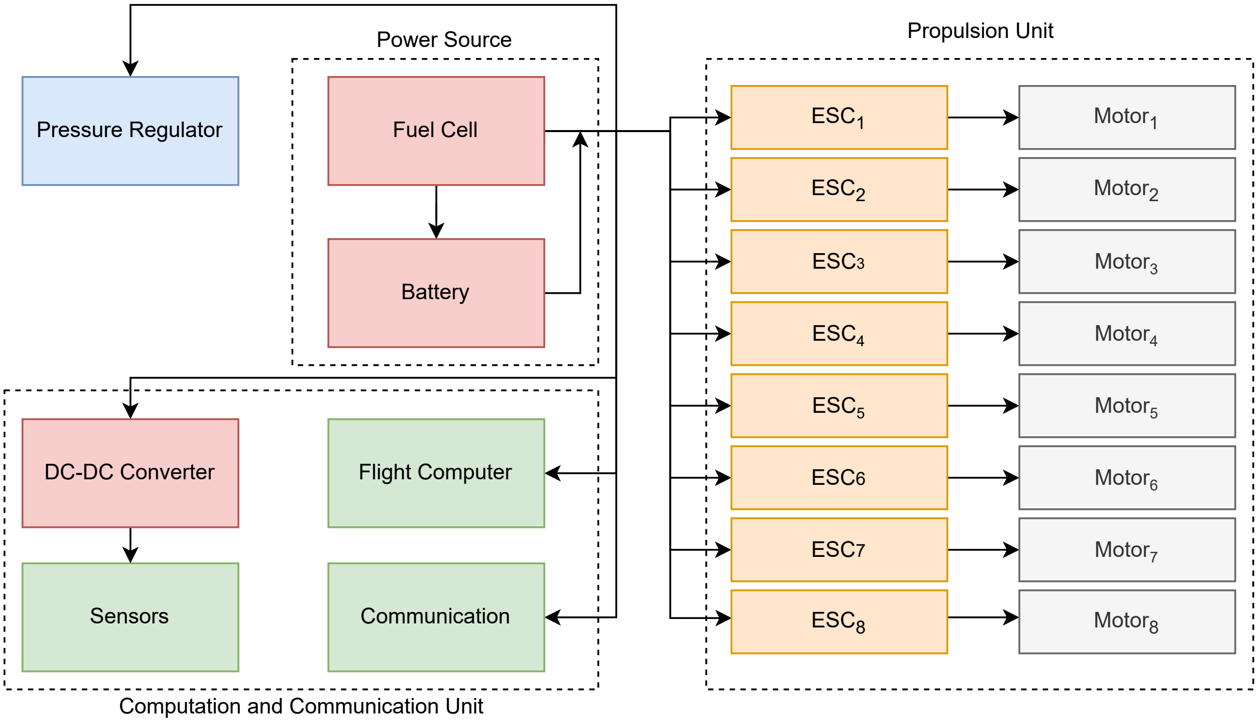


Figure 3.3: Electrical block diagram of the drone

3.1.5 Data handling block diagram

The data handling block diagram, depicted in Figure 3.4, provides an overview of the drone's data processing and storage architecture for wind turbine maintenance inspections. It illustrates the flow and handling of data within the system, including data acquisition, processing, storage, and transmission. Understanding this diagram provides insights into the interfaces between components and is particularly useful for the selection of the flight controller.

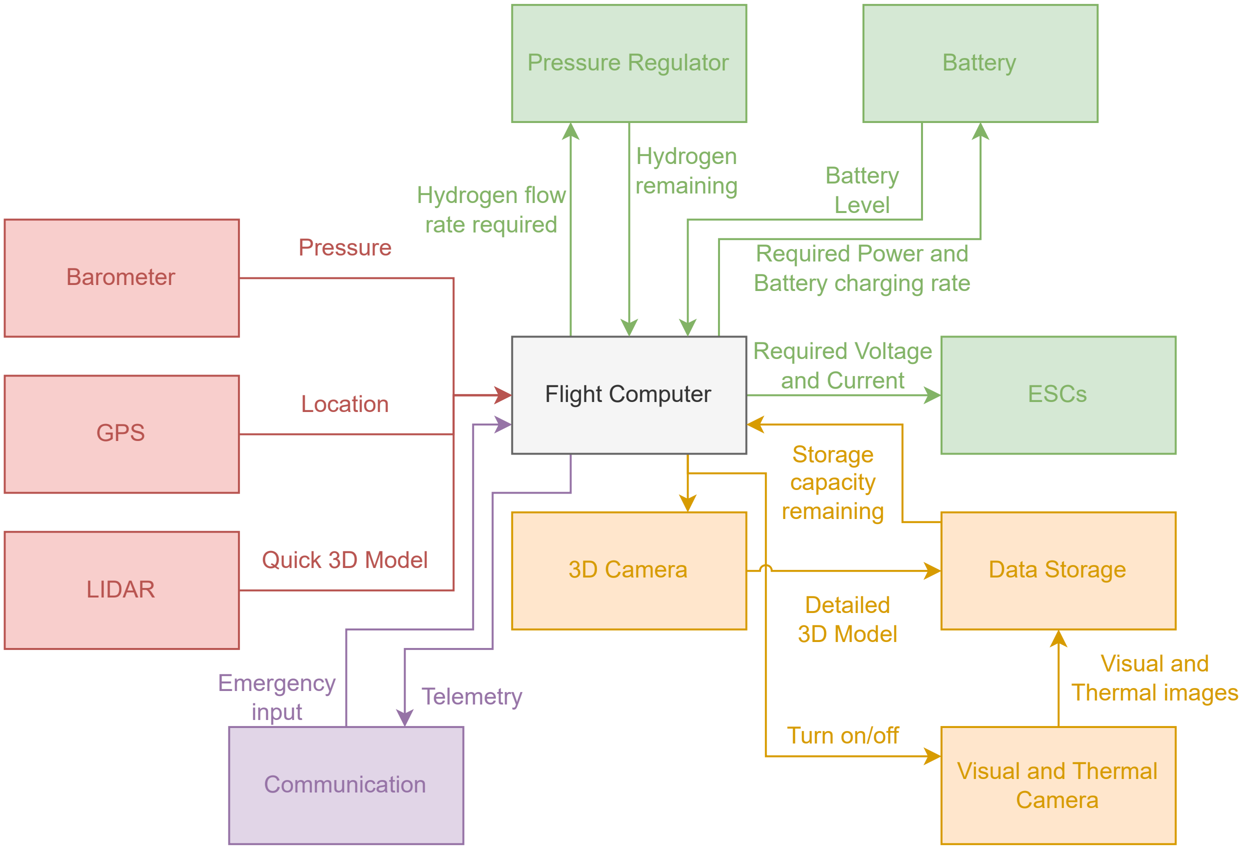


Figure 3.4: Data Handling block diagram

3.2 Mass, Power & Cost Budget

Author: Arham, Enes, Tomás

In this section, a comprehensive examination of the budget for TurbEye is undertaken, specifically focusing on the aspects of mass, power, and cost. As the drone needs to be commercially viable, understanding the intricate balance between these critical factors becomes crucial for the drone's successful design and operation, as well as its profitability. Table 3.2 showcases a detailed segmentation of the system's costs, along with the respective mass and power of the drone. A more elaborate cost breakdown shall be done afterwards alongside the production and operation costs in chapter 8.

Table 3.2: Mass, Power & Cost Budgets per component

Subsystem	**Component**	**Name**	**#**	**Mass (kg)**	**Power (W)**	**Cost (€)**
Power Unit	Fuel cell	A-2000	1	3.0[11]	-	25000[1]
	Backup battery	Voltaplex LiPo 1800 mAh	10	0.047[12]	-	50[1]
	Brackets	Brackets of the fuel cell	1	0.080[2]	-	30[1]
	Pressure regulator	Ultralight H2 Pressure Regulator	1	0.200[11]	≈ 0	4000 [1]
	Hydrogen Tank	H3 Dynamics A5	1	1.65[11]	-	
	DC converter	XLSemi XL4015	1	0.020[13]	≈ 0	5[13]
	Battery enclosing	Enclosing of the backup battery	1	0.100[2]	-	30[1]
	Fuel	Hydrogen gas	-	0.12[11]	-	1.2 [1,3]
Sensors	LIDAR	Livox Mid-360	1	0.265[14]	6.5 [14]	750[14]
	Barometer	BMP390	1	≈ 0 [15]	≈ 0[15]	2[15]
	IMU	MICROSTRAIN 3DM-CV7-AHRS	1	0.0083[16]	0.280[16]	672[16]
	GPS	HGLRC M100 MINI GPS	2	0.0027[17]	0.2 [17]	16 [17]
Compute Unit	Flight Controller	Pixhawk Cube Orange+	1	0.073[18]	14 [18]	410[19]
Comms	Transmitter	SIYI FM30	1	0.036 [20]	0.7[20]	50 [21]
	Receiver	SIYI FR Mini Receiver	1	0.0015[20]	≈ 0	
Payload	3D Scanner	MotionCam-3D Color L+	1	1.150[22]	60 [22]	13000 [23]
	Camera	DJI Zenmuse H20T	1	0.828[24]	27[24]	12000 [25]
	SD Card	Sony SF-G Series TOUGH UHS-II 128 GB	1	≈ 0	≈ 0	210 [26]
Structures	Cables	Custom Cable	1	0.049[2]	≈ 0	50 [1]
	Chassis	Custom Chassis	1	1.52[2]	10	250[1]
Propulsion	Propellers	T-Motor NS 26x92	8	0.048 [27]	-	135[27]
	Motors	T-Motor Antigravity MN6007II KV160	8	0.159[27]	285 [4]	130[27]
	ESCs	T-Motor Flame 60A	8	0.0735[27]	-	100[27]
TOTAL				**11.878**	**2400**	**59932**

Concerning the table's data collection, a majority of the items have been retrieved from the element's specific datasheet. Additionally, some of the components didn't have a listed price and only on request was an approximate price given, as was the case for most of the high-cost components. Finally, for the custom cable and chassis, an estimation was performed regarding the costs of the materials required, along with their manufacturing.

Comparing the previous mass estimation provided in the midterm report with the actual mass of the design, it can be concluded that the mass has increased by 1 kg. This is significant information since it demonstrates that the preliminary estimation was fairly accurate. However, a notable difference can be seen in the current power and cost budgets, as the previous model did not contain specific components. Thus, with all subsystems exactly defined, both budgets fell in the expected range.

3.3 Technical risk assessment

Author: Louis, Anton

In risk management, it is crucial to identify the possibility that a certain technical requirement of the system may not be achieved. This technical risk arises when the system fails to meet its performance requirements. Subsequently, the risk is assessed to determine the probability of failure and the potential consequences associated with it. Firstly, potential risks are identified in the following section.

---

[1] Approximate values from reference items or company conversations

[2] Taken from CAD

[3] Cost of fuel per trip

[4] For a thrust-to-weight ratio of 2

3.3.1 Risk identification

The identified technical risks along with the cause, event and consequence for the system are given in Table 3.7 and for the subsystems in Tables 3.5, 3.4, 3.5 and 3.6. The risks have been categorized into system risks, such as maximum take-off weight and being hit by a bird, and subsystem risks, such as a loss of a propeller for the propulsion system.

Table 3.3: System risk identification

Identifier	**Risk title**	**Cause**	**Event**	**Consequence**
TR-SYS-01	Endurance requirement	The drone is designed such that the endurance requirement is not met	The drone does not meet the endurance requirement	The operation might not be completed successfully
TR-SYS-02	Range requirement	The drone is designed such that the range requirement is not met	The drone is designed such that the range requirement is not met	The operation might not be completed successfully
TR-SYS-03	Water leakage	The drone encounters a water seepage	The drone cannot withstand rainfall of at least 40 mm/h	The drone's systems malfunction
TR-SYS-04	Snow and hail	Weather conditions	The drone encounters snow and hail conditions	The drone's systems malfunction
TR-SYS-05	Maximum take-off weight	The drone is designed such that its weight exceeds maximum take-off weight	The drone exceeds the maximum take-off weight	The operation might not be completed successfully
TR-SYS-06	Maximum altitude	The navigation malfunctions or the drone has to avoid an obstacle	The drone exceeds its maximum altitude	The drone does not conform with safety risk regulations
TR-SYS-07	Maximum characteristic length	The drone is designed to have a higher characteristic length than specified in requirements	The drone exceeds the maximum characteristic length	The drone does not conform with safety regulations
TR-SYS-08	Hit by wind turbine	The wind turbine being inspected is rotating	The drone is hit by the wind turbine	Significant damage caused to the drone's systems and wind turbine
TR-SYS-09	Hit by bird	Seabirds flying in the area surrounding the wind turbines	The drone is hit by a bird	Significant damage caused to the drone's systems
TR-SYS-10	Inspection system failure	Damage to drone, faulty wiring or software-related	Malfunctioning inspection system	Drone failing to perform an inspection on a wind turbine

Table 3.4: Propulsion System Risk Identification

Identifier	**Risk title**	**Cause**	**Event**	**Consequence**
TR-PROP-01	Loss of propeller	Damage to the drone, faulty wiring or motor malfunction	The drone loses functionality of one of its propeller units	The drone experiences loss of control
TR-PROP-02	Hit by lightning	Due to lightning the hydrogen tank explodes	The drone is in operation when it should have been observed from the weather forecast that it was unsafe to do so.	The drone is damaged beyond repair.

TR-PROP-03	Power failure	There are multiple causes for power/battery failure: faulty wiring, battery failure and insufficient charge	The drone experiences a power failure or battery failure	The drone's systems suffer significant damage, resulting in a potential loss of drone
TR-PROP-04	Hydrogen leakage	Mechanical failure, faulty seals, or damage to storage tanks or pipelines	Hydrogen leaks from the system	Increased risk of fire, explosion, and asphyxiation hazards
TR-PROP-05	Combustion hazards	Improper handling, storage, or ignition sources near hydrogen storage areas	Hydrogen ignites, leading to uncontrolled combustion	Fire, property damage, and potential harm to personnel
TR-PROP-06	Engine overdrawing or underdrawing	Too much or not enough current is provided	Engine fails	The drone will have to fly with fewer engines, and suffers from increased risk of fire hazard

Table 3.5: Structures System Risk Identification

Identifier	**Risk title**	**Cause**	**Event**	**Consequence**
TR-STR-01	Load conditions	The drone experiences loads that exceed design requirements	The drone is not able to withstand specified loading conditions	Significant damage to the drone's systems
TR-STR-02	Resonance	Eigenfrequency drone meets is equal to the frequency of the vibrations induced by the propulsion system	The structure of the drone will start resonating	Significant damage to the drone's systems
TR-STR-03	Corossion	Galvanic corrosion occurs due to interference of salty seawater with the chassis	The chassis will exhibit corrosion	The chassis might experience structural degradation over time
TR-STR-04	Loosening Bolts	Vibrations induced into the chassis by propulsion system	Bolts loosen	Components might detach from the chassis
TR-STR-05	Unstable landing gear	Landing skids are too small	Drone might fall over during landing	Damage to the drone's systems

Table 3.6: Control System Risk identification

Identifier	**Risk title**	**Cause**	**Event**	**Consequence**
TR-CON-01	Loss of communication	Drone out of range, faulty transceiver or receiver, faulty wiring	The communication link between drone and operator is lost	The operator is not able to take control over the drone
TR-CON-02	Wind gusts	The drone is designed to withstand wind gusts of 35[km/h]	The drone encounters wind gusts of more than 35 [km/h]	The drone loses control and flight stability
TR-CON-03	Flight termination system failure	The Drone is designed such that the Flight Termination System is faulty	The Flight Termination System fails	The drone cannot return to desired target
TR-CON-04	Hijacking	A third party hijacks the drone	The drone is designed without proper protection from third parties	Loss of control over the drone

TR-CON-05	Navigation error	GPS Antenna malfunction or Software Bug	The navigation system malfunctions, leading to following the flight path incorrectly	The drone cannot reach the desired target or the drone hits an element of the offshore wind infrastructure

3.3.2 Risk assessment

After briefly explaining each technical risk, it is necessary to provide an assessment of their severity and occurrence. A detailed description of each severity and occurrence for each risk is provided, followed by a risk map.

Severity of Consequence

- Catastrophic: Complete operation failure or significant failure to achieve technical performance objectives.
- Critical: Questionable operation success or significant reduction in technical performance.
- Significant: Operational delays or a noticeable reduction in technical performance
- Marginal: Degradation of secondary mission or slight reduction in technical performance
- Negligible: Minor inconvenience or negligible impact on the operation.

Probability of Occurrence

- Very High: Probability of occurrence above 80%
- High: Probability of occurrence between 60% and 80%
- Moderate: Probability of occurrence between 40% and 60%
- Probability of occurrence between 10% and 40%
- Very Low: Probability of occurrence below 10%

Table 3.7: List of risks and their likelihood and impact on the operation.

Identifier	Likelihood	Impact	Risk	Risk driver
TR-SYS-01	2	3	6	Design flaw
TR-SYS-02	2	3	6	Design flaw
TR-SYS-03	1	4	4	Weather conditions
TR-SYS-04	4	3	12	Weather conditions
TR-SYS-05	3	3	9	Design flaw
TR-SYS-06	3	4	12	Design flaw
TR-SYS-07	3	3	9	Design flaw
TR-SYS-08	4	5	20	Operational environment
TR-SYS-09	4	4	16	Operational environment
TR-PROP-01	3	5	15	Operational environment
TR-PROP-02	3	5	15	Weather conditions
TR-PROP-03	3	5	15	Design flaw
TR-PROP-04	2	4	8	Design flaw
TR-PROP-05	2	5	10	Design flaw
TR-PROP-06	3	4	12	Design flaw
TR-STR-01	3	3	9	Design flaw
TR-STR-02	1	5	5	Design flaw
TR-STR-03	2	1	2	Operational environment
TR-STR-04	2	4	8	Design flaw
TR-STR-05	2	3	6	Design flaw

Table 3.7: List of risks and their likelihood and impact on the operation.

Identifier	Likelihood	Impact	Risk	Risk driver
TR-CON-01	3	4	12	Drone telemetry
TR-CON-02	4	4	16	Weather conditions
TR-CON-03	2	3	6	Design flaw
TR-CON-04	2	5	5	Design flaw
TR-CON-05	3	4	12	Software Bugs

To visualize the identified risks and their severity, a colour-coded grid has been created, as shown in Figure 3.5. This grid provides a clear overview of the risks and helps identify those that are deemed unacceptable, requiring mitigation strategies to reduce their severity or probability. The risk is assessed by multiplying the likelihood and impact of the event, and if the resulting risk score is equal to or greater than 10, the risk is considered unacceptable. This visual representation highlights the risks that need immediate attention. Additionally, some risks that may not be deemed unacceptable but have relatively simple mitigation strategies should still be addressed to minimize their impact.

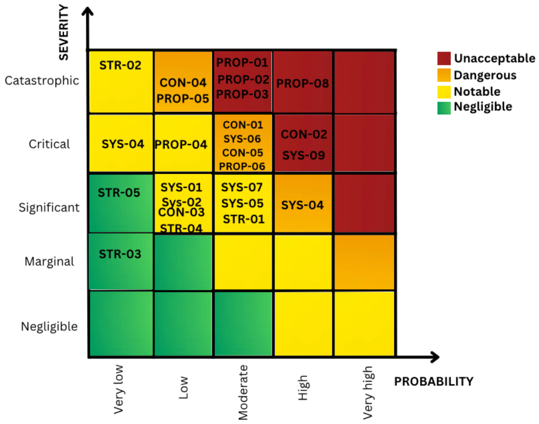


Figure 3.5: Technical Risk Map

3.3.3 Risk mitigation

To lower the impact of risks on the operation, risk mitigation techniques have to be drawn, which are given in Table 3.8. The mitigated technical risk map is given in Figure 3.6.

Table 3.8: Mitigation strategies drawn

Identifier	Risk	Mitigation strategy	Description
TR-SYS-01	6	Accept	The possibility that a higher endurance is needed than specified in the requirements is low and can be assessed before commencing the operation.
TR-SYS-02	6	Accept	The possibility that a higher range is needed than specified in the requirements is low and can be assessed before commencing the operation.
TR-SYS-03	4	Accept	The drone is already designed to withstand high amounts of rain, hence worse conditions are considered to be rare and operations can be scheduled appropriately.
TR-SYS-04	12	Reduce	An emergency signal is sent to the drone when the weather prediction system predicts snow or hail and the drone will abort its mission.
TR-SYS-05	9	Reduce	The target maximum take-off weight will include adequate contingencies so that the actual maximum take-off weight is within the set requirements.

Table 3.8: Mitigation strategies drawn

Identifier	Risk	Mitigation strategy	Description
TR-SYS-06	12	Reduce	The target maximum altitude will include adequate contingencies so that the actual maximum altitude is within the set requirements.
TR-SYS-07	9	Reduce	The target maximum characteristic length will include adequate contingencies so that the actual maximum characteristic length is within the set requirements.
TR-SYS-08	20	Reduce	The drone is equipped with an obstacle avoidance system.
TR-SYS-09	16	Reduce	The drone is equipped with an obstacle avoidance system.
TR-PROP-01	15	Reduce	The drone is equipped with one motor inoperative capabilities such that the drone can stabilize itself and bring itself to safety.
TR-PROP-02	12	Reduce	The drone is cleared for takeoff only after checking the weather forecast.
TR-PROP-03	15	Reduce	The drone is equipped with an auxiliary power unit to cope with such a power loss.
TR-PROP-04	8	Reduce	A hydrogen detector is installed.
TR-PROP-05	10	Reduce	A hydrogen detector is installed.
TR-PROP-06	12	Reduce	A sensor is installed that measures the current.
TR-STR-01	9	Reduce	The target maximum load will include adequate contingencies so that the actual maximum load is within the set requirements
TR-STR-02	5	Accept	The possibility is low and can be assessed before operations
TR-STR-03	2	Accept	Corrosion is a very slow process
TR-STR-04	8	Reduce	Add washers and check bolts regularly
TR-STR-05	6	Accept	Stability problems with the landing gear will be detected well before operations commence
TR-CON-01	12	Reduce	The drone is equipped with a redundant transmitter
TR-CON-02	16	Reduce	An emergency signal is sent to the drone when the weather prediction gives wind gusts above this value and the drone will abort its mission
TR-CON-03	6	Reduce	The system is encased by a Faraday cage, uses an auxiliary power system and computing unit
TR-CON-04	5	Reduce	The drone uses a secured communication channel
TR-CON-05	12	Reduce	The drone is fitted with a redundant navigation system

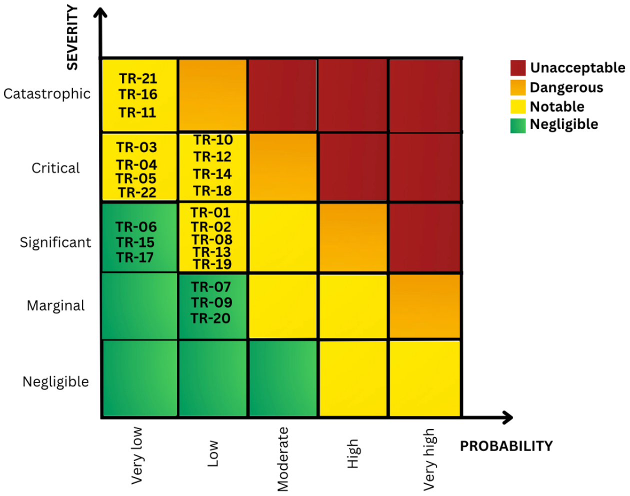


Figure 3.6: Mitigated Technical Risk Map

3.4 Compliance to Requirements

Author: Louis

Ensuring compliance with the requirements is a crucial aspect of the design. Requirements have been identified based on the stakeholders involved within the project, system criteria and subsystem criteria. The compliance matrix is discussed in that order in subsection 3.4.1, subsection 3.4.2 and subsection 3.4.3. A rationale is provided if a requirement is not met.

3.4.1 Compliance Stakeholders Requirements

In Table 3.9 the compliance matrix with the stakeholder's requirements is provided. The stakeholders for this project as identified in the baseline report are: Clients (CL), students (ST), Delft University of Technology (TUD), Environment and flight regulators (RE), Subcontractors and suppliers (CS), Technicians and operators (TO), Investors (IN) and Competitors (CO). From these stakeholder needs, the stakeholder requirements were formulated. The goal is to provide a common understanding of what their needs are, hence they give a common set of requirements and identify priorities and conflicts. As can be seen, it can be concluded that all stakeholder requirements have been met.

Table 3.9: Stakeholder requirements compliance

ID	Requirements	Compl.
AD-CL-02 (KEY)	The drone shall be able to operate autonomously.	✓
AD-CL-03 (KEY)	The drone shall offer a positive return of investment within a timeframe of 2 years.	✓
AD-CL-04 (KEY)	The drone shall be capable of using at least two different inspection methods, but not necesarilly at the same time.	✓
AD-CL-05 (KEY)	A full inspection of an offshore wind turbine shall be performed in less than 2 [hours].	✓
AD-CL-06 (KEY)	The drone shall be able to operate in nominal weather and environmental conditions experienced at offshore locations.	✓
AD-CL-07 (KEY)	The drone shall be able to perform NDT on wind turbine within reasonable accuracy.	✓
AD-CL-08 (KEY & DRIVING)	The drone shall operate in a safe manner.	✓
AD-ST-01 (DRIVING)	The project shall be completed in 10 weeks by 10 students.	✓
AD-ST-02 (KEY)	The project shall contribute to the educational experience of the student.	✓
AD-TU-01 (KEY)	A good relationship shall be maintained with entities of TU Delft for the duration of the project.	✓

3.4.2 Compliance System Requirements

System and mission requirements describe the system in general technical terms at a high level. They are formulated based on the stakeholder requirements as well as based on the functional analysis of the system. As can be seen in Table 3.10 some requirements will have to be tested in practice. This can be as of time constraint or as analysis reasons, e.g. simulating operations under 30m visibility conditions is very hard to model. Moreover in the compliance matrix it can be seen that not all requirements as of various reasons have been met:

- Req. AD-Sys-CL-03-4 (KEY): It turned out that 1.978 kg of payload is sufficient
- Req. AD-Sys-CL-03-5 (KEY): It turned out that 0.087 kW of power to the inspection module is sufficient
- AD-Sys-CL-4-1 (Driving): After a thorough analysis it determined that mounting only one module onto the drone is sufficient
- AD-Sys-CL-07-1 (KEY): An accuracy of 5mm is not necessary for the type of sensors that are currently used by the team
- AD-Sys-CL-07-2: A distance of 3.0m away from the wind turbine is sufficient
- AD-Sys-CL-08-13 (DRIVING&KEY): A floatation device has been discarded as it is not deemed necessary nor economically feasible

- AD-Sys-CL-08-2 (KEY): An Automatic Recovery System is not deemed necessary nor economically feasible
- AD-Sys-CL-08-8 (KEY): The drone shall not anymore operator on the wind turbine

Table 3.10: System requirements

ID	**Requirements**	**Compl.**
AD-Sys-CL-01-1	The drone shall be able to inspect wind turbine blades in any orientation.	✓
AD-Sys-CL-01-2	The material of the wind turbine blades shall be provided as input to the drone during the initialisation phase.	✓
AD-Sys-CL-01-3	The drone shall be able to inspect wind turbine blades deflecting at a rate of at most 5 [mm/s].	TBT
AD-Sys-CL-01-4	The drone shall be able to inspect all external surfaces of a wind turbine blade.	✓
AD-Sys-CL-02-1 (DRIVING & KEY)	The drone shall be able to navigate autonomously.	✓
AD-Sys-CL-02-2 (DRIVING & KEY)	The drone shall be able to land and take-off autonomously.	✓
AD-Sys-CL-02-3 (DRIVING & KEY)	The drone shall be able to perform inspection routines autonomously.	✓
AD-Sys-CL-03-1 (DRIVING)	The drone shall perform at least 10,000 inspection missions.	
AD-Sys-CL-03-2	The drone shall require a maximum down time of 20 [minutes] between inspections.	✓
AD-Sys-CL-03-3	The drone shall have replaceable parts.	✓
AD-Sys-CL-03-4 (KEY)	The drone shall be able to carry a payload of at least 2.5 [kg].	✗
AD-Sys-CL-03-5 (KEY)	The drone shall be able to provide a minimum power of 0.5 [kW] to the inspection module.	✗
AD-Sys-CL-04-1 (DRIVING)	The drone shall be able to mount different types of inspection modules.	✗
AD-Sys-CL-05-1 (DRIVING)	A fully charged drone shall have sufficient power to for an endurance of up to 2 [h] that includes up to 30 [min] maximum power delivered to the inspection module.	✓
AD-Sys-CL-05-2 (KEY)	The inspection routine shall not take more than 2 [hours].	✓
AD-Sys-CL-06-1 (KEY)	The drone shall be able to operate normally in the presence of wind and wind gusts of 55 [km/h]. [28]	✓
AD-Sys-CL-06-2	The drone shall be able to operate normally under visibility conditions of 30 [m].	TBT
AD-Sys-CL-06-3 (KEY)	The drone shall be able to operate normally in an environment with 95% humidity.	TBT
AD-Sys-CL-06-4 (KEY)	The drone shall be able to operate normally in rain of 40 [mm/h].	TBT
AD-Sys-CL-07-1 (KEY)	The drone shall be able to perform NDT testing if needed with an accuracy of 5 [mm] with respect to the desired measurement spot.	✗
AD-Sys-CL-07-2	The drone shall be able to acquire 3D images of the wind turbine within 1.0 [m] of it.	✗
AD-Sys-CL-07-3 (KEY)	The drone shall be able to detect damages of diameter >= 7 [mm] with a true positive rate of at least 95%.	TBT
AD-Sys-CL-07-4 (KEY)	The drone shall be able to detect damages of skin/beam dis-bonding >= 7 [mm] with a true positive rate of at least 95%.	TBT
AD-Sys-CL-08-1 (DRIVING)	The drone shall be equipped with a collision avoidance system.	✓
AD-Sys-CL-08-10	The drone shall withstand a maximum lateral acceleration of 1.2 [g].	✓
AD-Sys-CL-08-11	The drone shall withstand a maximum longitudinal acceleration of 1.2 [g].	✓
AD-Sys-CL-08-12	The drone shall be designed with a conservative design philosophy in mind.	✓
AD-Sys-CL-08-13 (DRIVING & KEY)	The drone shall support floatation in case of a sea-water landing.	✗
AD-Sys-CL-08-14 (KEY)	The drone shall communicate its geo-awareness state with the ground segment.	✓

Table 3.10: System requirements

ID	Requirements	Compl.
AD-Sys-CL-08-15	The drone shall communicate the available fuel level to the ground segment.	✓
AD-Sys-CL-08-16	The drone shall have at least 10% reserve energy of maximum energy.	✓
AD-Sys-CL-08-17 (KEY)	The drone shall not be controllable by third parties.	✓
AD-Sys-CL-08-18	The drone shall communicate the communication link status with the ground segment.	✓
AD-Sys-CL-08-2 (KEY)	The drone shall be equipped with an Automatic Recovery System (ARS).	✗
AD-Sys-CL-08-3 (KEY)	The drone shall be equipped with a Flight Termination System (FTS).	✓
AD-Sys-CL-08-4	The operator shall be able to take manually activate the FTS of the drone at any time during automatic operations.	✓
AD-Sys-CL-08-5	The operating temperature of the drone shall operate in environmental temperatures from -10 [degree C] to 40 [degree C].	TBT
AD-Sys-CL-08-6	The drone shall be able to sustain a hard landing of 2 [g] on its landing system.	TBT
AD-Sys-CL-08-7 (KILLER)	The drone shall not be rendered inoperational by lightning strike.	TBT
AD-Sys-CL-08-8 (KEY)	The drone shall only operate on the wind turbine after it has been turned off.	✗
AD-Sys-CL-08-9 (DRIVING)	The chassis shall withstand incoming hail that has an impulse upto 3 Ns.	TBT
AD-Sys-ST-01-1	The drone shall be limited to perform inspection from the external side of the turbine.	✓
AD-Sys-ST-01-2 (DRIVING)	The total time spent on the project shall not exceed 4000 [hours].	✓
AD-Sys-ST-01-3	The team shall limit the project to the design of the drone, as compared to building it.	✓
AD-Sys-ST-01-4 (DRIVING)	The project shall be limited to the design of a single drone, as compared to a swarm of drones.	✓
AD-Sys-ST-01-5 (DRIVING)	All parts and subsystems shall be developed and manufacturable in house or COTS.	✓
AD-Sys-ST-02-1 (KEY)	The students shall hand in all project deliverables not later than their respective deadlines.	✓
AD-Sys-ST-02-2 (KEY)	Each student shall fulfill both a technical and an organisational role within the team.	✓
AD-Sys-ST-02-3 (KEY)	The drone and its operations shall be designed to adhere to sustainability standards imposed by the team.	✓
AD-Sys-TU-01-1 (KEY)	The team shall adhere to the TU Delft rules for the duration of the project.	✓

3.4.3 Compliance Subsystem Requirements

Structures

Subsystem requirements describe the system in general technical terms at a lower level. They are formulated based on the system requirements as well as based on the functional analysis of the subsystem. As well as for the system requirements some requirements for the subystems will have to be tested in practice to validate compliance. The matrix can be found in Table 3.11. The rationale for not meeting all requirements is as follows:

- AD-Sys-CL-03-4-STRUCT-5: A non-detachable landing gear is lighter, moreover is it not deemed likely that the landing gear shall fail sooner then the chassis itself
- AD-Sys-CL-04-1-STRUCT-6 (DRIVING): The team has chosen to only use one inspection system, modularity is therefore not necessary
- AD-Sys-CL-08-13-STRUCT-18 (DRIVING): The materials will never be in contact with sea-water
- AD-Sys-CL-08-13-STRUCT-19 (DRIVING): A sea-water landing shall not occur

Table 3.11: Structural subsystem requirements compliance

ID	Requirements	Compl.
AD-Sys-CL-03-1-STRUCT-1	The landing gear system shall be able to withstand 1,000 nominal landings.	TBT
AD-Sys-CL-03-4-STRUCT-2 (DRIVING)	The drone shall be disassemblable.	✓
AD-Sys-CL-03-4-STRUCT-3	The lift generation system shall be detachable.	✓
AD-Sys-CL-03-4-STRUCT-4	The power unit shall be detachable.	✓
AD-Sys-CL-03-4-STRUCT-5	The landing gear shall be detachable.	✗
AD-Sys-CL-04-1-STRUCT-6 (DRIVING)	The interface mechanism of the inspection compartment shall be modular.	✗
AD-Sys-CL-06-4-STRUCT-7	No electrical wiring shall directly be exposed to the environment.	✓
AD-Sys-CL-08-5-STRUCT-8	The drone chassis shall withstand temperatures from -10 [degree C] to [40 degree C].	TBT
AD-Sys-CL-08-5-STRUCT-9	The battery housing shall withstand a temperature of 40 [degree C].	TBT
AD-Sys-CL-08-6-STRUCT-10 (DRIVING)	The landing system shall be able to withstand an impact of 2 [g] during a hard-landing.	TBT
AD-Sys-CL-08-7-STRUCT-11 (DRIVING)	The drone's structural materials shall be non-ablative.	✓
AD-Sys-CL-08-9-STRUCT-12 (DRIVING)	The drone chassis shall be able to withstand an impact load of 3[Ns].	TBT
AD-Sys-CL-08-10-STRUCT-13	The drone chassis shall withstand a maximum lateral acceleration of 1.2 [g]	✓
AD-Sys-CL-08-11-STRUCT-14	The drone chassis shall withstand a maximum longitudinal acceleration of 1.2 [g].	✓
AD-Sys-CL-08-12-STRUCT-15	A safety factor of 1.5 shall be applied to applied loads.	✓
AD-Sys-CL-08-12-STRUCT-16	All structural components shall have a margin of safety of 1.5 w.r.t. the limit loads.	✓
AD-Sys-CL-08-12-STRUCT-17	All chassis components of the drone shall be attached by means of rigid joints.	✓
AD-Sys-CL-08-13-STRUCT-18 (DRIVING)	The materials shall be compatable with sea-water.	✗
AD-Sys-CL-08-13-STRUCT-19 (DRIVING)	Upon a sea-water landing, the drone shall be in static equilibrium.	✗
AD-Sys-ST-02-3-STRUCT-20 (KILLER)	The materials used to build the drone's structure shall be at least 70% recyclable.	✓

Control

The compliance matrix can be found in Table 3.12. It can be seen that the design meets almost all requirements or compliance will have to be tested in the future. The requirements verification is found in Section 6.6.1. The rationale for not meeting all is as follows:

- AD-Sys-CL07-1-CS-5 (DRIVING): An accuracy of 5 mm with respect to the desired measurement point is deemed unnecessary for the chosen inspection methods
- AD-Sys-CL07-1-CS-5 (DRIVING): An accuracy of 5 mm with respect to the desired measurement point is deemed unnecessary for the chosen inspection methods

Table 3.12: Control subsystem requirements compliance

ID	Requirements	Compl.
AD-Sys-CL-02-1-CS-1 (DRIVING)	The drone shall be able to navigate without the intervention of the remote pilot on the control commands.	✓

Table 3.12: Control subsystem requirements compliance

ID	Requirements	Compl.
AD-Sys-CL-02-2-CS-2 (DRIVING)	The drone shall be able to land and take-off without the intervention of the remote pilot on the control commands.	✓
AD-Sys-CL-02-2-CS-3 (DRIVING)	The drone shall be able to perform inspection routines without the intervention of the remote pilot on the control commands.	✓
AD-Sys-CL-06-1-CS-4 (DRIVING)	The control subsystem shall cope with wind and wind gusts of 55 [km/h]. [28]	✓
AD-Sys-CL-07-1-CS-5 (DRIVING)	The control system shall provide the means for inspection module to be stationary with an accuracy of 5 [mm] w.r.t. the desired measurement point.	✗
AD-Sys-CL-07-1-CS-6	The position of the drone shall be known with an accuracy of at least 5 [mm].	✗
AD-Sys-CL-07-1-CS-7	The control system shall provide the means for drone to have identical performance regardless of which inspection module is fitted, if any.	✓
AD-Sys-CL-07-2-CS-8	The drone shall have a local navigation system with a range of 1 [m].	TBI
AD-Sys-CL-08-1-CS-9 (DRIVING)	The control system shall provide a change of path in order to avoid imminent collisions with obstacles.	TBI
AD-Sys-CL-08-1-CS-10 (DRIVING)	The control system shall be able to predict if a collision is going to occur with an obstacle.	TBI
AD-Sys-CL-08-4-CS-11	The control system shall be able to receive a manual command to trigger the FTS during automatic operations.	✓
AD-Sys-CL-08-9-CS-12	The control subsystem shall be able to stabilise the drone after an impact load of 3[Ns].	✓
AD-Sys-CL-08-14-CS-14 (DRIVING)	The drone shall be able to determine geo-awareness state variables.	✓
AD-Sys-CL-08-14-CS-15	The drone shall be able to determine its current fuel level.	✓
AD-Sys-CL-08-14-CS-18	The control subsystem shall facilitate immediate return to ground station in case reserve fuel level reaches 10%.	✓

Aerodynamics

The aerodynamic compliance matrix can be found in Table 3.13. For the aerodynamics subsystem it can also be concluded that all requirement have been met.

Table 3.13: Aeroydynamics subsystem requirements compliance

ID	Requirements	Compl.
AD-Sys-CL-05-2-AERO-1 (DRIVING)	The drone shall be able to accelerate its vertical translational speed with 0.2 [g].	✓
AD-Sys-CL-05-2-AERO-2 (DRIVING)	The drone shall be able to accelerate its horizontal translational speed with 0.2 [g].	✓

Performance

Regarding the performance system, as can be seen in Table 3.14, the following can be denoted regarding the requirements that have not been met:

- AD-Sys-CL-08-7-PERF-5: There is no ARS unit present on the drone
- AD-Sys-CL-03-3-PERF-15 (KEY): After analysis a payload capacity of 1.987 kg was deemed sufficient
- AD-Sys-CL-03-5-PERF-16 (KEY): After analysis a power provision of 0.087 kW was deemed sufficient

Table 3.14: Performance requirements

ID	Requirements	Compl.
AD-Sys-CL-03-1-PERF-1	The power unit shall be replaced or serviced after [TBD] inspections.	TBT

Table 3.14: Performance requirements

ID	Requirements	Compl.
AD-Sys-CL-03-1-PERF-2	The lift generation subsystem shall be replaced or serviced after [TBD] hours of flight. *	TBT
AD-Sys-CL-05-1-PERF-3 (DRIVING)	The power unit shall provide the drone with an endurance of upto 2 [h] that includes upto 30 [min] maximum power delivered to the inspection module.	✓
AD-Sys-CL-06-3-PERF-4 (DRIVING)	All parts of the drone shall be unaffected by humidity of 95%.	TBT
AD-Sys-CL-08-7-PERF-5	The ARS unit shall be protected against a lightning strike.	✗
AD-Sys-CL-08-8-PERF-6	The navigation system shall create paths that avoid the wakes of wind turbines.	✓
AD-Sys-CL-05-2-PERF-7	The drone shall have a thrust to weight ratio of at least 2.	✓
AD-Sys-ST-02-3-PERF-13	The drone shall be transported using vehicles with a maximum of 50g of CO2 emissions per km.	✓
AD-Sys-ST-02-3-PERF-14	The drone shall have a maximum of 500g of CO2 emissions per hour under the most energy intensive operating condition.	✓
AD-Sys-CL-03-3-PERF-15 (KEY)	The payload capacity of the drone shall be at least 2.5 [kg].	✗
AD-Sys-CL-03-5-PERF-16 (KEY)	The power unit shall provide inspection module with at least 0.5 [kW] of power.	✗

Organisation

Regarding the organisational requirements, as seen in Table 3.15, it can be seen that all requirements have been met.

Table 3.15: Organisational requirements

ID	Requirements	Compl.
AD-Sys-ST-02-1-ORG-1 (KEY)	The students shall follow the Gantt chart of the project.	✓
AD-Sys-ST-02-2-ORG-2	The students shall adhere to the roles described in the organogram.	✓
AD-Sys-TU-01-1-ORG-3 (KEY)	The team shall follow rules set out by the TU Code of Conduct.	✓
AD-Sys-TU-01-1-ORG-4 (KEY)	The team shall follow rules set out by Facility Management.	✓
AD-Sys-TU-01-1-ORG-5 (KEY)	The team shall follow rules set out by TU Delft Aerospace Engineering.	✓

4: Propulsion Subsystem Design

Author: Arham

The propulsion system of the drone comprises six main components: The propeller, motor, Electronic Speed Controller (ESC), fuel cell, hydrogen tank and back-up battery. The propeller generates thrust, the motor drives the propeller, and the ESC controls the power and voltage supplied to the motor. In terms of power delivery, the system includes a fuel cell and a backup battery for power generation. To supply the power source, a hydrogen tank and pressure regulator are utilized. Given the wide range of options available in the market, commercially available off-the-shelf (COTS) components were selected for the propulsion system.

In section 4.1, the assumptions of the model are listed. Following that, the database of commercial off-the-shelf (COTS) components is compiled and presented in section 4.2. The modeling of the propeller and motor is explained in section 4.3 and section 4.4, respectively. Various combinations of the propeller and motor are then evaluated in section 4.5. The chosen electronic speed controller (ESC) and fuel cell are modeled, and their performance limitations are described in section 4.6. The range of the configuration is calculated for a variety of speeds and drag coefficients in section 4.7. The modeling of the backup battery, which is needed to supply sufficient power for peak loads and act as a backup in case the fuel cell fails, is discussed in section 4.8. Sensitivity analysis is performed in section 4.9, followed by verification and validation in section 4.10 and section 4.11, respectively.

4.1 Assumptions

Author: Enes

To aid in the decision-making process, an analysis tool was developed using Python. This tool allowed for the estimation and evaluation of the propulsion system's performance, including but not limited to the thrust, torque and power. The tool was later verified and validated to ensure its accuracy and reliability. Various assumptions were made during the analysis which are outlined in Table 4.1.

Table 4.1: List of assumptions

Identifier	Description
ASM-PS-01	The components of the propulsion subsystem, including a fuel cell, motors and ESCs, have an instant and accurate reaction to an input.
ASM-PS-02	The components of the propulsion subsystem are all rigid and show no change in aerodynamic characteristics.
ASM-PS-03	The environmental conditions (density, temperature, viscosity, etc.) the drone operates in are constant and equal to ISA at sea-level conditions.
ASM-PS-04	The thrust and moment coefficient of the propellers do not vary with increasing RPM.
ASM-PS-05	The motors are brushless permenant magnet DC motors.
ASM-PS-06	The armature inductance and the transient process caused by switching elements within the motor are negligible.
ASM-PS-07	The backup battery has a minimum discharge capacity of 15% of the total capacity.
ASM-PS-08	The coaxial rotors have a 90% aerodynamic efficiency compared to rotors in a conventional layout.
ASM-PS-09	Aerodynamic forces and interactions are negligible besides thrust and drag originating from the drone.
ASM-PS-10	The drone is aerodynamically equivalent to a cuboid.
ASM-PS-11	The wind only has a horizontal component.
ASM-PS-12	The power consumption during the inspection is equal to 1.2 times the power required during hovering.
ASM-PS-13	The drag coefficient of the drone in all directions is equal to 0.80.

Table 4.1: List of assumptions

Identifier	**Description**
ASM-PS-14	The center of gravity is equidistant from all propellers.
ASM-PS-15	The thrust acts perpendicular to the propeller at all pitch angles.
ASM-PS-16	The performance of the components are constant with time and
ASM-PS-17	There are no environmental disturbances that affect the drone.
ASM-PS-18	The motors of the drone spin with the same RPM at all times.
ASM-PS-19	The pitching of the drone is instantenous.
ASM-PS-20	The mass of the drone is constant throughout the flight.
ASM-PS-21	The power loss from the cables are negligible.
ASM-PS-22	The chassis mass is assumed to be constant regardless of motor or propeller used.

4.2 Propeller and Motor Database

Author: Arham

As COTS components are used, a propeller and motor database was compiled for various options and all their relevant parameters were included. Considering that similar drones with high maximum take-off weights usually took propellers with diameters around 20 inches and motors with relatively low KV values (lower than 400), the database was mainly concentrated around that configuration.

A collection of 28 propellers is used, and their properties were found from the database on TytoRobotics [29] and their respective product pages [27] [30]. The propeller diameter ranges from 15 inches (0.381 m) to 62 inches (1.575 m), and the propeller pitch ranges from 4.5 inches (0.1143 m) to 24 inches (0.6096 m). The mass of each propeller ranges from 21 g to 526 g. Apart from size, the other factor that affects mass is whether it is made from carbon fiber or polymer core. Although carbon fiber is a much stronger material, it is not necessary unless the propeller is extremely large, such as the T-motor 62x24 CF and Xoar PJP-T-L 47x10. For smaller propellers, polymer cores can be used to significantly reduce weight.

A collection of 27 distinct motors and their properties were found by searching for a specific motor on TytoRobotics [29] and then recording all variants of it from the product pages [27][30]. Although a lower KV value is beneficial for performance, a trade-off can be observed with low KV motors being much heavier.

4.3 Propeller Characteristics

Author: Arham

To get an estimate for the performance of the propeller, a simple model is made. The formulas as described in eq. (4.1) and eq. (4.2) are used to calculate the thrust and moment produced [31].

$$T = C_T \rho \left(\frac{N}{60}\right)^2 D_p^4 \tag{4.1}$$

$$M = C_M \rho \left(\frac{N}{60}\right)^2 D_p^5 \tag{4.2}$$

In these N is the propeller angular speed (in RPM) and D_p is the diameter of the propeller, C_T and C_M are the dimensionless thrust and torque coefficients respectively, and ρ is the air density. ρ can be found using eq. (4.3).

$$\rho = \frac{T_0 p}{p_0 T} \rho_0 \tag{4.3}$$

T_0 and p_0 are in this equation the temperature and pressure at standard conditions. The temperature T and pressure p can be calculated using Equation 4.4 and Equation 4.5.

$$T = T_0 + \lambda h \tag{4.4}$$

$$p = p_0 \cdot \left(1 - \frac{\lambda h}{T}\right)^{\frac{-g_0}{\lambda R}} \tag{4.5}$$

C_T and C_M can be expressed as a function of propeller parameters, as shown below in eq. (4.6) and eq. (4.7) [31].

$$C_T = 0.25\pi^3\lambda\zeta^2 B_p K_0 \frac{\epsilon \arctan\left(\frac{H_p}{\pi D_p}\right) - \alpha_0}{\pi A + K_0} \tag{4.6}$$

$$C_M = \frac{1}{8A}\pi^2 C_d \zeta^2 \lambda B_p^2 \tag{4.7}$$

$$C_d = C_{fd} + \frac{\pi A K_0^2}{e}\left(\frac{\epsilon \arctan\left(\frac{H_p}{\pi D_p}\right) - \alpha_0}{\pi A + K_0}\right)^2 \tag{4.8}$$

H_p is the propeller pitch, B_p is the number of blades, and D_p is the blade diameter. The remaining coefficients are aerodynamic coefficients that describe the propeller properties, which also influence the Thrust T, Drag D, and Torque M. As these depend on the exact shape of the propeller, it is not possible to determine these coefficients at such an early stage of the design, and instead, typical values from literature and experimental data have been used. The values are listed in Table 4.2.

Table 4.2: Mean Parameters from Literature [31]

Parameter	**Value**
A, Aspect Ratio	5
ϵ, Oswald Factor	0.85
λ	0.75
ζ	0.5
e	0.83
C_{fd}, Zero-Lift Drag Coefficient	0.015
α_0	0
K_0	6.11

This means that the following inputs are left to the designers choice: Number of blades of the propeller B_p, the propeller diameter D_p, the propeller pitch H_p and the propeller's weight.

If data about the propeller from experiments is available, then more accurate values of C_M and C_T can be obtained. eq. (4.1) and eq. (4.2) can be used to estimate them, by re-arranging the equations to make C_M and C_T explicit and then take the mean of the result. It is shown in eq. (4.9) and eq. (4.10). Note that n is the number of readings used.

$$C_T = \frac{\sum_n \frac{T}{\rho\left(\frac{N}{60}\right)^2 D_p^4}}{n} \tag{4.9}$$

$$C_M = \frac{\sum_n \frac{M}{\rho\left(\frac{N}{60}\right)^2 D_p^5}}{n} \tag{4.10}$$

4.4 Motor Characteristics

Author: Arham

To run the propeller, the motor also needs to be sized to find its required power. This can be split into the motor voltage and current, both of which can be estimated using the following formulae [31]:

$$U_m = \left(\frac{M K_{\mathrm{V0}} U_{\mathrm{m0}}}{9.55\left(U_{\mathrm{m0}} - I_{\mathrm{m0}} R_{\mathrm{m}}\right)} + I_{\mathrm{m0}}\right) R_{\mathrm{m}} + \frac{U_{\mathrm{m0}} - I_{\mathrm{m0}} R_{\mathrm{m}}}{K_{\mathrm{V0}} U_{\mathrm{m0}}} N \tag{4.11}$$

$$I_m = \frac{M K_{\mathrm{V0}} U_{\mathrm{m0}}}{9.55\left(U_{\mathrm{m0}} - I_{\mathrm{m0}} R_{\mathrm{m}}\right)} + I_{\mathrm{m0}} \tag{4.12}$$

where K_{V0} is the KV value of the motor, U_{m0} is the no load voltage, I_{m0} is the no load current and R_m is the Motor resistance.

The power and efficiency can then be calculated as described in eq. (4.13) and eq. (4.14):

$$P_m = U_m \cdot I_m \tag{4.13}$$

$$\eta = \frac{N\frac{2\pi}{60}M}{P_m} \tag{4.14}$$

$$P_{tot} = P_m \cdot N_m \cdot 1.2 + P_{pay} \tag{4.15}$$

where P_m is the power per motor, N_m is the number of motors and 1.2 is the factor implemented to take into account that the aircraft is not consistently hovering and sometimes needs more power, P_{pay} is the power required by all other electrical components onboard the drone.

The required Energy capacity can then be calculated by multiplying the power by the required flight time as shown in eq. (4.16) and then the energy density of hydrogen is used to calculate the mass that is needed for the required energy capacity in eq. (4.17). Then, the amount of hydrogen can be calculated with eq. (4.17), using hydrogen's energy density (34000 Wh/kg). The hydrogen tank also needs to be large enough to carry the total amount of hydrogen and just calculating the weight of material required for the specified volume isn't sufficient, since the tank has a lot of appendages. Instead, the approximate mass of hydrogen needed was found, around 100 grams, and a corresponding off-the-shelf (COTS) hydrogen tank was chosen. The H3 Dynamics A-Series 5L tank was chosen as it allows for up to 120 grams of hydrogen storage.

$$E_{req} = P_{tot} \cdot T \tag{4.16}$$

$$m_{H_2} = \frac{E_{req}}{U_{H_2}} \tag{4.17}$$

The iteration process can now be performed by updating the required hydrogen mass instead of battery mass and iteratively updating until convergence. For the required hydrogen mass, an appropriate off-the-shelf hydrogen tank will be selected.

4.5 Configuration Selection

Author: Arham

To determine the optimal combination of motor and propeller, each motor is paired with each propeller, and the process described in Section 4.3 and Section 4.4 is utilized to calculate the total mass of each combination that fulfils the mission goals. The current ratio (I_{ratio}) is then computed, representing the total current drawn by the motor at a thrust-to-weight ratio of 2 divided by the maximum allowable current. This value should not exceed 1. The feasible options are presented in Table 4.3.

Table 4.3: Top Configurations and their respective properties

i	**Propeller**	**Motor**	$\mathbf{m_{H_2}}$ [kg]	**Mass** [kg]	**RPM** [-]	$\mathbf{I_{ratio}}$ [-]
1	T-Motor NS 26x85	T-Motor Antigravity MN6007II KV160	0.108	11.85	1840.63	0.73
2	T-Motor NS 26x85	T-Motor Antigravity MN6007II KV320	0.109	11.86	1841.95	0.79
3	T-Motor NS 26x85	T-Motor Antigravity MN7005 KV115	0.110	12.08	1858.69	0.85
4	T-Motor NS 26x85	T-Motor Antigravity MN7005 KV230	0.111	12.08	1858.74	0.91
5	T-Motor NS 26x85	T-Motor Antigravity MN6007 KV320	0.115	12.02	1854.11	0.91
6	T-Motor NS 24x72	T-Motor Antigravity MN5008 KV340	0.118	11.58	2347.64	0.91
7	T-Motor NS 24x72	T-Motor Antigravity MN6007II KV160	0.118	11.77	2367.05	0.65
8	T-Motor NS 24x72	T-Motor Antigravity MN6007II KV320	0.119	11.78	2368.74	0.69

Table 4.3 presents two feasible propeller options: the T-Motor NS 26x85 and the T-Motor NS 24x72. Choosing the T-Motor NS 26x85 results in a slightly heavier drone, as the NS 26x85 is 80 grams heavier when paired with the T-Motor Antigravity MN6007II KV160 compared to the NS 24x72. Additionally, the NS 26x85 has a lower headroom with a current ratio of 0.73 compared to 0.65 for the NS 24x72. However, both propellers operate comfortably within their current limits, and the small difference in headroom does not have a significant impact.

One major advantage of the NS 26x85 is its higher efficiency, leading to lower fuel consumption. With a 3-hour endurance, the NS 26x85 consumes 108g of fuel, while the NS 24x72 consumes 118g when paired with the same motor. When fully loaded (120g), the NS 26x85 will last for approximately 3.32 hours, while the NS 24x72 will last around 3.05 hours. The difference in endurance compensates for the slightly increased weight of the drone, making the NS 26x85 the preferred choice. The T-Motor Antigravity MN6007II KV160 was selected as the optimal motor to pair with the NS 26x85 due to its lower overall mass and lower fuel consumption.

Performance Evaluation

Now that the configuration has been decided, the performance of the chosen engine can be further expanded upon with the following plots.

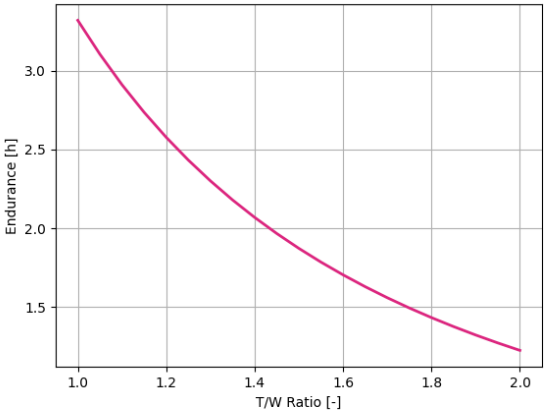


Figure 4.1: Endurance as a function of average thrust to weight ratio

Figure 4.1 illustrates how the endurance of the drone decreases when the average required thrust-to-weight ratio exceeds 1 (hovering). This is particularly relevant on windy and turbulent days when the drone needs to exert additional thrust to counteract aerodynamic forces. It is also relevant for cruise, as the thrust-to-weight ratio needs to be higher than 1 to have a horizontal component. Even at maximum thrust (2 times the weight), the drone still maintains an endurance of more than 1 hour, which is sufficient for it to abort the mission and return to the base. It is important to note that the correction factor of 1.2 applied in eq. (4.15) is applied separately and accounts for the fact that the drone is manoeuvring during the inspection and the system is not 100% efficient.

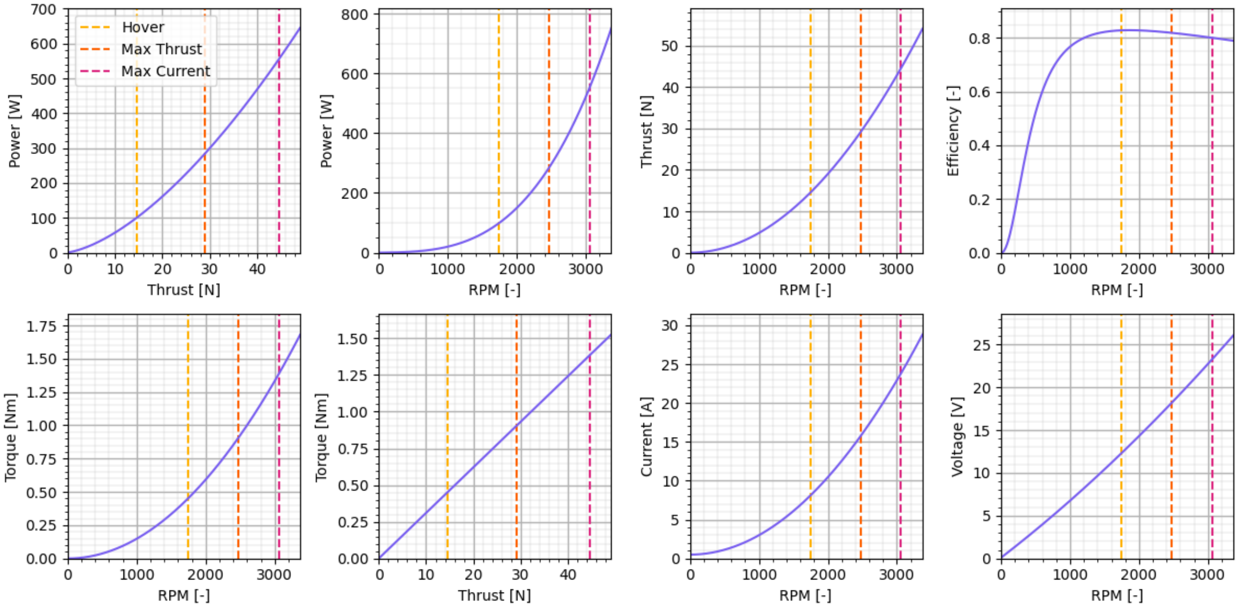


Figure 4.2: Performance Plots for the chosen Engine configuration

Figure 4.2 displays the performance characteristics of the engine, with vertical line indicators marking the hovering state (thrust-to-weight ratio of 1), the maximum needed thrust (thrust-to-weight ratio of 2), and the maximum thrust that the motor can deliver, limited by the maximum current. Several key observations can be made from these curves. First, the power required for double the thrust (from hovering to maximum needed thrust) is nearly three times higher, indicating a steep relationship between power and thrust. Similarly, the RPM of the engine also scales sharply with thrust, particularly with increasing power. However, the efficiency curve reveals that the hovering state is close to optimal, with an efficiency of over 80%. The maximum thrust and maximum current values are not significantly lower, making the engine well-suited for drone operations. Furthermore, the torque is plotted to examine its relationship with RPM, thrust, and power. The quadratic variation of torque with RPM aligns with eq. (4.2), while the linear relationship with thrust corresponds to eq. (4.1), with the gradient of the torque-thrust graph being $\frac{C_M}{C_T}$. The voltage-RPM graph, which appears almost linear, suggests that the magnitude of the N term in eq. (4.11) is significantly larger than $I_m R_m$, highlighting its dominant influence.

4.6 ESC and Hydrogen Fuel Cell

Author: Enes

An ESC in a drone regulates the voltage supplied to the motors, controlling the power output of the motors. The input voltage to the ESC is provided by the power generation subsystem, which is the hydrogen fuel cell. Since the ESC and fuel cell are heavily interconnected, it is crucial to model the two together and ensure compatibility in terms of voltage and current draws, which is necessary for the drone to meet its performance requirements.

4.6.1 Hydrogen Fuel Cell Selection

The fuel cell chosen for the drone is H3 dynamic's A-2000 advanced lightweight fuel cell system which can provide a continuous power of 2200 watts. On the market, very limited fuel cells are available suitable for a drone. Moreover, a requirement was that the chosen fuel cell could provide sufficient power for a thrust-to-weight ratio of 2, for which a power supply of 2200 watts was deemed necessary. The voltage of this fuel cell was modelled as two piece-wise linear functions, made to resemble the official datasheet, which can be seen in Figure 4.3.

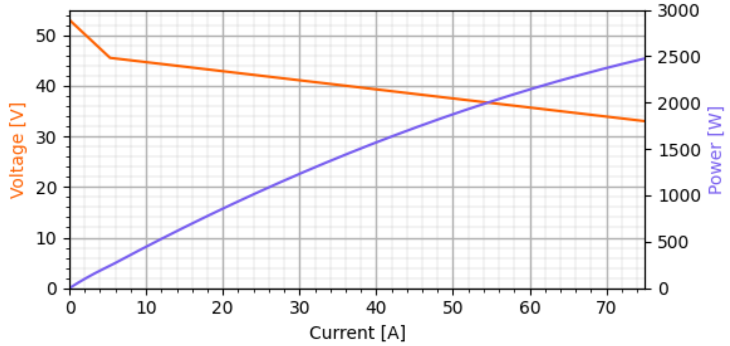


Figure 4.3: The output voltage and power of the Fuel Cell for current drawn

4.6.2 Electronic Speed Controller

The most important properties of the ESC are $I_{\mathrm{e\,max}}$ and $I_{\mathrm{e\,cont}}$ which are the maximum current and continuous current that can be drawn by the ESC, respectively. ESC manufacturers usually provide with these two values, $I_{\mathrm{e\,max}}$ and $I_{\mathrm{e\,cont}}$, which describe the current limits and ensure safe operation of the electronics. The R_{e}, internal resistance of an ESC, is another important property but it cannot be found easily in datasheets of off-the-shelf ESCs. However, there is an empirical relation between R_{e} and $I_{\mathrm{e\,max}}$ [32] as shown below in eq. (4.18).

$$R_{\mathrm{e}} = 32.6754 \cdot (I_{\mathrm{e\,max}})^{-0.7669} + 0.5269 \tag{4.18}$$

Based on the above it is possible to compute the current drawn and the voltage of the ESC using eq. (4.19) and eq. (4.20) [31], where the subscript m, e, b denote the motor, ESC, battery, respectively.

$$I_e = \frac{(U_{\mathrm{m}} + I_{\mathrm{m}} R_{\mathrm{e}}) \cdot I_{\mathrm{m}}}{U_{\mathrm{b}}} \tag{4.19}$$

$$U_e = U_{\mathrm{b}} - I_{\mathrm{b}} R_{\mathrm{b}} \tag{4.20}$$

With the hydrogen fuel cell selected and formulas for the properties of the ESC established performance plots of the ESC and Fuel Cell could be generated as seen in Figure 4.4. It is crucial to visually verify that the power required for hovering and maximum thrust is provided by the fuel cell through the ESCs.

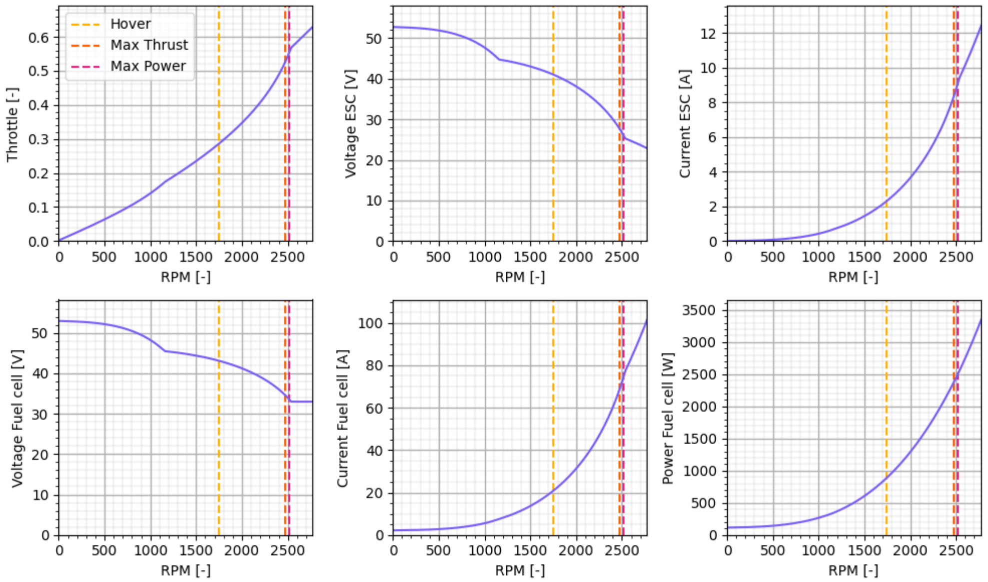


Figure 4.4: ESC and Fuel Cell Performance Plots

The performance plots have two kinks, one corresponding to the kink in the voltage-current graph and the other kink corresponding to the maximum power that can be sustained by the fuel cell. It is important to note that the power can go higher than the maximum for short bursts when combined with the LiPo battery (up to 8000W). The current for the ESC is also often a limiting factor, but in this case, the maximum allowable current in ESC is much higher than what the current provided by the power subsystem is, hence why it is not an issue.

4.7 Range Calculation

Author: Arham

Calculating the range of the drone is essential for its operation. Although it spends most of its mission profile in a near hover state, the number of inspections made per trip is largely influenced by how efficiently the drone can travel from one turbine to another. The range can be calculated using the airspeed (V) and the endurance(E) as shown in eq. (4.21). To calculate the airspeed the drone needs to tilt to have a component of thrust acting forward, however, that means that to still have sufficient thrust to counteract the weight, it requires a higher thrust-to-weight ratio and hence more power. This is shown in Figure 4.5a. To find the required thrust-to-weight ratio, the tilt/pitch angle (θ) can be used as shown in eq. (4.22).

$$R = V \cdot E \tag{4.21}$$

$$T/W = \frac{1}{\cos\theta} \tag{4.22}$$

The endurance for the thrust-to-weight ratio can be found in Figure 4.1 and V is the terminal velocity that can be found by equating thrust and drag. The horizontal component of thrust is shown in eq. (4.23), the drag is shown in eq. (4.24) and the resultant equation is eq. (4.25).

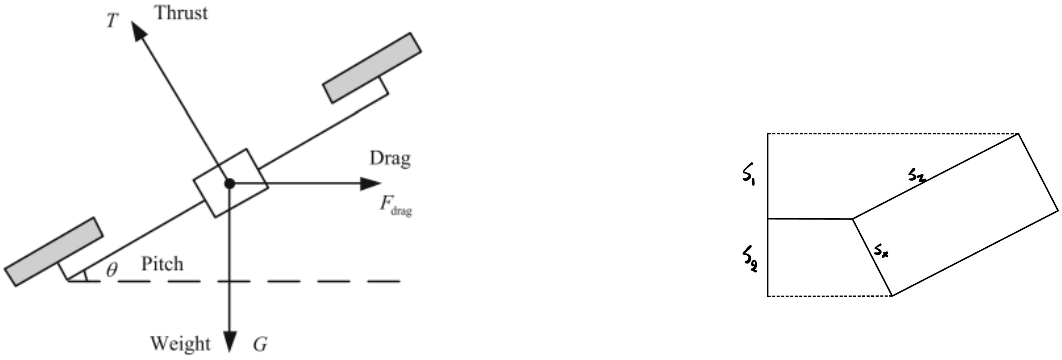
$$T \cdot \sin\theta = mg\tan\theta \tag{4.23}$$

$$D = C_D \frac{1}{2} \rho V^2 S \tag{4.24}$$

$$V = \sqrt{\frac{2mg\tan\theta}{C_D \rho S}} \tag{4.25}$$

m is the mass of the drone and g is the gravitational acceleration, C_D is the drag coefficient which is estimated based on the shape of the drone, S is the surface area of the drone faced by the wind (a function of pitch angle).

To calculate how surface area changes with θ, both the front view area S_x and top view area S_z are considered and their perpendicular areas are added up as shown in Figure 4.5b.

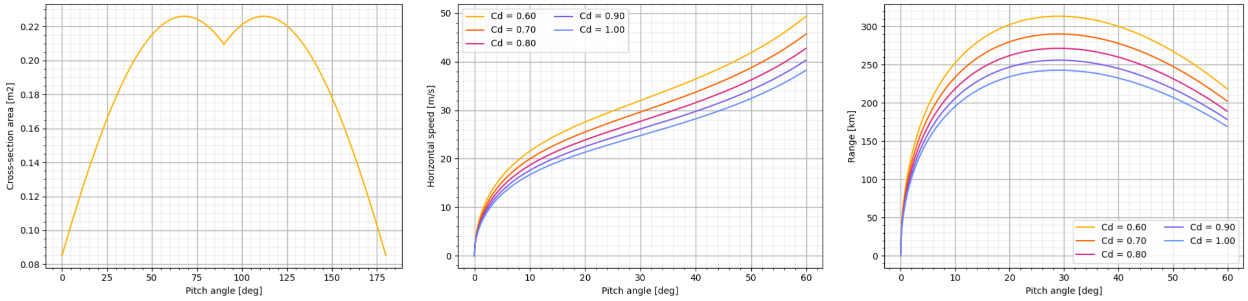


(a) Forces acting on the drone

(b) Surface area facing the airspeed

Figure 4.5: Modelling of the drone

The resulting surface area, airspeed and range for varying C_D valuesare shown in Figure 4.6a, Figure 4.6b and Figure 4.6c.



(a) Surface area as a function of θ and C_D (b) Speed as a function of θ and C_D (c) Speed as a function of θ and C_D

Figure 4.6: Results of the range calculation

4.8 Back-up battery

Author: Enes

A back-up battery is essential for a multiple of reasons. First of all in case of failure of the hydrogen system the battery should ensure the drone can still land safely. The second reason is more technical. Fuel cells suffer from having a transient response to an increase in power demand. This means that for example for the the H3 Dynamic's A-2000 fuel cell, the no-load to full-load power generation takes around 20 seconds. If a sudden ramp up of the power supply is necessary, for example to avoid a collision, a faster response time is needed. Batteries have a much faster response time compared to fuel cells [33].

The selection of the battery is based on a few aspects. First of all should the battery safely support high discharge. When the battery is activated a higher power usage is warranted from the propulsion system. This is around 50A, 37V and around 1850 Watts of power drawn. Moreover, the battery should be lightweight and have sufficient endurance. The battery endurance is calculated using eq. (4.26) as given by [31]. T_B is the endurance in minutes, C_b capacity of battery in mAh and C_{min} the minimum capacity of the battery. C_{min} is assumed to be 15%, corresponding to an average battery.

$$T_{\text{b}} = \frac{C_{\text{b}} - C_{\text{min}}}{I_{\text{b}}} \cdot \frac{60}{1000} \tag{4.26}$$

Three promising batteries were found from Voltaplex, which develop high discharge batteries for aerial applications. The properties of the batteries and the resulting calculations can be found in Table 4.4. The maximum discharge in batteries is given as a C-rate which is a unitless coefficient that is defined as the ratio of maximum discharge over battery capacity.

It can be seen that the only battery meeting the C-rate requirements for peak thrust is the 1800mAh, 63A battery while also providing the best endurance at 108 seconds at peak thrust. The other two options, although lighter, would limit the maximum discharge rate making it unsafe to draw enough current for peak thrust.

Table 4.4: List of potential back-up batteries and their properties

Battery	**Layout**	**Capacity** [mAh]	**Current** [A]	**Voltage** [V]	**C-rate** [-]	**Weight** [kg]	**Endurance** [s] **(Peak/Hover)**	**Required C-rate**[-]
Voltaplex LiPo	10S1P	1400	49	37	35	0.36	84 | 267.75	42.86
		1600	56			0.44	96 | 306	37.50
		1800	63			0.47	108 | 344.25	33.33

4.9 Sensitivity Analysis

Author: Arham

As part of the sensitivity analysis, all the mission profile parameters as well as the design parameters are changed slightly and the mass response of the drone is checked to see if any particular parameter can affect the drone weight significantly with a small variation. The reason the mass changes is due to the reduced/increased performance of the drone due to the parameter change. Doing so moreover supports verification of the model, as the mass variation can be checked for sensibility depending on a change in the input.

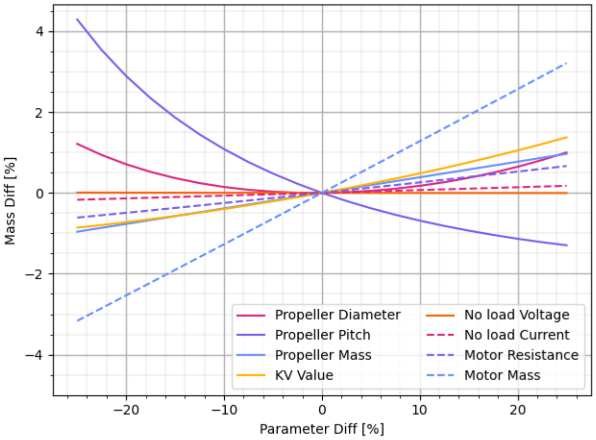


Figure 4.7: Sensitivity Analysis

The two most sensitive parameters in terms of their impact on the overall drone mass are the propeller pitch and motor mass. A 25% variation in these parameters can lead to up to a 5% increase in mass. However, it is important to note that both the propellers and motors are manufactured products with tight tolerances. Therefore, significant variations in these parameters are unlikely, and their impact on performance is expected to be minimal. On the other hand, the remaining parameters, including the motor parameters, have a negligible effect on the overall drone mass. A 25% variation in these parameters would only result in a 1% change in the total mass.

4.10 Verification

Author: Enes

The verification of the model is split up in two parts: unit tests and systems tests. These are discussed in subsection 4.10.1 and subsection 4.10.2 respectively. Whereas unit tests test separate function systems with the aim to isolate separate malfunctioning parts of the code system tests test the whole model. The verification data is obtained from hand-calculations.

4.10.1 Unit Tests

Unit tests are an essential tool for verifying the functionality of individual functions that form a part of a larger software model. Moreover, unit tests check that a single component operates correctly, making it easier to isolate and fix any issues in the application. Therefore, by ensuring that all individual functions work correctly, it can be confirmed that the complete module operates as intended. To conduct effective unit testing, analytical solutions and hand calculations must be provided, and their outputs compared against the respective code output to ensure the correct functioning of the code. The built-in Python package, *unittest*, provides a convenient way to perform these tests. For all of the unit tests, the tolerance used for the error is machine epsilon. A set of unit tests were performed alongside greater module tests. The list of unit tests are presented in Table 4.5. They consist of an ID, the test name and rationale.

The unit tests were implemented for the following python classes:

- `Propeller`: Models the propeller and is used to compute the thrust and torque generated for a given RPM. Tests given under UNIT_PROP_XX.

- `Motor`: Models the model and is used to compute the required voltage and current for a given thrust and torque. Tests given under UNIT_MOTOR_XX.
- `ESC`: Models the ESC and is used to compute the required voltage and current for a drone voltage and current, as well as to compute the throttle command given to the ESC. Tests given under UNIT_ESC_XX.
- `HydrogenTank`: Models the hydrogen tank and is used to compute the resulting tank mass for a given hydrogen fuel amount. Tests given under UNIT_TANK_XX.
- `FuelCell`: Models the hydrogen fuel cell and is used to get the voltage and power output for the current drawn. Tests given under UNIT_FUELCELL_XX.

Table 4.5: List of implemented unit tests for the performance tool.

ID	**Name Test**	**Explanation**	**Passed**
UNIT_PROP_01	Initializer test	Check if the initialization of the Propeller class was done correctly	✓
UNIT_PROP_02	Coefficient test	Check if the coefficients C_T and C_M match hand calculations.	✓
UNIT_PROP_03	Forces test	Check if the forces for a given RPM match hand calculations.	✓
UNIT_PROP_04	Required RPM test	Check if the required RPM for a certain thrust value match hand calculations.	✓
UNIT_MOTOR_01	Initializer test	Check if the initialization of the Motor class was done correctly.	✓
UNIT_MOTOR_02	Representation test	Check if the representation (__repr__) of motor class is correct.	✓
UNIT_MOTOR_03	Voltage & current test	Check if the current and voltage for a given RPM and Torque match hand calculations.	✓
UNIT_ESC_01	Initializer test	Check if the intialization of the ESC class was done correctly.	✓
UNIT_ESC_02	ESC resistance test	Check if the resistance calculation of ESC matches the results from the empirical formula, as well as verify that an ESC with higher maximum current has a lower resistance.	✓
UNIT_ESC_03	Throttle test	Check that the throttle command matches the results from hand calculations and that for a higher motor current and voltage, corresponding throttle command is higher.	✓
UNIT_TANK_01	Initializer test	Check if the initialization of the Hydrogen tank class was done correctly.	✓
UNIT_TANK_02	Tank & hydrogen mass test	Check if the mass of the tank match empirical formula.	✓
UNIT_FUELCELL_01	Initializer test	Check if the initialization of the Fuel Cell was done correctly.	✓
UNIT_FUELCELL_02	Voltage test	Check that the voltage for a given current is positive.	✓
UNIT_FUELCELL_03	Voltage limit test	Check that for a current over the limit, an error is raised.	✓
UNIT_FUELCELL_04	Power test	Check that the voltage and current computed for a given power match the power itself and if the current is below the maximum.	✓

4.10.2 System Tests

After the unit tests, system tests were performed on the overall performance tool, as seen in Table 4.6. System tests tests a much larger block of code compared to unit tests. The tests have been implemented for the following python classes:

- `Drone`: Generates drone from a given configuration file specifying the properties of the various components of the drone. Tests given under SYS_DRONE_XX.

- `ShelfDrone`: Generates a drone from off-the-shelft components. Tests given under SYS_SHELF_DRONE_XX.
- `DroneCombinator`: Generates all possible drone combinations from the list of components available, and sorts them by the final mass (if converged). Used to find the ideal drone combination. Tests given under SYS_COMBINATOR_XX.
- `SensitivityAnalysis`: Performs sensitivity analysis by creating drones from a given configuration and varying every parameters to see the effet on final mass. Tests given under SYS_SENSITIVITY_XX.

Table 4.6: List of implemented system tests for the performance tool.

ID	Name Test	Explanation	Passed
SYS_DRONE_01	Initializer test	Check if the initialization of the Drone class was done correctly.	✓
SYS_DRONE_02	Mass test	Check if the mass calculation of a drone configuration matches hand calculations.	✓
SYS_DRONE_03	Current test	Check if the current required for the propellers for a given torque and thrust matches the current drawn by the motor.	✓
SYS_DRONE_04	Endurance test	Check if the endurance computed matches the power draw from the power source	✓
SYS_SHELF_DRONE_01	Propeller test	Check if the off-the-shelf propeller matches the real properties of the propeller	✓
SYS_SHELF_DRONE_02	Motor test	Check if the off-the-shelf motor matches the real properties of the motor.	✓
SYS_SHELF_DRONE_03	ESC test	Check if the off-the-shelf ESC matches the real properties of the ESC	✓
SYS_SHELF_DRONE_04	Coefficient test	Check if the computed coefficients match calculations from the performance datasheet	✓
SYS_COMBINATOR_01	Initializer test	Check if the initialization of the Drone Combinator class was done correctly	✓
SYS_COMBINATOR_02	Create drones test	Check if the properties of the drones generated are sensible.	✓
SYS_COMBINATOR_03	Sort test	Check if the sorted drone list is sorted correctly.	✓
SYS_SENSITIVITY_01	Initializer test	Check if the initialization of the Sensitivity Analysis class was done correctly	✓
SYS_SENSITIVITY_02	Generate drones test	Check if the correct parameters are changed in the drones.	✓
SYS_SENSITIVITY_03	Drone parameter test	Check if the drones have different configurations, as in they are not the same.	✓

4.11 Validation

Author: Arham

Having passed all verification tests paved the way to start validation, in which the model is compared to experimental results obtained from Tytorobotics [29]. It involved the chosen best combination of propeller and motor, the T-Motor NS 26x85 and T-Motor Antigravity MN6007II KV160 respectively. The parameters compared were thrust, torque and power as well as efficiency. Current and voltage were not compared as the data provided battery current and voltage compared to motor current and voltage supplied by the model. The resulting graphs are shown in Figure 4.8.

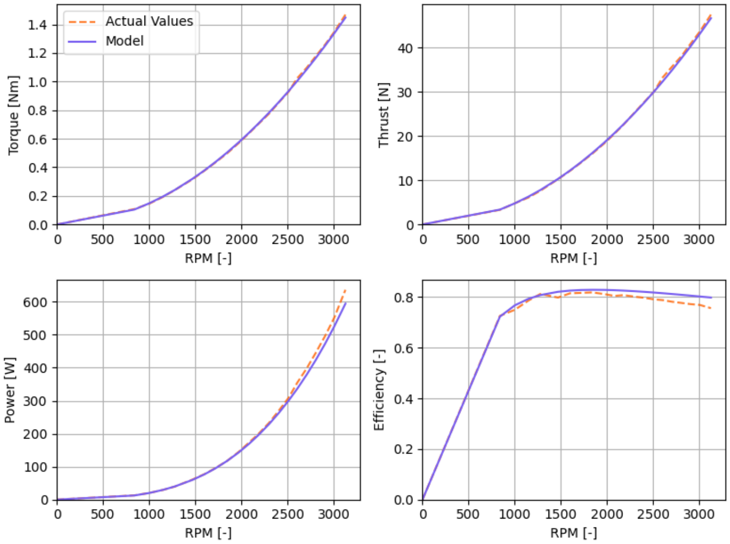


Figure 4.8: Comparison between experimental and model results for validation

The validation yielded very accurate results with the thrust and torque particularly following the experimental values almost exactly, the power and efficiency curves are also very accurate until towards the very end where the model starts to slightly underestimate the required power and overestimate the efficiency. The average Root Mean Squared Error (RMSE) values are provided in Table 4.7, alongside the mean value of the parameter which is used to normalize the RMSE.

Table 4.7: The RMSE and normalized RMSE of the four relevant parameters

Parameter	**RMSE**	**Mean value**	**Normalized RMSE**
Thrust	0.3859	21.623	0.01785
Torque	0.00827	0.672	0.0123
Power	14.421	213.305	0.0676
Efficiency	0.02266	0.7706	0.0294

5: Structures Subsystem Design

The chassis is the foundation of any drone, providing a strong framework for all other subsystems. As such, it is important that the chassis follows the requirements listed in Table 3.10. A structural drone anatomy is explained in section 5.1, in which the key components are decribed in detail. Moving on, the material choices are justified in section 5.2. A background on the development from the initial sketch can be found in section 5.3. On top of that for structural analysis, Finite Element model is set-up and simulated in section 5.4. Finally section 5.5 provides a conclusion with recommendations to further improve the chassis. Moreover, this section provides an outlook for the tests that will need to be conducted in the future in order to proof the design meets requirements.

5.1 Drone Anatomy

Author: Nachiket, Louis

The drone chassis is designed to accomodate the coaxial octocopter design [34], that demands two motors connected per propeller arm, attached to a central structure. A detailed CAD view of the chassis is shown below in Figure 5.1.

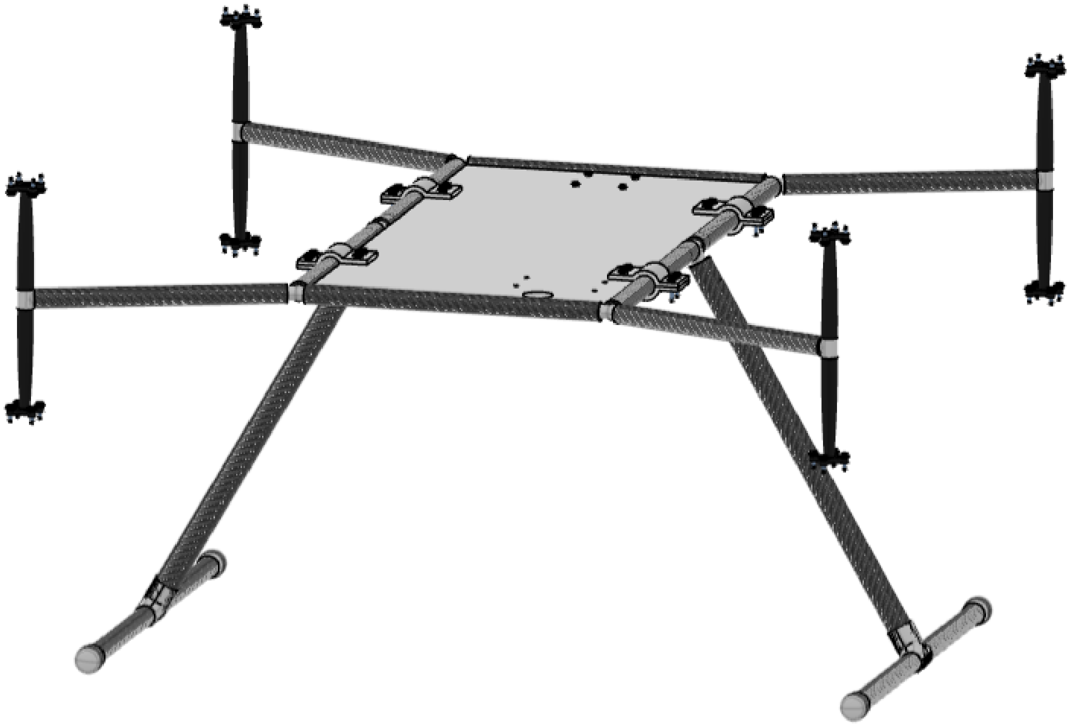


Figure 5.1: Overview of Complete Structure

As the first version of the drone is a prototype with low production rate it is important for the chassis to be easy to manufacture and assemble. Moreover, is it required that all components are accessible and can easily be replaced. The main structure has therefore been chosen to consist of hollow rods connected with attachments. A base plate has been added in the middle of the structure to allow for easy mounting of the other subsystems and their respective components. This includes components such as the fuel cell, hydrogen tank, battery, control box and inspection subsystem. Regarding the landing gear a belly positioned skid landing-gear has been chosen as of accessibility and stability reasons. The key components of the drone chassis are described as follows.

5.1.1 Base Plate

The central base plate governs the structural interfacing of the drone. It serves as a mounting platform for both payload and flight critical modules. For example, the camera is secured to the bottom face of the plate. Doing so allows the camera to gimbal in its full range of motion when inspecting the wind turbine at a downward facing angle. The other half of the bottom face plate comprises of bolt holes for fuel cell, battery and electronics compartment. Dimensions of the plate are found in Table A.1.

A high stiffness compliance in the base plate is achieved by means of a sandwich structure. While sandwich panels culminate to stiff structures, they are susceptible to failure in face wrinkling, and threaded insert pull-out [35]. The effect of face wrinkling is reduced by double clamp attachment of the hydrogen tank on the top face sheet. The double clamp attachment acts as a plate stiffener. To reduce the risk of insert pull-out, the first design iteration is over-designed. Specifically, the base plate has a large number of mounting holes to enhance distribution of the weight of different components.

Finally, material details of the honeycomb-metal panel are provided in section 5.2; corresponding thicknesses are tabulated in Table A.1.

5.1.2 Chassis Rods

Although the base plate serves as a mounting platform for bulk components, it cannot directly interface with rods leading to the landing gear or to the propellers. In order not to induce load paths created by the landing gear and propeller into the baseplate it was decide to integrate hollow circular rods on the perimeter of the baseplate. These loads can be induced in the event of an impact landing or a one-engine-inoperative situation.

5.1.3 Cross Rods

Cross rods connect the chassis to the propulsion mounting. These rods are angled by 21.5% with respect to the short side chassis rod. Reason for having the rods on an angle is to have the propellers in a square configuration, which heavily simplifies the control algorithm for the drone. The reader is kindly referred to Table A.1 for exact specifications of the cross rods.

5.1.4 Skid Landing Gear

The belly landing gear mandated a design change from four rods [34] to two skids. It was concluded that the original option would obstruct the view of the inspections systems cameras. Therefore, a skid landing gear was favoured as only two points of attachments are required. To prevent tip-over, the landing rod is angled at an angle of 30° to the vertical axis. The lengths of the landing rods are reported in Table A.1.

5.1.5 Attachments

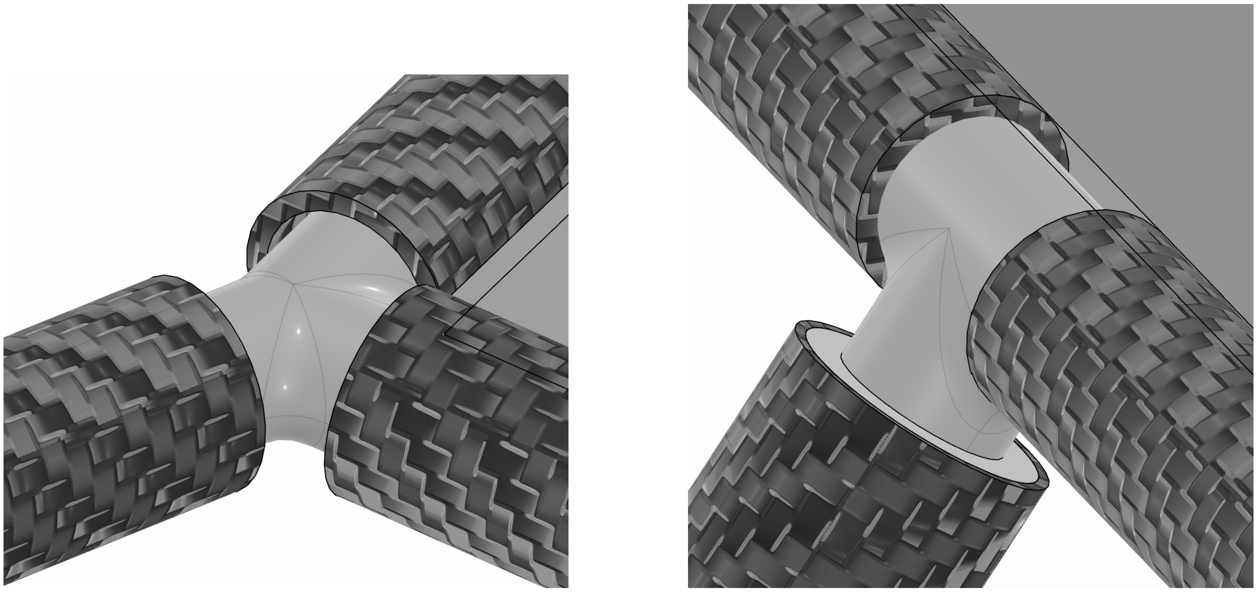
Thus far, it is clear that the structure of the drone fundamentally comprises of rods and plates. The design of their attachment mechanisms is paramount in maintaining overall structural integrity. Consequently, an array of connectors, brackets and clamps are custom designed.

3-point connectors

In-house developed 3-point (T) connectors are purposed to connect rods, namely the carbon tubes of the chassis/landing gear. Custom design of these brackets warrants precise control over the angle at which the connection is made. Such design freedom is particularly beneficial in designing the connector between chassis and cross tubes, as seen in Figure 5.2a. Recall that for control reasons, these cross rods need to be angled to maintain a square configuration relative to each other.

T-connectors tightly fit inside the hollow carbon fibre tubes. A tight or interference fit is especially useful for mounting the propellers as shown in Figure 5.2b. Such an interference fit increases the strength of the joint by providing some frictional resistance. The frictional resistance reduce the risk of the propeller rods simply

detaching during a peak thrust maneuvre.
Examples of these T-connectors can be seen in Figure 5.2.

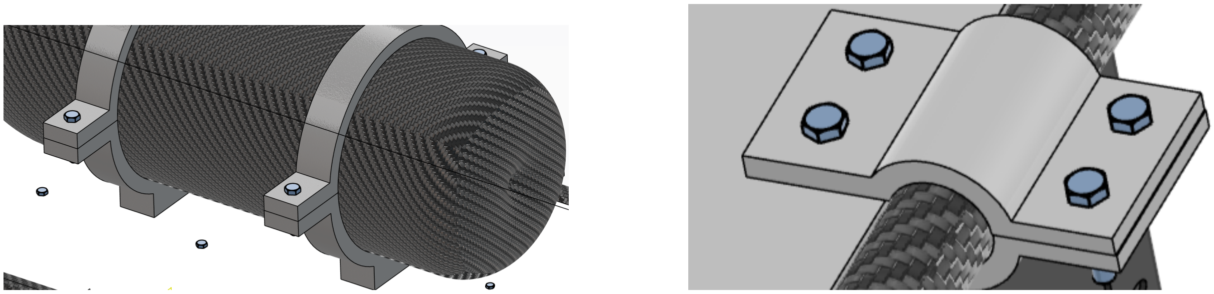


(a) Connection between chassis-cross tubes (b) Connection between chassis-landing gear

Figure 5.2: 3-point (T) connectors

Saddle brackets

Saddle brackets provide convenient solutions in clamping radial profiles using bolt and can easily be disassembled, thereby contributing to a modular structure especially for maintenance purposes. Tailored brackets are developed to connect the base plate with the chassis tubes, as seen in Figure 5.3b. Similar brackets in Figure 5.3a are utilised to mount the hydrogen tank to the base plate.



(a) Tank-base plate brackets (b) Base plate-chassis brackets

Figure 5.3: Saddle Brackets

Note that at least two pairs of saddle brackets should be used for any connections to eliminate undesired rotation about a single bolt.

Propulsion mounting

Regarding the propellers some challenges were induced by the fact that the propellers need to be half their diameter apart due to the wake introduced by the top propeller negatively effecting the aerodynamic performance of the other propeller. In order to do so a solid rod has been developed on which the propellers and engine can be mounted. The propulsion mounting is made from symmetric propeller rods and a 3-point connector. The parts shall be glued together as is not possible to print the complete part in one go. A render of the part can be found in Figure 5.4.

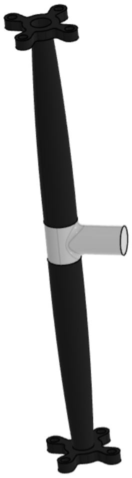


Figure 5.4: Propulsion mounting

It can be observed that the end of the propeller rods comprises of drilled holes for mounting on the motors. More details on the assembly are treated in Appendix A. Finally, note how the rods thicken at the root in order to maintain the same margin of safety for a higher internal moment.

All in all, the most noteworthy dimensions of the CAD model are tabulated in Table A.1.

Table 5.1: Top-level dimensions derived from CAD

Parameter	**Value**	**Unit**
Mass	1.52	kg
Base plate Length	600	mm
Base plate Width	300	mm
Propeller Pitch	818	mm
Propulsion rod length	230	mm
Ground Clearance	135	mm

5.2 Materials

Author: Nachiket, Louis

Careful selection of materials is driven by a range of factors such as high stiffness-to-weight ratio, ease of manufacturing, low cost, environmental sustainability and finally resistance to flammability. The material's fire resistance is applicable in case of a hydrogen fire. With the desired material properties in mind, a chassis material trade-off (similar to chassis layout) was conducted [34]. Carbon Fibre (CFRP) Epoxy Resin with Graphene Nano-platelets scored highest in the material trade-off and was therefore selected as the material of choice. Henceforth, this material shall be referred to as CFRP for brevity.

During the detailed design phase, adjustments were performed to the material selection. While CFRP is still the main chassis material- used in chassis rods, landing gear rods, and cross rods- other materials are deemed a better fit for certain custom parts. Additional materials include aluminum, Nomex and PETG.

Specifically, it has been decided to use PETG to make the connectors between the drone's hollow tubes. PETG excels in its ease of manufacturability, especially for complicated geometry. The connectors can easily be 3D printed; if production levels increase, the production process can be replaced with an injection moulding process. Moreover, PETG is lightweight and recyclable [36]. An extensive overview of all 3D printed brackets, clamps and connectors can be found in section A.1.

All in all, the base plate is to be made out of Aluminium Sandwich structure with adhesively bonded Nomex® honeycomb as the core material. It is preferred that aluminum face-plates are used instead of CFRP ones, namely for ease of drilling. Many mounting holes need to be drilled into the base plate to accommodate for components that are directly attached to it. Furthermore, the Nomex® honeycomb is especially suitable as the core material in providing a high resistance to bending, ie, a high $\frac{EI}{\rho}$. Nomex® has an additional benefit of being a flame resistant material[37].

5.3 Development from Initial sketch to CAD

Author: Nachiket

This section summarises the results of back-of-the-envelope structural sizing calculations that were used to generate a CAD model.

Firstly, a trade-off has been performed between several types for the structure of the drone in DSE Midterm Report [34]. Unfeasible options were quickly disregarded after which a trade-off and sensitivity analysis was performed on the remaining options. A coaxial octocopter came out to be the most suitable solution for the team's application after which a preliminary sizing tool in Excel® had been built with the aim to provide a starting point for the detailed design of the structure. An initial sketch and corresponding dimensions can be seen in Figure 5.5 and Table 5.2 respectively. The reader is referred to the DSE Midterm Report[34] for the methodology leading to the results presented in Table 5.2.

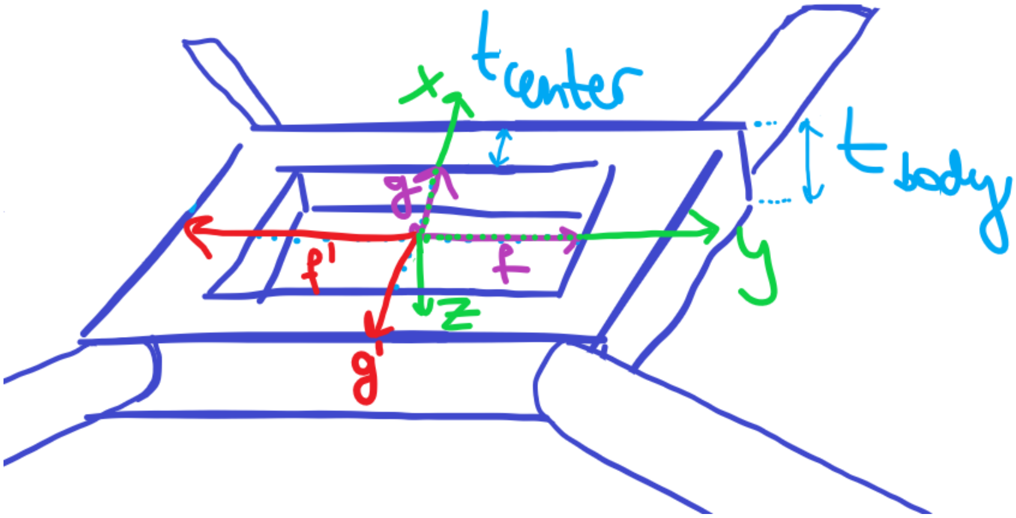


Figure 5.5: Hollow, rectangular drone chassis

Table 5.2: Preliminary structural sizing

Input	Value
Safety factor [-]	1.5
t_{center} [mm]	10
t_{body} [mm]	4
d [mm]	15
$r_{arm_{inner}}$ [mm]	10
t_{arm} [mm]	0.25
f [mm]	300
g [mm]	150
h [mm]	100

The dimensions provided in Table 5.2 lay the foundation of a CAD model adequately detailed to fabricate a first prototype. The complicated geometry of the drone advocates the structural analysis methodology to transition from closed-form analytical equations to computational mechanics, as described subsequently in section 5.4.

5.4 Structural Analysis: Finite Element Methods

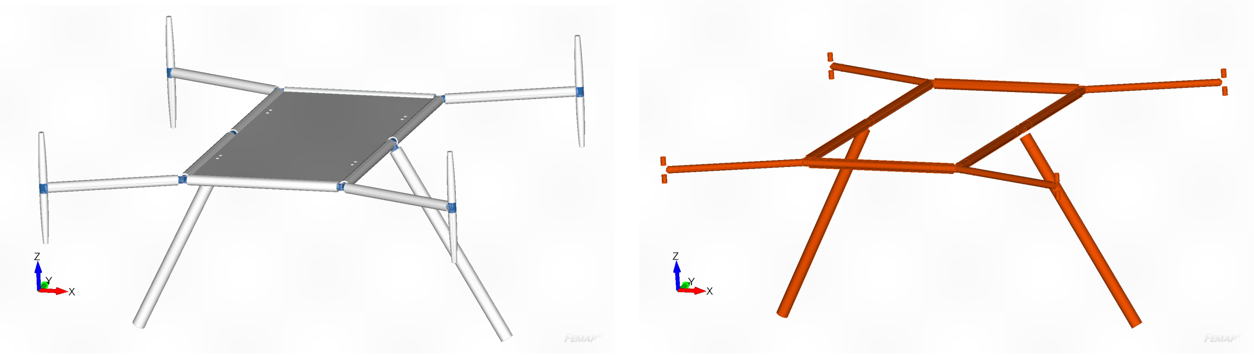
Author: Nachiket, Louis

A detailed CAD model, derived from preliminary chassis sizing, sanctioned high fidelity structural analyses. The objective of such analyses was to verify whether the drone adheres to a fail-safe, stiffness limited design philosophy. A secondary objective was to improve the structural design- possibly for reduced mass.

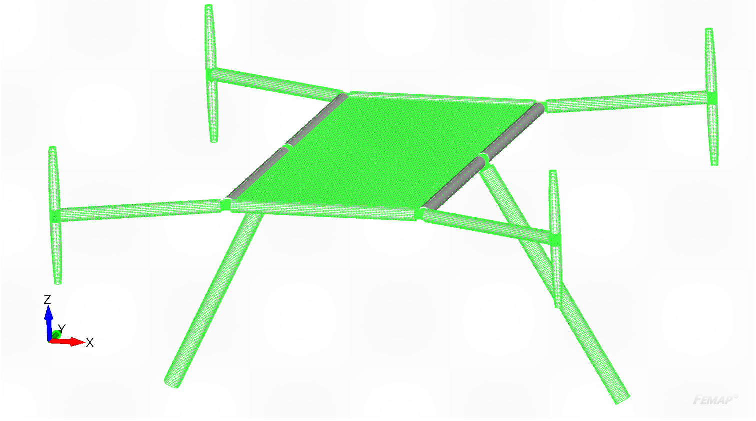
On top of that, composite structures made up of carbon-fibre lay-ups or honeycomb sandwiches invoke more complicated calculations than their isotropic counterparts. It is difficult, time-consuming to accurately implement such analytical models. Hence, an alternative analysis methodology, namely using commercial Finite Element Analysis (FEA) software was followed.

An FEA model has been created using Siemens Simcenter Femap®. The FEA model imports the CAD geometry as described in section 5.1. The FEA model set-up and pre-processing procedures are briefly outlined in subsection 5.4.1. Following the set-up stage, a wide range of static load cases, with relevant safety factors, are identified in subsection 5.4.2. Stiffness results from static analyses are discussed in subsection 5.4.2. In addition, modal analyses (see subsection 5.4.3) are carried out to identify the natural frequencies of the drone chassis. Identification of the drone's natural frequencies is critical to either avoid the external frequencies (eg: caused by rotor dynamics of propellers or wind gusts) that may lead to resonance, or even modify the design altogether. All FEA insights are mentioned in section 5.5.

5.4.1 Model Framework



(a) Surface Geometry (b) Glued Connections



(c) Meshed overlay

Figure 5.6: Simplified drone assembly used in FEM

CAD Surface Geometry

To begin with, a surface geometry of the drone chassis assembly was imported from CAD and appropriately simplified as shown in Figure 5.6a. Highly detailed part geometry (such as holes smaller than 3mm in diameter, sharp edges and fillets) was de-featured or simply removed. Namely, the bottom skid landing gear rods were omitted due to their overly intricate interface geometry.

Assembly Connections

It is important to note that the drone assembly is imported in multiple part that must be connected together for meshing. A linear glued connection property (weld type)[38] is set as the connection standard. All overlap region between two parts in geometric contact are assumed to be uniformly covered in glue. A visual of the glue surfaces is given in Figure 5.6b.

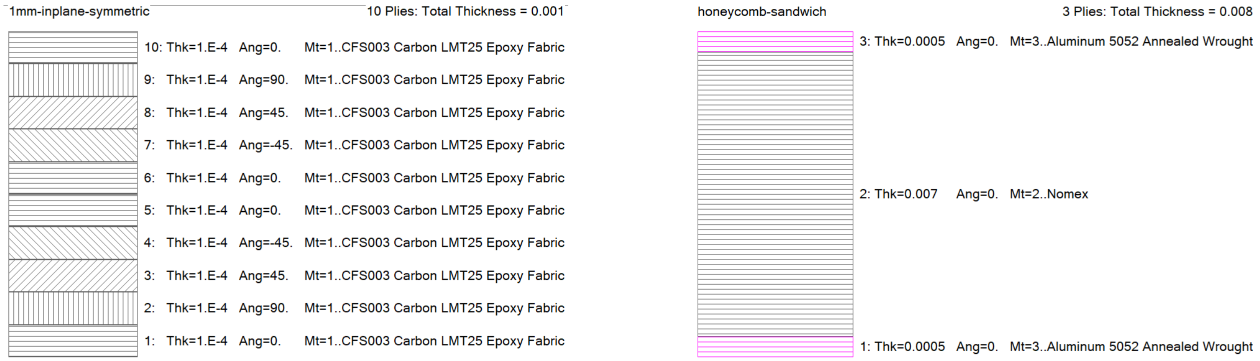
Materials and lay-up

Figure 5.6a shows that the drone chassis comprises of three major types of components: rectangular nomex/aluminum sandwich base plate, CFRP hollow cylindrical rods and PETG connectors. The material properties used in FEA are tabulated in Table 5.3.

Table 5.3: Materials used in FEA model

Material	**Type**	**Mass density** $[kg/m^3]$	$(\mathbf{E_x}, \mathbf{E_y})$ [GPa]	**G** [GPa]	ν [-]
CFS003 Carbon LMT25 Epoxy Fabric	2D Orthrotropic	1450	(53.6, 55.2)	2.85	0.042
Nomex	2D Orthrotropic	120	(3.95, 5.05)	1.6	0.2
Aluminum 5052 Annealed Wrought	Isotropic	2685	(70.3, 70.3)	Not used	0.33
Ultimaker PETG	Isotropic	1230	(1.94, 1.94)	Not used	0.35

Once materials are defined, it is possible to create lay-ups. The base plate has the stiffer orientation (E_x) of Nomex aligned with the longer side. The plate is then sandwiched (adhesively bonded) with equal thickness aluminum plates on top and bottom. Base plate lay-up is visualised in Figure 5.7b. Concerning the CFRP tubes, 10 orthrotropic thin plies are laid-up. As a first model, the lay-up is almost quasi-isotropic with the exception of 2 extra plies aligned along the axis of the lay-up. The symmetric lay-up $[45, 0, -45, 0, 90]_S$ degrees is depicted in Figure 5.7a. For composite cylindrical parts, a local cylindrical co-ordinate system is defined such that its z-axis aligns with the part's longitudinal axis. Consequently, surface laminates can be wrapped radially about the longitudinal axis of the cylinder.



(a) CFRP layup (1mm) for chassis rods (b) Nomex/aluminum sandwich (8mm) for base plate

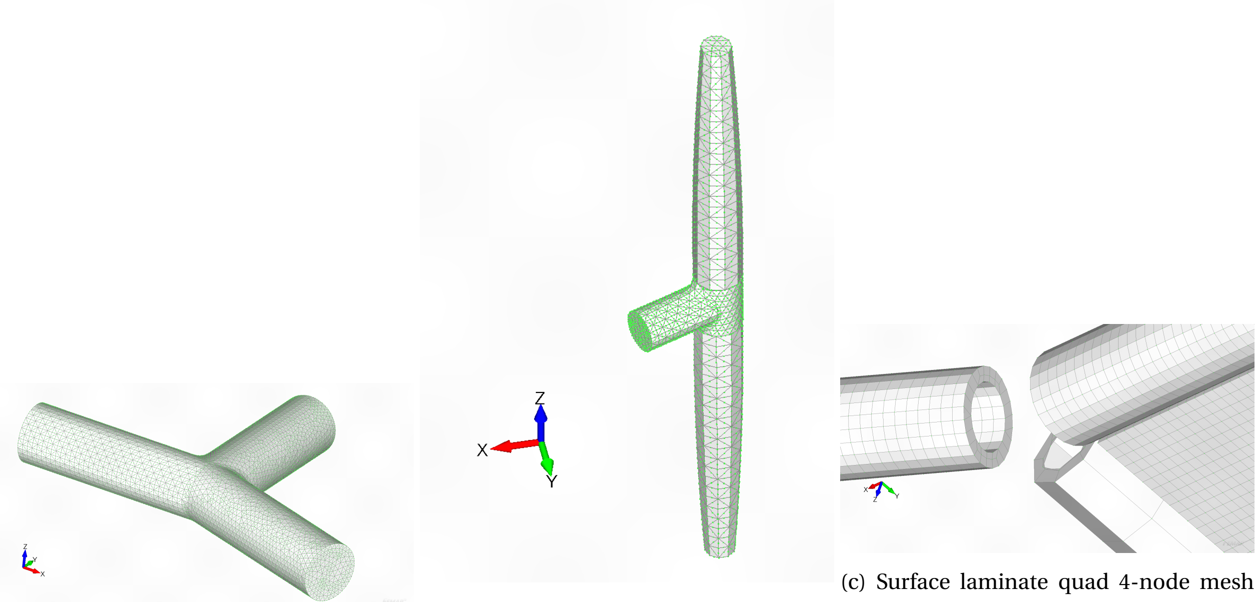
Figure 5.7: 2D Orthotropic lay-up

Observe that Figure 5.7a shows 10 plies of 0.1 mm thickness each, used to create a 1mm lay-up for a 1mm thick CFRP tube. Different thickness tubes demand a different total thickness for the lay-up. To achieve that, the ply thickness is varied while the number of plies and ply orientation is kept constant. For instance, a 3mm-layup for a 3mm thick tube has 10 plies of 0.3mm (instead of having 30 plies of 0.1mm thick).

Meshing

The drone model consists of 635,000 nodes, made up of 3D tetrahedal elements and 2D surface laminate elements. An image of the meshed model is provided in Figure 5.6c.

Tetrahedral elements were used on 'solid' geometry- comprising of all connectors and the propeller rods as shown in Figure 5.8a and Figure 5.8b respectively. Note that these solids are modelled to be isotropic. Alongside that, the 'surface' laminate quad meshes are shown in Figure 5.8c.



(a) Volumetric tetrahedral 10-node mesh, solid cross connector (size = 0.14)

(b) Volumetric tetrahedral 10-node mesh, solid propeller rods (size = 0.12) and connector (size = 0.18)

(c) Surface laminate quad 4-node mesh for orthrotropic lay-ups, hollow chassis tubes (size = 0.66-0.75) and base plate (size = 0.44)

Figure 5.8: Meshing

5.4.2 Linear Static Analyses

To obtain structural deformations of the drone assembly, linear static analyses were simulated for a range of load cases: static equilibrium on ground, peak thrust in air, hard landing and nominal hover. Deflection contours are presented in Figure 5.4.2, thus shedding qualitative insight into improving parts that contribute most to deflections. NX Nastran$^{®}$ [39] is used as the solver.

Load Cases

As multiple load cases are present during operational life of the drone, multiple load cases have to be analysed. An overview of boundary conditions, applied loads and constraints, can be found in Table 5.4 and Table 5.5 respectively. Load case A simulates static equilibrium on ground, involving the maximum take-off weight of the drone only as the applied force. For load case C the landing of the drone is simulated. Finally, load case B and D are similar for they model the drone in flight. In case B, maximum thrust that can be delivered by the propulsion system is added whereas in case D the drone's thrust is in static equilibrium with its maximum weight.

Table 5.4: Overview of Applied Loads

	Weight	Max. Thrust	Landing	Nom. Thrust
Load case A	🟩			
Load case B	🟩	🟩		
Load case C			🟩	
Load case D	🟩			🟩

To result in a statically determinate system, it is essential to introduce constraints for any applied loads. Special attention is drawn to load case B and D, where the drone is in flight. The FEA model assumes that during flight, the base of the landing gear remains fixed (as summarised in Table 5.5). By fixing the base of the landing gear, the majority of the deformations of the drone (such as bending of the base plate/propeller rods and axial elongation of the landing gear rods). By fixing any other geometry on the drone, such as the entire base plate, important stiffness information is lost- leading to an (incorrect) underestimation of the deformation contour.

Table 5.5: Overview of Fix Constraints

	Landing Gear Base	Centre of Rectangular Base Plate
Load case A	🟩	
Load case B	🟩	
Load case C		🟩
Load case D	🟩	

Weight Multiple payload/propulsion components are connected directly to the drone's chassis. As many of the components are commercial off-the shelf (COTS), it was not always possible to obtain a CAD model. Even in the case that a CAD model was obtained, it often contained overly detailed geometry that was unfeasible to mesh. It was further stipulated that COTS parts are already well analysed and should not be structurally altered. Therefore, these components were not included in the FEA model. Instead, component masses are modelled as (distributed) point loads. These point loads generally mimic the load path taken by bolts, from components to drone chassis. The location of the loads is taken from the assembly CAD model. Refer to Table 5.6 for the magnitude of the weight loads.

Further note that for static analyses, the weight of components has been modelled as applied point forces, rather than inertial matrices or point mass elements. In modal analysis (explained in subsection 5.4.3), it is explained how the weight forces are replaced with point masses.

Table 5.6: Component Weight Loads

Component	**Number of attachment points**	**Load per attachment point [N]**
Camera, gimbal, structured light	1	18.91
Fuel cell	6	4.91
Battery	4	1.52
Electronics Housing	4	0.49
Hydrogen Tank/Regulator Clamp 1	2	2.25
Hydrogen Tank/Regulator Clamp 2	2	7.93

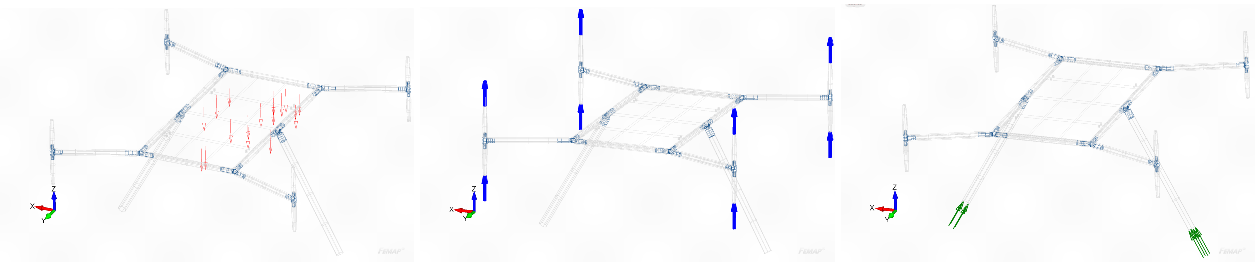
All weight loads, as tabulated above, are visualised in Figure 5.9a.

Thrust In practice, each motor is bolted into the end-plate of the propeller rod (shown in Figure 5.4). However, to reduce computational resources during meshing, the end-plate is omitted and thrust is applied directly on the top/bottom faces of the propeller rods- as depicted in Figure 5.9b. Maximum (as per motor specifications) and nominal (static equilibrium with 11.8kg) thrust magnitudes are given in Table 5.7. An extensive description of modelling assumptions on propulsive forces is treated by GNC in Table 4.1.

Table 5.7: Thrust Loads

	Load per upper motor [N]	**Load per lower motor [N]**
Maximum Thrust	32.87	21.06
Nominal Thrust	17.94	11.49

Landing In order to perform static analysis for a hard landing, it is said that drone free-falls under its own weight. This produces a total landing force of 116N, equally distributed across two landing rod. Refer to Figure 5.9c for visualisation.



(a) Weight (b) Thrust (c) Landing

Figure 5.9: Applied Forces

Results of Structural Deformations

Deformation contours are provided for each simulated load case. For select load cases, a safety factor (if used) is justified. It is possible to gather both maximum displacement values and qualitative insights on components that are most stressed-strained in each case. The insights from these results form the basis of future design iterations that are discussed in section 5.5. In some load cases, it is of interest to further include a Von Mises stress contour of structural parts.

Load case A: Static Equilibrium on Ground

Although the weight of non-structural components, as given in Table 5.6 does not exceed 75N, a safety factor of 1.5 on weight was utilised. While on ground, it is possible that the drone chassis may experience handling loads during operation or assembly. In Figure 5.10, the maximum sag is observed in the middle of the base plate- giving rise to a maximum downward displacement of 0.0028m (or 2.8mm). Figure 5.10

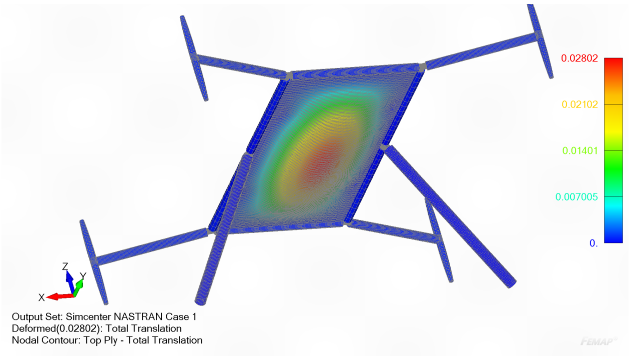
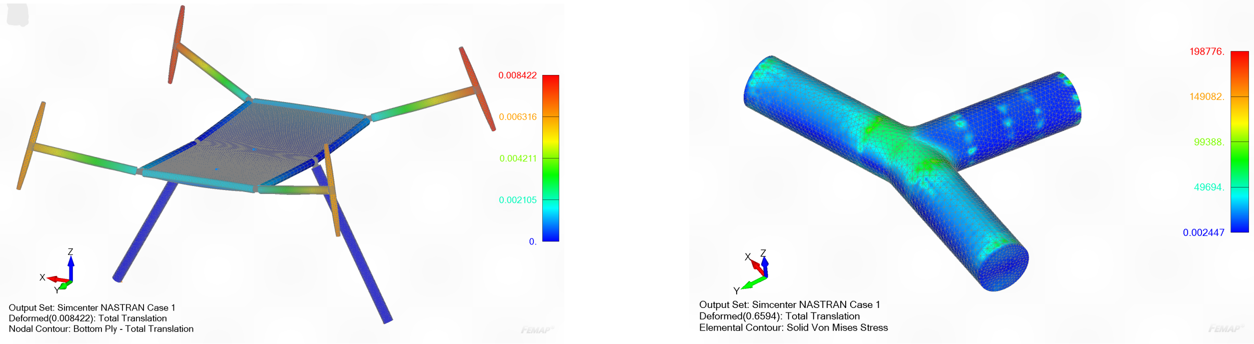


Figure 5.10: Total translation [m] contour

Load case B: Peak Thrust in Air

An ultimate load factor of 2 is applied on the maximum thrust described in Table 5.7. A higher safety factor has been chosen to compensate for the larger simulation uncertainties present in load case B, such as dynamic loadings. In Figure 5.11a a maximum deflection of 0.00842m (or 8.42mm) occurs at the tip of the propeller rods. Specifically, the aft propeller rods experience more deflection as the centre of gravity is slightly more concentrated in the front.

The deflection is caused by bending of the propeller rods, bending of the base plate under payload/propulsion weight and the bending of the chassis/propeller rod connector. It was observed however that the deflection is mainly caused by the chassis connector. From Figure 5.11b, a stress of approximately 60Mpa is present at the T-junction.



(a) Exaggerated View of Total translation [m] contour

(b) Von Mises Stress [Pa] contour on Chassis Connector

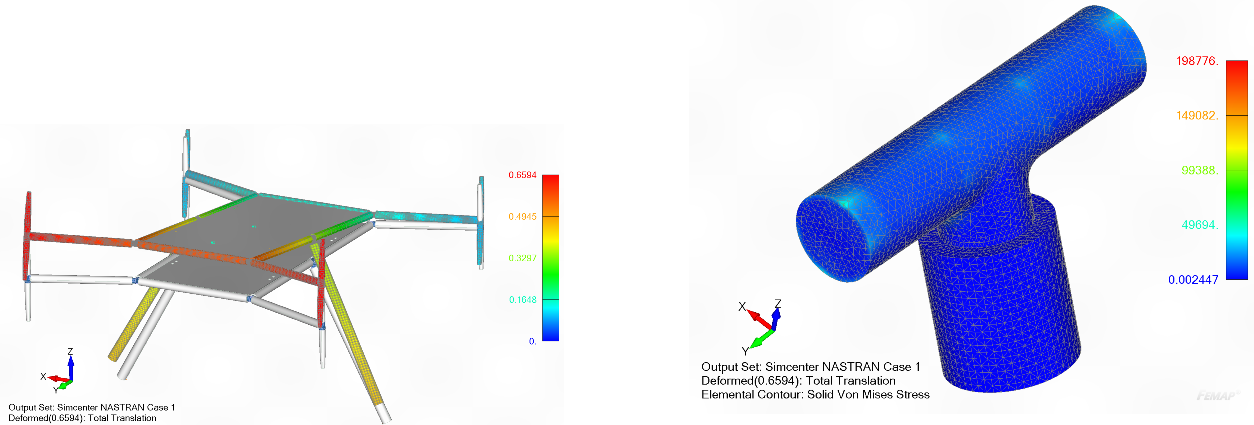
Figure 5.11: Stress-strain results of load case B

Load Case C: Hard Landing

The structure of the drone should be robust enough to withstand a hard landing. Such an event may occur due to a faulty control system, or if the FTS is activated, as previously elaborated in Table 3.6. To simulate a hard landing, the drone is modelled to land under its own weight. In other words, the drone free-falls ($1g$) without any deceleration from the propellers. Such a scenario is modelled by applying a load to the base of the landing gear whilst fully constraining the central node of the base plate. Fixing one node of the base plate (instead of the landing gear or the entire base plate) allows all other nodes to move freely. It is stipulated that such a static set-up most accurately mimics an equivalent dynamic or impact analysis (beyond the scope of the project).

Deformation contour, as visualised in Figure 5.12a, shows that applying a total landing force of 116N leads to a maximum deflection of 0.2074m. It seems that the entire drone chassis moves upwards at once. For a reference scale, The un-deformed drone is shown in white/grey. Notice that the deflections themselves are asymmetric. This is most likely attributed to an uneven weight distribution of the drone itself (namely the fuel cell/battery compartment).

On top of the stiffness contour, the stress behaviour of the landing gear T-connector is seen in Figure 5.12b. It was interesting to observe that the landing gear connector largely remained stress-free. This is most likely explained by the connector moving upwards freely, rather than deforming.



(a) Exaggerated View of Total translation [m] contour

(b) Von Mises Stress [Pa] contour on Landing-Gear/Chassis Connector

Figure 5.12: Stress-strain results of load case C

Load Case D: Nominal Hover

Although the critical load case for lift occurs when motors provide peak thrust, it is still important to estimate deformation during nominal hover. Such information, for example, can be useful for a later iteration of the control system that incorporates displacements of the payload module during an inspection.
With that in mind, nominal hover is simulated. Similar boundary conditions as load case B as applied here. The key difference is the magnitude of thrust applied. Unlike load case B, peak thrust is no longer used. Instead, the amount of thrust needed to maintain static equilibrium with the drone's weight is simulated. This nominal thrust can be found in Table 5.7. Simulated deformation contour is pictured in Figure 5.13. A maximum deflection of 0.00588m (or 5.88mm) occurs at the tip of the propeller rods.

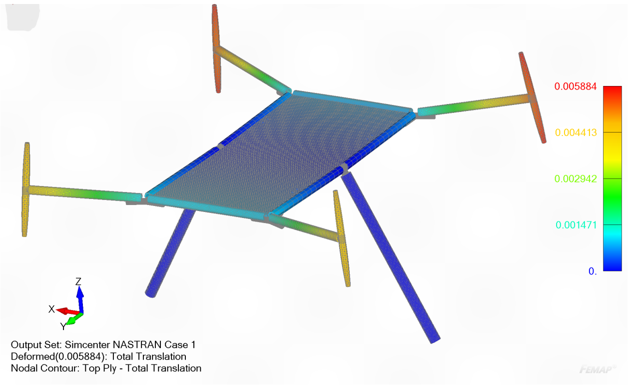


Figure 5.13: Load case D: Nominal hover

5.4.3 Linear Modal Analysis

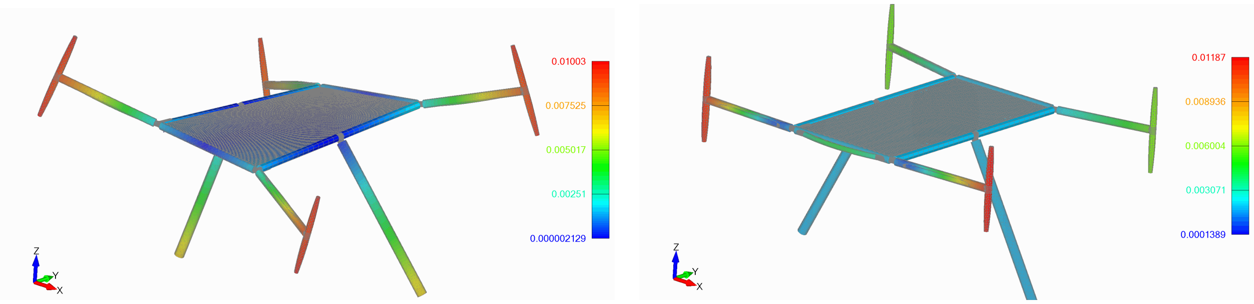
On top of static analyses, a linear modal analysis has been performed on the chassis to compute the eigen-modes of the structure. The most important part of such an eigen-analysis is to verify that the vehicle's natural frequency is sufficiently higher than possible excitation frequencies that can be encountered during flight. The bare minimum compliance is to check that the propeller frequency of 33.3 Hz $(\approx 2000 rpm)$ does not lead to drone's resonance. Resonance of the drone may severely reduce the overall safety, reliability and predictability of the vehicle.

Boundary Conditions

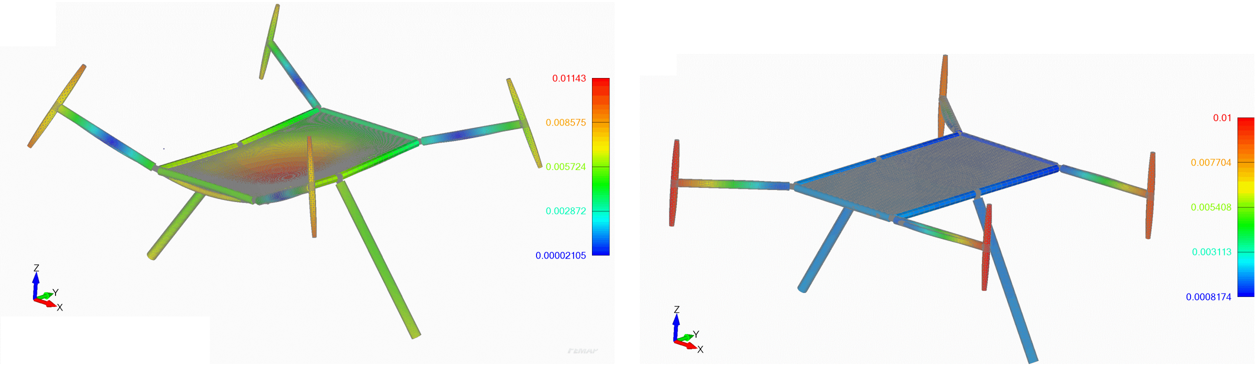
Recall that for static analyses, both loads and constraints (explained previously in subsection 5.4.2) were defined. However, for modal analyses, only constraints [40] are required. As a consequence, the weight loads described in Table 5.6 had to be replaced as point masses. These point masses were applied as 'mass elements' at the geometric centre of their components. In further analyses, point masses should be ideally be remodelled as inertial matrices to include the inertia of the components themselves that are attached to the drone.
In order to realistically model the drone's eigen-frequencies in flight, the drone is left **unconstrained** during the modal simulation. As a result, the first six eigen-modes represent rigid body motion. These include three degrees of freedom in translation (T_x, T_y, T_z) and in rotation (R_x, R_y, R_z). Their corresponding eigen-frequencies are effectively zero. Eigen-frequcies of rigid body motion can be neglected as they do not contribute to resonance. All in all, the non-rigid body modes of interest are provided in subsequent text.

Results of Non-Rigid Body Modes

Neglecting rigid body motion, the lowest four non-rigid modes are shown in Figure 5.14. After these shape-functions are scaled for the correct mass, the final eigen-frequencies are obtained. These frequency values are tabulated in Table 5.8.



(a) Mode 1 (b) Mode 2



(c) Mode 3 (d) Mode 4

Figure 5.14: Exaggerated Displacement Contours of Non-Rigid Body Modes

Table 5.8: Non-rigid body modes

Mode	**Frequency** [Hz]
Mode 1	39.48
Mode 2	47.92
Mode 3	53.07
Mode 4	62.31

All in all, the preliminary modal analysis verifies that the lowest non-rigid eigen-mode still remains above the frequency of the propellers. Nevertheless, the frequency margin remains within the same order of magnitude. Higher fidelity analysis or simply prototype testing is recommended to check the results of FEA. Should resonance be observed during testing, a design optimisation can be carried out. Such a design optimisation should aim to increase the frequency of the lowest eigen-mode without changing the topology of the drone structure or significantly increasing the mass of the drone.

Limitations FEA

It should be noted that even though the Von Mises stresses are computed next to the displacement the values are not deemed very realistic. In order to obtain realistic values all connections should be properly modelled which was regarding the time constraint of the DSE not possible. This means for example that 1D elements will have to be used for bolted connections instead of the glued connections used in the model.

Moreover, has the landing been modelled very simplistically. A much more accurate would be to simulate the impact during the landing through a dynamic analysis. Another, more practical, solution is to perform extensive testing on the landing gear after production.

It also should be noted that the model consists of a lot of nodes with a total of 635,000. As the current simulations performed were relatively simple this was not an issue but once more complex simulations will be performed, for example, dynamic, the mess size will need to be significantly reduced in order to keep computational time within limits. Moreover is it worth mentioning that a fatigue analysis has not been performed. The reason for this is the limited time available during DSE and the fact that fatigue for composites is much less likely to become problematic compared to metals such as aluminium. However, it is advised to properly check for this during the extensive testing phase.

5.5 Design iterations

Author: Nachiket, Louis

Although the FEA performed has its limitations, regarding the design a few things can already be denoted and could be taken into account during an iteration of the chassis design.

First of all, based on the analysis it could be concluded that the deformations of the chassis mainly results from the deformation of the connectors instead of the tubes. This can clearly be seen in Figure 5.13. An option could be to produce the connectors through additive manufacturing out of aluminium. Please be aware of the fact if doing so that aluminium in contact with CFRP results in galvanic corrosion. Although this process might take a while if would still be advised to coat either the brackets or tubes to prevent this from happening. To keep weight-increment to a minimum the inside of the connector could be made hollow. Besides, the tubes themselves can likely be made smaller as these show very little deformation.

Moreover, especially in load case A, it can be seen that the base plate deflects quite a bit and it might be necessary to stiffen the base plate in a new iteration. However, it should be noted that in the design the inertia of the hydrogen tank and fuel cell has not been modelled (the components have been modelled as point masses with no inertia). Furthermore, a glued contact constraint has been applied whereas a more realistic approximation would be to use a bolted connection on which a pretension is applied. Therefore, before stiffening the base plate in a new iteration, it is advised to first improve the Finite Element Analysis.

In addition, to reduce deflection upon landing (load case C), a design solution is to include a damper in the landing gear rod. Another useful iteration may be to increase the surface area of the bottom landing gear rod (possibly by changing the profile from round to square). This iteration may improve landing stability and ensure that the drone lands in an upright orientation.

Finally it should be denoted that the design of the structure is a first prototype with a low production rate. This entails that many components are off-the-shelf and the chassis structure has been chosen such that is easy to manufacture and assemble. All all components be easily accessible too. However, a final version of the chassis shall probably be more of a monocoque design which might become feasible once production rates increase. The main benefit of a monocoque design is reduced weight and thus increased endurance. In addition, especially once additive manufacturing methods become more advanced and cheaper, topology optimisation of the chassis might also be very interesting to look into.

6: Guidance, Navigation, and Control

Guidance, Navigation, and Control (GNC) can be considered the brains behind every UAV drone. It provides the demanded intelligence and control capabilities that the drone shall require to perform its operations effectively. Additionally, the autonomous nature of TurbEye makes such system even more crucial, enabling it to autonomously navigate through complex environments, adapt to dynamic situations, and execute tasks with precision.

As a matter of fact, each of the subsystems of GNC is a vital part by itself. Guidance treats the drone's trajectory planning, object avoidance, and path optimisation. Navigation details how the selected sensors (GPS, IMU, and Barometer) determine the drone's position, velocity, attitude, and angular rates. Finally, Control allows for the management of the drone's actuators in order to provide the necessary forces and moments to control the drone. An overview of the GNC architecture showing the different modules is given in Figure 6.1.

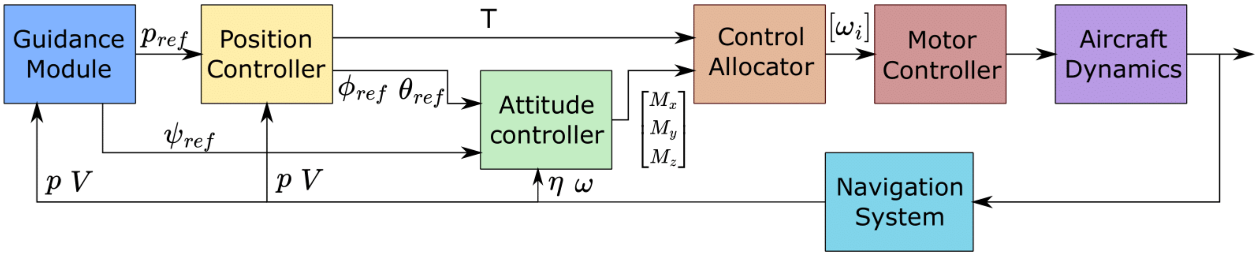


Figure 6.1: Integrated control architecture with position and attitude control

This chapter obeys the following structure: firstly, the sensor selection shall be conducted, succeeded by their respective modelling in Simulink® R2023a [41]. Secondly, the drone model used to simulate the GNC software is described. After this, each part of guidance, control and navigation is detailed. First, the control logic of the drone is given where both the control law and allocation of the drone has been designed. The second section described is navigation. The drone makes use of multiple sensors, whose output is used by the navigation logic to be able to estimate the drone's position and orientation. Lastly, the guidance algorithm is designed and explained, which takes care of planning the trajectory of the drone to be able to perform wind turbine inspections in an offshore wind farm. The chapter is closed off with verification of the GNC software.

6.1 Sensor Selection

Author: Tomás

Our drone is equipped with 4 sensors for position and attitude determination. The IMU (Inertial Measurement Unit) contains an accelerometer and a gyroscope; the accelerometer provides pitch and roll information, whereas the gyroscope additionally measures the yaw data [42]. Due to the IMU's significant drift in position determination, GNSS was used for the determination of the drone's positioning in the ECEF reference frame. Finally, by means of a barometer, one can obtain a more precise measure of the altitude. Moreover, devices are required for the three inspection techniques which were previously defined: visual testing, passive thermography, and geometry-based inspection.

After thorough research, specific COTS sensors were selected. Each sensor can be found in Tables 6.1 to 6.6, along with its relevant specifications, such as mass, weight, precision, etc.

Table 6.1: GPS: *HGLRC M100 MINI GPS [17]*

Characteristic	**Value**
Dimensions (l x w x h) [mm]	15 x 15 x 5.2
Mass [g]	2.7
Average Power [W]	0.2
Baud Rate [bps]	115200
Velocity accuracy [m/s]	0.05
Horizontal position accuracy [m]	2.0
Receiver Sensitivity [dBm]	-166

Table 6.2: Thermal Camera: *DJI Zenmuse H20T [24]*

Characteristic	**Value**
Dimensions (l x w x h) [mm]	167 x 135 x 161
Mass [g]	828±5
Average Power [W]	27
Image quality [MP]	20
Sensitivity [mK]	≤ 50
Range Precision [cm]	≤ 2

Table 6.3: Barometer: *BMP390 [15]*

Characteristic	**Value**
Dimensions (l x w x h) [mm]	2 x 2 x 0.75
Power Required [μW]	11.52
Accuracy [Pa]	±3

Table 6.4: 3D Scanner: *MotionCam-3D Color L+ [22]*

Characteristic	**Value**
Dimensions (l x w x h) [mm]	628 x 68 x 85
Mass [g]	1150
Average Power [W]	60
Scanning range [cm]	130 - 378
Static resolution [$\#points$]	2 Million
Static FPS	2
Static Precision [mm]	1.15
Dynamic resolution [$\#points$]	0.9 Million
Dynamic FPS	20
Dynamic Precision [mm]	1.68

Table 6.5: LiDAR (+ Additional IMU): *Livox Mid-360 [43] (+ TDK ICM-40609-D)*

Characteristic	**Value**
Dimensions (l x w x h) [mm]	65 x 65 x 60
Mass [g]	265
Power Required [W]	6.5
Horizontal FOV [deg]	360
Vertical FOV [deg]	-7 to 52
Range Precision [cm]	≤ 2
Point rate [$points/s$]	200,000

Table 6.6: IMU: *MICROSTRAIN 3DM-CV7-AHRS [16]*

Characteristic	**Value**
Dimensions (l x w x h) [mm]	38 x 24 x 8.6
Mass [g]	8.3
Maximum Power [mW]	280
Roll/Pitch static accuracy [deg]	0.25
Dynamic accuracy [deg]	0.5
Acc. Bias instability [µg]	18
Gyro Bias instability [deg/hr]	1.5

While studying the sensors which were ideal for the drone, some important considerations were identified:

- For the sake of redundancy, and TurbEye's fail-safe philosophy, an additional GPS sensor, barometer and IMU have to be integrated in the drone. Thus, two M100 MINI GPS, the LiDAR's integrated IMU, and the IMU's integrated barometer will fit this purpose. The payload devices were not duplicated, as their failure is not crucial for the functioning of the drone. Additionally, the selected flight computer also contains a secondary IMU, GPS, and barometer.
- The receiver sensitivity of a GPS refers to the minimum power level of the GPS signal that the GPS receiver can effectively detect and process. In this context, dBm stands for decibels relative to 1 milliwatt, which is a unit of power.
- In the initial research stage, it was believed that an RTK (Real-Time Kinematics) device would need to be added to the GPS to provide an accuracy in the order of centimeters. The implementation of this device

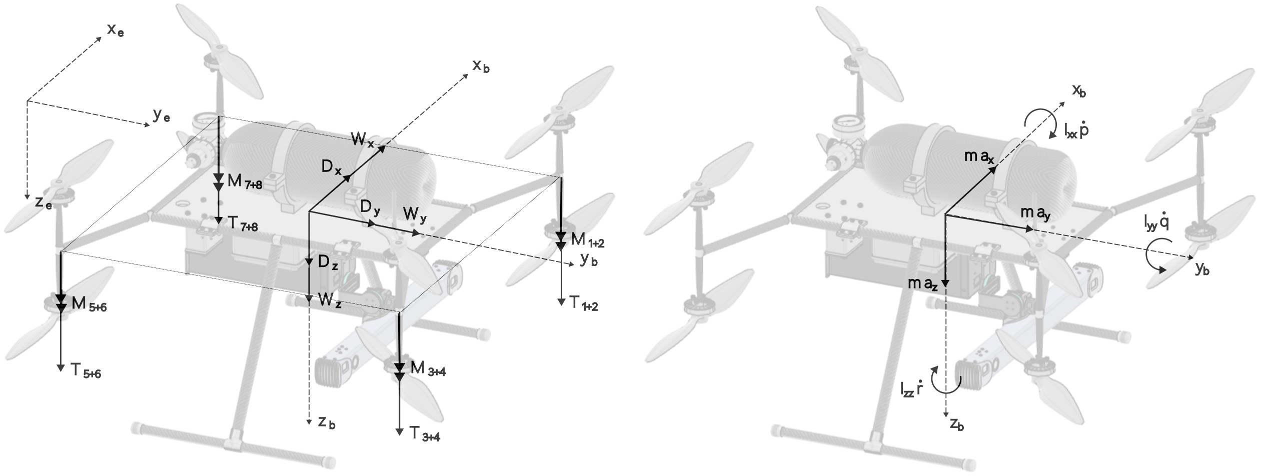
enhances precision by comparing the drone's location with a static base station which has its location defined by other methods besides GPS. However, after a meeting with BladeInsight's CEO, one perceived that there was no need for such, as there is no optimal location to place the base station.

- The selected LiDAR already integrates a high-quality IMU.
- The barometer's weight is considered negligible.

6.2 Drone Model

Author: Radu

In order to develop the GNC, a drone model was constructed in Simulink® This drone model is simplified with respect to its real world equivalent by a set of assumptions which are tabulated in Table 6.7. The simplified free body diagram of the drone model in flight, grouping the forces and moments generated by paired propellers, is displayed in Figure 6.2a, alongside the body-fixed and Earth-fixed reference frames. The kinematic diagram of the drone model is displayed in Figure 6.2b. The equations of motion are displayed in eq. (6.1), in which T and $[M_x M_y M_z]^T$ are obtained from eq. (6.2).



(a) Free body diagram of the drone model

(b) Kinematic diagram of the drone model

$$m\begin{bmatrix}a_x\\a_y\\a_z\end{bmatrix}=\begin{bmatrix}0\\0\\T\end{bmatrix}+\begin{bmatrix}N_x\\N_y\\N_z\end{bmatrix}+\begin{bmatrix}D_x\\D_y\\D_z\end{bmatrix}+\begin{bmatrix}W_x\\W_y\\W_z\end{bmatrix} \qquad \begin{bmatrix}I_{xx}&0&0\\0&I_{yy}&0\\0&0&I_{zz}\end{bmatrix}\times\begin{bmatrix}\dot{p}\\\dot{q}\\\dot{r}\end{bmatrix}=\begin{bmatrix}M_x\\M_y\\M_z\end{bmatrix} \tag{6.1}$$

Table 6.7: Drone model assumptions

Identifier	**Description**
ASM-GNC-Model-01	The Earth-fixed reference frame is set up using the North, East, Down (NED) convention and is assumed to be an inertial frame.
ASM-GNC-Model-02	The drone is rigid. (i.e. the drone's structure does not strain)
ASM-GNC-Model-03	The drone operates in the International Standard Atmosphere.
ASM-GNC-Model-04	The origin of the body-fixed reference frame is located at the center of gravity of the drone.
ASM-GNC-Model-05	The inertia tensor of the drone is diagonal, meaning that the only I_{xx}, I_{yy} and I_{zz} are non-zero terms.
ASM-GNC-Model-06	The mass and inertia tensor of the drone are constant in time.
ASM-GNC-Prop-01	The center of gravity of the drone and the points of action of the forces and moments generated by the propellers lie in the same plane.

Table 6.7: Drone model assumptions

Identifier	**Description**
ASM-GNC-Prop-02	The center of gravity of the drone lies at equal distances from the points of action of the forces and moments generated by the propellers.
ASM-GNC-Prop-03	The points of action of the propeller forces lie in the corners of a square with the side length of $d_p = 0.8175[m]$.
ASM-GNC-Prop-04	The propeller's thrust is aligned with the z-axis of the body-fixed reference frame.
ASM-GNC-Prop-05	The propellers are placed along the $45°, 135°, 225°, 215°$ lines in the x-y plane.
ASM-GNC-Prop-06	The propeller's thrust and moment are proportional to the square of the propeller's rotational rate and the air density: $F = k_F \rho \omega^2$ and $M = k_M \rho \omega^2$.
ASM-GNC-Prop-07	The aerodynamic coefficients of the propellers situated in the wake of another propeller are reduced by a factor of $1/k_c = 1.5607$. [44] [45]
ASM-GNC-Prop-08	The experimental data used to obtain the aerodynamic coefficients was taken at sea level conditions.
ASM-GNC-Drag-01	The drag coefficient of the drone is estimated to $C_D = 0.8$. [46] [47]
ASM-GNC-Drag-02	The point of action of the drag is at the center of gravity of the drone.
ASM-GNC-Wind-01	The wind can modelled using a Von Karman Wind Turbulence Model. [48]
ASM-GNC-Wind-02	The reference wind intensity at 6 [m] altitude is estimated to be 8 [m/s] [49] .
ASM-GNC-Wind-01	The weight of the drone is acting at its center of gravity.
ASM-GNC-Ground-01	The ground is represented by the x-y plane at z = 0.
ASM-GNC-Ground-02	Ground interaction generates reaction forces and prevents the drone from reaching positive z-direction coordinates in the NED reference frame.
ASM-GNC-Ground-03	The normal force point of action is at the center of gravity of the drone.
ASM-GNC-Motor-01	The motor responds like a first order system with a time constant of $\tau_m = 20[s^{-1}]$.

Propulsion system model

The propulsion system to be modelled consists of 8 propellers, co-axially placed in pairs on the 4 arms of the drone. The co-axial propellers are counter-rotating. In this manner, the upper propellers (1 and 5) and the lower propellers (4 and 8) are creating a positive moment. At the same time, propellers 2, 3, 5, and 6 are creating a negative moment. Overall, the assemblies of propellers 1&2 and 5&6 will be creating a resultant positive moment, while the other two will be creating a balancing negative moment. Additionally, this setup enables the control allocation to be designed similarly to a conventional quadcopter, but this will be discussed in further detail in subsection 6.3.2. The resultant thrust and moments generated by the propulsion system are computed using eq. (6.2).

$$\begin{bmatrix} T \\ M_x \\ M_y \\ M_z \end{bmatrix} = \begin{bmatrix} -k_F & -k_F k_c & -k_F & -k_F k_c & -k_F & -k_F k_c & -k_F & -k_F k_c \\ -k_F \frac{d_p}{2} & -k_F \frac{d_p}{2} k_c & -k_F \frac{d_p}{2} & -k_F \frac{d_p}{2} k_c & k_F \frac{d_p}{2} & k_F \frac{d_p}{2} k_c & k_F \frac{d_p}{2} & k_F \frac{d_p}{2} k_c \\ k_F \frac{d_p}{2} & k_F \frac{d_p}{2} k_c & -k_F \frac{d_p}{2} & -k_F \frac{d_p}{2} k_c & -k_F \frac{d_p}{2} & -k_F \frac{d_p}{2} k_c & k_F \frac{d_p}{2} & k_F \frac{d_p}{2} k_c \\ k_M & -k_M k_c & -k_M & k_M k_c & k_M & -k_M k_c & -k_M & k_M k_c \end{bmatrix} \times \begin{bmatrix} \omega_1^2 & \omega_2^2 & \omega_3^2 & \omega_4^2 & \omega_5^2 & \omega_6^2 & \omega_7^2 & \omega_8^2 \end{bmatrix}^T \rho \tag{6.2}$$

The aerodynamic coefficients were obtained by performing two least square fits on experimental data[50]. The data, among others, featured values for thrust and torque generated by the propeller against its angular velocity. After performing the least squares fits, the coefficient corresponding to the second power were extracted from the obtained polynomials. To obtain non-dimensional thrust and torque coefficients of the propellers, these values were divided by the standard sea level air density of $1.225[kg/m^3]$. The obtained coefficients were $k_F = 3.7257 \times 10^{-4}[m^4 rad^{-2}]$ and $k_M = 1.1455 \times 10^{-5}[m^5 rad^{-2}]$.

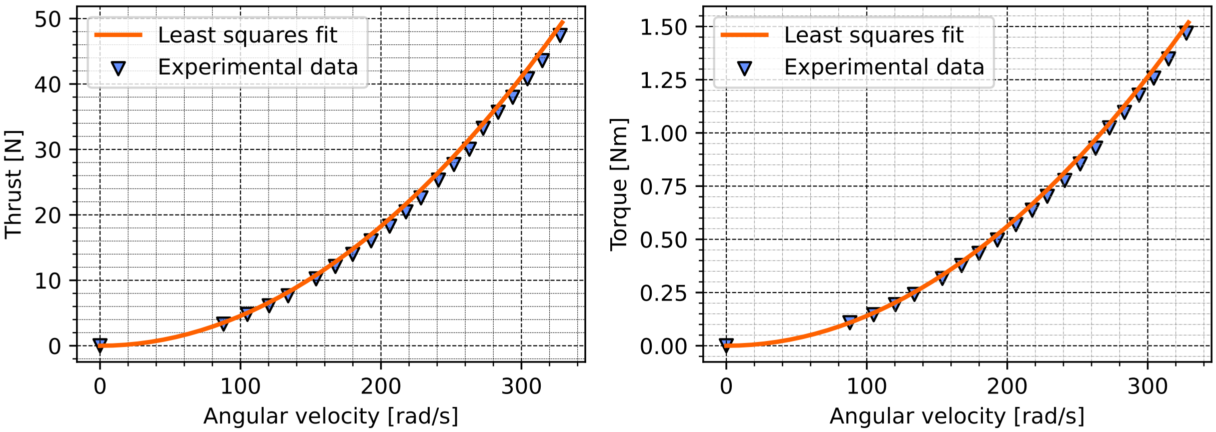


Figure 6.3: Plots of the thrust and torque curves versus the experimental data

Drag model

The drag that acts on the drone is modelled independently on all axis of the body-fixed frame. Each component of the velocity vector (expressed in the body-fixed reference frame), and the cross-sectional area of the drone with respect to that specific axis, are used to calculate the respective drag. The cross-sectional areas of the drone, with respect to that specific axis, are derived from CAD and reported in Table 3.1. The equation used to model drag on the drone is displayed in scalar form, and expressed in the body-fixed reference frame in eq. (6.3).

$$\begin{bmatrix} D_x \\ D_y \\ D_z \end{bmatrix}_b = -\frac{1}{2}\rho \begin{bmatrix} V_x^2 \\ V_y^2 \\ V_z^2 \end{bmatrix}_b \circ \begin{bmatrix} S_x \\ S_y \\ S_z \end{bmatrix} C_D \tag{6.3}$$

Wind model

The wind model was implemented using the already available *Von Karman Wind Turbulence Model (Continuous)* [48] in Simulink®. It was set to specification "MIL-F-8785C" and wind speed reference intensity of $8[m/s]$ at $6[m]$ altitude [49]. The model generates turbulence for the linear and angular velocities, which then are added to these respective velocities, as described by eq. (6.4) and eq. (6.5). For the sake of evaluating the performance of the GNC, the influence of the wind may not be taken into consideration in order to remove randomness.

$$\begin{bmatrix} V_x \\ V_y \\ V_y \end{bmatrix}_b + \begin{bmatrix} V_x \\ V_y \\ V_y \end{bmatrix}_{b,wind} = \begin{bmatrix} V_x \\ V_y \\ V_y \end{bmatrix}_b^* \tag{6.4}$$

$$\begin{bmatrix} p \\ q \\ r \end{bmatrix}_b + \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix}_{b,wind} = \begin{bmatrix} p \\ q \\ r \end{bmatrix}_b^* \tag{6.5}$$

Gravity model

The weight of the drone in the body-fixed reference frame is obtained by transforming the weight in the Earth-fixed reference frame. The transformation calculation, involving the Direction Cosine Matrix (DCM), is described in eq. (6.6).

$$DCM \times \begin{bmatrix} 0 \\ 0 \\ mg \end{bmatrix}_e = \begin{bmatrix} W_x \\ W_y \\ W_z \end{bmatrix}_b \tag{6.6}$$

Ground model

The ground is implemented through a number of steps: reaction forces and bounds for z-direction position/velocity. Should the drone be at z = 0 [m] in the Earth-fixed reference frame, then a normal force and reaction moments is modelled to act at drone's centre of gravity. Additionally, the model additionally prevents the drone from reaching positive z-direction position. The normal force acting on the drone is calculated by eq. (6.7) and eq. (6.8).

$$DCM^{-1} \times \begin{bmatrix} F_x \\ F_y \\ F_z \end{bmatrix}_b = \begin{bmatrix} F_x \\ F_y \\ F_z \end{bmatrix}_e \quad (6.7) \qquad DCM \times \begin{bmatrix} 0 \\ 0 \\ -F_{z,e} \end{bmatrix} = \begin{bmatrix} N_x \\ N_y \\ N_z \end{bmatrix}_e \quad (6.8)$$

Motor response model

The motor speed is related to the desired speed by first order differential equation, as described in eq. (6.9). The time constant had been chosen to be $\tau_m = 20[s^{-1}]$. [51]

$$\dot{\omega}_i = \tau_m \left(\omega_{des,i} - \omega_i\right) \quad (6.9)$$

Sensor Modelling

For each sensor, a specific Simulink®command block was used; the blocks *Three-axis Inertial Measurement Unit* [52], *GPS* [53], and *Pressure Altitude* [54] were applied to replicate the IMU, GPS, and Barometer sensors, respectively.

Regarding the IMU's inputs, it is necessary to specify the linear and angular accelerations, the angular velocities, the location of the center of gravity, and the gravity vector, all in terms of the fixed body frame. Since the fixed frame used by the block considers the z-direction positive upwards, gravity is considered to be negative. As outputs, one will retrieve the measured accelerations and angular rates.

In terms of the GPS, the Cartesian coordinates [m] and the velocity [m/s] in the specified local coordinate system are required as inputs. Subsequently, the relevant outputs are the updated coordinates (in the LLA coordinate system) and the velocity of the receiver after a given time step. Lastly, the barometer only requires the static pressure, subsequently providing the corresponding pressure altitude.

Verification

All subsystems that make up the drone model have been unit tested. The performed tests were tabulated in Table 6.8.

Table 6.8: Drone Model Tests

ID	Model Part	Test Description	Test Rationale	Result
UT-DM-01	Gravity model	Compares the expected weight vector with the one generated by the gravity model. The input for this test is the direction cosine matrix for an arbitrary set of Euler angles. The mass of the drone and the gravitational acceleration are constants.	Used to assess the accuracy of the calculation of the weight vector in the body-fixed reference frame.	Passed
UT-DM-02	Propulsion system model	Compares the expected thrust and moments vector with the one generated by the propulsion system model. The input for the test is an arbitrary set of angular velocities of the propellers and an arbitrary air density.	Used to assess the accuracy of the calculation of the resultant thrust and moments vector.	Passed
UT-DM-03	Motor response model	Compares the expected motors' responses with the respective responses generated by the model. The input for the test is an arbitrary set of motor angular velocity commands.	Used to assess the accuracy of the calculation of the motors' responses to the commanded angular velocities.	Passed

Table 6.8: Drone Model Tests

ID	Model Part	Test Description	Test Rationale	Result
UT-DM-04	Drag model	Compares the expected drag vector with the one generated by the drag model. The input is an arbitrary linear velocity vector and arbitrary air density.	Used to assess the accuracy of the calculation of the drag vector.	Passed

The air density is obtained using the built-in ISA model. [55] The wind is obtained using the built-in *Von Karman Wind Turbulence Model (Continuous)*. [48] The influence of the IMU is obtained using the built-in *Three-axis Inertial Measurement Unit* model. [52] The influence of the GPS is obtained using the built-in *GPS* model. [53] The influence of the barometer is obtained using the built-in *Pressure Altitude* model. [54] These modules have already been verified before release, as they are readily available in Simulink®, so no further verification is needed.

6.3 Control

Author: Mike

The inherently unstable nature of UAV configurations necessitates a rigorous approach to the analysis and design of UAV control technologies, as well as a thorough understanding of stability issues. This section gives a description of the control module used to stabilize the drone and get it to track the reference given by the guidance module. For that, it requires knowledge of the system state as feedback supplied by the navigation module.

The control module consists of two parts: the control law and the control allocation. Since the control problem is over-determined, meaning there are less signals to track than there are to command, the control law converts the reference signals into a virtual control input which matches the number of signals to be tracked, namely four; three positional signals and the yaw angle. The control allocation takes this virtual control input and determines the actual control inputs. In this case, the reference signals are the position in the earth-fixed reference frame $\begin{bmatrix} x_{ref} & y_{ref} & z_{ref} \end{bmatrix}$ and the yaw angle ψ_{ref}. The virtual control input is the thrust in vertical direction and moments about each body axis $\begin{bmatrix} T & M_x & M_y & M_z \end{bmatrix}$. The actual controllable inputs are the rotational speed of the eight propellers $[\omega_i]_c$, with $i \in \{1, 2, \ldots, 8\}$. The interaction of these two parts as well as the inputs and outputs of these are shown in Figure 6.4.

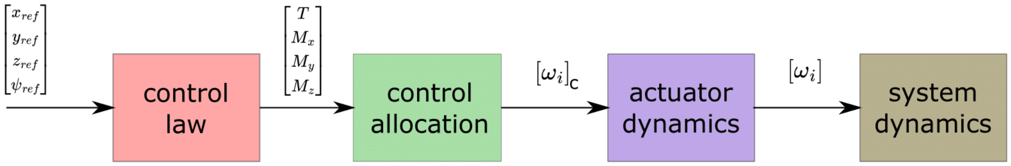


Figure 6.4: Control module interaction

6.3.1 Control Law

Author: Mike

For the control law, a consideration is made with regards to which control technique is used. A choice is to be made to whether a linear or nonlinear control technique is used. While the drone is a nonlinear system, a linear control technique can be used to control the drone based on the linearised equations of the model. For all practical purposes, when high performance and agility are not a necessity, a linear control technique gives the desired performance. Therefore, the choice is made to use a linear control technique. There are two basic linear control techniques considered, also described in [56, sec. 27.3].

The first control technique is to perform a successive loop closure. In this case, loops are nested by arranging that the reference commands for the inner loops are provided by the outer-loop controller: the outermost position control loop provides velocity commands and the outer velocity control loop provides a reference for the inner attitude control loop. The key advantage of this approach is the ability to handle limits on flight variables and actuator inputs because the reference commands can be saturated before being passed to the inner loop. The control loops are designed as to ensure that the inner control loops result in fast dynamics and each successive loop is slower than the previous one.

The second approach is to design a controller for the full dynamics, using for example a linear quadratic regulator. As such, it has the advantage that it can handle the fully coupled dynamics. The disadvantages are that it is very hard to handle state constraints and it is very sensitive to model errors and discrepancies.

Regardless of the selected control technique, it is customary to split the position and attitude control in two parts, as shown in Figure 6.5, from [31, pg. 253]. Therefore, the controller design for each of these parts is treated separately.

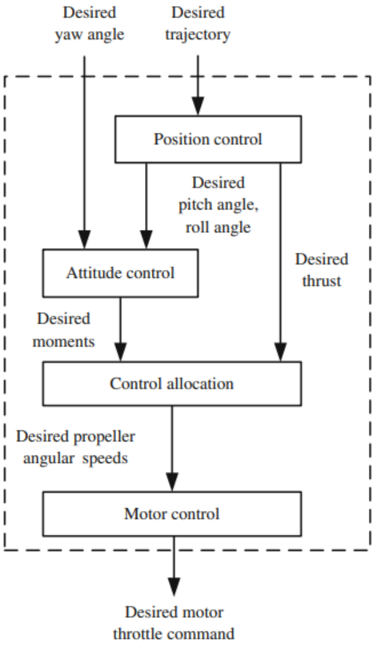


Figure 6.5: High-level control architecture diagram [31, pg. 253]

Attitude Control

For the attitude controller, both the above mentioned techniques are implemented and subsequently compared. The first technique is based on successive loop closure using proportional integral derivative (PID) controllers, while the second technique is based on full state feedback making use of linear quadratic regulator (LQR). The theoretical basis for both these approaches is described in many text books, such as [31] and [56], hence is not repeated here. Merely the design choices made in their implementation and subsequent evaluation is given.

The control architecture for the successive loop closure technique is given in Figure 6.6. The design parameters for this control technique are the control gains $(K_\phi, K_\theta, K_\psi)$ and (K_p, K_q, K_r). These gains are tuned based on loop shaping, which is a frequency domain design technique [57, Chap. 11]. This technique utilizes Bode plots and principles from frequency analysis to change the Bode plot response in such a way that the system has desirable characteristics. The control loop characteristics are shown in Table 6.9. The cross-over frequency dictates the rise time of the response, the gain margin indicates the robustness of the control loop and the phase margin determines the overshoot of the response. Since the moments of inertia around the x- and y-axis is different, the angular velocity loop is tuned to have the same response time. This is accomplished through having the same cross-over frequency. As design guidelines, the inner loop should be set 4 times faster than the outer loop, the phase margin greater than $30[deg]$ and the gain margin greater than $6[dB]$ [58, Chap. 10]. Hence, the inner loop cross-over frequency is set to be a factor of 4 lower than the outer loop. Then, the phase and gain margins still have desirable values.

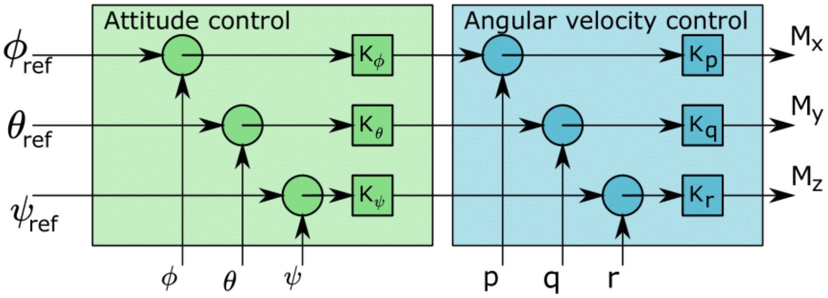


Figure 6.6: Successive loop closure inner attitude control loop

Angular Velocity Control

Control gains		Cross-over frequency [rad/s]	Gain margin [dB]	Phase margin [deg]
K_p	11.275	10.0	Inf	90
K_q	6.166	10.0	Inf	90
K_r	15.136	10.0	Inf	90

Attitude Control

Control gains		Cross-over frequency [rad/s]	Gain margin [-]	Phase margin [deg]
K_p	2.573	2.5	Inf	76
K_q	2.573	2.5	Inf	76
K_r	2.573	2.5	Inf	76

Table 6.9: Design parameters for successive loop closure technique

The control architecture for the second approach is given in Figure 6.7. It is based on the the LQR framework with an integrator embedded in it to improve tracking performance, as described in [56, sec. 27.3.4.1]. The LQR feedback gain matrix K_{lqr} is obtained by solving an optimal control problem, given in eq. (6.10). The control law is then given by eq. (6.11).

$$\begin{cases} J_{lqr} = \min\limits_{\boldsymbol{u}(.)} \int_0^{\inf} \left[\boldsymbol{x}(\boldsymbol{t})^T Q \boldsymbol{x}(\boldsymbol{t}) + \boldsymbol{u}(\boldsymbol{t})^T R \boldsymbol{u}(\boldsymbol{t})\right] dt \\ s.t. \dot{\boldsymbol{x}} = A\boldsymbol{x} + B\boldsymbol{u} \\ \boldsymbol{x}(\boldsymbol{0}) = \boldsymbol{x_0} \end{cases} \tag{6.10}$$

$$\boldsymbol{u}(\boldsymbol{t}) = K_{lqr}\boldsymbol{x}(\boldsymbol{t}) \tag{6.11}$$

The optimal control problem calculates the feedback gain based on the weights assigned to the states and inputs through the weight matrices Q and R. These are the tuning parameters which influence the performance of the controller. If a high weight is assigned to a certain state, it means that the state being away from the target value is punished more heavily. Hence, a higher weighted state converges to its target faster, while if a high weight is assigned to the inputs, the system will use its inputs more conservatively leading to a slower response. The state and input weight matrices chosen to give a desirable response are given in eq. (6.12). It can be noted that the yaw angle is set the have a slower response than the roll and pitch angles because the system becomes unstable if the weight is set too high.

$$Q = diag(\begin{bmatrix} 100 & 100 & 100 & 5 & 5 & 5 & 1000 & 1000 & 200 \end{bmatrix}) \qquad R = 1 \times 10^{-2} diag(\begin{bmatrix} 1 & 1 & 1 \end{bmatrix}) \tag{6.12}$$

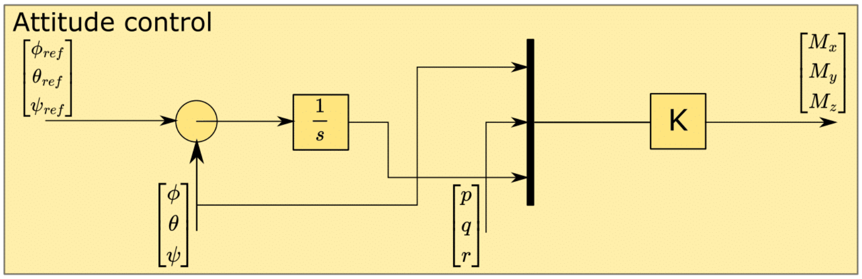


Figure 6.7: Full state feedback inner attitude control loop

A comparison is made based on the controllers' performance, robustness and stability characteristics. Firstly, a reference attitude trajectory is given to be tracked by the system. In order to compare the controllers' performance, the high fidelity model and actuator dynamics are used, but the navigation module is excluded. The system's responses are shown in Figure 6.8. As can be seen, the response for the pitch and roll angles nearly match between the two controllers, however, the PID-based controller is able to track the yaw reference faster than the LQR-based controller. If the response of the yaw angle in the LQR-based controller is attempted to be set faster, than the controller becomes unstable.

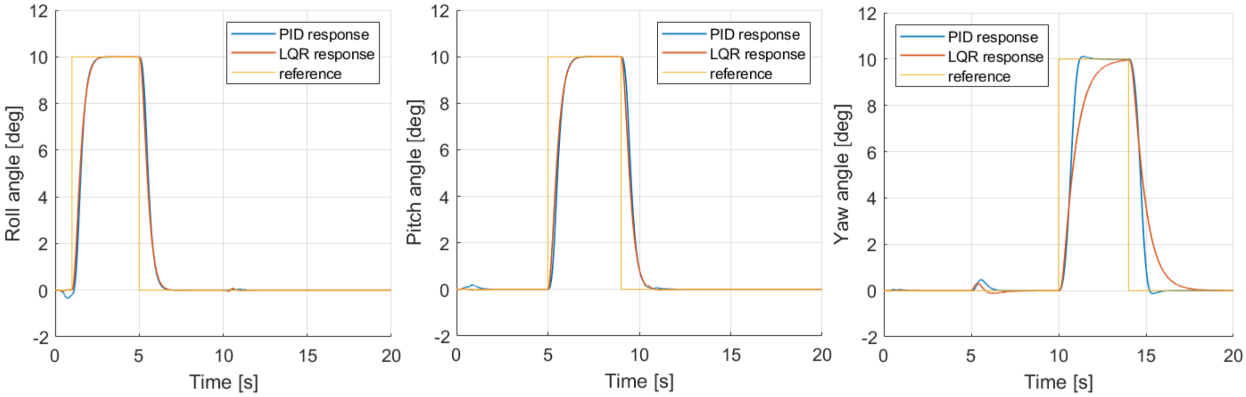


Figure 6.8: Comparison attitude control techniques

Secondly, the navigation module is included, hence the control module receives feedback from the navigation. Under these circumstances, while the PID-based controller's performance degrades, it still remains stable. The LQR-based controller, on the other hands, becomes unstable. This is due to the high values in the gain matrix K which are caused by the high weights in the weight matrix Q. Due to these high values in the gain matrix K, the noise is amplified and the system becomes unstable. To solve this issue, these values need to be decreased by changing the weight matrix Q. It is set to be $Q = diag(\begin{bmatrix}1 & 1 & 1 & 1 & 1 & 1 & 10 & 10 & 10\end{bmatrix})$. The results are shown in Figure 6.9. As can be seen, performance is exchanged for stability. Therefore, the PID-based controller is chosen as the attitude controller.

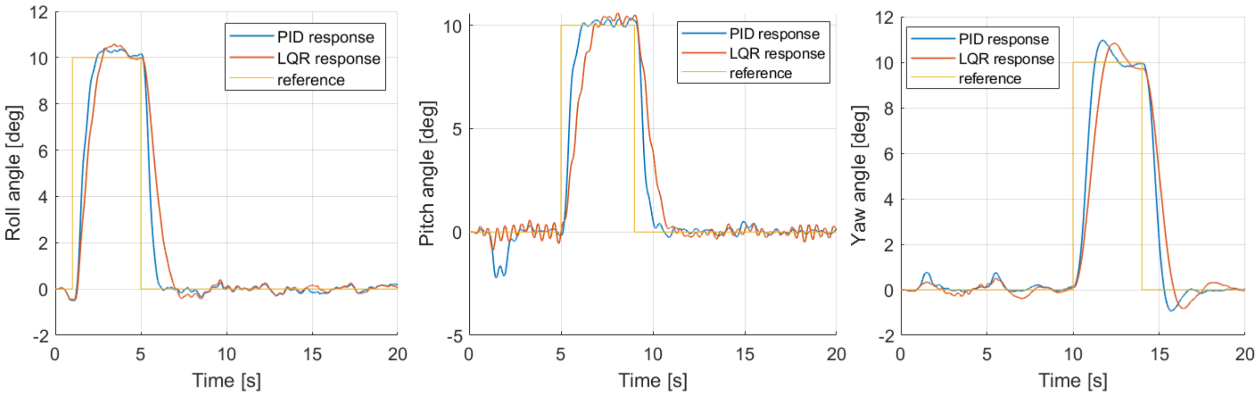


Figure 6.9: Comparison attitude control techniques including navigation

Position Control

The position control is constructed as a loop around the attitude controller. The position control then commands the desired pitch and roll angles to the attitude controller. Hence, the controller is constructed using the successive loop closure technique. The architecture of the position controller is shown in Figure 6.10. Two cascaded loops are used: position control loop and velocity control loop. Both control loops rely on proportional feedback control.

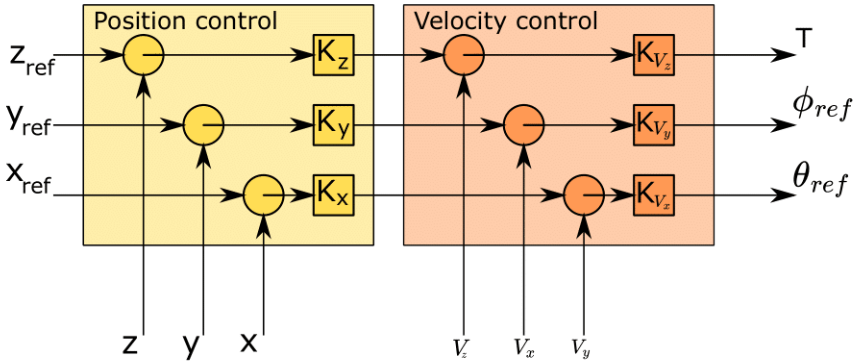


Figure 6.10: Outer position control loop

The design parameters for the position control are the control gains $\left(K_{V_x}, K_{V_y}, K_{V_z}\right)$ and $\left(K_x, K_y, K_z\right)$. These feedback gains are again tuned using loopshaping techniques [57, Chap. 11]. The control loop characteristics are shown in Table 6.10. For the horizontal position control loop, PID terms are included in the velocity control loop to get a desired frequency response. The parameters are tuned such that the control loop has a cross-over frequency of $2[rad/s]$, while maintaining a high phase margin. Subsequently, the horizontal position control loop is tuned to have a cross-over frequency of 4 times less than the inner loop, hence $0.5[rad/s]$. Then for the vertical position control, the velocity control loop is tuned to have a cross-over frequency of $3[rad/s]$ while in the outer loop, PID terms are included and tuned for a cross-over frequency of $0.75[rad/s]$. The integral term eliminates steady-state offset from the reference.

Velocity Control				
Control gains		Cross-over frequency [rad/s]	Gain margin [dB]	Phase margin [deg]
K_{V_x}	$-0.07413(\frac{3}{2}s+1)(1+\frac{2}{15s})$	2.0	Inf	115
K_{V_y}	$-0.07413(\frac{3}{2}s+1)(1+\frac{2}{15s})$	2.0	Inf	115
K_{V_z}	35.892	3.0	Inf	90.0
Position Control				
Control gains		Cross-over frequency [rad/s]	Gain margin [-]	Phase margin [deg]
K_x	0.6095	0.5	27.2	64.5
K_y	0.6095	0.5	27.2	64.5
K_z	$0.494(\frac{4}{5}s+1)(1+\frac{1}{4s})$	0.75	Inf	121

Table 6.10: Design parameters for successive loop closure technique

Full Control Law

For the full control law, use is made of a cascaded architecture with proportional, derivative and integral feedback control. The full control law architecture is shown in Figure 6.11. The reference signals are on the left, $\begin{bmatrix} x_{ref} & y_{ref} & z_{ref} & \psi_{ref} \end{bmatrix}$ of the figure. These are supplied by the guidance module. The control inputs are on the right, $\begin{bmatrix} T & M_x & M_y & M_z \end{bmatrix}$, which are passed on to the control allocation. The feedback signals are at the bottom of the figure, $\begin{bmatrix} x & y & z \end{bmatrix}$, $\begin{bmatrix} V_x & V_y & V_z \end{bmatrix}$, $\begin{bmatrix} \phi & \theta & \psi \end{bmatrix}$, and $\begin{bmatrix} p & q & r \end{bmatrix}$. These are given by the navigation module.

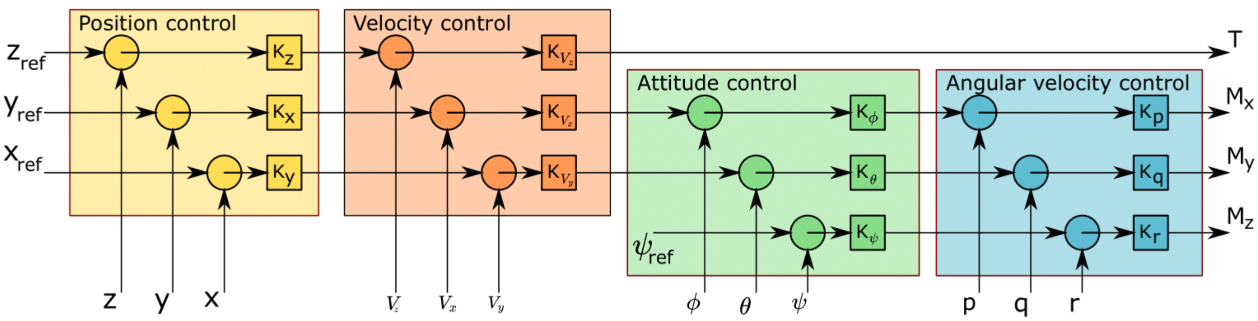


Figure 6.11: Control architecture

As previously mentioned, when using the successive loop closure technique, it is possible to put saturation and rate saturation limits on certain state by saturating the commanded value. This allows for more authority over the behaviour of the system since it is possible to set a limit to certain states. Further, this also prevents the controller to command undesirable values, leading to unstable behaviour. Essentially, it is possible to limit the drone's state to a desired state space. The saturation used for each state is given in Table 6.11. These saturation values were selected after iteratively simulating the drone's response under certain state saturation, until the drone's performance was as desired.

Table 6.11: Control law command saturation

Position Control		Velocity Control		Attitude Control		Angular Velocity Control	
x	-	V_x	$\pm 10 m/s$ $\pm 2 m/s^2$	ϕ	$\pm 20 deg$	ϕ	$\pm 40 deg/s$
y	-	V_y	$\pm 10 m/s$ $\pm 2 m/s^2$	θ	$\pm 20 deg$	ϕ	$\pm 40 deg/s$
z	$\pm 15 m/s$	V_z	$\pm 10 m/s$ $\pm 2 m/s^2$	ψ	$\pm 180 deg$	ϕ	$\pm 40 deg/s$

The control system performance in tracking a position reference is shown in Figure 6.12, with the position plotted against time. As can be seen, the drone successfully tracks the position reference with slight overshoots when it has to change direction of flight of roughly maximum $2.5[m]$. It should be noted that this is because it is flying at a velocity of $5[m/s]$ from which it needs to slow down to change direction. Considering the drone has a high inertia, it takes time to slow down the drone to zero velocity. Additionaly, Figure 6.13 also shows the flight path in the xz-plane, xyz-space and xy-plane. These simulations are performed with the high-fidelity model and navigation module.

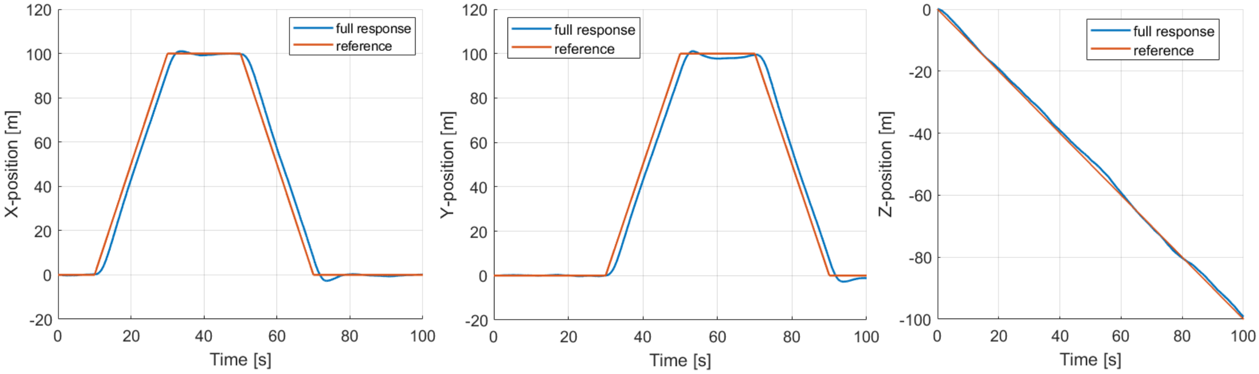
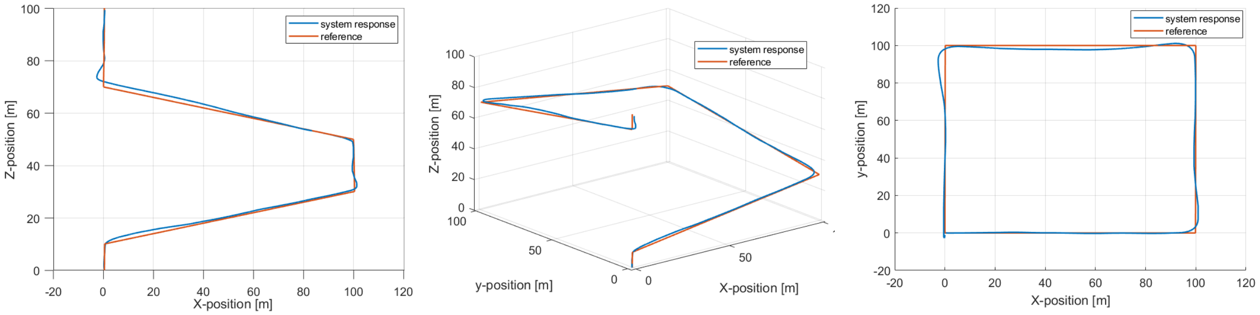


Figure 6.12: Full control system performance in tracking a position reference



(a) Flight path in xz-plane (b) Flight path in xyz-space (c) Flight path in xy-plane

Figure 6.13: Flight path

6.3.2 Control Allocation

Author: Radu

The control allocation sub-module is intended to translate the commanded control inputs, as specified by the control law, to actuator inputs, specifically commanded motor angular velocities. The first step is to account for gravity and drag in order to obtain desired thrust and moments to be generated by the propulsion system. The information used to account for these effects is obtained from the navigation module. They are estimated using eq. (6.13) and eq. (6.14). The moments remain unchanged, but the required thrust changes as described in eq. (6.15).

$$\begin{bmatrix} W_x & W_y & W_z \end{bmatrix}^T = DCM \times \begin{bmatrix} 0 & 0 & mg \end{bmatrix}^T \tag{6.13}$$

$$D_z = \frac{1}{2}\rho V_z^2 C_D S_z \tag{6.14}$$

$$T^* = T - W_z - D_z \tag{6.15}$$

Considering that the aim is to control 8 motors using 4 control inputs, then the system is over-determined. The proposed answer to this is to group the propeller-motor assemblies in pairs, in the same way they are placed on the arms of the drone, and to command the same angular velocity to both propellers. Every pair will consist of two contra-rotating propellers and the lower one is located in the wake of the upper one. Accounting for this, the combined thrust and moment coefficients of a propeller pair are described in eq. (6.16) and eq. (6.17), with $k_c = 0.6407$ as specified in section 6.2. The squared commanded angular velocities are obtained using eq. (6.19) and then the element-wise square root of the vector is extracted to get the actual values. Lastly, a saturation is applied to the commands in order to ensure they remain positive and below the threshold supported by the engines, previously specified in chapter 4, as described in eq. (6.18).

$$k_{F,2} = k_F(1+k_c) \tag{6.16}$$

$$k_{M,2} = k_M(1-k_c) \tag{6.17}$$

$$0 \leq \omega_i \leq 314 \left[\frac{rad}{s^2}\right] \tag{6.18}$$

$$\begin{bmatrix} \omega_{1,2}^2 \\ \omega_{3,4}^2 \\ \omega_{5,6}^2 \\ \omega_{7,8}^2 \end{bmatrix} = \begin{bmatrix} -k_{F,2} & -k_{F,2} & -k_{F,2} & -k_{F,2} \\ -k_{F,2}\frac{d_p}{2} & -k_{F,2}\frac{d_p}{2} & k_{F,2}\frac{d_p}{2} & k_{F,2}\frac{d_p}{2} \\ k_{F,2}\frac{d_p}{2} & -k_{F,2}\frac{d_p}{2} & -k_{F,2}\frac{d_p}{2} & k_{F,2}\frac{d_p}{2} \\ k_{M,2} & -k_{M,2} & k_{M,2} & -k_{M,2} \end{bmatrix}^{-1} \times \begin{bmatrix} T^* \\ M_x \\ M_y \\ M_z \end{bmatrix} \frac{1}{\rho} \tag{6.19}$$

6.4 Navigation

Author: Mike

For the drone to function autonomously, the state of the drone must be available to the controller at high accuracy and low noise. Accurate position is required to perform automatic control for precise inspections. The navigation module in the GNC system implements the estimation of the drone's state. For this, use is made of sensor fusion, which is a technique of optimally blending information from multiple sensors to obtain state estimates. The navigation module is divided in three parts: pre-processing, extended Kalman filter and post-processing, of which a block diagram is shown in Figure 6.14 giving the flow of data between the blocks.

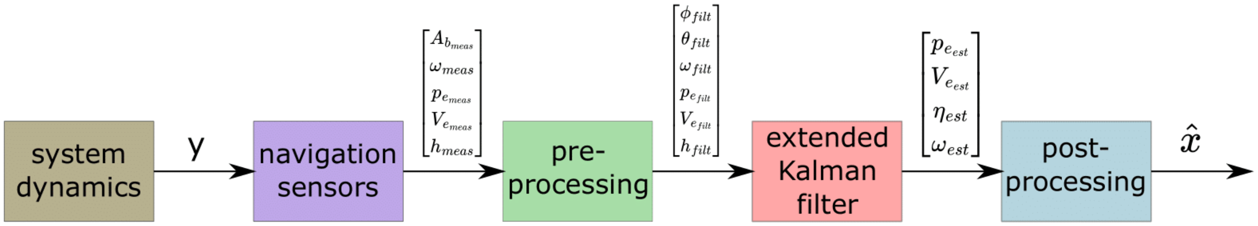


Figure 6.14: Navigation module interaction

The complete state of the drone comprises its position p_e, velocity V_e, attitude η and angular velocity ω. These are known as the navigation state. The sensors used to measure these quantities are called navigation sensors. The navigation sensors used on the drone are described in section 6.1 and consist of an Inertial Measurement Unit (IMU), a GPS receiver and a barometer.

6.4.1 Pre-processing

Before performing the sensor fusion, the sensor measurements are processed to filter out the high-frequency noise and already obtain an estimate of the roll and pitch angles. The architecture of the pre-processing submodule is shown in Figure 6.15, showing the inputs and outputs and different components within the submodule. The inputs are the measured proper acceleration $A_{b_{meas}}$, which is the acceleration of a body in its own instantaneous rest frame. Furthermore, measured velocity $V_{e_{meas}}$ and position $p_{e_{meas}}$, measured altitude h_{meas} and measured angular velocity ω_{meas} are also inputs. These are processed into the outputs which are the filtered roll ϕ_{filt} and pitch θ_{filt} angles, filtered position $p_{e_{filt}}$ and velocity $v_{e_{filt}}$ and the filtered angular velocity ω_{filt} and altitude h_{filt}.

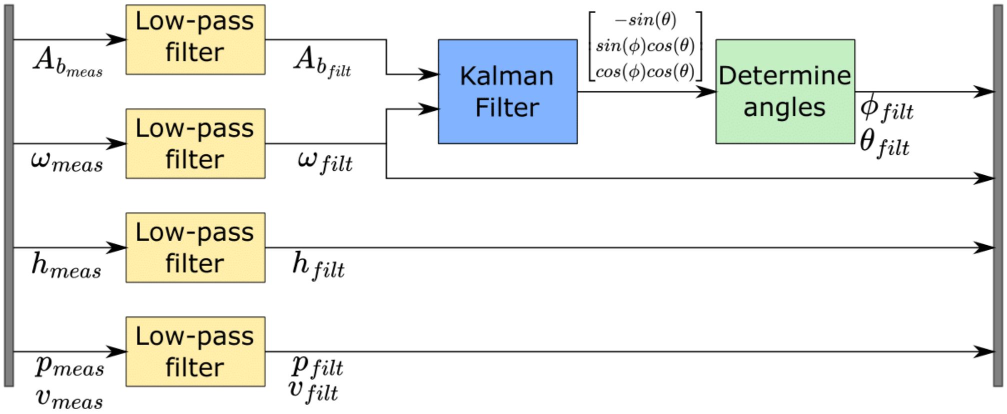


Figure 6.15: Diagram of pre-processing of sensor measurements

The Kalman filter block is used to estimate the roll and pitch angles from the IMU sensor measurements, as described in [31, Sec. 9.1.4]. Kalman filtering is a state estimation technique which optimally fuses the measurement data and model knowledge to obtain a mean estimate value of the state. The dynamic system under consideration is a linear discrete-time system, as given in eq. (6.20).

$$\begin{aligned} x_k &= Ax_{k-1} + Bu_{k-1} + w_{k-1} \\ y_k &= Cx_k + v_k \end{aligned} \tag{6.20}$$

The set of terms $\{w_k\}$ and $\{v_k\}$, with $k \in \mathbf{N}$, are the process noise and measurement noise, these indicate the uncertainty in the model and in the sensor measurements. They are assumed to be white, zero-mean, uncorrelated noise with known covariance matrices Q_k and R_k, respectively.

The Kalman filter obtains its state estimate through a sequence of steps. It is initialized with an estimate of the state as well as a covariance matrix reflecting the uncertainty in this initial state estimate. At each time step, it uses the system model to make a prediction of the current state based on the previous estimate. This prediction is the *a priori* state estimate. This estimate is then corrected using the measurement data to obtain the *a posteriori* state estimate. At each time step, also the covariance matrix of the state estimate is updated. An extensive theoretical description of the Kalman filter algorithm is given in [59].

In order to estimate the roll an pitch angles using a Kalman filter, both a state and measurement equation are required. The process state is set to be

$$\boldsymbol{x_k} = \begin{bmatrix} -sin(\theta) \\ sin(\phi)cos(\theta) \\ cos(\phi)cos(\theta) \end{bmatrix} \tag{6.21}$$

Then, the process model is given by eq. (6.22), where I is the identity matrix and T is the time step.

$$\boldsymbol{x_k} = \left(I + T \begin{bmatrix} 0 & r_{k-1} & -q_{k-1} \\ -r_{k-1} & 0 & p_{k-1} \\ q_{k-1} & -p_{k-1} & 0 \end{bmatrix} \right) \boldsymbol{x_{k-1}} + \boldsymbol{w_k} \tag{6.22}$$

For the measurement model, it is useful to consider the multicopter aerodynamic drag model for a drone since the specific force is directly measured by accelerometers, giving the measurement model for the Kalman filter [31, Sec. 6.2.2]. It is given by

$$\begin{bmatrix} A_{b_x} \\ A_{b_y} \end{bmatrix} = \begin{bmatrix} \dot{V}_{b_x} + gsin(\theta) \\ \dot{V}_{b_y} - gcos(\theta)sin(\phi) \end{bmatrix} = \begin{bmatrix} -\frac{k_{drag}}{m} V_{b_x} \\ -\frac{k_{drag}}{m} V_{b_y} \end{bmatrix} \tag{6.23}$$

The Laplace transform of V_{b_x} and V_{y_b} are expressed as

$$\begin{aligned} V_{b_x} &= -\frac{g}{s + \frac{k_{drag}}{m}} (sin(\theta))(s) \\ V_{b_y} &= \frac{g}{s + \frac{k_{drag}}{m}} (cos(\theta)sin(\phi))(s) \end{aligned} \tag{6.24}$$

while the Laplace transforms of A_{b_x} and A_{b_y} are given by

$$\begin{aligned} A_{b_x} &\approx gH(s)(sin(\theta))(s) \\ A_{b_y} &\approx -gH(s)(cos(\theta)sin(\phi))(s) \end{aligned} \tag{6.25}$$

where $H(s) = \frac{k_{drag}/m}{s + k_{drag}/m}$ is a low-pass filter with $H(0) = 1$. Hence, the low-frequency specific forces along the x- and y-axis in the body frame are

$$\begin{aligned} A_{b_x} &\approx gsin(\theta) \\ A_{b_y} &\approx -gcos(\theta)sin(\phi) \end{aligned} \tag{6.26}$$

In this analysis, it is assumed that the translational and rotational acceleration are zero. Hence, at high accelerations, the measurements contain more uncertainty. For this reason, the measurement noise covariance matrix R_k is set to be proportional to the magnitude of the measured acceleration; higher accelerations means a higher uncertainty in the measurements. The measurement model of the accelerometer is then given by

$$\boldsymbol{y} = -g \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \boldsymbol{x} + \boldsymbol{v_k} = \begin{bmatrix} gsin(\theta) \\ -gsin(\phi)cos(\theta) \end{bmatrix} + \boldsymbol{v_k} \approx \begin{bmatrix} A_{b_x} \\ A_{b_y} \end{bmatrix} \tag{6.27}$$

The sensor measurement $\boldsymbol{y}$ consists of the accelerations A_{b_x} and A_{b_y} which are measured by the accelerometer. The angular velocities $\begin{bmatrix} p & q & r \end{bmatrix}$ in the state matrix are obtained from the filtered gyrometer measurements. As a last step, the roll and pitch angles are determined from the state estimate through $\theta_{filt} = arcsin(-x_1)$ and $\phi_{filt} = arctan(x_2/x_3)$.

The low-pass filter blocks are implemented to obtain the low-frequency specific forces from the accelerometer and to filter out the high frequency noise from the other signals before processing them further. The filters are implemented using a build-in Simulink® block.

6.4.2 Extended Kalman Filter

An extended Kalman filter (EKF) is an extension to the conventional Kalman filter to work with nonlinear system models. This is achieved by linearising the nonlinear system at each time step and using the conventional Kalman filter algorithm. However, in this scenario, the state estimation is no longer optimal due to the linearisation errors that are inherent to EKF.

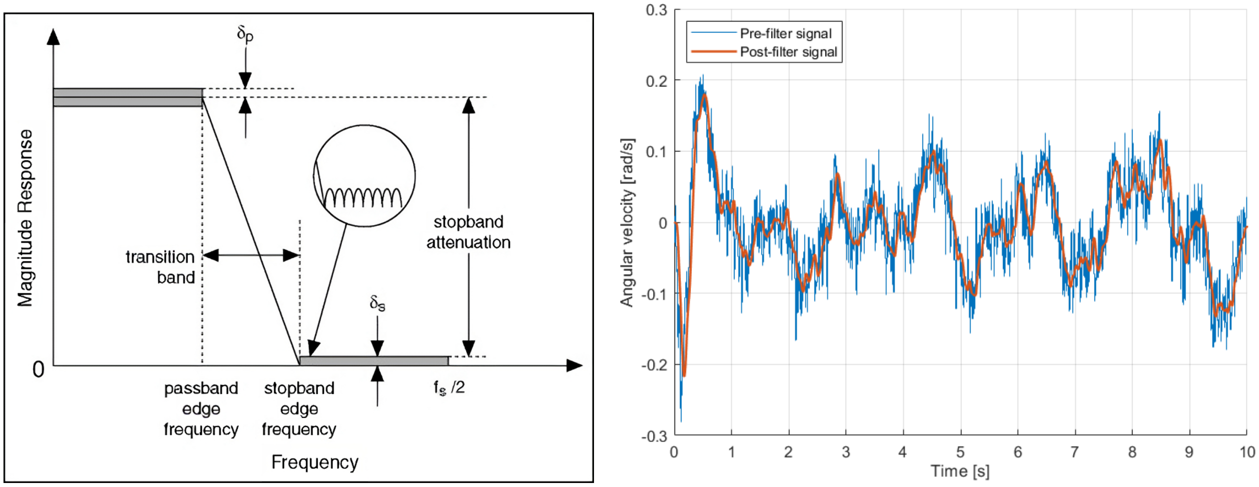
The extended Kalman filter is used to estimate the drone's full state, namely position p_e and velocity V_e in earth-fixed reference frame, the attitude η and angular velocity ω. For this, it uses four measurement inputs and the state transition function. The EKF algorithms are implemented through a build-in Simulink® block. This algorithm is the same as that for a conventional Kalman filter based on the linearisation of the nonlinear system. The configuration of the EKF is given in Table 6.12.

Table 6.12: Extended Kalman filter parameter configuration

	Implementation	**Equation**
State transition function	Equations of translational and rotational motion	$\begin{bmatrix}\dot{p}_e \\ \dot{V}_e \\ \dot{\eta} \\ \dot{\omega}\end{bmatrix} = \begin{bmatrix} V_e \\ DCM_{eb}\frac{F_b}{m} + g \\ \mathbf{T}_{Ib}\omega \\ I^{-1}\left(M - \omega \times I\omega\right)\end{bmatrix}$
Measurement 1	Kalman filter estimate (from pre-processing)	$\boldsymbol{y} = \left[\phi\ \theta\right]^T$
Measurement 2	Gyrometer	$\boldsymbol{y} = \left[p\ q\ r\right]^T$
Measurement 3	GPS	$\boldsymbol{y} = \left[x_e\ y_e\ z_e\ V_{x_e}\ V_{y_e}\ V_{z_e}\right]^T$
Measurement 4	Barometer	$\boldsymbol{y} = z_e$

6.4.3 Post-processing

The post-processing of the drone's state estimate is performed to further smooth out the signals. This is important since any noise which remains in the signal is amplified when passing through the PID controllers. Hence, in the post-processing submodule, each signal is passed through a low-pass filter. These low-pass filter are configured to have a passband edge frequency of 5 [Hz], stopband frequency of 100 [Hz], and minimum stopband attenuation of 50 [dB]. These properties are visualised on a magnitude plot in Figure 6.16a [60]. The filters are run at a sample rate of 1000 Hz. A comparison between a signal before passing through the filter and after is shown in Figure 6.16b. In this case, the roll rate p is selected as an example. As can be seen, the signal is smoothened without introducing a delay between the two signals.



(a) Low-pass filter properties [60]

(b) Post-processing filtering of state signal

Figure 6.16: Filter properties of post-processing submodule

6.4.4 Full navigation module

The state estimation performance of the full navigation module is evaluated in this section. The position estimation is shown in Figure 6.17. As can be seen, the estimation closely matches the true position of the drone closely. This is due to the fact that the GPS receiver gives us direct access to the position with an accuracy of 1 [m].

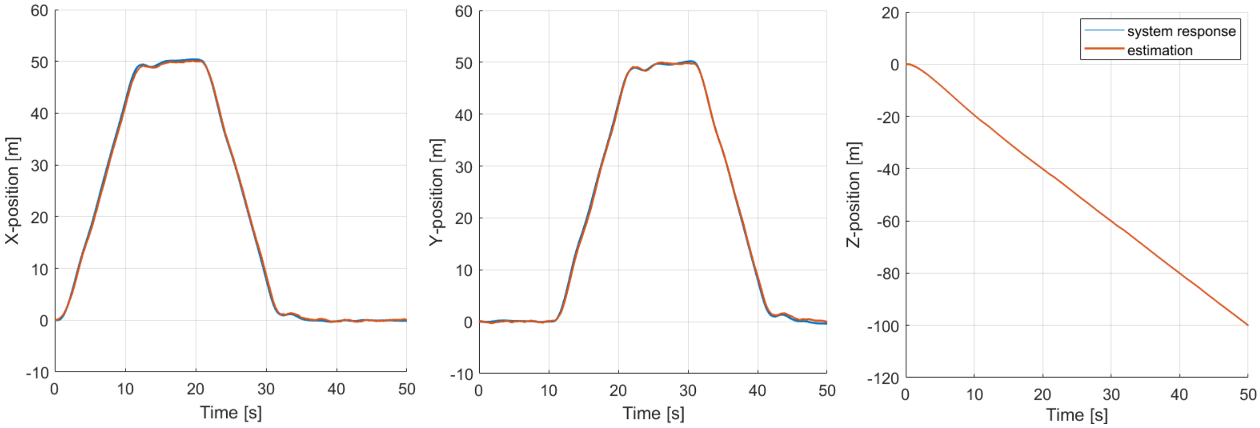


Figure 6.17: Position estimation

The velocity estimation is shown in Figure 6.18. The estimate also closely matches the true drone's velocity due to direct access to velocity data through the GPS receiver. This provides an accuracy of 0.1 [m/s].

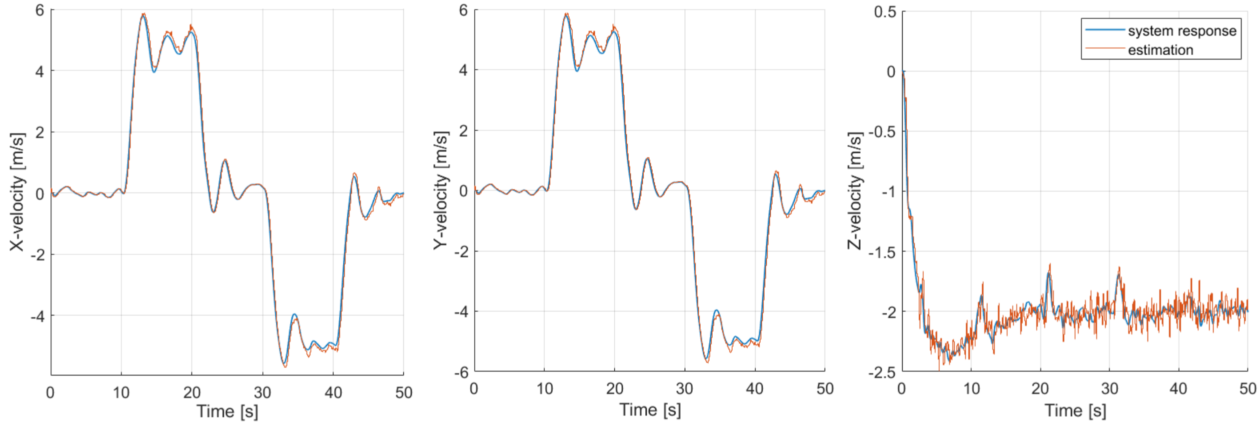


Figure 6.18: Velocity estimation

The attitude estimation is shown Figure 6.19. The estimation of the attitude is less accurate since none of the sensors directly measure the drone's attitude. The Kalman filter in the pre-processing module is capable of estimating the low-frequency roll and pitch angles, however fast angle changes are harder for the Kalman filter to estimate. This, however, is not an issue since the drone is still capable of tracking the position reference, which is the main goal of the control system.

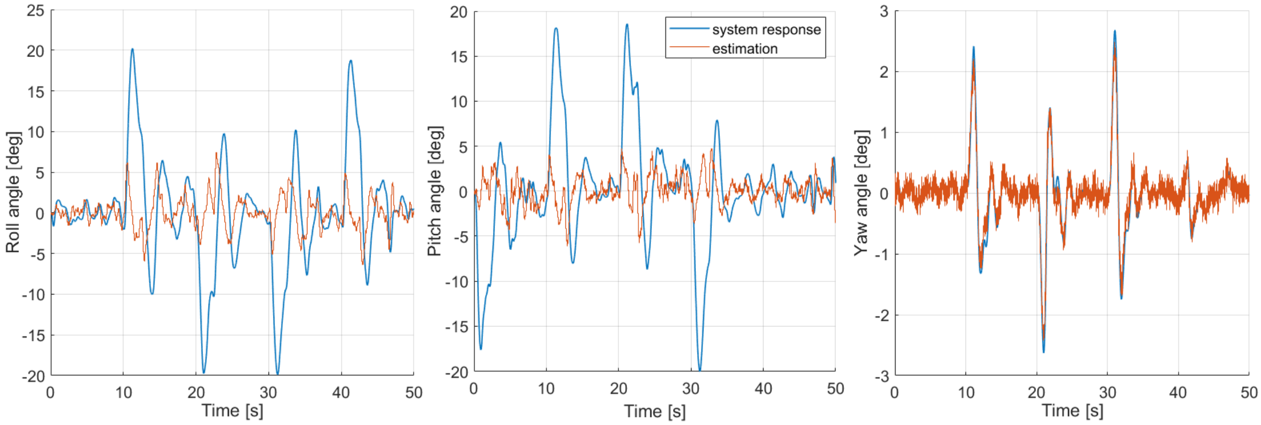


Figure 6.19: Attitude estimation

The angular velocity estimation is shown in Figure 6.20. As can be seen, the angular velocity estimate closely follows the actual angular velocity. This is due to the gyrometer giving direct access to these states.

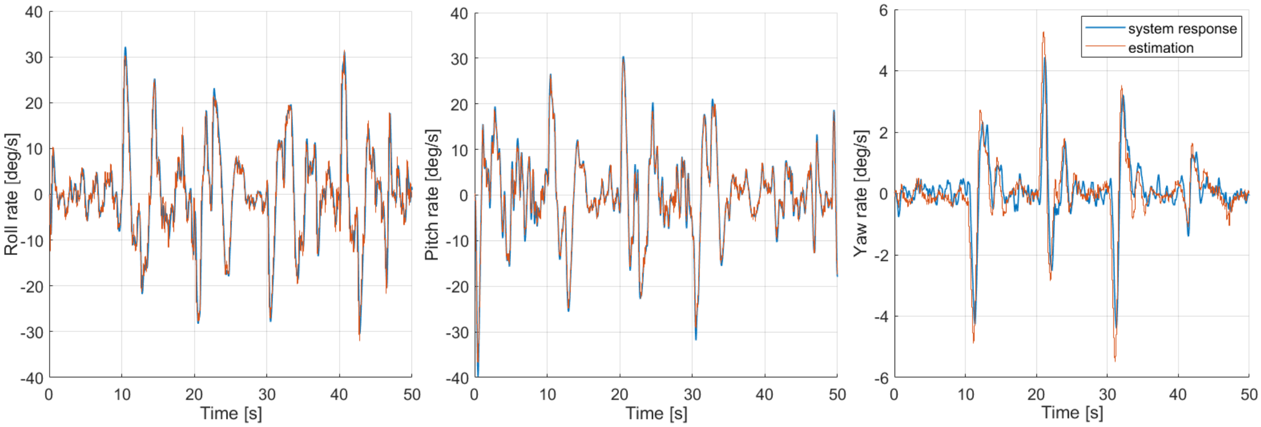


Figure 6.20: Angular velocity estimation

6.5 Guidance

Author: Michael and Jip

The design of the guidance logic of the drone was broken down into 3 major parts, namely Mapping, Motion Planning, and waypoint Following. The guidance logic is the last main component which compliments the navigation and control logic previously detailed in chapter 6. Together with the guidance, the drone model will be complete and can be used to simulate all aspects of a typical wind turbine inspection.

6.5.1 Mapping and Motion Planning

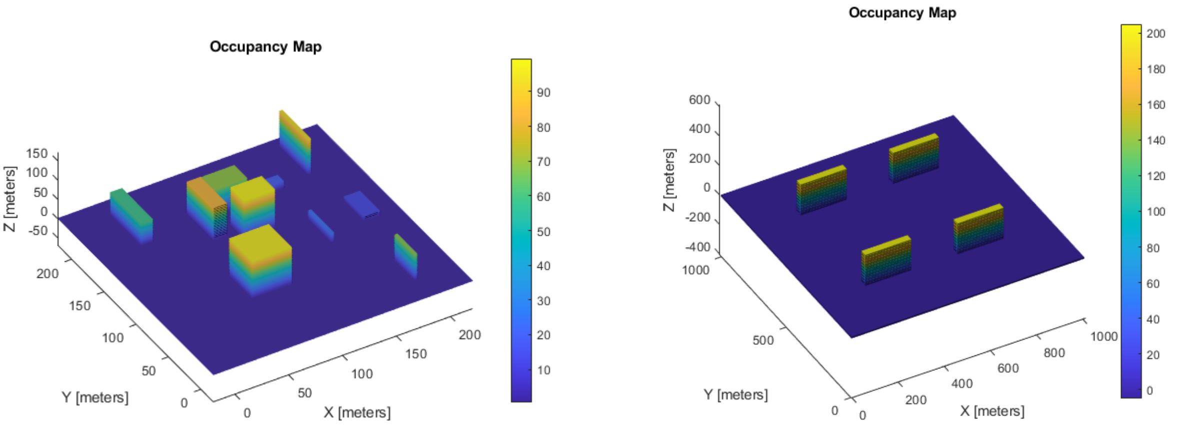
For navigation and positioning, a 3D map must be created of the region for the drone to traverse. Photogrammetry and LiDAR are the 2 most commonly used methods for drones, which generate 3D maps for guidance algorithms to be used. The focus in mapping regions for UAV drones to use is on capturing the 3D location and shape of the obstacles that are present in the region. UAV drones themselves are commonly used to map 3D regions using techniques such as simultaneous localization and mapping (SLAM), as it becomes very crucial for drones traversing high-density regions such as urban environments or small closed spaces. Mapping is the first component of the guidance logic, followed by motion planning.

Mapping

For the purpose of wind turbine inspections, the team's focus lies in possessing the information about the wind turbine's shape and its location, as this will allow us to generate a path around the wind turbine blades for inspection. As offshore wind turbines are however spread out very sparsely over regions tens of kilometres in length, mapping an offshore wind farm with a UAV drone becomes unnecessary and unfeasible. This is because it would result in a very sparse 3D map with no obstacles other than the wind turbines. Mapping the offshore wind turbine would also require significant additional logistic effort with little added value.

As a more effective approach, maps of the wind turbines are instead generated beforehand with the information provided by the client. Knowing the exact model, dimensions and locations of the wind turbines a 3D map can be generated.

Accurate maps of offshore wind farms can therefore be stored within MATLAB® as a 3D occupancy map. In this format, the 3D map is stored as a large matrix of equally spaced nodes in a 3D space, where each node is assigned a probabilistic value from 0 to 1, with each value indicating the probability of the node being occupied by an obstacle. The solid shape of the wind turbine can therefore be represented as a set of occupied nodes with value 1 in 3D space. Below is an example of how different obstacles can be represented in a 3D occupancy map in Figure 6.21a. Additionally, a rough representation of a cluster of 4 wind turbines has also been generated as shown in Figure 6.21b. Whereas typical uses of occupancy maps are for high-density regions of areas less than $200m^2$ at a resolution of 1 node per meter, a larger region can also be generated although at a larger computational expense. The distances between wind turbines in the 3D map have been limited in order to ensure a reasonable computation time for generating the map.



(a) Example of a 3D occupancy representing a random set of obstacles in a $200m^2$ map.

(b) Rough 3D occupancy map representing a cluster of 4 wind turbines in a $1000m^2$ map

Figure 6.21: 3D occupancy maps

Occupancy maps were chosen to store the 3D maps due to their efficient method of storing the 3D matrix as an octree structure. It is more efficient compared to storing the matrix as it subdivides the 3D region into 8 parts recursively to store each node and probability within the tree. Consequently, multiple neighbouring nodes with the same probability can be easily pruned from the tree where necessary to increase space efficiency.

Space efficiency becomes especially crucial once developing the motion planning algorithm as it must retrieve the probability value at each one of the nodes while searching for a valid path. The compatibility of 3D occupancy maps with motion planning algorithms within MATLAB® is another reason why occupancy maps are superior to other storage structures.

Motion planning

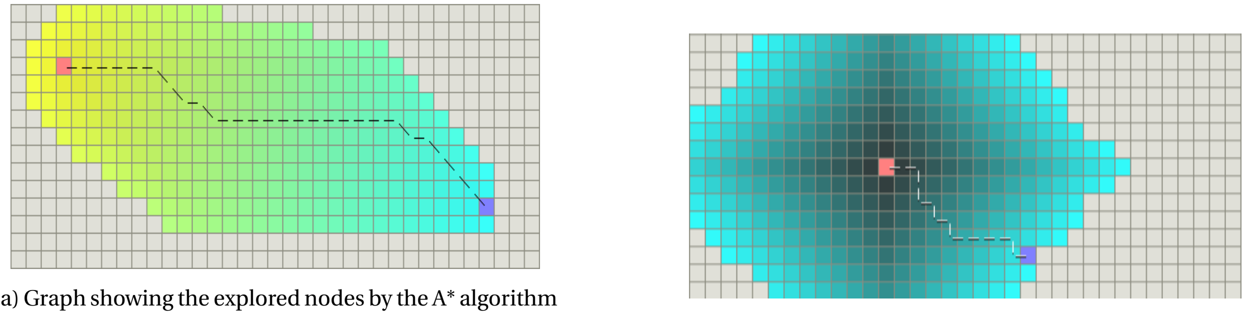
Motion planning entails creating a specific trajectory for the drone to follow in order to move it from point A to point B. This trajectory is generated as a sequence of states known as waypoints that specify the position and orientation of the drone at several instances along the path. With these generated waypoints the drone has a reference location and orientation that it must follow throughout the trajectory to get to the end goal. Several algorithms exist in literature that are able to generate these waypoints from point A to point B while avoiding obstacles in a given 3D occupancy map. The most prominent ones used for UAV drones are A* and the Rapidly-exploring Random Tree (RRT) algorithm, each with its own special variations that seek to improve certain performance parameters of the algorithm. As these are the two main algorithms used in path planning for UAV drones, a comparison of their general performance and characteristics will be performed to determine the most suitable for TurbEye's motion planning.

A* Algorithm

The A* algorithm is the most common path-planning algorithm, which is known as a node-based optimal algorithm as it visits nodes in a map sequentially. It can guarantee the shortest path between a given start and end goal. Neighbouring nodes around the start node are visited based on a cost function which is used to score which neighbouring nodes are the best to visit next in the iteration. Once the cost function has been evaluated for all neighbouring nodes, the neighbouring node with the lowest cost will be added to the path and therefore be visited in the next iteration. This total cost function is defined as the summation of two cost functions which take a node as input as shown in Equation 6.28.

$$f(n) = g(n) + h(n) \tag{6.28}$$

The first function $g(n)$ calculates the distance of the current neighbouring node to the start node, whereas the second function $h(n)$ is a heuristic cost function that estimates the distance from the current neighbouring node to the end node. By evaluating this cost function $f(n)$ the algorithm aims to prioritize selecting nodes that are in the area between the start and end node as opposed to visiting every neighbouring node. Several heuristic cost functions exist which calculate the distance to the goal node differently. They can be used to tune the behaviour of the algorithm and perform a trade-off between the quality of the final solution and the computation time. In Figure 6.22a shown below, it demonstrates how the A* algorithm explores specific neighbouring nodes compared to simply exploring all neighbours as in done by another algorithm shown in Figure 6.22b. Exploring fewer neighbouring nodes each iteration decreases the required computation time. The latter algorithm is Dijkstra's algorithm [61] which is a widely known path-finding algorithm. The blue colour represents the greater distance to the start node, whereas the yellow represents a high value of the heuristic function, which scales with the distance to the goal node. [62]



(a) Graph showing the explored nodes by the A* algorithm [62].

(b) Explored nodes by Dijkstra's Algorithm.[61]

Figure 6.22: Path finding algorithms

RRT Algorithm

The other prominent path planner is the Rapidly Exploring Random Tree (RRT) which is a sampling-based algorithm. The algorithm tries to find a path by generating an expanding tree of connected nodes from randomly sampled points in the map. It is initialized by adding the start and end goal as nodes into the tree. In each iteration, the algorithm randomly samples a point in the map. If there is no obstacle at that location, a new node is made. The algorithm then calculates which node in the current tree is the closest to this new node. If the new node can be connected to the closest node by a link without crossing an obstacle or exceeding the maximum link length, then the node and link are added to the total tree. This process is repeated until the tree expands enough to reach the goal node and a path can be created from the sequence of nodes and links by tracing back through the tree. The RRT algorithm has multiple hyperparameters that can be used to tune how the tree expands, such as the maximum link length, maximum iterations, and sampling strategy, making RRT highly adaptable. This however could require some hyperparameter tuning to extract optimal performance. An example of RRT is shown below in Figure 6.23, where the green lines connect the randomly sampled nodes in the space to form the tree.

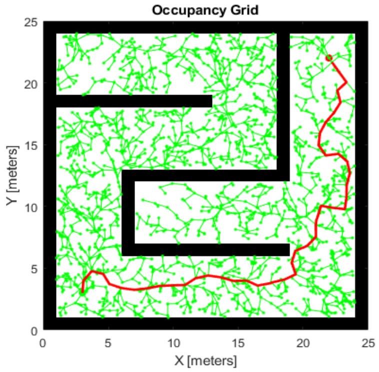


Figure 6.23: Generated tree by the RRT algorithm

Comparison of Motion Planning Algorithms

Additional variations of the A* and RRT algorithms exist which seek to enhance certain performance parameters. In order to determine which algorithm is most feasible to use, a comparison can be performed as shown in Table 6.13.

Table 6.13: Table comparing the performance of different path-finding algorithms.

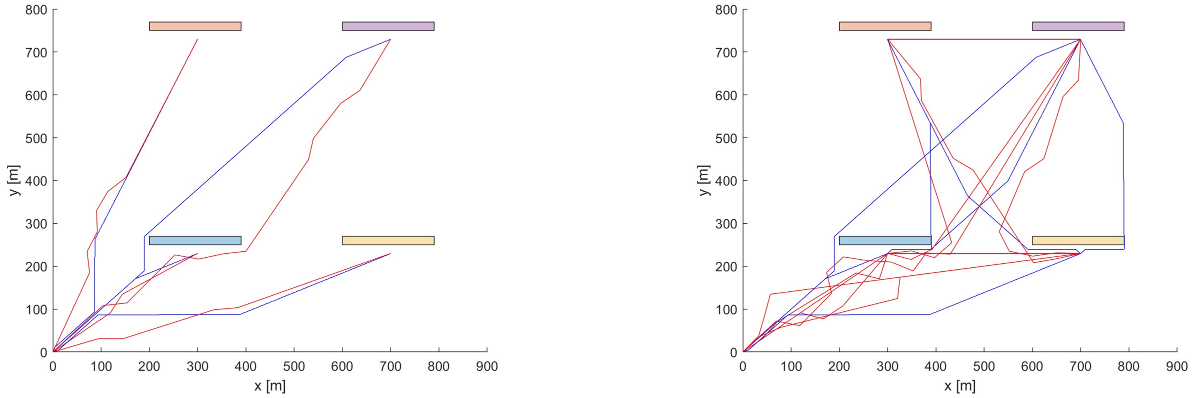
Name	**Description**	**Advantages**	**Disadvantages**
A*	Classic A* algorithm as described in Figure 6.5.1	Short computational time. Simple to implement. Behaviour can be customized using heuristic cost function $h(n)$. Can be modified to guarantee the shortest path.	Performance depends heavily on the heuristic cost function $h(n)$. The path does not consider vehicle dynamics and generates very sharp turns.
Hybrid A*	Similar to A* but also imposes additional constraints regarding the physical range of motion of the vehicle and generates a smoother path that the vehicle can realistically follow such as by avoiding 90° turns.	Short computational time. The path generated is realistic and smooth. The path is easier to follow by vehicle.	Path is longer compared to A*, computation time will depend on the strictness of imposed vehicle constraints. Does not guarantee the shortest solution
RRT	Classic RRT Algorithm as described in Figure 6.5.1.	Short computation time. Simple implementation, performance is very customizable. The algorithm also performs well in 3D.	Path is very segmented and contains a lot of unnecessary movement. May not converge to a solution. Tree expands uniformly throughout the map instead of expanding in the direction of the goal node.

Table 6.13: Table comparing the performance of different path-finding algorithms.

Name	**Description**	**Advantages**	**Disadvantages**
RRT*	Builds upon RRT by optimizing which node a new node is connected to each iteration, as it chooses a close neighbour with the lowest distance to the start node. It also improves RRT by rerouting the tree once a path is found, which generates a significantly shorter path.	Reasonable computation time. Simple implementation. Takes vehicle dynamics into account. Shorter and smoother final path compared to RRT. Performance is highly customizable	Not guaranteed to find a path. May require many iterations before a solution is found. Path still contains some unnecessary movement or turns. Slightly slower computation time than RRT
RRT*-Smart	Improves upon RRT* by sampling random points around the first generated path. From the first path found, key nodes are taken from the path, usually around obstacle vertices. The sampling strategy is then biased to sampling points in a radius around these key nodes in the next iteration. These additional nodes are then used to smoothen and optimize the path that was initially found.	Path found is short and smooth, respecting the vehicle dynamics. Reasonable computation time. Can reorder tree linkages to prevent unnecessary movement in the path. Works well in 3D. Converges faster to the goal node than RRT*	Slowest variant of RRT in terms of a worst-case scenario. Implementation is fairly complex. Requires many iterations to converge to the absolute shortest path.

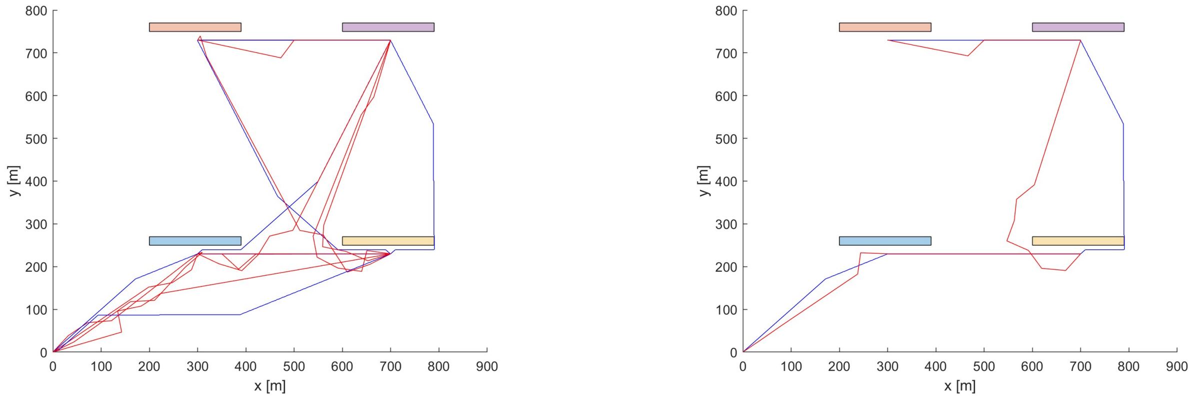
As a result of the performed comparison in Table 6.13, it was decided to focus the team's effort on using the A* and RRT* algorithms for the wind turbine motion planning. A* was chosen for its fast computation time, ability to guarantee the shortest path and ease of implementation. Likewise, RRT* was chosen due to its ease of implementation as well, and its significantly better performance over RRT for relatively little extra effort. Although RRT*-smart could have resulted in even shorter paths, its implementation is quite complex and less documented in the literature, likely requiring more time to research and implement for motion planning. Having chosen these 2 algorithms, a more rigorous quantitative analysis can be performed by implementing both algorithms in the guidance logic.

To identify the practical difference between RRT* and A* in the simulation scenario, all possible trips for the 4 wind turbine cluster map in Figure 6.21b were compared. Meaning for all combinations of wind turbines that can be visited between the start point and at least one wind turbine up to four turbines a path was generated by both algorithms. On average the A* routes are 4.18% shorter than the RRT* routes. For a route visiting all four turbines the path generated using A* was 6.99% shorter.



(a) Paths generated by RRT*(Red) and A*(Blue) algorithms for visiting 1 turbine.

(b) Paths generated by RRT*(Red) and A*(Blue) algorithms for visiting 2 turbines.



(c) Paths generated by RRT*(Red) and A*(Blue) algorithms for visiting 3 turbines.

(d) Paths generated by RRT*(Red) and A*(Blue) algorithms for visiting 4 turbines.

Figure 6.24: Paths generated by RRT*(Red) and A*(Blue) algorithms for visiting 1,2,3 or 4 turbines.

Inspection route

The inspection route for the drone is statically generated based on four parameters. These parameters are: nacelle height, blade radius, blade orientation (angle of closest blade in the positive direction w.r.t the z-axis) and windturbine orientation w.r.t. True North which is assumed to coincide with the y-axis (angle of closest blade in the positive direction w.r.t the y-axis) . Using these parameters a path is generated for the turbine inspection. Each blade gets inspected both from the front and back once, so as to ensure full image coverage of the blade. This is supported and shown by Figure 6.25. An example of a generated inspection path for arbitrary input parameters can be found in Figure 6.26. In this case, a nacelle height of 120m, blade radius of 80m, a blade angle of 30° w.r.t. the z-axis and a 90° angle w.r.t. to True North were used.

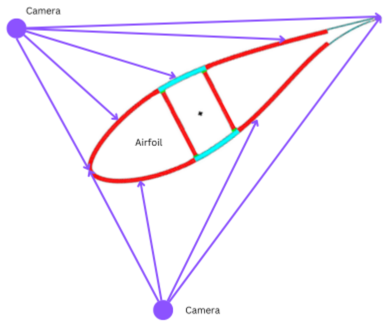


Figure 6.25: Justification for inspection path

Since the drone payload is attached by gimbal, it is assumed the camera can stay in the correct position during the inspection. Small oscillations in position will occur during an inspection due to external factors such as wind or other weather conditions. Internal factors can also cause small path deviations. The gimbal will ensure camera stability and therefore valid inspection data.

6.5.2 Waypoint Following

Once mapping and motion planning have been conducted, the next step is instructing the drone to follow the given trajectory. During operation the drone has two main objectives: going to the turbine that needs to be inspected and performing the inspection. The path to the required turbines is dynamically generated using the A* algorithm as discussed in Figure 6.5.1, whereas the inspection path is statically generated given the pose of the wind turbine. In order to implement the waypoint a Guidance Stateflow subsystem was implemented within the SIMULINK® environment. The inputs, outputs and main components will be further detailed in this subsection.

Input parameters

Three different types of parameters are associated with each waypoint and get passed to the drone during operation. Mode, Position and Parameters. Mode is the flying mode for the drone. Position are the desired x,y,z coordinates. Parameters are four custom parameters which can be customized based on mission objectives.

Operation Modes

There are seven different flying modes that can be utilised for drone operations. These modes get passed to the drone per waypoint as an integer, the integers in the list coincide with the index in the list below. The eight modes are:

Table 6.14: A table describing the drone operation modes.

ID	Mode	Description
1	Takeoff	The drone will only move in the z-direction to the first waypoint from the takeoff position
2	Waypoint	The drone follows a path of set waypoints.
3	Orbit	The drone will fly in a circle with a given radius.
4	Landing	The drone will move to the desired landing location and return the status that it has landed. And follows waypoints with higher accuracy.
5	Return to Home	The drone returns to its takeoff position and lands.
6	Inspection	A modified version of the waypoint following.
7	Hold	Drone hovers at the given coordinates.
8	Flight termination	All electrical on-board systems shut down.

Position

The position inputs are the x, y and z coordinates of the corresponding waypoint.

Parameters

The parameters are custom additional parameters that can be passed as input. This includes things such as desired yaw, turn radius, number of turns and desired speed.

Drone operation

The drone will take off in mode 1 and land in mode 4. It will operate in mode 2 when moving to the desired turbine and it will operate in mode 6 when performing the inspection. During normal operation the other modes will not be utilised. Mode 6 is a modified version of mode 2, the drone will operate at lower speeds and with higher accuracy in mode 6. In all other aspects both the modes coincide.

These modes work as follows: Based on the current position, the previous waypoint and the next waypoint a look-ahead-point gets generated, along with the desired yaw. A look-ahead point is the x, y, z coordinates of the next waypoint with a small margin added. The look-ahead-point slightly overshoots the coordinates of the waypoint in the opposite direction. This is done to improve the overall smoothness of the operation such that the drone does not slow down before every waypoint. This look-ahead point and yaw get fed as feedback to the control system, which in turn generate a response.

Moreover, the drone system waypoint following system uses three different statuses during operation. Status 0: moving to the desired waypoint, Status 1: waypoint has been reached, moving on to the next waypoint. Status 2: Drone has landed, operation is terminated. To determine if the drone has reached the desired waypoint, the distance between its position and the desired waypoint is determined. If the distance is within the set threshold (5m for mode 2 and 1m for mode 6) the system returns status 1 and it will move on to the next waypoint.

6.5.3 Generated and simulated path comparison

In this section the generated paths for inspection and operation get compared to the simulated paths for inspection and operation. For the operation path two path generation algorithms get compared. The RRT* path generation gets compared to the A* path generation algorithm.

Inspection path

In Figure 6.26 the generated and simulated inspection paths can be found. The generated path gets followed accurately. The simulated path slightly oscillates in the y-direction, however it is all within the set margins.

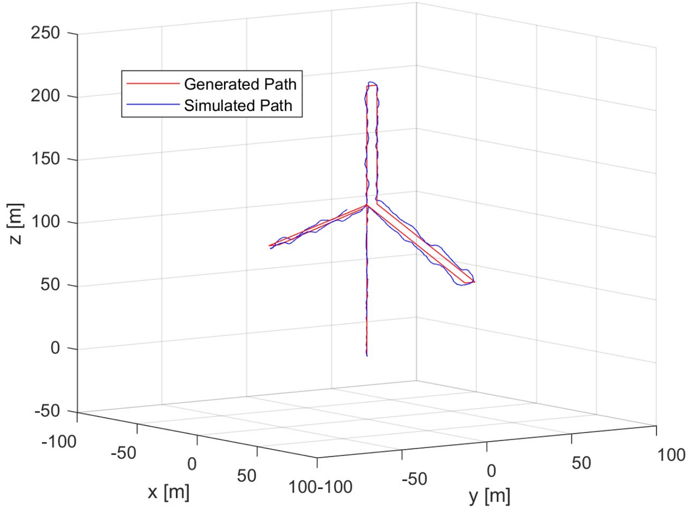
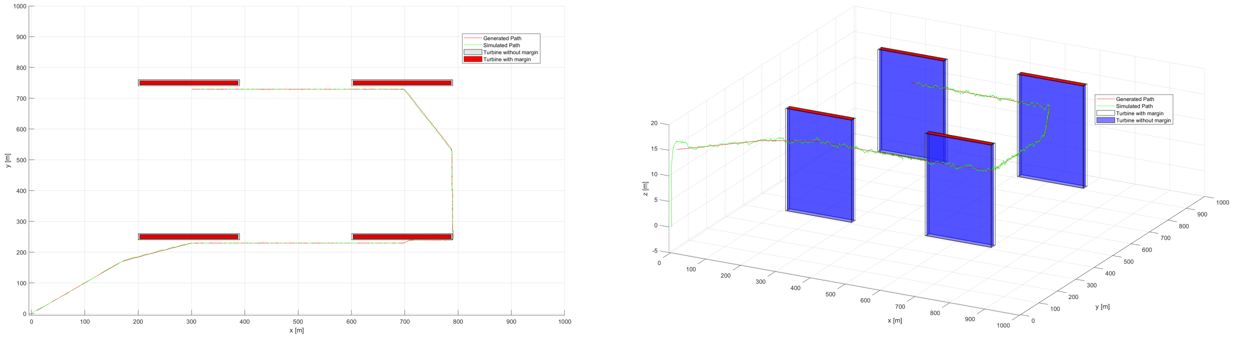


Figure 6.26: Generated path and simulated path for inspection of wind turbine.

Operation path

In Figure 6.27 the generated and simulated operation path can be found. As can be seen the path gets followed very accurately, excluding the overshoot after takeoff, which gets corrected quickly, the deviation from the generated path is small.



(a) Generated and simulated operation path in 2D perspective. (b) Generated and simulated operation path in 3D perspective.

Figure 6.27: Generated and simulated operation paths using A* algorithm.

6.6 Verification and Validation

Author: Mike

This section performs an extensive verification and validation procedure on the guidance, navigation and control algorithms. The procedure is used to develop the algorithms from a simulated model-based components in MATLAB/Simulink® environment to embedded software able to be run on hardware. This methodology is visualized in Figure 6.28. From the requirements specifications, the models are created within the Simulink®environment and each part of the model is traced back to the specific requirements, through which each requirement can also be verified. Subsequently, model coverage verification and modelling standard verification is carried out on the models. From these models, the embedded C-code is generated. This C-code is again traced back to specific parts of the models. On this C-code, code coverage verification and code standards verification is again completed. Lastly, software-in-the-loop (SIL) and processor-in-the-loop (PIL) tests are performed to test the functionality of the generated C-code.

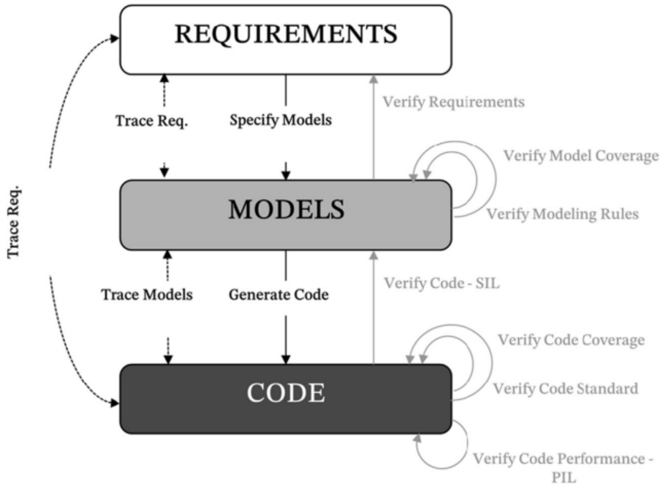


Figure 6.28: GNC software auto coding methodology

6.6.1 Model-In-the-Loop (MIL)

The main activities related to the GNC verification at the MIL level are listed below:

- Unit testing
- Algorithms verification (sensitivity analysis) for control and guidance
- Requirements verification
- Modeling standards verification
- Profiling of the models

Unit Testing

Author: Mike

Unit testing is performed on the pieces of MATLAB® code used in the simulation. Other blocks in the simulation do not require unit testing since these have already been verified by the provider of the add-ons which contain these Simulink®blocks.

In the complete simulation, six user-written MATLAB® function are utilized, which are in the pre-processing and extendend Kalman filter submodule within the navigation module. These MATLAB® code files are used to implement the process models and sensor models mentioned in eq. (6.22) and Table 6.12.

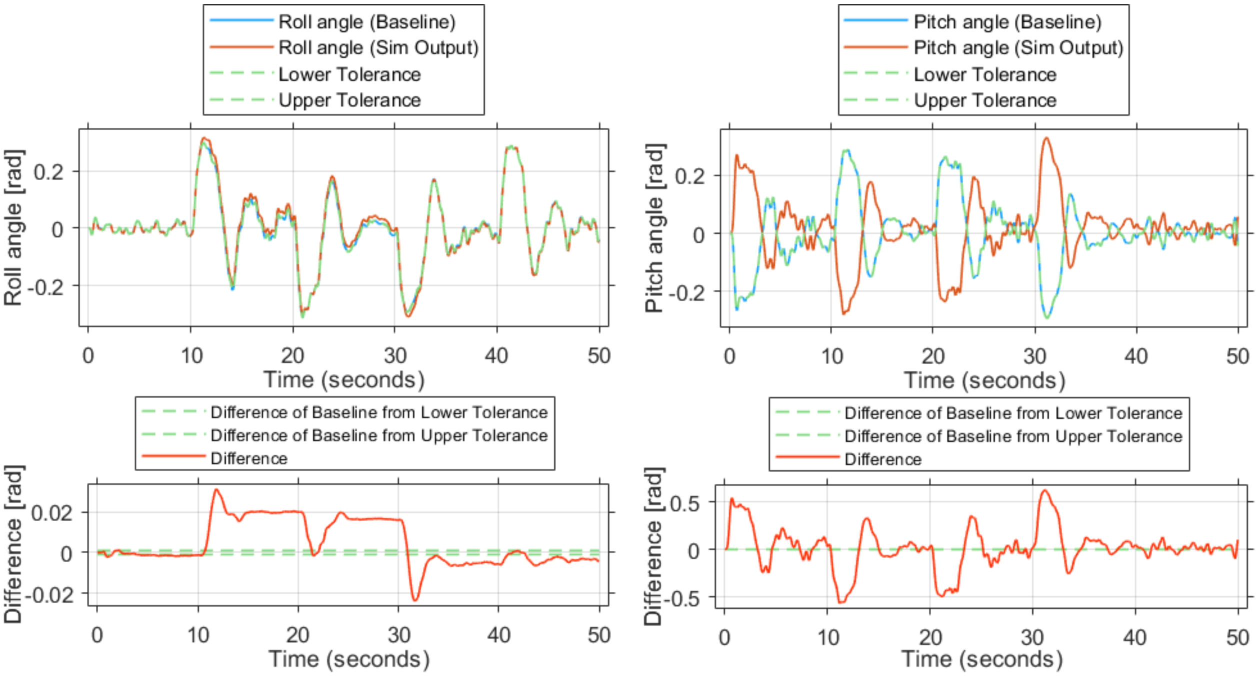
Several methods of unit testing these MATLAB® functions are used. The method used for each function and results are shown in Table 6.15.

Table 6.15: Unit testing of user-written MATLAB® functions

MATLAB function®	**Reference**	**Testing method**	**Results**
Process model	eq. (6.22)	User-written MATLAB® function is tested against baseline data, provided by simulated data.	Initially, results came out negative. After verifying with different sources from literature, a mistake in the original source was identified. Further elaborated on below.
Measurement model 1	Table 6.12	Visually inspected due to the simplicity of the code.	Passed visual check.
Measurement model 2	Table 6.12	Visually inspected due to the simplicity of the code structure.	Passed visual check.
Measurement model 3	Table 6.12	Visually inspected due to the simplicity of the code structure.	Passed visual check.
Measurement model 4	Table 6.12	Visually inspected due to the simplicity of the code structure.	Passed visual check.
State transition	Table 6.12	User-written MATLAB® function is tested against baseline data, provided by a build-in Simulink®function	All output signals pass the baseline test.

The original source used for the process model is [63], in which the process model is given as in eq. (6.29). This function is tested by giving it an initial state and propagating this state using a baseline angular velocity signal. The corresponding baseline roll and pitch angles are then compared to the roll and pitch angles obtained from this propagated state vector. The result for this process model is shown in Figure 6.29. As can be seen, while the roll angles roughly match up with an error of 0.03 [rad]. The pitch angles, however, are off by a phase of 180 [deg]. These results indicate a problem with the original source, hence other sources are checked to verify the process model [31] and [64]. These sources indicate eq. (6.22) to be the correct equation. The result for this updated process model is given in Figure 6.30.

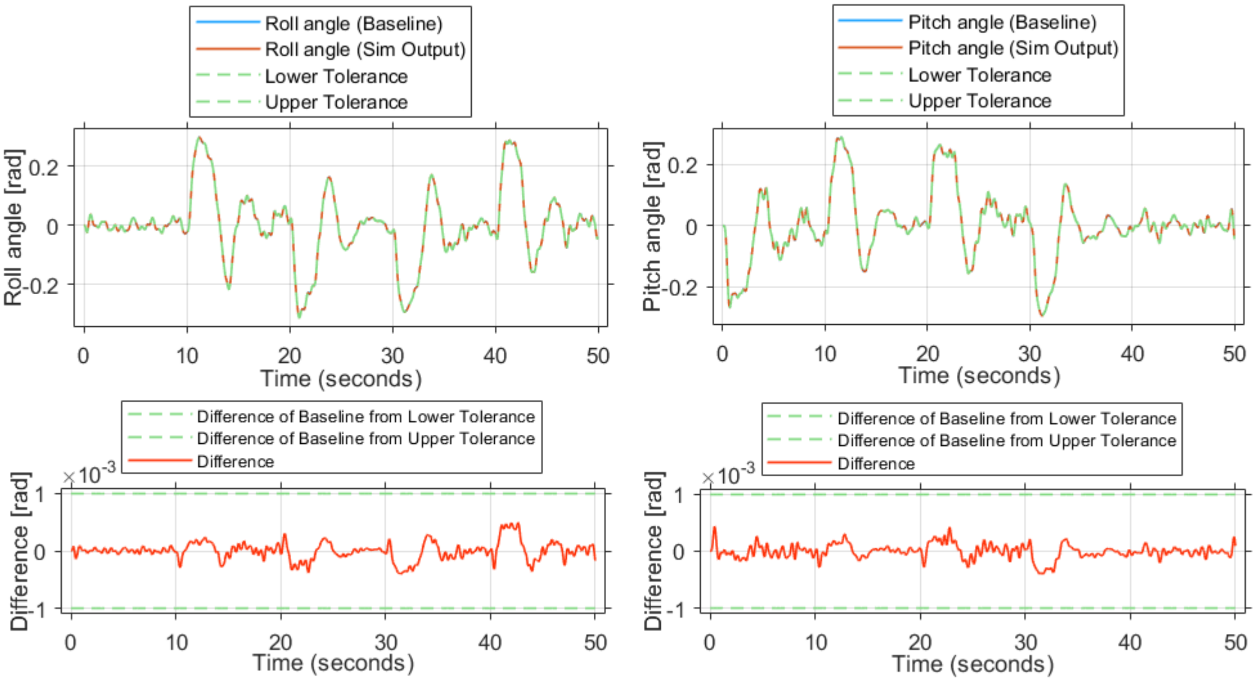
$$\begin{bmatrix} -sin(\theta_k) \\ sin(\phi_k)cos(\theta_k) \\ cos(\phi_k)cos(\theta_k) \end{bmatrix} = \left(I + T \begin{bmatrix} 0 & -r_{k-1} & -q_{k-1} \\ r_{k-1} & 0 & -p_{k-1} \\ -q_{k-1} & p_{k-1} & 0 \end{bmatrix} \right) \begin{bmatrix} -sin(\theta_{k-1}) \\ sin(\phi_{k-1})cos(\theta_{k-1}) \\ cos(\phi_{k-1})cos(\theta_{k-1}) \end{bmatrix} \tag{6.29}$$



(a) Baseline test for roll angle

(b) Baseline test for pitch angle

Figure 6.29: Unit testing result of erroneous process model function



(a) Baseline test for roll angle

(b) Baseline test for pitch angle

Figure 6.30: Unit testing result of correct process model function

Control Sensitivity Analysis

Author: Radu

Algorithms verification is concerned with the functional verification of the GNC algorithms via specific tests and sensitivity analysis. The GNC algorithms are already verified on a first level through simulating the model response in a simulation environment. An important aspect to consider in modelling a physical system is that the parameters are obtained through estimation from measurements. Therefore, the parameters used in the model possibly do not match with the physical reality. To test the robustness of the GNC system against the uncertainty in these parameters, a sensitivity analysis is performed. As the guidance logic is relatively independent of the navigation and control, the sensitivity analysis can be broken down into 2 parts. Firstly for the navigation and control logic and afterwards for the guidance logic. For navigation and control the system parameters for which a sensitivity analysis is performed are shown in Table 6.16. One specific run indicates that the uncertainty in the parameters is considered at the same time. They are considered together since they are related; e.g.: if the mass changes, then so do the inertia values. The probability distribution indicates how uncertainty is introduced into the parameters and the last columns give the parameters used for the probability distribution and the number of samples. The actual parameter samples can be visualized in Appendix B. For each evaluation, the root mean square error (RMSE) between the system response and the reference is determined. An analysis is then performed to determine how the performance is influenced by the parameters.

Table 6.16: System parameters evaluated for uncertainty

Run	Model parameters	Probability distribution	Distribution parameters	Samples
1	$m[kg]$	Normal distribution	$\mu = 12.0248$; $\sigma = 0.60124$	450
1	$I_{xx}[kgm^2]$	Normal distribution	$\mu = 1.12755$; $\sigma = 0.05638$	450
1	$I_{yy}[kgm^2]$	Normal distribution	$\mu = 0.61461$; $\sigma = 0.03073$	450
1	$I_{zz}[kgm^2]$	Normal distribution	$\mu = 1.51166$; $\sigma = 0.07558$	450
2	$k_F[m^4rad^{-2}]$	Normal distribution	$\mu = 3.72575 \times 10^{-4}$; $\sigma = 3.72575 \times 10^{-5}$	250
2	$k_M[m^5rad^{-2}]$	Normal distribution	$\mu = 1.14548 \times 10^{-5}$; $\sigma = 1.14548 \times 10^{-6}$	250
2	$k_c[-]$	Normal distribution	$\mu = 0.6407$; $\sigma = 0.06407$	250
2	$d_p[m]$	Normal distribution	$\mu = 0.8175$; $\sigma = 0.08175$	250
3	$\tau_m[s^{-1}]$	Uniform distribution	[5; 50]	100
4	$C_D[-]$	Normal distribution	$\mu = 0.8$; $\sigma = 0.12$	250
4	$S_x[m^2]$	Uniform distribution	[0.0735; 0.2205]	250
4	$S_y[m^2]$	Uniform distribution	[0.0425; 0.1275]	250
4	$S_z[m^2]$	Uniform distribution	[0.1047; 0.3142]	250

Run 1 The samples used for the first run are shown in Figure B.2. The plots on the diagonal indicate that distribution of the samples. The plots below the diagonal indicate the actual values of the parameters for every sampled data point. The results of this sensitivity analysis are displayed in Figure 6.31; the y axis represents the root-mean-square error (RMSE) of the tracking w.r.t each axis resulted from running the simulation with the parameter values taken as the sampled data points. These plotted results include 448 out of the 450 generated samples. The remaining data points were outliers and are listed in Table 6.17.

The results clearly show the strong negative correlation between mass and RMSE of tracking w.r.t. x and y position, as well as the strong positive correlation with RMSE of tracking w.r.t. z position. Outlier 2 further indicates that decreasing the mass too much will make the system unable to follow the reference. All in all, the accuracy in tracking of position on all axis varies negligibly with variations in mass, as long as the variations remain in the stability envelope. The lower bound of this stability envelope had been demonstrated to approximately be $m \geq 10.4788[kg]$. I_{xx}, I_{yy} and I_{zz} do not show very strong correlations with RMSE of tracking w.r.t. any axis, except for the strong negative correlation between I_{xx} and RMSE w.r.t. z position. Outlier 1 identified the lower bound for the stability envelope of I_{yy} to approximately be $I_{yy} \geq 0.5369[kgm^2]$. The other bounds of the stability envelopes had been dictated by the largest tested values that showed stability and all are tabulated in Table 6.18.

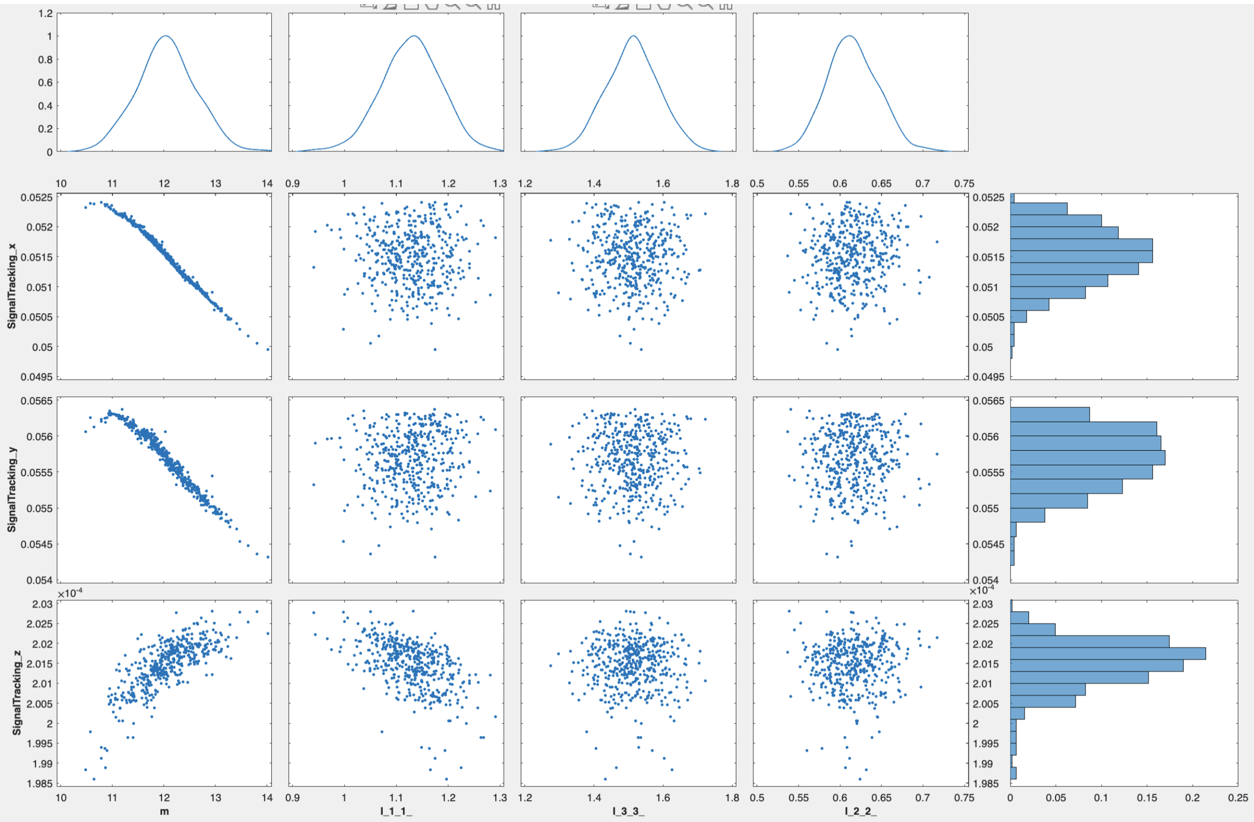


Figure 6.31: Sensitivity analysis results for m, I_{xx}, I_{yy} and I_{zz}, with y axis showing RMSE of the tracking w.r.t each axis resulted from running the simulation with the parameter values represented on x axis

Table 6.17: Outliers for the mass m and inertia I parameters

No.	**m** $[kg]$	$\mathbf{I_{xx}}$ $[kgm^2]$	$\mathbf{I_{yy}}$ $[kgm^2]$	$\mathbf{I_{zz}}$ $[kgm^2]$	$\mathbf{x_{pos}}$ $[m]$	$\mathbf{y_{pos}}$ $[m]$	$\mathbf{z_{pos}}$ $[m]$
1	11.9804	1.1523	0.5005	1.5643	0.0514	0.0555	2.0755×10^{-4}
2	10.3827	1.2308	0.6092	1.5026	5.3952	18.4547	1.0000

Table 6.18: Stability envelopes of m, I_{xx}, I_{yy} and I_{zz}

Parameter	**m** $[kg]$	$\mathbf{I_{xx}}$ $[kgm^2]$	$\mathbf{I_{yy}}$ $[kgm^2]$	$\mathbf{I_{zz}}$ $[kgm^2]$
Lower Bound	10.4788	0.9405	0.5369	1.2754
Upper bound	14.0124	1.2914	0.7173	1.7228

Run 2 The samples used for this run are shown in Figure B.1. Because k_F and k_M represent the thrust and moment coefficients of the propellers they are somewhat related and both have a strong positive correlation with the propeller diameter; e.g., one cannot see a drastic change without the other also being affected, as well as implying a larger or smaller propeller diameter, which in turn creates a change in distance between propellers, d_p. For this reason, a correlation between the 3 parameters was implemented when generating the samples, as can be seen in the in Table 6.19. The results of this sensitivity analysis are displayed in Figure 6.32; the y axis represents the root-mean-square error (RMSE) of the tracking w.r.t each axis resulted from running the simulation with the parameter values taken as the sampled data points. These plotted results include 243 out of the 250 generated samples. The remaining data points were outliers and are listed in Table 6.20.

The results indicate strong positive correlations between k_c and the RMSE of tracking w.r.t. x and y positions, as well as a strong negative correlation between k_c and RMSE w.r.t. z position. Other correlations were not significant for the analysis. The tracking w.r.t. y position is increasingly deprecated as k_c increases. This creates a lot uncertainty in the other axis as well and, for this reason, the team suspects that there appears to be a skewness between the d_p, k_F, and k_M and RMSE of tracking w.r.t. to all axes. Similarly to the previous run, the stability envelopes for the coefficients evaluated in this run are tabulated in Table 6.21.

Table 6.19: Correlation matrix of k_F, k_M, and d_p

-	$\mathbf{k_F}$	$\mathbf{k_M}$	$\mathbf{d_p}$
$\mathbf{k_F}$	1	0.8	0.8
$\mathbf{k_M}$	0.8	1	0.8
$\mathbf{d_p}$	0.8	0.8	1

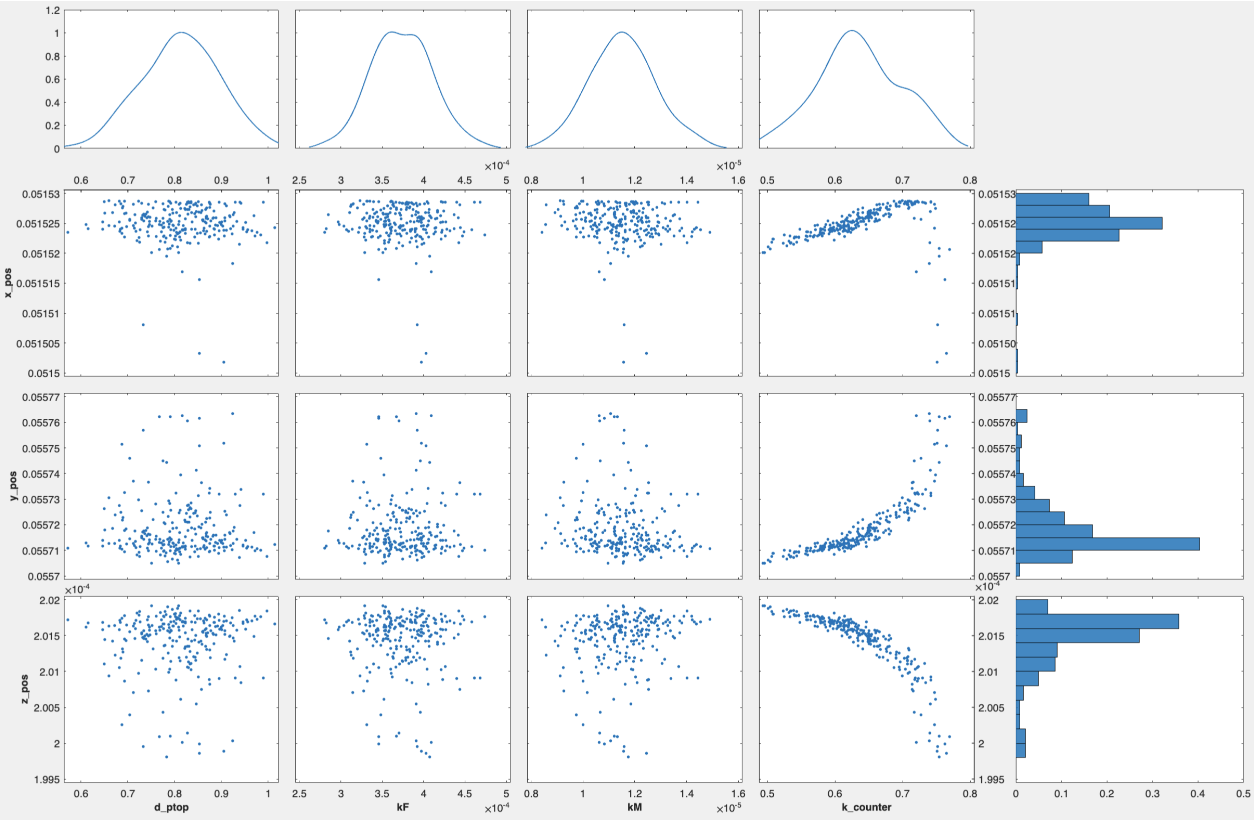


Figure 6.32: Sensitivity analysis results for $d_p, k_F, k_M and k_c$, with y axis showing RMSE of the tracking w.r.t each axis resulted from running the simulation with the parameter values represented on x axis

Table 6.20: Outliers for d_p, k_F, k_M and k_c

No.	$\mathbf{d_p}$ $[m]$	$\mathbf{k_F}$ $[m^4 rad^{-2}]$	$\mathbf{k_M}$ $[m^5 rad^{-2}]$	$\mathbf{k_c}$ $[-]$	$\mathbf{x_{pos}}$ $[m]$	$\mathbf{y_{pos}}$ $[m]$	$\mathbf{z_{pos}}$ $[m]$
1	0.9605	4.5318×10^{-4}	1.4274×10^{-5}	0.8491	0.0517	0.0556	2.0862×10^{-4}
2	0.6987	3.4504×10^{-4}	1.1997×10^{-5}	0.8367	0.0517	0.0557	2.0297×10^{-4}
3	0.7674	3.4951×10^{-4}	1.0727×10^{-4}	0.8334	0.0517	0.0557	2.0481×10^{-4}
4	0.7272	3.5279×10^{-4}	1.1922×10^{-5}	0.8175	0.0516	0.0557	2.0153×10^{-4}
5	0.7816	3.8568×10^{-4}	1.1863×10^{-5}	0.8101	0.0516	0.0557	2.0230×10^{-4}
6	0.7466	3.4249×10^{-4}	1.1359×10^{-5}	0.7910	0.0516	0.0557	1.9966×10^{-4}
7	0.8352	3.9944×10^{-4}	1.1645×10^{-5}	0.7717	0.0516	0.0557	1.9953×10^{-4}

Table 6.21: Stability envelopes of d_p, k_F, k_M and k_c

Parameter	$\mathbf{d_p}$ $[m]$	$\mathbf{k_F}$ $[m^4 rad^{-2}]$	$\mathbf{k_M}$ $[m^5 rad^{-2}]$	$\mathbf{k_c}$ $[-]$
Lower Bound	0.5722	2.8065×10^{-4}	8.4214×10^{-6}	0.4936
Upper bound	1.0136	4.7373×10^{-4}	1.4899×10^{-5}	0.7694

Run 3 The samples used for this run are shown in Figure B.3. The results of this sensitivity analysis are displayed in Figure 6.33; the y axis represents the root-mean-square error (RMSE) of the tracking w.r.t each axis resulted from running the simulation with the parameter values taken as the sampled data points. These plotted results include 95 out of the 100 generated samples. The remaining data points were outliers and are listed in Table 6.22.

The results fully meet the team's expectations as the ability of the system to track the reference is deprecated as τ_m decreases. The RMSE w.r.t. tracking of y position increases rapidly as soon as the value of $\tau_m \approx 10$ is approached. Furthermore, the system becomes unstable for values of $\tau_m \leq 7.4278$. Similarly to the previous runs, the stability envelope for the coefficient evaluated in this run is tabulated in Table 6.23.

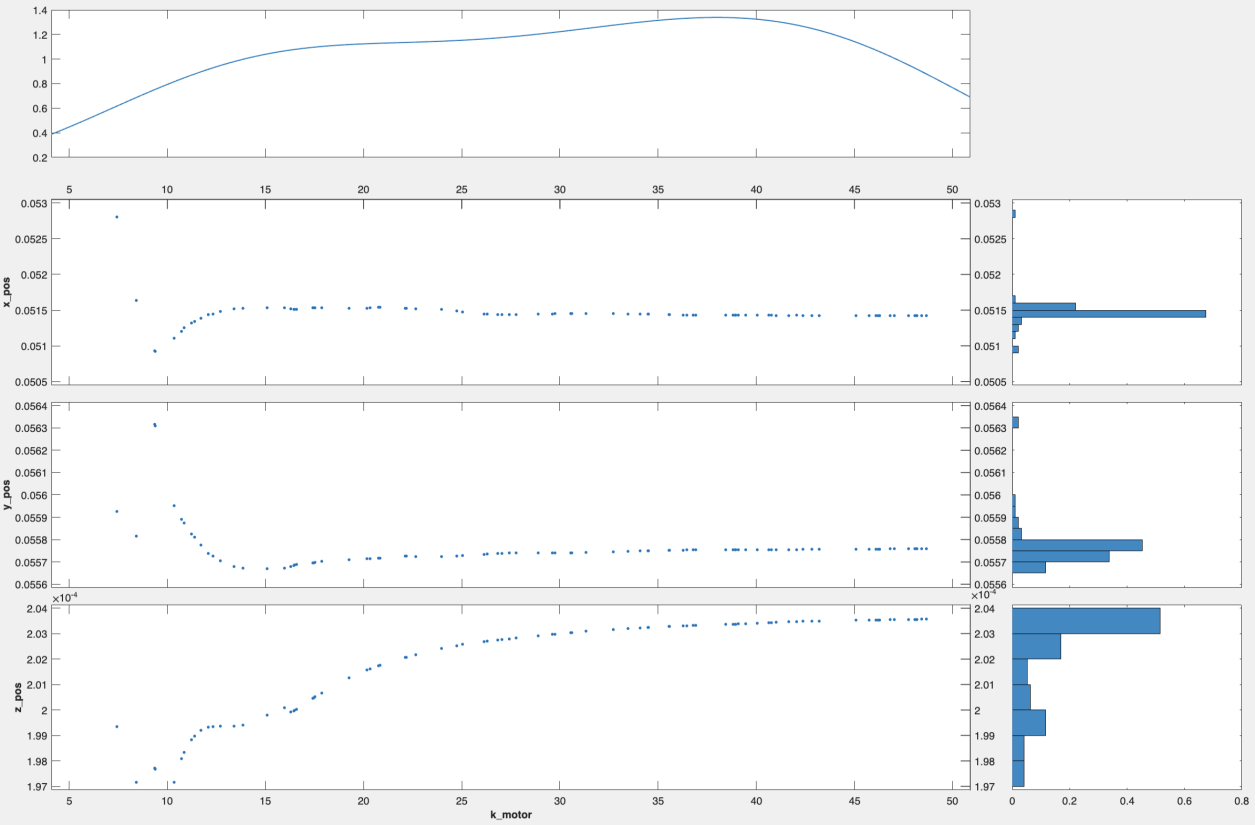


Figure 6.33: Sensitivity analysis results for τ_m (listed as "k_{motor}"), with y axis showing RMSE of the tracking w.r.t each axis resulted from running the simulation with the parameter values represented on x axis

Table 6.22: Outliers for τ_m

No.	$\mathbf{\tau_m}[s^{-1}]$	$\mathbf{x_{pos}}[m]$	$\mathbf{y_{pos}}[m]$	$\mathbf{z_{pos}}[m]$
1	5.5356	5.6140	11.5910	1.0000
2	6.4325	2.1117	10.7491	1.0000
3	6.5501	14.1944	4.7023	1.0000
4	7.0777	7.3721	13.2834	1.0000
5	7.4278	58.8969	81.8685	1.0000

Table 6.23: Stability envelope of τ_m

Parameter	$\mathbf{\tau_m}[s^{-1}]$
Lower Bound	7.4278
Upper bound	48.6767

Run 4 The samples used for this run are shown in Figure B.4. The results of this sensitivity analysis are displayed in Figure 6.34; the y axis represents the root-mean-square error (RMSE) of the tracking w.r.t each axis resulted from running the simulation with the parameter values taken as the sampled data points.

The results indicate strong positive correlations between: 1) C_D and RMSE of tracking w.r.t. all axis, 2) S_x and RMSE of tracking w.r.t. x axis, and 3) S_y and RMSE of tracking w.r.t. y and z axis. Similarly to the previous runs, the stability envelopes for the coefficients evaluated in this run are tabulated in Table 6.24.

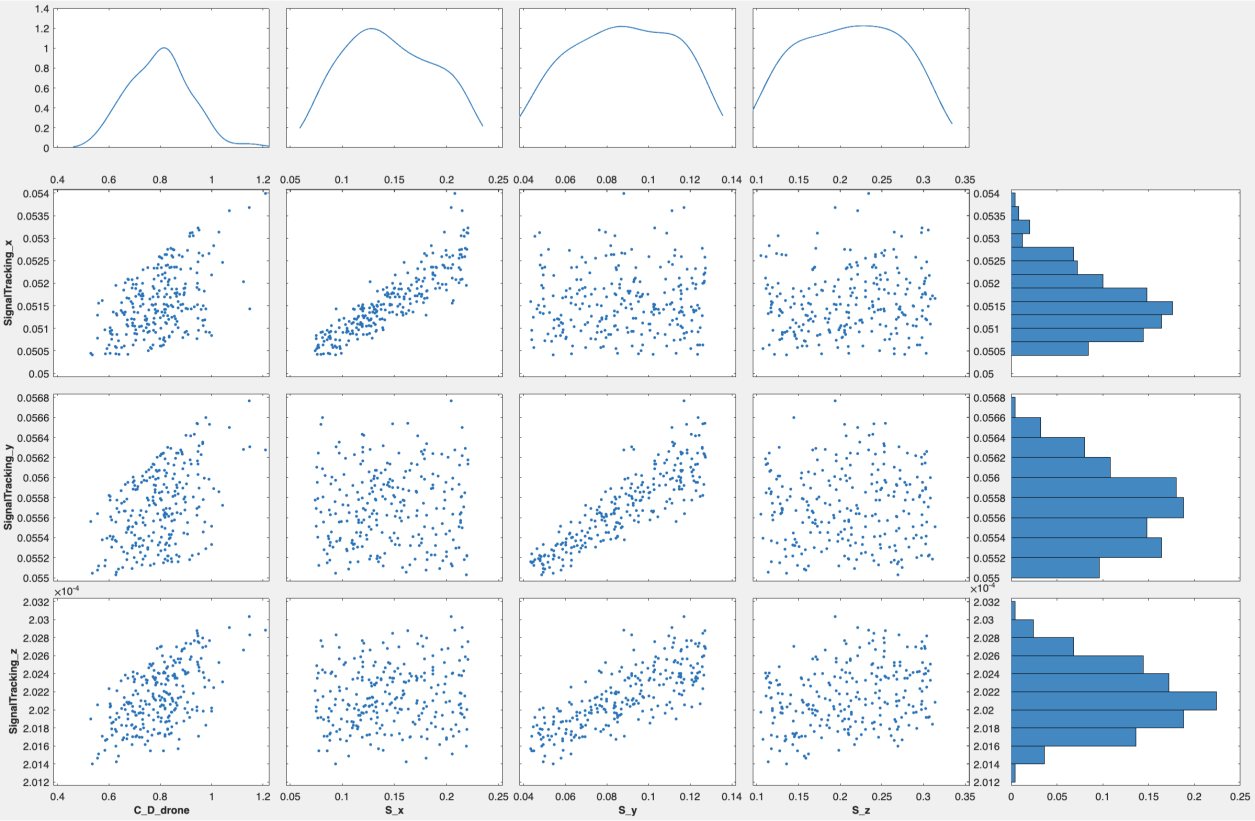


Figure 6.34: Sensitivity analysis results for C_D, S_x, S_y and S_z, with y axis showing RMSE of the tracking w.r.t each axis resulted from running the simulation with the parameter values represented on x axis

Table 6.24: Stability envelopes of C_D, S_x, S_y and S_z

Parameter	$\mathbf{C_D}[-]$	$\mathbf{S_x}[m^2]$	$\mathbf{S_y}[m^2]$	$\mathbf{S_z}[m^2]$
Lower Bound	0.5293	0.0737	0.0433	0.1048
Upper bound	1.2085	0.2204	0.1273	0.3136

Guidance Sensitivity Analysis

Author: Michael and Jip

Following the sensitivity analysis of the navigation and control, a sensitivity analysis including the guidance logic can also be performed. The main purpose of the analysis is to verify the trajectories that the A* algorithm generates, and consequently how accurately the drone is able to follow the trajectory during model simulation. To determine the robustness of the model, it was most appropriate to vary the end goal and trajectory to ensure that the model is able to generate paths towards any point within a given map with obstacles. With a generated path from the A* algorithm, the trajectory can be simulated and both paths can be plotted for comparison to measure the accuracy of the model.

As a first step, a random 3D occupancy map was generated with a much higher obstacle density. As opposed to the previously used 1000 X 1000m^2 wind farm map with only 4 obstacles, now a 200 X 200m^2 map with 50 randomly positioned obstacles will be used. Using a higher density of obstacles in a smaller area will require higher manoeuvrability and control of the drone, allowing us to better assess the performance of the model. Now within the 3D map, 50 random end goals are generated using the built-in random function within MATLAB®. The map with the random points is shown below in Figure 6.35. The blocks have a margin of 3 meter around them to allow for error in the accuracy of the drone. Moreover since the A* algorithm is used for shortest path generation, it often closely goes around obstacle edge points. Therefore this margin is essential to prevent the drone from "crashing" into the obstacles. The boxes in red are the true size of the obstacles and the grey box around them is the obstacle with the applied margin. The green dots are the randomly generated points.

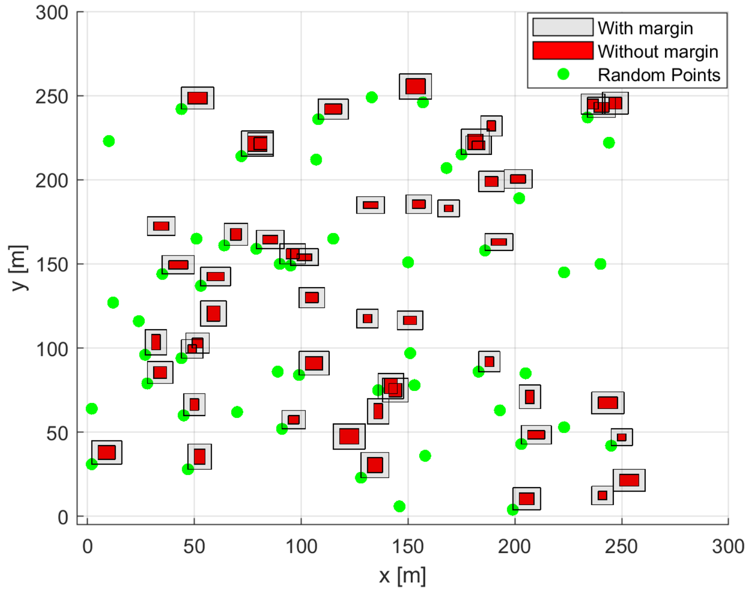
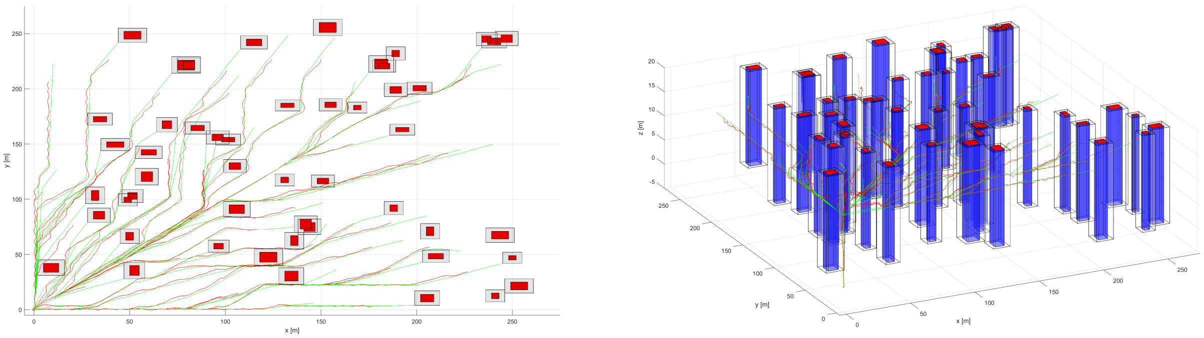


Figure 6.35: Figure showing the generated random occupancy map and randomly generated endpoints.

With Figure 6.35, the team can generate 50 corresponding trajectories and henceforth simulate the drone model 50 times. It was chosen to simulate 50 trajectories as it gives a large enough sample size to analyze the error between the generated path and the simulated path across multiple simulations. A larger sample size could in theory be used to obtain more data and increase the confidence in the performance of the model. Due to limited computational resources, however, a sample size of 50 was deemed to strike a good balance between computational time and retrieved information. Having performed the simulations, the generated and simulated paths can be plotted as shown in Figure 6.36.

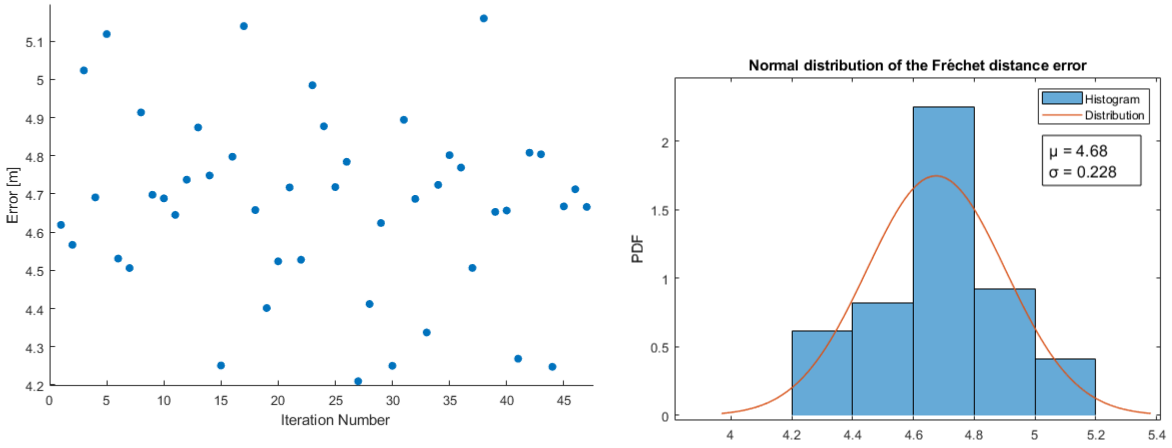


(a) Generated and simulated operation paths in 2D perspective.

(b) Generated and simulated operation paths in 3D perspective.

Figure 6.36: Generated and simulated operation paths for 50 random end goals in a random 3D map.

As can be seen in Figure 6.36, the drone model and path generation is robust enough that a valid path can be generated to reach each end goal. Simulating these generated paths has also resulted in all simulations being able to reach the end goals while trying to follow the generated path. To be able to quantify the error of the simulated path from the generated path the discrete Fréchet distance function was used [65]. This function allows one to calculate the distance between 2 lines that are defined by a sequence of discrete points, it can thus be used as a measure of error between the generated and simulated path. The calculated error and distribution for all 50 simulations have been plotted in Figure 6.37.



(a) Figure showing the error between simulated and generated path for each iteration.

(b) Normal distribution of the error for 50 simulations.

Figure 6.37: Figures plotting the error of the simulated paths compared to generated paths.

Out of the 50 simulations, there was only 1 outlier regarding the path error which had an error of 24.3m. Regarding the rest of the data, the simulated paths had an average error of 4.68m from the generated path with a standard deviation of only 0.228m. This shows that the drone model is quite accurate and capable of following a reference trajectory and avoiding obstacles. Considering that wind turbines are spaced very sparsely with a separation of at least 1km, an average error of 4.68m is remarkably good. This result indeed verifies that the guidance, navigation and control modules are integrated correctly and are able to perform well enough to perform wind turbine inspections.

Requirements Verification

Author: Mike

The GNC requirements drive the model development and implementation. A set of requirements were specified at the beginning of the software lifecycle, which are used to describe the complete behaviour of the algorithms implemented by the model. This model is then used to verify the requirements due to the ease of generating simulations that demonstrate the correct implementation of the algorithms.

In this context, the MATLAB®/Simulink®environment is used for model development and for the verification of the models' requirements. Use is made of the Simulink®add-on *Requirements Toolbox* for verification and testing activities that enable continuous testing and verification throughout the development process.

The set of GNC subsystem requirements, as also given in the compliance matrix Table 3.12, are verified. The requirements are grouped together to complete their verification. The first set of requirements is given in Table 6.25. These are requirements which are concerned with the autonomous aspect of the drone.

Table 6.25: Autonomous requirements verification

Identifier	**Requirement**	**Verification**
AD-Sys-CL-02-1-CS-1	The drone shall be able to navigate without the intervention of the remote pilot on the control commands.	Verified by providing a tracking signal to the drone for autonomous tracking. See Figure 6.38.
AD-Sys-CL-02-2-CS-2	The drone shall be able to land and take-off without the intervention of the remote pilot on the control commands.	The drone is simulated doing a take-off and landing. See Figure 6.38.
AD-Sys-CL-02-2-CS-3	The drone shall be able to perform inspection routines without the intervention of the remote pilot on the control commands.	The drone is simulated successfully tracking an inspection path in Figure 6.26.

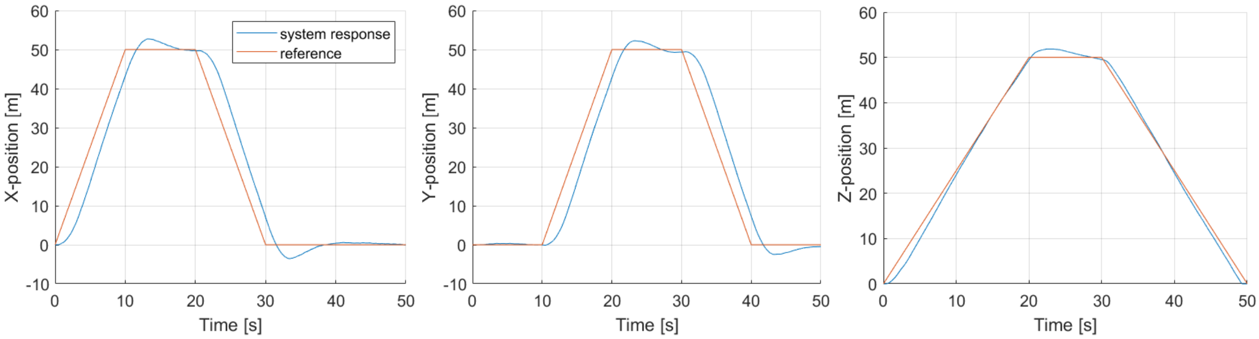


Figure 6.38: Autonomous requirements verification

The second set of requirements contains the requirements which specify the drone's robustness against external disturbances. They are given in Table 6.26. Simulations are performed to verify this set of requirements.

Table 6.26: Robustness requirements verification

Identifier	**Requirement**	**Verification**
AD-Sys-CL-06-1-CS-4	The control subsystem shall cope with wind and wind gusts of 55 [km/h].	The drone is simulated under the influence of a wind gust. See Figure 6.39 in which a wind gust activates at 20 [sec] for 0.5 [sec].
AD-Sys-CL-08-9-CS-12	The control subsystem shall be able to stabilise the drone after an impact load of 3 [Ns].	The drone is simulated under the influence of a 3 [Ns] impact load. See Figure 6.40. Tested up to 20 [Ns].

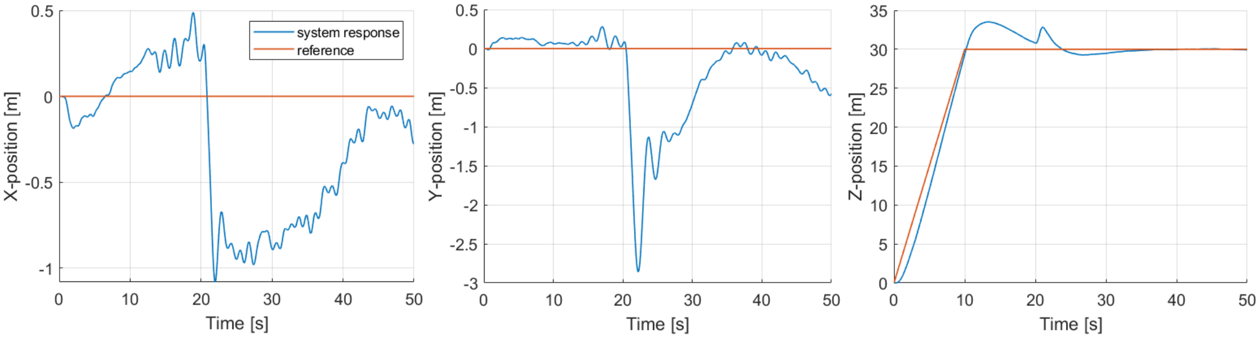


Figure 6.39: Wind gust requirements verification

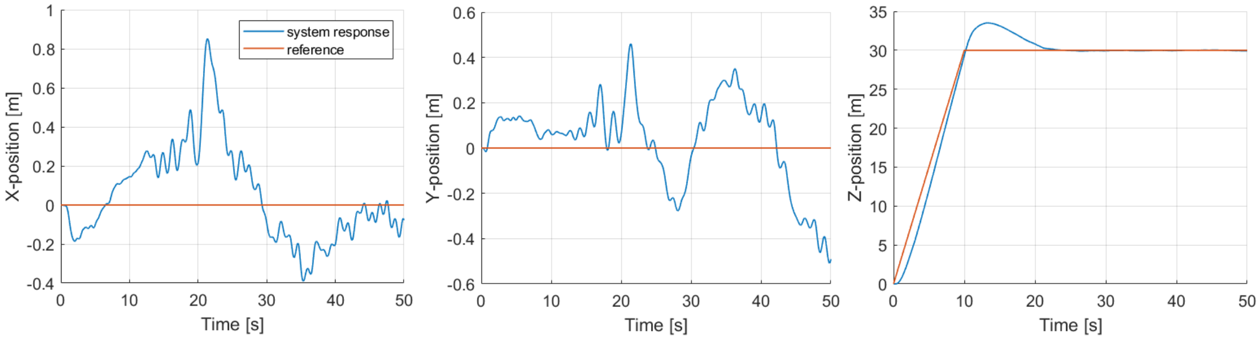
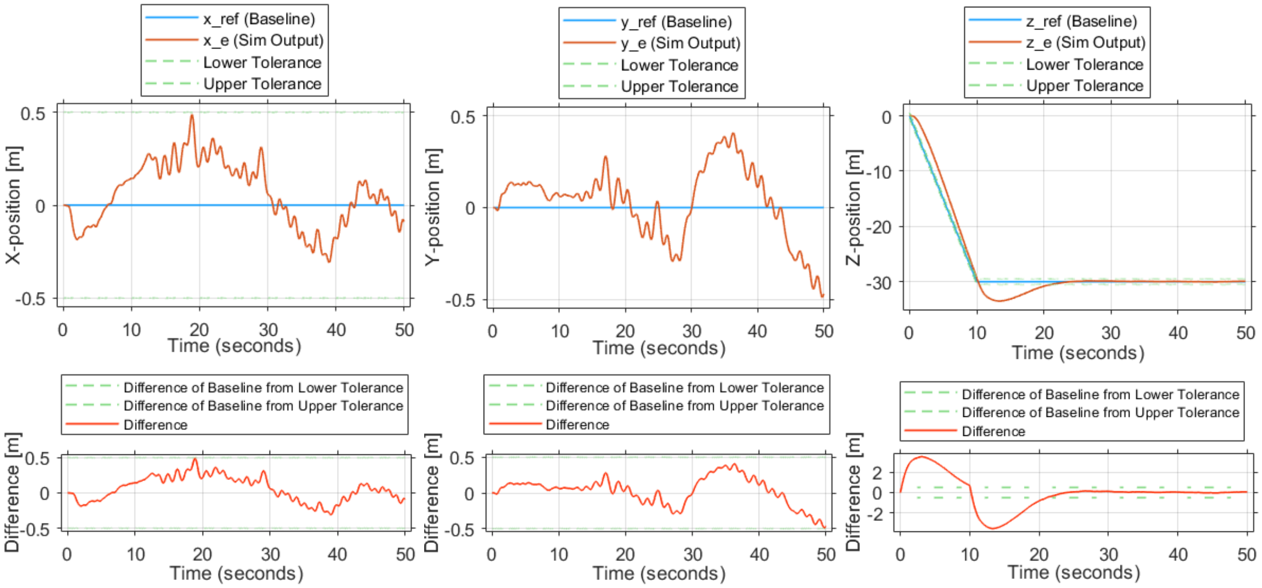


Figure 6.40: Impact load requirements verification

The third set of requirements specifies required accuracies of the drone's navigation and control systems. These are given in Table 6.27.

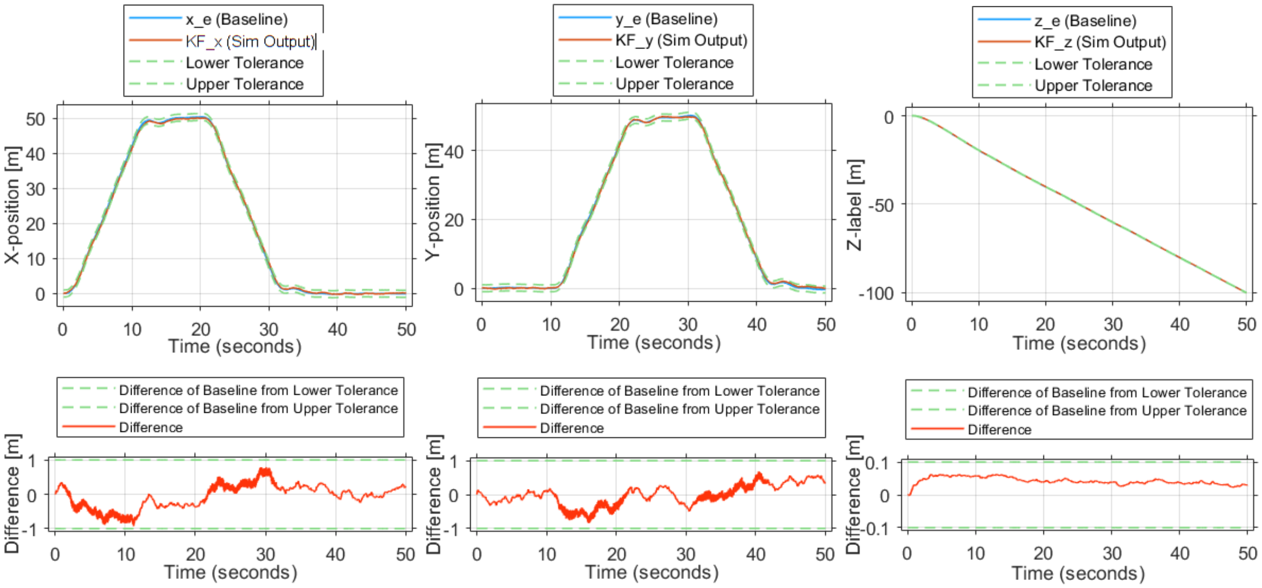
Table 6.27: Accuracy requirements verification

Identifier	**Requirement**	**Verification**	**Rational**
AD-Sys-CL-07-1-CS-5	The control system shall provide the means for inspection module to be stationary with an accuracy of 5 [mm] w.r.t. the desired measurement point.	This requirement is not verified, as shown in Figure 6.41. After the drone has converged to its stationary position, it is able to stay withing 0.5 [m] of the target position.	This requirement was initially put in place to enable inspection techniques requiring high accuracy. However, the visual inspection currently used does not require this accuracy.
AD-Sys-CL-07-1-CS-6	The position of the drone shall be known with an accuracy of at least 5 [mm].	This requirement is not verified, as shown in Figure 6.42. The position is known with an accuracy of 1 [m] in the horizontal direction and 0.1 [m] in the vertical direction.	This requirement was initially put in place to enable inspection techniques requiring high accuracy. However, the visual inspection currently used does not require this accuracy.



(a) Control system accuracy in x-direction (b) Control system accuracy in y-direction (c) Control system accuracy in z-direction

Figure 6.41: Control system accuracy requirement verification



(a) Navigation system accuracy in x-direction (b) Navigation system accuracy in y-direction (c) Navigation system accuracy in z-direction

Figure 6.42: Navigation system accuracy requirement verification

The fourth set of requirements, given in Table 6.28, contains the collision avoidance specifications. They specify how the drone should behave when encountering an obstacle. Based on these requirements, the collision avoidance system is constructed.

Table 6.28: Procedural requirements verification

Identifier	Requirement	Verification
AD-Sys-CL-08-1-CS-9	The control system shall provide a change of path in order to avoid imminent collisions with obstacles.	To be implemented.
AD-Sys-CL-08-1-CS-10	The control system shall be able to predict if a collision is going to occur with an obstacle.	To be implemented.
AD-Sys-CL-07-2-CS-8	The drone shall have a local navigation system with a range of 1 [m].	To be implemented.

The last set of requirements contains requirements with regards to logging drone status and transmitting it to the operator as well as implementing general safety procedures. These requirements are shown in Table 6.29.

Table 6.29: Procedural requirements verification

Identifier	Requirement	Verification
AD-Sys-CL-07-2-CS-7	The control system shall provide the means for drone to have identical performance regardless of which inspection module is fitted, if any.	Not applicable since there is only one inspection module.
AD-Sys-CL-08-4-CS-11	The control system shall be able to receive a manual command to trigger the FTS during automatic operations.	The guidance module has a flight state implemented which shuts off the propeller motors.
AD-Sys-CL-08-14-CS-14	The drone shall be able to determine geo-awareness state variables.	The drone has state estimation implemented which determines the geo-awareness state variables.
AD-Sys-CL-08-14-CS-15	The drone shall be able to determine its current fuel level.	The drone is able to determine the hydrogen fuel level using the pressure regulator.
AD-Sys-CL-08-14-CS-18	The control subsystem shall facilitate immediate return to ground station in case reserve fuel level reaches 10%.	The guidance module has a flight mode implemented which activates a return to base when fuel level reaches 10%

Modelling Standards Verification

Author: Mike

During modelling standards verification, the Simulink®models are checked against modelling standards for two purposes. The first one being to check the model for modelling conditions and configuration settings that cause inaccurate or inefficient simulation of the model. Furthermore, it is also to ensure that consistent modelling guidelines are applied across the subsystems. The second purpose is to check against guidelines such that the C-code generated from the model complies with software certification standards. For these checks the, *Model Advisor* of Simulink®is used.

For the modelling standards specifying basic rules for modelling, the *MAB Modeling Guidelines*[1] are used. These are guidelines stipulated by the *MathWorks Advisory Board* (MAB). The purpose of these modelling guidelines is to provide a common understanding by modelers of control system models.

For the software certification standards, use is made of the *DO-178C* standard[2] guidelines. The *DO-178C* standard defines a set of objectives for software to be certified for use in airborne systems. It is the primary document by which the certification authorities such as *FAA, EASA* and *Transport Canada* approve software-based aerospace systems. By adhering to the guidelines, the generated C-code complies with the *DO-178C.*

---

[1] https://nl.mathworks.com/help/slcheck/ref/model-advisor-checks-for-mab-modeling-guidelines.html

[2] https://nl.mathworks.com/solutions/aerospace-defense/standards/do-178.html

Profiling of the Models

Author: Mike

In profiling a model, the time it takes to run the code is measured. This allows to identify which functions are consuming the most time, after which an evaluation is done for possible improvements to the model. The profiling of the integrated GNC model is performed, for which a simulation of $50[sec]$ in real time is run. The results are shown in Table 6.30, in which a break-down into each individual module is given. The total time is the time used to execute the constituent blocks, while the self time is the time taken by its own execution. The calls indicate the number of times the profiled code called the function.

Table 6.30: Model profiling after optimization

Module	**Total time** [s]	**Self time** [s]	**Calls**
Navigation	78.824	0.765	150012
Model	17.138	0.0	0
Control	13.648	0.0	0
Sensors	4.037	0.076	50508
Allocation	1.089	0.0	0
Total	129.908	12.027	800034

From the model profiling, it is clear that the navigation module consumes the more than 60% of the simulation time. Further expanding the navigation module in its subcomponents, it is revealed that the low-pass filter are the cause of the high computation time. The low-pass filter block has a setting to utilize code generation to simulate the functionality of the block. This significantly speeds up the execution of the low-pass filter blocks. Along with removing redundant blocks, this results in a 72 % improvement in run-time. The new model profiling is shown in Table 6.31.

Table 6.31: Model profiling after optimization

Module	**Total time** [s]	**Self time** [s]	**Calls**
Navigation	13.061	0.508	150012
Model	8.201	0.0	0
Control	1.380	0.0	0
Sensors	3.445	0.056	50508
Allocation	0.803	0.0	0
Total	36.443	6.887	800034

While a significant improvement in execution time has been achieved, for the purpose of performing sensitivity analysis, further improvements can be achieved by generating C-code of the control law, control allocation, and navigation modules. These modules are then no longer accessible from MATLAB® as they run the C-code. However, the only module to which access is required for sensitivity analysis is the model module in which the model parameters are configured. In addition, the simulation can be run in accelerator mode. This simulation mode replaces any interpreted code used in the simulation. The resulting computation times are given in Table 6.32. Then, a simulation with a simulated time of 50 [sec] takes a run-time of 14.082 [sec], which is an improvement of almost 90% over the original run-time.

Table 6.32: Model profiling after C-code generation

Module	**Total time** [s]	**Self time** [s]	**Calls**
Navigation	10.223	0.147	50005
Model	0.031	0.0	4
Control	0.151	0.0	0
Sensors	0.838	0.051	50508
Allocation	0.127	0.0	0
Total	14.082	2.364	600014

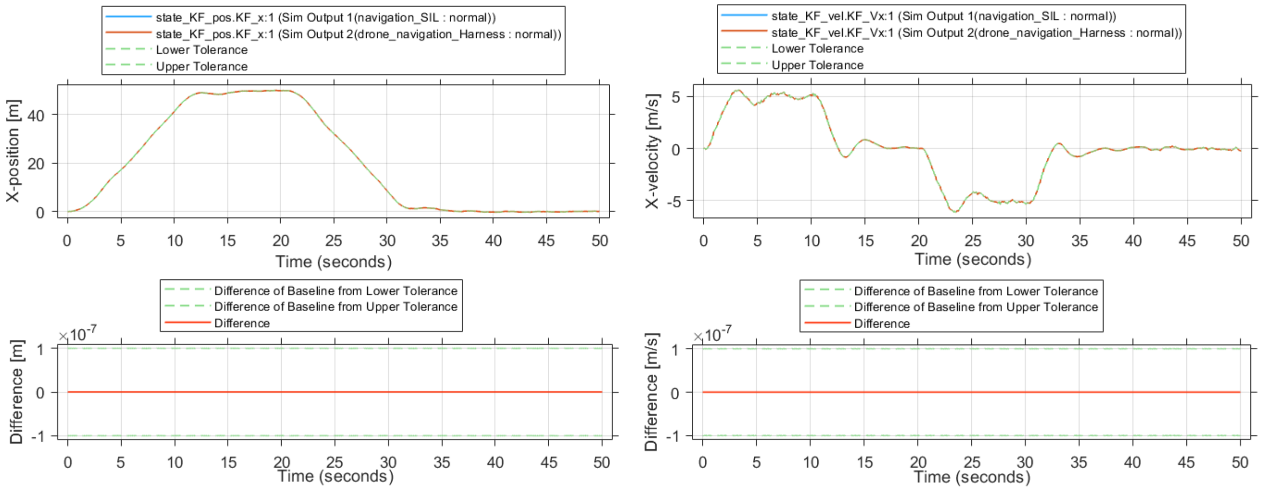
6.6.2 Software-In-the-Loop (SIL)

Author: Mike

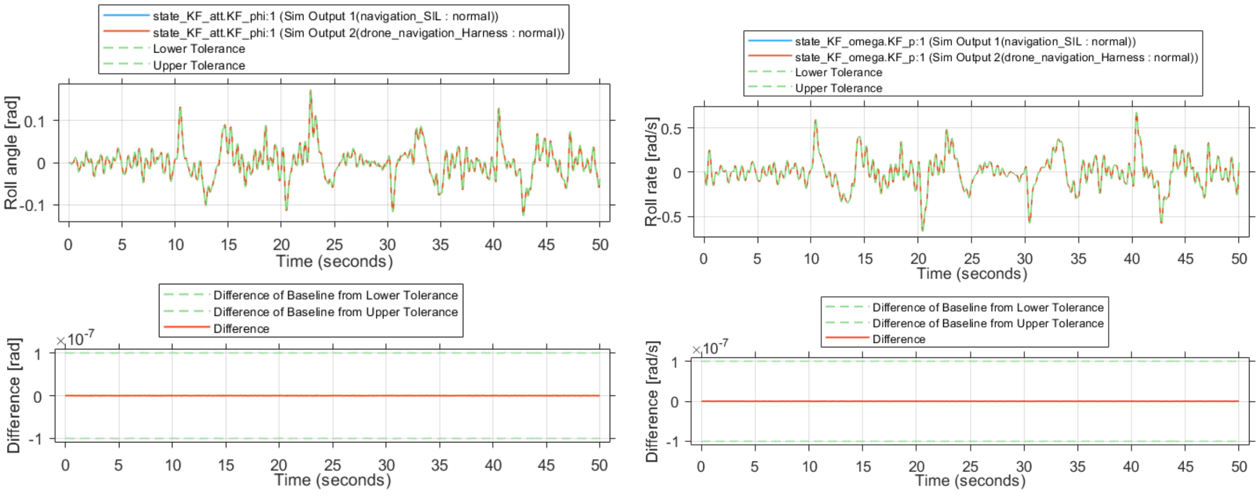
In the GNC software design, development and verification strategy, autocoding of the GNC MATLAB®/Simulink®models is used to generate code optimized for embedded systems, with the purpose of using this code for the flight computer. Hence, after GNC algorithm verification at model level, autocoding is pefromed. In order to verify that the GNC code behaves as the models, testing of GNC generated code is executed in the Software-In-the-Loop environment.

There are three modules for which SIL testing is performed, these include the control law, control allocation, and navigation modules. Each of these modules is auto generated and tested for equivalence against the corresponding Simulink®model behaviour.

Starting with the navigation module, a test harness is created for both the auto generated module and the Simulink®module. A test input is captured from the full integrated GNC system. All output signals of the modules are then compared for equivalence. An absolute tolerance between both signals of 1×10^{-7} is used, which corresponds to single float precision. Figure 6.43 shows the results for the first component of each output vector. However, all signals passed the test, hence the auto generated code functions as expected.



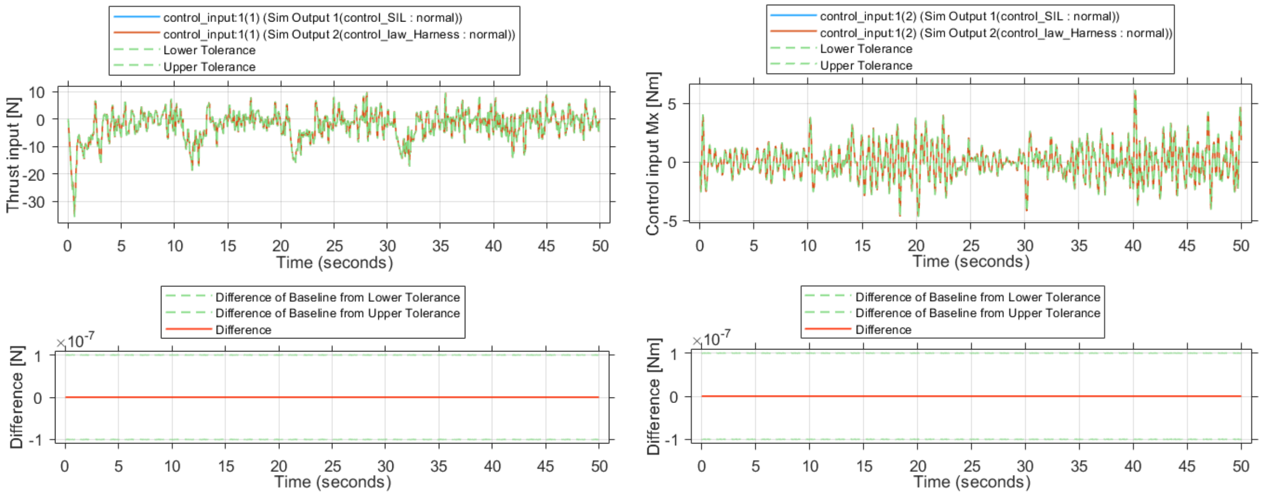
(a) Equivalence of x-position (b) Equivalence of x-velocity



(c) Equivalence of roll angle (d) Equivalence of roll rate

Figure 6.43: Equivalence test of auto generated C-code for navigation module

The next module tested in a SIL environment is the control law module. Again, the inputs of the module are captured from a closed-loop simulation and these are fed into both the Simulink®module and the auto generated C-code module. The output of both is then tested for equivalency with an absolute tolerance of 1×10^{-7}. The results for the thrust input and control input in x-direction are shown in Figure 6.44. However, all signals pass the test of equivalence, hence the auto generated code functions as expected.

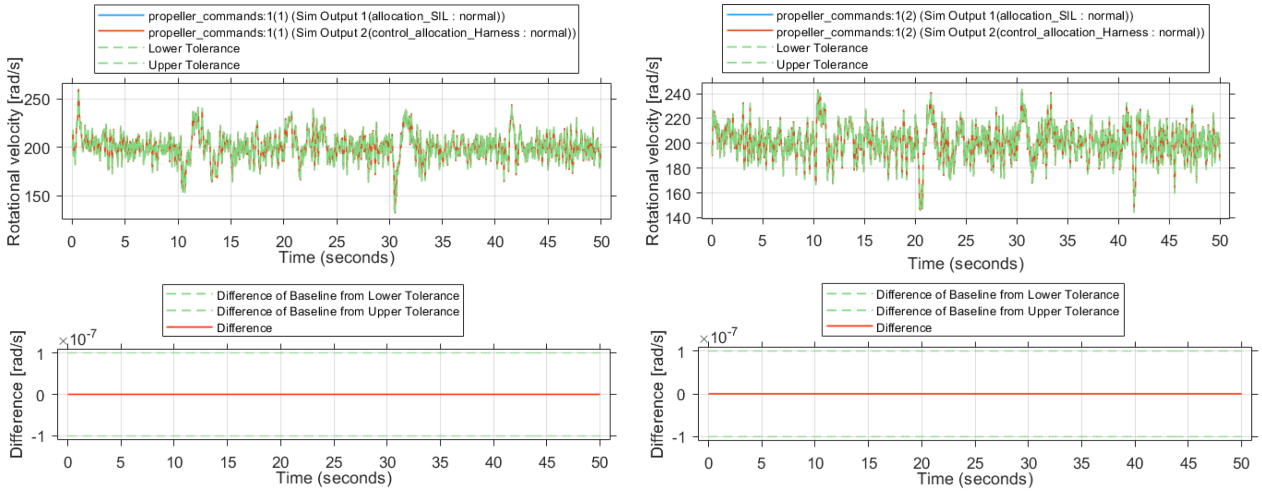


(a) Equivalence of thrust input

(b) Equivalence of control moment around x-axis

Figure 6.44: Equivalence test of auto generated C-code for control law module

The same procedure is followed for the control allocation module. Again, an absolute tolerance of 1×10^{-7} is used since this corresponds to single float precision. Results for two of the outputs are shown in Figure 6.45, however, the other signals also passed the equivalency test. Therefore, the auto generated code functions as expected.



(a) Equivalence of propeller rotational velocity 1

(b) Equivalence of propeller rotational velocity 2

Figure 6.45: Equivalence test of auto generated C-code for allocation module

Completing this verification step gives confirmation that the generated C-code functions as its corresponding modules in the Simulink®model. This is an important step in the verification process and in obtaining flight software capable of running on the flight computer. Since the next step in this process is to run the software on the flight processor, the main errors which could be encountered are due to processor limitations or issues in configuring the real-time operating system which handles communication between the software modules as well as sending data from the processor to a target receiver. These aspects are dealt with in a next stage of the GNC software verification process, namely the Processor-In-the-Loop (PIL) testing. However, this is not dealt with in this report.

7: Drone Operations & Logistics

7.1 General Overview

Author: Tomás, Anton, Jip

In this section, a general overview of the operations and logistics regarding the drone is given. In order to ease its comprehension, the operations and logistic process can been visualised in the form of a flow diagram, found in Figure 7.1.

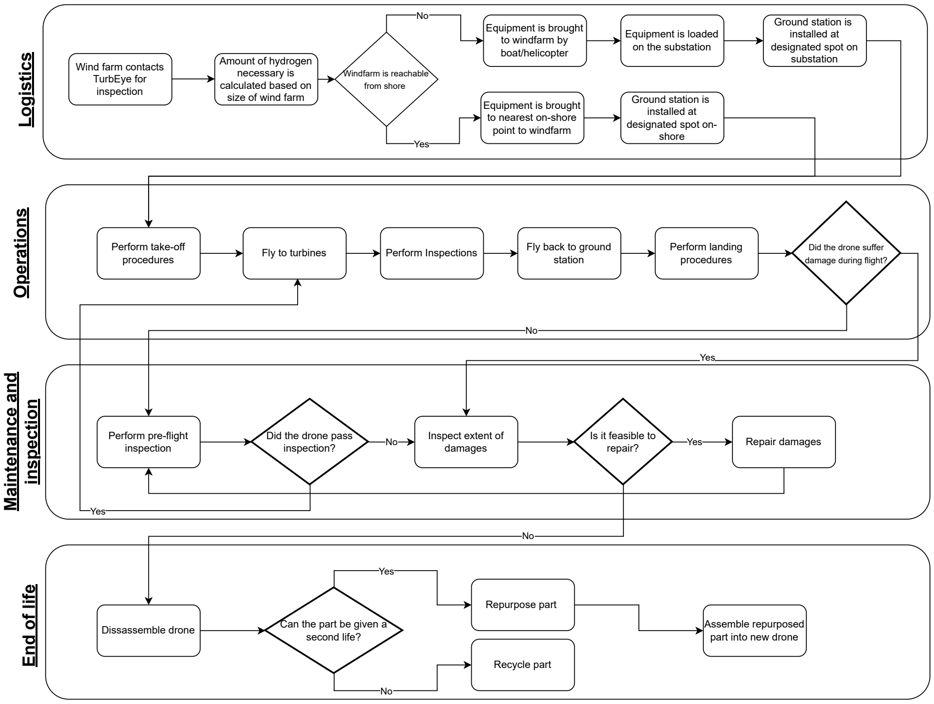


Figure 7.1: Operations and logistics flow diagram

Logistics

As it can be observed above, the logistics behind the subsequent mission depends on the proximity of the wind farm to the shore. In case of a near-shore scenario, i.e. the wind farm is located close enough for the drone to travel back and forth between the shore and the farm for refueling, the ground station would be established on the shore at the nearest feasible point to the wind farm. In such circumstance, the drone would commence its operations from the shore.

However, in the case where the wind farm is situated far from the shore, making it impractical for the drone to make frequent round trips, a ground station needs to be set up at the Operational Sub-Station (OSS) of the wind farm. This entails transporting all the required equipment, including the drone and a 120-litre hydrogen tank, to the wind farm site. Subsequently, the drone will carry out its operations from this established ground station. The mentioned hydrogen tank has a capacity that can sustain approximately 50 operations before requiring a refill or replacement, which would be sufficient to inspect roughly 250 wind turbines, assuming that a drone's operation can inspect 5 wind turbines, on average.

Operation

The operational procedures encompass five stages that the drone will undertake during a typical mission. Prior to takeoff, the drone undergoes a pre-flight checklist to ensure all systems are operational and sensors are calibrated. This checklist should include sensor calibration, battery and fuel tank check, communication verification, among others. During operations, the drone will firstly deploy from the station, until it reaches a safe flying altitude which complies with certification. Secondly, it shall approach the first wind turbine. The inspection of the entire turbine should take approximately 30 minutes, as mentioned in chapter 1 After completing its inspection, the drone shall fly towards the next wind turbine in its pre-programmed route. If there are no more inspections required, or the drone has to be re-fuelled, it will return to the ground station. During landing procedures, the data gathered during flight, by means of an SD card, will be retrieved in order to post-process the information, and the hydrogen tank shall be refuelled/ replaced. At the end of the operation block, a decision on whether any noticeable damage occurred during operation is made. If no damage has been reported, the cycle is repeated by starting at the pre-flight check box. Should any damage occur to the drone, one must return to damage inspection.

Maintenance and Inspection

The maintenance and inspection block starts with a routine pre-flight inspection, similar to the former take-off procedures. Then, if the inspection is passed, it is suitable to commence operations again. If any damage is detected, the drone will be further inspected. Should the drone be feasible to repair, reparations will be carried out and it will return to the pre-flight inspection block. If it is not feasible to repair the drone, one will move to the end of life procedures.

End-of-life

The end-of-life procedures will start with the disassembly of the drone. Non-damaged parts will be assessed for the possibility of re-use. If they can be repurposed, they will be used for the production of a new drone. If they cannot be repurposed, materials will be appropriately recycled.

7.2 RAMS Characteristics

Author: Enes, Arham, Tomás

Reliability, Availability, Maintainability and Safety (RAMS) are critical aspects of an engineering design that characterize certain properties. This section provides an in-depth analysis of these characteristics, highlighting their technical significance and impact on the drone's performance, up-time, maintainability, and safety standards.

7.2.1 Reliability

By definition, reliability is the probability of a system to consistently perform its intended functions without failure or breakdown within a specified period of time. Due to the lack of available data regarding the failure rates of each component, calculating reliability numerically is not feasible. Thus, in order to assess TurbEye's degree of reliability, the drone's critical components have to be identified, as well as its failure modes.

Concerning its critical components, one can conclude that the elements which are paramount for the drone's longevity are the fuel cell, hydrogen tank, and its flight computer. The main reason behind this is because the design applied to TurbEye follows a **fail-safe philosophy**. Therefore, by making primary components redundant, such as the propellers, motors, and navigation sensors, it can be ensured that the failure of one shall not be catastrophic, as there is always a backup component capable of performing the same function. Additionally, a safety factor of 1.5 has been applied to the structural design. Given their high cost, the fuel cell and its hydrogen tank do not have a redundancy element on board, making them the most pivotal components. It is certain that the design philosophy applied increases the weight of the drone. However, this is a price worth paying, as the drone is already a high-cost investment, and the potential consequences of losing it far outweigh the minimal increase in weight resulting from the added preventive measures.

A fault tree depicting the main failure modes of the drone is provided in Figure 7.2 to highlight the critical failure paths and their contributing factors. This diagram is essential as it allows for a systematic visual representation of potential failure scenarios, enabling us to identify and prioritize mitigation measures (which shall be discussed in the risk analysis chapter 3.3), allocate resources effectively, and enhance the overall reliability by addressing the root causes of failures.

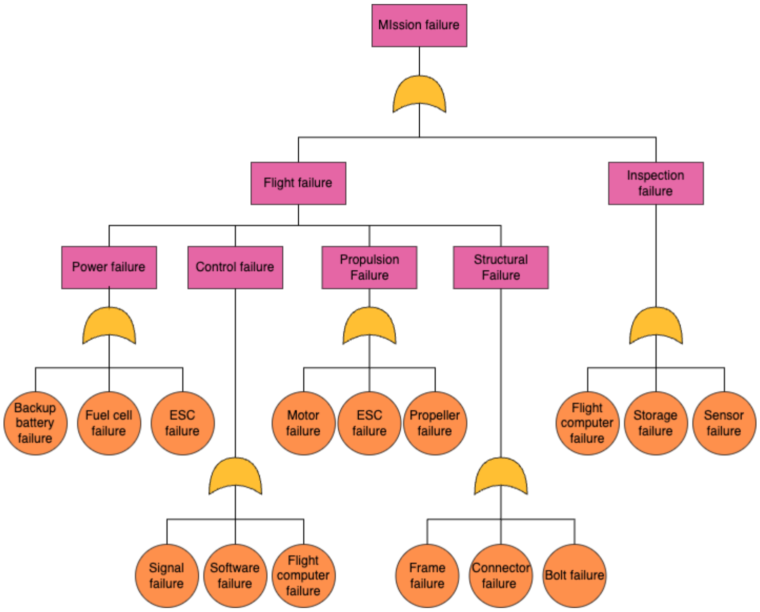


Figure 7.2: Fault Tree

As visible above, the entire operation can fail due to two high-level cases: due to an airborne anomaly or a problem during the inspection. Concerning the latter, such could be the result of a flaw in the flight computer, storage, or in the inspection sensors. Nevertheless, a flight failure is undoubtedly more critical than an inspection failure. This type may result from a power, control, propulsion, or structural failure. Each of these can then be subdivided into more specific failures, which shall be further explained in subsection 7.2.4.

In summary, one must be concerned mainly with the fuel cell and its accompanying components. However, all mentioned failure modes should be treated with equal importance in order to avoid any mishaps.

7.2.2 Availability

The drone's availability can be distinguished into two aspects: its availability due to the system's manufacturing speed and its operational availability resulting from its low maintenance requirements. In the system's first years, the former is not a major concern, as there should not exist a need to build more drones, since a single drone should be sufficient to inspect all wind turbines, assuming simultaneous maintenance of wind turbines will not be required.

Furthermore, TurbEye's operational availability is dependant on its maintainability and reliability. The greater the reliability of a system, the higher its availability, as fewer failures result in reduced downtime. This is ensured by the redundancy philosophy applied, previously mentioned above. Simultaneously, an easily maintainable system increases the likelihood of availability by enabling efficient maintenance practices and prompt issue resolution. Such shall be discussed in the next subsection.

In order not to exhaust operators, one will consider an average daily operation time of 12 hours, which consists of two shifts of 6 hours. Due to the thermal camera, the drone can also perform during night time, but this shall not be treated for now. A single trip should take 3 hours (plus 30 minutes in case of an extraordinary event), whereas refuelling and pre-flight inspections last around 45 minutes. Therefore, taking into account an average of 4.8 inspections per trip and 3 trips per working day, this provides a total of approximately 15 wind turbine inspections per day. To put it into perspective, the world's biggest wind farm, *Hornsea 2*, has 165 turbines [66]. Thus, it would be possible to inspect *Hornsea 2* entirely in only 11 days.

7.2.3 Maintainability

Maintenance is essential for safe, reliable and long-term functioning of any engineering product. To this end, an outline of both scheduled and non-scheduled maintenance activities are given in this section.

To enhance the maintainability of the product, the design incorporates a modular approach with replaceable holders for tanks, fuel cells, and other appendages. This design allows for easy inspection or replacement of specific components without the need for extensive disassembly. The increased accessibility ensures that all components are readily reachable for maintenance purposes.

The design also utilizes standardized interfaces between the distinct subsystems. This means that the bolts, cables, and other connections use standardized dimensions and materials, often incorporating commercial off-the-shelf (COTS) components. This approach allows for easy connection and re-connection of components and reduces the risk of errors in maintenance activities.

To obtain the health status of the drone and troubleshoot issues without requiring physical access, a robust telemetry system needs to be implemented. This system will enable remote monitoring and provide self-diagnostic capabilities, reducing downtime by allowing proactive maintenance planning.

The maintenance diagram of the drone is given in Figure 7.3. Telemetry is recorded by the drone, and limited telemetry is provided to the operations base due to communication constraints, which is checked constantly for any discrepancies. If a discrepancy is detected, the drone ceases its inspection operations and returns to the OSS. The drone is then brought back to the workshop for further diagnostics and maintenance.

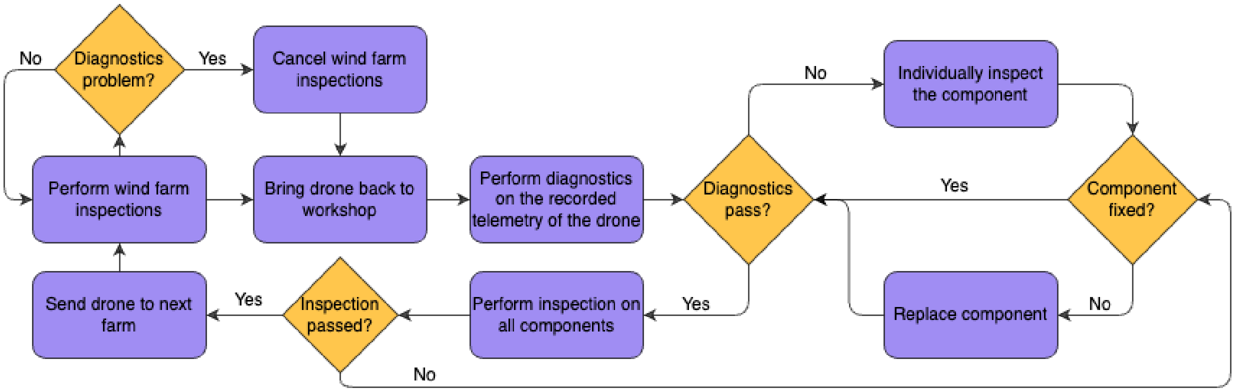


Figure 7.3: Maintenance diagram of the drone

The drone is also brought back to the workshop after the wind farm inspections are complete. First diagnostics are performed on the recorded telemetry data, and if an issue is found with a specific component, the component is further diagnosed and inspected. If the component cannot be repaired, it is replaced. Afterwards, diagnosis is again performed to see if the fixing of the component was helpful.

If the diagnosis passes after the component is fixed, inspection and maintenance is performed on the components. A list of scheduled maintenance is given in Table 7.1. If during the inspection, a problem is encountered with a component, it's fixed or replaced. After the inspection is completed, the drone is ready for the next wind farm inspection.

Table 7.1: List of scheduled maintenance of the drone

Component	**Scheduled Maintenance**
Fuel Cell	Identify any leaks Check the voltage output Replace filters if required Inspect the fans
Hydrogen Tank & Pressure Regulator	Inspect the tank for damages Perform pressure testing Check for leaks Lubricate the valves Inspect the burst disk
Backup Battery	Inspect for damages Check the output voltage Charge the battery to 50%
Propellers & Motors	Inspect for damages Lubricate components
Sensors	Inspect for damages Calibrate sensors Test sensors for accuracy
Connections	Inspect for damages Tighten fasteners and connections
Cameras	Inspect for damages Clean lens and gimbal Lubricate gimbal Verify saving of data to storage
Cables	Inspect for damages Check for loose cables

Apart from the scheduled maintenance performed, the drone will undergo a more extensive inspection every year of its operations. To ensure that the drone is performing as expected, the drone undergoes extensive testing as defined in subsection 10.2.3. Even though this would increase expenses marginally, given the cost of the drone and critical nature of inspections it is deemed worth it.

7.2.4 Safety

The safety of autonomous drones used in wind turbine maintenance inspections is of paramount importance. If the drone were to sink into the ocean, it could cause significant damage to marine life as well as result in the loss of a costly drone and significantly delay operations. Additionally, as the drone autonomously navigates environments and interacts with critical infrastructure like wind turbines, ensuring its safe operation becomes essential to protect personnel, property, and the environment. This section focuses on the various aspects of safety that must be considered when deploying autonomous drones for wind turbine inspections. There are a variety of methods to ensure safety, and the main ones are listed below.

Risk Assessment and Mitigation

In section 3.3, a technical risk assessment was performed to identify potential hazards to the operation of the drone. The assessment was conducted up to a subsystem level to ensure that specific risks were identified and addressed. Risk mitigation measures were then implemented to ensure that appropriate actions are taken to prevent any harm to the drone itself or the surrounding environment. By systematically assessing and mitigating risks, the safety and reliability of the drone are enhanced during its operation.

Built-in redundancies and emergency response

The drone has built-in redundancies, such as multiple motors & propellers, GPS, IMU, and barometers. In case of a motor or propeller failure, the drone can still fly safely. If the drone diagnostics show that one of the components isn't functioning properly, the drone initiates a return-to-base sequence and cancels inspection operations.

The return-to-base sequence is a sequence in which the drone withholds any turbine inspection operations and immediately returns to the OSS for examination. This happens if the telemetry data and the diagnostic software of the drone indicate an issue within any subsystem.

Furthermore, the drone is equipped with a backup battery in case the hydrogen fuel cell fails to provide power and for during certain manoeuvres where additional power is required. If the fuel cell fails, again a return-to-base sequence is triggered so that the fuel cell can be examined to resolve any issues.

Collision avoidance

The drone is going to be equipped with both global as well as local real-time collision avoidance. Global collision avoidance is knowing the obstacles beforehand and planning a route around them as explained in subsection 6.5.1. Local real-time collision avoidance will be implemented to the drone, which is live scanning of objects around the drone and modifying the route to avoid any collisions.

Frequent Maintenance

As explained in subsection 7.2.3, regular maintenance is performed on the drone every time it returns to the workshop after inspecting a wind farm. This ensures that the drone remains in optimal condition for its next mission, and any potential issues are identified and addressed proactively. Additionally, a yearly detailed maintenance is also conducted to thoroughly inspect the drone and identify any hidden defects that may have been missed during regular maintenance. This comprehensive maintenance approach aims to maximize the drone's reliability and performance, minimizing the risk of failures or malfunctions during its operation.

Burst disk

Safety is more of a concern when having a pressurized hydrogen storage on-board. To prevent any safety risks caused by hydrogen pressurization the hydrogen subsystem is equipped with a burst disk, which is a component that releases hydrogen to prevent damage in-case the pressure inside exceeds a certain threshold.

7.3 Wind Farm Traversal Problem

Author: Enes, Arham

A major pre-flight procedure involves planning out the trips required by a drone to efficiently cover all the wind turbines in a wind farm. It is necessary to follow a strategy to visit each turbine while ensuring that each trip has sufficient fuel, as an non optimized traversal path would require more trips potentially increasing downtime and cost.

Various methods were explored, including k-means clustering and the travelling salesman problem (TSP), but they proved to be ineffective for this specific scenario. Additionally, the traditional implementation of k-means clustering did not provide a way to limit the number of elements within each cluster, resulting in several trips with an excessive number of turbines to inspect and insufficient fuel to cover them all.

While the TSP is suitable for problems with cases where returning to the base isn't necessary, in this case, the drone needs to return to refuel after each trip, making it unsuitable. The vehicle routing problem (VRP), which is TSP with additional constraints of having to return to the base every once in a while was deemed a better problem statement. More specifically, since there is only one drone making multiple trips, the problem can be defined as Multi-Trip VRP (MTVRP). Initially a custom-made algorithm was developed, and certain optimizations were performed. Then, an industry-developed algorithm was used: OR-Tools by Google [67].

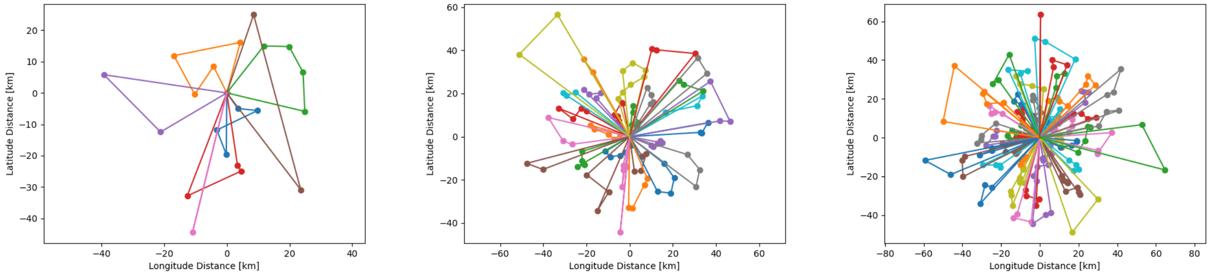
The problem is formulated as follows: *What is the minimum amount of trips taken to cover all the wind turbines in a wind farm?*. The inspection time for a single turbine is assumed to be 30 minutes (which was found to be on the higher end based on market analysis), and a maximum of 90% of the hydrogen capacity of the tank is used to allow for some margin in case of sub-optimal routes and other disturbances. The speed of the drone and the required power for the drone can be referenced from section 4.7, corresponding to the angle for maximum range.

7.3.1 In-house Heuristic Algorithm

The in-house algorithm's logic consists of the following steps:

- The first turbine is selected as the one that is closest to the base that has not been inspected yet
- The next turbine is selected as the one closest to the first turbine that hasn't been inspected yet
- The algorithm checks if there is sufficient fuel to go to the next turbine, perform the inspection and return to base. If there isn't it returns to base immediately
- The process repeats until the drone can't inspect another turbine either due to a lack of fuel or if there are none remaining

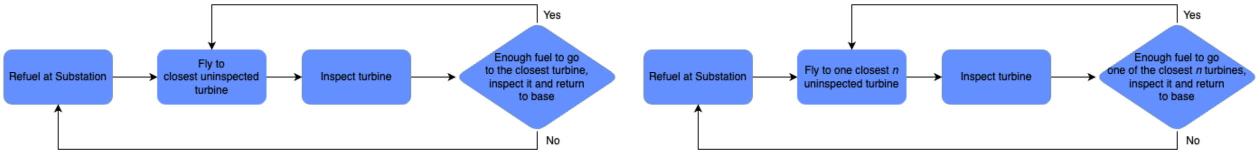
The routing algorithm can be initially tested on a randomly distributed set of coordinates. The results for different sets of points can be observed in Figure 7.4.



(a) Route planning for 20 randomly generated turbines (b) Route planning for 100 randomly generated turbines (c) Route planning for 200 randomly generated turbines

Figure 7.4: Route plans for different sets of turbines

The time complexity of the algorithm is found to be $O(N^2)$. Although this is not ideal, the sample sizes used for wind farms typically do not exceed 200, so the scaling is not problematic, and the execution is nearly instant. Intuitively, it can be observed that the algorithm is not optimal and may perform poorly in certain edge cases. To enhance the robustness of the model, the algorithm can be iterated with the option of randomly selecting the closest 2 or 3 neighbours. The new route is then compared to the original route, considering the number of inspections first and, if they are equal, evaluating the amount of hydrogen used. The best route is saved in a JSON file and can be compared to iterations conducted at a later stage, perhaps with a higher number of neighbours. The original and updated algorithm is shown in Figure 7.5.



(a) Logic for original algorithm (b) Logic for updated algorithm

Figure 7.5: Logic Diagrams

The in-house developed algorithm provides a good starting solution, however has a lacking optimization algorithm which is very dependant on the initial route. For this reason, the team decided to search for an optimization algorithm for this specific problem.

7.3.2 OR-Tools VRP Algorithm

An industry developed tool is OR-Tools, which is an open source library developed by Google that has implementations of a wide range of algorithms to popular optimization problems, with VRP being one of them. The algorithm within the VRP uses Guided Local Search (GLS) which is a meta-heuristic algorithm that is used to solve optimization problems by minimizing the cost between nodes [68].

GLS utilizes Local Search (LS) algorithms, which are simple optimization algorithms that search the neighbours of a potential solution for better cost. Cost is the parameter to be minimized, which is the number of trips in this case. LS has the advantage of finding solutions very fast, but has a downside of getting stuck in a local minima. This becomes a problem, as there might exist other minima that are better solutions [69].

To improve this shortcoming, various algorithms are built utilizing LS, with one of them being GLS. GLS kicks in when LS is trapped in a local minima and modifies the cost function, widening the solution space to find a better solution. A working flow of GLS can be found in Figure 7.6.

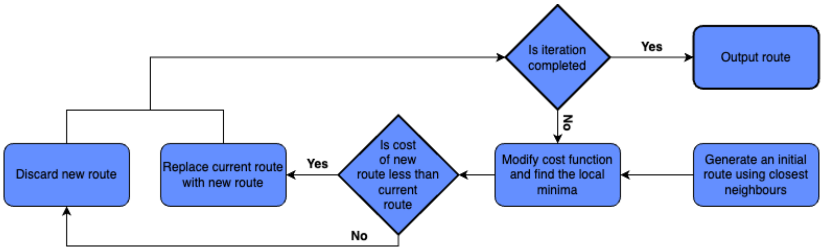


Figure 7.6: Diagram representation of the GLS algorithm

After the final implementation of the GLS algorithm using OR-Tools, all three of the algorithms are run on a real-life wind farm and the corresponding results are presented and compared in subsection 7.3.3.

7.3.3 Hornsea 2 Routing Results

All three models were tested on a real-life example. The Hornsea 2 wind farm, which consists of 165 wind turbines, is recognized as the world's largest [66]. It spans 70 kilometres in width and 20 kilometres in height, with its offshore substation (OSS) slightly skewed to the side. The wind turbines are arranged in an L-shape pattern, as depicted in Figure 7.7.

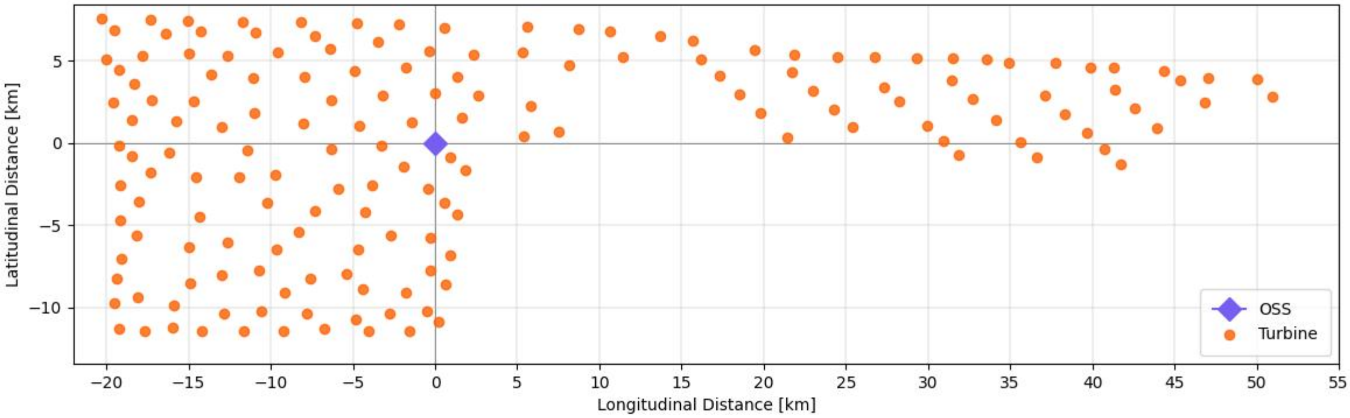
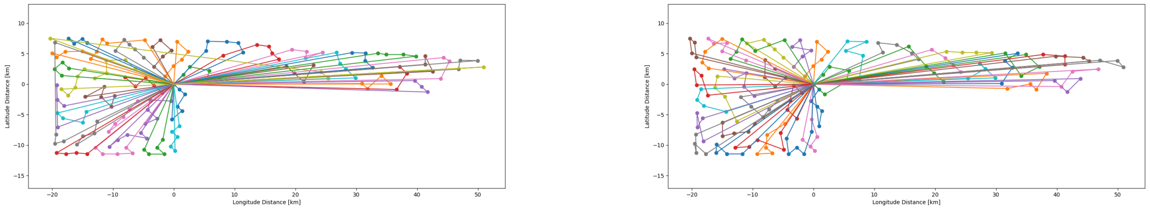
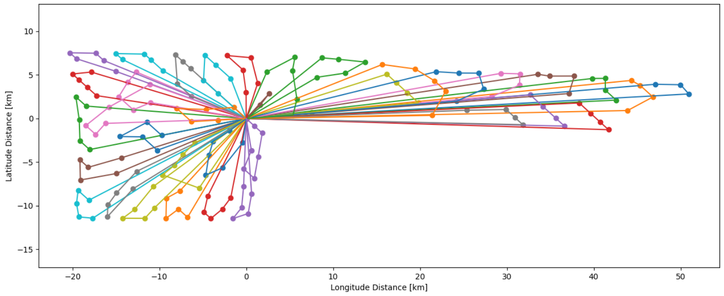


Figure 7.7: Layout of the Hornsea Project 2wind farm



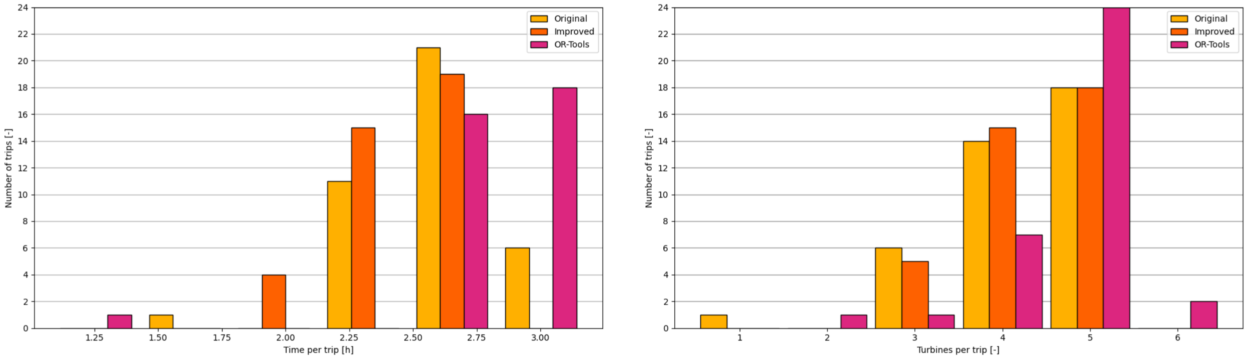
(a) Route planning for Hornsea using original algorithm (b) Route planning for Hornsea using updated algorithm



(c) Route planning for Hornsea using OR-Tools algorithm

Figure 7.8: Route plans for Hornsea 2

Although they are challenging to interpret, some details can already be observed. The original model makes a few awkward decisions, such as choosing a route that travels all the way from (-20, 8) to (22, 3), resulting in a significant waste of energy and fuel. On the other hand, the updated model appears to be much more effective in this regard but is still quite chaotic with a lot of overlapping routes. The OR-Tools algorithm, however, is very consistent and methodical. It has no outlying trips that take unnecessarily long. This observation can be further supported by referring to Figure 7.9.



(a) Flight time per trip (b) The number of turbines per trip

Figure 7.9: Time taken and turbines inspected for all the models

As expected, for the most part, the original algorithm performs just as well as the improved one. While the original algorithm demonstrates comparable performance, it is in the edge cases where the improved model truly shines. The number of trips inspecting 5 turbines remains the same, and the number inspecting 4 turbines only increases by 1. However, the trip that only visited a single turbine no longer exists. Additionally, Figure 7.9a demonstrates that, on average, the inspections take longer, indicating that the improved model makes better use of the drone's endurance capacity.

In the OR-Tools model, there was one trip that lasted less than 1.5 hours and only checked two turbines. While some trips in the OR-Tools model exhibit lower efficiency, most of the trips make full use of the drone's endurance capacity and last close to 3 hours. Consequently, most of the trips inspect 5 turbines each, and one trip even checks 6 turbines. These improvements are supported by the data presented in Table 7.2, which indicates that the improved model requires 38 trips compared to the original's 39 trips. It is worth noting that the OR-Tools model only requires 35 trips. Both custom models exhibit higher hydrogen consumption and increased flight time to cover all the turbines.

The reduced number of trips also leads to lower total downtime between them, as a single refueling session can take approximately 30 minutes.

Table 7.2: Parameters for Comparison between models

Parameter	Original	Updated	OR-Tools
Trips [-]	39	38	35
H_2 used [kg]	3.802	3.755	3.647
Total flight time [hrs]	101.45	100.36	98.08

7.4 AI Damage Detector

Author: Tomás

As a state-of-the-art system, the inspection data retrieved from TurbEye, comprised of multiple images and video files, will be post-processed by an external computer. One shall be equipped with a professional GPU able to use a dirt and damage detector, implemented with Artificial Intelligence.

7.4.1 Data collection

Prior to the creation of a model, a dataset containing drone images of a wind turbine was found, along with each images' YOLO label [70]. YOLO (You Only Look Once) is an object detection algorithm which takes as input an image (or video) and predicts bounding boxes which specify the location and class label of objects within the image, along with its confidence in the prediction [71]. Each respective YOLO label has the following text file format: "<class of object> <x-coordinate of center> <y-coordinate of center> <box width> <box height>". All the dimensions of the file have been normalized by the height or width of the figure, which have a size of 371x586 pixels. An example can be seen in Figure 7.10.

Figure 7.10: YOLO algorithm example

As it can be seen in the image above, it is able to recognise the specified object (a horse) with high prediction confidence, along with its location in the image. Likewise, for the report's intended usage, the YOLO format can provide valuable information about the presence and location of damage or dirt within the images.

7.4.2 First Model: Convolutional Neural Network

To begin with, a simple binary image classifier was developed using TensorFlow [72]. Using this Convolutional Neural Network (CNN) model, an image was classified as "Dirt/Damage" or "No Dirt/Damage", depending whether dirt and/or damage were found in the *.png* file. Although it is very rudimentary, it is a good first iteration to verify if the dataset is large enough to train more advanced models. The binary classifier can be found in an Google Collab folder.

Theoretical Background

A CNN is a powerful deep learning model designed for image analysis. It takes advantage of the inherent spatial hierarchies in images to perform effective pattern recognition. The key idea behind CNNs is the use of convolutional layers, which apply filters to the input image for the detection of local features, such as edges and corners. These filters are learned during training, enabling the network to automatically extract relevant visual patterns.

Moreover, CNNs employ pooling layers to downsample the feature maps and retain important information while reducing computational complexity. In order words, the model is shrinking/compressing the information while keeping the important details intact. From there, the network's architecture includes fully connected layers that aggregate the learned features and map them to the desired output classes, enabling the intended classification.

Additionally, a useful technique is also applied: parameter sharing. It is used to efficiently learn and recognize patterns in images by employing previously found sets of learned features across different parts of an image. The final output layer of the AI model classifies the image by assigning it to the class with the highest confidence level.

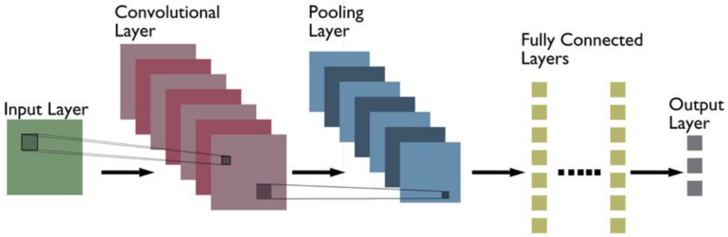


Figure 7.11: Typical CNN Architecture [73]

Performance analysis and results

In order to evaluate the model, one can retrieve valuable metrics which can help one comprehend if the model fits the dataset in hand.

- **Accuracy** is a measure of how often a classification model correctly predicts the correct class label. It is calculated by dividing the number of correct predictions by the total number of predictions. However, accuracy may be unreliable if any class imbalance exists. Thus, other parameters have to be assessed, such as precision and recall.
- **Precision** measures how many predictions of an individual class are truly positive out of the instances it predicts as positive.
- **Recall** aims to gather what proportion of actual positives are correctly classified from the actual positive instances in the dataset.
- **F1 Score** This metric combines the two previous metrics into a single index, providing a balanced measure of precision and recall. This allows for a comprehensive evaluation of a model's effectiveness in situations where both aspects are important.

In order to accommodate computational constraints, the tuning process for this specific model involved resizing all the images in the dataset to a scale of 256x256 pixels. This resizing ensures that the images are more manageable in terms of memory and processing requirements, allowing for smoother training and inference processes.
Applying the metrics derived from this tuned model to TurbEye's intelligent agent yields the results depicted in Figure 7.12. These results provide a comprehensive evaluation of the agent's performance, highlighting key metrics such as precision, recall, and F1 score. By analyzing these metrics, one can assess the effectiveness of the intelligent agent in detecting and classifying the presence of dirt/damage.

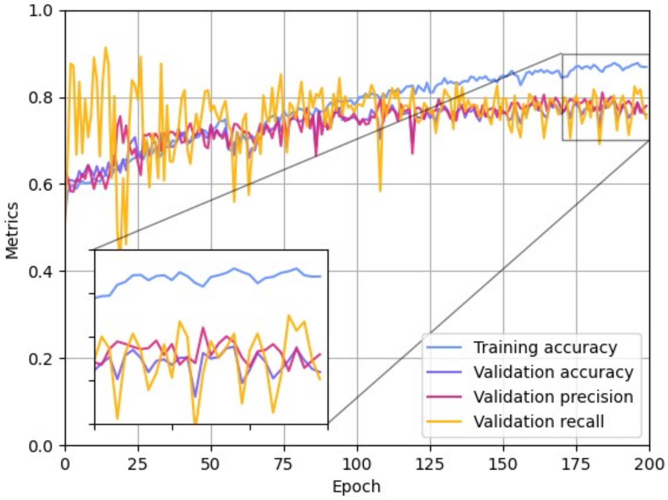


Figure 7.12: Metrics for CNN Model

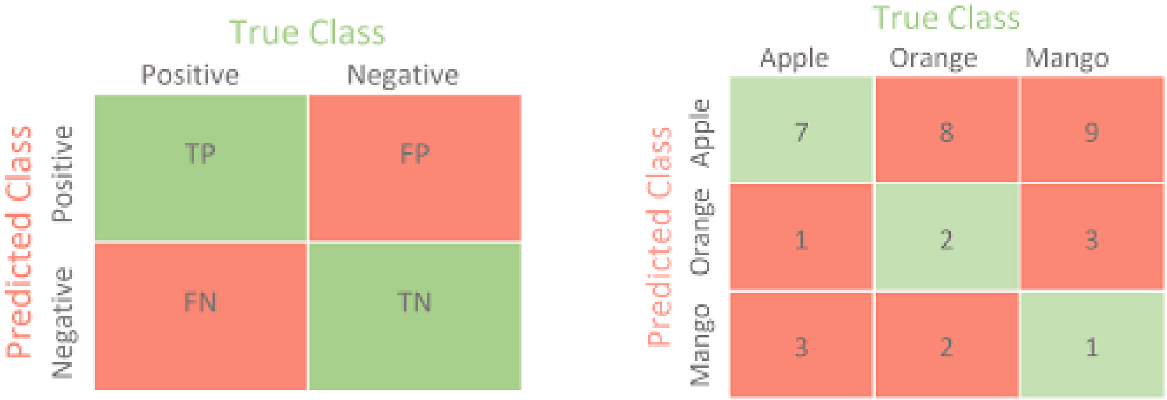
Based on the results obtained after 200 epochs, the model demonstrated commendable performance. During the training phase, it achieved an accuracy of 86.9%. Moving on to the validation phase, the model maintained a solid accuracy of 75.9%, indicating its ability to generalize well to unseen data. Furthermore, it exhibited a precision of 77.9%, which implies that a significant portion of the predicted positive instances were indeed correct. Additionally, the model displayed a recall of 75.1%, indicating its capacity to correctly identify a substantial proportion of the actual positive instances in the dataset. Overall, an F1 score of 0.765 was achieved, proving that these results showcase the model's strong performance and its potential for effective utilization in basic damage detection.

7.4.3 Second Model: YOLOv7

In order to avoid unnecessary maintenance procedures, distinguishing dirt from damage is essential. Thus, the second model shall consist of 3 distinct classes: dirt, damage, and background (no object was found). Furthermore, it was intriguing to investigate the feasibility of generating similar YOLO identification boxes for all the classes. After a thorough analysis, a research project regarding the usage of YOLOv7 was found as fit for a second iteration of the model, as "it surpasses all known object detectors in both speed and accuracy", according to the article's authors [74]. For a high-value dataset, YOLOv7 is capable to identify which class, dirt or damage, is present, as well as its exact location on the image. To put it differently, it provides an image similar to Figure 7.10. Due to the complexity of the model, such will not be explained further.

Algorithm Implementation

In order to implement the model, the wind turbine dataset had to be shuffled, avoiding any inherent ordering or patterns. Additionally, only the YOLO-labeled images were used, reducing the original dataset to only 2995 images. Subsequently, it was split into training, validation, and testing subsets, containing 85%, 7.5%, and 7.5% of the dataset, respectively. In order to evaluate the model, a multiclass confusion matrix shall be assembled. Such table is a matrix representation of the prediction results on the test subset. This shall be crucial to retrieve all the important metrics of the YOLOv7 model.



(a) Binary Confusion Matrix [75]

(b) Multiclass Confusion Matrix [75]

Figure 7.13: Types of Confusion Matrices

Figure 7.13a displays a binary confusion matrix, which represents two classes commonly known as positive and negative. Inside the matrix, four metrics can be found: True Positive/True Negative (TP/TN) represents the number of predictions which were predicted positive/negative and were in reality positive/negative. On the contrary, False Positive/False Negative (FP/FN) refers to the number of predictions which were predicted incorrectly as positive/negative.

Performance analysis and results

The metrics applied for the CNN model can also be used here, as they can be derived from the provided confusion matrices:

Accuracy: $\frac{TP+TN}{TP+FP+FN+TN}$ **Precision:** $\frac{TP}{TP+FP}$ **Recall:** $\frac{TP}{TP+FN}$ **F1 Score:** $\frac{2\cdot\text{Precision}\cdot\text{Recall}}{\text{Precision}+\text{Recall}}$

Although the YOLOv7 model has three classes, just as Figure 7.13b, the "Background" class has no assigned value *per se*. Thus, a value for each metric can be obtained by individually considering "Damage" and "Dirt" classes. For instance, if the team solely consider the "Apple" class from Figure 7.13b, it will retrieve the subsequent values:

$$TP = 7 \qquad TN = 2+3+2+1 = 8 \qquad FP = 8+9 = 17 \qquad FN = 1+3 = 4$$

Accuracy ≈ 0.416 **Precision** ≈ 0.292 **Recall** ≈ 0.636 **F1 Score** $= 0.4$

After calculating each parameter by class, the unweighted mean of each metric shall be retrieved, providing the macro metrics of the entire model. Thus, by applying the same methodology to the YOLO dataset, Figure 7.15a is obtained. Furthermore, accuracy is unfortunately not graphed by the YOLOv7 model. Therefore, only the accuracy retrieved for the best F1 score shall be computed, by means of a confusion matrix (Figure 7.15b). Finally, an example of the model's performance is shown in Figure 7.14.

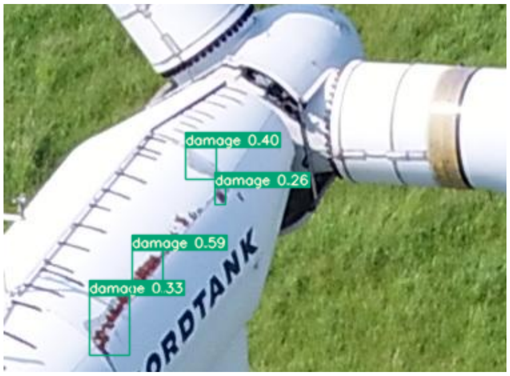
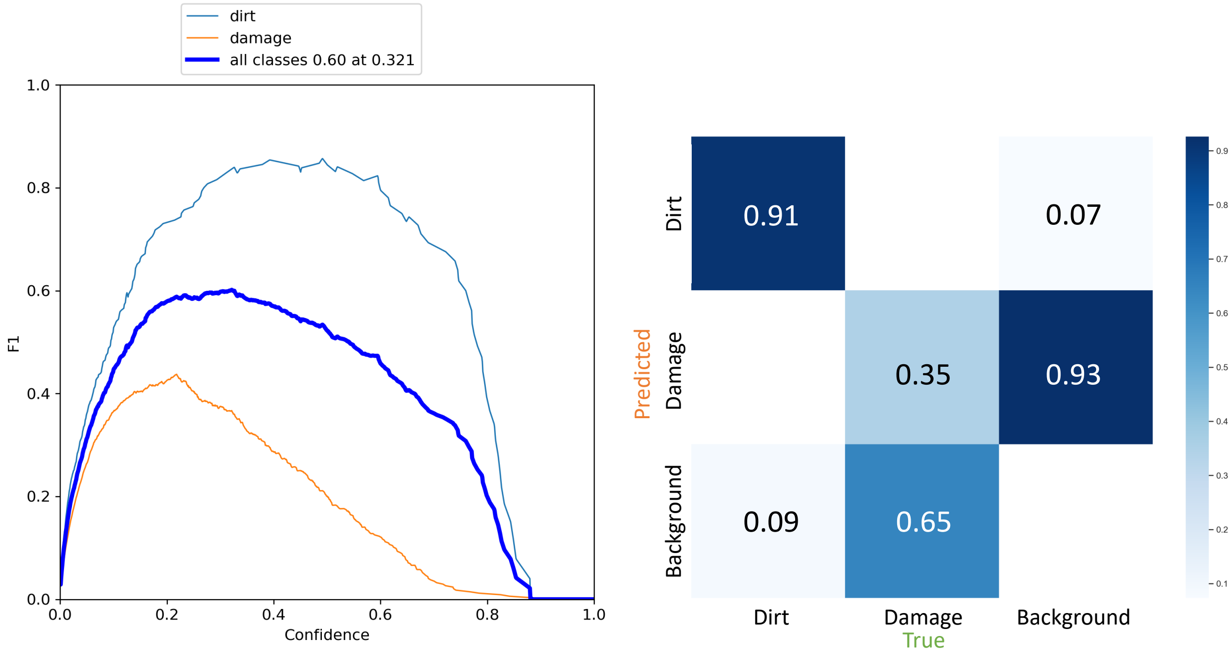


Figure 7.14: Test Example



(a) F1 Score

(b) Confusion Matrix

Figure 7.15: YOLOv7 Outputs

Thus, as it can be seen, the model is able to recognise dirt very well, having a much higher F1 score than damage. Such may be due to the larger presence of dirt compared to damage in the training data.
Moreover, the reason why the "Background" class is never perceived as itself is because there are no YOLO labels for the "Background" class. As a matter of fact, it is the lack of a label which makes the model consider it part of the background, i.e., a label will never be predicted as the latter. This explains why the last diagonal element of Figure 7.15b is equal to zero.
Also, the mean F1 score seems to peak at 0.6 for a model confidence of 32% and decays to zero for a confidence of 86%. This indicates that the model's reliability decreases as the confidence threshold increases, meaning that the model is unfortunately unreliable and cannot be used with high certainty at the moment. Unfortunately, the model's performance is relatively low, with an accuracy of 42%. This suggests that the model's predictions align with the correct outcomes only 42% of the time, indicating that there is room for improvement in its predictive capabilities. Most likely, the major reason for such low accuracy is the lack of data, as YOLOv7 is a model commonly used with datasets containing around 50,000 images, compared to the 2,995 images of the current dataset. Another reason could be the quality of the YOLO labels not being good enough for the model to correctly predict what is considered as background, dirt, or damage.

8: Financial Overview

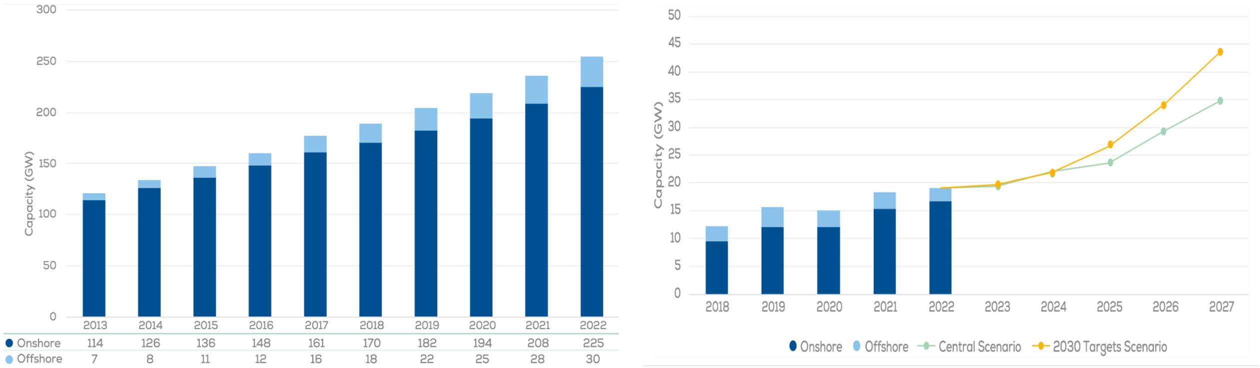
To create a profitable business, an extensive analysis has to be made of all the costs that might occur. Additionally, the potential market gap and achievable market has to be evaluated as well. All of these aspects will be discussed in this chapter in more detail.

8.1 Market volume for the product

Author: Anton

TurbEye currently focuses its attention on the offshore wind-farm market within Europe, rather than expanding globally. Therefore, the analysis will exclusively consider offshore turbines in Europe to gain a comprehensive understanding of the potential market size. As of 2022, Europe boasted a total wind energy capacity of 255 GW. Among this capacity, onshore turbines contributed 225 GW, while offshore turbines accounted for 30 GW. This offshore capacity is generated by 6,121 grid-connected but is expected to keep growing as more and more projects are initiated regarding offshore wind farms, hence increasing the market volume as well [76]. As can be seen from Figure 8.1b, a clear upwards trend is present in the installed wind power capacity and is expected to keep growing as can be seen in Figure 8.1b. This expected growth is dictated by what additional capacity has to be installed to meet Europe's goals for 2030 [77].

As all these turbines have to be inspected at least twice a year, the total market volume equates to 12,242 potential inspections. Again, this number is expect to grow significantly as more capacity has to be installed towards the future.



(a) Installed wind power capacity in Europe, 2013-2022

(b) New installations in EU - WindEurope's scenarios

Figure 8.1: Total wind capacity installed and to be installed

8.2 Achievable market share for the product

Author: Anton

As this market is a rather saturated market, striving newcomers are rare. However, TurbEye is confident that, due to its innovative approach, increasing the flight time by using a fuel cell instead of batteries, roughly 2000 inspections should be an achievable market share and is expected to grow 10% annually in the first years as TurbEye will establish a stronger position in the market and therefore attracting more costumers.

8.3 Revenue

Author: Anton

Taking into consideration all costs of the drone's operation, a competitive, yet profitable rental price should be established. This is estimated to be 800€ per inspection. However, it should be noted that the competitive nature of the current market makes it rather hard to get an accurate estimation of competitors' pricing, but it is believed that this is a significant reduction in price compared to competitors as one of the driving costs of off-shore inspections is being drastically decreased: the transportation cost. This equates to a revenue of 1.6 million euro's in the first year, after which it will increase with 10% a year as determined before.

8.4 Production cost of the product

Author: Anton

As discussed in section 3.2, most components are off-the-shelf components, hence no machining costs are present. However, the drone has to be assembled. This will be done by one of the team members at TurbEye and is assumed to take a 100 hours, with a salary of 80,000€/year and an average working year containing 2,000 hours, equating to a labor cost of 4,000€. Including the price of all the components, the total production cost results in 63,550€.

8.5 Operational cost

Author: Anton

The operational costs are divided into different sections to explain them into more detail. Note that throughout the years, an average inflation rate of 3.2% is taken into account and applied to all reoccurring costs [78].

Hydrogen cost

As the drone will use hydrogen as an energy source, the price of it should be taken into account. As discussed in chapter 9, TurbEye aims to use green hydrogen to be as sustainable as possible. The price of this type of hydrogen currently lies around 8 euro's per kilogram, and, on average, 0.120 kg is needed for a single trip. If indeed the achieved market share is 2000 inspections, and 4.8 inspections are performed on average per trip, which equates to 400€ in hydrogen costs per year.

Transportation Costs

As the drone is aimed to be operated on offshore locations, a boat will often be required to transport the drone. This mode of transportation, based on an interview held with an external expert (private), is rather expensive, coming at a cost of 10,000€ per day of using a boat. However, compared to battery operated drones, who require the boat to be present at all times to carry out the operation, the hydrogen drone only needs one boat trip to the wind-farm. For a total wind-farm, two boat trips are needed. Hence, this results in a boat operation cost of 20,000€ per farm. This is a drastic improvement compared to the competitors in the field. For example, if the time of a total wind-farm inspection would be ten days, this would cost competitors operating battery powered drones 100,000€ in boat transportation, compared to TurbEye who would only have a cost of 20,000€. This drastic reduction in fixed costs (80,000€) positions TurbEye as a highly attractive and competitive solution in the market.

Employee Cost

A competitive salary of 80,000€ is provided for each team member. For a team of ten, this equates to an annual salary cost of 800,000€. It is believed that attracting very capable engineers and paying them a competitive salary will be beneficial for the company in the long run.

Office Space Cost

The office space should be able to facilitate the 10 employees of TurbEye. An office space was found in Delft which provides the perfect work environment for the team, the price for the office is 120€/m^2/year, with an area of 269m^2 this comes down to 32,280€ per year [79].

Maintenance Cost

The maintenance cost of the drone is hard to predict because the maintenance needed is mainly caused by environmental factors and circumstances that TurbEye cannot control. However, a budget of 10% of the drone's cost is reserved for potential maintenance costs. For a drone price of 60,000€, this comes down to 6,000€ being reserved per year for smaller reparations. For larger expenses, the drone will have an insurance to cope with these costs, this is more discussed in section 8.5.

Drone Insurance

Since the drone is a very valuable piece of equipment, an insurance should be taken to cover with unexpected damages, theft, etc. A quote was made by Baloise [80] for an insurance of this kind. The proposed quote covers aspects like civil liability, CASCO (Casualty & Colision), theft, hacking of the drone, loss of inspection data, among others, all accompanied with respective deductibles. The total amount of the insurance premium comes down to 6861.31€ per operational year. An overview of the deductibles can be found in Table 8.1

Table 8.1: Insurance Coverage

Coverage	Deductible
Civil Liability	250 EUR
Geographical Extension	2,500 EUR
Casco	5% of insured value, min. 250 EUR
Theft	10% of insured value, min. 250 EUR and max. 750 EUR
Loss of Visual Material	500 EUR
Recovery Costs	2,500 EUR

Summary operational costs

An overview of all operational costs and revenues of the first operational year can be found in Table 8.2, and the same information given in the cash flow diagram in Figure 8.2

Table 8.2: Financial Information

Category	Item	Amount
Inspection [8.3]	Number of Inspections	2,000
	Price per Inspection	800 €
	Revenue	(+) 1,600,000 €
Hydrogen [8.5]	Hydrogen per kg	8 €
	Hydrogen per trip	0.120 kg
	Inspection/trip	4.8
	Hydrogen Cost	(-) 400.00 €
Transportation [8.5]	Cost of Boat	10,000 €
	Average Wind farm Size	100
	Number of Farms per Year	20
	Transportation Cost	(-) 400,000 €
Drone [3.2]	Drone Cost	(-) 59,993 €
Employees [8.5]	Number	10
	Salary	80,000 €
	Salary Cost	(-) 800,000 €
Office [8.5]	Rental Price per Month	2,690 €
	Office Cost	(-) 32,280 €
Insurance [8.5]	Insurance for Drone	(-) 6,861.31 €
Maintenance [8.5]	Maintenance cost	(-) 6,000 €

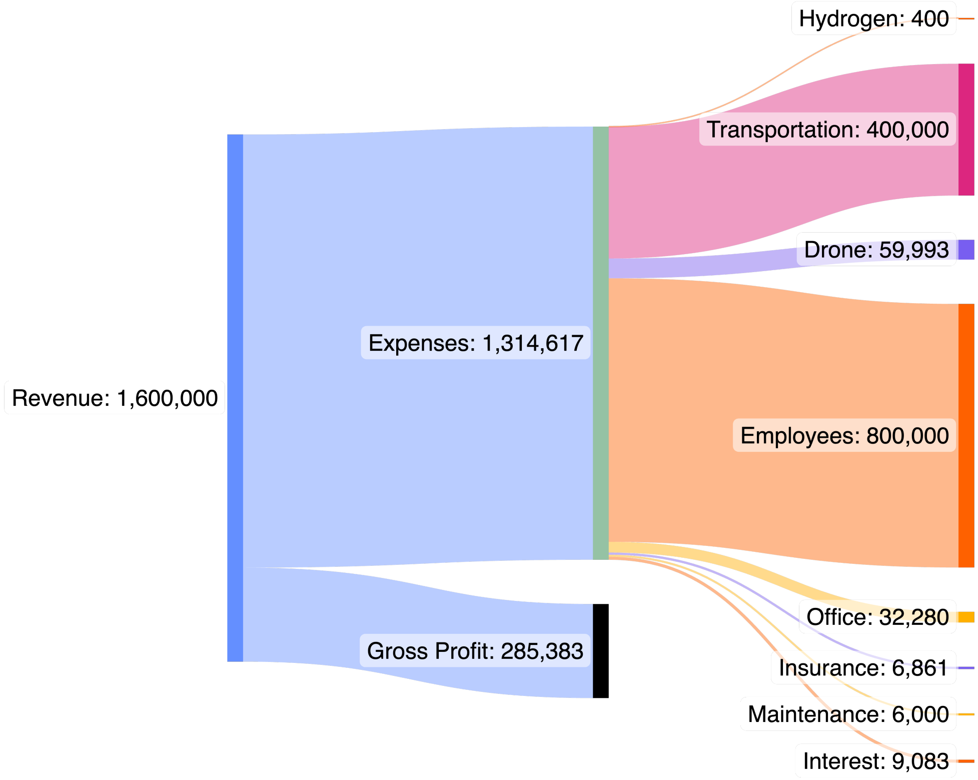


Figure 8.2: Cash flow diagram of revenue and expenses of the operations

8.6 Loan

Author: Anton

In order to have sufficient capital to start up the company and make the initial investment for the drone, a loan has to be taken. Since TurbEye will start its operations instantly after acquiring the drone, it is assumed that 125,000€ is sufficient to get the company started. It will cover the initial investment of the drone and will also provide sufficient cash to operate in the first months without relying on the payment of clients. The repayment of the drone will be done in the following manner.

Table 8.3: Loan Repayment Schedule

Year	To Pay	Capital	Interest	Residual Capital
Year 1	34,083.34 €	25,000.00 €	9,083.33 €	102,083.33 €
Year 2	32,083.33 €	25,000.00 €	7,250.01 €	77,083.33 €
Year 3	30,083.33 €	25,000.00 €	5,083.33 €	50,000.00 €
Year 4	28,083.34 €	25,000.00 €	3,249.99 €	25,000.00 €
Year 5	26,083.33 €	25,000.00 €	1,083.34 €	0.00 €

8.7 Expected Lifespan

Author: Anton

Naturally, the lifetime of the drone is an important aspect as this can be a limiting factor in the ROI. High-end drone models currently have a life expectancy that lies around 3 years. This is mainly due to the fact that the drone's batteries decay over the number of cycles the batteries goes through. However, since the TurbEye drone will make use of a power-cell the decay of batteries will not be the limiting factor anymore, which is why a higher life expectancy is assumed. For the business model of TurbEye, the drones expected lifetime is set at 5 years operational after which the drone would have to be replaced. This is a significant improvement compared to the competitors in the market.

8.8 Summary

Author: Anton

By adding the operational costs, revenue and cost of having a loan, a financial summary can be made. Note that each year an expected growth of 10% and an average inflation of 3.2% is assumed. Doing these calculations for 5 years (operational lifetime of drone), a total return on investment of 30.83% is obtained using Equation 8.1, which equates to a net profit of just over 3 million euros. Overall, it should be noted that this is a rough estimate and that predicting revenue in this field is rather hard due to its competitiveness and therefore limited available information. However, based on these calculations, it is concluded that it is indeed a profitable, compelling business opportunity.

$$\text{RoI} = \frac{\text{Net return on investment}}{\text{Total value of investment}} \times 100\% \tag{8.1}$$

Table 8.4: Financial Summary

Year	**Revenue (€)**	**Costs (€)**	**Profit (€)**	**RoI (%)**
Year 1	1,600,000	1,324,741.31	275,258.69	17.20
Year 2	1,760,000	1,301,025.05	458,974.95	26.08
Year 3	1,936,000	1,338,129.11	597,870.89	30.88
Year 4	2,129,600	1,376,755.03	752,844.97	35.35
Year 5	2,342,560	1,416,271.99	926,288.01	39.54
Total	9,768,160	6,756,922.49	3,011,237.51	30.83

9: Sustainable Development Strategy

For the design of a wind turbine inspection drone, a holistic approach was followed in which environmental, societal and economic impacts are considered. The team aims to minimise the negative impact of the drone on the environment, while maximising its benefits to the society and economy. For this end, the team derived design goals & choices from a sustainability framework developed by UN: Engineering for Sustainable Development [81].

9.1 Sustainable Choices

Author: Enes

The design goals derived from these principles and their corresponding design choices are described in Table 9.1.

Table 9.1: List of design goals derived from the Engineering for Sustainable Development Framework

Goal ID	Goal Name	Design Choice ID	Design Choice Name
SDS-01	Energy efficiency	SDS-ENERGY-01	Lightweight materials
		SDS-ENERGY-02	Autonomous path planning
		SDS-ENERGY-03	Route planning
		SDS-ENERGY-04	Post-flight processing
		SDS-ENERGY-05	Large range
SDS-02	Renewable energy source	SDS-RENEWABLE-01	Green hydrogen
SDS-03	Autonomous capabilities	SDS-AUTONOMY-01	Autonomous path planning
		SDS-AUTONOMY-02	Route planning
		SDS-AUTONOMY-03	Autonomous inspection
SDS-04	Recycling and waste reduction	SDS-RECYCLE-01	Modular design
		SDS-RECYCLE-02	Fail-safe philosophy
SDS-05	Environmental pollution	SDS-ENV-01	Hydrogen fuel cell
		SDS-ENV-02	Minimize transportation
SDS-06	Social responsibility	SDS-SOCIAL-01	Data responsibility

The design goals & choices are elaborated as follows:

- **Lightweight materials:** The drone will be made out of lightweight materials, and its components will be chosen with weight in mind to reduce weight and power requirements, thereby conserving energy. The use of graphene nanoplatelets (GNP) offers advantages such as an 8% higher elastic modulus, 6% higher tensile strength, and 40% higher impact strength compared to normal carbon fibre [82]. This means that a lower mass is needed to handle the load cases experienced by the drone.
- **Autonomous path planning:** The drones will have autonomous flying capabilities. This will ensure consistent performance of the drone throughout its flights. Additionally, this allows efficient navigation in dynamic conditions.
- **Route planning:** An optimized VRP algorithm will be used to generate routes to be followed to inspect turbines in a wind farm. This will reduce fuel usage, time spent as well as downtime. Compared to a simple heuristic algorithm, this results in fuel savings of around 5%, reducing the number of refuels by 10% and the total flight time by 3%, as seen in Table 7.2.
- **Post-flight processing:** The drone will not have real-time processing of inspection footage, but post-flight processing. This conserves useful energy during the flight and reduces the power requirements of the drone.
- **Green hydrogen:** The drone will be mainly powered by a hydrogen fuel cell, using green hydrogen as a source. Green hydrogen is a renewable and sustainable energy source, and utilizing it will reduce the carbon footprint of the drone. More information about power generation is given in section 9.2

- **Large range:** The hydrogen-powered drone will have higher endurance compared to conventional battery-powered drones which reduces refueling time and the number of trips made back and forth to the OSS for refueling. This means more inspections can be performed in a given amount of time, decreasing the need of a swarm of drones.
- **Autonomous inspection:** The drone will perform inspections autonomously. This prevents safety risks to the maintenance crew caused by dangerous working conditions.
- **Modular design:** The drone's components will have a modular design, allowing easy replacement, repair and re-purposing of the components. This way, the disposal of the entirety of the drone will usually not occur, and the irreparable components of the drone will be replaced. This helps significantly reduce waste and re-using of components.
- **Fail-safe philosophy:** The drone will be designed with fail-safe design philosophy and redundancies implemented into the drone as well as made from durable materials. This will help use the drone for a long time and get the most out of the resources used.
- **Life cycle analysis:** Life cycle analysis of the drone will be conducted, to assess and minimize the environmental impact. The analysis will consider various factors like manufacturing processes, re-purposing possibilities, end-of-life recycling options and disposal methods of components.
- **Hydrogen fuel cell:** The drone will have a hydrogen fuel cell that does not emit any pollutants, except for water. This helps prevent environmental pollution.
- **Minimize transportation:** The drone will require only one two-way trip to the wind farm and won't require trips between wind turbines (which is the case for conventional inspections), thanks to the drone's high range. This reduces the carbon footprint of the operations due to a significantly less amount of transportation. As an example, inspecting the Hornsea Project 2 would require the boat to travel 535 km for the conventional case, whereas the boat would only travel 180 km using TurbEye's solution, which is almost a three fold less.
- **Data responsibility:** To limit the recording of private individuals, recordings will be made only in the vicinity of a wind turbine. Individuals in the vicinity of the wind farm, mainly the crew working at the OSS, will also be informed beforehand about the drone's recording capabilities.

Sustainable design choices that are going to be implemented in the future design steps are as follows:

- **Life cycle analysis:** Life cycle analysis of the drone will be conducted, to assess and minimize the environmental impact. The analysis will consider various factors like manufacturing processes, re-purposing possibilities, end-of-life recycling options and disposal methods of components.
- **Carbon calculations:** Carbon emission calculations of the drone, its components and the operations will be conducted to better understand the environmental implications.
- **Collision avoidance system:** The drone will be equipped with a collision avoidance system, which will prevent collisions, including with the wind turbine. This prevents potential damage to the energy infrastructure.

These design goals also promote some of the UN's 17 Sustainable Development Goals [83], as seen in Table 9.2.

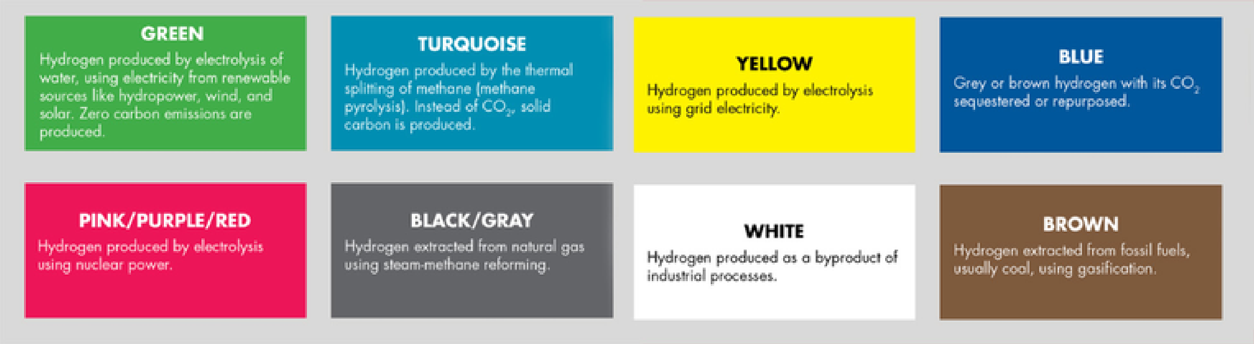
Table 9.2: List of UN Sustainability Goals that are promoted within the design

UN Sustainability Development Goal	SDS	Explanation
7: Affordable and Clean Energy	SDS-02	Promoting wind turbines by making it more competitive in the energy markets and accelerating its adoption
8: Decent Work and Economic Growth	SDS-05 SDS-06	Improving working conditions and safety of maintenance crew and reducing cost and time of maintenance, making wind turbines more competitive and contributing to economic growth
9: Industry, Innovation and Infrastructure	SDS-01 SDS-03	Promoting a lightweight, autonomous design, increasing efficiency and promoting innovation in the wind energy industry
12: Responsible Consumption and Production	SDS-01 SDS-02 SDS-04	Promoting modular design waste reduction and sustainable manufacturing methods
13: Climate Action	SDS-01 SDS-02 SDS-05	Promoting low-carbon propulsion and reducing the carbon footprint of logistics operations

9.2 Hydrogen Production

Author: Arham

To be sustainable, it is imperative to know the emissions caused by the choice of fuel. Although hydrogen does not cause any direct carbon emissions, it does release carbon emissions during production, compression and transport. Therefore, all of these factors should be kept in consideration to be as sustainable as possible.



GREEN
Hydrogen produced by electrolysis of water, using electricity from renewable sources like hydropower, wind, and solar. Zero carbon emissions are produced.

TURQUOISE
Hydrogen produced by the thermal splitting of methane (methane pyrolysis). Instead of CO_2, solid carbon is produced.

YELLOW
Hydrogen produced by electrolysis using grid electricity.

BLUE
Grey or brown hydrogen with its CO_2 sequestered or repurposed.

PINK/PURPLE/RED
Hydrogen produced by electrolysis using nuclear power.

BLACK/GRAY
Hydrogen extracted from natural gas using steam-methane reforming.

WHITE
Hydrogen produced as a byproduct of industrial processes.

BROWN
Hydrogen extracted from fossil fuels, usually coal, using gasification.

Figure 9.1: Different Hydrogen Production Methods [84]

About 99.6% of the hydrogen produced in the world is derived from hydrocarbons, mainly consisting of Grey and Brown hydrogen. Green hydrogen, which accounts for only 0.1% of global production, is significantly more expensive compared to hydrocarbon-based alternatives. An explanation of different types of hydrogen can be seen in Figure 9.1

Since all the other hydrogen production methods still rely on fossil fuels to some extent, the two main viable options remaining are pink and green hydrogen. Green hydrogen can be produced from various sources, including solar, wind, and hydropower. Considering that the inspections are being conducted on wind turbines, it is logical to utilize the available wind energy on-site.

Converting electricity to hydrogen requires substantial investments in electrolysis plants, making it impractical to build a hydrogen plant solely for the purpose of refuelling drones. However, as offshore wind farms move further from shore and deeper into the ocean, the losses in the cable corridor become more significant [85]. In such cases, it becomes necessary to consider on-site energy conversion into hydrogen and transporting the hydrogen by ship.

This approach aligns well with the use of hydrogen drones, as they can be refuelled on-site. Depending on the size of the wind farms, it may even be feasible for the drone to remain on-site throughout the year.

To incentivize on-site hydrogen production, offering a discounted price to wind farms that incorporate a hydrogen plant can be a viable option. This discounted price would help reduce the costs incurred by the wind farm operator, making the investment in hydrogen production more economically attractive. By providing this financial incentive, it would encourage the integration of hydrogen plants within wind farms and promote the adoption of cleaner energy solutions.

Furthermore, referring to the IPCC WP3 report [86], it is evident that wind energy has one of the lowest life cycle emissions compared to other energy sources. Although the levelled cost of wind energy is higher than that of non-renewable options like coal and gas, it remains competitive with the cost of nuclear energy and is significantly cheaper than solar energy. This indicates that if hydrogen is produced on-site using wind energy, it would result in some of the cleanest and least emission heavy hydrogen available. Additionally, by producing hydrogen on-site, the costs and emissions associated with transportation can be minimized.

One potential concern is whether using green hydrogen would impact profitability. However, considering that less than 4 kg of hydrogen is required to inspect the entire Hornsea 2 wind farm, the cost of hydrogen can be a maximum of 40 euros, which when compared to other operational expenses. Therefore, even if a premium is paid for green hydrogen, it would not have a significant effect on profitability.

10: Conclusion and Project Outlook

10.1 Conclusion

Author: Louis

It can be concluded that the wind turbine market is continuously growing, with increasing resources allocated towards sustainable and renewable energy. Offshore wind energy is predicted to become increasingly significant in the near future. As a result, there is a growing need for automated inspections, leading the team to focus on the development of a fully autonomous drone propelled by hydrogen for complete offshore wind turbine inspections. The drone can take off from either the wind farm substation or the shore if the wind farm is sufficiently close-by.
The main strengths of the drone include its increased range, making it suitable for inspecting entire wind farms. With full autonomy, the drone can perform hands-off inspections without the need for human intervention. This allows TurbEye to offer cheaper and faster maintenance inspections.
The design of the drone has been finalized as a co-axial octocopter powered by hydrogen. It incorporates three different inspection methods, namely visual, passive thermography, and geometry-based inspections, to provide a comprehensive assessment of the wind turbine's condition. The drone is engineered to have a range of 280 [km] using a full tank of 120 [grams] of hydrogen, enabling it to fly for up to 3.5 [hours]. With these specifications, the drone can efficiently inspect the Hornsea 2 wind farm, which consists of 165 turbines, in just 35 trips when taking off from the substation. The main specifications are detailed in Table 10.1.

Table 10.1: Top Level Specifications

Characteristic	Value
Range	280 [km]
Endurance	3.5 [hrs]
Mass	11.8 [kg]
No. of inspection per trip	5 inspections[1]
Hydrogen Mass	0.12 [kg]
Fuel Cell Power Output	2000 [W]
Thrust per engine	6.5 [kg]

10.2 Future Development

This section focuses on future project design and development. In subsection 10.2.1 recommendations are provided for the further design of the drone as, due to the limited time available during the DSE, the team has not been able to finish the detailed design phase. Having the development recommendations implemented, the project can transition towards the execution phase in which the drone can be produced, tested and operated. Finally, also an end-of-life retirement plan is provided for the drone. These activities are outlined in Figure 10.1. The post-DSE Gantt Chart can be found after Appendix A.

10.2.1 Design Recommendations

Author: Enes, Arham, Louis

Regarding the design of the drone, various improvements can be made with respect to the various subsystems. The most important recommendations are listed below:

---

[1] Average number of inspections per trip tested on Hornsea 2

Control

Significant improvements can be made mainly to the control model of the drone, which is crucial for achieving precise and stable flight operations.

1. It is advised to develop a non-linear control model to increase robustness, improve maneuverability and adaptability to different flight conditions.
2. The way-point following algorithm will be enhanced to allow for smoother trips, reducing the energy consumed to travel the path and creating a more realistic, less oscillatory trajectory that the drone would be able to follow.
3. The routing algorithm used to generate the optimized route to be followed within a wind farm will also be integrated into the guidance subsystem to provide waypoints for the drone to follow.
4. Since safety measures are paramount for this operation, emergency control procedures are also to be developed for cases in which engine(s) is inoperative. This involves developing control algorithms that can detect engine failure and adapt control inputs accordingly to ensure controllability and safety during such cases. Moreover, a return-to-base flight mode will be developed to have the drone return to base in critical conditions.
5. From the final CAD, a scaled model can be acquired to obtain the aerodynamic coefficients inside a wind tunnel to further improve the model.

Propulsion

Regarding the propulsion system, the main improvements possible are regarding the simulation of the dynamic conditions, modelling interdependencies and updating performance parameters:

1. The propulsion can mainly be further developed through simulation of the dynamic conditions. This includes modeling changing weight during the flight, changing environment conditions with height and implementing the dynamic response of the components such as motors, ESCs as well as the fuel cell.
2. Make the interdependencies like the chassis weight dynamic based on the size and weight of the engines.
3. Revise the performance parameters, such as range and endurance, based on the final model characteristics, including the updated drag coefficient, dimensions, and the shifted center of gravity.

Structures

For the structures system, there are several areas where significant improvements can be made. These recommendations aim to optimize the Finite Element Analysis (FEA), implement design iterations based on FEA results, and create a more integrated design. The main recommendations are as follows:

1. Enhance the FEM model by incorporating proper modeling of various connections, such as adding contact constraints and bolted connections.
2. Consider running dynamic simulations, including scenarios like impact landings, to achieve more realistic analysis results. Prioritize improving meshing techniques to handle the computational expense associated with dynamic simulations, as the current FEM model contains 635,000 nodes.
3. Based on the analysis, focus on addressing deformations primarily caused by connectors rather than tubes in the chassis. Explore the possibility of utilizing additive manufacturing with aluminium to produce more robust connectors.
4. Evaluate the need to reinforce the baseplate, particularly in load case A, where significant deflection was observed. However, before implementing any modifications, improve the Finite Element Analysis by accurately modeling the inertia of the hydrogen tank and fuel cell, which are currently represented as point masses. Additionally, consider replacing glued constraints with bolted connections where appropriate, as the use of pre-tensioned bolted connections may enhance the overall performance.

5. Recognize that the current prototype design with low production rates utilizes off-the-shelf components and a chassis structure that facilitates easy manufacturing and assembly. If production rates increase, consider a future iteration adopting a monocoque design, which can reduce weight and improve endurance.
6. Investigate the potential of utilizing advanced and cost-effective additive manufacturing methods for topology optimization of the chassis. This approach can further optimize the design by achieving optimal structural configurations while reducing weight.
7. Enhance the CAD model of the drone by including all mounts, wires, and other connections.

Deep Learning Model

Regarding the YOLOv7 model, it was concluded that it was still not suitable for utilisation in real-life scenarios, due to its low accuracy and F1 score. The following improvements to the AI model are therefore advised:

1. The model should be improved by using a more complex neural network layer layout initially, but also by the increasing datasets acquired through actual wind turbine inspections throughout the drone's operation.
2. The model is advised to be trained through a more computationally powerful workstation, in order to decrease computing time.

10.2.2 Production Phase

In this phase, the drone enters the production stage. Commercial off-the-shelf (COTS) components will be purchased, while the remaining components will be manufactured in-house. The acquired components will be transported to the assembly area, where the final assembly of the drone will take place. Following the assembly process, quality assurance procedures will be carried out to ensure the drone meets the required standards. Once the quality assurance is complete, the drone will undergo testing to verify its performance and functionality.

10.2.3 Testing Phase

In this phase, the drone will undergo a series of tests to validate its performance and ensure that it meets the intended requirements and expectations. The structural integrity of the drone will be tested under critical loading cases to verify its rigidity. Control inputs will be applied to the drone within a controlled arena to assess its response and manoeuvrability. Performance tests will be conducted to ensure that the drone meets the specified performance requirements. Additionally, the drone will be subjected to extreme operating conditions such as extreme temperatures, humidity, rain, and hail to evaluate its functionality under these challenging environments. Integration tests will be performed to confirm that all the electrical components work together seamlessly. Once the testing phase is successfully completed, the drone will be ready for operational use.

10.2.4 Operation Phase

During the operational phase, several actions will be taken to facilitate the drone's use and maintenance. These include obtaining an all-risk insurance policy to cover any potential risks associated with the drone. Additionally, a large hydrogen storage tank will be purchased for the refueling station, ensuring a reliable source of fuel for the drone. A landing pad will also be acquired to provide a designated area for the drone to land safely. Simultaneously, the main office will actively engage with various companies to secure clients for inspection services. The drone, along with the large storage tank, will be transported to the clients' locations to conduct the inspections. Once the inspections are completed, the drone will be returned for inspection and maintenance. If any repairs are necessary, both the drone and its components can be repaired individually due to the modular design. However, in the event that the drone is deemed irreparable as a whole, it will be retired from service.

10.2.5 Retirement Phase

If the drone is determined to be irreparable, it will be returned to the repair shop where it will be disassembled. Any components that are still functional and deemed suitable for reuse in future drone production or repairs will be set aside. However, if the components are not expected to be reused, they will be sold. Given that most of the components are commercially available off-the-shelf (COTS) items, there is a market for them. In the case that certain components are not suitable for repair or reuse, recycling agencies will be contacted to ensure proper recycling procedures.

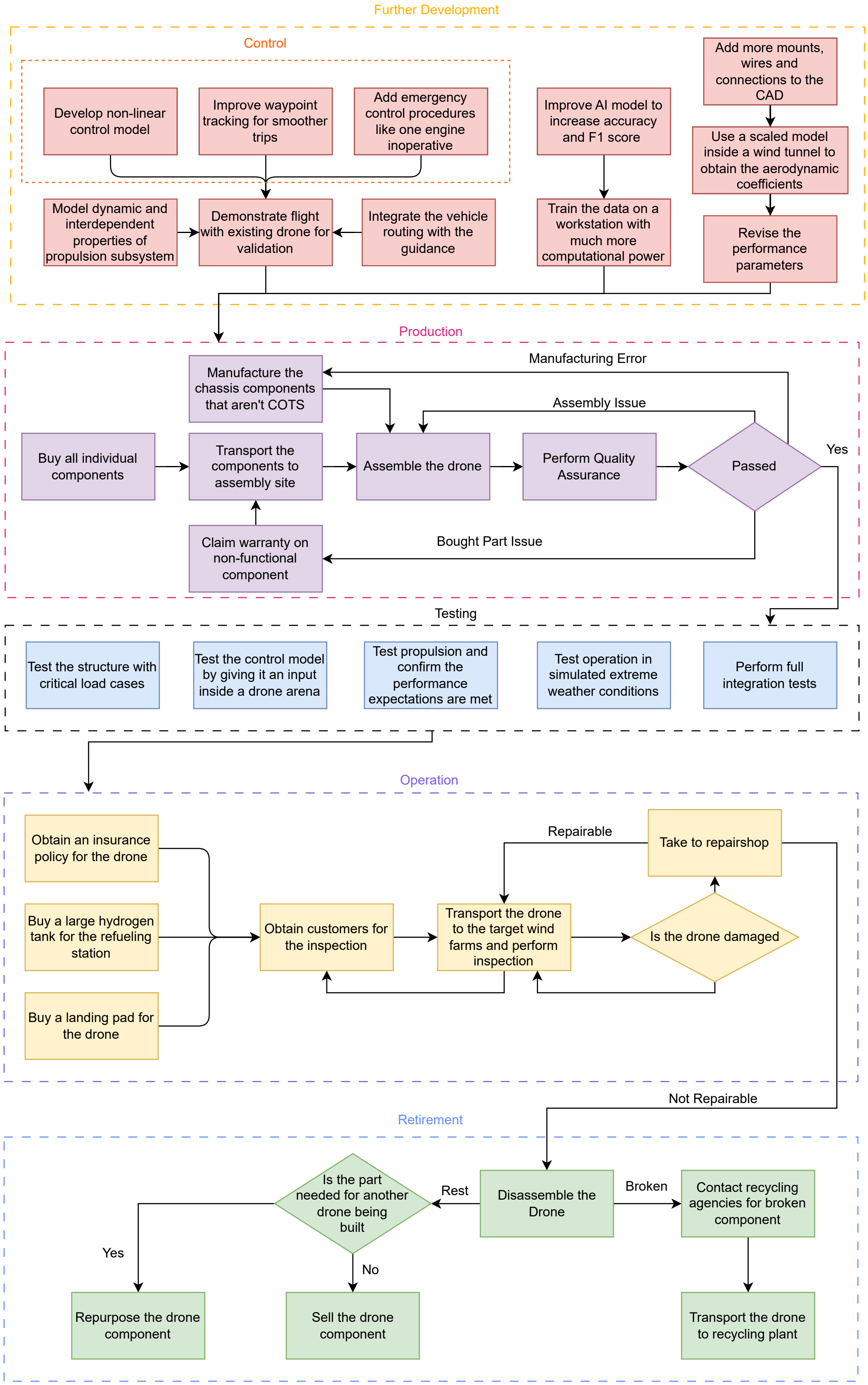


Figure 10.1: Project Design and Development Logic

Bibliography

[1] Ember, *Electricity Data Explorer*, Retrieved on 4 May 2023. [Online]. Available: https://ember-climate.org/data/data-tools/data-explorer/.

[2] P. Bojek, "Wind electricity," IEA, Tech. Rep., Sep. 2022. [Online]. Available: https://www.iea.org/reports/wind-electricity.

[3] M. Hagenbeek, S. van den Boom, N. Werter, *et al.*, "The blade of the future: wind turbine blades in 2040," TNO, Tech. Rep., May 2022. [Online]. Available: https://publications.tno.nl/publication/34639372/T881x3/TNO-2021-R12246.pdf.

[4] Voliro AG, *GRAVITY-DEFYING ROBOTICS*, Accessed on 4 May 2023. [Online]. Available: https://voliro.com.

[5] Skyspecs, *Simplify Renewable Energy Asset Management*, Accessed on 4 May 2023. [Online]. Available: https://skyspecs.com.

[6] Sulzer & Schmid Laboratories AG, *Powerful data analytics and autonomous blade inspections*, Accessed on 4 May 2023. [Online]. Available: https://www.sulzerschmid.ch.

[7] BladeInsight, *Allying powerful technology with blade expertise to optimize O&M*, Accessed on 4 May 2023. [Online]. Available: https://www.bladeinsight.com.

[8] Smulders Group, *Passionate about steel*, Retrieved on 3 May 2023. [Online]. Available: https://www.smulders.com/en/.

[9] Insider News, *How Wind Turbine Technicians Risk Their Lives to Keep Blades Spinning | Risky Business*, Accessed on 21 June 2023. [Online]. Available: https://www.youtube.com/watch?v=bNBQKlHt6rg%5C&t=38s.

[10] H3 Dynamics, *H3 Dynamics website*, Accessed on 12 June 2023. [Online]. Available: https://www.h3dynamics.com.

[11] H3 Dynamics, *High performance hydrogen fuel cell systems*, Accessed on 21 June 2023. [Online]. Available: https://www.h3dynamics.com/hydrogen-fuel-cell-systems-for-air-mobility.

[12] Voltaplex, *Voltaplex LiPo Battery Pack*, Accessed on 21 June 2023. [Online]. Available: https://voltaplex.com/lipo-battery-1800mah-wtuu.

[13] MELIFE, *4 Pack 5A DC-DC Adjustable Buck Converter*, Accessed on 21 June 2023. [Online]. Available: https://www.amazon.com/Adjustable-Converter-MELIFE-1-25-36V-Regulator/dp/B08BJSR75Z.

[14] LIVOX, *Mid-360*, Accessed on 11 May 2023. [Online]. Available: https://www.livoxtech.com/mid-360.

[15] *BMP390*, Bosch. [Online]. Available: https://www.bosch-sensortec.com/products/environmental-sensors/pressure-sensors/bmp390.

[16] *MicroStrain 3DM-CV7 Series*, Parker Hannifin Corporation. [Online]. Available: https://www.microstrain.com/sites/default/files/8400-0141_3DM_CV7_Datasheet.pdf.

[17] *M100 Mini*, HGLRC. [Online]. Available: https://www.hglrc.com/products/hglrc-m100_mini-gps.

[18] *PX4 User Guide - CubePilot Cube Orange+ Flight Controller*, Accessed on 21 June 2023, PX4. [Online]. Available: https://docs.px4.io/main/en/flight_controller/cubepilot_cube_orangeplus.html.

[19] 3DXR, *3DXR Cube Orange+ Listing*, Accessed on 21 June 2023. [Online]. Available: https://www.3dxr.co.uk/autopilots-c2/the-cube-aka-pixhawk-2-1-c9/cube-autopilot-and-combos-c10/cubepilot-the-cube-orange-imu-v8-standard-set-with-ads-b-carrier-board-p5328.

[20] *FTM30 Radio Module User Manual*, V1.1, SIYI Techology Co.,Ltd., Shenzen, Feb. 2021. [Online]. Available: https://discuss.ardupilot.org/uploads/short-url/tnkYUwyqKYUXSFmPduVjQHH0E1F.pdf.

[21] SIYI, *SIYI FM30 Radio Module 2.4G Transmitter*, Accessed on 21 June 2023. [Online]. Available: https://shop.siyi.biz/products/siyi-fm30?VariantsId=10559,10561.

[22] *MotionCam-3D L+*, Photoneo. [Online]. Available: https://www.photoneo.com/products/motioncam-3d-l-2/.

[23] Schweiger Shop, *Motioncam-3D Listing*, Accessed on 21 June 2023. [Online]. Available: https://www.schweiger-shop.de/motioncam-3d.html.

[24] *Zenmuse H20(T)*, dji Enterprise. [Online]. Available: https://enterprise.dji.com/zenmuse-h20-series/specs.

[25] DroneLand, *Drone Land webpage*, Accessed on 21 June 2023. [Online]. Available: https://droneland.nl/.

[26] Sony, *Sony TOUGH-G UHS-II 128GB SD Card*, Accessed on 21 June 2023. [Online]. Available: https://www.amazon.com/Sony-Performance-Blazing-SF-G128T-T1/dp/B07H9GDFRW?th=1.

[27] *T-motor - the safer propulsion system*, Accessed on 09 June 2023. [Online]. Available: https://uav-en.tmotor.com/.

[28] Koninklijk Nederlands Meteorologisch Instituut, *Dutch Offshore Wind Atlas*, Accessed on 21 June 2023. [Online]. Available: https://www.dutchoffshorewindatlas.nl/.

[29] Tyto Robotics Inc., *Tyto Robotics motor and propeller database*, Accessed on 11 May 2023. [Online]. Available: https://database.tytorobotics.com/.

[30] *Xoar*, Accessed on 09 June 2023. [Online]. Available: https://www.xoarintl.com/.

[31] Q. Quan, *Introduction to multicopter design and Control*, 1st ed. Springer Singapore, Jun. 2017. [Online]. Available: https://doi.org/10.1007/978-981-10-3382-7.

[32] J. Jeong, H. Shi, K. Lee, and B. Kang, "Improvement of Electric Propulsion System Model for Performance Analysis of Large-Size Multicopter UAVs," *Applied Sciences*, vol. 10, no. 22, Nov. 2020. [Online]. Available: https://doi.org/10.3390/app10228080.

[33] Z. Huang, J. Shen, S. H. Chan, and Z. Tu, "Transient response of performance in a proton exchange membrane fuel cell under dynamic loading," *Energy Conversion and Management*, vol. 226, Dec. 2020. [Online]. Available: https://doi.org/10.1016/j.enconman.2020.113492.

[34] A. Hendrickx, E. B. Koca, J. van Sommeren, *et al.*, "Designing an autonomous drone for wind turbine maintenance inspection: Midterm report," Technische Universiteit Delft, Tech. Rep., Apr. 2023.

[35] K. Almaz, F. de Vries, M. Lentner, and P. Nae, "Trinity Exercise report," Technische Universiteit Delft, Tech. Rep., Oct. 2022.

[36] TWI-Global, *WHAT IS PETG? (EVERYTHING YOU NEED TO KNOW)*, Accessed on 21 June 2023. [Online]. Available: https://www.twi-global.com/technical-knowledge/faqs/what-is-petg.

[37] *DuPont Nomex 818 Technical Data Sheet*, DuPont. [Online]. Available: https://www.dupont.com/content/dam/dupont/amer/us/en/safety/public/documents/en/DPT16_21668_Nomex_818_Tech_Data_Sheet_me02_REFERENCE.pdf.

[38] A. Jensen and K. Hamilton, *Connections (linear contact) - a seminar for femap v11.2.2 users*, Siemens, 2016. [Online]. Available: https://appliedcax.com/support-and-training/technical-online-seminars/seminars/Femap-and-NX-Nastran-Technical-Seminar-Connections-Linear-Contact.pdf.

[39] *NX Nastran User's Guide*, Siemens, 2013. [Online]. Available: https://docs.plm.automation.siemens.com/data_services/resources/nxnastran/10/help/en_US/tdocExt/pdf/User.pdf.

[40] G. Laird and M. Breedlove, *Dynamic Analysis User Guide*, Siemens, 2019. [Online]. Available: https://appliedcax.com/docs/femap/Dynamic-Analysis-User-Guide-A-user-guide-for-FEMAP-and-NX-Nastran-users.pdf.

[41] *Matlab version: 9.13.0 (r2023a)*, The MathWorks Inc., Natick, Massachusetts, United States, 2023. [Online]. Available: https://www.mathworks.com.

[42] A. Kaviyarasu, *Fundamentals of attitude estimation*, Accessed on 09 June 2023. [Online]. Available: https://mitindia.edu/images/pdf/avionics_ppt/Fundamentals%5C%20of%5C%20attitude%5C%20Estimation.pdf.

[43] *Mid-360*, Livox. [Online]. Available: https://www.livoxtech.com/mid-360.

[44] M. Brazinskas, S. D. Prior, and J. P. Scanlan, "An Empirical Study of Overlapping Rotor Interference for a Small Unmanned Aircraft Propulsion System," *Aerospace*, Oct. 2016. [Online]. Available: https://doi.org/10.3390/aerospace3040032.

[45] C. M. Simões, "Optimizing a Coaxial Propulsion System to a Quadcopter," Dept. Engenharia Mecânica, Instituto Superior Técnico, Av. Rovisco Pais, 1049-001 Lisboa, Portugal, Tech. Rep. [Online]. Available: https://fenix.tecnico.ulisboa.pt/downloadFile/563345090412782/Resumo.pdf.

[46] D. Xin, J. Zeng, and K. Xue, “Surrogate drag model of non-spherical fragments based on artificial neural networks,” *Powder Technology*, vol. 404, May 2022. [Online]. Available: https://doi.org/10.1016/j.powtec.2022.117412.

[47] J. F. Moxnes, Ø. Frøyland, I. J. Øye, *et al.*, “Projected area and drag coefficient of high velocity irregular fragments that rotate or tumble,” *Defence Technnology*, vol. 13, pp. 269–280, Aug. 2017. [Online]. Available: https://doi.org/10.1016/j.dt.2017.03.008.

[48] *Von karman wind turbulence model (continuous)*, The MathWorks, Inc. [Online]. Available: https://nl.mathworks.com/help/aeroblks/vonkarmanwindturbulencemodelcontinuous.html.

[49] G. G. A. Venkitachalam, “High Altitude Wind Resource Assessment - A study of the North Sea wind conditions using the Dutch Offshore Wind Atlas,” TNO, Tech. Rep., Jan. 2020. [Online]. Available: https://www.dutchoffshorewindatlas.nl/publications/publications/2020/01/01/internship-report---gopan-gopalan-achary-venkitachalam.

[50] *T-Motor Antigravity 6007II-160 with T-Motor P26x85*, Tyto Robotics Inc., Feb. 2023. [Online]. Available: https://database.tytorobotics.com/tests/k7qq/t-motor-antigravity-6007ii-160-x-t-motor-p26-8s.

[51] D. W. Mellinger, “Trajectory Generation and Control for Quadrotors,” Ph.D. dissertation, University of Pennsylvania, 2012. [Online]. Available: https://repository.upenn.edu/edissertations/547.

[52] *Three-axis inertial measurement unit*, The MathWorks Inc., Natick, Massachusetts, United States, 2023. [Online]. Available: https://nl.mathworks.com/help/aeroblks/threeaxisinertialmeasurementunit.html.

[53] *Gps*, The MathWorks Inc., Natick, Massachusetts, United States, 2023. [Online]. Available: https://nl.mathworks.com/help/nav/ref/gps.html.

[54] *Pressure altitude*, The MathWorks Inc., Natick, Massachusetts, United States, 2023. [Online]. Available: https://nl.mathworks.com/help/aeroblks/pressurealtitude.html.

[55] *Isa atmosphere model*, The MathWorks Inc., Natick, Massachusetts, United States, 2023. [Online]. Available: https://nl.mathworks.com/help/aeroblks/isaatmospheremodel.html.

[56] K. P. Valavanis and G. J. Vachtsevanos, *Handbook of Unmanned Aerial Vehicles*, 1st ed. Springer Dordrecht, 2015. [Online]. Available: https://doi.org/10.1007/978-90-481-9707-1.

[57] N. S. Nise, *Control Systems Engineering*, 4th ed. John Wiley & Sons, Ltd, Sep. 2003.

[58] F. Pace, E. Paolini, F. Sanfedino, *et al.*, “Chapter Twelve - GNC verification and validation,” in *Modern Spacecraft Guidance, Navigation, and Control*, V. Pesce, A. Colagrossi, and S. Silvestrini, Eds., Elsevier, 2023, pp. 647–684. [Online]. Available: https://doi.org/10.1016/B978-0-323-90916-7.00012-3.

[59] D. Simon, *Optimal State Estimation: Kalman, H Infinity, and Nonlinear Approaches.* John Wiley & Sons, Ltd, Jan. 2006. [Online]. Available: doi.org/10.1002/0470045345.

[60] NATIONAL INSTRUMENTS CORP., *Filter Specifications (Digital Filter Design Toolkit)*, Accessed on 21 June 2023. [Online]. Available: https://www.ni.com/docs/en-US/bundle/labview-digital-filter-design-toolkit-api-ref/page/lvdfdtconcepts/dfd_filter_spec.html.

[61] E. W. Dijkstra, “A note on two problems in connexion with graphs,” *Numerische Mathematik*, vol. 1, pp. 269–271, Dec. 1959. [Online]. Available: https://doi.org/10.1007/BF01386390.

[62] *Introduction to A**, Accessed on 13 June 2023. [Online]. Available: http://theory.stanford.edu/~amitp/GameProgramming/AStarComparison.html.

[63] D. Q. Dung, J. Sun, and V. N. Le, “Attitude estimation Algorithms using low cost IMU,” *International Journal of Control and Automation*, vol. 8, pp. 113–126, Nov. 2015. [Online]. Available: doi.org/10.14257/ijca.2015.8.11.12.

[64] B. Candan and H. E. Soken, “Robust Attitude Estimation Using IMU-Only Measurements,” *IEEE Transactions on Instrumentation and Measurement*, vol. 70, pp. 1–9, Aug. 2021. [Online]. Available: doi.org/10.1109/TIM.2021.3104042.

[65] Z. Danziger, *Discrete Frechet Distance*, Accessed on 21 June 2023. [Online]. Available: https://nl.mathworks.com/matlabcentral/fileexchange/31922-discrete-frechet-distance.

[66] Orsted, *Hornsea Project Two Offshore Wind Farm Safety Zone Application*, Accessed on 21 June 2023. [Online]. Available: https://orstedcdn.azureedge.net/-/media/www/docs/corp/uk/hornsea-project-two/how02offex-safety-zone-application-03767092a.pdf?rev=1952f1fa61904a7eba9e30abcc5a4e28&hash=A640A9DCCB9F1ABA07A229035B884D08.

[67] Google, *About OR-Tools*, Retrieved on 13 June 2023. [Online]. Available: https://developers.google.com/optimization/introduction.

[68] *OR-Tools User's Manual*, Google. [Online]. Available: https://acrogenesis.com/or-tools/documentation/user_manual/manual/metaheuristics/GLS.html.

[69] A. Alsheddy, C. Voudouris, E. P. K. Tsang, and A. Alhindi, "Handbook of heuristics," in R. Martí, P. M. Pardalos, and M. G. C. Resende, Eds. Springer, Cham, Aug. 2018, ch. Guided Local Search, pp. 261–297. [Online]. Available: https://doi.org/10.1007/978-3-319-07124-4_2.

[70] A. Foster, O. Best, M. Gianni, A. Khan, K. Collins, and S. Sharma, *Drone Footage Wind Turbine Surface Damage Detection*, Accessed on 21 June 2023, 2021. [Online]. Available: https://www.kaggle.com/datasets/ajifoster3/yolo-annotated-wind-turbines-586x371.

[71] R. Kundu, *YOLO: Algorithm for Object Detection Explained*, Accessed on 21 June 2023. [Online]. Available: https://www.v7labs.com/blog/yolo-object-detection#how-does-yolo-work-yolo-architecture.

[72] TensorFlow. "Convolutional Neural Network (CNN)." Accessed on 21 June 2023. (2023), [Online]. Available: https://www.tensorflow.org/tutorials/images/cnn.

[73] A. Kumar, *Different Types of CNN Architectures Explained: Examples*, Accessed on 21 June 2023, May 202. [Online]. Available: https://vitalflux.com/different-types-of-cnn-architectures-explained-examples/.

[74] C.-Y. Wang, A. Bochkovskiy, and H.-Y. M. Liao, *YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors*, Accessed on 21 June 2023, Jul. 2022. [Online]. Available: https://arxiv.org/abs/2207.02696.

[75] J. Mohajon, *Confusion Matrix for Your Multi-Class Machine Learning Model*, Accessed on 21 June 2023, May 2020. [Online]. Available: https://towardsdatascience.com/confusion-matrix-for-your-multi-class-machine-learning-model-ff9aa3bf7826.

[76] WindEurope, *Wind Energy in Europe 2022: Statistics and the Outlook for 2023-2027*, Accessed on 21 June 2023. [Online]. Available: https://windeurope.org/intelligence-platform/product/wind-energy-in-europe-2022-statistics-and-the-outlook-for-2023-2027/#interactive-data.

[77] European Council, *Fit for 55: The EU plan for a green transition*, Accessed on 21 June 2023. [Online]. Available: https://www.consilium.europa.eu/en/policies/green-deal/fit-for-55-the-eu-plan-for-a-green-transition/.

[78] Statista, *Inflation rate in the Netherlands*, Accessed on 12 June 2023. [Online]. Available: https://www.statista.com/statistics/276708/inflation-rate-in-the-netherlands/.

[79] Funda in Business, *Delftechpark 11*. [Online]. Available: %7Bhttps://www.fundainbusiness.nl/kantoor/delft/object-41265349-delftechpark-11/%7D.

[80] *Baloise*, Accessed on 12 June 2023. [Online]. Available: https://www.baloise.be/nl/prive.html?gclid=CjwKCAjwhJukBhBPEiwAniIcNTTAsG0XGUUKOM4NoqWy9QnZpiIDAFfHGJjpp46fE2uxfDCLHAw4ZxoC9M8QAvD_BwE.

[81] UNESCO, *Engineering for sustainable development*, Accessed on 16 May 2023. [Online]. Available: https://unesdoc.unesco.org/ark:/48223/pf0000375644.locale=en.

[82] B. Suresha, G. Hemanth, R. Hemanth, and N. P. Lalla, "Role of graphene nanoplatelets and carbon fiber on mechanical properties of PA66/Thermoplastic Copolyester Elastomer Composites," *Materials Research Express*, vol. 7, Jan. 2020. [Online]. Available: doi.org/10.1088/2053-1591/ab648d.

[83] United Nations, *THE 17 GOALS*, Retrieved on 3 May 2023. [Online]. Available: https://sdgs.un.org/goals.

[84] A. Teunissen and M. Dittrich, *Hydrogen*, Accessed on 09 June 2023. [Online]. Available: https://www.tebodin.bilfinger.com/services/energy-transition-solutions/hydrogen/.

[85] J. W. Langeraar, *Double the energy from wind, with hydrogen as primary energy carrier*, Accessed on 21 June 2023. [Online]. Available: https://hy-gro.net/newsitem/double-the-energy-from-wind-with-hydrogen-as-primary-energy-carrier.

[86] T. Bruckner, D. Perczyk, A. McKinnon, E. Hertwich, and L. Fulton, "Climate Change 2014: Mitigation of Climate Change," in Feb. 2015, pp. 1329–1356. [Online]. Available: https://doi.org/10.1017/CBO9781107415416.025.

[87] 3M, *3M™ Scotch-Weld™ Structural Plastic Adhesive DP8005*, Accessed on 21 June 2023. [Online]. Available: https://www.3m.com/3M/en_US/p/d/b40066451/.

A: Production Plan

Author: Louis

The goal of this appendix is to provide a production plan for the drone's assembly. As the chassis is the skeleton of the drone on which all other components shall be mounted, it will have to be produced first. The next step is to mount the propulsion and control system. Finally, the inspection system is integrated. Please note that even though the team put significant effort in making the design as detailed as possible, additional mounts for inspection, control and propulsion components may still be required. Therefore, the component list is not fully complete. However, the chassis has been fully designed- including all bolts and brackets necessary to start production.

A.1 Chassis

A.1.1 Preparation components

The modular construction of the chassis means that it is relatively easy to produce. The carbon fiber tubes and base plate are Commercial Off-The Shelf (COTS) parts; custom brackets and connectors are 3D FDM printed. Besides that holes will need to be drilled in the base plate. Last step in the preparation of all parts is to sand the edges of the CFRP tubes at the edges and make them dust and grease-free (eg: using iso-propanol). Similar surface preparation must be done for the connectors in order for the glue to adhere to the surface. Main benefit of using glued connections over bolted connections is the reduced weight. An overview of the parts of the chassis, their materials, and the production processes required is given in Table A.1.

Note that all dimensions are in **mm** unless otherwise specified.

Table A.1: Chassis Components Table

Description	Material	Manufacturing	Treatment	Qty
Chassis Tube (290 x ⌀20 x 3)	CFRP	Off-the-shelf	Sand, degrease and dust	4
Chassis Tube (300 x ⌀20x 3)	CFRP	Off-the-shelf	Sand, degrease and dust	2
Chassis Tube (250 x ⌀20 x 3)	CFRP	Off-the-shelf	Sand, degrease and dust	4
Landing Gear Tube (460 x ⌀20 x 1)	CFRP	Off-the-shelf	Sand, degrease and dust	2
Landing Gear Tube (420 x ⌀25 x 1)	CFRP	Off-the-shelf	Sand, degrease and dust	2
Chassis corner T-connector	PETG	3D Printing	Sand, degrease and dust	4
Landing gear/Chassis T-connector	PETG	3D Printing	Sand, degrease and dust	2
Landing gear T-connector	PETG	3D Printing	Sand, degrease and dust	2
Landing gear tips	Rubber	Off-the-shelf	Degrease and dust	4
Engine mounting T-connector	PETG	3D Printing	Sand, degrease and dust	4
Engine mounting rods	PETG	3D Printing	Sand, degrease and dust	8
Base plate saddle brackets	PETG	3D Printing	Potential drilling	8
Base plate core (600x300x7)	Nomex Honeycomb	Off-the-shelf	Drilling	1
Base plate face sheets (600x300x0.5)	Aluminum	Off-the-shelf	Drilling	1
Bolts (M4x25)	Grade A steel	N/A	N/A	8
Bolts (M4x20)	Grade A steel	N/A	N/A	8
Nuts (M4)	Grade A steel	N/A	N/A	16

A.1.2 Assembly Chassis

After finishing part production, the chassis can be assembled easily. The first step involves gluing the tubes and connectors together- as seen in Figure A.1. Sufficient curing time must be allow for proper adhesive bonding. For the bonding the 3M™ Scotch-Weld™ Structural Plastic Adhesive DP8005 is advised as it is a structural adhesive specifically for bonding of thermoplasts with minimal surface preparation [87].

Figure A.1: Chassis skeleton

In the mean time, the propulsion mounting and landing gears can be assembled. The assembly of the landing gear is very straightforward. To begin with, the landing gear tubes should be mounted together using the in-house developed T-connectors. The rubber tips can be subsequently mounted on the edges of the bottom tube. Doing so should give the result as seen in Figure A.2a.

The assembly of the propulsion mounting is also straightforward. The 3D printed engine mounting should be glued to the engine mounting connectors. Due to the geometry of the propulsion mounting, it is not advisable to 3D print the entire structure at once. Problems with wasteful support structures and overhang angles are anticipated. Therefore, it was chosen to bond together different parts of the engine mounting. The result should be as seen in Figure A.2.

(a) Landing Gear

(b) Propulsion Mouting

Figure A.2: Peripheral Assemblies

The entire chassis can now be fully assembled once the chassis skeleton, engine mounting and landing gear subassemblies are ready. The landing gear and engine mounts can be glued to the skeleton. Moreover, the base plate can be bolted on the skeleton. Doing so result in the complete chassis as visualised in Figure A.3.

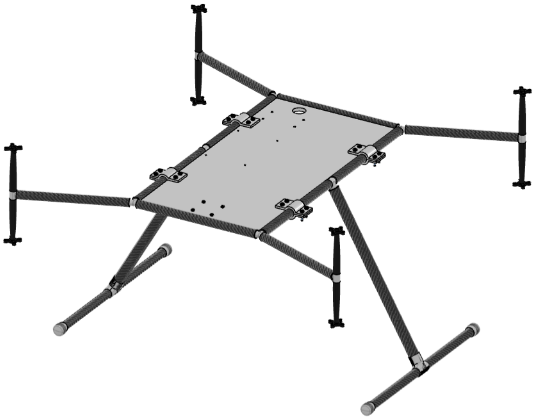


Figure A.3: Complete Chassis Structure

A.2 Propulsion system

A.2.1 Propeller/engine mounting

With the completed structure of the chassis, actual engines and propellers can be integrated. The components necessary can be seen Table A.2. The mounting process is simple: each engine is bolted onto the propulsion mount through four M4x12 thread screw bolts. The propeller can then easily be screwed onto the engine. This process should be repeated eight times for all eight propellers. The final result is depicted in Figure A.4.

Table A.2: Propeller Mounting Components Table

Description	Material	Manufacturing	Treatment	Qty
MF2211 Propeller	N/A	Via T-Motor	N/A	8
Antigravity MN6007II KV160 Engine	N/A	Via T-motor	N/A	8
Screw Thread (M4x12)	N/A	N/A	N/A	32

Figure A.4: Engine/propeller bolting

A.2.2 Hydrogen system

After adding the engines and propellers, the rest of the system components can be added. This includes the fuel cell, hydrogen tank and auxiliary battery unit. The first step is to manufacture the brackets necessary to keep the hydrogen tank in position. Similar to the brackets for the chassis, the parts shall be 3D printed. It may be necessary to post-process the brackets- holes may have to be drilled through cleanly. All other components necessary to mount the hydrogen system are bought off-the-shelf. An overview of the different parts and their manufacturing methods can be seen in Table A.3. Please note that the battery box for back-up purposes contains 10 Voltaplex Grade-A cells as selected in section 4.8.

Table A.3: Hydrogen System Components Table

Description	**Material**	**Manufacturing**	**Treatment**	**Qty**
A5 hydrogen tank	N/A	Via H3Dynamics	N/A	1
Fuel Cell A-2000	N/A	Via H3Dynamics	N/A	1
Pressure regulator	N/A	Via H3Dynamics	N/A	1
Battery Box	N/A	N/A	N/A	1
Lower Hydrogen Bracket	PETG	3D Printing	Potential drilling	2
Upper Hydrogen Bracket	PETG	3D Printing	Potential Drilling	2
Bolts (M5x80)	Grade A steel	N/A	N/A	4
Bolts (M5x30)	Grade A steel	N/A	N/A	8
Nuts (M5)	Grade A steel	N/A	N/A	8

Incorporating the hydrogen system renders the drone as shown in Figure A.5. Notice that key components (namely the tank) are connected through bolts to allow for easy replacement, thereby promoting a truly modular drone.

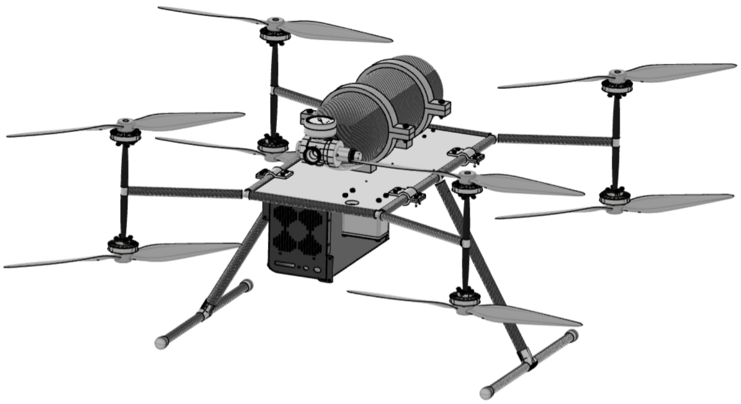


Figure A.5: Operational Drone Assembly (w/o payload)

A.3 Control System

Having added the propulsion system means the rest of the control system can now be mounted as well. This includes the transmitter/receiver, Lidar sensor and control box. The control box houses the barometer, GPS, Inertial Measurement Unit, DC-DC converters and ESC's. Following this, wiring should be added. The result should be a fully working drone. The readiness of the drone is adequate to start first tests of the control system. An overview of the various components can be seen in Table A.4.

Table A.4: Control System Components Table

Description	**Material**	**Manufacturing**	**Qty**
FM30 FR Mini-Receiver	N/A	Via SIYI	1
FM30 Transmitter	N/A	Via SIYI	1
Control Box	N/A	N/A	1
Lidar MID-360	N/A	Via Livox	1
Wiring	N/A	N/A	N/A
Bolts (M3x16)	Grade A steel	N/A	4
Bolts (M4x60)	Grade A steel	N/A	4
Nuts (M3)	Grade A steel	N/A	4
Nuts (M4)	Grade A steel	N/A	4

A.4 Inspection System

In the wake of successful flight demonstration of the drone, the payload can at last be integrated. Assimilating the inspection system includes the depth camera, stereo/infrared camera and gimbal. The component list consists of parts as seen in Table A.5. It should be noted that due to time constraint, not all connections have been modelled in CAD. Hence, some brackets and bolts still need to be added to the component list. The full assembled drone can be seen in Figure A.6.

Table A.5: Inspection System Components Table

Description	**Manufacturing**	**Qty**
Gimbal	N/A	1
Zenmuse H20T	Via DJI	1
Motioncam-3D	Via Photoneo	1

Figure A.6: Fully Assembled Drone (with Payload)

B: Parameter samples for sensitivity analysis

Author: Radu

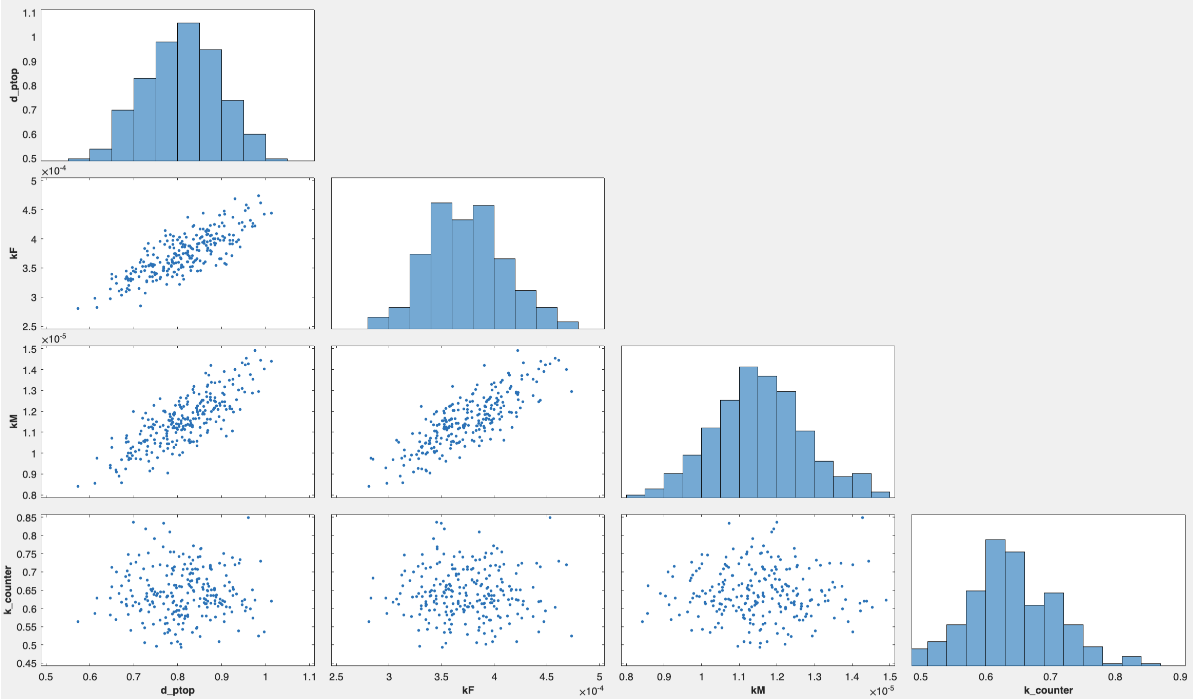


Figure B.1: Parameter samples for k_F, k_M, k_c and d_p

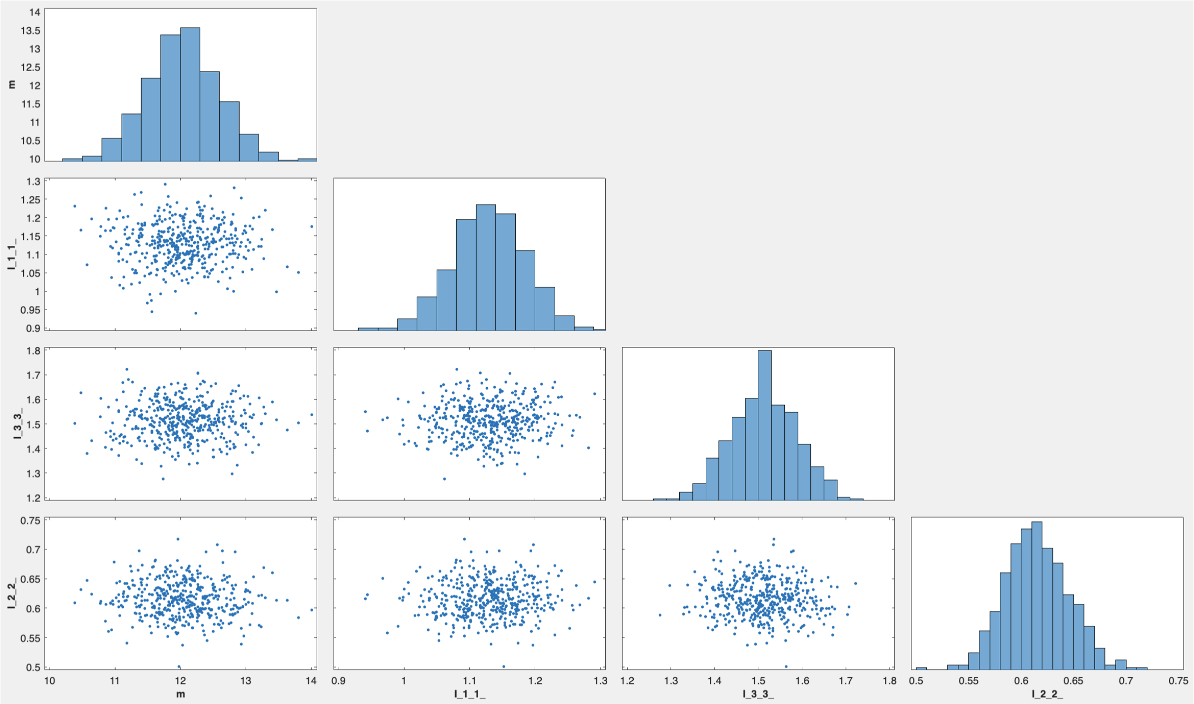


Figure B.2: Parameter samples for m, I_{xx}, I_{yy} and I_{zz}

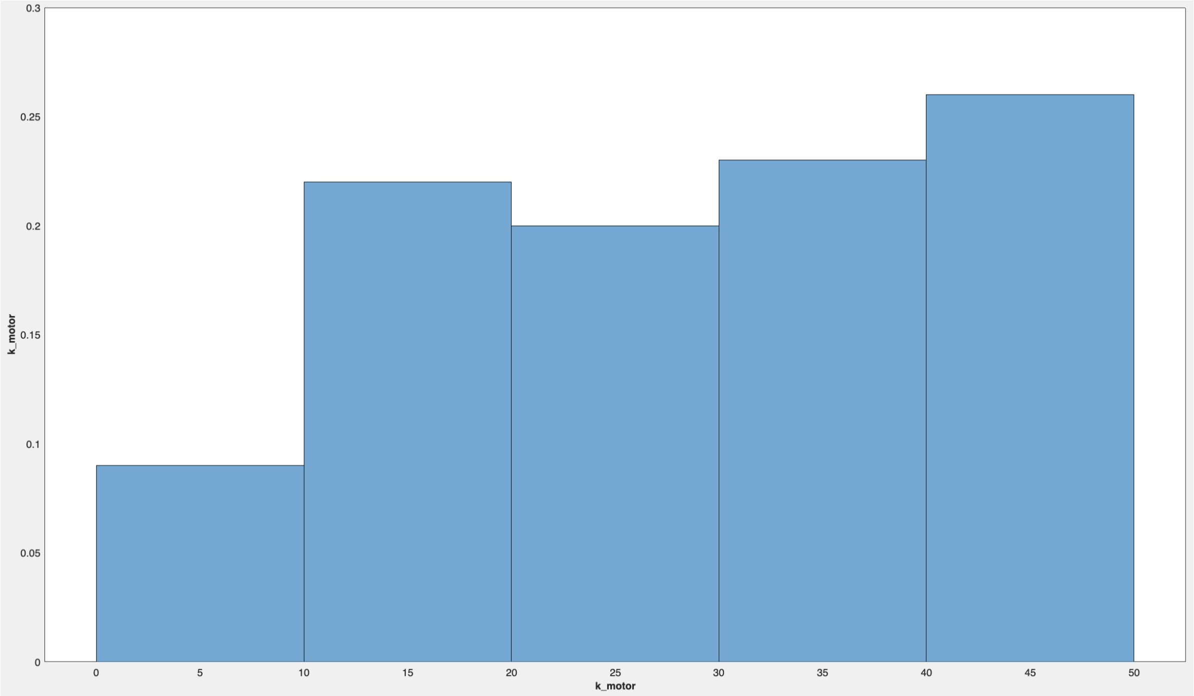


Figure B.3: Parameter samples for τ_m (listed as "k_motor")

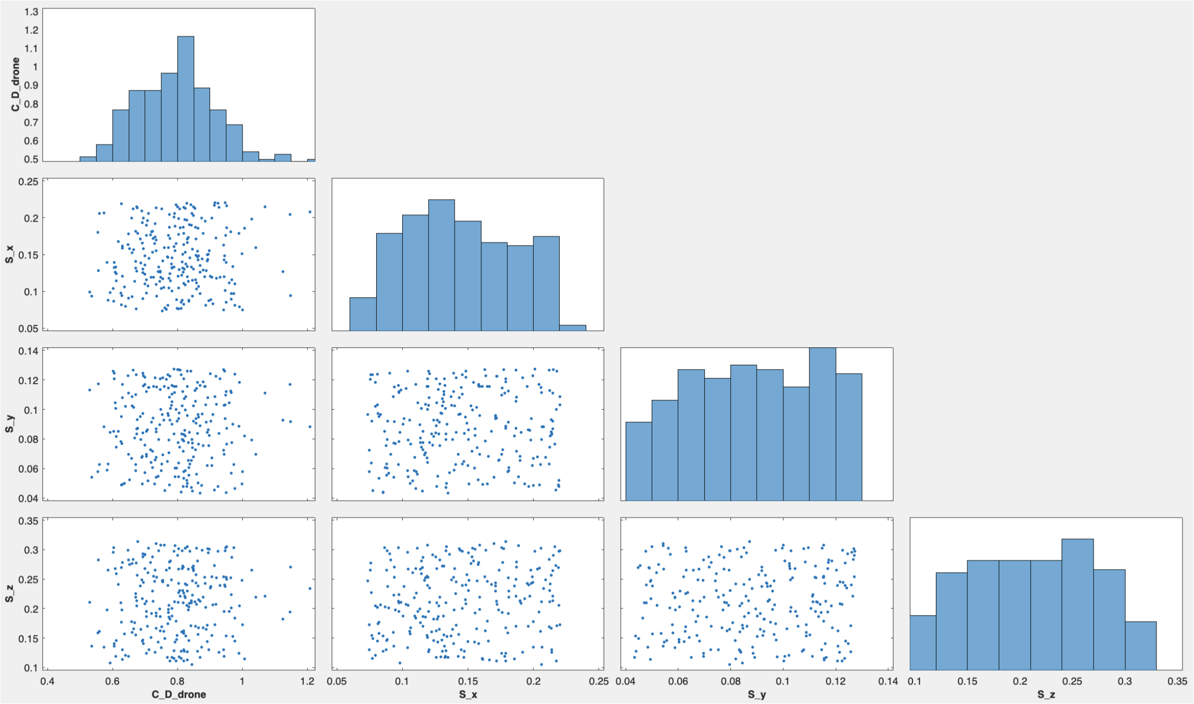
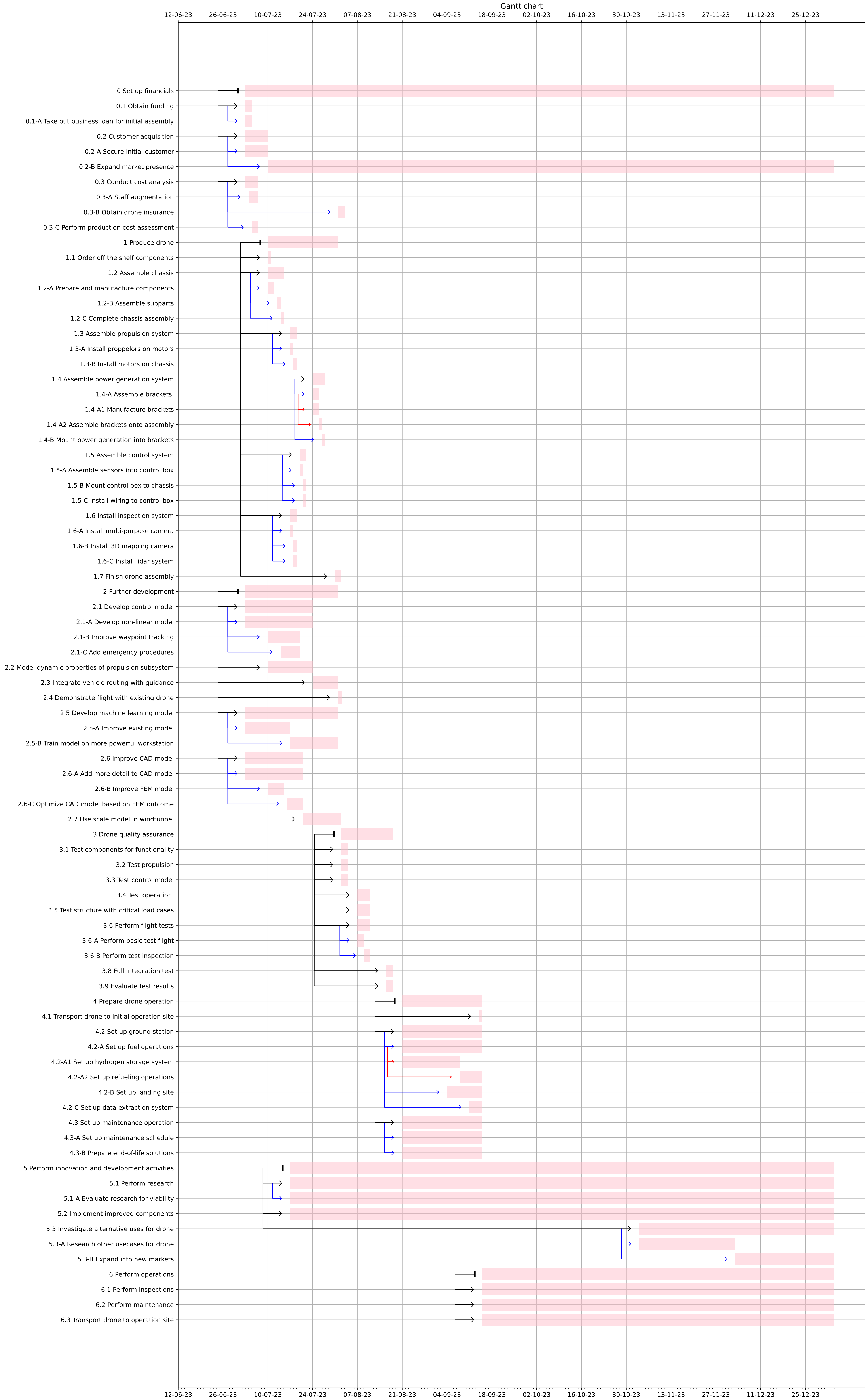


Figure B.4: Parameter samples for C_D, S_x, S_y and S_z


Gantt chart
12-06-23
26-06-23
10-07-23
24-07-23
07-08-23
21-08-23
04-09-23
18-09-23
02-10-23
16-10-23
30-10-23
13-11-23
27-11-23
11-12-23
25-12-23
0 Set up financials
0.1 Obtain funding
0.1-A Take out business loan for initial assembly
0.2 Customer acquisition
0.2-A Secure initial customer
0.2-B Expand market presence
0.3 Conduct cost analysis
0.3-A Staff augmentation
0.3-B Obtain drone insurance
0.3-C Perform production cost assessment
1 Produce drone
1.1 Order off the shelf components
1.2 Assemble chassis
1.2-A Prepare and manufacture components
1.2-B Assemble subparts
1.2-C Complete chassis assembly
1.3 Assemble propulsion system
1.3-A Install proppelors on motors
1.3-B Install motors on chassis
1.4 Assemble power generation system
1.4-A Assemble brackets
1.4-A1 Manufacture brackets
1.4-A2 Assemble brackets onto assembly
1.4-B Mount power generation into brackets
1.5 Assemble control system
1.5-A Assemble sensors into control box
1.5-B Mount control box to chassis
1.5-C Install wiring to control box
1.6 Install inspection system
1.6-A Install multi-purpose camera
1.6-B Install 3D mapping camera
1.6-C Install lidar system
1.7 Finish drone assembly
2 Further development
2.1 Develop control model
2.1-A Develop non-linear model
2.1-B Improve waypoint tracking
2.1-C Add emergency procedures
2.2 Model dynamic properties of propulsion subsystem
2.3 Integrate vehicle routing with guidance
2.4 Demonstrate flight with existing drone
2.5 Develop machine learning model
2.5-A Improve existing model
2.5-B Train model on more powerful workstation
2.6 Improve CAD model
2.6-A Add more detail to CAD model
2.6-B Improve FEM model
2.6-C Optimize CAD model based on FEM outcome
2.7 Use scale model in windtunnel
3 Drone quality assurance
3.1 Test components for functionality
3.2 Test propulsion
3.3 Test control model
3.4 Test operation
3.5 Test structure with critical load cases
3.6 Perform flight tests
3.6-A Perform basic test flight
3.6-B Perform test inspection
3.8 Full integration test
3.9 Evaluate test results
4 Prepare drone operation
4.1 Transport drone to initial operation site
4.2 Set up ground station
4.2-A Set up fuel operations
4.2-A1 Set up hydrogen storage system
4.2-A2 Set up refueling operations
4.2-B Set up landing site
4.2-C Set up data extraction system
4.3 Set up maintenance operation
4.3-A Set up maintenance schedule
4.3-B Prepare end-of-life solutions
5 Perform innovation and development activities
5.1 Perform research
5.1-A Evaluate research for viability
5.2 Implement improved components
5.3 Investigate alternative uses for drone
5.3-A Research other usecases for drone
5.3-B Expand into new markets
6 Perform operations
6.1 Perform inspections
6.2 Perform maintenance
6.3 Transport drone to operation site
12-06-23
26-06-23
10-07-23
24-07-23
07-08-23
21-08-23
04-09-23
18-09-23
02-10-23
16-10-23
30-10-23
13-11-23
27-11-23
11-12-23
25-12-23