Lockheed Martin Aircraft Maintenance Drone

AE3200 Design Synthesis Exercise: Final Report

Group 05

Supervisor: C. de Wagter

July 02, 2019

Delft University of Technology

TU Delft Delft University of Technology

Challenge the future

This page was intentionally left blank.

LOCKHEED MARTIN AIRCRAFT MAINTENANCE DRONE

AE3200 DESIGN SYNTHESIS EXERCISE: FINAL REPORT

An elaboration on the design of a Lockheed Martin fuel tank inspection drone.

by Group 05:

Erik van Baasbank	**4554825**
Edoardo Barbieri	**4539990**
João Bessa Da Cruz Pereira	**4532570**
Ioan Bîrgăoanu-Acăei	**4545249**
Boris Englebert	**4532783**
Jan Hendrik Farr	**4555538**
Max Kemmeren	**4553209**
Arthur Reijns	**4556100**
Andreas Schelfhout	**4211960**
Daan Verkooij	**4564251**

Delft, July 2, 2019
Document Version: 1.7

Supervisor:	Christophe de Wagter
Coaches:	Qingqing Ye
	Xiaodong Li

Distribution list

Supervisor:	Christophe de Wagter
Coaches:	Qingqing Ye
	Xiaodong Li
PMSE tutor:	Sam van Elsloo
Lockheed Martin contact:	Christopher L. Colaw

Changelog

Version 1.0	Draft Version
Version 1.1	Feedback Christophe de Wagter (Final review)
Version 1.2	Feedback other groups (Non-technical parts)
Version 1.3	Feedback Sam van Elsloo (Non-technical parts)
Version 1.4	Feedback from Xiaodong Li (Executive overview)
Version 1.5	Feedback from Qingqing Ye (Conclusion)
Version 1.6	Updated CFD analysis and implementation of navigation in simulation
Version 1.7	Redacted the patentable part of the design

CONTENTS

LISTINGS

LIST OF FIGURES

List of Tables

List of Algorithms

List of Symbols

Sign	Description	Unit	Sign	Description	Unit
A_d	Disk area	m^2	A	Area	m^2
C_d	Drag coefficient	–	C_l	Lift coefficient	–
C_{P_0}	Static power coefficient	–	C_{T_0}	Static thrust coefficient	–
$C_{l_{max}}$	Maximum lift coefficient	–	D	Diameter	m
K_d	Derivative gain	–	K_i	Integrator gain	–
K_p	Proportional gain	–	K_u	Ultimate gain	–
M	Mach number	–	N	Processed nodes	–
P	Power	W	Re	Reynolds number	–
R	Resistance	Ω	T_u	Marginally stable response oscillatory period	s
T	Temperature	K	T	Thrust	N
V_∞	Freestream velocity	m/s	Δt	Time Step	s
α	Temperature coeffiecient	1/K	η	Efficiency	–
ϕ	Roll angle	rad	ψ	Yaw angle	rad
ρ	Resistivity	Ωm	ρ	Density	kg/m^3
τ_d	Derivative time constant	–	τ_i	Integrator time constant	–

Sign	Description	Unit	—	Sign	Description	Unit
θ	Pitch angle	rad		b^*	Effective branching factor	–
d	Distance	m		d	Solution depth	–
$e(t)$	Process variable error	–		l	Cable length	m
n	Revolutions per second	Hz		p	Roll rate	rad/s
q	Pitch rate	rad/s		r	Yaw rate	rad/s
$u(t)$	Controller output	–		v_x	Velocity in x direction	m/s
v_y	Velocity in y direction	m/s		v_z	Velocity in z direction	m/s

List of Abbreviations

3D Three Dimensional.

AMD Aircraft Maintenance Drone.
ARTIS Autonomous Robotic Tank Inspection System.
AWG American Wire Gauge.

BEC Battery Eliminator Circuit.
BEMT Blade Element Momentum Theory.
BPC Battery Protection Circuit.

CAD Computer-aided Design.
CAGR Compound Annual Growth Rate.
CFD Computational Fluid Dynamics.
CMOS Complementary Metal–Oxide–Semiconductor.
CPU Central Processing Unit.

DOT Design Option Tree.
DSE Design Synthesis Exercise.

ESC Electronic Speed Controller.

FOM Figure of Merit.
FOV Field Of View.
FPC Flexible Printed Circuit.
FPS Frames Per Second.
FPU Floating Point Unit.

GNC Guidance, Navigation & Control.
GND Ground.
GNSS Global Navigation Satellite System.

HSV Hue Saturation Value.

IChE In Channel Effect.
ILoCE In Low Corner Effect.
IMU Inertial Measurement Unit.
IP International Protection.
ISA International Standard Atmosphere.
IUpCE In Upper Corner Effect.

JST Japan Solderless Terminal.

LED Light Emitting Diode.
LM Lockheed Martin.

MET Materials, Emissions and Toxicity.
MMA Motor Mixing Algorithm.
MMRF Multiple Moving Reference Frame.
MNS Mission Need Statement.
MRO Maintenance, Repair & Overhaul.

NFPA National Fire Protection Association.

PCB Printed Circuit Board.
PDB Power Distribution Board.
PFTE Polytetrafluoroethylene.
PID Proportional Integral Differential.
PMSE Project Management and System Engineering.
PnL Perspective-n-Line.
PnP Perspective-n-Point.
POS Project Objective Statement.

RANSAC Random Sample Consensus.
RF Radio Frequency.
RGB Red Green Blue.
ROV Remotely Operated Vehicle.
RPM Revolutions per minute.

SD Secure Digital.
SDA Serial Data.
SFC Standard F Connector.
SIMPLE Semi-Implicit Method for Pressure-Linked Equations.
SLAM Simultaneous Localization and Mapping.
SLC Serial Clock.
SLS Selective Laser Sintering.

ToF Time of Flight.
TU Technological University.

UNEP United Nations Environmental Program.

VO Visual Odometry.
VPU Visual Processing Unit.

WBS Work Breakdown Structure.
WFD Work Flow Diagram.

1

Executive Overview

In an effort to reduce the time, costs, and health hazards involved with the inspection of aircraft fuel tanks, Lockheed Martin has commissioned the design of an autonomous inspection drone for fuel tanks to DSE Group 05 at the Aerospace Engineering faculty, Delft University of Technology. This report entails the conceptual design process of the autonomous robotic tank inspection system - or ARTIS - that has been designed in this endeavor. The mission need statement, or MNS, which outlines the reason for developing ARTIS is therefore as follows:

Mission Need Statement Aircraft Maintenance Drone (MNS-AMD): *DSE Group 05 will reduce the time and cost required for the inspection of aircraft fuel tanks.*

First of all, the trade-off summary chapter presents a concise summary of the trade-off that was performed during the midterm stage, in which a final concept was chosen from three preliminary concepts. The following three concepts were proposed:

- *The Air Travel Drone*: this concept optimizes flight time and minimized complexity. Its only way of travel is flying, while it still has the option to statically land.
- *The Hybrid Drone:* this concept optimizes the use of power. This concept can not only fly, but also ride or crawl, thereby decreasing the power usage while inspecting.
- *The Power Cord Drone:* This drone is connected to an outside power source via a cable. It can thus fly for unlimited time and does not have the added weight of a battery. On the other hand, additional risks and complexity are introduced. This concept optimizes data transfer and power capacity.

The three concepts are evaluated based on cost, sustainability, performance, risk and complexity. These trade-off criteria are weighted based on a requirement classification that quantifies the importance of several trade-off aspects and the subsequent correlation between the requirement classification and the criteria. Each concept is then scored for each of the weighted criteria, after which the final trade-off scores presented in table 1.1 are attained.

Table 1.1: Trade-off execution table

	Cost	Sustainability	Performance	Risk	Complexity	Final Score (0-10)
Weight	11 %	3 %	28 %	40 %	17 %	-
Concept 1	9	6.8	7.5	8	6.6	7.62
Concept 2	9	6.4	5.5	5.5	4.6	5.70
Concept 3	8	6.4	8.75	7	4.8	7.14

From table 1.1 it becomes clear that concept 1, the air travel drone, wins the trade-off, and is therefore the concept that is to be developed in-depth for the final design phase.

The market analysis is presented in the next chapter and reveals that long term profitability could originate from more markets than merely the aircraft maintenance industry, such as the oil and railroad industry [9, 10]. The main source of revenue for ARTIS , however, is still projected to originate from the MRO aircraft industry, for which the entire industry is set to be valuated at $115 B before the end of the next decade compared to $77 B in 2015 [11]. Nevertheless, regulatory and administrative practises could hinder the predicted growth by stricter safety management requirements and high market entry barriers. Lockheed Martin itself currently maintains three aircraft projects that are viable for tank inspection with ARTIS, and has a global infrastructure in place for the phased implementation of ARTIS in existing maintenance procedures [12]. As such, successful market penetration will aim at the military aerospace section and the commercial aviation

sector, for which the total fuel tank inspection costs incur $42.9 M and $500 M, respectively, on a yearly basis [13–16].

Existing competition in the aircraft MRO sector is scarce, with only a handful companies offering inspection services by virtue of autonomous drones. Moreover, these companies mainly focus on surface inspection instead of aircraft fuel tank inspection. Long-term efforts will additionally strive to capture market shares outside the aircraft MRO sector, in which the marine segment and off-shore solutions are of prime importance in the projected revenue streams.

After that, the requirements analysis chapter presents a complete overview of the subsystem requirements as well as the user requirements and constraints, which provide insight in the design process and how the drone should be operated and maintained. The most important requirements are determined to concern the risk of spark generation that can ignite the fumes inside the tank and potentially cause irreparable damage to the aircraft, and these are labelled as category 1 driving requirements. The remaining requirements are split into category 2 driving requirements (unsuccessful mission execution without further harm or injury when not met), key requirements (mission can still be executed but performance is diminished when not met), and others (no harm to mission execution when not met but beneficial to mission performance when met). The majority of category 1 and 2 driving requirements is in regard to performance and risk.

The technical risks that are characterized by a probability of occurrence and severity higher than medium are presented in the technical risk analysis chapter, where the risks that are unlikely to occur and result in superficial damage can be found in [6] and [17]. The risks that are identified primarily involve system integration errors and damage to the tank, either from system malfunctioning or sparks, collision damage, communication errors, operational risks (e.g. getting stuck inside the tank, data corruption, or inaccurate damage detection), payload and electronics risks, and structural and manufacturing risks. A risk reporting map is constructed, which reveals that several risks had an unacceptable combination of likelihood and severity as they are placed in the red top-right corner of the matrix. These risks specifically are sought to be mitigated during the detailed design phase, after which a post-mitigation risk map can be set up.

Numerous technical resources, namely the masses of the designed subsystems, hardware cost, and flight time, are derived from the driving and key requirements, after which a design specification values for each of the technical resources and a corresponding contingency plan are devised. The contingency plan consists of a Technical Performance Measurement (TPM) method to track the contingencies of the designated technical resources for several levels of design maturity. The levels of design maturity that are used start at the beginning of the detailed design phase and end at the actual flight hardware test, at which the TPM values should converge to 1 [18].

The next logical step is commence with the actual design of ARTIS. The propellers are sized to have a diameter as large as possible within the given size constraints with the purpose of minimizing power usage, which resulted in a propeller diameter of 12.4 cm. In order to accommodate for the surrounding ducts and a layer of protective foam, the disk area was effectively reduced to 10 cm. Several tested and appropriately sized propellers were selected from the UIUC database on propeller performance and geometry data, after which it was determined that none of the examined propellers demonstrated optimal static (hover) performance, expressed by the Figure of Merit (FOM) [19]. It is proposed that better static performance can be achieved by altering the GWS DD 4025 propeller to include a more optimal airfoil and an improved blade pitch angle. Three airfoils were selected that possessed beneficial aerodynamic properties at the operating Reynolds numbers, after which the AH-7-476 airfoil was finally chosen because of its high $C_l - C_d$ ratio and $C_{l_{max}}$, combined with a low average C_d [20]. Analyzing the performance of the redesigned GWS DD 4025 using JBLADE showed an average increase in the FOM of 24 % compared to the regular GWS DD 4025.

The aerodynamic interactions between the propeller and the structural components were subsequently investigated by performing a simplified CFD simulation, in which only one arm of the drone was analyzed for simplicity's sake. At hover condition (T = W), the total drag (pressure and friction drag) generated by the arm structure is found to be 0.13 N, which implies that the propellers will need to spin around 9.4 % faster compared to a free rotating propeller. Most of this drag can be alleviated with a curved duct inlet, which has low

pressure on the top side due to the incoming flow velocity of the propeller, and high pressure on bottom side due to flow stagnation.

Aerodynamic disturbances can also be caused by interactions with the wall (the wall effect). Using a 2D CFD simulation, a relation between the distance to the wall and lateral force generated by pressure differences was quantified.

The corresponding motor was selected from the eCalc database based on the motor's mass as well as power usage and associated battery mass for a given gravimetric energy density and flight time, and a minimal thrust-to-weight ratio requirement of 2. This resulting motor that was picked is the RotoX RX1404, which has a mass of 8.5 g including cabling from the ESC to the motor [21]. Moreover, this motor is able to sustain a thrust at least twice the drone weight, and is brushless and as such does not create sparks during operation, thereby mitigating risks **A3**, which concerns induced sparks during operation, and **P1**, which deals with the power required exceeding the available power, respectively.

Structural design starts by determining the loads the structure needs to withstand, for which it was determined that the most critical loads occurred during an impact with the tank floor when the drone falls from the tank ceiling at 1 m height, taking into account a 4/3 safety factor. As the toughness of the material alone cannot be considered enough to absorb the impact energy, energy absorbing polymeric foam is added to possible impact areas on the drone. As the amount of foam that can be applied to the sides of the duct is limited by the fact the drone has to fit through 8x12 inch inspection holes, a small layer of medium density (MD) foam is applied to the sides of the duct, whereas a larger layer of lower density (LD) foam can be applied to the top and bottom duct surfaces due to more available space [22]. Furthermore, as the maximum forces exerted on structure are high and can be applied from a multitude of positions, CFR Nylon with 30 % carbon power is used, which is still isotropic when 3D printed [23].

For static electricity dissipation, the structural components directly exposed to the airflow from the propeller are covered in aluminum foil, which makes these components conductive and as such makes the drone easier to ground. Since covering the propellers with aluminum foil might lead to damage to the foil as a result of the high aerodynamic forces, the carbon content of the propellers is increased in an effort to make the propellers more conductive [24]. The carbon content in the propellers is increased to 50 %, which makes injection moulding the desired manufacturing option for the propellers [25]. Table 1.2 summarizes the materials used and the quantities required for the final design. The resulting structural design meets the impact requirement and as a result mitigates risks **A6** (damage to fuel tank structure due to collisions) and **S1** (undetected manufacturing flaws).

Table 1.2: Materials quantities used in the final design

Component	Material	Weight
Propeller	CFR 50 %	7 g
Frame	CFR 30 %	74 g
Dense foam	Polyurethane	12 g
Light foam	Polyurethane	11 g
Static dissipation foil	Aluminum	3.63 g

With the materials known, the frame design was performed by approximating the frame structure by straight beam elements and then applying the matrix displacement method to estimate the loads in all the members and the total mass of the structure, while adhering to the structural mass budget that was set. The element thicknesses were adapted iteratively until the calculated stresses in the members did not exceed the failure stress of the material and the estimated structural mass stayed within the set mass budget. The structural propeller design was performed by analyzing the imposed aerodynamic loading and conducting a subsequent stress analysis. This resulted in structural propeller design for which the estimated mass amounts to 0.178 g.

To ensure proper knowledge of the operational environment, the design contains 4 cameras, an IMU, and 6 time of flight sensors, one on each face of the drone. Two of these cameras are used to provide stereovision, while the other two serve to provide a full 180° view of the tank for inspection purposes. The camera used is the CAM1320, it can take 13.2 MP images, and 1080 p video. The pictures and one video feed are stored on microSD cards which can be inserted into the drone's motherboard. This motherboard also contains the

aforementioned IMU, which can operate at up to 1000 Hz all computational capabilities needed, including a CPU and VPU, Central - and Visual Processing Unit. The VPU is specifically optimized for processing images and is used to run the guidance algorithms.

The aforementioned electronics need to be powered, this is done with a four celled lithium-polymer battery. The total capacity of this battery is 15.2 Wh, with a C-rating of 10. This means the battery can deliver an amperage at which it is empty after 1/10 hours, which is sufficient for ARTIS's power needs. This power is delivered to power distribution board, which converts the voltage down and then provides the LEDs, ESCs, and motherboard with power. The ESCs serve to deliver the right amount amps and voltage to the motors, depending on signals provided by the motherboard.

One of the most important requirements states that ARTIS should not generate any sparks at all. One such source is the trioboelectric effect[26], which is what makes friction generate static charge. This effect leads to the drone gaining a potential difference with respect to the tank. The proposed solution to this effect is to make the drone conductive in areas which are prone to the triboelectric effect. This is to dissipate the static charge before it can build up. This solution can however only be verified through testing which is still to be done. If this solution is not considered to be effective enough it can be combined with another solution like an anti-static agent.

In the realization of autonomous flight, it is first and foremost important to identify the interfaces between the principles of guidance, navigation, and control, for which the framework that is used in the GN&C design is shown in figure 1.1.

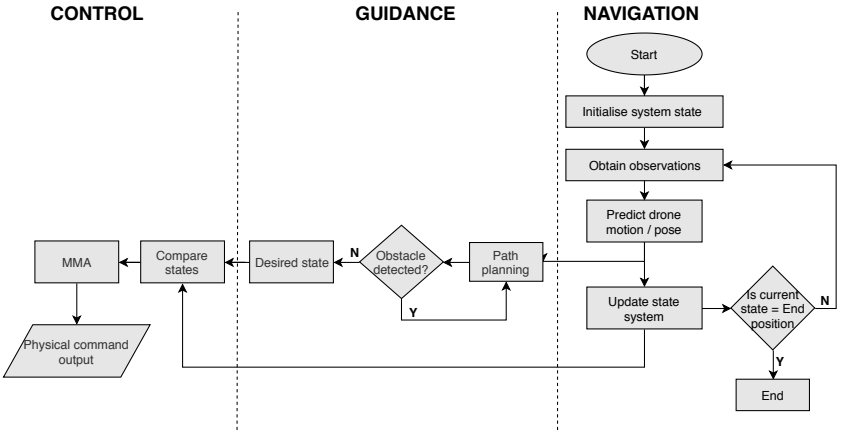


Figure 1.1: GNC interfaces framework flow diagram

The localization problem for navigation is tackled by means of sensor data from the cameras, IMU, and time of flight sensors. Indirect visual odometry (VO) was deemed most appropriate for ARTIS given the nature of the operating environment and on-board capacity [27, 28]. The sensor features introduced in turn lead to stereo visual inertial odometry with time of flight sensors as the navigation algorithm that will be run. The computer vision algorithms are run fully on the on-board VPU. Lines and circles were selected as descriptors for the environment as they are able to represent the majority of the items in the fuel tank. The pre-processing of the images for the detection methods then requires RGB conversion to gray scale and blurring, after which an edge detector can be applied. By applying existing algorithms to the sensor input, the camera's position and motion can be estimated. Given this estimation, together with the data from the IMU and Flight distance sensors, a final estimated is obtained by using a Kalman filter. A test for different lighting scenarios subsequently revealed that the lighting scenario implemented in ARTIS is advantageous for increasing the number and accuracy of detected features. Having detected obstacles, the generation of a collision-free path is required, with an accurate environment representation storing information about occupancy being needed. Such task is tackled via a voxel grid map, in which the fuel tank domain is divided into a grid of smaller sub-elements (voxels) [29]. The A^* algorithm is then chosen as path-planning solution, employing best-first search in order to find the least-cost path from a starting grid point to the target location. Proper flight control and stability are ensured by virtue of a cascaded controller that consists of six PID controllers, where the three attitude and the altitude controllers are nested in the inner loop and the position controllers are placed in outer loop. The output command is then run through the motor mixing algorithm (MMA), which outputs the motor outputs [4]. The choice for PID controllers instead of nonlinear controllers

was made based on the ease of implementation of PID controllers and the sufficient performance of PID controllers in analogous quadcopters [4, 30, 31]. The performance of the proposed control scheme evidently has to be verified and validated in order to confirm that PID controllers yield sufficient performance in the design for ARTIS as well.
The control scheme was then built in Simulink using MATLAB's aerospace blockset quadcopter package as a basis, after which the controllers were individually tuned using the Ziegler-Nichols tuning method [4, 32].

The final design is presented in figure 1.2. It shows two colors of foam, gray and red, being medium density and low density, respectively.

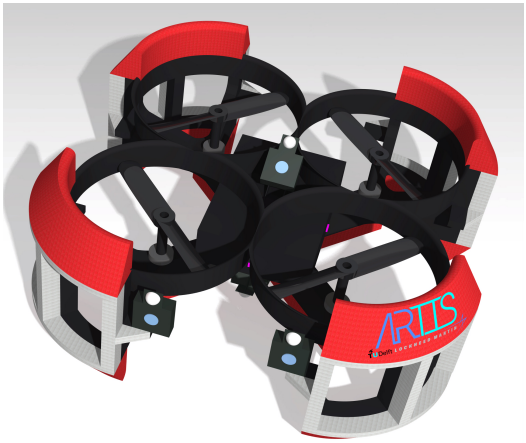


Figure 1.2: Top view render of the final ARTIS layout

ARTIS has a total mass of 285 g, with a total power usage of 85 W for hover. A total distribution of mass and power usage can be found in figure 1.3. This power usage, combined with the aforementioned 15.2 Wh battery, leads to a total flight time of 10 minutes and 40 seconds.

Continuing on the subject of time, a fuel tank compartment consisting of 4 sections can be inspected in 4 minutes and 33 seconds. Extrapolating this to the full wing gives a flight time for the inspection of about 28 minutes. This is done through key points, at which the drone stops moving and rotates 360° in steps of 36°, while the camera field of view is 45°. This allows for a contingency in images if the positional accuracy is off by a few degrees. The images taken at these points are then processed after ARTIS has left the tank.

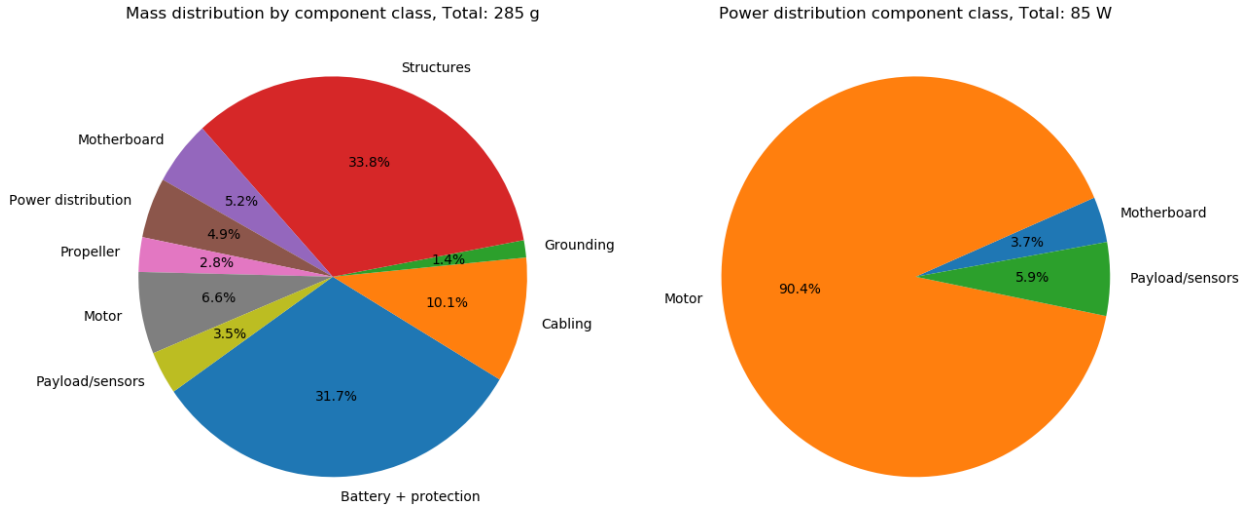


Figure 1.3: Mass and power distribution of ARTIS

Since the design still has some uncertainties a sensitivity analysis is performed, this looks at how a change in the design would affect it. The first thing looked at is the possible benefits of the ducts, these would decrease the power required by the engine to hover. This can increase the flight time of ARTIS to up to 20 minutes.

Another possibility is including a solution for grounding, this is estimated to increase the mass by about 6 g this would then reduce the total flight time by 20 s. This means the mass and flight time requirements can still be met with a grounding solution.

For general sensitivity, a total of 12 g or 5.7 W can still be added before the 10 minutes requirement is no longer met. This means there is still some room for changes in the design.

Contingencies were also tracked per subsystem in the design, the overall contingency ended up exactly at the point that was predicted, being 25 g. For the subsystems cabling mass was largely underestimated (330 %) and structural mass was also underestimated (28 %). Payload mass, propulsion mass and hardware costs were all overestimated by about 50 %. Finally the battery mass was underestimated slightly by 20 % and flight time was very close at a mere 1.6 % underestimation.

The verification process for control and stability comprises two notions, namely verification of the linear model itself by confirming that it works as intended and verification of the performance of the linear model compared to the nonlinear model. For the former, several unit and system tests demonstrates that the linear model indeed functions as intended. For the latter, the step response characteristics between the linear model and nonlinear model are compared, which showed that differences between the linear model and nonlinear model do exist. However, these differences were either deemed beneficial in terms of the actual response in the nonlinear model, or not detrimental enough for the performance of ARTIS with regard to control and safety requirements. The Simulink model that was used to tune the PID controllers is therefore considered verified.

The MATLAB programs for guidance purposes were checked for internal consistency via extensive debugging and by writing unit testing functions. Having assessed no programming errors were present, the subsequent activities aimed at establishing the programs were computing the correct outputs. False obstacles were added, up to 2 % of the total occupancy map, leading to a rougher path. The algorithm performed as expected, giving results in a reasonable range of values. To further confirm the MATLAB programs fulfill their intended purpose, voxels were randomly shuffled, leading to a maximum value of 1 % false or missed obstacle detection before collision. All of the written functions, besides being free of errors, performed as expected, yielding results in reasonable ranges, thus hinting at being correctly formulated.

The navigation algorithm is mostly composed of functions from the open source library OpenCV, and as such the functions have been verified and validated thoroughly already. The verification for navigation therefore consists of verifying the implementation of the open source functions. The feature detection implementation by testing different input images, adding shaking or blur, and altering the lighting conditions. It was found that the performance was sufficient, especially since IMU data can mitigate the problems caused by possible failure of line detection. Moreover, a second test was carried out that confirmed the possibility of feature matching and tracking on the fuel tank structure, using a Lucas-Kanade optical flow method [33].

The control and navigation were further validated by writing simulation software that runs both the control algorithm and an open source navigation algorithm. By doing this it was possible to validate that the control algorithm works in the typical operating conditions it will encounter in the fuel tank. The possibility to use VIO algorithms was also validated while showing why using corners as features is sub-optimal in the fuel tank and why line feature are better in this case. The simulation itself was verified by comparing several test cases against hand calculation (free fall, constant angular acceleration, etc).

Given the limited time scope, it was impossible to properly validate the guidance algorithms. Instead, possible validation tests are proposed, starting with implementing the guidance algorithms in the simulation or in a real-life drone. Subsequently, random obstacles could be introduced in the predetermined path to confirm whether the guidance algorithms function as such that the drone will avoid these obstacles. Also, a deliberately erroneous path could be provided to the algorithms in that it directs the drone to fly through a solid rib section instead of a hole, and then to examine if the drone is able to autonomously correct for this error by passing through the actual hole, after which it should return to its assigned path.

For structures, the code verification has been performed by checking that the results for all cases satisfy equilibrium of forces and of moments. Furthermore, for symmetric loading cases, the results were checked to satisfy symmetry. The MATLAB tool that was used was verified using a finite element simulation in ANSYS, which reveals that the MATLAB tool provides an accurate estimation for symmetrical load cases, but not for asymmetrical loading. The results from the FEM analysis were considered and this was then adapted in the tool, after which the results from the tool and FEM did correspond. The structural design is thus successfully verified with Ansys. Similarly, A FEM analysis was used in an effort to verify the propeller design, which

revealed that the maximum stresses in the propeller are below the tensile strength of the material, which also verifies the propeller design. The validation for the structure is deemed to be out of the scope of this projected, as it would require actual testing of the structure.

Verification of the propeller's increased efficiency (expressed by the Figure of Merit) is verified with CFD simulation. The propeller geometry was imported in ANSYS Fluent, and using a Multiple Moving Reference Frame the propeller was spun at a radial velocity of 11 000 RPM (the estimated velocity needed for hover). This simulation resulted in a FOM value of 0.698, slightly lower than predicted by JBLADE, but still a 21.8 % improvement over the best reference propeller. Further testing will have to be performed to validate this result.

For grounding, payload and electronics, no numerical or analytic models are used, which implies that verification cannot be performed for these subsystems and that focus is solely put on validation. The illumination requirement was validated by means of using an LED source in the form of a smartphone flashlight to illuminate an area of 1 m^2 inside a wing-box. This test was nevertheless considered quite inaccurate, which means that a safety factor of 2 is applied to the illumination requirement and that LEDs are selected accordingly. The Simulink color detection algorithm was validated by testing the detection capabilities on a white sheet of paper with marks. The test demonstrated that the color detection algorithm was able to detect the green marks, which thus validated the detection algorithm. Due to the fact that the test conditions differ from the actual conditions in the tank, it is advisable to increase the HSV thresholds to increase the chances of detecting of an anomaly and thereby mitigating risk **C6** (inaccurate detection of fuel tank damage/corrosion), even if that might increase the chance of a false detection.

For the grounding or explosion proofing, it is proposed that it should first be tested whether only some material for dissipation of charge would perform sufficiently in general static charge tests. If this is not the case, the addition of an anti static agent should be tested, after which other solutions can possibly be explored as well if this combination proves to perform insufficiently. As sparking should be prevented at all times in the operating environment, an electrostatic charge test was carried to measure the charge build-up on a hovering drone, which in the end proved to be inconclusive due to the interaction of the present human bodies with the electric field. Improved testing conditions for evaluating the electrostatic charge build-up and distribution, as well as for the general tests regarding the proposed solution, are therefore proposed.

A reliability, availability, maintainability and safety analysis of ARTIS reveals and determined several important characteristic of the product. A failure rate of 1/90min has been estimated at this preliminary stage. This failure rate is driven by the autonomous flight system as hardware reliability was determined to be high on a relative level. Maintenance inspections were laid out, with an outline of frequent visual inspections and both planned and unplanned thorough inspections of ARTIS. Spare parts are to be included with ARTIS and reordered after a certain threshold is reached in order to ensure the availability of ARTIS. Safety is also required in order to ensure hazard free operations and pubic acceptance.

The operational environment and breakdown establish the actions to be taken throughout ARTIS' life cycle from ordering up to discarding. ARTIS will operate within C-130 fuel tanks, each C-130 wing contains 3 separate fuel bays. The fuel bays contain different types of rib structures, some resembling a truss structure and others resembling flat plates with circular or oval cut outs. Figure 1.4 shows a accurate representation of the fuel tank environment. Furthermore fueling pipes may also be found along the compartments. The design of ARTIS takes into account all these environment features and more. Once ARTIS is proved to operate efficiently such an environment, it may start to be applied in different aircraft such as to expand its presence in the MRO market.

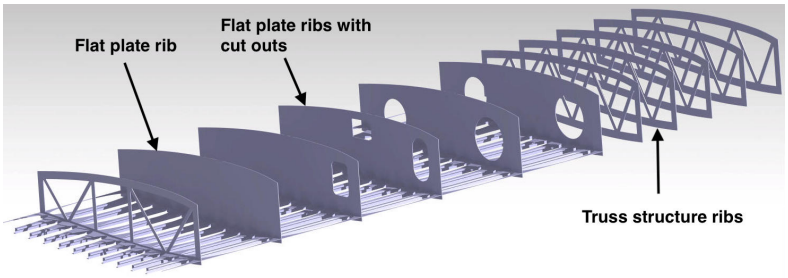


Figure 1.4: Side view of the ribs in a C-130 wing

The operational breakdown consists of a high level of initialization, deployment, inspection and retrieval. The operational breakdown also deals with possible issues that can arise during operations and how they can be managed. The total operation time is estimated to be approximately 236 minutes for one inspection.

A financial analysis is presented with the purpose of gaining insight in the financial benefits ARTIS could provide Lockheed Martin during the first year of sales (anno. 2020). To come up with an efficient estimate a Cost Break-down Structure (CBS) is set up that segments the total cost into system costs, relating to production and assembly, and supporting costs, where the supporting costs are further divided into shipping & delivery, maintenance & support, research & development, sales & marketing, and management. The segmented supporting costs were estimated by means of engineering judgment and verified through a research study concerning corporate cost breakdowns. By assessing the fixed and variable costs under the production and assembly costs, it is determined that the total system costs amount to €1433.65, which is below the €2500 **AMD-NF-CNST-Cost-01** system cost requirement. Back-engineering the calculation, including the supporting costs, then leads to a total first year cost of €10240.35. An analysis on the Return of Investment (RoI) shows that €8.3 million is earned in the first year of sales, which implies for Lockheed that the inspection costs saved amount to €30 million. The RoI for the first year is high due to the engineering team agreeing to low wages and research and development assistance from TU Delft. Nevertheless, one must remain critical on the results, as e.g. compound effects and USD to EUR conversion rates are not included in the RoI computation.

ARTIS' sustainability is also analyzed, this is done by first looking at the general effect of what this project is trying to achieve. This effect is deemed to be very positive, with the main contribution in that respect being the elimination of the many health hazards to any personnel tasked with tank inspection [34–38]. Furthermore ARTIS aims to reduce time and energy spent on this task.

The environmental impact of ARTIS' lifecycle has also been analyzed through the UNEP EcoDesign Strategy Wheel [39]. For this wheel a minimum score was set, this score was reached with, in most categories, a fair amount of leeway as can be seen in figure 1.5. The worst part about ARTIS from a sustainability perspective is the battery.

Figure 1.5: EcoDesign strategy wheel for ARTIS

With the conclusion of the detailed design phase, it is necessary to revisit the risks in order to determine how they have been managed and implemented in the design. Ideally ARTIS has been designed with inherit safety principals in mind. With a summary of the mitigation measures applied during the design to the previously determined risks, the shortcoming with regards to risk can be identified and dealt with.

Those risks that are determined to not have been managed appropriately are further mitigated, and new risks that have only been identified during the design stage are also addressed.

The last phase of this project stage is to verify compliance with the requirements. There were no driving (category 1 and 2) or key requirements that were not met to there full extent. Most requirements were fully met, one was partially met and nine are to be tested or investigated, meaning there was not information available at the moment to verify their compliance. All driving requirements were fully met with the exception on one (**AMD-NF-CNST-sfty-02** The drone shall not create sparks during inspection.) which is still to be tested and investigated. Overall, the requirement compliance analysis showed promising results.

As far as post DSE development is concerned, the logical next step would be to build a prototype of ARTIS, after which thorough testing should be performed. These tests ought to comprise the entire functionality of ARTIS, i.e. propulsion tests, static charge tests, flight endurance tests, autonomous flight tests, and structural tests, which will reveal any possible design flaws. After the prototype testing phase is completed, actual full-scale testing within the operating environment should be done, which opens the door to verification of the product. This then leads to the start of the production phase, during which a detailed marketing plan can also be formulated. This market plan should be reassessed regularly to assess the potential scalability of ARTIS to other possible applications. Simultaneously, it is advised to continue research & development with the purpose of improving and innovating the current design.

2
INTRODUCTION

The rise of automation is continuously generating demand for smart solutions within the Aerospace and Defence sectors. More specifically, the market for inspections and maintenance is changing rapidly. At this stage, humans are subjected to dangerous working-environments, while slowly but surely drones are finding their way into use, providing for better and faster inspections [40].

At Lockheed Martin, fuel tank inspections form an important aspect of the total inspection program, as in this report it is estimated that $42.9 M (i.e. €38.30 M) is spent annually on obtaining information about fatigue cracks and other anomalies inside fuel tanks. Currently, this process is performed by humans, and is considered open for process improvement, due to its hazardous, lengthy and costly nature. The fumes remaining from flight make it a hazardous environment, and thus long venting time is required. Lockheed Martin has therefore called out for innovation. Although at this point crawling robots have been investigated by Lockheed, they prove to be sufficient only for smooth surfaces outside of fuel tanks [41]. Therefore, by imposing an assignment for the Design Synthesis Exercise at TU Delft, Lockheed Martin hopes to accelerate the steps towards autonomous and efficient inspection inside fuel tanks. After generating key, driving, and killer requirements, the following associated mission need statement has been developed.

Mission Need Statement Aircraft Maintenance Drone (MNS-AMD): *DSE Group 05 will reduce the time and cost required for the inspection of aircraft fuel tanks.*

A solution to the above posed mission need is presented in this report, following from an intensive 10 week design process. The group has established what specifically the team would like to achieve. Therefore, a project objective statement has been established. It can be constructed as follows:

Project objective statement Aircraft Maintenance Drone (POS-AMD): *to demonstrate to Lockheed Martin the possibilities of employing an autonomous MAV to reduce their aircraft fuel tank inspection time and cost.*

Three worked-out concepts followed from previous reports, of which each maximized one aspect of the mission, and minimized the risk of another. The best solution was chosen via a trade-off. Throughout the report, the solution will be referred to as the Autonomous Robotic Tank Inspection System, or ARTIS.

Bearing in mind the above, this report will inform about the decisions made and steps taken to reach ARTIS' final design. To structure this effectively, the report is set out as follows. In chapter 3, the design concepts from the previous report are briefly shown. The winning design is presented, with their associated trade-off scores. In chapter 4, SWOT is used as a tool to forecast market prospects. Chapter 5 presents the requirement from the customer, as well as the ones generated in previous reports, and summarizes the significance of the most important ones. In chapter 6, the key risks are given attention to, as well as how they are mitigated. Chapter 7 shows how resources are managed, and establishes the cost of the individual components, as well as a contingency strategy. Chapter 8 present the design and materials selected ARTIS, and chapter 9 discusses the internal hardware components of the drone. In chapter 10, attention is given to the guidance, navigation and control. The required hardware's functionality is touched upon, as well as how G, N and C are merged into a simulation. Chapter 11 presents the final design, with the configuration and layout of the hole drone. Then, chapter 12 dives into verification and validating all found results. chapter 13 discusses the operations and logistics, and chapter 14 covers how the group foresees the cost breakdown, as well as the return of investment. Chapter 15 explains how sustainability is incoorporated in the design, and chapter 16 tells how this design complies with the mitigated risks. Lastly, it is checked whether requirements are complied with in chapter 17. Further recommendations are then provided after a conclusion.

3
TRADE-OFF SUMMARY

In this chapter, the trade-off executed in the midterm report [6] will be summarized and the concept to be designed in this report will be highlighted. First, in section 3.1 the three design options that were analyzed are briefly elaborated on. Then, the setup of the trade-off is shown in section 3.2. The results of this trade off are mentioned in section 3.3. Section 3.4 then shows the analysis of the the trade-off.

3.1. DESIGN CONCEPTS

The following three design concepts were proposed:

- *The Air Travel Drone*: this concept optimizes flight time and minimizes complexity. Its only means of travel is flying, while it still has the option to statically land.
- *The Hybrid Drone:* this concept optimizes the use of power. This concept can not only fly, but also ride or crawl, thereby decreasing the power usage while inspecting.
- *The Power Cord Drone:* This drone is connected to an outside power source via a cable. It can thus fly for unlimited time and does not have the added weight of a battery. On the other hand, additional risks and complexity are introduced. This concept optimizes data transfer and power capacity.

These concepts were designed in more detail to properly analyze which one is most suitable for the final design. The most important characteristics per design are stated in table 3.1.

Table 3.1: Most important characteristics per concept

	Concept 1	**Concept 2**	**Concept 3**
Mass [g]	270	270	174
Power cons. [W]	77	34	101
Flight time [min:s]	10:24	18:55	-
Inspection time [mm:ss]	16:03	54:13	20:35
Hardware cost [€]	615	620	804

3.2. TRADE-OFF SETUP

The trade-off criteria setup and weighting is visualized in table 3.2. In the top row, the trade-off criteria are presented. In the left column, the requirement classification factors are presented. These factors quantify how important a certain aspect is based on the requirements. The correlation between the classification factors and the criteria is shown in the sand colored boxes. By taking the total sum and the weighted average, the weights for each criterion was computed as seen in the last row. It can be seen that for this mission, the risk is vital. Furthermore, performance is also important. Less important are the complexity, cost and sustainability.

Table 3.2: Criteria weighting table

Requirement classification factors	Score [-]	[%]	Cost	Sustainability	Performance	Risk	Complexity
Safety & Damage control	200	56	2	0	5	10	2
Inspection Time & Quality	60	17	4	1	10	7	7
Regulatory compliance	20	5.6	3	2	4	8	6
Stability & Control	26	7.3	0	0	9	8	5
Level of autonomy	18	5	3	1	8	9	8
Cheapness	8	2.2	10	5	7	9	4
Maintenance	10	2.8	3	6	3	6	5
User-friendliness	8	2.3	2	0	6	5	8
Eco-friendliness	4	1.1	0	10	3	7	2
Total sum	354	100	880	258	2204	3150	1368
Weight	-	-	**11%**	**3%**	**28%**	**40%**	**17%**

Each of the concepts was scored on a scale from 0-10 on all criteria. The scores for each criteria were based on several features. For example, the performance score is based on the flight time, inspection time, payload mass and camera obstruction. For a more detailed explanation the reader is referred to [6].

3.3. Trade-off Results

Every concept was scored on all criteria as can be seen in table 3.3. It can be seen that concept 1 won the trade-off with a final score of 7.62. Concept 2 and 3 obtained a score of 5.70 and 7.14 respectively.

Table 3.3: Trade-off execution table

	Cost	**Sustainability**	**Performance**	**Risk**	**Complexity**	**Final Score (0-10)**
Weight	11%	3%	28%	40%	17%	-
Concept 1	9	6.8	7.5	8	6.6	7.62
Concept 2	9	6.4	5.5	5.5	4.6	5.70
Concept 3	8	6.4	8.75	7	4.8	7.14

3.4. Trade-off Sensitivity

Now that the trade-off was performed, the trade-off sensitivity was analyzed. This was done via four tests. First, the concepts scores were changed from a linear 0-10 scale to a coarser scoring of 0, 5 or 10. With these scores concept 1 remained the winner. Secondly, it was computed by how much each of the criterion weights should be changed to obtain a different trade-off winner. If a criterion weight would have to change by more than 50 % it would be considered that the trade-off is too sensitive to a change in inputs. It was found for all criteria that the weights would have to be changed by at least 100 % to result in a new winner. Thirdly, the importance of each individual criterion was determined by setting the criteria weight to 0 one by one. Only if both the risk and complexity weights were set to 0, concept 2 overtook concept 1. Finally, the flight time for concept 1 was set to 0, thereby lowering the performance score. It was found that in this case it still won. Thus it can be concluded that concept 1 won justly. The *Air Travel Drone* concept will therefore be considered for this final design stage. More details on the other two concepts, including layout, and mass and power budgets, can be found in the midterm report[6].

4
Market Analysis

In this chapter the market the drone will operate in will be analyzed. Furthermore, a forecast for the ARTIS' market position is done. Finally, a SWOT analysis for the ARTIS regarding the market position is constructed.

4.1. Market Overview

An autonomous drone for inspection in hazardous environments lends itself to versatile applications in different markets. Short-term efforts will be put in the design with the aircraft maintenance industry segment in mind, but long-term profitability could emerge from other markets. All values reported in the present section are reported in US dollars, to keep consistency with the cited sources.

4.1.1. Maintenance Industry

Among others, inspection and maintenance in the oil industry [9] is one of the main drivers of autonomous services demand. Also, petrochemical, off-shore solutions, power plants for radiation monitoring and civil applications like pipes inspection are on the lookout for innovative robotics solutions due to the increased productivity and lower health dangers [40].

One feasible possibility for the ARTIS services is maintenance demand in the marine segment. In the case of marine robotics, the Remotely Operated Vehicles (ROVs) market was foreseen to grow at near 14% CAGR (compound annual growth rate) reaching around $3.12 B billion by 2023 [42].

Furthermore, in the US only, from the overall operating revenue of $65.76 million in the railroad industry, $63.24 million were designated for freight transportation, requiring quick, reliable and possibly autonomous solutions [10].

4.1.2. Aeronautical Industry

The current target industry for the ARTIS is the Maintenance, Repair and Overhaul (MRO) aircraft industry. For an autonomous inspection service driving efficiency, reducing shutdown times and increasing productivity, substantial opportunities for profitability are expected from a market estimated at $77 B in 2018 [11] and $82 B the following year [43].

Furthermore, the overall MRO market spend is foreseen to reach $96.9 B in 2021, striking to a market evaluation of $115 B before the end of the next decade [11]. The current commercial air transport fleet is the main catalyst in the MRO market worth, followed by military aircraft. A visual overview of the MRO spend of 2015 can be seen in figure 4.1 [44].

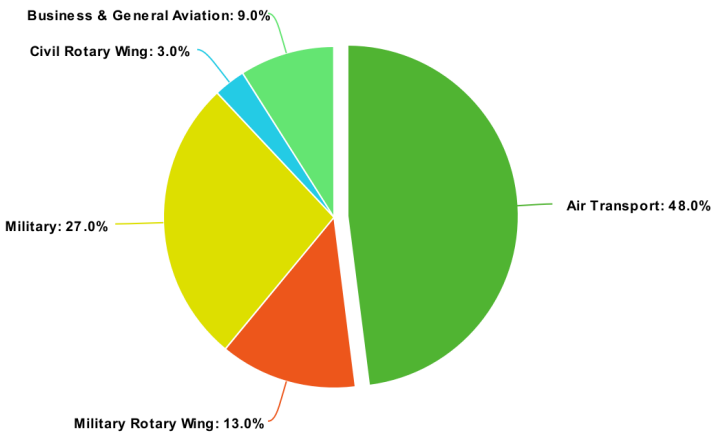


Figure 4.1: 2015 MRO spend for the global aerospace industry

The initial project stage being under commission from a military and aerospace defence company, a good market consolidation in the military segment could see later market expansion towards the commercial aviation sector, capturing higher MRO market share.

Growth is projected in various market segments, with 3.7 % annual growth rate for the global aircraft fleet, 3.8 % increase in 2018 for air cargo demand, and 4.4 % growth rate for cargo capacity. Demand for MRO services is expected to increase at a steady pace, with the market forecasting 4.5 % annual growth rate and a market value of $115 B in 2028 [11, 43].

Attention shall be paid at administrative practises and compliance. The predicted growth could be hindered by uncertainties over global regulations, particularly aimed at reducing emissions, and the implications of international agreements and policies (see Paris Climate Accord and Brexit) [45]. The result might be impeded trades, causing delays in international shipping, or transport of product components. Also, reduced emission regulations might cause an increase in price decrease in availability of transport, as this accounts for a large part of global emissions [45].

Hinting at the need for autonomous practises, EASA maintenance regulations are going in the direction of more robust rules related to safety management requirements [46].

Strict regulatory practises posing serious barriers to market entry, limit the number of companies joining the MRO sector. In particular, significant restrictions are present to entries in the wide- and narrow-body market [47]. This will not affect the company directly, but might decrease the future market size, as it affects potential client-airliners with wide-and narrow bodies.

Having assessed the size, growth and structure of the industry, attention was given to barriers of entry and regulations. Excellent opportunities for increased demand in the MRO sector are present. Aerospace executives strongly believe the most efficient approach to counteract the rise in fuel and material costs is via leveraging data analytic services and through predictive maintenance and aircraft health monitoring via innovative practices [11]. Opportunities for the ARTIS to gain large market shares are evident. Furthermore, the previously mentioned cargo growth is backed by Amazon company, planning to build a $1.5 B international air cargo hub, supporting a fleet of more than 100 air cargo aircraft [11].

The possibilities for the ARTIS to gain profits from the cargo growth, military expansion and commercial aviation sector shall be further analyzed in later stages of the project development.

4.1.3. Lockheed Martin

Lockheed Martin corporation is currently working on 45 aircraft projects, 3 of which are viable for fuel tank inspection with ARTIS (the C-130,C-5 and LM-100J). Since the focus aircraft for the ARTIS project is the C-130, the opportunities regarding this aircraft will be studied in more detail. Other aircraft models provide scalability and growth opportunities in the future.

Lockheed Martin now provides various frameworks for the maintenance of the C-130. They provide a 'Contract Maintenance Team' where experts perform and aid customer planning and execution of maintenance operations. Furthermore, there are 13 'Lockheed Martin Hercules Service Centres' (LMHS, or MROC) around the world with capabilities for maintenance services. Finally, Lockheed Martin offers advanced training to customers in maintenance operations for the C-130 [12]. This existing infrastructure provides opportunities for the phased implementation of ARTIS in existing maintenance procedures.

Lockheed Martin's 2019 first quarter results show an increase in aeronautical sector sales of 26% compared to the same period in 2018, similarly, the sectors operational profit increased by 23%. The majority of the increase is related to the F-35 program, yet increased sales can also be attributed to the C-130 project with 5 C-130J deliveries in the first quarter of 2019 compared to 3 deliveries in the same period of 2018 [48].

4.2. ARTIS Market Forecast

The market forecast for the ARTIS project highlights the estimated market characteristics and costs of the product.

The discussion thus far hinted at two pockets of growth for the ARTIS. Successful market penetration will aim at two distinct segments: military aerospace and defence companies, with Lockheed Martin being the target customer, and the commercial aviation sector.

Due to the confidential nature of Lockheed Martins financial data, a series of informed assumption were determined. These assumptions were supported and driven by the projects Lockheed supervisor via personal communication [14].

Lockheed produces approximately 24 C-130 aircraft per year. During production and test flights, fuel tank maintenance is scarcely required. Once delivered and during operation, the C-130 fuel tank must be inspected on average once every 4 flights. Estimating that each week 500 C-130 aircraft are flown (world-wide) 4 times per week, fuel tank inspections would be performed 500 times per week (approximately 26 000 per year)[14] .

Regarding commercial aviation, there were more than 25 000 commercial passenger aircraft in service in 2019 [16]. It can be assumed that the fuel tank of a commercial aircraft is inspected every time a maintenance A-check[1] is performed, and that on average an A-check is performed once every month on every aircraft in service. Given this data, it can be estimated that 300 000 fuel tank inspections are performed per year. Given that the commercial aircraft industry is expected to grow, this value would naturally increase with time.

An important point of analysis for the market forecast lies in estimating the industry costs currently incurred by Lockheed Martin for the C-130 aircraft maintenance. Workforce employment data for the maintenance and repair industry implies average hourly earnings of $25 [13], to which insurance, other human resources and non-variable costs shall be added. As confirmed by a Lockheed Martin representative [14], this amounts to approximately $150 per person. Assuming 2 workers per inspections, 1 inside the fuel tank and 1 providing support from the outside, and an approximate duration of 5.5 hours [14], yields $1650 required per inspection. The previous rational concluded that 26 000 C-130 fuel tank inspections are performed per year, thus leading to a total cost of $42.9 millions per year.

Given the number of inspection for commercial aircraft 300 000 and the same inspection costs as previously discussed, fuel tank inspections for the commercial industry would incur costs of half a billion USD per year.

These calculation assume an equal cost distribution and inspection duration per aircraft, it is thus an estimate to get a rough idea of the industry costs.

A comprehensive understanding of the profitability opportunities of the ARTIS shall include an analysis of the competition. Existing and emerging companies shall be looked at to effectively assess eventual market shares for the developed product.

Of the existing companies, only a few are heavily involved in the MRO sector, mainly focusing on surface cases like fuselage inspection. Table 4.1 shows a list of companies making use of the mentioned technology.

Table 4.1: Current companies making use of autonomous drones for inspection and maintenance

Company Name	**Relevance**	**Url**
Aiir Innovations	Software development for automatic detection of aircraft engines surface defects.	`https://www.aiir.nl/`
Flyability	Indoor inspection of inaccessible places using drones.	`https://www.flyability.com/`
Luftronix	Drone powered inspections of aircraft for external use cases.	`https://www.luftronix.com/`
Airbus	Drone-based advanced indoor inspection	`https://www.airbus.com/`
Shell	Advanced plant inspection at high altitudes	`https://www.shell.com/`
Xyrec	Robotic solution for aircraft coating maintenance	`https://www.xyrec.com/`

[1] A-check consists of a general inspection of the interior and exterior of the aircraft. It is typically performed every two weeks or every month [15].

Of the above, none specifies the intent of aircraft fuel tank inspection. Thus, the limit to the obtainable market shares are mostly internal or driven by the market dynamics overview above. An analysis of the strengths, weaknesses, opportunities and threats (i.e. SWOT) faced by the proposed ARTIS solution is offered in section 4.3.

Aiming at an extended market reach, long-term efforts will strive to capture market shares in comparable industries. From the discussion above, the marine segment, off-shore solutions and freight transportation will represent a linchpin step in the ARTIS revenues stream.

4.3. SWOT Analysis

A detailed market analysis requires the identification of strengths, weaknesses, opportunities and threats of the project. The analysis is performed here in the form of a SWOT diagram as seen in figure 4.2.

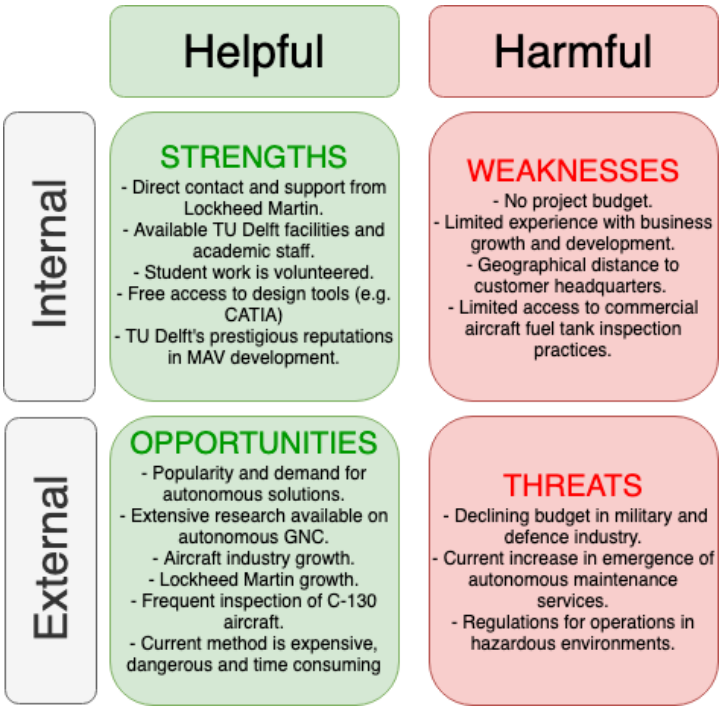


Figure 4.2: Market analysis SWOT diagram

Focusing on the strengths and opportunities, there are a various points that strengthen the potential of the project. The direct contact and support from Lockheed Martin ensures the existence of a large client, therefore eliminating the need to spend resources to attract the initial investor/client. The availability of TU Delft's resources, support and prestige bring further advantages to the project. The growth of the aircraft industry and of Lockheed Martin bring further opportunities since spending and demand should increase in parallel, and with most organizations looking to implement and develop autonomous solutions, ARTIS is in a good position to attract customers.

The SWOT analysis is helpful in identifying the strengths of the team, which shall be directed towards market shares acquisition in the defense aircraft maintenance drone industry during the first years of operations. It further supports the chosen focus on Lockheed Martin as the key stakeholder.

5
Requirements Analysis

In this chapter an analysis on both the subsystem and the user requirements will be performed. First, the functional breakdown structure and the functional flow diagram are provided to identify subsystem requirements. Next, the most important requirements per subsystems will be listed to obtain a better understanding for the design process. Then, the user requirements and constraints will be stated, showing how the user should operate and maintain the drone.

5.1. Functional Breakdown Structure and Functional Flow Diagram

The functional breakdown and functional flow diagram of the ARTIS' mission are depicted in the two following pages. The goal of those diagrams is aid in the understanding of how the mission is performed, which can help in further identification of subsystem requirements. Keeping these diagrams in mind, the requirements describing the ARTIS mission are set out in section 5.2 and section 5.3.

5.2. Subsystem Requirements

In this section, the most important requirement per subsystem are listed. In the following chapters, ARTIS will be designed in accordance to these requirements. For more global system requirements the reader is referred to [6].

5.2.1. Requirements for all subsystems

- **AMD-NF-CNST-Sfty-03 (D1)**: No parts of the drone that are exposed to the fumes in the fuel tank shall have a temperature that exceeds the auto-ignition temperature of the fumes.
- **AMD-NF-CNST-Sfty-02 (S,D1)**: The drone shall not create sparks during inspection.
- **AMD-F-ST&M-Join-03 (D1)**: The joints in the drone shall be air and liquid tight.
- **AMD-NF-CNST-Regu-02 (D1, K)**: The equipment in the drone shall only consist of ATEX/NEC approved equipment for operation in at least a Zone 1 working environment.
- **AMD-F-CNST-Material-01 (D1)**: The materials used in the drone which are exposed to the environment shall be resistant in the given environment.

5.2.2. Aerodynamic requirements

- **AMD-F-AERO-Ent-01 (D2):** This requirement has been redacted due to patent reasons.
- **AMD-F-AERO-Thr-01:** The drone shall be able to provide a thrust to weight ratio of 2.
- **AMD-F-AERO-Duc-01:** The drone shall use ducted fans.
- **AMD-F-AERO-Thr-02:** The drone shall be able to provide different thrusts per propeller.

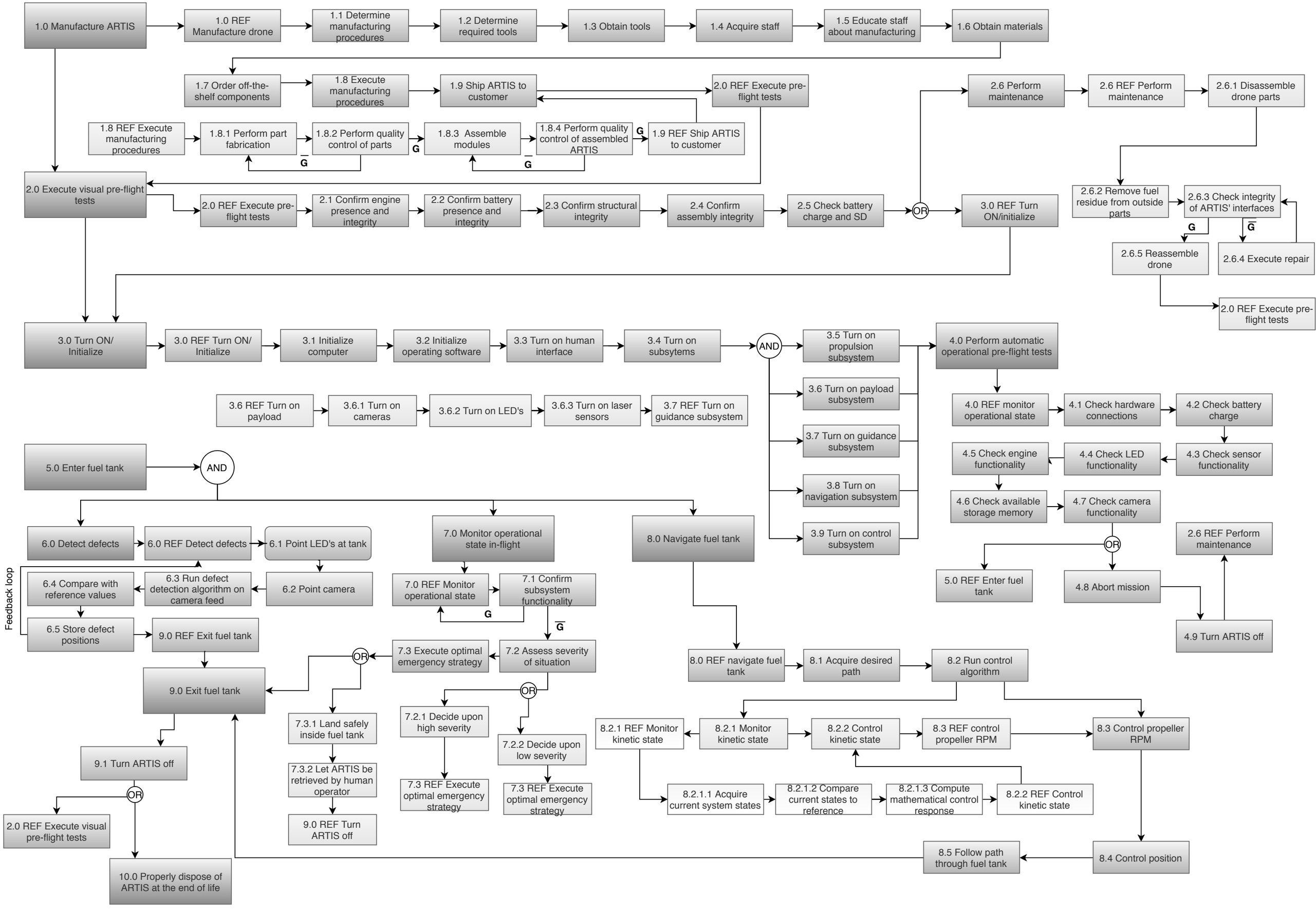
5.2.3. Structural requirements

- **AMD-NF-CNST-Sfty-01 (S,D1)**: The drone shall not damage the inside of the fuel tank during inspection.
- **AMD-NF-CNST-Sfty-06 (S,D1)**: Impact pressure on the fuel tank structures must be reduced to 57 kg/cm^2 in all circumstances.
- **AMD-NF-CNST-Sfty-07 (S,D1)**:Impact force on the structure shall be less than the forces occurring when 150 pound people inspect the structure.
- **AMD-F-ST&M-Join-01 (D1)**: The joints in the drone shall be able to sustain operational loads.
- **AMD-F-ST&M-Imp-01 (S,Y)**: The drone structure shall not plastically deform due to an impact of 1 kg/cm^2.

5.2.4. Electronics requirements

- **AMD-F-P&P-Endu (S,Y)**: The drone shall have a minimum flight time of 10 minutes.
- **AMD-NF-LFCC-Dura-05**: The total electrical power needed for one drone inspection shall not exceed the total electrical power needed for venting the tank for (To obtain from LM) hours.

- Inspect Fuel Tank
  - 1. Exit Fuel Tank
    - 1.1 Locate exit point
    - 1.2 Turn off
  - 2. Enter Fuel Tank
  - 3. Detect Defects in tank
    - 3.1 Navigate fuel tank
      - 3.1.1 Establish path
        - 3.1.1.1 Control position
          - 3.1.1.1.1 Control kinetic state
            - 3.1.1.1.1.1 Determine control variable input
              - 3.1.1.1.1.1.1 Determine rpm for each propeller
          - 3.1.1.1.2 Monitor kinetic state
            - 3.1.1.1.2.1 Acquire kinetic state variables
              - 3.1.1.1.2.1.1 Acquire height
              - 3.1.1.1.2.1.2 Acquire velocity X-direction
              - 3.1.1.1.2.1.3 Acquire velocity Y-direction
              - 3.1.1.1.2.1.4 Acquire velocity Z-direction
              - 3.1.1.1.2.1.5 Acquire Q1
              - 3.1.1.1.2.1.6 Acquire Q2
              - 3.1.1.1.2.1.7 Acquire Q3
              - 3.1.1.1.2.1.8 Acquire Q4
              - 3.1.1.1.2.1.9 Acquire q
              - 3.1.1.1.2.1.10 Acquire p
              - 3.1.1.1.2.1.11 Acquire r
    - 3.2 Detect green masks on white background
      - 3.2.1 Brighten tank
      - 3.2.2 Point camera
      - 3.2.3 Store position
      - 3.2.4 Store footage on SD carsds
      - 3.2.5 Run algorithm on camera feed
        - 3.2.5.1 Compare with reference
  - 4. Turn on/Initialize
    - 4.1 Initialize computer
    - 4.2 Turn on human interface
    - 4.3 Turn on subsystem
      - 4.3.1 Turn on propulsion
      - 4.3.2 Turn on payload
        - 4.3.2.1 Turn on LEDs
        - 4.3.2.2 Turn on Cameras
        - 4.3.2.3 Turn on laser sensors
      - 4.3.3 Turn on human interface
      - 4.3.4 Turn on power system
      - 4.3.5 Turn on computer
        - 4.3.5.1 Load operating software
      - 4.3.6 Turn on Navigation
      - 4.3.6 Turn on guidance
      - 4.3.7 Turn on control
  - 5. Monitor operational state
    - 5.1 Check sensor functionality
    - 5.2 Check hardware connections
    - 5.3 Check battery charge
    - 5.4 Check light functionality
    - 5.5 Check engine functionality
    - 5.6 Check available storage memory
    - 5.7 Check camera functionality
    - 5.8 Determine wheter or not to declare emergency
  - 6. Perform pre flight checks
    - 6.1 Check structural integrity
      - 6.1.1 Check presence of engines
      - 6.1.2 Check battery integrity
    - 6.2 Check assembly integrity
    - 6.3 Perform maintenance as needed
  - 7. Manufacture drone
    - 7.1 Order materials
    - 7.2 Order off-the-shelf components
    - 7.3 Obtain tools
      - 7.3.1 Determine required tools
    - 7.4 Acquire staff
      - 7.4.1 Instruct staff on manufacturing procedures
        - 7.4.1.1 Develop manufacturing procedures
    - 7.5 Perform quality control
      - 7.5.1 Perform quality control on parts
      - 7.5.2 Perform quality control on final assembly
    - 7.6 Execute manufacturing procedures
      - 7.6.1 Assemble modules
  - 8. Properly dispose of the drone
    - 8.1 Disassemble drone
    - 8.2 Send parts to relevant recycling plants
      - 8.2.1 Recycle foam
      - 8.2.2 Recycle battery
  - 9. Run emergency procedures when necessary
    - 9.1 Evaluate emergency strategy
      - 9.1.1 Evaluate closest viable exit point
      - 9.1.2 Evaluate closest viable landing space
      - 9.1.3 Evaluate possibility of backtracking
    - 9.2 Evaluate severity of situation
      - 9.2.1 Decide upon low severity
      - 9.2.2 Decide upon high severity
    - 9.3 Execute emergency strategy
      - 9.3.1 Land safely
      - 9.3.2 Exit tank
      - 9.3.3 Signal that an emergency has occured
      - 9.3.4 Turn off
  - 10. Perform maintenance
    - 10.1 Disassemble ARTIS
    - 10.2 Remove any fuel residue
    - 10.3 Reassemble ARTIS
    - 10.4 Execute any needed repairs
    - 10.5 Check integrity of interfaces


1.0 Manufacture ARTIS
1.0 REF Manufacture drone
1.1 Determine manufacturing procedures
1.2 Determine required tools
1.3 Obtain tools
1.4 Acquire staff
1.5 Educate staff about manufacturing
1.6 Obtain materials
1.7 Order off-the-shelf components
1.8 Execute manufacturing procedures
1.9 Ship ARTIS to customer
2.0 REF Execute pre-flight tests
1.8 REF Execute manufacturing procedures
1.8.1 Perform part fabrication
1.8.2 Perform quality control of parts
1.8.3 Assemble modules
1.8.4 Perform quality control of assembled ARTIS
1.9 REF Ship ARTIS to customer
G
G
G
G
2.6 Perform maintenance
2.6 REF Perform maintenance
2.6.1 Disassemble drone parts
2.0 Execute visual pre-flight tests
2.0 REF Execute pre-flight tests
2.1 Confirm engine presence and integrity
2.2 Confirm battery presence and integrity
2.3 Confirm structural integrity
2.4 Confirm assembly integrity
2.5 Check battery charge and SD
OR
3.0 REF Turn ON/initialize
2.6.2 Remove fuel residue from outside parts
2.6.3 Check integrity of ARTIS' interfaces
G
G
2.6.5 Reassemble drone
2.6.4 Execute repair
2.0 REF Execute pre-flight tests
3.0 Turn ON/ Initialize
3.0 REF Turn ON/ Initialize
3.1 Initialize computer
3.2 Initialize operating software
3.3 Turn on human interface
3.4 Turn on subsytems
AND
3.5 Turn on propulsion subsystem
3.6 Turn on payload subsystem
3.7 Turn on guidance subsystem
3.8 Turn on navigation subsystem
3.9 Turn on control subsystem
4.0 Perform automatic operational pre-flight tests
3.6 REF Turn on payload
3.6.1 Turn on cameras
3.6.2 Turn on LED's
3.6.3 Turn on laser sensors
3.7 REF Turn on guidance subsystem
4.0 REF monitor operational state
4.1 Check hardware connections
4.2 Check battery charge
4.3 Check sensor functionality
4.4 Check LED functionality
4.5 Check engine functionality
4.6 Check available storage memory
4.7 Check camera functionality
OR
5.0 REF Enter fuel tank
4.8 Abort mission
2.6 REF Perform maintenance
4.9 Turn ARTIS off
5.0 Enter fuel tank
AND
6.0 Detect defects
6.0 REF Detect defects
6.1 Point LED's at tank
6.2 Point camera
6.3 Run defect detection algorithm on camera feed
6.4 Compare with reference values
6.5 Store defect positions
Feedback loop
9.0 REF Exit fuel tank
9.0 Exit fuel tank
9.1 Turn ARTIS off
OR
2.0 REF Execute visual pre-flight tests
10.0 Properly dispose of ARTIS at the end of life
7.0 Monitor operational state in-flight
7.0 REF Monitor operational state
7.1 Confirm subsystem functionality
G
G
7.2 Assess severity of situation
OR
7.2.1 Decide upon high severity
7.2.2 Decide upon low severity
7.3 REF Execute optimal emergency strategy
7.3 REF Execute optimal emergency strategy
7.3 Execute optimal emergency strategy
OR
7.3.1 Land safely inside fuel tank
7.3.2 Let ARTIS be retrieved by human operator
9.0 REF Turn ARTIS off
8.0 Navigate fuel tank
8.0 REF navigate fuel tank
8.1 Acquire desired path
8.2 Run control algorithm
8.2.1 REF Monitor kinetic state
8.2.1 Monitor kinetic state
8.2.2 Control kinetic state
8.3 REF control propeller RPM
8.3 Control propeller RPM
8.2.1.1 Acquire current system states
8.2.1.2 Compare current states to reference
8.2.1.3 Compute mathematical control response
8.2.2 REF Control kinetic state
8.5 Follow path through fuel tank
8.4 Control position

- **AMD-NF-LFCC-Bat-05**: The drone battery shall be rechargeable in 1 h.
- **AMD-NF-LFCC-Bat-02**: The drone shall be able to notify the operator if the battery charge level gets below 5 %.
- **AMD-NF-LFCC-Bat-03**: The drone battery shall be replaceable in 15 minutes.

5.2.5. Payload requirements

- **AMD-F-PAYL-Cam-01 (D2)**: The camera shall provide RGB images.
- **AMD-F-PAYL-Cam-02 (S,Y)**: The camera shall have a resolution of at least 5 MP/ft^2.
- **AMD-F-PAYL-Cam-03 (Y)**: The drone camera images shall be linked to the corresponding position and orientation of the drone within the tank.
- **AMD-F-PAYL-Ill-01 (Y)**: The illumination source shall provide an illumination of 100 lm/m^2.

5.2.6. Guidance, Navigation and Control requirements

- **AMD-F-GN&C-Navi-02 (D2)**: The drone shall be able to stabilize itself following a disturbance input.
- **AMD-F-GN&C-Navi-03 (Y)**: The drone shall be able to navigate without the use of GPS.
- **AMD-F-GN&C-Cont-05 (Y)**: The drone shall be able to obtain a pointing accuracy of 5°.
- **AMD-F-GN&C-Guid-05 (Y)**: The drone shall be able to determine its orientation within 2°.
- **AMD-F-GN&C-Guid-03 (Y)**: The drone shall be able to determine its own position within 0.5 cm with respect to an obstacle.
- **AMD-F-GN&C-Navi-01 (Y)**: The drone shall update its position at least every 0.1 s.
- **AMD-F-GN&C-Guid-02 (Y)**: The drone shall be able avoid obstacles within the fuel tank.
- **AMD-F-GN&C-Guid-01 (Y)**: The drone shall be able to follow a predetermined path with an accuracy of 5 cm within the fuel tank.

5.3. User Requirements and Constraints

Next to the technical requirements regarding the drone design, the user requirements are also listed. These requirements describe how the user should operate the product. This includes operation, maintenance, storage, transport, and disposal. In addition, user constraints are stated, containing information on how the user should not operate and use the drone.

5.3.1. Operational requirements

- **AMD-NF-LFCC-Ope-01:** The user shall ensure that the drone is being operated in the operational temperature range.
- **AMD-NF-LFCC-Ope-02:** Before operation, the user shall ensure that the drone turns on as required.
- **AMD-NF-LFCC-Ope-03:** Before operation, the user shall ensure that all drone subsystems operate as intented.
- **AMD-NF-LFCC-Ope-04:** Before operation, the user shall ensure that no more than 5 % of the fuel remains in the tank after draining.

5.3.2. Maintenance requirements

- **AMD-NF-LFCC-Main-01**: The drone shall get a full maintenance check after every four inspections.
- **AMD-NF-LFCC-Main-02**: During full drone maintenance, it shall take no longer than 2 hours to ensure all drone functions work accordingly.
- **AMD-NF-LFCC-Main-03**: During full drone maintenance, all electronic interfaces shall be checked to ensure a electric current runs through the cables.
- **AMD-NF-LFCC-Main-04**: During drone maintenance, the drone shall be able to be disassembled and cleaned from fuels residues.
- **AMD-NF-LFCC-Rep-03**: During drone inspection, a certified drone repair employee should be present within a radius of 10 km at all times.
- **AMD-NF-LFCC-Rep-04**: During drone inspection, certified drone repair tools should be present within a radius of 10 km at all times.

5.3.3. Storage and Transport requirements

- **AMD-NF-LFCC-Stor-01:** The drone shall be stored in the operational temperature range.
- **AMD-NF-LFCC-Stor-02:** The drone shall be stored in a dry environment.
- **AMD-NF-LFCC-Dura-04**: The drone battery shall be charged with electrical power generated by a source that does not emit substances that are harmful to the environment.
- **AMD-NF-LFCC-Rep-02**: At least one spare part of each of the drone parts shall be present within 10 km of where the inspection occurs.
- **AMD-NF-LFCC-Tran-01:** The drone shall be transported in the same environment as it is stored in.

5.3.4. Disposal requirements

- **AMD-NF-LFCC-Bat-01**: Replaced batteries shall be given to an external party that is specialized in recycling batteries in a sustainable way.
- **AMD-NF-LFCC-Disp-01**: All drone materials for which recyclability processes exist shall be recycled at the end of life of the drone.
- **AMD-NF-LFCC-Disp-02**: All drone materials that cannot be recycled shall be dumped in a way that does not release additional harmful substances for the environment.

5.3.5. User constraints

- **AMD-Constraint-01:** The user shall not operate the drone in environments where the drone is not designed for.
- **AMD-Constraint-02:** The user shall not reverse engineer the drone to use the design for their own profit.
- **AMD-Constraint-03:** The user shall not sell the drone to third parties without written permission from the manufacturer.
- **AMD-Constraint-04:** The user shall not let third parties use the drone without written permission from the manufacturer.
- **AMD-Constraint-05:** The user shall not let the drone be operated by personnel without a drone operations certification.

6
TECHNICAL RISK ANALYSIS

An overview of the likelihood and consequences of unmitigated risks faced during the drone's operational life is given in this section. The identified hazards are all high-severity, being characterized by a likelihood of occurrence and severity higher than medium before any design mitigation activity is performed.

Previous reports ([17] and [6]) thoroughly assessed every risk, the ones unlikely to occur and inducing superficial damages included. The focus of the present discussion is thus on likely risks causing major damages and hindering the mission success.

The definitions used for severity ranking, i.e. how severe are the consequences to the drone effectiveness due to the occurrence of the mentioned event, and likelihood ranking are reported in table 6.1 and based on a qualitative assessment.

Table 6.1: Likelihood and severity ranking definitions for risk assessment.

Likelihood Ranking	High	Frequent occurrence
	Medium-high	Rate of occurrence of event is of concern for drone operations
	Medium	Rate of occurrence of event is of minimum concern for drone operations
	Medium-low	Rate of occurrence of event is not of concern for drone operations
	Low	Event unlikely to occur
Severity Ranking	Catastrophic	Damage well beyond repair, with hazardous financial and health consequences for human personnel
	High	Damage beyond repair leading to mission failure
	Medium-high	Major damage, can be repaired but will be temporarily unavailable
	Medium	Major damage, can be repaired and remains partially functional
	Medium-low	Moderate damage, can be repaired and remains fully functional
	Low	Superficial damage

A1. ERRORS WITH SYSTEMS INTEGRATION.

Likelihood Assessment: Medium likelihood. This event occurs for example in the case that the electronic circuitry is wrongly integrated, or the propellers are miss-aligned. The integration of different subsystems is a challenging aspect of working in a large team on the same design. Through experience, it is evident that clear communication and effective system engineering is required in order to prevent such risks.
Severity Assessment: Medium severity. The consequences of this event will in most cases be detected before the product is delivered. Errors with system integration are usually evident and uncovered during the testing phase. Upon delivery only small integration errors could be expected with no significant consequences.

A2. EXPLOSION INDUCED BY SYSTEM MALFUNCTIONING.

Likelihood Assessment: Medium-high likelihood. Given the hazardous condition found within (unvented) aircraft fuel tanks, the likelihood that electronic devices operating on the inside of the tank (without appropriate design characteristics) cause the ignition of fuel vapour is substantial. Explosions can be triggered by sparks in the ARTIS motors, other electronic systems and from collisions.
Severity Assessment: Catastrophic consequence. The explosion or ignition of fuel tank fumes can cause dangerous an uncontrollable fires in the aircraft's wing structure. Such an event would not only lead to mission failure, it would also as a consequence incur financial loses and furthermore put the safety of bystanders at risk.

A3. INDUCED SPARKS DURING OPERATIONS.

Likelihood Assessment: Medium-high likelihood. Sparks during operations are likely to occur if no preventive measures are put in place. Sparks can be generated due to static electricity on the drone, due to motors and electronic systems.
Severity Assessment: Catastrophic severity. Sparks would cause the ignition of surrounding fuel vapours, leading to extremely dangerous hazards for all the personnel supervising the inspection operations, with damages well beyond repair.

A4. Broken parts due to collisions remain undetected inside fuel tank.

Likelihood Assessment: Medium-low likelihood. The event that parts of the drone break inside the fuel tank can be triggered by collision within the tank or due to fatigue. In both cases the user should easily become aware that the tank should be inspected for broken parts. The ARTIS concept design has a relative low complexity, thus minimizing the number of parts that can be expelled.
Severity Assessment: Medium-high severity. In the event that broken parts remain undetected inside the fuel tank, the part may be ingested by the fuel system and thus maintenance would be required to clear the fuel system. Since fuel systems are equipped with filters, the part would in no case reach the aircraft's engine.

A5. Communication failure.

Likelihood Assessment: Medium likelihood. In cases where communication with the drone is required, such as recall of the drone from operations due to external emergencies, the communication system may fail. Communication with ARTIS is expected to be minimal since the drone operates mostly on autonomous principals. Yet due to the contained environment of the fuel tank, the likelihood that interference blocks communication is considerable.
Severity Assessment: Medium severity. Depending on the occasion leading to the need for communication with the drone the severity of this event is variable. As a worst case scenario, an emergency such as a misunderstanding with regards to the fuel drainage can cause the drone to operate in a tank which still contains fuel and thus poses a high risk of ignition if communication fails to call the drone out.

A6. Damage to fuel tank structure due to collisions.

Likelihood Assessment: High likelihood. Due to the nature of the mission, the drone is expected to navigate through the intricate and tight environment of the fuel tank. Given the small margins for error, the likelihood that a collision occurs is high, especially when navigating through ribs.
Severity Assessment: Medium-high severity. Depending on the nature of the collision, the damage is variable. Yet without preventive measures, in most cases the damage can be expected to be more than acceptable. Although damage due to drone collision would never have a critical consequence on the wing structure due to the low drone mass, an impact may nevertheless require repair.

C1. Vehicle gets stuck inside fuel tank.

Likelihood Assessment: Medium likelihood. The possibility that the vehicle gets stuck inside the fuel tank arises in this case when the drone is still fully functional but either trapped in confined space without the ability to exit, or in the case where parts of the drone get caught in certain fuel tank features.
Severity Assessment: Medium-high consequence. The consequence of this event is the possibility of fuel tank damage as well as the possibility of timely and expensive retrieval operations.

C2. Loss of stability and control.

Likelihood Assessment: Medium likelihood. Control modes are theoretically constructed with the possibility for inaccurate definitions. The mission profile commissioned by Lockheed Martin is specific, nevertheless, unknown disturbances are still to be expected and are challenging to plan for. The presence of fumes, aerodynamic flows and structural elements result in a challenging operational environment for the control of the drone.
Severity Assessment: Medium-high consequence. Inaccurate control modes could cause incorrect drone responses thus enabling instability and unexpected responses. As a consequence the drone can become uncontrollable, stuck inside the fuel tank and damages to the inside environment may occur.

C3. Data corruption during on-board processing.

Likelihood Assessment: Medium likelihood. Due to the complexity of embedded algorithms and limited on board computation and storage capabilities it is possible that data is corrupted, overwritten or exceeds available storage capacity.
Severity Assessment: Medium-high severity. Data corruption could result in complete failure of the GNC subsystem in case critical data is involved as it would not be able to process images and manage data nominally.

C4. Inaccurate position knowledge.

Likelihood Assessment: Medium-high likelihood. Autonomous localization algorithms typically generate cumulative position errors, if this error exceeds a certain threshold, the drone may not have enough knowledge to accurately determine its true position.
Severity Assessment: Medium-high severity. If the drone losses position knowledge during operations it may not be able to navigate within the fuel tank. Furthermore it could also collide with the fuel tank if it does not know its true position.

C5. Inaccurate obstacle avoidance and path planning.

Likelihood Assessment: Medium-high likelihood. The difficulty of implementing accurate solutions for guidance and navigation purposes lowers the probability of a collision-free path. Furthermore, the amount of obstacles to be encountered and the wall vicinity causing aerodynamic interference is an another key point of concern.
Severity Assessment: Medium-high consequence. The consequences for this event are both the failure to fully inspect the fuel tank and the possibility of colliding and damaging the fuel tank. The first consequence can lead to the possibility of a false negative inspection.

C6. Inaccurate detection of fuel tank damage/corrosion.

Likelihood Assessment: Medium likelihood. Image processing and feature extraction rely on an accurate characterization of edges, corners and interest points in images [49]. For such tasks, complex algorithms are required, thus bringing the likelihood level of inaccurate detection to a high level when no mitigation is performed.
Severity Assessment: Medium severity. Inaccurately identifying corrosive spots inside the fuel tank environment results in compromising mission success. No catastrophic consequences would be observed since no damages to the wing box or the drone itself would be encountered and since inspections are performed regularly (every four days) the feature should be eventually detected. Furthermore, ARTIS is expected to perform better than manual inspection, which is also prone to inaccurate definitions based on human error.

E1. Electronics subsystem failure.

Likelihood Assessment: Low likelihood. Short circuits may arise, leading to partial electronics subsystem failure during nominal operations. Such event is very unlikely due to the flight readiness of the already proposed design [6], expected to improve even further in later design iterations.
Severity Assessment: High severity. As outlined in other mentioned risks, power distribution represent a major point of concern in the ARTIS design. Failure of the electronics subsystem would thus be catastrophic, leading to immediate mission abortion, the ARTIS being stuck in the fuel tank, and collisions with items in the hazardous environment.

P1. Power required exceeds available power.

likelihood Assessment: Medium likelihood. Maximum operating conditions require higher power supplies than nominal ones, thus impeding efficient power distribution.
Severity Assessment: Medium severity. Malfunctioning of subsystems due to inefficient power delivery harms the ARTIS capabilities. Below average performance of subsystems and, in the worst case scenario, short circuits or obstruction of power stores are possible consequences.

P2. Propulsion sub-system failure.

Likelihood Assessment: Medium likelihood. Electrical engines considered for the ATIS, being flight-proven and with a relatively high Technology Readiness Level, can be considered reliable from a mechanical standpoint. Given the constant motion of the propulsive units and fatigue effects due to extensive operational cycles, unexpected failures are possible.
Severity Assessment: High severity. Detrimental consequences on the ARTIS performance is caused by mechanical failure of the propulsive unit. Loss of stability and control is the consequence of not having fail-safe mechanisms or redundant architectures in place. Collisions with fuel tank elements would then occur.

P3. Insufficient range and endurance during flight operations.

Likelihood Assessment: Medium-low likelihood. This risk is dependent on several factors, from power distribution to overall subsystem integrations. For instance, excessive power demands by the onboard computer would limit the power availability for propulsion, thus complicating the range estimations.
Severity Assessment: Medium-high severity. The ARTIS would get stuck inside the fuel tank due to insufficient range and endurance during flight operations. Multiple charges would then be required to complete the mission inspection.

PL1. Image quality is not sufficient for analysis.

Likelihood Assessment: Medium-low likelihood. The ARTIS cameras are flight-proven solutions [6], thus it is unlikely an insufficient RGB image quality is produced.
Severity Assessment: High severity. Inadequate image quality would prevent the ARTIS from detecting corrosion and, in the worst case, hinder the GNC subsystem operations.

PL2. Payload mass or power exceeds set budget.

Likelihood Assessment: Medium-high likelihood. The on-board computer shall be multifunctional, demanding high power demands and possibly a heavier weight than estimated during the first design iteration, thus exceeding the computed budgets.
Severity Assessment: Medium severity. Mass and power distributions above the estimated contingency values can lead to an under-performing design. The different functions being met during the design stage, unexpected consequences could appear during the ARTIS operations, e.g. a limited flight time due to unpredicted excessive power demands.

PL3. Payload subsystem failure upon impact.

Likelihood Assessment: Following on from requirements **AMD-F-ST&M-Join-01** and **AMD-F-ST&M-Imp-01**, damage inducing collisions with the fuel tank are unlikely to occur, and if it were to happen, limited damages are expected in case of requirements compliance, thus lowering the likelihood of such risk.
Severity Assessment: High severity. If a payload item were to fail, severe consequences would occur. For instance, camera malfunctioning could prevent the GNC subsystem to work with sufficient accuracy, thus hindering the ARTIS' path planning and obstacle avoidance capabilities. Mission failure would then be the inevitable consequence.

S1. Undetected manufacturing flaws.

Likelihood Assessment: Medium-low likelihood. Although unlikely, manufacturing defects could go unseen in the complex drone layout. In particular, wiring of electronics or misorientation of propellers and motors could be difficult to spot with a naked eye.
Severity Assessment: Medium-high severity. If not detected and handled early in the ARTIS production stage, manufacturing flaws, depending on their extent on the overall performance, could lead to mission abortion.

S2. Unexpected manufacturing resource limitations.

Likelihood Assessment: Medium-high likelihood. The manufacturing procedures of the ARTIS will be performed in-house or via third parties involved in the production. In the former case, insufficient knowledge of manufacturing procedures may end up in excessive demands of resources. In the latter, the design not being a market standard could result in limited supplies.
Severity Assessment: Medium-low severity. The design life-cycle would be slowed down, not proceeding to the next stage in the production until new stock supplies are delivered.

S3. Structural sub-system failure upon impact.

Likelihood Assessment Medium likelihood. The hazardous environment present in the inside of fuel tanks is characterized by the presence of gasses, structural items and cross-sectional beams, compromising the movements of the drone due to collision probabilities.
Severity Assessment High severity. Collision leading to failure of structural components would lead to mission failure and prevent the drone abilities to conduct its operations.

The risks can be prioritized by use of a risk matrix, reporting the likelihood of occurrence on one axis and the severity on the other, as shown in figure 6.1.

Risk Mitigation and compliance strategy

All above mentioned risks will be carefully kept in mind throughout the ARTIS design process and mitigated with proposed solutions. These solutions will be discussed during the detailed design, when relevant subsystems are being developed. An assessment of the overall effect of the risk mitigation solution, including a post-mitigation risk map, is given in chapter 16. Furthermore, in this section all risks which are not mitigated during the detailed design will be discussed and managed appropriately.

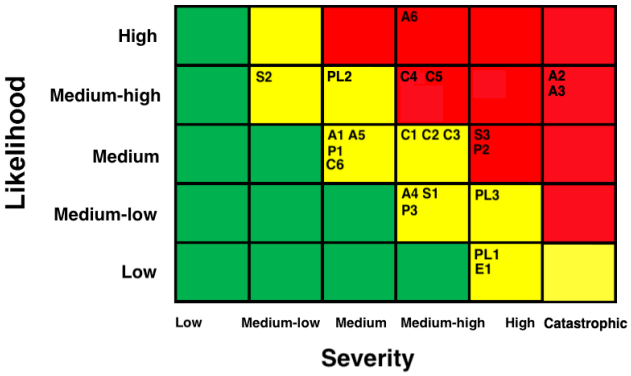


Figure 6.1: Pre-mitigation Risk Map

7

Resource and Budget Allocation

Technical resources, such as computing capacity or mass, are typically scarce and have the tendency to grow in an undesired direction as the design progresses to higher levels of maturity [18]. It is important that these scarce technical resources are identified and that a contingency management strategy is formulated to allow for undesired technical resource growth as the design process advances. A preliminary technical resource and budget breakdown, including a contingency plan, was presented in the baseline review [17]. This chapter provides a revised and more detailed technical resource and budget breakdown, as well as a concurrent contingency plan, for the final phase of the design.

7.1. Technical Resource and Budgets Definition

The requirements presented in chapter 5 can be analyzed to distinguish potential technical resources that are relevant with respect to the scope of the mission. This revealed that the technical resources that relate driving and key requirements most notably comprise the flight time, total mass, and hardware cost. Since only using the total mass as a technical resource may result in additional difficulty in keeping track of the origin of the mass increase in later design stages, it was decided to break down the total mass into subsystem masses. This subsystem mass breakdown for the technical resource and budget allocation was defined based on the initial subsystem mass specified in the conceptual design stages [6]. The subsystem breakdown after the midterm phase is shown in figure 7.1, and this percentage-wise breakdown is intended to be adhered to for increasing levels of design maturity, and as such a potential total mass increase.

The design specification values for flight time, total mass, and hardware cost are directly dictated by requirements **AMD-F-P&P-Endu, AMD-NF-CNST-Mass-01** ([6]), and **AMD-NF-CNST-Cost-01** ([6]), respectively. The contingencies starting at the midterm design stage follow directly from the values and estimates established for the conceptual design trade-off. Since the conceptual design was sized for an initial total mass of 270 g while the design specification value for total mass is 300 g per requirement, the contingency for total mass at this design stage becomes 10 %. Similarly, the contingencies for flight time, with a design specification value of 600 s, and total hardware cost, with a design specification value of €2500, become 4 % and 75 %, respectively, based on midterm design stage values of 624s and €615.

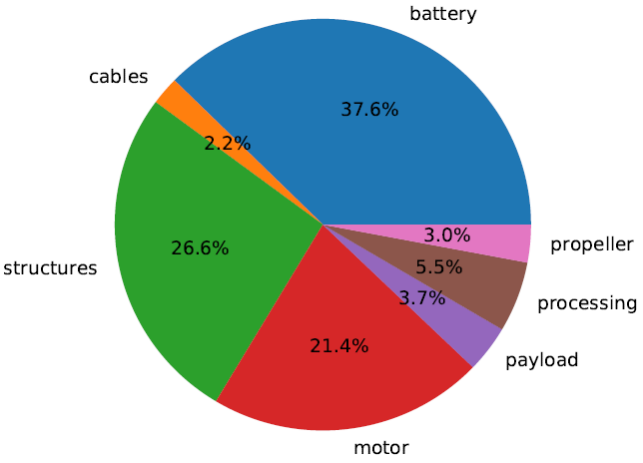


Figure 7.1: Mass breakdown per subsystem at the start of the detailed design phase

The contingencies of the technical resources can subsequently be evaluated by virtue of defining an ordinal scale for the different levels of design maturity, as is shown below for increasing levels of design maturity [18].

1. Lay-out calculations (MTR), i.e. the design stage that is achieved at the end of the midterm review.
2. Final design calculation (FDR), i.e the design stage that is achieved at the end of the final design review
3. Prototype development tests
4. Measurement qualification tests
5. Flight hardware tests

The resource and budget allocation can then be summarized as is shown in table 7.1. The contingency values for later design stages were acquired by scaling the contingency values that were achieved at the end of the

midterm design stage. Evidently, it holds that the contingencies for all the subsystem masses are the same, as they are all derived from the contingency on the total mass at the end of the midterm stage.

In table 7.1, the orange rows denote the design stages that will be achieved within the scope of the DSE, whereas the blue rows denote the design stages that will only be done post-DSE.

Table 7.1: Resource & budget allocation including contingency allowances for the final design phase

	Contingency (%)							
Design Maturity	Total Mass	Structural Mass	Payload Mass	Battery Mass	Propulsion Unit Mass	Cabling Mass	Flight Time	Hardware Costs
Lay-out calculations (MTR)	10	10	10	10	10	10	20	15
Final design calculations (FDR)	5	5	5	5	5	5	5	5
Prototype development tests	2	2	2	2	2	2	2	2
Measurement qualification tests	1	1	1	1	1	1	1	1
Flight hardware tests	0	0	0	0	0	0	0	0
Design specification value	300 [g]	80 [g]	28 [g]	113 [g]	73 [g]	7 [g]	600 [s]	2500 [€]

7.2. Contingency Management Strategy

To see how the design has propagated after the midterm report, the contingencies are visualized in figure 7.2. Here, one can compare for all three TPM (technical performance measurement) parameters, whether or not the design at this stage meets the target values specified in [17]. For the total mass, the target value and actual value are identical, causing the blue and yellow dashed line which is composed of two individual lines. The total mass meets the target value exactly as the iteration tool used in [6] uses flight time as input to end up at a certain final mass. This caused the flight time to not meet the target value. At this stage, the flight time is 10 minutes and 24 seconds, instead of the target of 12 minutes. In the later stage, care should thus be taken to ensure that the flight time never ends up being lower than 10 minutes. Finally, it can be seen that the hardware cost dropped down a lot. Where a value of €2125 was allowed at this point, the total hardware cost only turned out to be €615

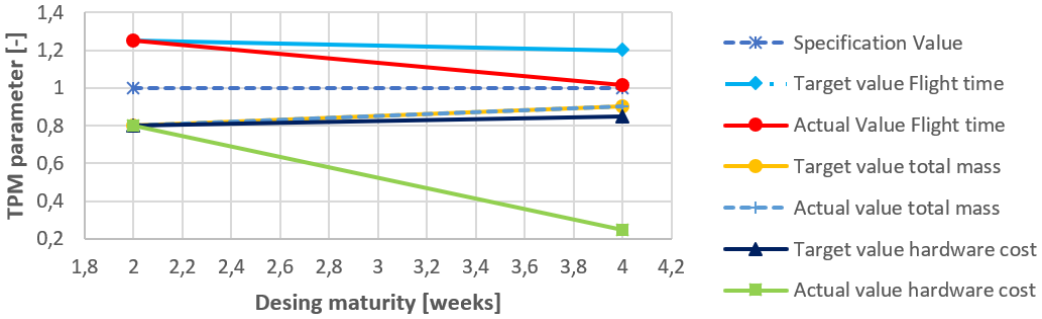


Figure 7.2: Design contingencies after the midterm phase

For the final design stage, the flight time should be carefully managed to prevent it from going down even further. The total mass will be kept at the target value to ensure that the mass will never exceed the specification value. Lastly, the hardware cost is allowed to increase. This allows the design to include more expensive and possibly better hardware options. The resource and budget allocation of the final design will be elaborated on in section 11.4. Here, the development of the total mass, total flight time and total hardware cost will be further analyzed.

8
DESIGN AND MATERIALS

This chapter gives an overview of the main ARTIS hardware being designed. This includes the propellers, the motors and the surrounding structure. Also, suitable materials are analyzed. In addition, a CFD analysis is performed to gain more insight on the interference of the airflow on the ARTIS design.

8.1. AERODYNAMIC CHARACTERISTICS

In this section the aerodynamic characteristics of the ARTIS will be analyzed. First, the propellers of the drone will be sized via blade theory and a program called JBLADE. Later, a CFD analysis performed on the aerodynamic performance of the ducts and structural elements and wall interference is proposed.

8.1.1. AERODYNAMIC PROPELLER SIZING

The main aspect of the drone's aerodynamics are the thrust generation and power requirement of the propellers. A theoretical derivation to calculate the ideal power required for a given thrust can be done using momentum theory. Equation (8.1) relates the ideal power (P_{ideal}) to the thrust (T), propeller disk area (A_d) and fluid density (ρ) [50].

$$P_{ideal} = \sqrt{\frac{T^3}{2\rho A_d}} \tag{8.1}$$

The propeller area should be as large as possible to minimize power usage due to their inverse relationship. However, because of the size requirement (**AMD-NF-CNST-Size-01**) the propeller size is limited geometrically. With a maximum diagonal of 30 cm, the maximum propeller diameter is 12.4 cm, as seen in figure 8.1. To accommodate for propeller ducts and a layer of protective foam, the maximum diameter is reduced to 10 cm.

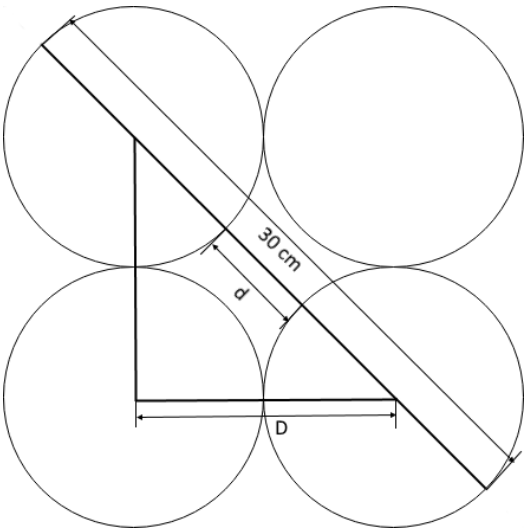


Figure 8.1: Geometry of quad-propeller configuration. Using Pythagoras results in d = 5.2 cm and D = 12.4 cm

The University of Illinois at Urbana-Champaign hosts a publicly accessible database of UAV propeller performance and geometry data [19]. Given the size requirement, several reference propellers of similar size were selected from this database. Static (hover) performance is characterized by static thrust and power coefficients (C_{T_0} and C_{P_0}) and Figure of Merit (FOM). These coefficients are normalized using fluid density, revolutions per second and propeller diameter as seen in equation (8.2) and equation (8.3). The Figure of Merit is used to quantify efficiency, the conventional efficiency factor $\eta = \frac{TV_\infty}{P}$ being unsuitable for static conditions ($V_\infty = 0$). It is calculated using equation (8.4) [51].

$$C_{T_0} = \frac{T}{\rho n^2 D^4} \tag{8.2}$$

$$C_{P_0} = \frac{P}{\rho n^3 D^4} \tag{8.3}$$

$$FOM = \sqrt{\frac{2}{\pi}} \frac{{C_{T_0}}^{3/2}}{C_{P_0}} \tag{8.4}$$

The propeller with the highest FOM and a similar size is the GWS DD 4025. It has 2 blades, a diameter of 10.16 cm and a FOM of 0.572. In [51] it is suggested that the tested propellers in the database are not optimal for hover conditions and that a FOM between 0.7 and 0.8 should be possible. The chord and twist angle distributions along the blade span are known for the GWS DD 4025 and can be seen in figure 8.2, but the airfoil is not. Therefore, an attempt to improve this propeller's FOM is done by selecting an optimal airfoil and potentially changing the pitch angle of the blades.

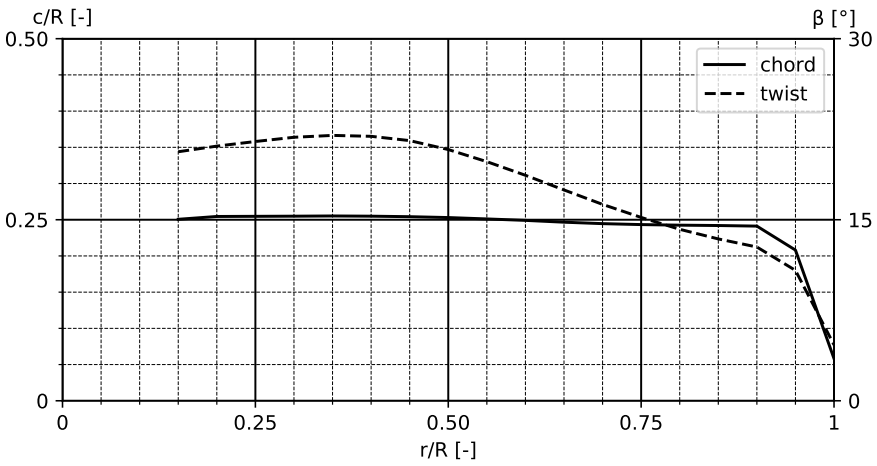


Figure 8.2: Chord and twist distribution of GWS DD 4025

To select an airfoil, the first step is to determine the operating Reynolds and Mach numbers. Because of the wide range of airspeeds along a propeller blade, the blade section at 75 % of the blade span is commonly used as reference [52]. Using the test results from the database, the GWS DD 4025 has to spin at an RPM of approximately 13 000 to produce 75 g of thrust, which translates to a reference velocity of 45.8 m/s. Using ISA atmospheric conditions at sea level [53] results in a Reynolds number of approximately 40 000 and a Mach number of 0.137.

The criteria for an optimal airfoil for a propeller that minimizes power requirement are a high maximum C_l-C_d ratio (to minimize drag) and a high $C_{l_{max}}$ (to minimize RPM). Three airfoils are selected: Eppler E61, Eppler E63 and AH-7-476 [20]. The lift-drag polars of each of these airfoils at the operating conditions described above are generated using XFOIL [54], with 250 panels for each, and can be seen in figure 8.3.

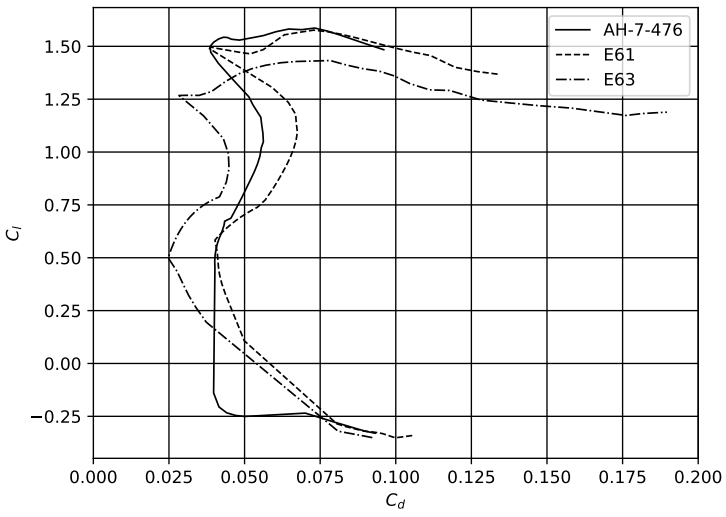
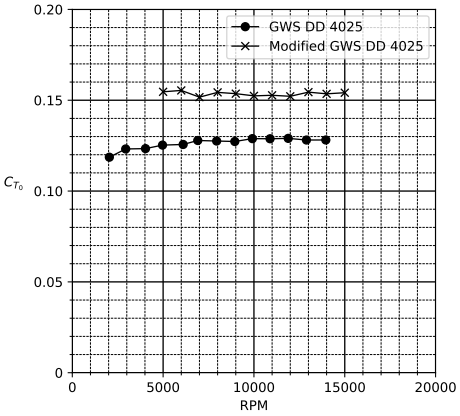


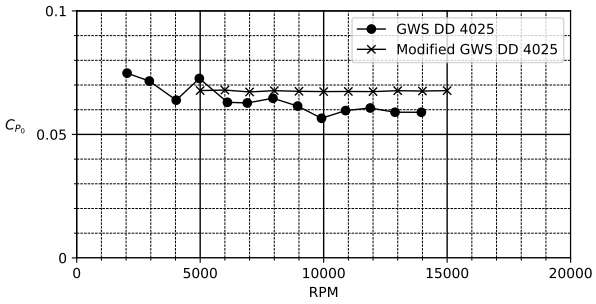
Figure 8.3: Lift-drag polars of selected airfoils, $Re = 40000$, $M = 0.137$

The Eppler E63 has the highest maximum C_l-C_d ratio of the three, but its $C_{l_{max}}$ is significantly lower than the other two. The Eppler E61's polar is very similar to that of the E63, with a slightly lower maximum C_l-C_d ratio and a higher $C_{l_{max}}$. The AH-7-476 has almost the same maximum C_l-C_d ratio and $C_{l_{max}}$ as the E61 but the average C_d value is lower. Since propellers are subject to a wide range of angles of attack, the AH-7-476 has the upper hand and is therefore selected.

Using a program called JBLADE, the performance of the propeller can be computed. It uses Blade Element Momentum Theory (BEMT) to calculate thrust generation and power required at various values of RPM and V_∞ [55]. The propeller is analyzed with an RPM range of 5000 to 15000 in increments of 1000, and a V_∞ of 0.01 m/s (the lowest possible setting in JBLADE). The resulting C_{T_0} and C_{P_0} values can be seen in figure 8.4. The average FOM of the modified propeller is 0.711, an improvement of 24 %.



(a) Static thrust coefficients



(b) Static power coefficients

Figure 8.4: Comparison of UIUC propeller test and JBLADE analysis of modified propeller

To reduce the risk of damaging the fuel tank with the propellers during a potential collision (see risk **A6** from chapter 6), a duct with a height of 17.5 mm and a thickness of 0.2 mm has been added around each propeller to protect the blade tips. The ducts have been sized based on the remaining mass budget after structural design was completed. The duct with the maximum length was designed within the mass constraints. Not only does it reduce the severity of impact damage, but it also has positive effects on the propulsive performance by reducing the effect of tip vortices [56]. Also by curving the inlet, additional lift is generated by the duct, so an inlet lip curvature of 2.5 mm is applied.

8.1.2. Aerodynamic interactions

Aerodynamic interactions can be divided in two categories: interactions with the environment and interactions with the structural components.

8.1.2.1. Environment interaction

Because the drone is flying in a fuel tank, it is expected that there will be flow interaction with the walls, namely the influence of wall effects on the power loading of the propellers. The power loading (PL) is defined as the ratio between thrust generated and power required, as seen in equation (8.5). A higher PL is beneficial for the drone's endurance, but acts as a disturbance on the drone's control system and could pose an associated risk.

$$PL = \frac{T}{P} \tag{8.5}$$

Wall influence can be separated in four categories: in ground effect (IGE), in ceiling effect (ICE), in channel effect (IChE) and in wall effect (IWE). All of these effects have been investigated in [1], using a controlled experimental setup.

For IGE, the relation between height-radius ratio (h/R) on the power loading is plotted in figure 8.5. At a h/R ratio of 2.5, the PL starts to increase regardless of disk loading (DL, thrust per disk area), and increases more significantly as height decreases. The propeller used in this design has a radius of 5 cm, equating to a height of 12.5 cm above the ground. Because of the counteracting nature of the ground effect, and the drone does not intend to land in the fuel tank, it does not pose a significant risk.

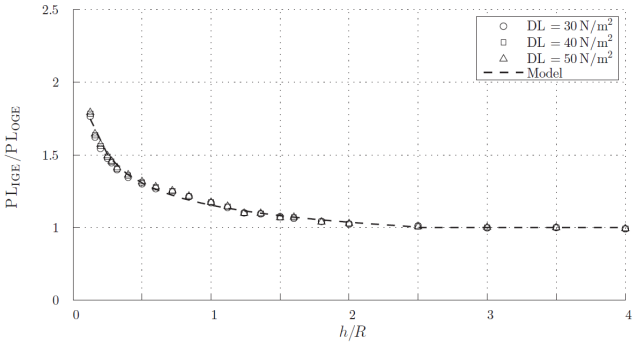


Figure 8.5: PL increase as function of distance above ground [1]

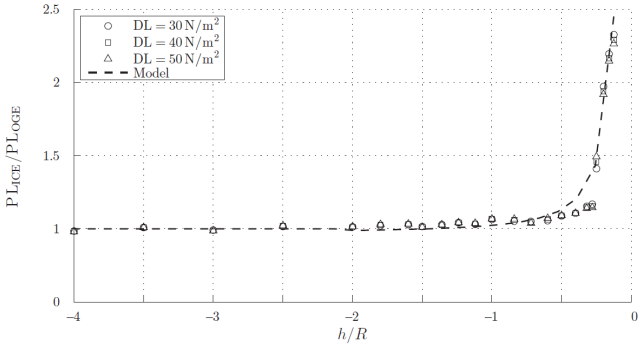


Figure 8.6: PL increase as function of distance below ceiling [1]

A similar relationship has been derived for ICE, which is visualized in figure 8.6. Here the PL stays rather constant, up to a height-radius ratio of -1 below the ceiling where it increases slightly. Above h/R = -0.25 the PL shoots up significantly. In this case this increase in PL does pose a risk, as the drone will get sucked against the ceiling if it can't correct its altitude in time. The foam that sticks out above the propellers provides a clearance of 2 cm (h/R = −0.4) in case of a collision with the ceiling, but nevertheless this should be avoided as much as possible to mitigate the risk of losing controllability.

In channel effect is the influence of both the ground and ceiling on the propeller's PL. As demonstrated by [1], this effect can be assumed as a superposition of IGE and ICE. The last effect is due to the proximity of a wall

perpendicular to the plane of the propeller disk. The change in PL is insignificant and therefore not shown here. The propeller jet deflects slightly towards the wall, as seen in but the lateral force induced by this could not be measured and is therefore deemed negligible.

8.1.2.2. STRUCTURE INTERACTION

To investigate the interaction with structural components, a CFD simulation was performed in ANSYS Fluent, using a slightly simplified model of one of the drone arms. It is expected that the structural elements and part of the compartment underneath the duct will generate a non-neglible amount of drag caused by the airflow coming out of the duct. The aim of this simulation is to quantify the amount of this drag.

The geometry is enclosed in a fluid cube with a rib length of 1 m. To simplify the mesh generation and flow calculation, the propeller has been left out and replaced with a fan boundary condition causing a pressure jump of 81.64 N, similar to the wall interaction simulation described above.

The fluid is meshed using tetrahedra with a standard size of 15 mm, growth rate of 1.2, minimum defeature size of 0.48 mm and minimum curvature size 0.96 mm. Because of the complex geometry of the drone arm, a face sizing of 0.5 mm is applied to all the surfaces of the geometry except the top and bottom of the duct where a sizing of 0.1 mm is used to capture the thickness of the duct wall. This resulted in a mesh of approx. 10 740 000 elements with a minimum and average orthogonality of 0.15 and 0.77 respectively.

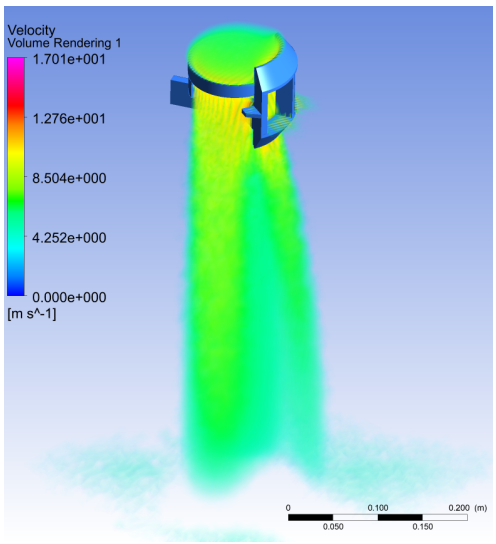


Figure 8.7: Volume rendering of velocity magnitude

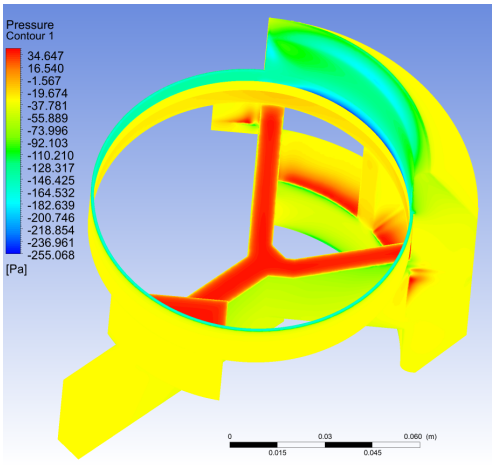


Figure 8.8: Pressure contours on drone structure

Besides from the pressure conditions described above, all boundaries are set as walls with a no-slip condition. Turbulence is modelled with a realizable k-ϵ model with standard wall functions [57]. A steady-state solution is calculated using a SIMPLE algorithm [58] with second-order discretization methods. Residual convergence criteria are set to 1×10^{-4} for all variables. When all convergence criteria were met, the total drag force calculated by ANSYS alternated between two values with an amplitude in the order of 10^{-4} , which is deemed accurate enough for this purpose.

A volume rendering of the flow velocity magnitude around the drone arm can be seen in figure 8.7. The velocity threshold for rendering is set at 4 m/s (lower velocities are not rendered). The flow accelerates up- and downstream of the duct due to the pressure differences, and a split in the flow can be observed downstream. This split is caused by the Y-shaped structure below the duct. There is also a bit of flow coming out of the open holes on the side of the structure. The pressure distribution on the drone arm structure can be seen in figure 8.8. As expected, the high pressure areas are directly below the duct, which causes pressure drag on the arm in vertical direction. Also the side holes cause some high pressure generation due to the flow going through them. The total drag (pressure and friction drag) on the structure is found to be 0.13 N, or 19.6 % of the generated thrust. This implies that the propellers will have to spin approximately 9.4 % faster (assuming a constant C_{T_0}). It should noted however that the curved duct inlet was omitted in the simplification of the drone arm structure, which would partially negate the downwards drag.

8.2. Motor Characteristics

The motor required to fly the ARTIS depends on the total mass of the drone. To select a proper motor, the eCalc database was analyzed for the optimal motor. This database consists of over 7000 suitable motors used in quadcopter design. For each of those motors, the power consumption was computed. As an additional requirement, the motors are required to provide a thrust to weight ratio of at least 2 as stated in **AMD-F-AERO-Thr-01**.

The power required to hover using a motor is determined by the rotational speed Ω, as is shown by equation (8.6) and equation (8.7), whereas the torque Q_m of the propeller is found using equation (8.8). In these equations, the diameter is set to 10.16 cm based on figure 8.1. The k_T is equivalent to the C_{T_0} of the propeller (0.1289), as well as the C_{p_0} having a value of 0.06734:

$$\Omega = 2\pi\sqrt{\frac{\left(\frac{W}{4}\right)}{k_T \rho D^4}} \quad (8.6) \qquad P_{shaft} = C_{p_0}\rho\left(\frac{\Omega}{2\pi}\right)^2 D^5 \quad (8.7) \qquad Q_m = \frac{P}{\Omega} \quad (8.8)$$

The electric voltage requirement of the motor is calculated for each motor using equation (8.9) (K_v, $\Re$, and

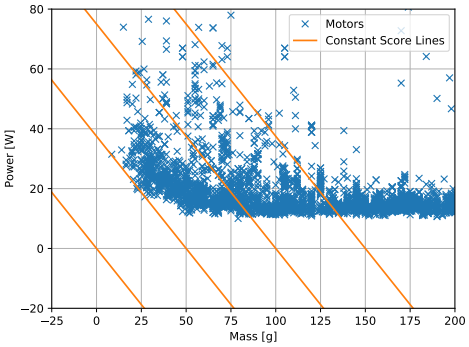


Figure 8.9: Trade off scoring for motor selection

Figure 8.10: The RotorX RX1404

i_0 from eCalc database [59]). Equation (8.9) is solved numerically for v. Based on this, the current and power can be computed with equation (8.10) and equation (8.11).

$$Q_m = \left[\left(v - \frac{\Omega}{K_v}\right)\frac{1}{\Re} - i_0\right]\frac{1}{K_v} \quad (8.9) \qquad i = \left(v - \frac{K_v}{\Omega}\right)\frac{1}{\Re} \quad (8.10) \qquad P_{electric} = i \cdot v \quad (8.11)$$

To select the optimal motor a trade-off between power consumption and the mass of the motor is performed. The optimal motor is the one that minimizes the following scoring function:

$$\Phi = m_{motor} + P_{motor} \cdot \left(\frac{t_{flight}}{\rho_{battery}}\right) \quad (8.12)$$

Equation (8.12) computes the sum of the motor mass and the battery mass needed to deliver the power of the motor. See figure 8.9 for a visual representation of this trade-off. The blue crosses represent all motors that are located in the eCalc database. The orange lines are lines with constant Φ, so the optimal motor is located in the direction perpendicular to the orange line towards to bottom left [6].

During an iterative process, the optimal motor with lowest Φ score was selected as the motor for the ARTIS. This motor is the RotorX RX1404, shown in figure 8.10 having a mass of 8.5 g including cabling from the ESC to the motor [21]. This motor requires a 3-4S voltage input and has a stator diameter of 14 mm. In addition, this motor can provide a thrust of at least 2 times the drone weight. In hover, the power consumption per motor is 19.12 W. As this is a brushless electromotor, it does not create sparks during operation, thereby reducing risk **A3**. Furthermore, as this motor can provide at least twice the required power to hover, risk **P1** is mitigated.

8.3. Material Characteristics

A detailed overview of suitable materials for the manufacturing of the drone was presented in [6]. In this section the material choice is presented.

Structural design starts from determining the loads the structure needs to withstand. The critical loads on the structure are those occurring during impacts. As the flight speed inside the tank will be low, the most powerful impact can occur at the end of a drop from the tank ceiling. The maximum height of the C-130 fuel tank is approximated to be 0.75 m. Accounting for manufacturing defects, such as small notches left over from 3D printing support structure or local thinning of material, leading to stress concentrations, a safety factor of 4/3 is applied. Therefore, the structure is designed to resist the ultimate loads occurring at the end of 1 m drop.

The energy to be absorbed by the structure during impact is the kinetic energy of the drone. The kinetic energy is equal to the potential energy at the height of 1m: $m_{drone} \cdot g \cdot h = 2.943$J, rounded up to 3 J for a conservative estimation. The structure needs to be able to absorb this energy at the maximum impact

pressure possible, which can occur when the drone falls on a sharp edge such as a stringer corner. Relying of the toughness of the structural material alone to absorb the impact energy is not reliable (difficult theoretical task on the edge of material science and research), nor practical as a larger volume of material is required than in the regular case of carrying the loads. Using polymeric foam as the main impact energy absorption material is the lightest solution. The foam toughness required is directly proportional with the available space for foam. As the drone has to fit through an elliptical hole of 30.48 cm by 20.32 cm, with a propeller diameter of 10 cm, the maximum space available for the structure around each duct is 2 cm. Allowing for a structural width of 1 cm, the maximum space available in height for the foam on the side is 1 cm. Assuming that the contact area is infinitesimally small (point load), and assuming the foam resistance to compression is linear on average throughout the compression range, the work of the foam reaction force is equal to $W_{reaction} = F_{reaction} \cdot \Delta X$, where ΔX is the compression range assumed to equal the foam thickness. $W_{reaction}$ needs to match the impact energy in order for the entire energy to be absorbed, leading to a reaction force of $F_{reaction} = \frac{3\,\mathrm{J}}{0.01\,\mathrm{m}} = 300\,\mathrm{N}$. In order to determine the required foam toughness, with the impact force determined above, a realistic impact area of $1\,\mathrm{cm}^2$ is assumed, corresponding to a sharp edge impact. The impact pressure is $P_{impact} = \frac{F_{impact}}{A_{impact}} = \frac{300\,\mathrm{N}}{0.0001\,\mathrm{m}^2} = 3\,\mathrm{MPa}$. Based on the assumption that the foam compresses linearly, the foam yield stress equals $\sigma_{yield} = 3\,\mathrm{MPa}$. A rigid polymer foam (MD) with a density of approximately $140\,\mathrm{kg/m}^3$ will be used [22]. As more space is available on the top and bottom surfaces for foam, a different foam with a lower density can be used, thus leading to lower structural loads. The procedure can be repeated, leading to a maximum reaction force of 150 N and a foam density of $47\,\mathrm{kg/m}^3$ corresponding to rigid polymer foam (LD) [22].

The maximum forces exerted on the structure are high and can be applied from a multitude a positions, therefore an isotropic material with a high specific strength is required. CFR Nylon with 30 % carbon powder filling has an ultimate strength of 260 MPa and a density of $1280\,\mathrm{kg/m}^2$, leading to a specific strength of 0.2 [23], which is higher then regular Nylon with a specific strength of 0.071 or Aluminum 2045-T4 with a specific strength of 0.17 [60]. CFR Nylon is the material of choice.

Table 8.1: Material quantities used in the final design

Component	Material	Weight
Propeller	CFR 50 %	7 g
Frame	CFR 30 %	74 g
Dense foam	Polyurethane	12 g
Light foam	Polyurethane	11 g
Static dissipation foil	Aluminum	3.63 g

For static electricity dissipation, the duct, the struts and the part of the main compartment exposed to direct airflow from the propeller will be covered in aluminum foil with a thickness of 0.016 mm. This is the typical commercial thin aluminum foil thickness. The total surface area to be covered by foil is $0.084\,\mathrm{m}^2$, as calculated from the final CATIA model. The added mass of this foil is 3.63 g. This addition of aluminum makes the ducts conductive, making the drone easier to ground. The propellers will not be covered in foil, as the high aerodynamic forces might damage it. Instead, the propellers will be manufactured by injection molding with nylon reinforced with 50 % carbon long fibers [25]. The higher the carbon content, the more conductive the material is, therefore having a higher carbon infill percentage is a mitigation procedure [24]. More on the grounding of the drone will be stated in section 9.4. Table 8.1 summarizes the materials used and the quantities required for the final design.

Requirement **AMD-F-CNST-Material-01 (D1)** is met as all materials used are immune to the chemical environment. Requirements **AMD-NF-CNST-Sfty-01 (S,D1)**, **AMD-NF-CNST-Sfty-06 (S,D1)** and **AMD-NF-CNST-Sfty-07 (S,D1)** are met as the maximum impact pressure when foam absorbs the energy is $30.5\,\mathrm{kg/cm}^2$, while the maximum load exerted on the fuel tank is 300 N which is lower than the weight of a 150 pound person, and no sharp edges are exposed which could scratch the paint coating of the tank.

Risk **A6** is mitigated by meeting requirements **AMD-NF-CNST-Sfty-01 (S,D1)**, **AMD-NF-CNST-Sfty-06 (S,D1)** and **AMD-NF-CNST-Sfty-07 (S,D1)**. Risk **S1** is mitigated by using 3D printing as the main manufacturing process for the structure. This makes the process more predictable and the results repeatable as the process is largely automatic.

8.4. Structural Characteristics

In this section, the ARTIS' main structure is designed. This includes the main frame that connects the ducts, foam and main compartment. Also, the propellers designed aerodynamically in section 8.1.1 are analyzed for structural integrity in this section.

8.4.1. Frame Design

The input of the iterative drone arm structure design process are the maximum forces $F_{vertical}$ = 150 N, $F_{horizontal}$ = 300 N and the total mass budget of 90 g for the structure. All structures during the iteration process were approximated as frame structures made of straight beam elements. The final structure was approximated as in figure 8.11. A render of the final design is shown in figure 8.12 for ease of visualization, even though it was only created at the end of the process. A tool was written in MATLAB which uses the matrix displacement method to estimate the loads through all members, the maximum occurring Von Mises stresses and the total mass of the structure. All the information used about matrix methods is contained in [61]. The following information about matrix methods is taken from [61].

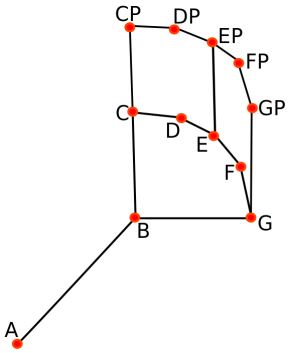


Figure 8.11: Frame approximation of the arm structure

Figure 8.12: Render of arm structure

Matrix methods are based on beam theory and conservation of energy methods (virtual work). The member end forces **Q** and the member end displacements **u** can be related with equation (8.13):

$$\boldsymbol{Q} = \boldsymbol{k}\boldsymbol{u} + \boldsymbol{Q_f} \tag{8.13}$$

where $\boldsymbol{Q_f}$ is the member fixed-end force vector expressed in local coordinates:

$$Q_f = \begin{bmatrix} FA_b & FS_{by} & FS_{bz} & FT_b & FM_{by} & FM_{bz} & FA_e & FS_{ey} & FS_{ez} & FT_e & FM_{ey} & FM_{ez} \end{bmatrix}^T \tag{8.14}$$

with FA_b, FS_{by}, FS_{bz}, FT_b, FM_{by}, FM_{bz} being the axial force, shear forces, torsional moment and moments around the axis Y and Z at end b, and the other components being the counterparts at the other end of the element, called e. $\boldsymbol{k}$ is the member stiffness matrix expressed in local coordinates:

$$k = \frac{E}{L^3}\begin{bmatrix} AL^2 & 0 & 0 & 0 & 0 & 0 & -AL^2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 12I_z & 0 & 0 & 0 & 6LI_z & 0 & -12I_z & 0 & 0 & 0 & 6LI_z \\ 0 & 0 & 12I_y & 0 & -6LI_y & 0 & 0 & 0 & -12I_y & 0 & -6LI_y & 0 \\ 0 & 0 & 0 & \frac{GJL^2}{E} & 0 & 0 & 0 & 0 & 0 & -\frac{GJL^2}{E} & 0 & 0 \\ 0 & 0 & -6LI_y & 0 & 4L^2I_y & 0 & 0 & 0 & 6LI_y & 0 & 2L^2I_y & 0 \\ 0 & 6LI_z & 0 & 0 & 0 & 4L^2I_z & 0 & -6LI_z & 0 & 0 & 0 & 2L^2I_z \\ -AL^2 & 0 & 0 & 0 & 0 & 0 & AL^2 & 0 & 0 & 0 & 0 & 0 \\ 0 & -12I_z & 0 & 0 & 0 & -6LI_z & 0 & 12I_z & 0 & 0 & 0 & -6LI_z \\ 0 & 0 & -12I_y & 0 & 6LI_y & 0 & 0 & 0 & 12I_y & 0 & 6LI_y & 0 \\ 0 & 0 & 0 & -\frac{GJL^2}{E} & 0 & 0 & 0 & 0 & 0 & \frac{GJL^2}{E} & 0 & 0 \\ 0 & 0 & -6LI_y & 0 & 2L^2I_y & 0 & 0 & 0 & 6LI_y & 0 & 4L^2I_y & 0 \\ 0 & 6LI_Z & 0 & 0 & 0 & 2L^2I_z & 0 & -6LI_z & 0 & 0 & 0 & 4L^2I_z \end{bmatrix} \tag{8.15}$$

In here, A is the cross sectional area, L is the element length, E is the modulus of elasticity of the material used, G is the shear modulus of the material used, J is the torsional stiffness, and I_y and I_z are the moments of inertia expressed in local coordinates. The transformation matrix between global coordinates and the local element coordinates can be constructed as in equation (8.16): In which $\boldsymbol{O}$ is a null 3x3 matrix, and the $\boldsymbol{r}$ matrix is expressed separately for regular elements and for elements with an angle of pitch of 90° because of the singularity in Euler angles. For regular elements, $\boldsymbol{r}$ is described by equation (8.17), and for vertical elements $\boldsymbol{r}$ is described by equation (8.18),

$$T = \begin{bmatrix} \boldsymbol{r} & O & O & O \\ O & \boldsymbol{r} & O & O \\ O & O & \boldsymbol{r} & O \\ O & O & O & \boldsymbol{r} \end{bmatrix} \tag{8.16}$$

$$\boldsymbol{r} = \begin{bmatrix} r_{xX} & r_{xY} & r_{xZ} \\ \frac{-r_{xX}r_{xY}cos(\psi)-r_{xZ}sin(\psi)}{\sqrt{r_{xX}^2+r_{xZ}^2}} & \sqrt{r_{xX}^2+r_{xZ}^2}cos(\psi) & \frac{-r_{xY}r_{xZ}cos(\psi)+r_{xX}sin(\psi)}{\sqrt{r_{xX}^2+r_{xZ}^2}} \\ \frac{-r_{xX}r_{xY}sin(\psi)-r_{xZ}cos(\psi)}{\sqrt{r_{xX}^2+r_{xZ}^2}} & -\sqrt{r_{xX}^2+r_{xZ}^2}cos(\psi) & \frac{-r_{xY}r_{xZ}sin(\psi)+r_{xX}cos(\psi)}{\sqrt{r_{xX}^2+r_{xZ}^2}} \end{bmatrix} \tag{8.17}$$

$$\boldsymbol{r} = \begin{bmatrix} 0 & r_{xY} & 0 \\ -r_{xY}cos(\psi) & 0 & sin(psi) \\ r_{xY}sin(\psi) & 0 & cos(\psi) \end{bmatrix} \tag{8.18}$$

Where r_{xX}, r_{xY}, r_{xZ} are defined as follows:

$$r_{xX} = cos(\theta_{xX}) \tag{8.19}$$

$$r_{xY} = cos(\theta_{xY}) \tag{8.20}$$

$$r_{xZ} = cos(\theta_{xZ}) \tag{8.21}$$

The global stiffness matrix can be obtained from the local stiffness matrix as follows: $\boldsymbol{K} = \boldsymbol{T}^T \cdot \boldsymbol{k} \cdot \boldsymbol{T}$. Similarly, the fixed-end force vector in global coordinates can be obtained by: $\boldsymbol{F_f} = \boldsymbol{T}^T \cdot \boldsymbol{Q_f}$. From the stiffness matrix of each element expressed in global coordinates, the system stiffness matrix has to be assembled. For the final iteration of the structure geometry, shown in Figure 8.11, there are 14 elements in the frame and 11 nodes. Each node has 6 degrees of freedom, leading to 66 degrees of freedom for the entire frame. Therefore, the system stiffness matrix is 66x66. The system to be solved by the computer is: $F_{global} = K_{system} \cdot D_{global}$, where F_{global} is the global system force vector (size 1x66) and D_{global} is the global system displacement vector (size 1x66). The global displacement vector can be disassembled into the local node displacements for each element, $\boldsymbol{d_{local}}$. From the end displacements of each member, using the local stiffness matrix, the forces and moments acting at both ends of each elements in local coordinates can be obtained with $F_{local} = k \cdot d_{local}$. From the local force vector, the thin walled beam assumption is made to calculate torsional stress with equation (8.23) and the flexure formula, equation (8.22). The maximum Von Mises stress occurring within a beam element can be calculated with equation (8.24).

$$\sigma = \frac{M \cdot y}{I} \tag{8.22}$$

$$\tau = \frac{T}{2 \cdot A \cdot t} \tag{8.23}$$

$$\sigma_v = \sqrt{0.5 \cdot [(\sigma_{11}-\sigma_{22})^2 + (\sigma_{22}-\sigma_{33})^2 + (\sigma_{33}-\sigma_{11})^2 + 6(\sigma_{12}^2+\sigma_{23}^2+\sigma_{31}^2)]} \tag{8.24}$$

For the last geometrical iteration of the structure, the element thicknesses were adapted iteratively until the calculated stress did not exceed the failure stress of the material, and the estimated mass of the structure remained within the budget.

Requirement **AMD-F-ST&M-Join-03 (D1)** is met as all structural members are 3D printed hollow tubes which are air and liquid tight. Risk **S1** can be mitigated by inspecting the structure after each manufacturing stage. After 3D printing the integrity of all elements can be visually inspected. The cabling holes can be obstructed and the structure pressurized to check that the structure is air and water tight and that no pore has been created during manufacturing that would jeopardize structural integrity and violate the regulation requirements. After foam is glued on the structure, the quality of the bonded joints can be checked by a worker. Risk **S2** related to manufacturing, as they relate to the structural elements can be mitigated. By 3D printing a prototype, manufacturing difficulties can be reduced significantly.

A detailed overview of risk **S3** is presented in table 8.2. The percentages are defined by the area protected by structure and exposed area. The propeller ducts are especially vulnerable from top and bottom impacts with loose hanging lines or cables, as they can get in the way of the propellers. The risk of damaging impact can be further reduced by reducing the risk of impacts occurring, with accurate navigation and control.

Table 8.2: Drone damaging collision probability with defined obstacles

Flat Surface	Stringer	Stringer at diagonal angle	Rib when crossing	Vertical Pipe	Horizontal Pipe	Loose hanging lines/cables	Exit/Entry Points
Safe 100 %	Safe from top 58 % Safe from side 70 % Safe from bottom 100 %	Same as stringer	Same as stringer	Same as stringer	Safe from top 58 % Safe from side 100 % Safe from bottom 100 %	Top/Bottom: Unsafe Safe from side 70 %	Same as stringer

8.4.2. Propeller Structural Design

Propeller design is dictated by aerodynamic performance and has to maintain structural integrity throughout the lifetime. Therefore, it is necessary to perform stress analysis of the propellers. The analysis for clockwise and anticlockwise propellers is identical, therefore only the clockwise propeller is presented.

The propeller blades can be analyzed as beams with a distributed loading and variable cross-section. The following loading cases are present:

- Axial loading due to the centrifugal force occurring during rotation.
- Bending loading due to the aerodynamic loading.

The axial loading distribution is determined using the mass distribution from the CATIA model of the propeller, and the angular velocity at the maximum RPM. The lift and drag forces are decomposed into axial and normal forces in the blade coordinate system. The load distribution is approximated conservatively as linear functions along the span, as in figure 8.13. In red is the lift distribution, and in blue are the resultant forces of the distribution. The centroid location and the area moments of inertia along the span are obtained by importing the airfoils into AutoCAD. Even though the tip cross sections are smaller, because of the low bending moment, the maximum tensile and compressive stresses occur at the root of the blades where the bending moment is maximal, and the tensile stress due to centrifugal forces are highest. From this conservative analysis performed by hand, the maximum stress found in the blades is 78.1 MPa.

The propellers need to have a 23 mm shaft in order to spin in the duct section above the electronics compartment. This value is obtained from the CATIA model. The shaft needs to withstand the torsion during all operating conditions and the lift force. Making the thin walled assumption, equation (8.23), equation (8.22) and equation (8.24) can be used again to size the radius and thickness of the shaft. For a shaft thickness of 0.4 mm and a radius of 2.5 mm, the maximum Von Mises stress occurring during regular operation is 1.12 MPa. The estimated mass of the propeller shaft is 0.178 g.

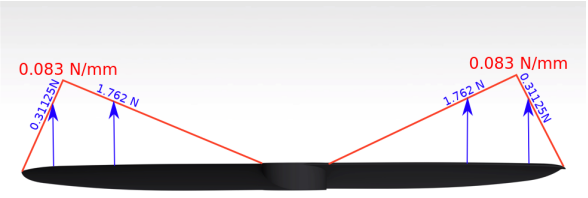


Figure 8.13: Approximated aerodynamic forces on propeller

9
POWER AND ELECTRONICS

This chapter will elaborate on the power and electronics configuration of ARTIS. First section 9.1 shows some diagrams which were used in the design. section 9.2 covers the payload and sensors used for the detection of green primer. In addition, it explains which electronic hardware will be used to perform the mission. section 9.3 elaborates on the power storage and distribution. Finally, the grounding of the drone to prevent static discharges is stated.

9.1. COMMUNICATION AND DATA HANDLING

This section aims at providing an overview of the hardware, data and communication of ARTIS within itself and its environment. This can be used in the design of the electronics subsystem. First section 9.1.1 will talk about the internal hardware and the connections between those components. These will be shown within a hardware block diagram. Section 9.1.2 is a more in depth diagram for the internal data flows and handling of said data. Lastly section 9.1.3 will show an internal/external communication diagram that will help in understanding the general operation of ARTIS.

9.1.1. HARDWARE BLOCK DIAGRAM

ARTIS consists of multiple components needing to be assembled and work together as one system. To get an overarching view of all the components a hardware block diagram is created and can be seen in figure 9.1. The indepth explanation of the components can be found in section 9.2 and section 9.3 A brief description of the components function and/or the data flow is shown within the diagram. The amount a certain component is present within the system is indicated with a small number in the corner of the block. Moreover a simplistic explanation of the structure of ARTIS is present, this is visualized as overarching dashed blocks containing multiple component. Within the diagram the components that have the grounding symbol are all connected to the aluminum with the purpose of dissipating static charge, as more thoroughly explained in section 9.4.4. The hardware within ARTIS also has dedicated software implementation which will be explained and visualized in section 10.2 and figure 10.2.

9.1.2. DATA HANDLING BLOCK DIAGRAM

Within ARTIS different types and amounts of data are handled by different components and transferred between the components, with figure 9.2 showing a general overview of this. For the data transmission, different types of interfaces are used. The data flow speed at each line is the total for all the components, e.g 1244 Mbit/s is considering both stereovision cameras. The ToF sensors will use SDA/SLC, the camera will use a 30 pin SFC which transfers raw RGB data, and the ESC will be controlled through a 2 pin JST connector. The mentioned components are chosen and described within section 9.2. These interfaces are used to transfer data from the motherboard to the components. Internal data transfer within the motherboard is integrated into the circuit board. The VPU will run an algorithm to process video feed for navigation purposes. This processed data will be passed to the CPU which in its turn will run the guidance and control algorithm for path planning and flight control to ensure a safe and efficient inspection. The

Table 9.1: Outgoing communication during inspection

Function	Type	Data amount in bits
Position (x,y,z)	float	96
Low battery charge <10%	flag	1
SD card 95% filled	flag	1
Component failure	flag	1
Safe mode Activated	flag	1
Emergency abort received	flag	1
Returning to exit	flag	1
Reached exit	flag	1
Landed	flag	1
Motors switched off	flag	1
Recording stopped	flag	1
Suspected missed section	flag	1
Total	-	107 (14 bytes)

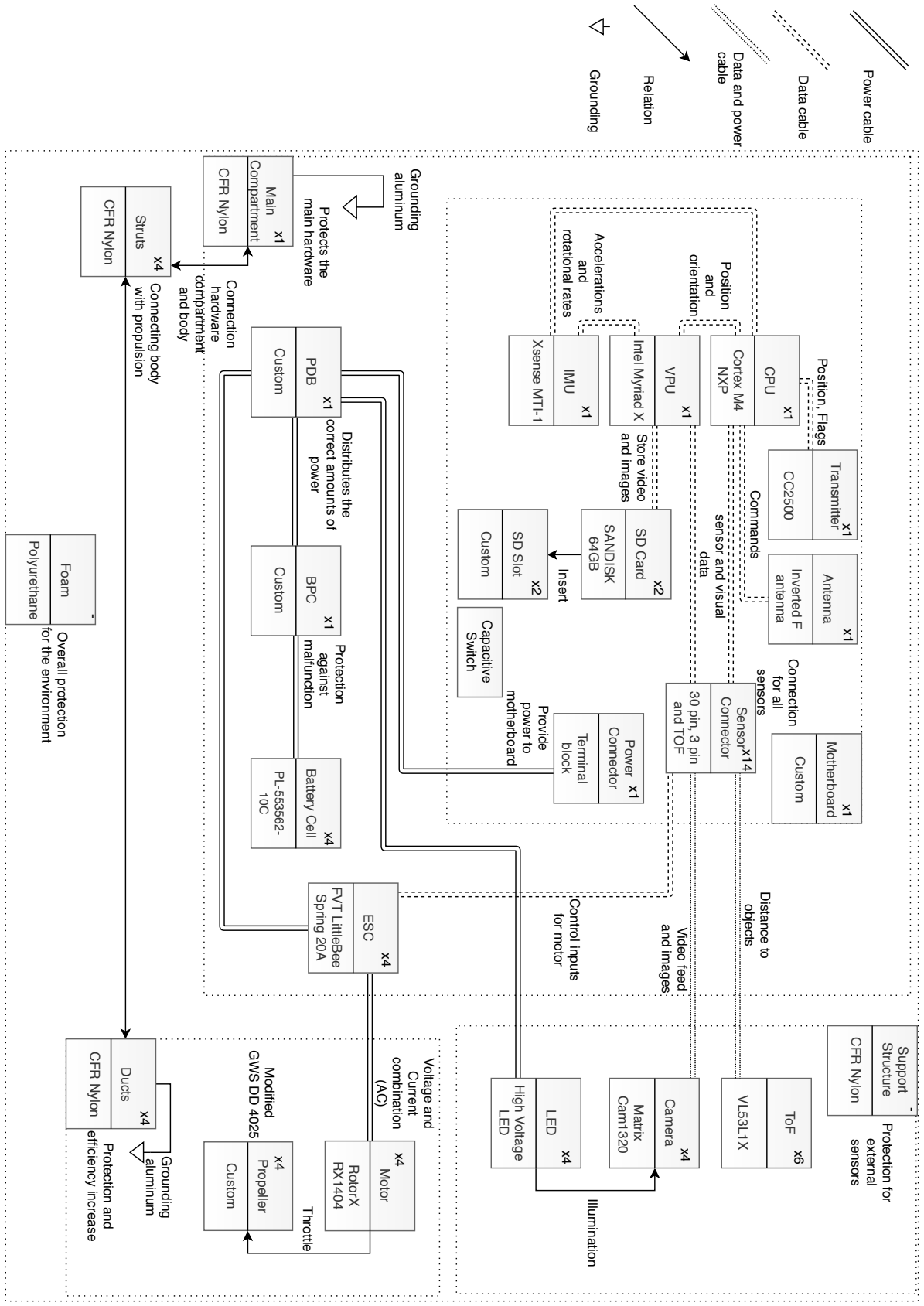


Figure 9.1: Hardware block diagram for ARTIS

video feed and images are stored onto SD cards. Only the video feed for one of the stereo vision cameras is stored due to negligible difference in FOV since the images provided by both are nearly identical. The asterisks at the "2x" connected to the camera indicate that operations at a key point are different. Namely at a key point one of the stereo vision cameras will also be used to take images. The speed at which the images are taken is reduced to 1.3 image per second, on average. This is based on the yaw speed of ARTIS at the key points. Reducing the speed at which the images are taken will not cause data overflow for the SD card to which the images are written. The SD cards will be taken out after inspection and post data processing using the green primer detection algorithm will be done by the operator. As ARTIS will communicate with the operator the information it will send is shown in table 9.1. The position is a 32 bit float number and all three coordinates are sent. All the other information is in the form of 1 bit flags. These flags will update the operator on the status of ARTIS. When a problem arises ARTIS will change the flag and proceed to solve the problem if possible. If this is not possible then an abort mission flag can be sent by the operator, which will have ARTIS try to exit the tank instead. This external communication will help locate the drone when it might get stuck or fail, decreasing the severity of risk **C1**.

For the data handling it's also important to consider that compression is an option, this would reduce data transfer rates, which is quite beneficial. Currently raw 10-bit RGB generated by the cameras, but the VPU also supports H265 and H264 compression algorithms. These can compress the data with a factor of up to 10. Currently this is not implemented in the design because of concerns for ARTIS' computational capacity and heat generation.

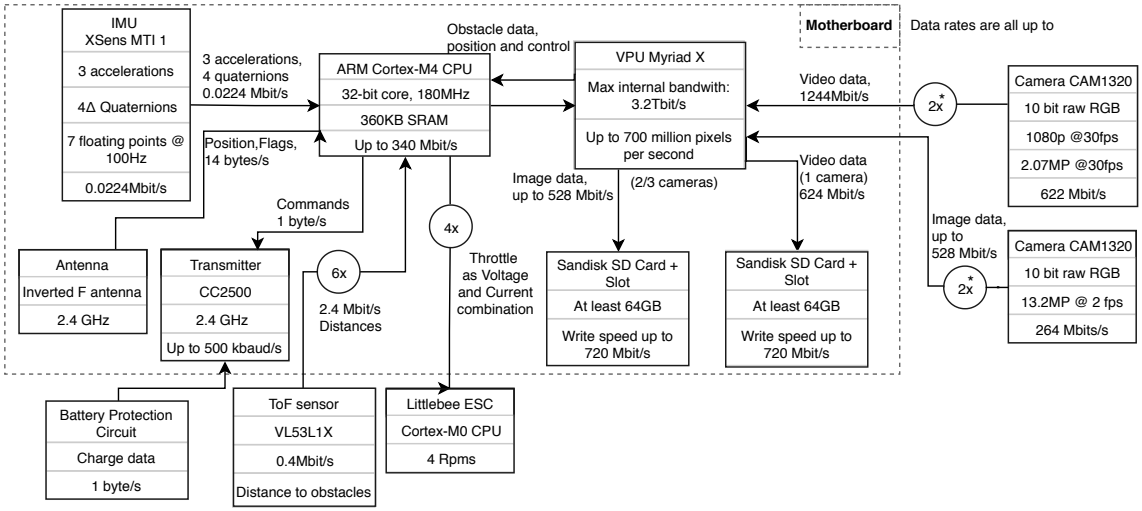


Figure 9.2: Data handling block diagram for ARTIS with * indicating nominal conditions

9.1.3. Communication Flow Diagram

After the in-depth diagram about the actual data a more general overview of the communications within the system is given in the form of a communication flow diagram. This diagram was constructed to obtain a better overview of the internal communication of the systems. With the knowledge obtained from this diagram, the electronics components can be picked better as it is now known how the internal communication is established. Within this diagram both in-going and out-going communications are included, as well as the internal communications. As can be seen the drone interacts in multiple ways with the environment. It will get visuals via the camera, as well as motion tracking and the laser will reflect off surfaces for distance measurements. Furthermore, for the external communication the drone stays continuously in contact with the off-board monitoring unit, which can be any computer with an RF module, to update the operator. After the inspection the SD cards can be taken out and inserted into an off-board computer and processed by the operator. For the internal communication a simplified version of the components and flows between those is shown. The asterisk again indicates the nominal condition, with a condition change when a key point is reached. For more details on the internal data handling see section 9.1.2.

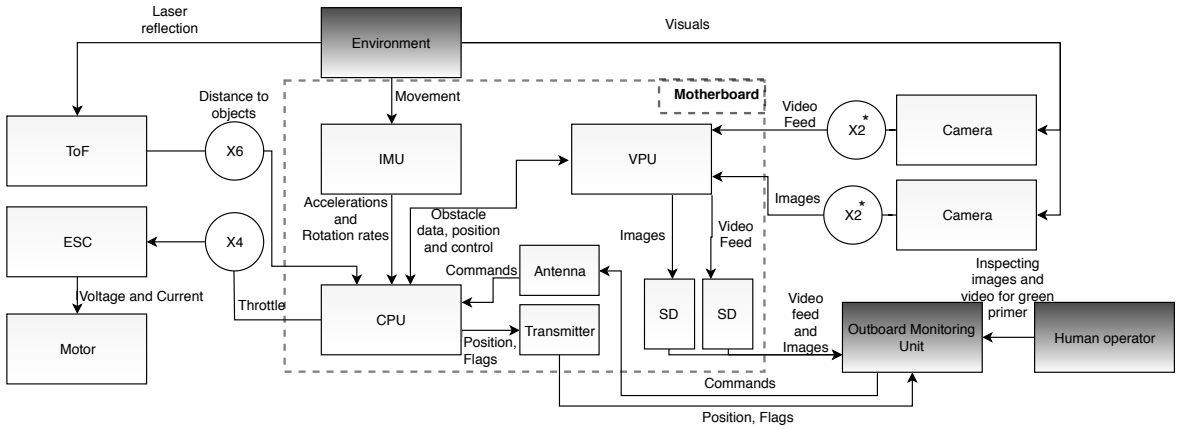


Figure 9.3: Communication flow diagram for ARTIS with * indicating nominal conditions

9.2. Electronics

This section elaborates on the electrical components inside ARTIS. This includes the payload, sensors, and other electronic components.

9.2.1. Payload and Sensors

The payload is the part of the drone that actually performs the required mission: visually inspect the tank's surface. To do this, the payload consists of cameras and LED lights. To detect anomalies, an off board algorithm to detect green primer in the tank is elaborated on. In addition, sensors that measure the distances to the environment are included.

9.2.1.1. Camera

The cameras carried by the ARTIS will perform two tasks simultaneously: performing visual inspection on the fuel tank surface, and providing the GNC algorithms with footage of the environment for proper navigation. In this part, the cameras will be analyzed for visual inspection. To select an adequate camera, one first has to look at the requirements being set for the camera:

- **AMD-F-PAYL-Cam-01 (D2)**: The camera shall provide RGB images.
- **AMD-F-PAYL-Cam-03 (Y)**: The drone camera images shall be linked to the corresponding position and orientation of the drone within the tank.
- **AMD-F-PAYL-Cam-02 (S,Y)**: The camera shall have a resolution of at least 5 MP/ft^2.

AMD-F-PAYL-Cam-03 (Y) does not pose any direct requirement on the camera itself: it merely states that if the camera spots an anomaly, the on board computer should store the position of that location. Therefore, the camera selection will be done based on **AMD-F-PAYL-Cam-01 (D2)** and **AMD-F-PAYL-Cam-02 (S,Y)**. From this it becomes apparent that the camera can be very basic: a camera module capable of making RGB images with a resolution of 5 MP/ft^2 will work. As it is important to make the cameras as light and with low power consumption as possible, only small camera modules will be considered.

Before selecting a specific suitable camera, it is first required to have more knowledge on ensuring a certain resolution, so that risk **PL1** is mitigated. Equation (9.1) was set up to compute the camera resolution in MP/ft^2, distance and observed object inclination. The inputs from the camera itself are the camera resolution, the horizontal FOV and the vertical FOV.

$$Res = \frac{MP}{tan(\frac{VFOV}{2})tan(\frac{HFOV}{2})|\mathbf{p}-\mathbf{c}|^2\frac{(\mathbf{p}-\mathbf{c})\cdot\mathbf{n}}{|\mathbf{p}-\mathbf{c}|}} \tag{9.1}$$

The denominator of this equation scales the camera resolution based on the area it observes, the distance to the object and the orientation of the object. $\mathbf{p}$ is the point that is observed, $\mathbf{c}$ the point at which the camera is located and $\mathbf{n}$ the normal vector of the observed point.

In table 9.2, the four best performing cameras for this mission have been stated, with the most important camera specifications being listed, such as mass, power usage, resolution, field of view and cost. From this table, the optimal camera for the ARTIS can be selected. The Matrix-CAM1320 has the highest resolution, a reasonably low mass and a given power usage of only 0.223 W [62]. An advantage of this camera is the confidence on the value of the power usage, since for the other cameras the power consumption is only an estimation. In addition, the FOV angles suffice for this mission and the cost is within the budget. Therefore, the Matrix-CAM1320, shown in figure 9.4, was chosen as the camera option for the final design.

Table 9.2: Camera options

	Raspberry Pi V2 [63]	**Matrix - CAM1320 [62]**	**NanoPi DUO2 [64]**	**Raspberry Pi Zero [65]**
Mass [g]	3.0	3.0[1]	2.0	3.0
Power cons. [W]	<1.0[2]	0.223	<1.0[2]	<1.0[2]
Resolution [MP]	8.0	13.2	5.0	5.0
VFOV / HFOV [°]	62.20 / 48.8	60 / 45	66 / 49.50	53.50 / 41.41
Cost [€]	29.95	21	7.99	16.95

Further aspects that are important to consider in the camera analysis are the mass of the plastic board (the blue board in figure 9.4) and the required illumination of the camera sensor to ensure smooth and bright image quality. The mass of the surrounding plastic board contributes to a large extent to the camera module mass. For the visual inspection only the camera is needed and further processing is done via the on board computer chip, therefore this board can be removed. The mass of one camera was therefore estimated at only 1 g.

The Matrix-CAM1320 requires 1.2 V and provides stable images in a temperature range of 0 °C to 60 °C. It uses an CMOS 13.2 MP image sensor with OmniBSI-3 technology, which means that the individual pixels are captured one at a time. For cameras taking images at high movement speeds, this causes the images to be blurry, since the camera moves over a distance while the camera is still capturing a single image. For ARTIS, this will not be a problem as it is expected to fly with low velocity (around 0.1 m/s). It was checked with a similar smart phone CMOS camera that taking images with the camera moving at low velocities does not affect the quality of the images. The image data output is done via a 4 lane MPI serial output which is compatible with the CPU that is described in section 9.2.2. The data is in a 10-bit RGB raw output format with 8 kbits of embedded one-time programmable memory. The low light performance is not explicitly stated, which makes the illumination source selection even more important. Using an illumination source will ensure that the cameras will never encounter low light conditions that may negatively affect the performance.

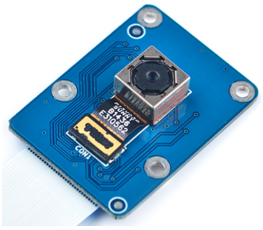


Figure 9.4: The Matrix-CAM1320

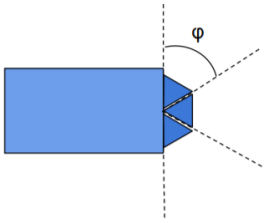


Figure 9.5: Orientation of the cameras on the drone

The most suitable camera orientation on the drone body is visualized in figure 9.5. Here, three cameras are shown: one angled upwards 60°, one facing straight forward and one angled downwards −60°, with ϕ being the FOV of one camera. In this set up, the cameras cover a full vertical 180° field of view. Furthermore, two front facing cameras will be used to provide the drone with stereo vision, useful for navigation. The separation distance between those two stereo cameras is discussed in section 10.3.1. In total, four cameras

[1] The mass of this camera is not exactly known. This value has been derived from the dimension of the camera module by comparing it to the Raspberry Pi V2 and looking at the volume of the camera module.

[2] This power consumption is not known exactly and is thus estimated to be less than 1 Watt under nominal operations.

will thus be used so that visual inspection and navigation can be performed optimally. The two front facing cameras will provide continuous video feed at 30 frames per second for navigation. The video quality is then set to 1080p. The up and down facing cameras do not have to provide continuous video and can thus take high quality 13.2 MP images. At some points in the tank, the drone will make a full yawing rotation. During this motion, one of the two front facing cameras will stop recording video and will switch to high quality images as well. In this way, the entire tank can be inspected with full 180° FOV at high resolution. How exactly this inspection will take place is stated in section 11.2.2.

9.2.1.2. Detection of green primer

The anomalies in the fuel tank will become apparent when a green primer bleeds through the coating. Therefore, crack inspection can be performed on the drone by running an algorithm that detects the specific green color of the primer. By storing this video the algorithm can be run over it, after the mission has ended. Anomalies in the tank can then be seen easily when the video and images are checked. This is not done on board since it would require significant computational power for the high resolution images.

The color detection algorithm that is used is a color detection algorithm included in Simulink. In this a reference color as well as some thresholds are set. As an input it takes the camera video feed and images and looks at the RGB values of each pixel. This RGB code is converted to the HSV color space. It then compares HSV values with the reference color and determines for each pixel if it is inside the set margin. All pixels that are within the margin are given a predefined color, in this case a very bright red for easy visualization when reviewing the inspection video. When the algorithm is run over a video feed, the same thing happens. This algorithm is described in pseudocode in algorithm 9.1. Furthermore, the stored camera footage is linked to a time stamp that can later be used to identify the location of certain finding and to meet requirement **AMD-F-PAYL-Cam-03 (Y)**.

Algorithm 9.1 Pseudocode for color detection algorithm

```
Convert input RGB pixel's color p into HSV color space as q
for H, S and V for each color do
    if the difference between the reference color and q is less than the threshold then
        Make the output color r red
        Mark the time stamp in the video at which detection has occurred
    else
        Output color equals the input color r = p
    end if
end for
Output r in RGB color space
Output list of timestamps at which detection has occurred.
```

Figure 9.6 exemplifies the results of the color detection algorithm. It has been tuned to detect the green primer bleeding through even when it is very faintly present, and to have a large range of luminosity in its detection (the V threshold in the HSV color space), ensuring a low chance of any missed detection (thus mitigates risk **C6**), but also increasing the chance of having a false positive. As a solution, after running the algorithm off-board, a human can compare the processed footage with the original footage to ensure the accuracy of the detection.

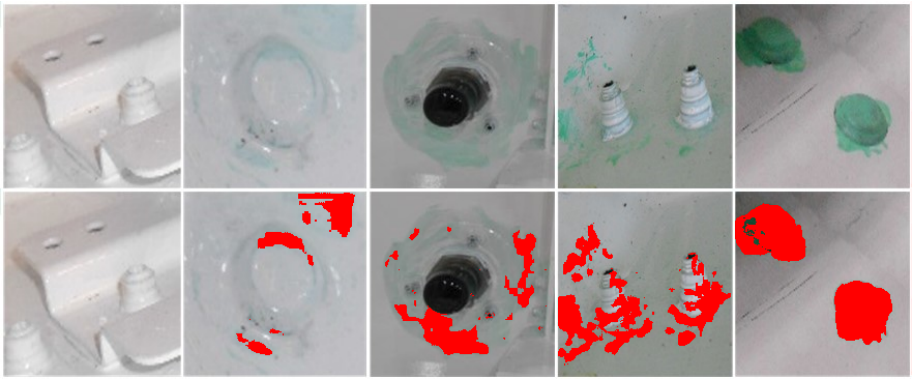


Figure 9.6: Color detection algorithm ran on provided images

9.2.1.3. Illumination source

For illumination, only one requirement needs to be met:

- **AMD-F-PAYL-Ill-01 (Y)**: The illumination source shall provide an illumination of 100 lm/m^2

To ensure sufficient illumination, equation (9.2) was composed. Here, the number of lumens of a specific LED light source is degraded with the distance to the object and the illuminated area.

$$\frac{lm}{m^2} = \frac{lm}{tan(\frac{VFOV}{2})tan(\frac{HFOV}{2})|\mathbf{p}-\mathbf{c}|^2} \tag{9.2}$$

The selection of an illumination source is based on the mass, power usage and brightness. Extensive online research was performed into LED sources, where a High Voltage LED was found to be suitable for this mission [66]. This LED only consumes 1 W of power and was estimated to have a mass of 1 g with a view angle of 120°. During drone operations it will be ensured that the illumination does not drop below 100 lm/m^2. To ensure illumination of the entire camera FOV, at least two LEDs are required. Moreover to obtain lighting from two different angles on each part of the surface, a total of four LEDs will be used. In this way, adequate illumination is ensured along the entire FOV and the surface is illuminated in a homogeneous way.

9.2.1.4. Time of Flight sensors

ARTIS will use six time of flights sensors to estimate the distance to the surrounding walls. On all sides of the drone a sensor will be placed. The VL53L1X Time of Flight Distance Sensor was chosen as the optimal option for ARTIS. It has a mass of only 0.25 g and consumes only 0.0014 W of power [67]. This low mass, in combination with the low camera and LED mass, makes sure that risk **PL2** is mitigated. Furthermore, it measures distances up to 4.0 m under the default settings. This is sufficient for its intended use. As the tank on the inside has a reflective coating, the sensors will most likely not encounter measurement problems. The way these sensors will be used for GNC will be elaborated on in chapter 10.

9.2.2. Motherboard

The on board processing of the ARTIS will take place on the motherboard. This motherboard consists of the connector board, the IMU, the CPU, the VPU, the transmitter, the antenna and the MicroSD chips with slots and a capacitive switch for switching ARTIS on and off. Furthermore, this board needs to be connected to the cameras, ESCs, ToF sensors and a power source to allow for data and power transfer. First, the hardware that is placed onto the board will be selected. After that, the specific connections and the motherboard connector board are chosen.

First the IMU was chosen, its function being the measurement of accelerations and rotational rates of ARTIS, as is explained in more detail in section 10.1.1. The chosen IMU is the Xsense MTI1 [68]. The IMU has a size of 12.1 mm × 12.1 mm, mass equal to 0.66 g and power usage of 0.044 W.

Secondly the chosen CPU is a Cortex-M4 NXP, the use of which will be explained more thoroughly in section 10.1.2. The dimensions of the CPU are 12 mm × 12 mm, mass equal to 0.5 g and power usage of 0.95 W.

Lastly, the VPU is selected. This VPU was chosen due to its low power (1 W) consumption and high computational power of 4 TFLOPs [69]. More on the usage of the VPU in ARTIS can be found in section 10.1.2. It has a size of 8.1 mm × 8.1 mm, mass equal to 0.5 g and uses 1.0 W of power.

9.2.2.1. Data Storage

The video data that the ARTIS will capture needs to be stored on board if one wants to review the footage of the inside of the tank. Using this footage, one can not only check for green primer, but also for loose cables and other anomalies. The data will be stored on a micro SD chip. One SD can store up to 90 MB/s. The upwards and downwards oriented cameras taking images at 13.2 MP at 2 fps store data continuously at 66 MB/s. For the two front facing cameras, only one needs to store the visual data. Recording video at 2.07 MP at 30 fps leads to a storage of 78 MB/s. In addition, at key points in the tank one of the two front facing cameras will switch to taking high quality images. At those points, the one camera still taking video still stores them as before. The three other cameras take images at 13.2 MP at 0.75 fps so that footage can be written with writing speed of the SD cards. Taking all of this into consideration, leads to the fact that one SD card is not enough. The data will thus have to be stored on two separate SD cards, one for the image feed of the upwards and downwards oriented cameras (and for the images of one front facing camera at key points, mentioned in section 11.2.2) and one for the video feed of one of the two front facing cameras.

With an estimated flight time of 600 s, as stated in requirement **AMD-F-P&P-Endu (S,Y)**, the total storage for the front facing camera turns out to be around 47 GB and for the up and down cameras around 40 GB. Both SD cards will thus have a storage space of 64 GB. This was also shown in the data handling block diagram, shown in figure 9.2. These two SD cards are estimated to have a total mass of 0.5 g. Besides the SD cards, ARTIS also needs slots were the SD cards can be placed. Two such slots will be used and mounted on the ARTIS motherboard, having a mass of around 2.5 g in total, based on a similar slot [70]. The chosen slot is heavier, but ARTIS does not need the entire unit. Therefore, the required part was estimated to have a lower mass. The total data storage module thus has a mass of 3.0 g. The SD cards consume negligible power. By using SD slots for on board data storage this aids in mitigating risk **A5**.

These microSD slots are integrated on the motherboard, which can be accessed through a small latch on the side of the main compartment. This latch is to be sealed with a rubber to ensure the air-tightness of the drone internals. Some alternative methods can also be considered to ensure air tightness, such as adding an IP66 (Dust and water proof) or higher certified charging/data port. This would eliminate any required rubbers, which are often hard to recycle. This could however lead to a weight increase so requires further investigation.

9.2.2.2. Communication

The final electronic components that are included in the ARTIS are an antenna and a 2.4 GHz RF transceiver. These components are added to allow for simple communication with the operator, such as commands to leave the tank during an emergency. Also, the drone can send the real time position to the outside operator so that the drone can be found in the tank at any time. Lastly it shall send flags to keep the operator up to date on the state of the drone and notify if a certain problem arises. These features mitigate the severity of risk **C1**. Together, these components have a mass of 2.0 g and have an estimated maximum power usage of 1.0 W [71].

9.2.2.3. Integration of Electronics

All of the electronics mentioned in this subsection are integrated onto the motherboard. A schematic representation, to scale, can be found in figure 9.7. For the printed circuit board including pins, the mass has been estimated to be 6.8 g, this is including a capacitive power switch, ESCs and ToF sensors. The mass was computed based on the required size to fit all components and the density of common circuit boards. The power connector to the motherboard is a PCB terminal block having a mass of 0.3 g [72]. The 4 connectors to the cameras are 30 pin FPC connectors of 0.3 g per piece [73]. Combining this with the VPU, CPU, SD slots, IMU, antenna plus a RF transceiver (0.5 g, 0.5 g, 3.0 g, 0.66 g and 2.0 g, respectively), yields a total mass for the motherboard of 12.96 g.

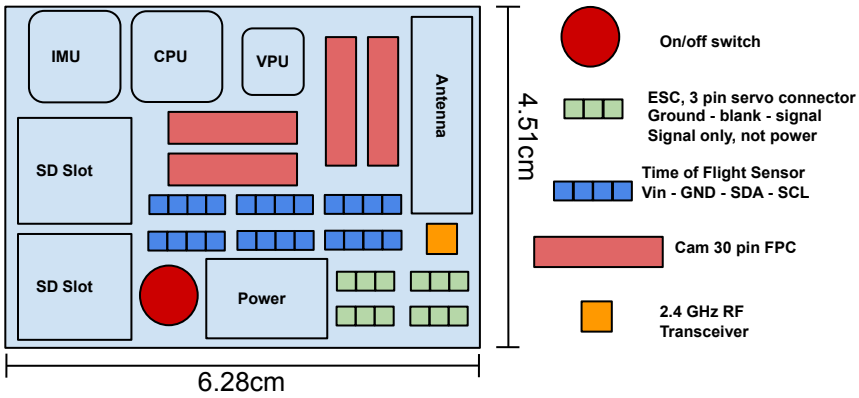


Figure 9.7: Motherboard schematics to scale

9.2.3. Electronics hardware overview

In table 9.3, all the selected electronics components can be found. In the last row, the total mass, power consumption and cost of the electronics can be seen.

Table 9.3: Overview of all ARTIS electronics components

Component	**Name**	**Mass** [g]	**Power consumption** [W]	Cost
Motherboard PCB	Custom	6.8	0.0	€1 [74]
Camera FPC connector (4x)	30 Pin FPC connector	0.3	0.0	€1.50 [73]
Power connector	PCB terminal block	0.3	0.0	€0.30 [72]
Camera (4x)	Matrix-Cam1320	1.0	0.892	€21 [75]
Time of flight sensor (6x)	VL53L1X	0.25	0.0084	€11 [76]
CPU	Cortex M4 NXP	0.5	0.95	€2 [77]
VPU	Intel Myriad X	0.5	1.0	€90 [78]
IMU	Xsense MTI-1	0.66	0.044	€17.5 [79]
LED (4x)	High Voltage LED	1.0	1.0	€0.30 [66]
Micro SD (2x)	SANDISK 64GB	0.25	0.060	€12 [80]
Micro SD slot (2x)	Custom	1.25	0.0	€2 [70]
RF Transceiver, Antenna	TI CC2500, Inverted F	2	1.0	€4 [71]
Total	-	**22.46**	**5.066**	€300

9.3. Power

This section will elaborate on the power storage and distribution inside ARTIS. The power storage will be done using batteries. This power will be distributed via cabling to the power distribution board, which in turn distributes the power to all other electronic components. The power required by the motors will be distributed via electronic speed controllers. During the design for the power provision within ARTIS the following requirements where kept in mind.

- **AMD-F-P&P-Endu (S,Y)**: The drone shall have a minimum flight time of 10 minutes.
- **AMD-NF-LFCC-Bat-05**: The drone battery shall be rechargeable in 1 h.

9.3.1. Energy Storage

The energy used to power all electronic components of ARTIS is stored in a battery. This battery delivers power via a cable to the power distribution board. In addition, a battery protection circuit is placed in between to protect the battery from over current.

9.3.1.1. BATTERY

Battery selection was performed by analyzing many different online available batteries. These were analyzed for several parameters, such as capacity, voltage, C-rating, energy density, power density, dimensions and cost. Only lithium polymer batteries were considered, since these are not only light but also most sustainable[81]. The collection of 172 different batteries can be seen in figure 9.8. In the figure, two batteries are marked in a different color; The battery with the highest energy density, being the orange square. The green star is the battery that was selected in the end as described in the next paragraph. For both batteries the lower entry is the C-Rating. The reason for displaying the battery with the highest energy density is to illustrate the fact that the energy density is not the only thing that needs to be considered, as it was in the end not selected due to the C-rating being significantly lower than desired. For discharging purposes, the battery is required to have a continuous C-rating of at least 7C. A caveat with this graph is that it also includes single cells, which are not batteries per se.

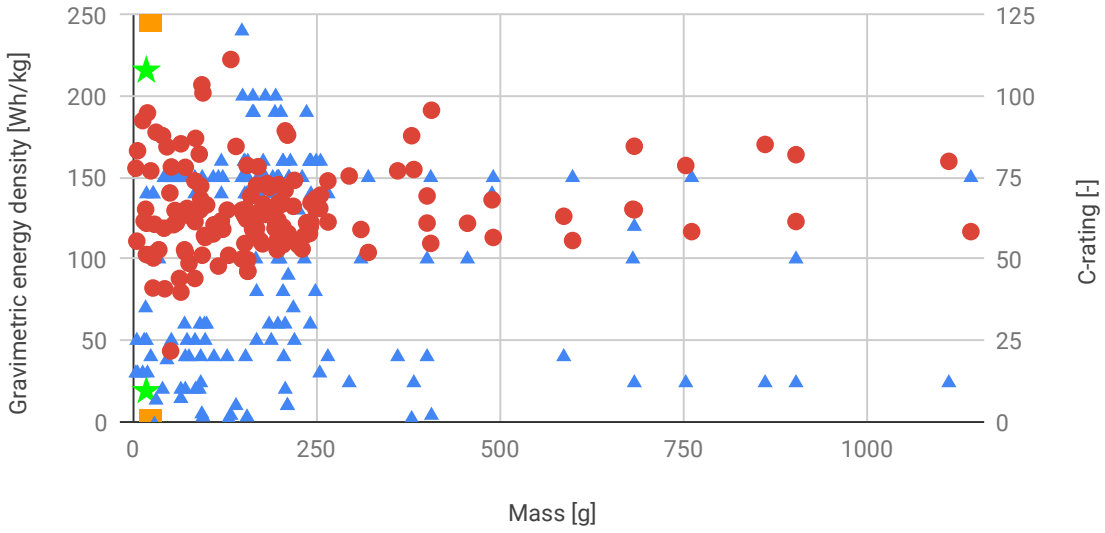


Figure 9.8: Gravimetric energy densities, battery mass and C-rating

The optimal battery selected to power the drone consists of four PL-553562-10C cells. Its specifications can be found in table 9.4. For this cell a maximum charging current is 1C, this is sufficient to charge the battery in 1 hour [8]. Due to the battery being made up of single lithium polymer cells, no protection or cabling is included. For this reason the following subsection will look into the protections required for the battery.

Four cells are chosen is because as this is required by the motors as specified in section 8.2. Furthermore having several cells in a battery allows one to convert the voltage to a constant, lower voltage, while this voltage is still fairly high. This relatively high voltage can then be converted down to the voltages required by each component. Converting a voltage down means having fewer losses and less mass needed than for converting a voltage to a higher voltage. Having a higher voltage also means fewer losses and thus less heat generated.

Table 9.4: PL-553562-10C cell characteristics [8]

Model	Capacity after 300 cycles	Voltage	C-Rating	Mass	Gravimetric Energy Density	Cost	Dimensions
PL-553562-10C	1050 mA h = 3.8 Wh	3.7 V (14.8 V for 4 cells)	10 Const	22 g	176.59 kg/Wh	€8.83	62 mm, 35 mm and 5.5 mm

9.3.1.2. BPC

A battery protection circuit disconnects the battery in any hazardous situations to prevent damage to the battery. The situations that need to be taken into account include; over and under voltage, over current and short circuit. The battery's properties are the following: Maximum discharge rate of 10 C or 10.5 A, this is over current. For voltages the values need to be considered per cell. The discharge cut-off voltage is 2.75 V, below this would be under voltage. Charge cut-off voltage is 4.2 V, above would be over voltage.

In the case of a multi-cell battery, the BPC should also serve as a charge balancer to ensure the batteries do not become unequally charged. This also means the battery is charged through the BPC. Another charge related feature included in the BPC is a charge gauge.

Looking at existing protection circuits [82], PCB-S4A8S-GS is a 2.4 g battery protection circuit for a 14.8 V battery. Its over charge, over discharge and over current ratings are slightly different from what is needed for this design, but the mass is considered to be a good first estimate. Adding a small contingency will make the BPC mass 3 g. The size is taken to be the same at 50 mm, 16 mm and 4 mm.

This BPC also significantly reduces the severity of the risk that is labeled as **E1**, due to the impact now only leading to mission failure instead of causing damage to the drone or worse. This also means **A2** is mitigated.

9.3.2. Power Distribution

This subsection describes how the power is transferred across the drone. Section 9.3.2.1 describes the ESCs, section 9.3.2.2 details the sizing of the power distribution board and section 9.3.2.3 finally shows the sizing and selection of the cables within the drone.

9.3.2.1. ESC

Power to the motor is delivered through an electronic speed controller (i.e. ESC). The required amperage and voltages are found in section 8.2 and are 17.2 A and 0.91 V during hover. For this reason an ESC that can deliver at least 20 A continuously was considered ideal. The "FVT LittleBee Spring 20A ESC" [83] can deliver this continuously and deliver up to 25 A in bursts. Including power and signal cables it has a mass of 4 g per unit. These will be placed in the centre compartment of the drone, which will then be connected with 3 wires to the motor.

9.3.2.2. PDB

The power distribution board distributes the power that it obtains from the battery through the BPC to all the other components that require power. The advantage of not including the power distribution of the motors/ESCs on the motherboard is that there is no interference due to the high Amperage. The power distribution board takes power only from the battery. The power then gets distributed to the motherboard, ESCs and LEDs. The motors obtain the required power via the ESCs. The power distribution is shown in figure 9.9. Here, the components that require power are shown as well as the required voltage and current per component. To ensure that all components operate at the proper voltage, a voltage regulator is integrated on the power distribution board, shown as a red diamond.

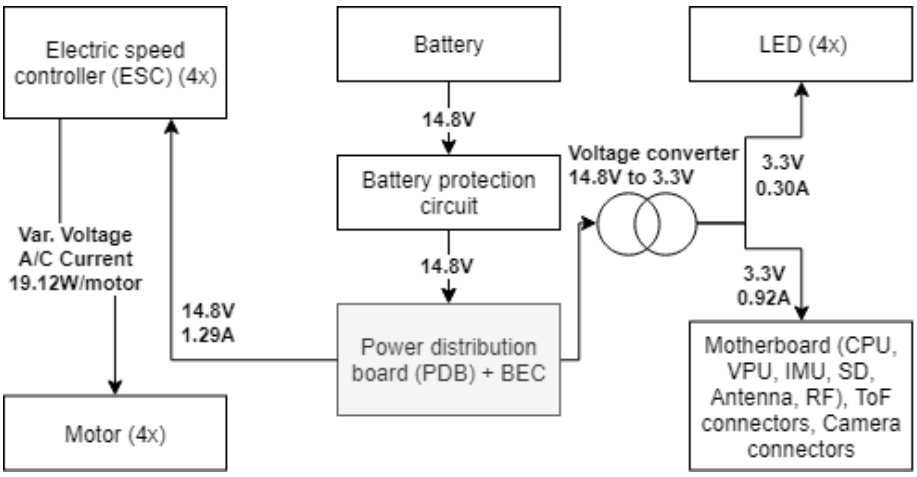


Figure 9.9: Electrical block diagram

Sizing the PDB is done by comparison to existing drones of similar size and the number of connections on the board. The connections needed are displayed in figure 9.10. Every component has a blue and a grey connection, representing the positive and negative poles. The ESCs are displayed in the four corners of the PDB and do not need a voltage regulator as essentially they are one themselves. The ToF sensors, cameras, CPU, VPU, microSD, IMU and RF transceiver are all on the motherboard and receive power from there.
In figure 9.11, a similar and commonly used PDB, being used as a reference for the design, is shown. Its mass is 9.1 g, including voltage conversion and battery eliminator circuit (BEC). It can handle up to 46 A, which is more than the battery can deliver continuously (40 A). The PDB for this drone is designed according to figure 9.9. The aforementioned BEC's function is to keep the voltage output of the battery constant. Instead of a nominal 14.8 V, down to 5 V. This means the voltage delivered to any other components remains constant regardless of battery life. The ESCs regulate their own voltage and are therefore not connected to this voltage. The PDB for ARTIS is taken to be the same size as the one in the reference image shown, the mass however is taken to be lower, at 6 g, this is due to the fact that the shown PDB has two voltage converters, while ARTIS would only need one.

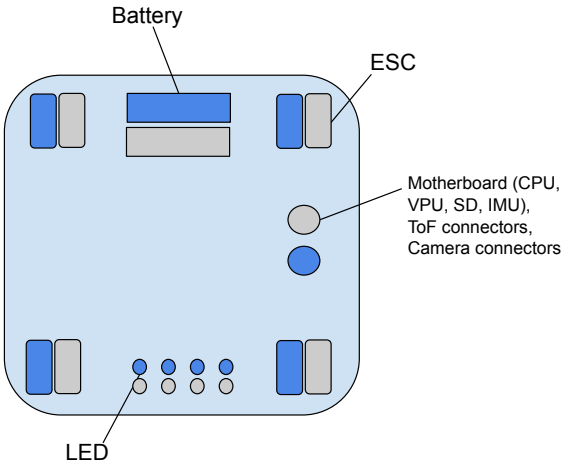


Figure 9.10: Power distribution board schematics

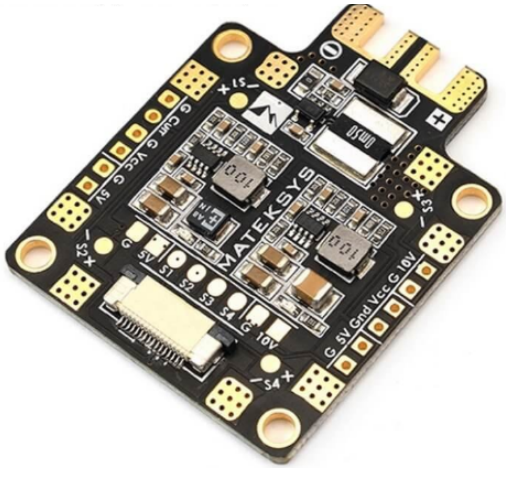


Figure 9.11: Similar PDB used for drone design [84]

9.3.2.3. CABLING

With the PDB sized, the next step will be connecting the PDB to the battery and the other components. The voltages and amperage needed are shown in figure 9.9. According to regulations and to mitigate the risk of creating sparks (risk **A3**), all cables will be ran through raceways and will thus not be exposed to the explosive environment complying with **AMD-NF-CNST-Regu-02 (D1, K)**. When deciding on cabling it is important to state the difference between chassis wiring and power distribution wiring. Power distribution wiring con-

cerns bundles of cables, since the drone has no such bundles the values for chassis wiring are used [85]. KabelTronik PTFE wires with a copper core and silver coating are then used for any wires [86]. All wires are from the same manufacturer to cut down on distribution costs and transportation emissions.

The camera is connected to the motherboard with a 30 pin FPC cable, these cables have a mass of 0.01 g/mm [87]. The ESCs are connected via a 3 pin JST connector, in which the centre pin is blank. The mass of these cables is already included in the mass of the ESC. Finally the connectors to the ToF sensors are with 4 separate pins each, two for power distribution and two for data transfer. The required amperages, American Wire Gauge (AWG), length and resulting mass for all cables are shown in table 9.5.

Table 9.5: Cable lengths and masses

Cable	**Type**	**Amps** [A]	**AWG**	**L** [mm]	g/m	**Amount of Cables**	**total** [g]
Batt-BPC	PTFE	Up to 10	20	30	8.4	2	0.48
BPC-PDB	PTFE	Up to 10	20	30	8.4	2	0.48
MB-TOF sensors	PTFE	0.0061	32	40	1.2	12	0.58
PDB-MB	PTFE	0.92	28	30	2.0	2	0.12
PDB-ESC	-	-	-	30	included in ESC	8	included in ESC
ESC-Motor	-	-	-	100	Included in Motor	4	Included in Motor
PDB-LED	PTFE	0.303	32	50	1.2	4	0.24
PDB-LED Outer	PTFE	0.303	32	150	1.2	4	0.72
MB-ESC	-	-	-	20	included in ESC	8	included in ESC
MB-camera	30 pin FPC	-	-	50	1	2	0.10
MB-camera Outer	30 pin FPC	-	-	150	1	2	0.30
MB-Lasers	PTFE (IT)	0.0061	32	40	1.2	12	0.58
						total cabling:	**3.59**

Lastly the efficiency of the cabling within ARTIS shall be considered using equation (9.4) and equation (9.3) for resistance within a wire and temperature effect for resistivity of a material [88, 89].

$$\rho = \rho_{20°C} \cdot (1 + \alpha \cdot \Delta T) \tag{9.3}$$

$$R = \frac{\rho l}{A} \tag{9.4}$$

Combining these equations with the wires chosen an efficiency approximation is performed. ρ indicates the resistivity and α is the temperature dependence coefficient. The cables consist mostly of copper with a silver coating. As silver has a higher conductivity than copper, making the assumption of using pure copper will negatively affect the results for efficiency. The characteristics of copper are $\rho_{20°C} = 1.7 \times 10^{-8}\,\Omega$m, $\alpha = 4.041 \times 10^{-3}$/K [88, 89]. Combining these characteristics with the cable lengths in table 9.5 and a temperature of 45 °C, the calculated voltage drop becomes negligible: < 1 %. For general operations the power loss due to cable efficiencies is shown to be so little that no iteration is needed for the battery as no capacity increase is required due to losses.

9.3.3. Power Hardware Overview

In table 9.6, all the selected power hardware components can be found. The total mass and power consumption are included in the last row. The BPC ad PDB cost values are based on commercially available ones, but within ARTIS custom ones will be implemented.

Table 9.6: Overview of all ARTIS power components

Component	Name	Mass [g]	Power consumption [W]	Cost [€]
Battery cell (4x)	PL-553562-10C	22	0.0	8.83 [8]
BPC	Custom	3.0	0.0	8.43 [90]
ESC (4x)	FVT LittleBee Spring 20A	4.0	0.0	11.29 [83]
PDB	Custom	6.0	0.0	7.95 [91]
Cabling	PTFL Type Cabling	3.5	0.0	15.55 [92]
Total	-	**121.77**	**0.0**	**112.41**

9.4. GROUNDING

This section will describe the proposed solution for grounding the drone and preventing electrostatic discharges. The final solution determined in this section might not fit the definition of 'grounding' per se, but it is referred to as grounding since this fits what is aimed to be accomplished in this section, namely to prevent the fuel tank from exploding.

No other drone has been designed to inspect fuel tanks in a dangerous environment like ARTIS has. As described in chapter 4, this means a large untouched market is available. Exploring this market however requires the application of creative solutions to mitigate the many risks involved with operating in such an environment. The grounding subsystem in its essence serves to ensure the requirement labeled as **AMD-NF-CNST-Sfty-02** is met, along with the mitigation of accompanying risks **A2** and **A3**. They are stated below.

- **AMD-NF-CNST-Sfty-02 (S,D1):** The drone shall not create sparks during inspection.
- **A2.** Explosion induced by system malfunctioning
- **A3.** Induced sparks during operation

9.4.1. BACKGROUND INFORMATION

Electrostatic charge is generated through the triboelectric effect, i.e. the transfer of electrons due to the termination of contact between two materials [26]. In the case of ARTIS this effect occurs mainly due to the friction between the air and the rotors. This leads to a potential difference between the tank and the drone, which in turn leads to the generation of sparks, if this difference is sufficiently high.

The energy of a spark can be approximated with equation (9.5) [93], in which E is the spark energy, V is the spark voltage, and C is the capacitance of the capacitor that is formed between ARTIS and the nearby object. If this energy exceeds approximately 2 mJ (at 45 °C), the vapors will ignite [94].

$$E = 0.5CV^2 \tag{9.5}$$

Further relevant information on this problem includes the fact the coating on the inside of the fuel tank is not conductive [14]. This means any solutions using the tank to bleed charge are reduced in effectiveness.

The fuel tank also contains the anti-static agent or conductivity improver known as Stadis450 [95]. Stadis450 has an evaporation rate of 1.4 butyl acetate, which is a medium evaporation rate [96]. This is the same rate as ethyl alcohol and almost 5 times as fast as water. This means that Stadis450 is also present within the vapors in the tank. This will likely have a positive effect on the charge of the drone since minuscule particles in the air can take away static charge [97].

Relative humidity also has a very low effect on the permittivity constant which has a direct effect on capacitance [98]. This effect however amounts, at most, to a less than one percent increase, and can thus be considered negligible.

9.4.2. Possible solutions

In figure 9.12, a design option tree for the solutions to the grounding of ARTIS can be found. This subsection will elaborate upon all branches of this tree to find the most suitable solution.

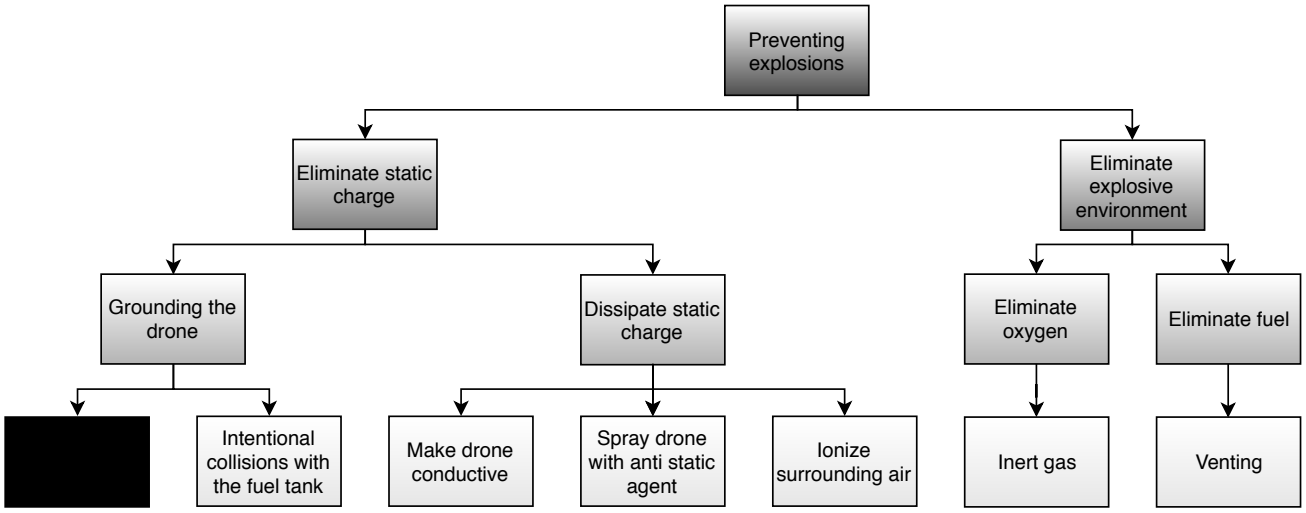


Figure 9.12: Explosion prevention design option tree

9.4.2.1. Main grounding solution

To ensure no static discharges will occur during operation, a grounding solution was designed that would likely mitigate this risk. After discussion with the client, Lockheed Martin, it was decided to leave out this solution from this report as they are investigating possible patenting of this idea. For this reason, some parts of this report are marked black, such as in figure 9.12 and figure 12.27. This indicates that the main grounding solution is described here, but left unreadable as requested by the client.

9.4.2.2. Intentional Collisions with the Fuel Tank

A second option for grounding the drone is via intentional collisions with the fuel tank. If the drone were to touch the fuel tank structure at regular time intervals, any build up of static charge would dissipate into the structure. This however comes with numerous issues:

- As the tank structure is coated on the inside, any static charge that is build up in the drone might not dissipate quick enough during a short term collision;
- As it is hard to know how rapidly the charge builds up exactly on the drone, estimating the time interval between two successive collisions becomes difficult;
- The GNC computations of the drone become computationally intensive as the drone needs to stabilize itself more often;
- The inspection time of the drone would increase significantly as it has to diverge from the optimal path to bump into the fuel tank;
- At some points it might happen that no fuel tank structure is close enough to the drone to reach within the desired time, leading to more static charge than anticipated and allowed.

These issues show why the method of intentional collisions is not considered a suitable option for ARTIS.

9.4.2.3. Make ARTIS Conductive

This solution means making the entirety of ARTIS' externals conductive, either through a change in the materials used, or through the application of an anti-static agent to the externals of the drone. This would help against the buildup of static charge by dissipating some charge due to conducting it and also distributing the charge over the entire drone. This means there are no built up charge concentrations.

One could assume that the majority of the charge is generated at the propeller and ducts. This is because these two components operate at the highest speeds and thus experience most friction leading to an increase

in the triboelectric effect. With this assumption it is advantageous to make these components conductive so that their charge can be dissipated better. The propellers could be made of a conductive material, likely carbon black. The ducts, likely made of aluminum, will also be grounded to that same circuit.

A suitable anti-static agent could be Stadis 450, which is also used in the JP8 used for the C-130[99]. Liquid anti-static agents do not affect the mechanical properties of polymers [100].

The problem with using Stadis 450 would however be that Stadis 450 is highly toxic [96]. This would however eliminate an advantage of ARTIS which is the elimination of any health hazards for tank inspection personnel. These hazards could be mitigated through masks similar to those used for the personnel entering the tank for inspection.

9.4.2.4. Ionizing the Air

The following solution requires an external device ionizing the air, giving the gasses in the air a charge. This will allow the drone to bleed off charge into the air, effectively eliminating any charge[101]. This is however not considered to be an effective solution since it would require the air being blown at the surface in which the charge needs to be eliminated [102]. It is logistically not possible to blow air at ARTIS from all points. Furthermore this would introduce turbulence due to the blowing of air into the tank, harming the performance of ARTIS.

9.4.2.5. Eliminating the Possibility of Explosion

If there is no way for an explosion to occur within the operation environment no method to eliminate static discharges is required. This is shown on the right side of figure 9.12 as eliminating the explosive environment. This can be done most intuitively by removing the fuel from the environment by venting as is done currently in inspections done by humans. This would however remove part of the time saving advantage ARTIS gives and is considered less than ideal.

A solution that is perhaps more interesting is eliminating the oxygen in the tank to prevent any ignition from occurring. Most gasses require at least 10% oxygen to be present for ignition to occur [93]. A suitable inert gas to vent the tank with would be nitrogen. Venting the tank with nitrogen would increase ratio of nitrogen to oxygen, making a gas mixture that can not explode under operational temperatures. This venting would take shorter than venting all the fuel out, because by using nitrogen not all oxygen has to be vented out. One consideration is the cost of nitrogen gas. On average, N_2 costs 5 cent per m^3 [103]. The fuel tank volume of the C-130 is equal to 25.36 m^3 [104]. This leads to a cost of around €1.27 to fill all tanks with nitrogen. The nitrogen cost itself can thus be considered negligible.

The downsides of eliminating the possibility of an explosion can be summarized as follows:

- Both removing fuel and lowering the oxygen in the tank lead to an increased time of venting. This partially defeats the purpose of ARTIS;
- Venting the fuel out does not lead to additional costs as Lockheed Martin currently uses this method. Venting the tank with nitrogen does lead to additional costs as Lockheed Martin needs to buy, transport, and store the nitrogen. Also, the current venting machine needs to be modified such that it can take in nitrogen to vent.
- More personnel is required during drone operations to connect the venting machine. This defeats the full independence of the drone.

In conclusion, venting the tanks is not considered optimal for the mission. Venting the tank takes time, adds additional costs and makes the drone less autonomous. This method will thus not be further investigated, but if it later turns out that no other methods work, this method will be kept in mind.

9.4.3. Selected (combination of) solution(s)

For ARTIS' mission the most suitable solution was found to be the conductive propellers and ducts. This solution is lightweight, cheap and easy to implement into the design. This solution also ensures mitigation of risk **A3**.

This solution will still be tested if it works as intended. If these tests show that the conductive ducts and propellers, even with the addition of the redacted solution, do not ground the drone well enough, an anti static agent will be sprayed on the drone to make it fully conductive. In combination with the redacted solution, this will ensure complete charge dissipation. This anti static agent also has low costs. It however does come with health hazards[96], which is why it is not implemented directly.

To check if the proposed solution works as intended, multiple tests will have to be performed to verify the solution. These tests will show whether or not the conductive parts by themselves are enough to prevent explosion in the tank due to electrostatic discharges. These tests will also check if any charge builds up outside of the conductive components. It will then also show whether or not an application of the anti static agent spray is required. These tests will be elaborated on in chapter 12, section 12.5. Some of these tests also serve to increase the understanding of the charge buildup in drones. Along with the tests, recommended actions for the results of the tests are described too.

9.4.4. Implementation of selected solution

In reality more than just the ducts need to be made conductive, this is because the main compartment and the struts connecting the main compartment to the motors are also directly in the wake of the propellers. This means that according to the assumption that only parts in direct contact with the flow are generating charge. They are thus also to be made conductive according to the solution selected in the previous section. This is done by covering these exposed components in aluminum foil, which can then dissipate any charge being generated on the drone. The mass required for this 3.63 g.

10
AUTONOMOUS FLIGHT

This chapter aims to show the building blocks towards providing ARTIS with autonomous flight capabilities. First, the required hardware and their functionality is addressed. Then, the relation between guidance, navigation and control is displayed via a flowchart. Lastly, the Control, Guidance and Navigation are discussed, respectively.

10.1. REQUIRED GNC HARDWARE

In order to allow for autonomous flight, ARTIS is equipped with the following hardware. The functionality is touched upon, while the mass, size and other specifications of the hardware can be found in section 9.2.2.

10.1.1. CAMERA, TIME OF FLIGHT SENSOR AND IMU

As follows from section 9.2.1.1 and section 9.2.1.3, ARTIS' front face is equipped with two CAM1320 cameras with accompanying LEDs. This will allow for stereo-vision, facilitating computer vision algorithms.

For more accurate navigation, all six faces of ARTIS are equipped with a ToF sensor to measure the distance to the surrounding walls. It can measure distances up to 4.0 m under the default settings, which is sufficient for the navigation. Having a ToF sensor especially helps the two front cameras with hole detection. Namely, the front cameras alone will not be able to detect enough depth features to distinguish a hole from a wall. The distance measurement will provide the knowledge to distinguish both.

Lastly, the drone contains an IMU, namely the XSens MTI-1. The IMU contains accelerometers for linear acceleration measurements and a gyroscope for angular rate measurements. All the measurements are done in the body frame. This state of the art device will run at 1000 Hz for angular rates and at 100 Hz for linear acceleration rates. The response for angular rates is faster, which is desired as small offsets in angles might impose serious risk. Other MTI versions (2,3 and 7) were considered too. Their added benefit is the capability of measuring free acceleration, heading tracking, referenced yaw, and GNSS receiver support. These measurements are not required for ARTIS' operations, so MTI-1 was deemed most suitable.

10.1.2. CPU AND VPU

For computational power, ARTIS is equipped with a CPU and VPU. For the CPU, a 32-bit ARM Cortex-M4 microcontroller was selected. It was chosen as it contains a FPU which handles IEEE-754 floating points particularly well [105]. To allow for more visual processing power, Intel's Movidius Myriad X was placed amongst the CPU. This cutting edge VPU supports a 720p stereo pair, and runs 4 trillion operations per second, at the cost of extremely low power usage [69]. This makes it very suitable for techniques like VO.

10.1.3. PAYLOAD AND SUPPORTING SYSTEMS

For the payload, four Matrix - CAM1320 RGB cameras are used. Two of them are simultaneously used for navigation. These provide enough resolution for the navigation and comply with **AMD-F-PAYL-Cam-02 (S,Y)**, both is under the condition that it flies with the speed associated with the calculated inspection time and correct distances. Its specification are stated in section 9.2.1.1.

10.2. SOFTWARE DIAGRAM AND GN&C INTERFACES

Before building the GN&C algorithms, it can be helpful to construct an overview of all the pieces of software that have to be written as well as of the interfaces between these pieces of software. A useful tool for visualizing this overview is the software diagram, which for ARTIS is shown in figure 10.1 [106]. Figure 10.1 will be used to develop the software for ARTIS itself and the software interfaces.

More specifically, to create a better insight in what will be elaborated on in this chapter a general overview is created for GNC. The relation between the navigation, guidance, and control is displayed in figure 10.2.

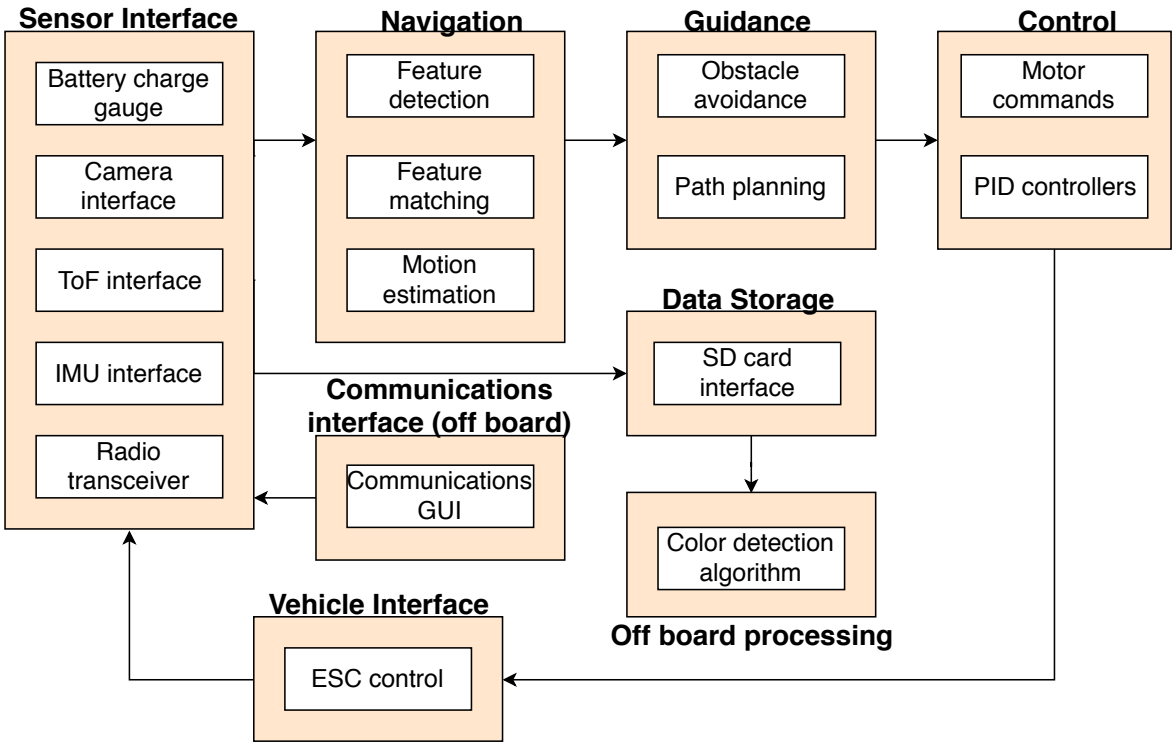


Figure 10.1: Software block diagram for ARTIS

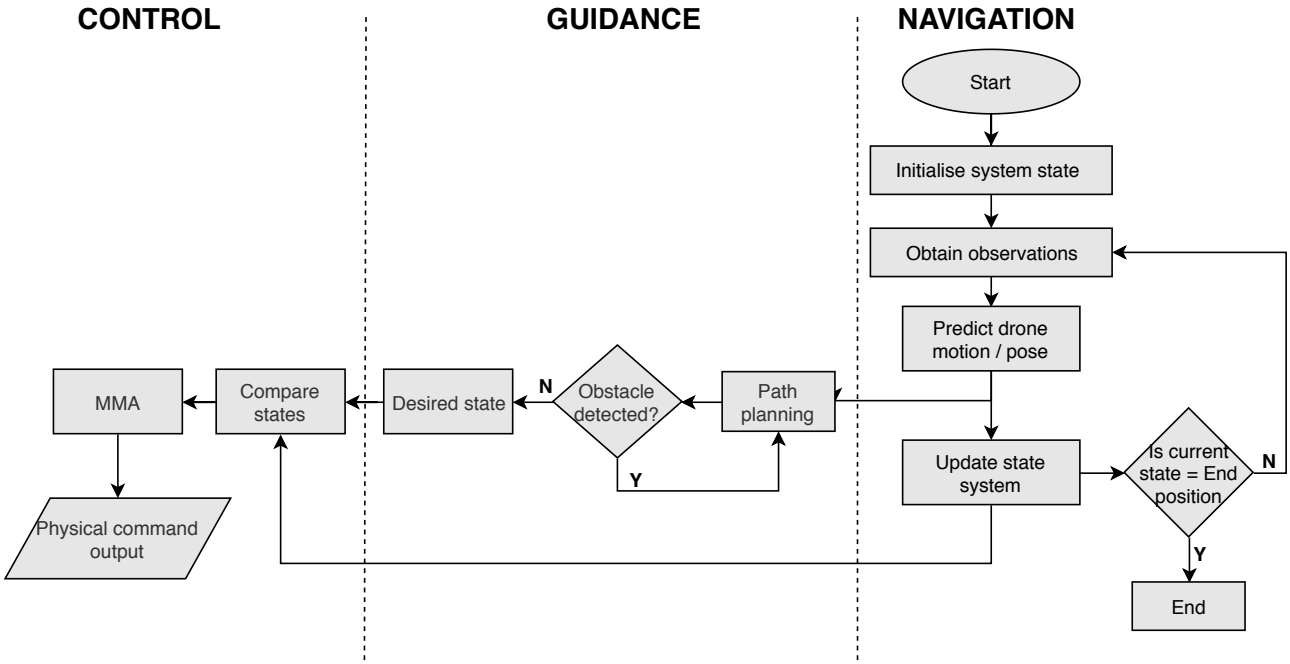


Figure 10.2: GNC framework flow chart

As evident from the chart, navigation is concerned with feature matching within the tank and as such determining position and motion, guidance focuses on path planning and obstacle detection and avoidance, and control is responsible for controlling the motion, position, and attitude of ARTIS as instructed by guidance and navigation.

10.3. NAVIGATION

In this section the principles and methodology used for the autonomous navigation of ARTIS are discussed and explained. Navigation entails the estimation of vehicle position and motion. This localization problem shall be approached using sensor data from multiple sources, namely from cameras, an IMU and ToF sensors.

Autonomous navigation is usually approached through two fundamental principles, VO and SLAM [28]. While VO simply determines the vehicles position and motion SLAM attempts to simultaneously map its environment. For ARTIS the use of SLAM appears to be inappropriate given the repetitive nature of aircraft fuel tanks and the on-board capacities of ARTIS. The fuel tanks features are repetitive and not distinct, rendering a map of this environment thus raises various problems regarding ambiguity and re-detection of features. The environment inside a fuel tank is explained in more detail in section 13.2.1. Furthermore, the storage and processing of the mapping feature requires additional on-board processing capacities and storage. Offering no clear advantage, the decision has been to use VO as opposed to SLAM. Since SLAM is an additional component for VO, if it is found that its implementation is profitable it could be implemented in future design versions or upgrades.

Regarding VO methods, two main procedural distinctions are the use of indirect or direct methods. Indirect methods implement an image pre-processing step before motion estimation, while direct methods skip this process and go directly from the image to motion estimation by optimizing photometric error [107]. Indirect methods commonly depend on feature detection and matching. The main advantage of direct methods is that it can use and reconstruct the whole image input as well as make decisions on more complete information, reducing the vehicle position error. Nevertheless, as a consequence the methods are slower and at their current stage of development require higher processing capacities [27]. Furthermore, given the artificial lighting conditions inside the fuel tank, in which the light source moves with the drone, major concerns arise regarding the robustness and flexibility of direct methods which rely on photometric pixel intensity. Due to this phenomenon, subsequent image scenes may not have similar photomoteric distributions such as to obtain a correlation for motion estimation. Given these consequences, a indirect VO methods shall instead be pursued.

10.3.1. ALGORITHM DESCRIPTION

The available sensors on-board ARTIS have already been discussed and have been selected with possible navigation applications in mind. Stereo cameras simultaneously observe features on two frames that are spatially separated by a known distance. Stereo cameras were chosen in order to reduce the steps required to reconstruct 3D information from two to one (compared to monocular camera) as well as providing scale invariance, the cameras shall be separated by 75 mm. The implementation of an IMU support the adjustment of position estimates, especially at moments in which the camera movement is fast, thus reducing the efficiency of feature detection and matching. Finally, ToF sensors can be used to support the position estimation especially with regards to vertical positioning within the fuel tank. Furthermore, a forward facing sensor can help determine if a detected shape is covered or hollow. A flow chart depicting the logic of the methods can be seen in figure 10.3. The computer vision algorithms shall run fully on the on board VPU.

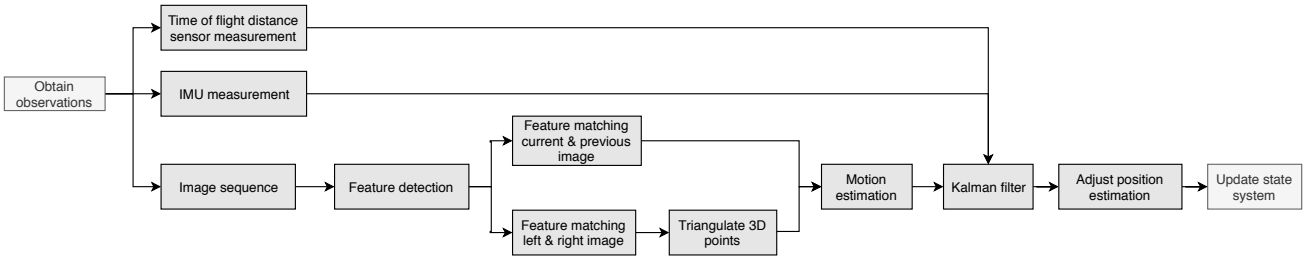
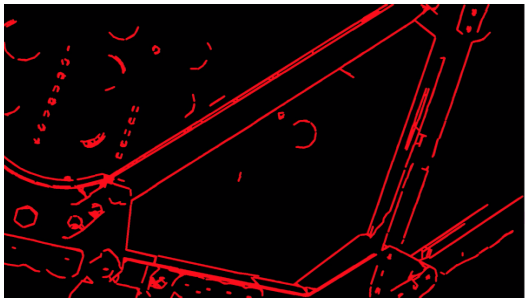


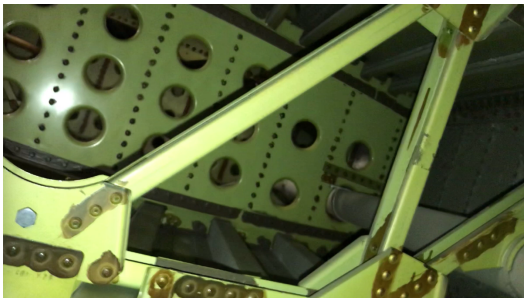
Figure 10.3: Navigation method flow chart logic

Looking at figure 10.3, each procedure within the localization method shall be discussed separately. The discussion shall start from the point at which the image sequence (i.e left and right camera images) have been obtained and made available. The first step given these images is to detect their features. Here it is advantageous that the variety of features that can be found inside the fuel tank are mostly know, this allows for a custom choice of feature detectors which can be optimized for the environment. To this end, lines and circles or ellipses were chosen as descriptors for the environment. Most structural items within the fuel tank can be described using lines (e.g. stringers, rib struts and tubes) while circles aid in describing rib cut-outs and exit holes.

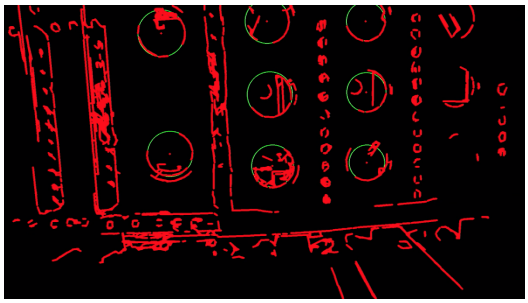
Images must be pre-processed before the feature detection methods are applied. These methods involve the conversion of the RGB images to gray scale and the blurring of the image using a Gaussian function. The blur reduces the noise in the image and improves the performance of the detection methods. Having formatted the images appropriately, a canny edge detector can be used to detect edges in the images. The edges can then be used to detect lines and circles using Hough Transforms. This method leads to the results shown in figure 10.4.



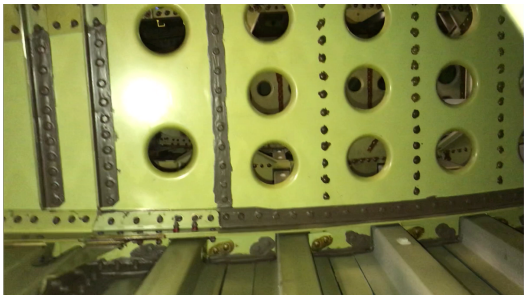
(a) Line detection from input image



(b) Fuel tank truss input image



(c) Line and circle detection from input image



(d) Fuel tank rib structure input image

Figure 10.4: Line and circle feature detection

The images shown in figure 10.4b and figure 10.4d are snapshots from a video taken inside a Dakota aircraft fuel tank, while figure 10.4a and figure 10.4c are from the corresponding video with the application of the discussed feature detection algorithm. The algorithm was tested using videos with different lighting scenarios, one using direct artificial light that moved along with the camera (the one shown in figure 10.4), one with no artificial lighting and one with indirect artificial lighting that did not move along with the camera. The first

case was shown to be the most advantageous for increasing the number and accuracy of detected features, it is also the way in which lighting is implemented in ARTIS. The line detection algorithm discussed ran real time with video recordings at 30 fps on a laptop equipped with a Intel Core M-5Y51 CPU (2 cores) at 1.10 GHz.

Having detected the features on each frame from the stereo input, the features have to be matched between left and right images as well as from one of the current image frames to one of the previous image frames. Feature matching thus refers to the process of finding out what feature is the same in both the images that are being compared. This process will then allow for the tracking of features and subsequently the triangulation of the observed features in 3D space. Figure 10.5 visually depicts the matching process between frames. Given that the feature detection algorithms shall detect lines and circles, these features are to be taken advantage of when performing feature matching. Various solutions for line feature matching have been proposed, tested and used [108–110], thus line feature matching is a feasible and reliable choice for feature matching, these methods should also match circles or ellipses as these shapes shall also be partially or fully constructed with line segments.

Having matched features between left and right stereo images, the 3D position of the camera with respect to the matched features can be triangulated. The observation of a point via the two images is illustrated in figure 10.6. Once common line features have been established, perspective-n-point algorithms (PnP) can be used in order triangulate the 3D pose of the cameras with respect to the observations. The minimum number of points used in PnP algorithms is 3 [111]. In this case, since line features are being used, the problem is posed as a perspective-n-line (PnL) algorithm. Furthermore, compared to PnP, PnL is more suitable for texture-less indoor environments. PnL has been thoroughly researched, and the methods robustness and accuracy has progressed in recent years [112]. Since these algorithms are dependent on a sample of points, they are vulnerable to errors due to outliers. In order to mitigate this effect and make the triangulation more robust, RANSAC can be integrated with PnL. The integration of RANSAC with PnP algorithms has been researched, is commonly used and available on open source platforms [113]. Depending on the specific PnL algorithm used, the complexity can range between $\mathcal{O}(n)$ and $\mathcal{O}(n^2)$ [114].

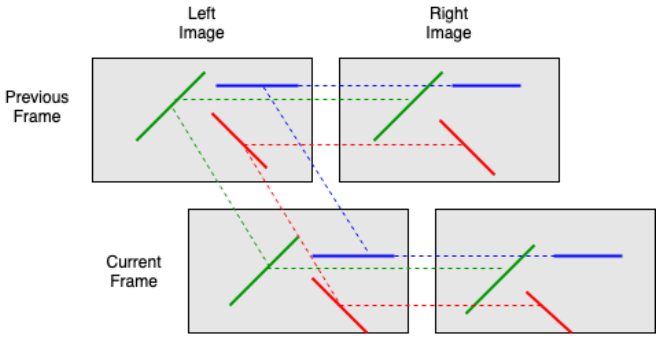


Figure 10.5: Stereo feature matching

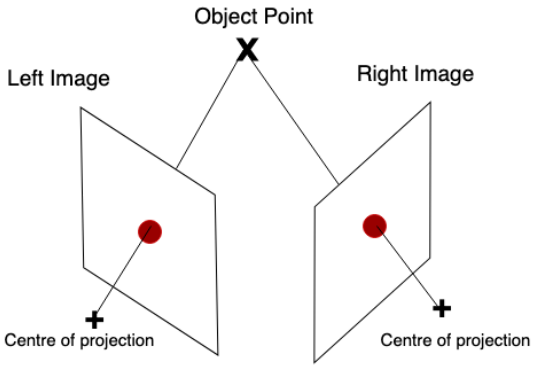


Figure 10.6: Stereo feature observation

Having triangulated the observed features from both the current and previous stereo pairs, the incremental camera poser can be recovered using the tracked features and triangulated points. This problem is commonly solved using absolute orientation methods [111]. Absolute orientation obtains the relative camera transformation from the coordinates of stereo-matched and triangulated features as represented in figure 10.7. This transformation is performed through geometric operations. Similarly to the triangulation algorithms, this methods will have associated errors resulting both from stereo and temporal feature matching. Once again, this can be mitigated by the use of RANSAC.

To this point, the navigation framework has been discussed to the extent that localization can be performed autonomously. Nevertheless, in order to improve the framework and mitigate navigation related risks, the framework is further optimized. This is done by the integration of the IMU and ToF sensors.

IMUs can be integrated in order to obtain position and orientation information in a short time step since they suffer from drift over longer time scales. They are therefore ideal to use when the standard navigation algorithm or cameras are temporarily unable to provide satisfactory outputs. This can occur for example in the case that images are blurred or distorted and when the movement of the camera is fast and sudden. This feature improves the overall performance of the navigation framework as well as mitigating risk **C4** (chapter 6) by lowering it's likelihood of occurrence. The process of using IMU measurements to obtain orientation and position information is called dead-reckoning [2]. The accelerometer and gyroscope measurements obtained from the IMU are integrated to position and orientation as shown in figure 10.8.

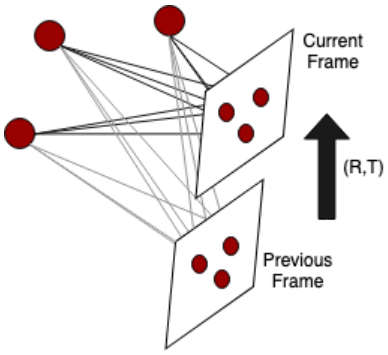


Figure 10.7: Depiction of time wise motion estimation

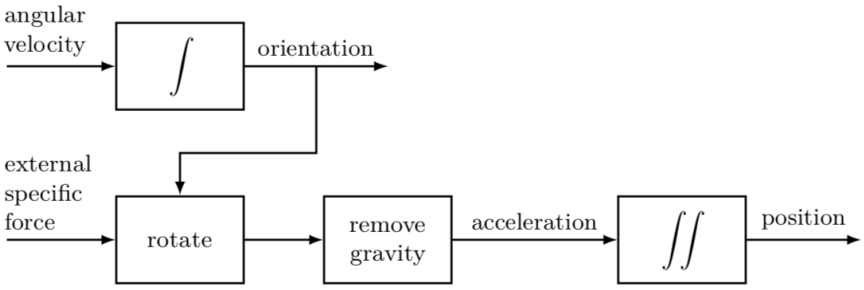


Figure 10.8: Dead reckoning with IMU [2]

The estimations resulting from the IMU measurements can then be integrated with the estimations from the visual odometry procedures using a Kalman filter which minimizes the error between two variables based on the mean of the squared error. The use of a kalman filter is common practice in visual inertial odometry algorithms [115].

The ToF sensors further improve the navigation framework by providing information on the distance to surrounding features. An ToF sensor is placed on all six planes of ARTIS. Like the IMU, the measurements from the ToF sensors are integrated after the position estimate from image sequences. Two major advantages of these additional sensors are the collision prevention and feature depth determination possibilities. By providing additional measurements on the distance to a certain object, the ToF sensors can be used to warn ARTIS when it is too close to an obstacle. This is especially useful for top and bottom collisions since ARTIS only has forward vision and thus no vision in these planes. By using a forward ranging ToF sensor, it can also be used to determine the depth of a certain observed feature to determine if a detected shape is hollow or covered, and thus if ARTIS can or can not travel trough it. Furthermore, these features are considered to be risk mitigating factors, specifically mitigating the likelihood of risk **C5** (chapter 6).

The use of ToF sensors provide additional risk mitigation characteristics. In the case of loss of position knowledge, the downwards pointing ToF sensor can be used to assist a controlled landing by providing the distance to the fuel tanks bottom panel, thus mitigating the severity of risk **C4** by preventing violent impacts with the fuel tank. Finally, by processing the ToF sensors measurements in the CPU, separately from the other navigation algorithms running in the VPU, the same controlled landing mitigation feature can be applied to the case in which the VPU stops function nominally, thus mitigating the severity of risk **C3**.

A final, important consideration for the integration of all sensors is their calibration. The cameras, IMU and ToF sensors must be appropriately calibrated before operations in order to obtain an accurate and reliable navigation algorithm. Appropriate calibration also helps mitigate risk **C4**. The sensors should also be periodically checked and re-calibrated if necessary. A procedural summary of the navigation algorithm structure discussed in this section can be seen in figure 10.9.

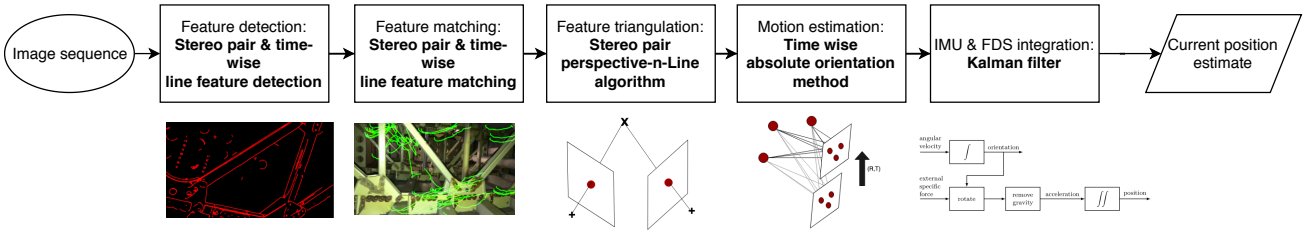


Figure 10.9: Overview of navigation algorithms

10.3.2. Navigation velocity

When testing the feature detection methods, it was clear that the movement of the camera influenced the quality of the output. As previously discussed, the IMU is used in order to correct for such periods where the image quality was poor. Nevertheless, these effects should be minimized in order for navigation to perform adequately. To this end the maximum velocity that ensures sufficient camera quality must be determined. The velocity can be estimated by determining how fast the camera can move in order to meet the shutter speed value required for a stable image. This shutter speed can be calculated from the cameras frames per second value. Equation (10.1) shows the linear relationship used to obtain the maximum transverse velocity for navigation. In the equation, $Hres$ represents the horizontal pixel resolution, $npixel$ the number of shifted pixels during exposure time, $HFOV$ is the horizontal field of view, FPS is the camera frames per second and d is the distance to the recorded surface. All components in the equation are constants except for d and $npixel$.

$$v = \frac{\frac{npixels}{Hres} \cdot 2 \cdot tan\left(\frac{HFOV}{2}\right) \cdot d}{\frac{1}{2 \cdot FPS}} \tag{10.1}$$

Table 10.1 lists the velocities, pixel blur and camera displacements during exposure time for navigation given the distance to the recorded surface. The values shown correspond to the use on $npixel = 3$ since this was the limiting case for acceptable blur. The distance is limited at 40 cm as discussed in section 9.2.1.1 due to resolution constraints. The pixel blur is calculated using equation (10.2), in most cases, a pixel blur significantly higher than 1 may cause issues [116].

$$B = \frac{v \cdot \frac{1}{FPS} \cdot Hres}{HFOV} \tag{10.2}$$

Table 10.1: Velocity for navigation at $npixel = 3$

Distance to surface d [cm]	Camera displacement [mm]	Velocity [cm/s]	Pixel blur [-]
10	0.129	0.7766	0.3
20	0.259	1.5533	0.6
30	0.388	2.3299	0.9
40	0.518	3.1066	1.2

According to Sieberth *et al.* [117], a camera displacement of 0.377 mm causes a failure of feature detection points of 72 % while a camera displacement of 0.529 mm corresponds to a 60 % failure. Looking at table 10.1 the limiting case is the 40 cm distance constraint, for which the pixel blur is acceptable and the detected features are satisfactory. Note that this estimate is performed on the basis of navigation direction transverse to the cameras axis. When moving along the forward axis, the velocity may be higher and can not be easily estimated. Furthermore, the velocities in table 10.1 are not strict limits but instead suggestions in order to decrease the risk of navigation failure.

10.3.3. Further Considerations

It will become clear in section 11.2.2 that the stereo cameras used for navigation should also be able to provide high resolution images for inspection. Unfortunately, while in use for navigation, the resolution is not high

enough. Therefore it will be discussed in section 11.2.2 how one of the stereo cameras should be deactivated for navigation and instead used for high resolution images when hovering for a short period of time. During this period, the navigation will have to mutate in order to perform as without the camera. Since this is only required for short hovering periods, this possibility appears to be feasible. The loss of stereo vision does result in scale variance, nevertheless during the short periods this method is applied, scale variance poses no significant consequence since ARTIS would be hovering in a constant point.

During the inspection periods in which the stereo cameras are used for inspection as opposed to navigation, the navigation shall be performed using only the IMU and the ToF sensors. The top and bottom ToF sensors are used to determine ARTIS' constant altitude while hovering while the IMU shall be used to estimate its orientation while rotating during hovering and provide a rough position estimate. While the ToF sensors provide direct altitude data, the IMU needs to integrate measurements as shown in figure 10.8. The vertical position estimated from the ToF sensors can be combined with those from the IMU using the Kalman filter while the other position and orientation information is direct from the IMU.

10.4. GUIDANCE

Path planning and obstacle avoidance are activities of utmost importance for ARTIS' performance, being one of main points of concern of key risks and requirements (see **AMD-F-GN&C** requirements in chapter 5 and risk **C5** in chapter 6). ARTIS' guidance shares features with standard robotic motion planning problems, those being partial knowledge of the fuel tank environment, and flight in a three-dimensional domain filled with obstacles [118]. The present discussion addresses all the factors that influence and determine the autonomous vehicle guidance, generating an obstacle-free path taking into account the spatial characteristics of encountered obstacles while modifying the trajectory of ARTIS in real time so as to avoid collisions [119].

As was hinted in figure 10.2, autonomous flight is characterized by intricacies between the navigation, guidance and control aspects of ARTIS. In abstracting the characteristics of the guidance process, those relations shall be kept in mind. Figure 10.10 presents a high-level overview of the guidance process, highlighting the dependence on control and navigation inputs and outputs. The remainder of the present discussion will follow the logic of the guidance process, firstly expanding on the mapping of the fuel tank environment and the implemented algorithm for path planning.

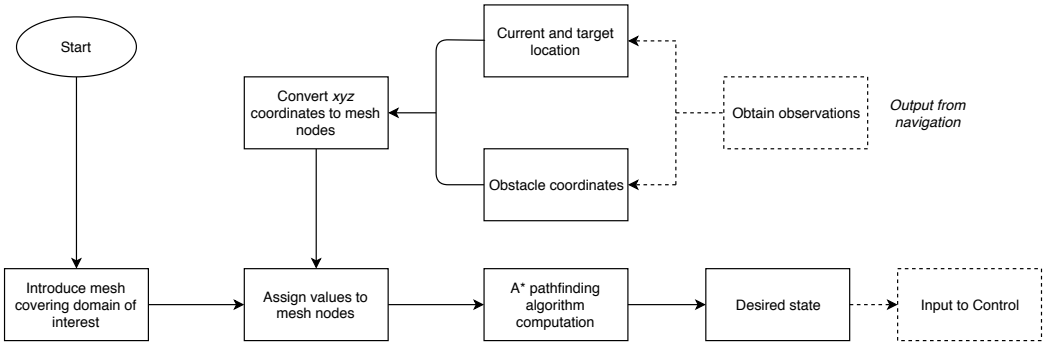


Figure 10.10: Flow chart logic of guidance process

10.4.1. ENVIRONMENT REPRESENTATION

To successfully generate a collision-free path, an accurate environment representation storing information about occupancy is needed. The easiest solution for 3D applications is the use of a voxel grid, in which the domain is segmented into a grid of smaller sub-elements (voxels), with each voxel storing information about its occupancy [29]. In research literature a grid representation of the environment is often implemented for motion planning algorithms of autonomous UAVs ([120, 121] among others). For guidance purposes, the spatial extent of the fuel tank of the C-130 will thus be modelled as a map with occupancy data. The reason for this choice, besides the substantial use in simulations and subsequent real-world applications, lies in the ease of implementing path planning algorithms, most of them relying on attractive or repulsive values assigned to grid points [122]. The effect of these values could then be leveraged for computing obstacle-free

paths in a cluttered environment.

The fuel tank environment is discretized by introducing a mesh covering the domain of interest, with the obstacle representation at the grid nodes to be interpreted as one of two possible discrete states using Boolean values that store information regarding the safety of moving to a specified node in the planned path [123]. Furthermore, each site of the C-130 virtual fuel tank environment has a definite neighbourhood of bordering cells, with a non-toroidal geometry, denoted by the presence of fixed boundaries, modelling the enclosed space [124].

The graph connecting such nodes has a correspondence with 3D point coordinates in the xyz space as, when storing occupancy locations, those can be expressed in either grid or world coordinates. In a multidimensional array representation, each grid node, or voxel sub-element, is characterized by three subscripts for indexing, as shown in Figure 10.11. Figure 10.11 shows an example start and goal position, characterized by the three subscripts $(1,1,1)$ and $(2,2,9)$, respectively.

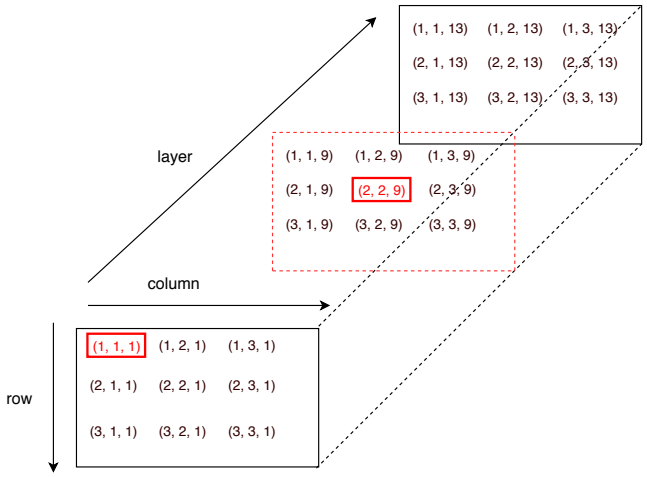


Figure 10.11: Start and target location characterized by subscripts in a multidimensional array

The number of grid points in x, y and z directions is strictly dependent on the level of physical complexity that needs to be captured and the memory capabilities of ARTIS. For instance, the inadequacy of the lattice model to reproduce the exact system geometry can be mitigated by improving the grid refinement in regions where the physical environment is characterized by high complexity. Such discussion shall be further developed in section 10.4.4.

10.4.2. A* Path Planning Algorithm

A^* algorithm was chosen for planning the path between multiple nodes due to its tested accuracy [125]. Compared to other relevant path planning algorithms (see Bellman-Ford and Vector Force Fields among others), A^* employs best-first search in order to find the least-cost path from a starting grid point to the target location. By performing the path finding computation per compartment of the fuel tank (hence having start and goal node for each compartment), the path computation is relatively short, hence an optimal fit for the capabilities of A^*, being considered among the most common and efficient used algorithm in short path finding problems [126]. Furthermore, its implementation is developed and extensively documented [122].

The A^* algorithm belongs to the category of best-first search models, expanding on nodes closest to the goal [127]. The C-130 fuel tank, filled with obstacles in a closed environment, is characterized by possible free locations that the A^* sees as the vertices and edges of an undirected graph, with the grid coordinates describing the location of each vertex [128]. Each time a node is expanded, an estimate of the remaining distance to the destination, the so-called heuristic value, informing the next move, is needed [129]. A^* is guaranteed to find a path if one exists, but how quickly it does so depends heavily on the quality of the heuristic [129]. Ideally, the heuristic estimates of the remaining distances should be as close as possible to the actual remaining distances, but not greater. If these estimates are too small, the frontier, containing nodes ARTIS has seen but not explored yet, will expand unnecessarily and more time will be wasted [130]. In fact, if the heuristic were all zero, every vertex in the graph that is not further away than the closest goal node, would have been explored, which is what Dijkstra's algorithm, another widely used path finding solution, does [131]. On the other hand, if the heuristic is overestimated, A^* will find a path if it exists, but it may not be the shortest path. Furthermore, the choice of heuristic highly depends on the nature of the problem [132].

Used to focus the search and evaluate the path cost to the goal, a Manhattan distance is often executed on grid-like environments as heuristic, its formulation being $d(A,B) = | A_x - B_x | + | A_y - B_y | + | A_z - B_z |$, for two points A and B. The efficiency of the Manhattan distance as heuristic is mostly leveraged for a square grid, thus not completely resembling the C-130 fuel tank environment [133].

Another possibility, guaranteed not to overestimate the true cost of the optimal path between the nodes, is the Euclidean distance, which was implemented in the MATLAB guidance program [134]. The chosen heuristic is heavily implemented in formulations of A^* algorithms and, for certain applications, more accurate than the mentioned Manhattan distance [135].

In a terrain with minimum movement cost D, i.e. the lowest cost between two adjacent nodes whose effect on the algorithm performance is expanded on in section 10.4.4, the Euclidean distance was formulated as shown in algorithm 10.1.

Algorithm 10.1 Pseudocode formulation of Euclidean distance

```
function EUCLIDEAN DISTANCE(node)
    dx = abs(node.x - goal.x)
    dy = abs(node.y - goal.y)
    dz = abs(node.z - goal.z)
    return D * sqrt(dx * dx + dy * dy + dz * dz)
end function
```

In the A^* algorithm implementation, performed with MATLAB, the heuristic distance from each vertex to the destination is stored in the h value, whereas the distance of the current node from the start position is given by the g value (see algorithm 10.2). Two lists will be maintained, one of the so-called open vertices and one of the closed vertices. At the beginning of the algorithm, the starting vertex is added to the list of open vertices, thus becoming the current vertex, as shown in green in the left part of figure 10.12, where the portion of an example 2D weighted graph from start node A to target node Z is presented.

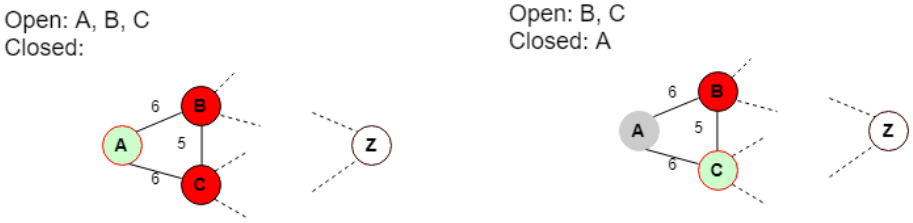


Figure 10.12: Current vertex switching from node A to C

Its g value, the distance ARTIS travelled from the start to get to it, is then computed. In the first case, the distance of the starting location to the starting location is obviously zero, as shown in table 10.2, which also reports example heuristics for the 2D example graph. The g value for the starting location is then added to the heuristic distance from the starting location to the goal position, the outcome being stored in the f value [136].

The vertices adjacent to the current vertex are then added to the list of open vertices (in red in the left part of figure 10.12) and their f values are calculated by adding their distance from the start location to their h values. The vertices ARTIS came through to get to each of these are stored in the so-called previous vertex list (last column of table 10.2). At this point, the starting location is added to the list of closed vertices. A new current vertex is then selected from the list of open vertices, the most promising one being the one with the lowest f value, as shown in green in the right part of figure 10.12.

Table 10.2: Required values for initial steps of A^* algorithm

Vertex	dist. from A (g)	Heuristic (h)	f = g + h	Prev. vertex
A	0	17	17	
B	6	18	24	A
C	6	13	19	A

Figure 10.12 shows a 2D example of the described steps, with the current vertex switching from node A to C, and node A being added to the list of closed vertices in the subsequent step. The heuristic values to the target goal Z and other required variables are shown in table 10.2.

Towards the last stages of the path finding algorithm, there will be only a few vertices to open, one of them being the destination, the one with the lowest f value. Once the path is found, the search is over. Since every vertex has a record of the vertex that came before it, these can now be traced back to obtain the path sequence ARTIS followed. Notice that ARTIS does not need to visit every single vertex in the graph. The pseudocode formulation of A^* is shown in algorithm 10.2 [137].

Algorithm 10.2 Pseudocode formulation of A^* path finding algorithm

```
Initialize open and closed lists
Set initial vertex as current
Compute heuristic distance of initial vertex to destination h
Compute f value for initial vertex (f = g + h, g = 0)
while current vertex is not the destination do
    for each vertex adjacent to current do
        if vertex not in closed list and not in open list then
            Add vertex to open list
        end if
        Compute distance from start (g)
        Compute heuristic distance to destination (h)
        Compute f value (f = g + h)
        if new f value < existing f value or there is no existing f value then
            Update f value
            Set parent as the current vertex
        end if
    end for
    Add current vertex to closed list
    Remove vertex with lowest f value from open list and set it as current
end while
```

The MATLAB implementation of the algorithm begins by setting up the open and closed lists. Then, the starting vertex is made current. Being a special case, it is processed outside of the main loop. Its h value is then computed, followed by its f value. The main loop comes then into play, running until the destination will be found, i.e. until the destination becomes the current vertex.

Inside the main loop, another loop deals with some of the current vertex potential successors, its neighbours which are not yet closed. As each neighbour is visited it is added to the open list and its g, h and f values are calculated.

The f values of the current vertices' neighbours are then updated. If there is already a f value, and the new one is smaller, the existing f value is replaced. At this stage, the parent of each neighbour is also set to be the current vertex, which keeps track of the paths.

When each pass of the inner loop comes to an end, the current vertex is closed and the new current vertex is selected from all of the open vertices, namely the one with the lowest f value.

The outer loop repeats to process each current vertex in the same way. Once the outer loop has run its course, the program has generated all of the information needed by ARTIS. The final path distance will be the g value of the current vertex, and it can be traced back through its parents to derive the path that was followed.

10.4.3. Code Implementation

An explanation of the discretization approach and the details of the A^* algorithm were given. The present section aims at providing a more comprehensive understanding of the actual code implementation and how the different procedures glue together.

The MATLAB programming language was chosen due to the ease of performing multi-dimensional matrix multiplications. Furthermore, for path planning and obstacle avoidance, a language optimized for speed of development would be beneficial, hence the implementation in a general-purpose programming language like MATLAB [138]. As far as the implementation running on the drone goes, a lower level programming language like C++ will be used.

Five MATLAB files were defined, with functions accepting the full paths to those files as inputs. The relations between the different files and functions are schematically represented in figure 10.13.

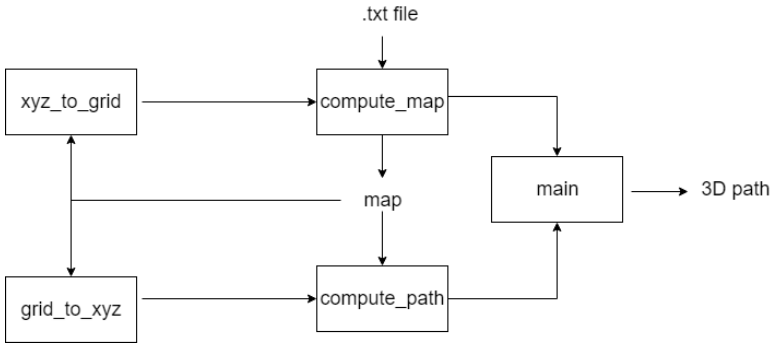


Figure 10.13: Overview of MATLAB functions with inputs and outputs for ARTIS guidance

The navigation solutions explained in section 10.3 allow momentaneous storing of the observation data, in the form of current and target location, whereas the stereo images generate a depth map leading to information about obstacle locations, as shown in figure 10.10. For the developed MATLAB code to function, the output from navigation is then structured as a text file and fed into the guidance channel (the actual implementation of the algorithms would be performed in C++). Such operation is performed at a rate of 100 Hz, from the IMU data rate (see section 10.1.1), hence allowing for real-time trajectory replanning. For the purposes of the present discussion, a one-time repetition of the actions performed by ARTIS is explained, but it is worth stressing how the GNC subsystem works simultaneously, with no separate actions being performed or isolation of processes. In fact, the inputs and outputs are constantly being fed back to the different subroutines.

The $compute\_map$ function handles the .txt input file, outputting the map structure array which groups data in the form of numeric, logical, and cell arrays data types. The map structure contains the discrete environment representation of the fuel tank, whose explanation was given in section 10.4.1.

The consequences of inaccurate obstacle avoidance and path planning laid out in risk **C5** (see chapter 6) are mitigated by adding margins to the obstacle coordinates provided by navigation. The occupancy grid map, generated by looping through the obstacles, assigns logical values to grid nodes in the obstacle locations, as well as the adjacent ones with margins, thus preventing the possibility of collision.

The A^* algorithm is formulated in $compute\_path$, requiring map as input argument value. The functions $compute\_path$ and $compute\_map$ make use of the $grid\_to\_xyz$ and xyz_to_grid functions, dealing with the conversion of indices in the occupancy grid into 3D points in xyz space and vice-versa, respectively. Furthermore, the structure map, used as input by xyz_to_grid, which is fed back to $compute\_map$, as shown in figure 10.13, is already defined when the latter is called, so as to avoid the inconvenience of an infinite loop. Finally, in $main$, the MATLAB interpreter reads the source file and executes the code found. The output, in the form of $Nx3$ coordinates, is the optimized 3D collision-free path from start to goal location through N nodes.

10.4.4. Performance Analysis and Results

This section aims at assessing the guidance programs overall performance and the effect of code design changes on the computational speed. The guidance solutions will be executed by ARTIS' 32-bit ARM Cortex-M4 (CPU), introduced in chapter 9, requiring a not negligible amount of resources for running.

The grid point refinement of the C-130 fuel tank's virtual environment plays a major role in the time complexity of the coded programs. Apparently, the domain is replaced with a number of finite nodes or voxels sub-elements introducing discretization errors. Shallow mesh refinement is expected to consume less computational resources than a finer one, with the drawback of not accurately capturing the exact system geometry. The voxel grid map might then misrepresent the actual C-130 fuel tank environment, leading to the A^* algorithm being unable to compute a collision-free path. Conversely, a finer mesh refinement drastically increases the computational costs by expanding the number of nodes and decreasing the smallest item dimension. In figure 10.14, the effect of varying grid refinements is plotted against the total elapsed time for A^* to run and the number of nodes visited by the algorithm.

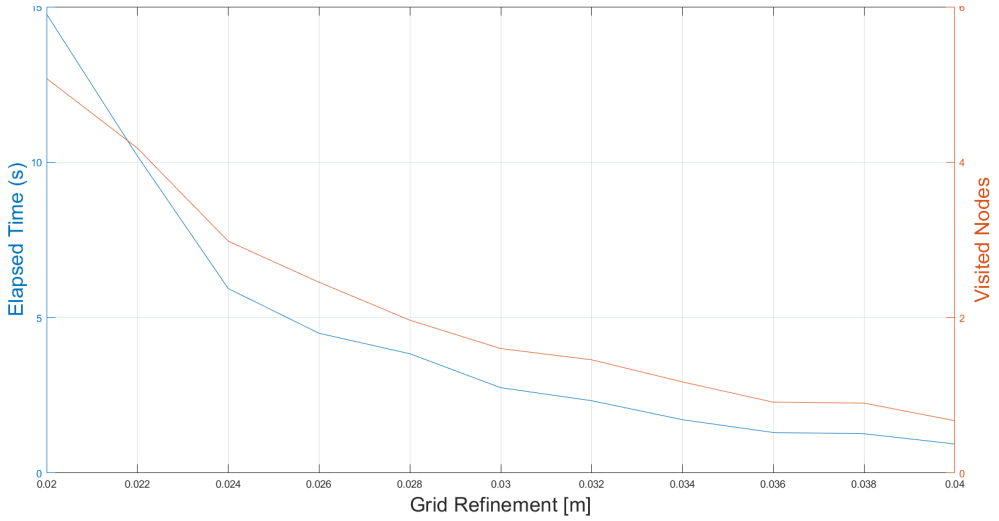


Figure 10.14: Grid Refinement Effect on A^* performance

The environment chosen is loosely representative of the fuel tank, with a 20 m length in the span-wise direction. Obstacles have been added, resembling the ones encountered in the C-130, the main goal being the study of the effect of the grid refinement on the A^* algorithm performance. The horizontal axis of the above plot shows the grid refinement, ranging from voxel width of 2.0 cm to a node every 4.0 cm in the span-wise direction of the wing box. The shallower the refinement, the lower the number of nodes seen by the path-planning algorithm and the time complexity. The main conclusion to be retained from above plot is the observed trend, with higher resolution consuming more time resources. The plot is presented to get a feeling for the numbers, but the final variable to be chosen for the grid resolution is strictly dependent on the C-130 fuel tank environment and the integration with the embedded systems on-board of the ARTIS. For the purposes of the present discussion, having to trade between elapsed time and required complexity, a node location every 4.0 cm was chosen. It is believed this refinement is sufficient to capture the environment complexity while requiring relatively low computational resources. The selected grid refinement was then used while running the algorithm in the other scenarios analyzed in the present discussion.

Another factor affecting the MATLAB code performance is the minimum movement cost D between two adjacent nodes, present in the A^* heuristic introduced in section 10.4.2. The impact of the cost value on the algorithm is shown in figure 10.15.

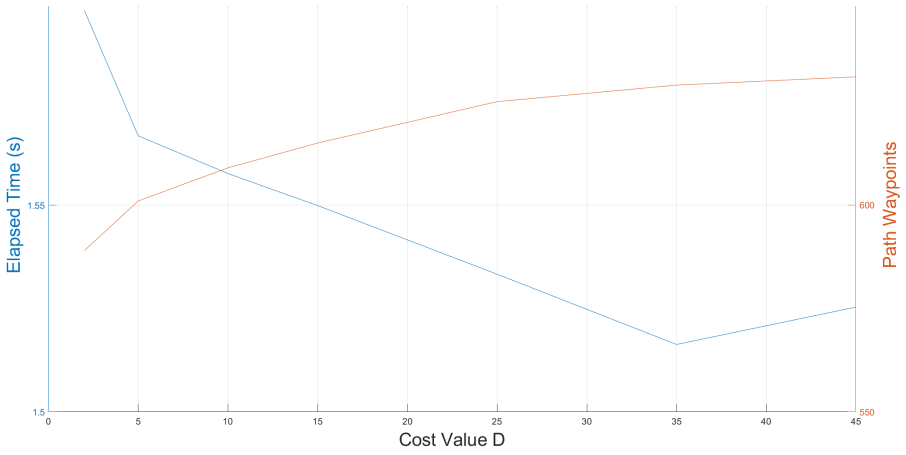


Figure 10.15: Effect of Cost Value from Heuristic Distance Function on A^*

The trend observed in figure 10.15 illustrates the features of the chosen heuristic. For a given f, as the cost value D, and, as a consequence, the heuristic h increase, the g value (i.e. the distance travelled from the start to get to the current node) loses its relative importance. A "greedy" best-first search approach is then used, since, in the limit of h growing unbounded (i.e. $\lim_{h\to\infty}$), no backtracking is being performed by the algorithm. In this case, ARTIS gets stuck in unsuccessful paths, increasing the depth of the solution to get to the goal location. Conversely, since no backtracking is being performed (g is too small compared to h), the algorithm is only looking at nodes along its path, thus overall visiting less nodes and requiring lower time. Of course, in the long-run, as the path waypoints keep on increasing, the elapsed time tends to increase as well, as shown by the final peak. The same setting for the map discretization and algorithm implementation used to obtain the results shown in figure 10.14 were used also in this case. What can be argued from figure 10.15 is that the algorithm is performing as expected, but no final choice can be made regarding the most efficient cost value D, since the trade-off between elapsed time and visited nodes shall be performed when studying the real operating environment of ARTIS. In the rest of the discussion, an indicative value $D = 2.0$ will be used.

When running the MATLAB programs, the files $compute\_map$ and $compute\_path$ are the two most computationally involved, with heavy memory requirements. When coding the programs, variables were stored as 2-byte (16 bit) signed integers. Hence, a first-level estimate of the demands imposed on the memory by the discretized approach to path planning can be found by multiplying the number of visited nodes of the algorithm by the number of bits per node. In the MATLAB files shown in figure 10.13, 5 arrays were initiated, those being the back-tracking map, with each vertex storing a record of the vertex that came before it, the multi-dimensional list of open and closed vertices for the A^* algorithm, and the f and g values (see algorithm 10.2 for clarification). Each visited node is thus characterized by 5 variables, representing values carrying information about the mentioned 5 grids. With each variable being a 2-byte integer, 80 bits are stored per variable, after the MATLAB codes are executed. Using the conditions for efficiency discussed thus far (i.e. a node every 4.0 cm and the cost value D set at 2.0) resulted in 6729 visited nodes when running A^*. The combination of the discretized approach to modelling the C-130 fuel tank environment and the chosen path planning algorithm resulted in a memory requirement of $6729 \times 80 = 538320$ bits. The computed value is only to be intended as a rough estimate to get a feeling for the numbers since no temporary variables were considered in the calculation and the choice of environment would heavily influence the results due to higher (or lower) grid nodes.

Furthermore, the main source of computational complexity lies in the A^* algorithm. The Big-O notation, defining the upper bounds of an algorithm, can be used to assess its theoretical worst-case scenario [139]. Denoting the number of successors generated by a given node per state by b and the depth at which the solution node was detected by d, the time complexity of the A^* path-planning algorithm is then given by: $O(b^d)$ [140]. Depending on the application of the algorithm, i.e. the environment in which ARTIS operates, different results can be obtained. Hence, depending on the size of the input data set, the performance will

grow exponentially. Computing the order of magnitude with no reference values would yield no insights into the time complexity of the algorithm. Hence, to get a feeling for the numbers, the effective branching factor b^* will instead be computed. It represents the number of successors generated by a typical node for a given search problem [141], and efficient algorithms characterized by a robust heuristic are those with a low effective branching factor, the optimal case being $b^* = 1$ [140]. With the total number of processed nodes denoted by N, the formal definition of the effective branching factor is shown in equation (10.3).

$$N = b^* + (b^*)^2 + ... + (b^*)^d \tag{10.3}$$

With no closed form solution available, a close guess is given by $b^* \approx N^{\frac{1}{d}}$ [141]. The A^* algorithm was run on a sample environment with obstacle coordinates, stored in the $text$ file (see figure 10.13), aiming to loosely resemble the C-130 fuel tank. The total number of visited nodes was 6729, whereas the depth of the solution, i.e. the number of xyz coordinates after which the solution node was found, resulted 571. Substituting the numbers resulted in the effective branching factor being equal to $b^* \approx N^{\frac{1}{d}} = 6729^{\frac{1}{571}} \approx 1.02$. The latter being close to the optimal effective branching factor confirmed the feasibility of the Euclidean as heuristic and the A^* algorithm as a path planning solution.

Section 10.4.2 briefly touched upon Dijkstra's algorithm, of which A^* can be considered an informed variation [142]. As mentioned before, A^* employs best-first search to find the least-cost path from a starting grid point to the target location, according to the $f = h + g$ value introduced in algorithm 10.2. Dijkstra's algorithm differs in its use of a non-informative heuristic (i.e. $h = 0$) [143]. Having argued the time complexity and memory requirements of A^*, it is instructive to perform a final comparison between the two algorithms. Slightly altering the $compute\_path$ MATLAB code, a version of Dijkstra's algorithm was obtained. Running the two algorithms produced the results reported in table 10.3.

Table 10.3: Comparison between pathfinding algorithms

Algorithm	***Path Waypoints***	***Visited Nodes***	***Elapsed Time [s]***
Dijkstra	571	18897	2.35
A*	571	6729	0.895

The depth of the solution proposed by Dijkstra's algorithm is the same as the one of A^*, since both algorithms are guaranteed to find the most optimal path as long as the heuristic is admissible (doesn't overestimate the cost to the target) [129]. The Euclidean distance definition makes it an admissible heuristic, hence A^* finds the most optimal path, in this case consisting of 571 waypoints. On the other hand, Dijkstra, with the non-informative heuristic $h = 0$, is still guaranteed not to overestimate the target since it also does not overestimate the cost [144].

Dijkstra's algorithm employs a simplistic cost function, at the expense of more than doubling the total number of processed nodes and elapsed time. The result is in accordance with the discussion regarding the advantages of A^* over other solutions and serves as further evidence of the merits of the proposed algorithm.

All of the above discussion regarding the performance analysis of the guidance solution was said to be executed on a MATLAB-generated environment representative of the C-130 fuel tank. A display of the planned trajectory inside such hazardous background with the drone size constraint being included is offered in figure 10.16, with the thick black line representing the planned path, from start to goal location. Assigning a size to the ARTIS resulted in the exclusion of certain paths, as occupied voxels too close to each other, constraining a safe movement of the drone, would have to be discarded. As can be argued, the dimensions in x and z direction are largely overestimated, with only the length in the span-wise direction being representative of the fuel tank environment. This setting has been done on purpose, to check the performance of A^* and the discretization approach over a larger domain, thus requiring larger use of computational resources and processed nodes.

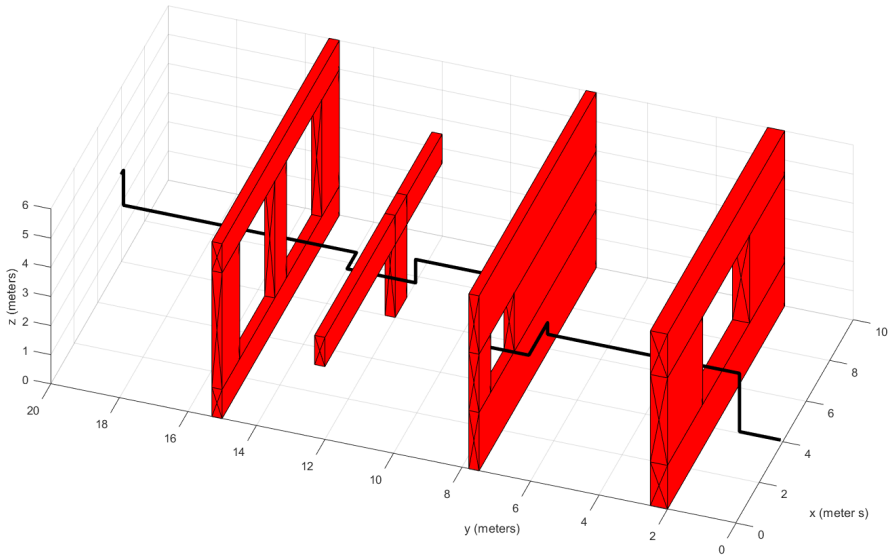


Figure 10.16: Path planning & obstacle avoidance visualization using A^* algorithm

The performance analysis and results obtained are thus encouraging that a discrete approach to the mapping procedure could be a suitable solution. A source of potential improvement lies in the non-negligible memory requirement imposed by the discretization and proposed algorithm for path planning. Alternative approaches to the guidance solution to be further investigated in later iterations of ARTIS' design are proposed in section 10.4.5.

10.4.5. Future Approaches to guidance problem

Trading possible paths via A^* algorithm in a discrete environment has proven to work efficiently and with sufficient performance. Still, it can be argued that a highly complex environment may require increased voxel grid resolution and demanding memory requirements. As a preliminary design, ARTIS is believed to perform all of its functions with satisfying accuracy, but for later stages, different approaches to the guidance problem may be supported.

Yanamoto et al. used a B-spline (i.e. a basis spline) curve to generate curvature continuous paths [145]. The local trajectory replanning would then require lower computational resources since the grid nodes along the path need not be stored. A limited amount of path waypoints would still need to be formulated, but the trajectory generation would then become an optimization problem with the splines being fit through the points of the planned path to generate a smooth trajectory. Hence, the main difference with a discrete approach is that the route itself would not be represented by a set of discrete points but by polynomial segments [29].

Different approaches have been explored when considering a continuous domain instead of a discrete one, with the minimization of the line integral of a cost-weighting function to compute the path being a frequently used one [146]. Curvature continuous paths can be determined by using continuity conditions demanded for each spline segment [147].

The main drawback of a continuous approach to environment representation and path planning is the complexity of implementing such a solution which, with the limited resources the team had to deal with, was deemed inappropriate.

10.5. Control and Stability

This section shines light upon how the drone intends to self-stabilize. For this, a piece of software has been adopted and adjusted from the asbQuadcopter package in MATLAB, which provides a model-based design approach for writing quadcopter flight control logic [148]. The inputs and outputs have been adjusted, such that they represent the model used for simulation. The inputs are $[x, y, z, v_x, v_y, v_z, \theta, \phi, \psi, \omega_x, \omega_y, \omega_z]^T$ which

include the three position coordinates, three velocities and three attitude angles and their rates, respectively. The outputs are $[p, q, r, T]$, in which T represents the thrust. These outputs are then ran through the Motor Mixing Algorithm (MMA) which converts the desired state into thrust for the individual propellers.

Writing the motor mixing algorithm requires knowledge of the spin directions of the rotors and the rotor configuration. In order to balance the torques generated by the spinning rotors and to negate potential pitch and roll interactions with yaw, the spin direction of the rotors is defined by having opposing rotors with respect to the symmetry plane spin in opposing directions [4]. For the rotor configuration with respect to the body frame, a cross configuration is chosen as it is considered more desirable than a plus configuration. Although plus configurations generally result in improved acrobatic qualities, cross configurations generally provide more stability [31], which is beneficial for the purpose of photography and navigation in the tank. The resulting rotor configuration in the drone body frame as well as the rotor spin directions are visualized in figure 10.17.

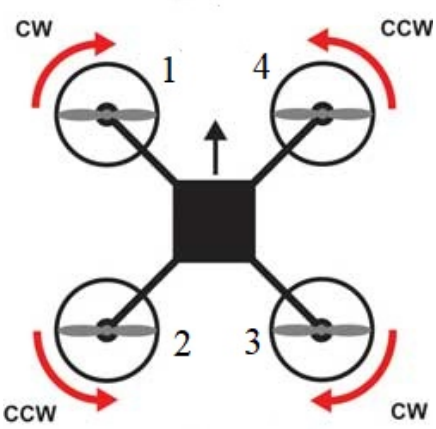


Figure 10.17: Rotor configuration in the drone body frame and rotor spin directions [3]

10.5.1. Flight Controller Technique and Logic

Before deciding on the control architecture, it is imperative that appropriate control techniques and schemes are selected with respect to the characteristics of the plant. It should be noted that the drone, i.e. the plant in this scenario, is nonlinear and multivariable in nature. Moreover, since the drone possesses of six degrees of freedom while only having four actuators, namely the four motors, the system is said to be underactuated. This in turn results in a strong coupling between the actuators and certain degrees of freedom, most notably between pitch and longitudinal motion and roll and lateral motion [149].

Looking at selecting a controller, both linear- and non-linear controllers should be considered. For non-linear controllers, there are sliding mode controllers, back-stepping controllers, and non-linear dynamic inversion controllers (NDI controller). The latter one ranks high up, as this controller has proven to be relatively easy to implement for non-linear control [150], and is promisingly robust when used as *Incremental* Non-linear Dynamic Inversion (INDI). This novel method does however have a technology readiness level of 6 out of 9, as it is not yet "proven in operational environment" [150]. Also, looking at the requirements critically, the only part of the model which is in fact non-linear is the transformation of angles to linear accelerations, which holds a sine function. Remaining under 10°, as expected, the maximum error would however only be a mere 1.6 %. Within this range, the system can thus be assumed linear, and thus a PID controller will suffice.

For the linear controllers, the most applicable linear controller is the PID controller [31]. PID controllers are generally easier to implement as well as to design and can be more intuitively assessed, based on the P, I, and D components, than is the case for nonlinear controllers [30]. It has been shown that PID controllers possess enough capability to track the desired system state of quadcopters [31, 149]. The drawback of applying a linear controller is that a linear controller only yields predictable control performance around hover, and as such performs with increasing error in the event that the drone veers off from its nominal state too excessively. ARTIS is however not expected to perform many, if none at all, aggressive manoeuvres in which is deviated significantly from the nominal state. Therefore it is expected the PID suffices. Should the controller be unable to satisfy the **AMD-F-GN&C-Cont** requirements, the PID can in turn be updated to include e.g. gain scheduling to better approximate the non-linearity of the system or be replaced by a nonlinear controller [31].

For the general overview of the control part, a flowchart has been established, which is shown in figure 10.18. Similarly to the code, this was adopted and adjusted from the MATLAB series on quadcopter control [4]. The loops for roll and pitch contain an inner- and outer loop, of which the outer loop controls the position. The inner loop adds the required control in order to attain the desired attitude. The total is then inputted into the MMA, and consequently, the motors make the drone move as desired.

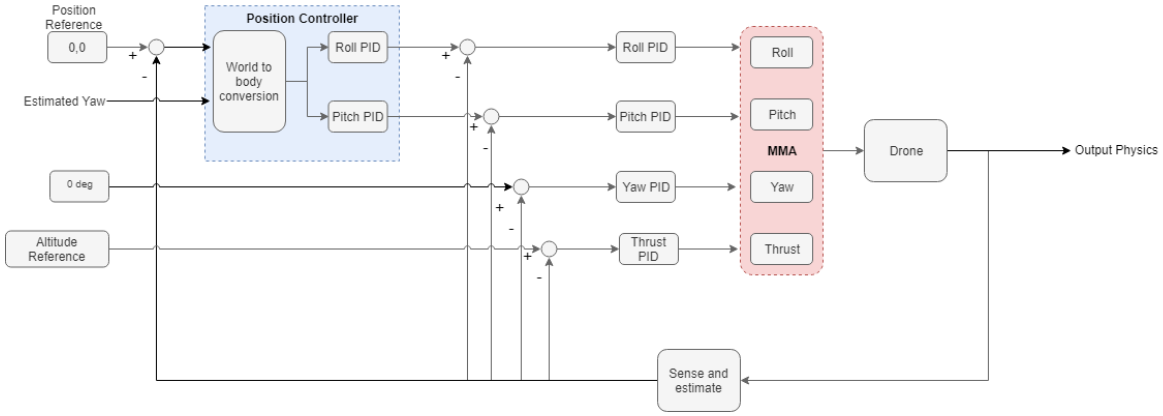


Figure 10.18: Control scheme with six PID controllers

10.5.2. Drone and Flight Environment Model

The quadcopter package in the MATLAB Aerospace blockset provides a nonlinear model that comprises nonlinear representations of the airframe, which is modelled using three-dimensional rigid body dynamics, the flight environment e.g. air density and temperature, and the sensors, which incorporate sensor noise and bias [4]. However, even though the nonlinear model most accurately represents reality it is unsuited to be used for the linear analysis and tuning of linear PID controllers, for which a linear model of the drone and flight environment is desirable. Accordingly, the nonlinear model needs to be linearized for the tuning of the PID controllers, after which the nonlinear model can be leveraged for the verification of the performance of the tuned PID controllers [4].

The main flow of the inner- and outer control loops are displayed in figure 10.18. The actual MATLAB version of this, used for gain tuning, is displayed in figure 10.19. Evidently, the latter resembles the former closely.

10.5.3. Flight Controller Tuning

The tuning of the PID controllers was performed by building a linear model based on the quadcopter project in MATLAB its Aerospace blockset [4]. Accordingly, the sensors were removed from the model in an effort to remove all nonlinear components from the model. The states are as such directly fed into the controllers, thereby assuming that the controllers know the states precisely. The resulting Simulink tuning model is shown in figure 10.19, where the control scheme including the six PID controllers and MMA can be seen on the left hand side of the model.

The PID controllers are subsequently tuned by virtue of the Ziegler-Nichols method. For the standard PID control law, as is shown in equation (10.4), the Ziegler-Nichols method provides estimates for the proportional gain K_p as well as the integral time τ_i and derivative time τ_d, and as such for K_i and K_d [32].

$$u(t) = K_p e(t) + K_i \int_0^t e(t')\,dt' + K_d \frac{de(t)}{dt} = K_p \left(e(t) + \frac{1}{\tau_i} \int_0^t e(t')\,dt' + \tau_d \frac{de(t)}{dt} \right) \tag{10.4}$$

The principle behind the Ziegler-Nichols method is that K_p, τ_i, and τ_d are estimated using the proportional gain for which the system becomes marginally stable, while setting the integrator and derivative gains to zero, and the period of the oscillations of the marginally stable response. For a given proportional gain for which marginal stability is achieved, also known as the ultimate gain K_u, and the corresponding oscillation period T_u, the values for K_p, τ_i, and τ_d for a specific control type (e.g. PI or PID) are calculated by using the formulae presented in table 10.4 [32].

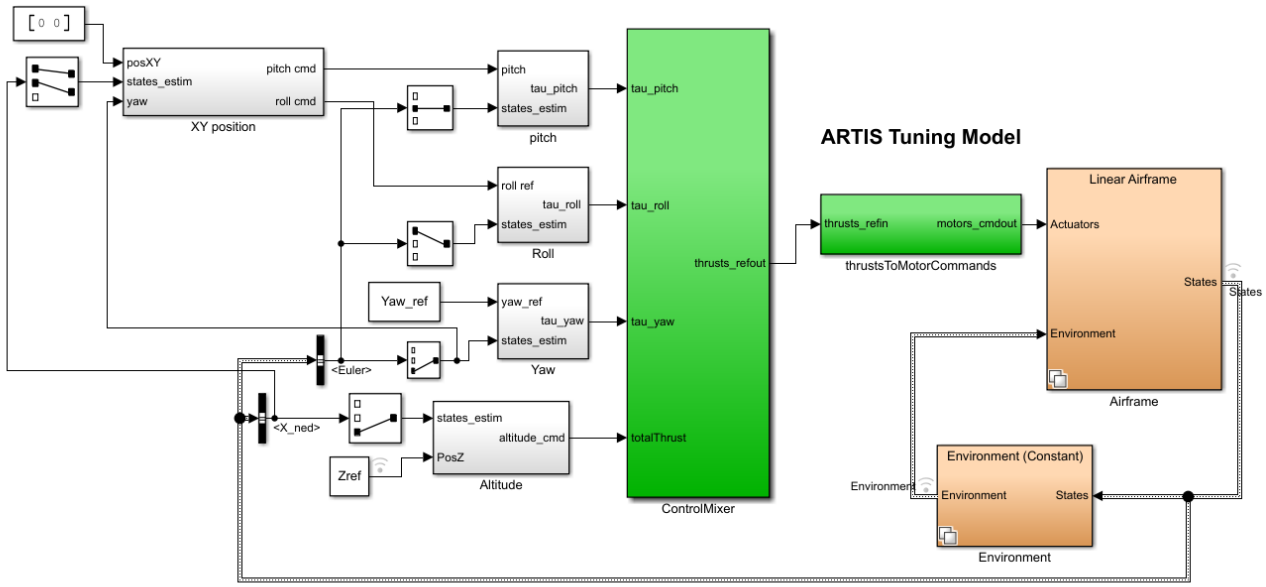


Figure 10.19: Simulink controller turning model [4]

Table 10.4: Formulae for K_p, τ_i, and τ_d for several control types for the Ziegler-Nichols tuning method [32]

Control Type	K_p	τ_i	τ_d	$K_i \left(= \frac{K_p}{\tau_i}\right)$	$K_d \left(= K_p \tau_d\right)$
P	$0.5K_u$	-	-	-	-
PI	$0.45K_u$	$\frac{T_u}{1.2}$	-	$0.54\frac{K_u}{T_u}$	-
PD	$0.8K_u$	-	$\frac{T_u}{8}$	-	$\frac{K_u T_u}{10}$
Classic PID	$0.6K_u$	$\frac{T_u}{2}$	$\frac{T_u}{8}$	$1.2\frac{K_u}{T_u}$	$\frac{3}{40}K_u T_u$
No overshoot	$\frac{K_u}{5}$	$\frac{T_u}{2}$	$\frac{T_u}{3}$	$\frac{2}{5}\frac{K_u}{T_u}$	$\frac{K_u T_u}{15}$

The "classic PID" control type in table 10.4 is used to tune the attitude controllers, whereas the "no overshoot" control type is used to tune the altitude and position controllers. The former is done to generate a relatively fast response for attitude corrections, while the latter is done with the purpose of minimizing the chance of the drone colliding with the walls of the fuel tank while updating its position and altitude. The resulting controller should lead to an effective mitigation of risk **C2**, the loss of stability and control.

The inner loop PID controllers (i.e. the attitude and altitude controllers) are tuned one at a time, while setting the gains for the other inner loop controllers to zero. The Ziegler-Nichols method is then employed to analyze and determine the appropriate response and corresponding gains. The outer loop controllers are subsequently tuned analogously, except for the fact that the inner loop controllers are now active and ensuring proper orientation and altitude [4]. An example of the Ziegler-Nichols tuning method is displayed for the altitude controller in figure 10.20, in which the integrator and derivative gains are set to zero and the proportional gain is increased to acquire marginal stability, and figure 10.21, which shows the no-overshoot system response after applying the PID gains that are attained after using table 10.4. The response can then be modified by slightly altering the gains to achieve the desired response.

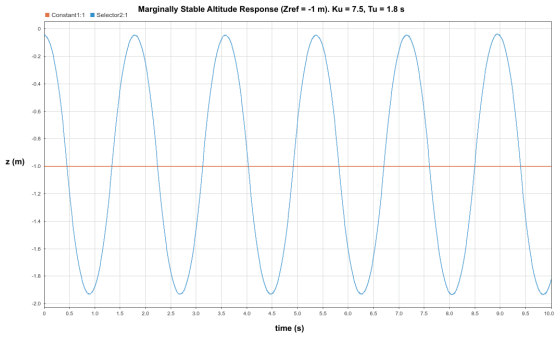


Figure 10.20: Marginally stable altitude response $(K_p = K_u = 7.5, T_u = 1.8\text{s})$

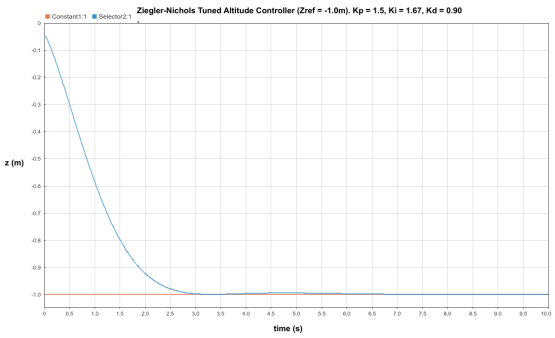


Figure 10.21: Tuned altitude response using the "no overshoot" control type $(K_p = 1.5, K_i = 1.67, K_d = 0.9)$

The full set of the PID controller gains that were acquired from the above described method are presented in table 10.5.

Table 10.5: K_p, K_i, and K_d for the six PID controllers

PID Controller	K_p	K_i	K_d
Altitude (z-position)	1.5	1.67	0.90
Yaw	0.081	0.073	0.0223
Pitch	0.0038	0.0016	0.0023
Roll	0.0038	0.0016	0.0023
x-position	-0.180	0.0765	0.140
y-position	0.180	-0.0765	-0.140

10.6. Simulation

This section highlights the construction of the simulation that was built with the purpose of visualizing the performance of the navigation, guidance, and control algorithms. The simulation can as such be used to validate the GN&C subsystem, which is further discussed in section 12.1.2

10.6.1. Physics

It is important that the physics necessary to build the simulation are first thoroughly analyzed. This subsection discusses the physics that are used in the simulation.

10.6.1.1. Reference Frames

In figure 10.22 the right-handed coordinate system used for the simulation is introduced.

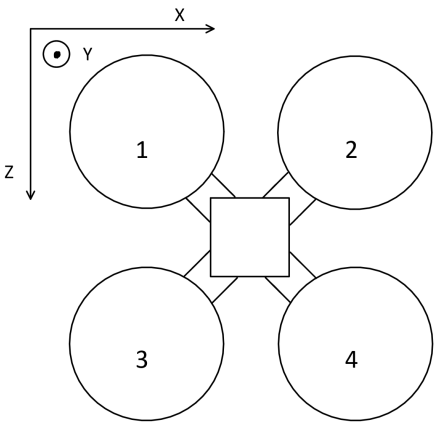


Figure 10.22: Reference frame and propeller numbering used for simulation

10.6.1.2. Rigid Body Kinematics

The drone inside the simulation is a rigid body. 6 DoF rigid bodies have a state vector containing position, velocity, angular rates, and orientation (in this case in quaternion form):

$$\mathbf{x} \equiv \begin{bmatrix} x & y & z & v_x & v_y & v_z & \omega_x & \omega_y & \omega_z & q_w & q_x & q_y & q_z \end{bmatrix}^T \tag{10.5}$$

The ODE describing the motion of a rigid body is (note the reference frame in figure 10.22):

$$\mathbf{f}(\mathbf{x}) = \dot{\mathbf{x}} = \begin{bmatrix} v_x & v_y & v_z & a_x & a_y & a_z & \alpha_x & \alpha_y & \alpha_z & \dot{q}_w & \dot{q}_x & \dot{q}_y & \dot{q}_z \end{bmatrix}^T \tag{10.6}$$

In equation (10.6), $\begin{bmatrix} \dot{q}_w & \dot{q}_x & \dot{q}_y & \dot{q}_z \end{bmatrix}^T$ is given by:

$$\begin{bmatrix} \dot{q}_w \\ \dot{q}_x \\ \dot{q}_y \\ \dot{q}_z \end{bmatrix} = \dot{\mathbf{q}} = \frac{1}{2} \begin{bmatrix} 0 \\ \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} \times \mathbf{q} \tag{10.7}$$

$$\mathbf{q} \equiv \begin{bmatrix} q_w \\ q_x \\ q_y \\ q_z \end{bmatrix} \tag{10.8}$$

Please note that "×" refers to quaternion multiplication. In equation (10.7), $\mathbf{q}$ is defined as:
The computation for linear acceleration and angular acceleration will be treated in section 10.6.1.3.

10.6.1.3. Quadcopter Kinetics

While section 10.6.1.2 describes the kinematics of a general rigid body, the present section features the kinematics used to describe the linear and angular accelerations of a quadcopter.

The linear acceleration is given by the forces acting on the quadcopter and its mass:

$$\mathbf{a} = \frac{\mathbf{F}(\mathbf{T})}{m} \tag{10.9}$$

The angular acceleration is given by the moments of inertia and the moments acting on the quadcopter. Note that the moments are given in body frame and are therefore transformed into world frame by means of two quaternion multiplications.

$$\boldsymbol{\alpha} = \begin{bmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{bmatrix}^{-1} \left(\mathbf{q} \times \mathbf{M}(\mathbf{T}) \times \mathbf{q}^{-1} \right) \tag{10.10}$$

There are multiple forces acting on the quadcopter that are modelled: the 4 propeller forces, gravity, and propeller flapping drag. The propeller forces are 4 point forces acting through the geometrical center of the propellers. The force imparted on the drone by propeller i is equal to $\frac{mg}{4} T_i$, where T_i is defined such that 0 is no thrust and 1 is the thrust required by all propellers to hover. These forces are acting in y-direction in the body frame and are transformed into world frame by the quaternion $\mathbf{q}$. Gravity is acting in the negative y-direction of the world frame. The flapping drag (coefficient $\frac{1}{2} \left[\frac{\text{kg}}{\text{s}} \right]$) is acting in the opposite direction of the velocity. Since the velocity is in world frame, this force does not have to be rotated.

$$\mathbf{F}(\mathbf{T}) = \underbrace{\mathbf{q} \times \left(\begin{bmatrix} 0 & 0 & 0 & 0 \\ \frac{mg}{4} & \frac{mg}{4} & \frac{mg}{4} & \frac{mg}{4} \\ 0 & 0 & 0 & 0 \end{bmatrix} \mathbf{T} \right) \times \mathbf{q}^{-1}}_{\text{4 propeller forces}} \underbrace{- \begin{bmatrix} 0 \\ mg \\ 0 \end{bmatrix}}_{\text{gravity}} \underbrace{- \frac{1}{2} \left[\frac{\text{kg}}{\text{s}} \right] \mathbf{v}}_{\substack{\text{propeller} \\ \text{flapping drag}}} \tag{10.11}$$

Besides the sum of forces, also the sum of moments has to be defined. Gravity—assuming a uniform gravitational field—does not produce any moments around the center of mass. The drag in this case is also assumed to not produce any moments. The magnitude of the moments created by the propeller forces around the x and z axis are equal to there magnitude multiplied by the moment arm: $\frac{mg}{4} \frac{d_{frame}}{2} T_i$. Around the the y axis, the magnitude of the moment is equal to the moment the motor exerts on the propeller. This moment is given by $Q = \frac{Q}{T} T = \frac{k_{Q_0} \rho n^2 D^5}{k_{T_0} \rho n^2 D^4} T = \frac{k_{P_0} D}{k_{T_0} 2\pi} T$ [151].

$$\mathbf{M}(\mathbf{T}) = \begin{bmatrix} \frac{d_{frame}}{2} & \frac{d_{frame}}{2} & -\frac{d_{frame}}{2} & -\frac{d_{frame}}{2} \\ \frac{k_{P_0} D}{k_{T_0} 2\pi} & -\frac{k_{P_0} D}{k_{T_0} 2\pi} & -\frac{k_{P_0} D}{k_{T_0} 2\pi} & \frac{k_{P_0} D}{k_{T_0} 2\pi} \\ -\frac{d_{frame}}{2} & \frac{d_{frame}}{2} & -\frac{d_{frame}}{2} & \frac{d_{frame}}{2} \end{bmatrix} \frac{mg}{4} \mathbf{T} \tag{10.12}$$

10.6.1.4. Numerical Solution

To solve the ODE proposed in section 10.6.1.2 and section 10.6.1.3 the widely used 4th order Runge-Kutta method is used [152]. It is explained in equation (10.13) – equation (10.18).

$$\mathbf{x}_{n+1} = \mathbf{x}_n + \frac{1}{6}(\mathbf{k}_1 + 2\mathbf{k}_2 + 2\mathbf{k}_3 + \mathbf{k}_4) \tag{10.13}$$

$$t_{n+1} = t_n + \mathrm{d}t \tag{10.14}$$

$$\mathbf{k}_1 = \mathrm{d}t\,\mathbf{f}(t_n, \mathbf{x}_n) \tag{10.15}$$

$$\mathbf{k}_2 = \mathrm{d}t\,\mathbf{f}\left(t_n + \frac{\mathrm{d}t}{2}, \mathbf{x}_n + \frac{\mathbf{k}_1}{2}\right) \tag{10.16}$$

$$\mathbf{k}_3 = \mathrm{d}t\,\mathbf{f}\left(t_n + \frac{\mathrm{d}t}{2}, \mathbf{x}_n + \frac{\mathbf{k}_2}{2}\right) \tag{10.17}$$

$$\mathbf{k}_4 = \mathrm{d}t\,\mathbf{f}(t_n + \mathrm{d}t, \mathbf{x}_n + \mathbf{k}_3) \tag{10.18}$$

The timestep $\mathrm{d}t$ is chosen at runtime such that the state of the drone is known at intervals of 1000 Hz for the control algorithm. Besides this interval, the simulation will also make steps to generate the state at the time of every frame that is drawn to the screen (60 Hz under nominal conditions).

10.6.2. Computer Graphics

To generate the computer graphics the simulation uses Microsoft's Direct3D 11, which is part of DirectX 11. Direct3D 11 provides a graphics pipeline that is programmable by shaders. The pipeline can be seen in figure 10.24. The simulation uses only the required shaders: Vertex Shader and Pixel Shader. The vertex shader is used to transform all vertices in the scene from model space into screen space via the model, view, and projection matrices. The pixel shader will then be run on all pixels of a given triangle and compute the color output.

For finding the color of each pixel the Phong Lighting Model is used [153]. The Phong Lighting Model uses simple algebra to give the appearance of ambient, diffuse and specular lighting in a scene. It uses a few vectors: **L** is the vector from a pixel to the light source, **N** is the unit normal vector of the surface that the pixel belongs to, **V** is the unit vector from the pixel to the camera, **R** is the reflected unit light vector. These vectors can be seen in figure 10.23

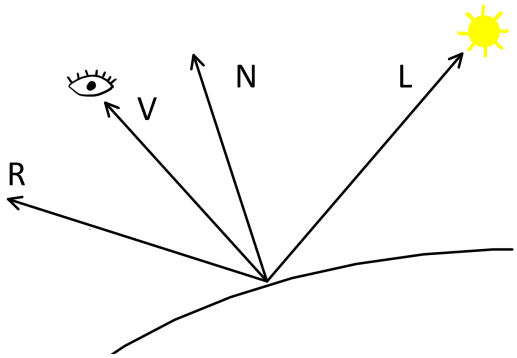


Figure 10.23: Vectors used for Phong Lighting

Colors in this section are expressed in the following format: $\begin{bmatrix} R & G & B & A \end{bmatrix}$ (red, green, blue, alpha (opacity)).

The Phong Lighting model considers three components for lighting: ambient (**A**), diffuse (**D**), and specular (**S**). Those components are added up in equation (10.19). The ambient lighting is just a flat color per material that is applied to a pixel independent of the lighting. The purpose of this is to be able to see the scene even without any lighting. It also simulates the effect of light that bounces multiple times in a scene so that shadows are not perfectly dark.

$$\mathbf{C} = \mathbf{A} + \mathbf{D} + \mathbf{S} \tag{10.19}$$

$$\mathbf{A} = \begin{bmatrix} A_R \\ A_G \\ A_B \\ 1.0 \end{bmatrix} \tag{10.20}$$

The diffuse lighting component, defined in equation (10.21), is given by a color $\mathbf{K}_d$ multiplied by the dot product of the normal vector and the light direction vector. Additionally it is multiplied by the inverse of the distance between the pixel and the light source squared.

$$\mathbf{D} = \mathbf{K}_d \max\left(\mathbf{N} \cdot \hat{\mathbf{L}}, 0\right) \frac{c}{|\mathbf{L}|^2} \tag{10.21}$$

The specular component is similar to the diffuse component, however the dot product of the reflected light vector instead of the light vector and the view vector instead of the normal vector is used. The dot product is further taken to the power of the variable N_s. This value determines how small and sharp the specular highlights will be.

$$\mathbf{S} = \mathbf{K}_s \max(\mathbf{R} \cdot \mathbf{V}, 0)^{N_s} \frac{c}{|\mathbf{L}|^2} \tag{10.22}$$

The vector $\hat{\mathbf{R}}$ is defined in equation (10.23);

$$\mathbf{R} = \hat{\mathbf{L}} - 2\mathbf{N}(\hat{\mathbf{L}} \cdot \mathbf{N}) \tag{10.23}$$

The result of running the full Direct3D 11 pipeline with the lighting model described above can be seen in figure 10.25. This figure shows the drone inside the fuel tank, with the four internal camera views shown on the left side.

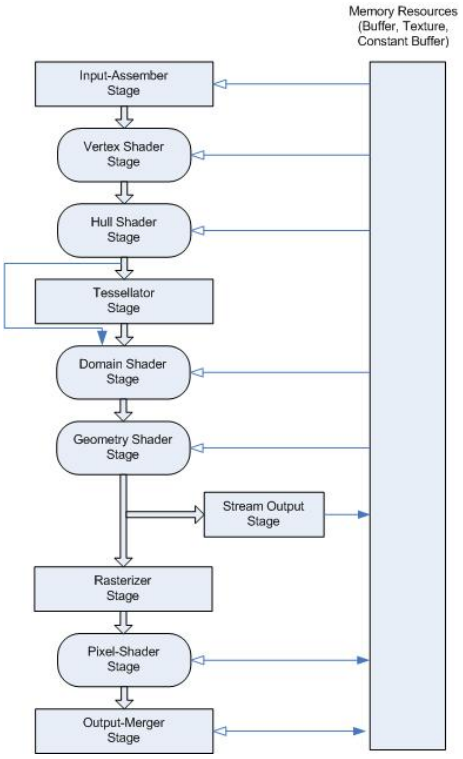


Figure 10.24: Direct3D 11 graphics pipeline [5]

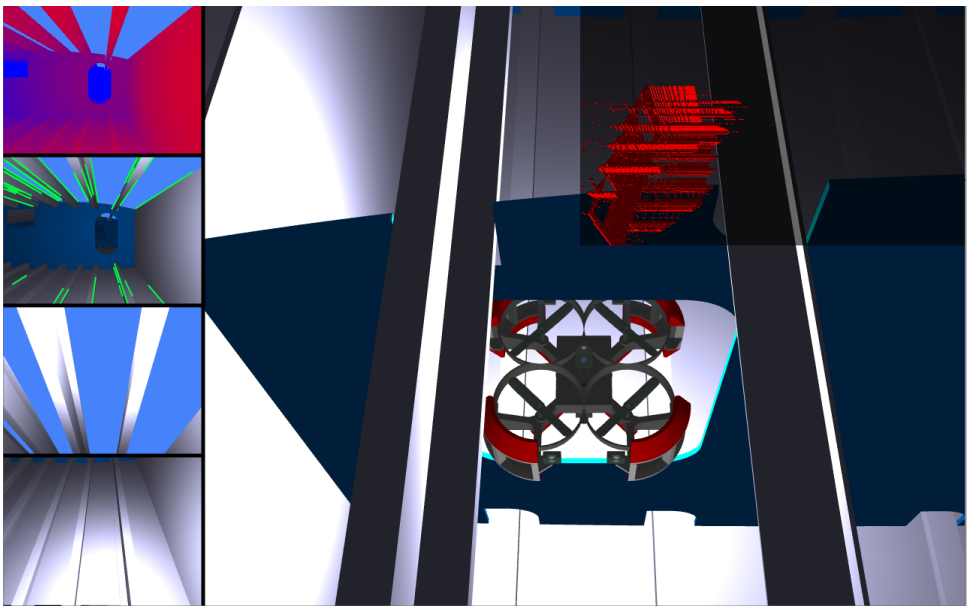


Figure 10.25: Screenshot of simulation. Left side shows internal camera views (first one is the depth map generated from the stereo images overlaid onto one of the images, second one shows line detection on stereo image); external view on the right; top-right shows voxel map generated from the depth map shown on the top-left

10.6.3. Control Interface

In order to implement the control software into the simulation, the control interface needs to be developed. This subsection features the development of the control interface.

The control algorithm was exported from Simulink using the C coder. Interfacing it was limited to accessing some global variables and calling the step function. However, one complication was that the reference frames between the simulation and the control algorithm as well as the propeller numbering was different.

To address this, all variables passed into the control algorithm are first transformed from simulation into control reference frame. After the control is run, the thrust output is converted from the propeller numbering used by the control to that used of the simulation.

10.6.4. Navigation Interface

Because of time constraints, the navigation algorithm explained in section 10.3 was not implemented but rather an open source VIO algorithm was used, namely MSCKF-VIO [154].

MSCKF-VIO is implemented on top the ROS framework. As such it can not run on Windows, like the simulation. To get around this issue, the navigation algorithm is run inside a Linux Virtual Machine and all necessary information such as the stereo video feed and IMU data is transferred to the VM over 3 different UDP sockets.

The stereo images are sent uncompressed (8 bit grey-scale) over two different sockets to make sure they are not flipped on the receiving end. To achieve this, the images first have to be copied from the GPU framebuffer to a separate texture that is not multi-sampled. Then the appropriate sections (part of the rendered frame that contains the stereo images) are copied to two more textures (one for each camera) that is mappable to CPU memory. The images are then copied from those textures in GPU memory to CPU memory and then sent over the socket. After they are received on Linux the server application assembles the ROS messages and publishes them. While this system adds some overhead, it is one of the easiest ways to make the simulation and the navigation run on two separate operating systems.

The result of running the navigation on the simulation data can be seen in figure 10.26. The algorithm ARTIS would be using post-DSE is using line features instead of corner features. An example of line features being detected in the simulation can be seen in figure 10.25.

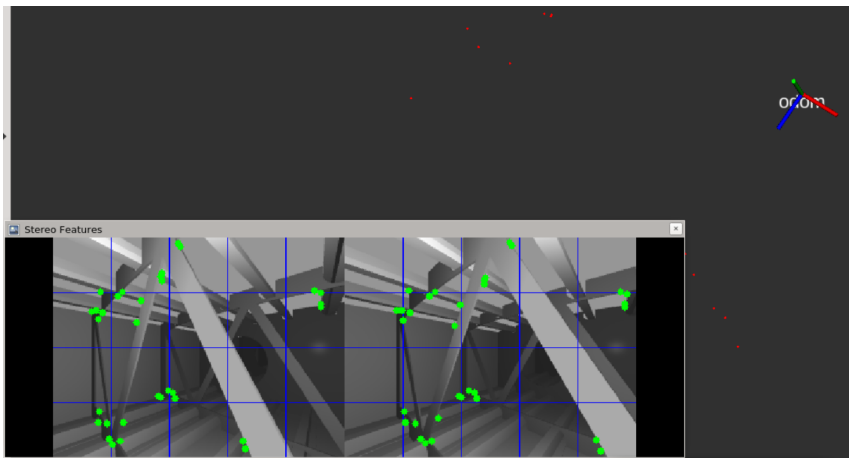


Figure 10.26: Navigation algorithm running on simulation

10.6.5. Guidance Interface

The guidance algorithm described in section 10.4 relies on having a voxel map to operate on. This voxel map has to be generated from the stereo images taken by ARTIS. This is a multistage highly parallel task that can be performed on the VPU. The first step is to generate a depth map from the stereo images. The SDK for the Intel Myriad X has built-in functionality for this. The next step is to generate a texture that does not only contain depth for each pixel but relative position from the camera in x, y, and z direction. This can be computed by multiplying a unit vector representing the direction each pixel has from the camera (depends on the camera model used (in the simulation this is the pinhole model)) by the depth. An example of this intermediary representation can be seen in figure 10.27. This texture is then sampled, and the position in world frame added to the relative position (rotated in to world frame first). The resulting position is then rounded onto the voxel grid and the resulting voxel position added to the list of occupied voxels. The resulting voxel map is then displayed in the simulation using instanced rendering and can be seen in figure 10.25.

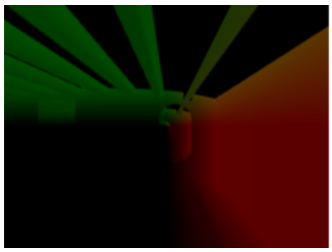


Figure 10.27: Texture with relative position in x, y, and z direction expressed as RGB components

11
Final Design Configuration and Performance

This chapter summarizes the final design of the ARTIS. First, the final configuration is presented. Then, the performance of the ARTIS is analyzed by looking at the mass and power breakdown and the performance inside the tank. The design then is analyzed for sensitivity to a change in inputs and finally, the resource and budget allocation for the final design is reflected upon.

11.1. Configuration and Layout

This section presents the final layout of ARTIS. This shows how all components and structures described previously come together in one design. In figure 11.1, the final ARTIS layout can be seen. On the front facing side, the two stereo vision cameras with a LED above them are visible. On the top, the third camera with LED is shown. The fourth camera is placed on the bottom and can not be seen in this image. The red parts on the four ducts are foam, which is added to protect both the drone and the fuel tank from impact damage. The drones name, TU Delft and Lockheed Martins logo are shown on the red foam to show the ARTIS' main parties.

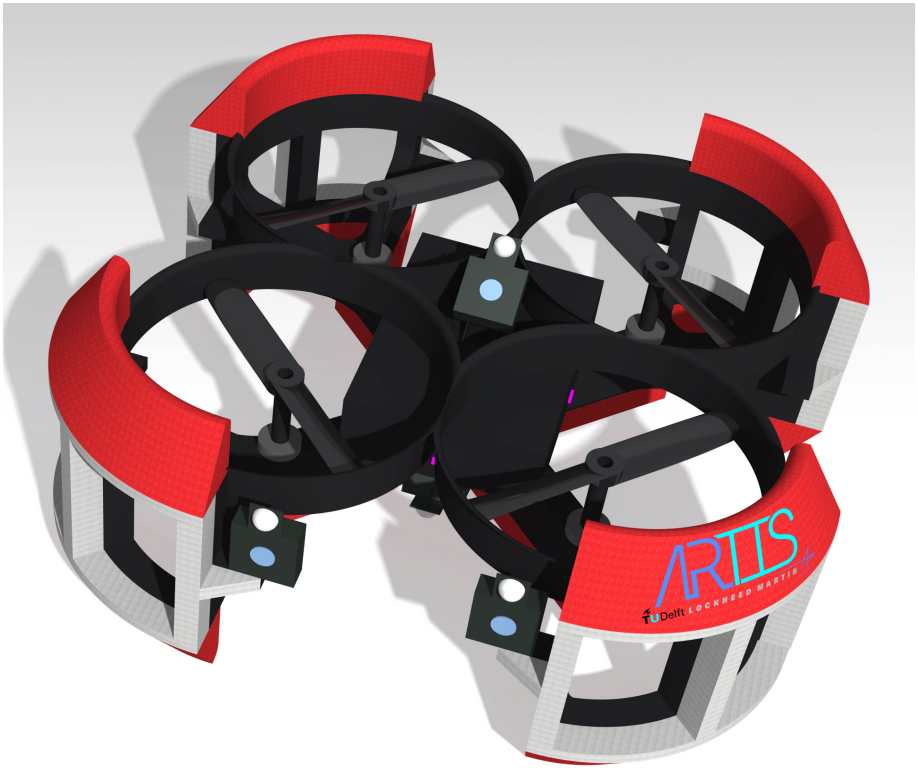


Figure 11.1: Top view render of the final ARTIS layout

Figure 11.2 shows an exploded view of the main compartment of ARTIS. Here, the internal layout of the body compartment can be clearly seen. The main body compartment was sized to be as small as possible to block the least airflow, but to still fit all electrical components. Furthermore, all important parts of the drone are indicated via arrows.

Figure 11.2: Exploded view of the inside compartment of ARTIS

Lastly, technical drawing of ARTIS is shown in figure 11.4. In this figure, the key dimensions of the drone are visualized to show the size of the drone. Please note that all dimensions are put in millimeters. Also, as the maximum ARTIS dimension, the diagonal, has a value of 285.35 mm, **AMD-NF-CNST-Size-01 (S,D2)** has been met [6].

11.2. Performance Analysis

The performance of ARTIS will be analyzed in two ways. First, the mass and power divisions of the final design are investigated. The main contributors to the final mass and power are stated to get an overview of ARTIS' most dominant components. After, the performance of the drone in the fuel tank will be elaborated on. The path in the tank is described and based on that the inspection time is computed to present a complete overview of the mission.

11.2.1. Mass and power division

The performance of this drone is analyzed by looking at the final mass and power divisions. Here, the mass and power consumptions are computed by adding all components stated in table 9.3 and table 9.6, as well as the propellers, the motors and the structure. The mass and power division of the final ARTIS design in presented in figure 11.3. For clarity, some components are grouped together in classes. For example, the CPU, the VPU, the IMU, the SD slots, the main circuit board and the connectors are grouped under the name 'motherboard'. The payload/sensors class includes the cameras, LEDs, and time of flight sensors. The power distribution consists of the PDB and the ESCs.

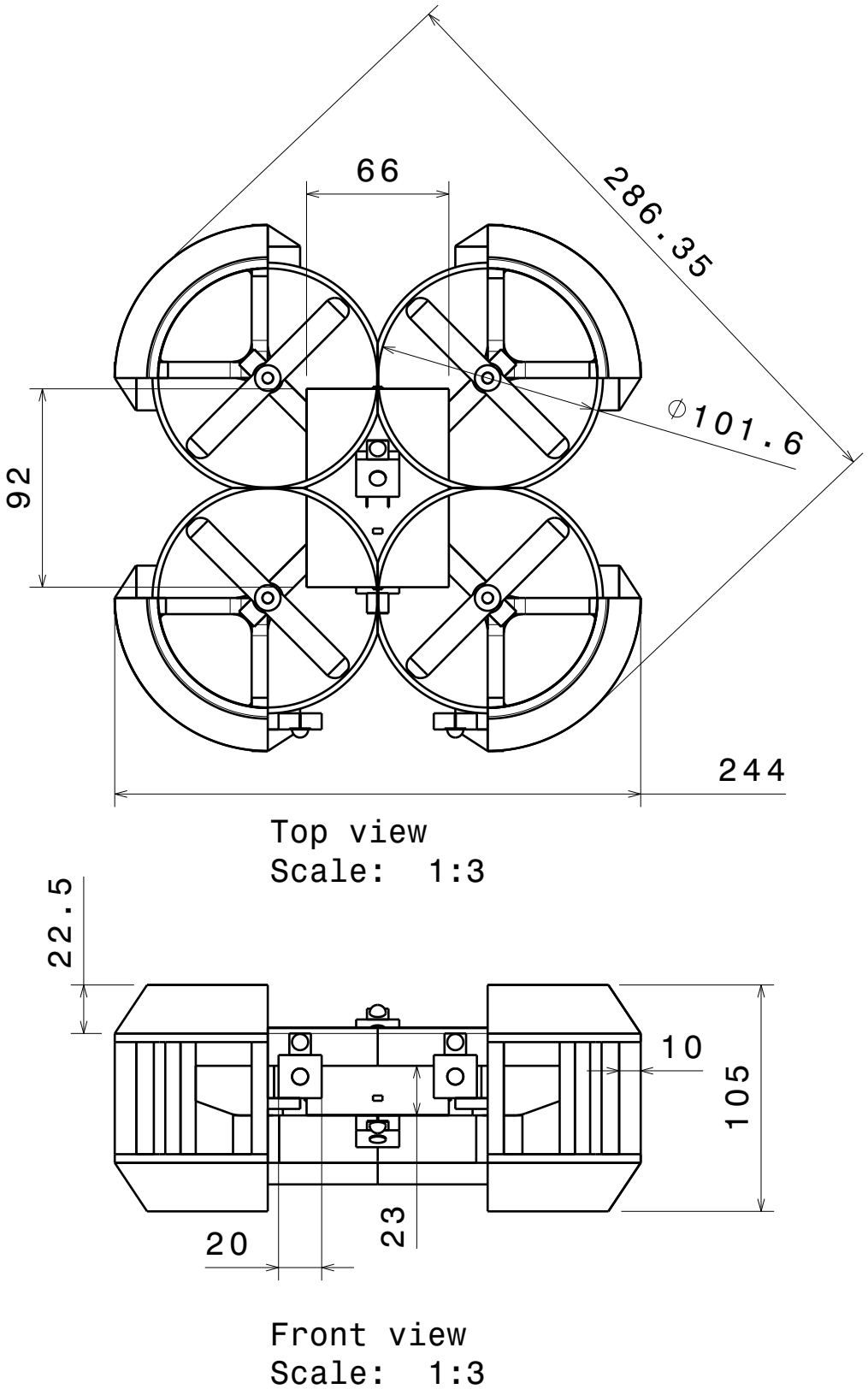


Figure 11.4: Technical drawings of the ARTIS to scale

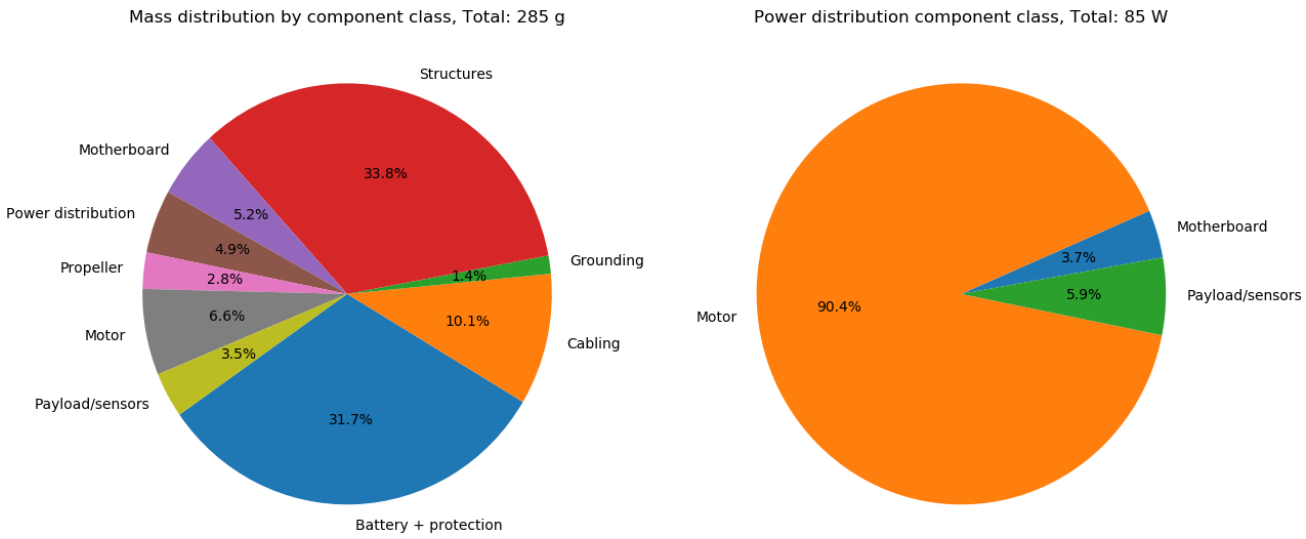


Figure 11.3: Mass and power distribution of ARTIS

As can be seen, the structure, battery and motors take up the largest part of the ARTIS mass with a combined share of over 65 %. The third biggest contributor is the cabling, taking up over 10 % of the total mass. This is due to not only the regular cabling stated in table 9.5, but also due to the cabling that is included in the motors and the ESCs. For this reason, the motor and power distribution weight turned out lower in this pie chart.

For the power consumption, the motors require almost all the ARTIS' power. The payload and sensors follow at significantly lower places, taking up 6 % of the power. Finally, the computations on the motherboard take up only 3.7 % of the total ARTIS power. With a total power consumption of 85 W and a battery with a capacity of 15.2 Wh, the total ARTIS flight time is computed at 640 s.

11.2.2. Inspection of the fuel tank

The fuel tank of the C-130 is divided in 6 parts. Each wing contains 3 separate sections in which fuel is stored: fuel cells close to the fuselage, the main fuel compartments around the middle, and a section in the wing tip. For path planning and fuel tank inspection, only the middle main compartment of the C-130 will be analyzed. This fuel tank compartment in shown in figure 11.6, including the ribs in this section with the locations of the holes. More on the operational environment is stated in section 13.2.1.

The actual inspection of the fuel tank as done by ARTIS will be done by following a path that has been set out beforehand. The path is structured between travel points (red circular points) and key points (orange stars). A travel point is a point where ARTIS navigates to to simply move through the tank. A key point on the other hand is a point at which ARTIS inspects the tank. Here, it performs a full 360° yawing rotation. This will be done by 10 intermediate yawing motions of 36° each. This is done to get some overlap in between consecutive images as the cameras have a 45° HFOV. The location of the key points was chosen such that each point in the tank would be sufficiently close to the cameras at some point so that a resolution of 5 MP/ft^2 would be met at any location. With the current division of 3 key points per section, the lowest obtained resolution is 7.3 MP/ft^2, meaning **AMD-F-PAYL-Cam-02 (S,Y)** is met. ARTIS is placed in tank at the top left at the curly arrow. It then follows the dashed orange arrows and passes by all key and travel points. It leaves the tank at the bottom right curly arrow. For the simulation, all key and travel points' location and orientation a specified with respect to a reference point, located at the top left in figure 11.6 (blue dot). For the pre planned path, the set up coordinates are structured as (x-coordinate, y-coordinate, z-coordinate, yaw angle, true/false). For each point to be visited, the coordinate gives the desired location, orientation and a true/false value indicating whether or not ARTIS should switch to the image mode and take high quality pictures.

Another significant detail for these key points is that one of the front facing cameras will temporarily be used for 13.2 MP pictures instead of video. Having three cameras take images at that resolution at 2 fps leads to a data rate that cannot be handled by the microSD storage. For this reason the cameras only take pictures

at 1 fps at key points, it does this after every 36° rotation. Stopping one of the cameras from generating a video feed has the consequence of having no stereo-vision. This means the navigation is not operational during one of these key-points, which is considered not to be problematic since the drone is supposed to be stationary. For this the IMU can be used in combination with the time of flight sensors. This means that for the short time in which it is only rotating it is not problematic. According to calculations done with the gain tuned control system, a angular rotation step of 36° takes 0.4 s. Including damping out the overshoot and taking 3 images, this becomes about 1.3 second. After each rotation of 36° degrees, the cameras take a picture and the rotation continues. A full rotation thus takes about 13 s. This means the actual camera fps is lower, at about 0.75 fps.

The total path length in the tank is computed by summing the distances between the key points, the distances in between a key point and a travel point, and the distances of flying trough a hole. This first distance is equal to $10 \cdot 2.558/4 = 6.396$ m. When flying from a key point to a travel point to align with the center of a hole, the distance was computed using $d = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2 + (z_1 - z_2)^2}$. This summed up to 2.425 m. Flying through holes ended up being 1.449 m. The total path length is computed at 10.27 m. The velocity at which ARTIS flies was estimated at 0.1 m/s. To be extra safe while flying through the holes, the velocity is halved to 0.05 m/s. These velocities are not to be confused with the maximum velocity for navigation described in equation (10.1), as this described the sideways velocity which does not happen during inspection. Using these velocities and also taking into account 12 rotations of 13 s each, the inspection time of this fuel tank turns out to be 273.2 s. As ARTIS has a flight time of 640 s, it can easily inspect this tank compartment. To find the total inspection time of all 6 C-130 fuel tank compartments, this computation can be extrapolated to obtain a reasonable estimate. All 6 fuel tanks consist of 24 sections in total. Therefore, as the tank in figure 11.6 has 4 sections, the estimated total inspection time for all 6 tanks is around 1639.2 s, or 27 minutes and 19 seconds. This is thus only the time required by ARTIS to fly inside all tank sections; the total time one complete inspection takes is elaborated on in section 13.2.2.

11.3. Sensitivity Analysis

To see how the design responds to a change in input, a sensitivity analysis is performed on the final design. As the design might still be subject to small changes, it is vital to know how the design responds to these changes. This analysis is performed by looking into the most obvious changes that might be included in the further design. These changes include the beneficial effect of ducts, the addition of a grounding solution and addition of additional components.

11.3.1. Benefits of ducts on hover performance

In the design of ARTIS, possible benefits of the ducts on aerodynamic performance are not taken into account. However, studies have shown that ducts around propellers can decrease the required power to hover by up to 50 % in theory [56]. This is a promising benefit to ARTIS, but not further researched due to time constraints. For this reason, for several decreases in power consumption the new performance is analyzed. The results are stated in table 11.1. It can be seen that the flight time clearly increases with the added benefits that ducts might bring. With a 38.1 % decrease in power consumption, ARTIS will even fly for 20 minutes.

Table 11.1: Drone performance with reduced power consumption

Power decrease [%]	**Power consumption [W]**	**Flight time [m:s]**
10	73	12:23
20	62	14:31
30	52	17:31

11.3.2. Addition of grounding solution

As stated in section 9.4, a grounding solution is implemented to prevent fuel tank explosions. The ARTIS design including this grounding solution is therefore analyzed by looking at the mass and power budgets stated in figure 11.5. It can be seen that the design now has mass of 291 g and consumes 87 W. The flight time reduced to 10 minutes and 22 seconds. The requirement regarding the total mass is still met, and the inclusion of a grounding solution is thus considered a viable option.

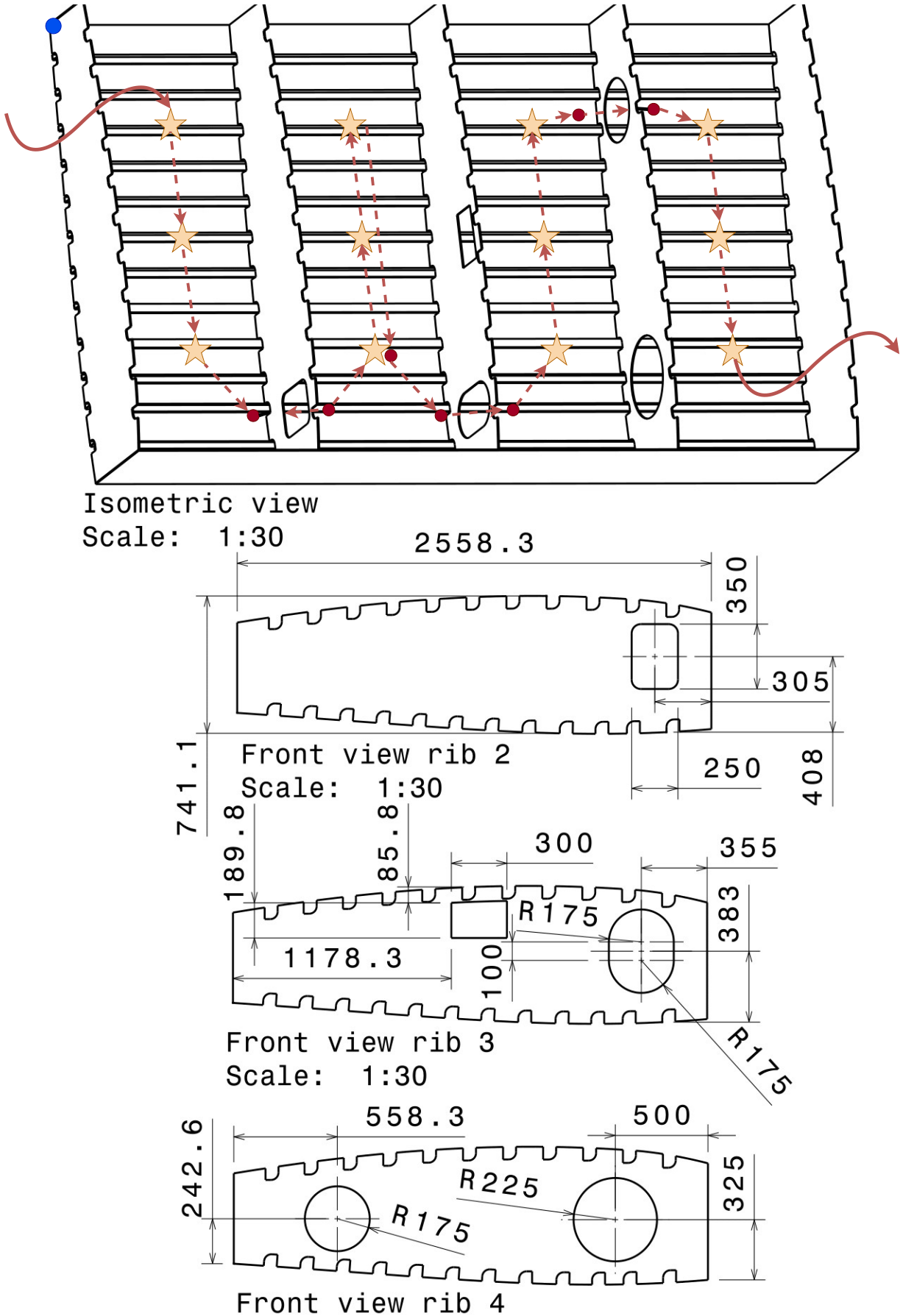


Figure 11.6: Path of ARTIS in the C-130 middle fuel tank including hole placement in the ribs to scale

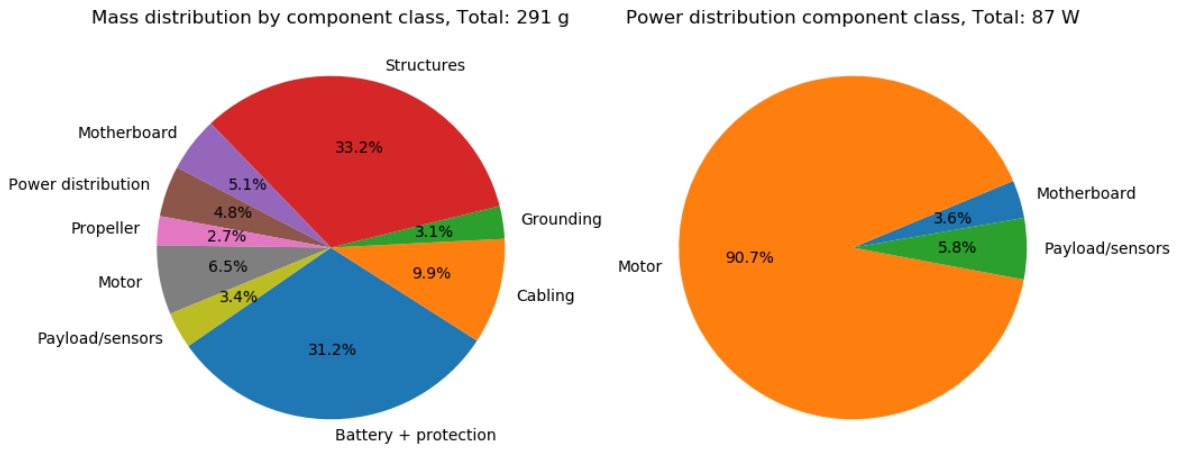


Figure 11.5: Mass and power distribution of ARTIS including a grounding solution

11.3.3. Overall design sensitivity

Finally, several design changes are analyzed for their effect on the ARTIS mass, power and flight time. The design changes that are analyzed are adding mass, adding power consumption and reducing the battery capacity. The changed values for those aspects were computed such that the drone would meet both the 10 minute flight time requirement and the 300 gram mass requirement. The results can be seen in table 11.2. It can be seen that the design would still just meet the requirements with either a mass increase of 12 g, a power consumption increase of 5.7 W or a battery capacity decrease of −6.3 %Wh. These values show that the design can still be tweaked a bit while still meeting both requirements. However, it does show that the final design total mass contingency of 5 % does not translate to meeting the requirements directly. According the the 5 % contingency, 15 g could still be added to the drone to still meet the 300 g requirement. The sensitivity analysis however shows that only adding 12 g already leads to a violation of the 10 min flight time requirement. The actual contingency on the design is thus 12 g and not 15 g. This leads to a stricter margin for adding more mass to the final design and still meeting the requirements. On the other hand, as no benefits of adding ducts is taken into account at this point, no problems regarding requirement violation is expected.

Table 11.2: Required changes to just meet the requirements

Design change	Changed value	Mass [g]	Power consumption [W]	Flight time [min:s]
Adding mass	+12 g	297	90	10:00
Adding power	+5.7 W	285	90	10:00
Reducing battery capacity	−6.3 % Wh	285	85	10:00

11.4. Final Design: Resource and Budget Breakdown

For the final design, the resource and budget breakdown is analyzed to check how to technical performance parameters developed over time. As stated in table 7.1, the contingencies for all subsystem masses, flight time and hardware cost is 5 %. How these parameters have developed until the final design is presented in table 11.3.

Table 11.3: Final design target value compliance

Value/Contingency	Total Mass	Structural Mass	Payload Mass	Battery Mass	Propulsion Unit Mass	Cabling Mass	Flight Time	Hardware Costs
Design specification value	300 g	80 g	28 g	113 g	73 g	7 g	600 s	€2500
Final design stage cont.	5 %	5 %	5 %	5 %	5 %	5 %	5 %	5 %
Final design tar. value	285 g	76 g	26.6 g	107.35 g	69.35 g	6.65 g	630 s	€2375
Final design ac. value	285 g	97 g	9.5 g	88 g	32.52 g	28.65 g	640 s	€1434
Compliance tar. value	0.0%	27.6%	-64.3%	-18.0%	-53.1%	330.8%	1.6%	-42.6%

In the first row, the design specification value is mentioned. This value represents the desired value that the parameters should have when the design is completely finished. In the second row, the contingency for the current design phase, the final design, is presented. This value is obtained from table 7.1. The target value for which the final design should be designed is presented in the third row. The fourth row then shows the actual value of the parameter after the final design. The final rows gives an overview of whether or not the actual value of a parameter is above or below the target value. If this value is beneficial to the mission, so a lower mass and cost, and higher flight time, the cell is marked green as this is desired. Please note that for the final design actual value, the masses of the subsystems shown in the table do not add up to 285 g, as during the design more subsystems were defined. For example, the electronics in ARTIS are not stated in table 11.3, while they of course also take up some mass.

As can be seen in the table, the structural mass and cabling mass have exceeded their target value for the final design phase. This is due to two main reasons:

1. During the design of the final drone, it was found that the drone should consist of more parts and components than initially thought off. For example, during the midterm design phase, components such as the PDB, motherboard connector, aluminum grounding coating and BPC were not considered at all. During the final design, more knowledge on drone design was obtained and that led to the addition of several components. This caused a different division in total mass than initially expected.
2. The design specification values (row one of table 11.3) were determined during the conceptual design phase of the drone. At this point, too little knowledge was present to obtain accurate specification values. As the design progressed to the final stage, more insight is drone mass and power division was obtained. For example, it was found that around 100 g of structures is a more reasonable estimate than the 80 g that was initially assumed.

These two reasons explain why some parameters did not meet the target value. Please note that for the cabling, also the cabling that is delivered with the motors and the ESCs is included in the 28.65 g. This inclusion also causes the cabling actual value to be higher than the target value.

To see how the total mass, flight time and hardware cost contingency compliance has developed over time, figure 11.7 was constructed. This figure is a continuation of figure 7.2. The values of total mass, flight time and hardware cost of the final design (11 weeks) have been added.

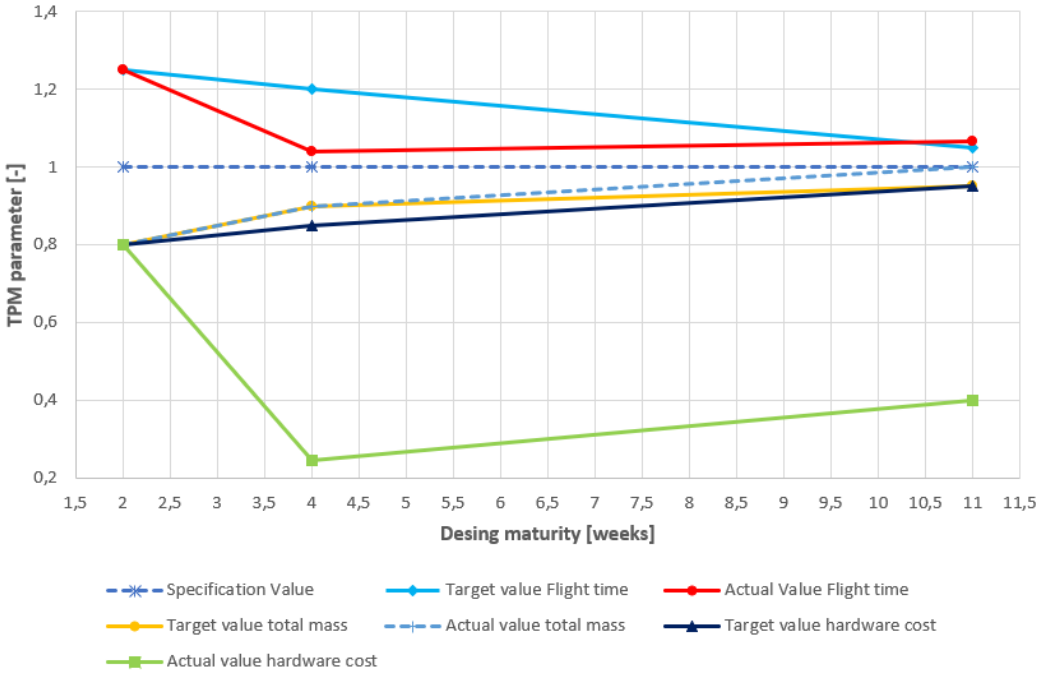


Figure 11.7: Contingencies on total mass, flight time and hardware cost for the final design

In this figure it can be seen that the actual flight time value for the final design exceeded the target value. This thus means that when looking at the flight time, the drone is well designed and no issues arise for in this area. The added flight time compared to the midterm is mainly due to the higher capacity of the selected battery. For the total mass, the actual value and the specification value were identical for the midterm phase. For the final phase, both values are again the same. In further design, the total drone mass should thus be carefully monitored so that the 300 g requirement is not exceeded. However, as the actual total mass value is does not yet cross the specification value, no rigorous actions should be taken. The contingency on the total mass of ARTIS was used in the design process to select components that would ensure that the final mass stays below 285 g at this stage. Finally, the hardware cost was considerably lower than the target value at the midterm stage. For the final design, this still is the case. For the hardware cost for the final design the total system cost (production and assembly), as will be elaborated on in figure 14.2. This value is around €1461 , which is still lower than the upper limit of €2500 This leaves room for either further improvement of the design, or a more profitable operation for both the selling company as the user.

12
VERIFICATION AND VALIDATION

It is imperative that the design that has been presented from chapter 8 through chapter 11 is thoroughly verified and validated to ensure that the predicted performance is representative of reality. This chapter presents the verification and validation of the established design.

12.1. GUIDANCE, NAVIGATION, AND CONTROL

This section presents the verification and validation efforts that have been executed for guidance, navigation, and control. Verification for guidance, navigation, and control were carried out separately, using e.g. unit testing and system testing. The validation is done for the entire GN&C subsystem at once by running the guidance, navigation, and control software simultaneously in the simulation.

12.1.1. VERIFICATION

As mentioned before, this subsection highlights the separate processes that were undertaken for the verification of navigation, guidance and control.

12.1.1.1. NAVIGATION

Section 10.3 discussed the framework of autonomous navigation. Here, certain characteristics of the framework shall be verified.

Firstly, the verification of feature detection algorithm is discussed. Most of this algorithm is composed of functions from the open source library OpenCV. This is a popular library among academics and professionals, and its functions have been used and tested thoroughly. Therefore the functions themselves shall not be verified or validated, instead their implementation will be discussed.

The feature detection code was tested with a variety of input images, both of wing-box structures and other objects such as to test the robustness of the algorithm. Since certain parameters in the algorithm need to be tuned for specific scenarios, the outputs varied in accuracy. Three test examples can be seen in figure 12.1. It can be observed that most lines are accurately detected, nevertheless, the right most most image shows a lot of noise, this can be improved by changing the parameters for the lined detection and edge detection functions within the algorithm.

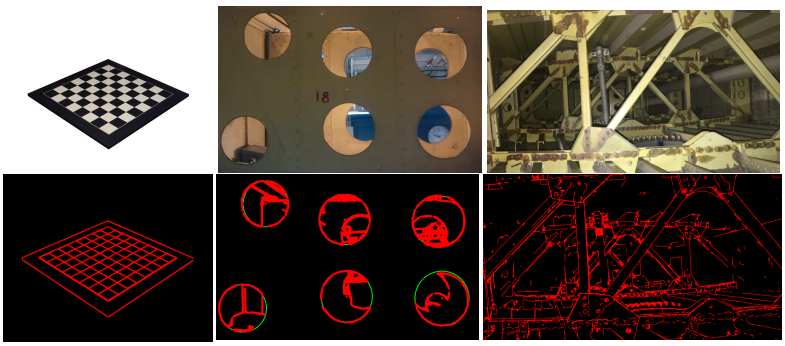


Figure 12.1: Line detection verification tests

While verification of the feature detecting algorithms was done using still images, the algorithms are validated using actual video footage from the fuel tank of a Dakota aircraft wing. This fuel tank contains features that are extremely similar to those found in the C-130 fuel tank, namely the rib truss structures found continuously along the wing. The videos were performed using a CMOS camera with led lighting. Furthermore shaking, sudden movements and blurry views were simulated in order to check the performance of the algorithm in a realistic scenario. Different lighting conditions were also analyzed in order to determine the

influence on the accuracy of the algorithms. Figure 12.2, figure 12.3 and figure 12.4 exemplify the performance of line detection methods under three different lighting conditions. It can be seen that direct and strong lighting is beneficial, but that even under poor lighting conditions, lines can still be detected.

Figure 12.2: Direct lighting

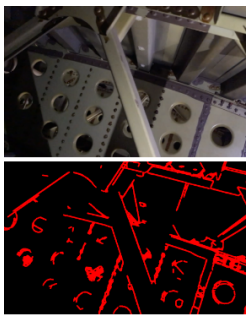


Figure 12.3: Low lighting

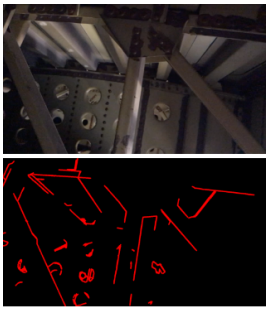


Figure 12.4: No lighting

Overall the performance shown in the video testing was positive. Detection did fail when the camera movements were sudden and fast, nevertheless the IMU is used to mitigate the effect of these movements on the navigation algorithms.

A second test was performed in order to verify the possibility of feature matching and tracking on the fuel tank structure. The algorithm used was the Lucas-Kanade Optical flow methods from OpenCV [33]. This method is based on the detection of corners as opposed to lines or edges, thus it is different form the proposed solution. Once again it ran real time with video recordings at 30 fps on a laptop equipped with a Intel Core M-5Y51 CPU (2 cores) at 1.10 GHz. Nevertheless it verifies the possibility of tracking features in the target environment. Figure 12.5 shows an image of the tracking of features in the fuel tank for a few seconds. The overall direction the tracking method showed was similar to the actual camera motion (right transverse motion) with the exception of outliers showing vertical displacement instead. A final observation that can be made from the tracking tests, is that corner features would not have been a suitable choice for the target environment. The tests showed that after a small time period, the estimated direction diverges a lot from the actual motion and becomes very chaotic. This is a reasonable outcome since corners are very ambiguous in the environment, with a high density of corners close to each other, that can be easily mismatched. Furthermore, due to depth differences, some corners are detected between overlapping features, the detection's are then constrained by the overlapping depth features and move erroneously. Overall, edge detection's (which are used for the detection of lines) should be a more appropriate choice.

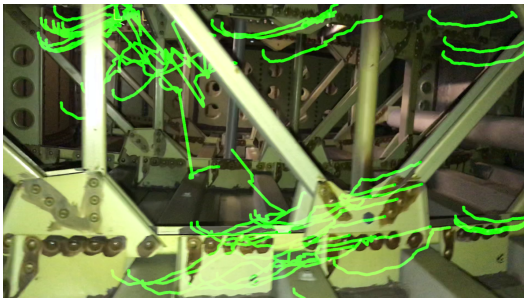


Figure 12.5: Lucas-Kanade optical flow feature tracking method

12.1.1.2. GUIDANCE

The verification procedures for the guidance solutions proposed in section 10.4 aim at checking the correctness of the codes and establishing the MATLAB programs are computing the intended outputs.

The first performed activity consisted of fixing the codes and debugging the programs. Syntax errors were checked running the MATLAB code files in the Editor. Early warnings and errors messages about the code were thus spotted, and the files were modified based on these messages. Syntax errors were also easily identified via the compiler.

Having detected and fixed the easy bugs in the code, a more thorough analysis was then performed. The files presented in figure 10.13 were graphically debugged using the MATLAB Debugger functionality [155]. In particular, unspotted issues were arising when running the *compute_path* file. Via the mentioned MATLAB

feature, breakpoints were set to pause the file execution and examine the chosen portion of the code where the problems were expected to originate from. After setting the breakpoints and running the file from the Command Window, MATLAB paused at the first breakpoint in the program, highlighting whether the error occurred or not. With this approach, most issues were resolved.

The next performed procedure aimed at testing the functions by creating script-based unit tests. Each unit test analysed the different outputs of the MATLAB functions, ensuring their correct functionality according to the intended purposes. Following MATLAB guidelines to generate efficient script-based unit tests [156], the *assert* function tested the different chosen conditions, throwing an error in case the testing did not run to completion. After fixing the bugs spotted via the mentioned method, the output of the unit testing procedure, as applied to the *compute_map* function, was as shown in figure 12.6.

Name	Passed	Failed	Incomplete	Duration	Details
'compute_mapTest/compute_mapTest'	true	false	false	0.017248	[1×1 struct]

Figure 12.6: Output of script-based unit tests for chosen MATLAB function

The same outcome was obtained for all the functions being defined (see figure 10.13 for reference).

Having assessed no programming errors were present in any of the written programs, the subsequent verification activities aimed at checking the codes were computing the right thing and with which level of accuracy. Section 10.4.2 outlined how margins had been added in the occupancy map, so as to mitigate risk **C5** (inaccurate path planning and obstacle avoidance). Different margin values were used to check whether the map discretization and A^* algorithm would still perform as intended. With the virtual environment used throughout section 10.4, the maximum margins that could be added to each obstacle, before the algorithm wouldn't converge to a solution, were 7 cm, hence higher than the voxel resolution of 4 cm. This result is strictly dependent on the environment in which the algorithm is run, thus it shall not be taken as a final reference value. The main conclusion that can be retained is that with the obtained margin being in the range of expected values, the programs are not wrongly coded and produce reasonable results.

The next illustrative test being executed dealt with the consequences of false obstacles detection. The occupancy map, storing information on each voxel in the form of a boolean value, was modified so as to mimic an inaccurate obstacle recognition. The initial out-of-range results (as much as 20 %!) suggested a wrong implementation of the ARTIS size constraint, which was subsequently fixed. The effect of having more obstacles being detected than actually present in the environment could then be studied. Randomly changing the voxel occupancy values anywhere along the map resulted in differing results, since a false obstacle in front of a small interstice would be enough to avoid converge of the A^* algorithm when including margins and the ARTIS size constraint, whereas an an extra obstacle in an area already discarded from the path would have no consequence. Although heavily dependent on the environment, for the one used throughout section 10.4 it was computed that, on average, around 2 % of false obstacles could be added before the algorithm would be incapable of planning the path. The effect on the planned path is shown in figure 12.7.

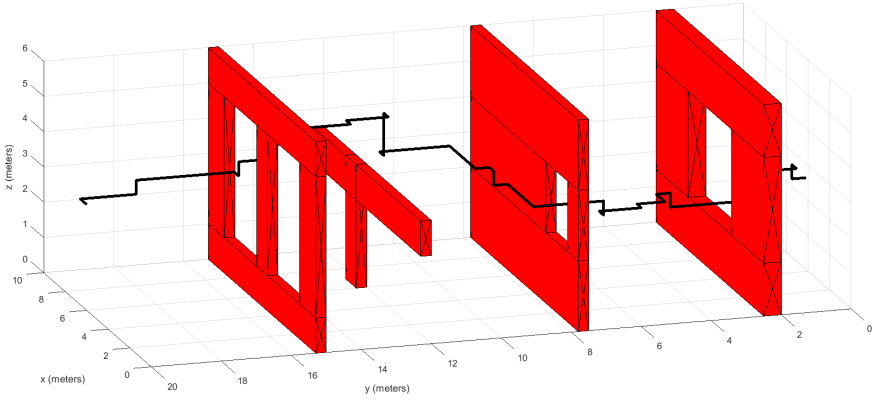


Figure 12.7: Rough path-planning in an environment with 2% false obstacle detection

As can be observed, a rougher path is being planned, since the ARTIS is detecting and avoiding false obstacles which are not actually present in the environment. The obtained value (i.e. 2 %) is not to be considered as a reference estimate since it is an average extremely dependent on the environment in which ARTIS operates, but the conclusion that can be drawn from the present discussion is that the algorithm does perform as expected, with a correct inclusion of the ARTIS size affecting the choice of path. Furthermore, the elapsed time substantially increased due to false obstacle detection (to around 8 s for 2 % false detections, whereas it was lower than 1 s for a completely accurate detection), hence confirming the reasonableness of the obtained results and the potential correctness of the MATLAB programs.

To further verify the MATLAB programs perform as expected, it was chosen to randomly shuffle the voxels with a pre-specified percentage. An inaccurate navigation system would incur in both missed and false detection, and it is thus illustrative to study the implications on the guidance solutions. Of all the nodes in the grid environment, on average around 1 % of them could be moved and randomly rearranged before the ARTIS would collide with an obstacle. This is visually shown in figure 12.8.

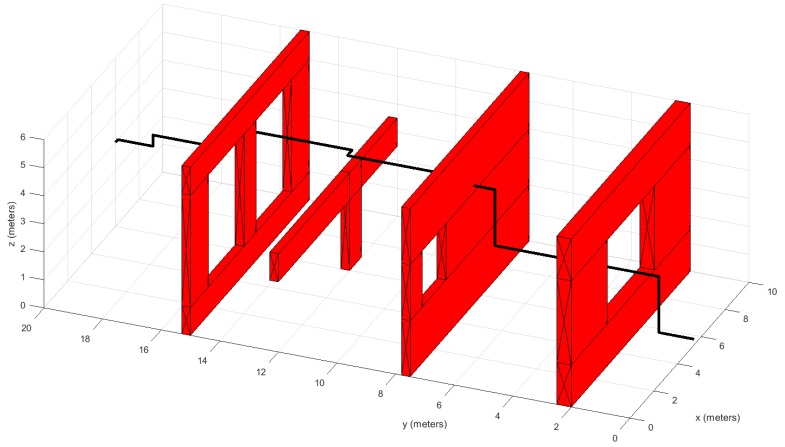


Figure 12.8: Missed obstacle detection and collision after shuffling 1 % of grid node values

ARTIS fails to detect the obstacle and collides into one of them. As expected, the A^* algorithm is unaware of its presence and plans that which it believes is an optimal collision-free trajectory. An average value of 1 % is deemed reasonable given the environment being used, further strengthening the conviction that the developed MATLAB codes are free of errors and fulfill their intended purposes with satisfying performance.

12.1.1.3. CONTROL AND STABILITY

As mentioned in section 10.5.2, the quadcopter model in the Matlab aerospace blockset provides a nonlinear model of a Parrot minidrone. This is in turn modified and expanded to a linear model for ARTIS with the purpose of tuning the six PID controllers. Verification for the control system therefore consists of two distinct verification processes, namely model verification and performance verification. The verification of the nonlinear model has already been confirmed based on the Parrot minidrone, whereas the newly constructed linear tuning model requires additional verification. This can be achieved by performing both unit tests and system tests. The nonlinear model is then used to verify the performance of the linear model, and as such the performance of the designed controller. This verification process for the control system is visualized in figure 12.9 [4].

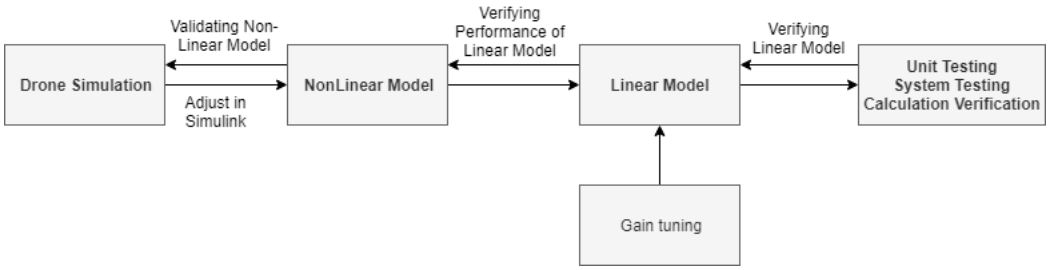


Figure 12.9: Control and stability verification process [4]

Verification of the linear model can be performed by first testing the constituent units of the model, which is also known as unit testing [157]. The linear model in Simulink consists of many blocks code, which are each verified independently. Fortunately, Simulink represents these pieces of code as actual blocks and arrows, making it easier to verify. It was decided to conduct the following unit tests, where the other controllers are turned off during the execution of a unit test.

1. The reference altitude is set to zero meters while adhering to the regular controller gains, as is shown in figure 12.10, after which the response of the altitude controller is monitored. As the initial value provided to the controller is zero meters (i.e. ARTIS is sitting on the ground), the altitude controller response should output a zero altitude command, thereby maintaining an altitude of zero meters. It can be seen from figure 12.11, which shows the altitude response to the altitude controller shown in figure 12.10, that this is indeed the case, and the altitude controller unit is therefore considered verified.

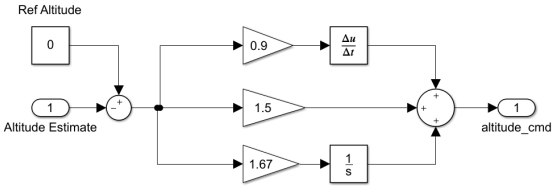


Figure 12.10: Simulink PID layout for the altitude unit test

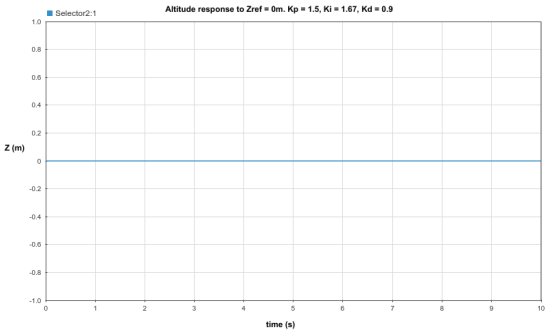


Figure 12.11: Altitude controller response for a reference altitude of Z_{ref} = 0m and an initial value of Z_{init} = 0m

2. The yaw controller gains are all set to zero while a step input of 0.3rad is provided as yaw reference. Logically, as the controller is effectively turned off by setting the gains to zero, there should not be any yaw response in this scenario. The actual yaw response is displayed in figure 12.12, which confirms the expected behavior and therefore verifies the yaw controller.

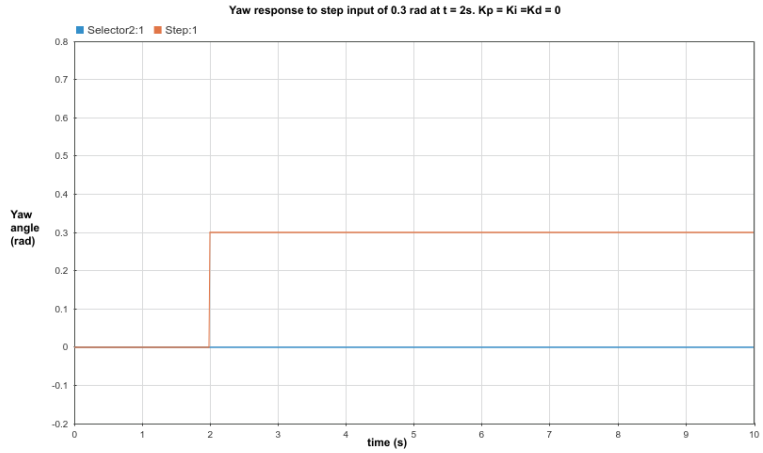


Figure 12.12: Yaw response for the proposed yaw controller unit test $(K_p = K_i = K_d = 0)$

3. The same step input is provided as reference to both the pitch and roll controllers. Since the drone is symmetric with respect to the pitch and roll axes, the responses should approximately be the same as well. Again, this behavior is confirmed by figure 12.13 and figure 12.14 as well as by table table 12.1, which shows the rise time and overshoot of the roll and pitch responses to further visualize the similarities. Evidently, the differences between the numerical values for the rise time and overshoot can be considered negligible, and verification of the roll and pitch controllers has thus been confirmed.

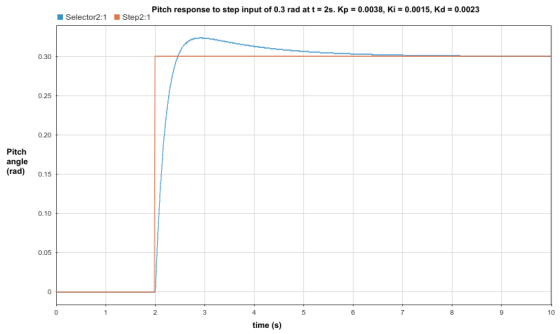


Figure 12.13: Pitch response to a unit step input of 0.3 rad at t=2 s

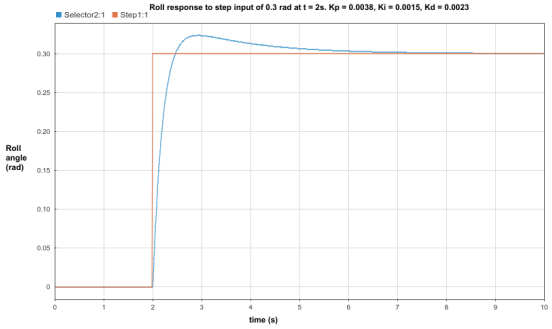


Figure 12.14: Roll response to a unit step input of 0.3 rad at t=2 s

Now that the individual blocks of code work and output the signals as desired, they are combined into a subsystem, which ought to verify that the outer loop controllers, which control position in the horizontal plain, work in coherence with the inner loop controllers. Evidently, testing whether the outer loop controllers function as desired while the attitude and altitude controllers are active and controlling their respective states effectively tests the functionality of all the unit blocks and as such of the entire control system. Providing a change in the reference position and monitoring the position response can therefore be used as a system test for the linear model.

Table 12.1: Rise time and overshoot for the roll and pitch unit test responses

	Pitch response	**Roll response**
Rise time [s]	0.311	0.309
Overshoot [expressed as % of final value]	107.80	107.83

In doing so, it was decided to provide consecutive step inputs of 1 m in x- and y position, as a result of which the reference position is first updated to $(x, y) = (1, 0)$ and then to $(x, y) = (1, 1)$, provided that the reference position at the start is $(x, y) = (0, 0)$. Since a right-handed coordinate frame is used this implies that the drone in the world frame first moves right and then moves forward, all the while simultaneously maintaining attitude and altitude.

The resulting position response is displayed in figure 12.15, where the step input in x-direction is provided at $t = 2$ s and the step in put in y-direction is provided at $t = 5$ s. From figure 12.15, it is visible that the position responses track to the updated reference position without overshoot, and the system test thus verifies the outer loop controllers and thereby the control system in the linear model.

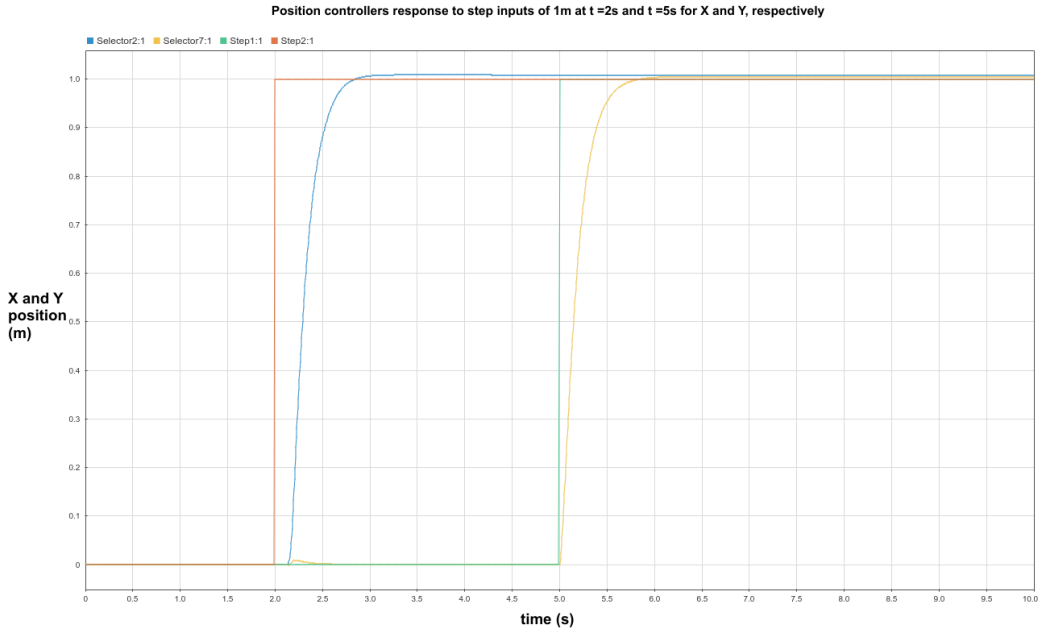


Figure 12.15: Responses of the outer-loop position controllers to step inputs at $t = 2$s and $t = 5$s

The performance of the linear model can be verified by analyzing the characteristics and differences of controller responses in the linear model versus the nonlinear model when a step input is provided.

From figure 12.16 and figure 12.17, which show the yaw responses to a step input for the linear and nonlinear models, differences in most prominently the overshoot and settling time can be noted. Nevertheless, the trend of the response for yaw appears to be mostly similar. The fact that the trend of the response is similar was also apparent for other controllers, but these responses are not shown here for brevity.

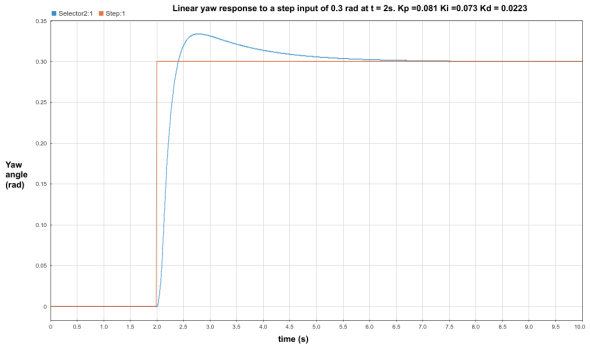


Figure 12.16: Yaw response in the linear model to a step response of 0.3 rad

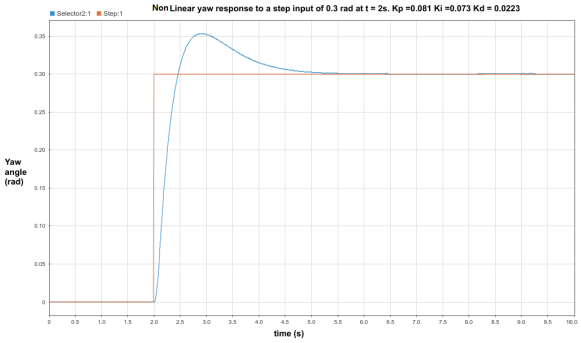


Figure 12.17: Yaw response in the nonlinear model to a step response of 0.3 rad

Instead of the showing the responses, however, the differences in step response characteristics can be analyzed to clarify the differences in the responses between linear and nonlinear model further. The step response characteristics that were analyzed comprise the following:

- Delay time, i.e. the time it takes until 10% of the final value is reached.
- Rise time, i.e the time it takes to advance from 10% to 90% of the final value.

- Settling time, i.e. the time it takes for the response to remain within 5% of the final value.
- Overshoot, i.e. how much the response overshoots the final value.

The step response characteristics for the six PID controllers in the linear model versus the nonlinear model are presented in table 12.2 and table 12.3. It should be noted that pitch and roll display exactly the same response characteristics as a result of symmetry, and as such the responses to a step input for x and y positions is the same as well.

Table 12.2: Rise time and delay time characteristics for the linear model versus the nonlinear model

	Rise time [s]			**Delay time** [s]		
	Linear model	**Nonlinear model**	**Error (%)**	**Linear model**	**Nonlinear model**	**Error (%)**
Altitude	2.029	2.412	15.88	0.128	0.198	54.69
Yaw	0.486	0.307	-58.31	0.069	0.073	4.17
Pitch and roll	0.280	0.249	-12.45	0.057	0.059	3.51
(x, y) **position**	0.438	0.513	17.12	0.127	0.165	29.92

Table 12.3: Overshoot and settling time characteristics for the linear model versus the nonlinear model

	Overshoot [% of final value]			**Settling time** [s]		
	Linear model	**Nonlinear model**	**Error (%)**	**Linear model**	**Nonlinear model**	**Error (%)**
Altitude	100	100	0	3.523	3.968	11.21
Yaw	111.3	117.7	5.44	1.917	2.256	15.03
Pitch and roll	110.87	113.87	2.71	1.810	1.261	-43.53
(x, y) **position**	100	100	0	0.729	0.756	3.70

It can be noted from table 12.2 and table 12.3 that differences between the linear and nonlinear model are apparent. As this model is tuned to match the overshoot as closely as possible, the most notable errors are for the altitude response delay time, the pitch and roll response settling time, and yaw response rise time. It should be assessed further whether the current gains still result in acceptable responses for the nonlinear model. For the yaw rise time and pitch and roll settling time, however, the error of the nonlinear model with respect to the linear model is negative, as a result of which the error of the nonlinear response decays faster than the linear response. It can thus be concluded that the nonlinear response is actually beneficial compared to the earlier found linear response.

As can be seen from the tables, there is tuned for as little as possible overshoot for position and altitude. Notably, the result is that it goes at the cost of increased delay time. This is acceptable, as a fast response in x and y is not paramount for the safety of the drone. A slow response in x and y does not increase risk of hitting obstacles, and the induced battery drainage by the extra time needed is negligible. This extra time is above all accounted for in section 11.2.2. Also, regarding the error between the linear and nonlinear, the greater delay time in the nonlinear model is not problematic enough to require a nonlinear updated tuning scheme for the PID gains.

From the above, the performance of the linear model for the gain tuning process is deemed sufficient with respect to the nonlinear model, which therefore allows the predetermined controller and corresponding PID gains to be used in the simulation and possible real-life operations.

12.1.2. VALIDATION

Since it is outside the scope of the DSE to build our final product the closest one can get to a full systems level validation is to run a simulation. The simulation is described in section 10.6. In the following is a description of the validation that was performed using the simulation.

12.1.2.1. CONTROL

The control algorithm as exported from the Simulink model using the C Coder was run inside the simulation. This allows validating its performance under the normal operating conditions of the drone.

By running the control algorithm in the simulation we were able to visually inspect the stability of the drone, while running it on a non-linear model that was developed separately to the one used to check performance in section 12.1.1.3.

Furthermore, the linearization of the drone model was checked for validity under normal operating condition the drone would encounter inside the tank. In section 10.5.1 it was established that for pitch and roll angles under 10° the model is sufficiently linear. During a test run of the drone in the simulation the maximum roll and pitch angles were 3.66°, well within the range of linearity.

Another aspect that is validated is the algorithms ability to deal with measurement errors. The noise density for the angular rates is given as $0.07\,\frac{\text{deg}}{\text{s}\sqrt{\text{Hz}}}$. This gives a value of $\sigma^2 = \sqrt{\int_0^{255\,\text{Hz}} \left(0.07\,\frac{\text{deg}}{\text{s}\sqrt{\text{Hz}}}\right)^2 \text{d}x} = 0.139676^2\,\frac{\text{deg}}{\text{s}}$ for the standard deviation of the angular rates (using a value of 255 Hz for the Bandwidth). By the same approach a value of $\sigma = 0.2397419\,\frac{\text{m}}{\text{s}^2}$ was determined. With these values added to the actual correct values for angular rates and linear acceleration the drone is not noticeably disturbed (maximum disturbance measured was 0.9 mm). It has to be noted however, that the IMU noise is not gonna be the biggest source of disturbance. Especially the aerodynamic disturbances such as non-steady flow and interactions with the walls are notable.

Besides running step inputs in Matlab, visually inspecting that the overshoot is small enough to not cause any problems in the fuel tank is of great importance.

12.1.2.2. NAVIGATION

Running the MSCKF-VIO algorithm on the stereo images created by the simulation gives valuable inside into the feasibility of performing stereo VIO in the environment of the fuel tank.

One concern of running VIO inside the fuel tank is whether or not enough features will be detected. As can be seen in figure 12.18 the algorithm is finding numerous features in the stereo pair. As can be seen not all of the detected corners are correct though. Some of the corners that are detected are just the intersection of two lines at different depth. This is the main reason why the algorithm that is planned to be implemented on ARTIS is using lines instead of corners.

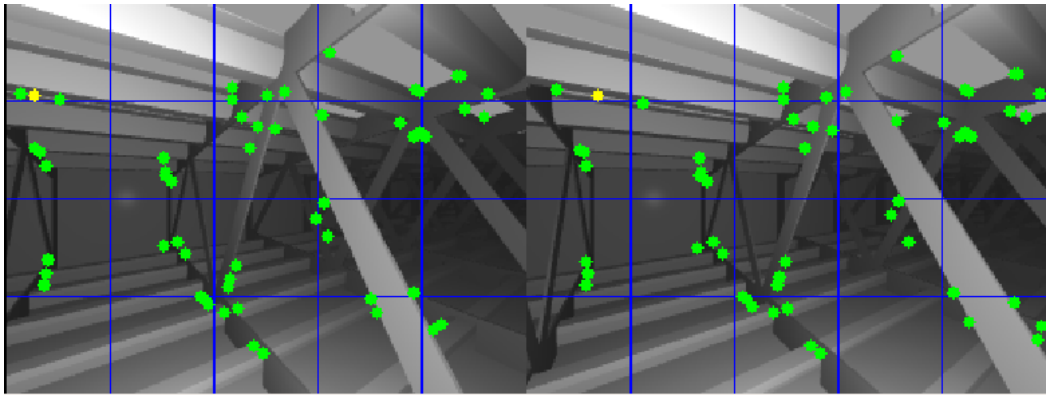


Figure 12.18: Stereo features inside the simulation

Another aspect to validate is that the camera field of view chosen is big enough to successfully find and match enough features. From figure 12.18 it can be seen by inspection that the FOV is big enough even with objects that are within decimeters of the drone.

12.1.2.3. GUIDANCE

Time constraints have not allowed implementing the guidance solutions, explained in section 10.4, into the 3D simulation (see section 10.6) whose purpose was validating the GN&C subsystem. This section aims at giving the reader an overview of future approaches to validate the guidance models.

The first approach to validation will be implementing the guidance programs into the mentioned simulation, assessing whether the models match the expected outcomes within decided accuracy requirements. Aiming at checking the simulated data with real-life experiments, the next step would consist of implementing the proposed algorithms onboard an existing drone. Possible deviations between the computational outcomes of the simulation and the experimentally observed data will then be assessed, minimized and resolved.

The codes shall be tested in several environments, one resembling the C130 fuel tank and one differing, assessing the robustness of the guidance solution. Changing the environment, domain and obstacles in a real-world scenario would be a good validation procedure of the developed programs.

Furthermore, a final experimental test could be inputting a wrong path planning in the guidance codes, to check whether the A^* algorithm would recognize the infeasibility of the proposed path and still effectively avoid the encountered obstacles.

12.1.2.4. General Design

There are 3 aspects relating to the general design that were validated using the simulation. These aspects are whether or not ARTIS fits through the holes inside the wing, whether the combined field of view of the cameras is big enough, and whether ARTIS can inspect all parts of the fuel tank.

Firstly, the drone does fit through all holes in the simulated fuel tank with enough margin to safely pass them. Secondly, the combined field of view of the cameras enables inspection of floor, ceiling, and walls. Thirdly, all relevant parts of the fuel tank are accessible by the drone for inspection.

12.2. Structures & Materials

In this section the frame and propeller design verification processes undertaken are presented. Furthermore, the necessary tests required to validate the designs are outlined.

12.2.1. Frame design verification

The airframe design verification process is broken down into code verification and numerical model results verification.

12.2.1.1. MATLAB tool code verification

The code verification has been performed by checking that the results for all cases satisfy equilibrium of forces and of moments. Furthermore, for symmetric loading cases, the results were checked to satisfy symmetry.

12.2.1.2. MATLAB tool results verification

The structure designed with the matrix methods implemented in MATLAB are verified using Ansys static structural analysis. Design verification is especially necessary in this case as the frame analysis method is not capable of calculating stress concentrations. Furthermore, the frame approximation assumes that the beam elements are slender, which is not an accurate description especially for members BC and BD from figure 8.11. The join area between members B-C and C-CP, and members B-G and G-GP has a significant influence on the stress distribution which can not be modelled as a frame. Finite element analysis is necessary to verify that no premature failure will occur locally due to a stress concentration. A very fine mesh of tetrahedral elements with a side lengths of 0.5 mm was used in order to be able to observe the occurring stress concentrations,

Figure 12.19: Fine meshing with tetrahedron elements with side of 0.5 mm

as in figure 12.19. For coarser meshes the value of
stress at concentration points would be averaged over a larger area, thus lowering the calculated concentration value. The bodies were combined with boolean operations in SpaceClaim before analysis in order to eliminate the numerical errors associated with inter-body contact. This unification is realistic as the entire structure will be manufactured by 3D printing in one single process. No bonds will be present, so no contact definitions are required.

In order to simplify the discussion, the "node" nomenclature from figure 8.11 is maintained, with the mention that the real structure does not have nodes. The following critical load cases have been considered, with the forces determined in section 8.3:

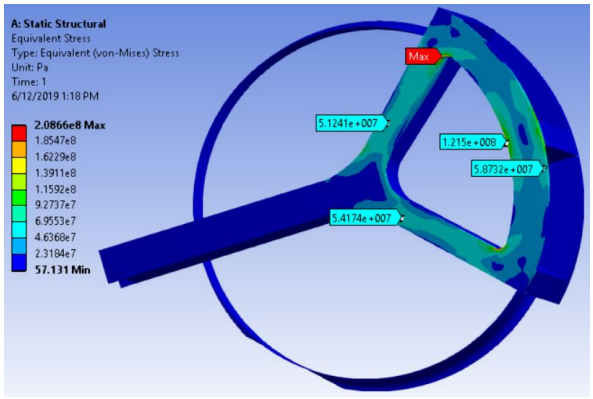
1. 300 N force normal to surface at node E. This load is critical for the upper arch structure, i.e. the structure approximated by nodes C, D, E, F, G.
2. 300 N force normal to surface at node EP. This load is critical for the upper arch structure, i.e. the structure approximated by nodes CP, DP, EP, FP, GP.
3. 300 N force normal to surface applied next to node CP, on top arch. This load is critical for members C-CP, B-G an A-B because of the extra torsion induced by the large asymmetrical loading.
4. 150 N force vertically applied next to node CP, on top arch. This is the area in which the maximum vertical load would create the highest stresses in member A-B because of the large bending moment couple with torsion.

Final design results are presented in figure 12.20 and in table 12.4. An iterative approach was required to reach those results, as the sharp edges at structural joints had to be rounded with a minimum of mass added. The maximum stress occurring in all load cases is below the failure stress, except for case in figure 12.20c where a stress concentration occurs at the root end of the main strut. That concentration can be ignored however, as the joining with the other struts will reinforce that area.

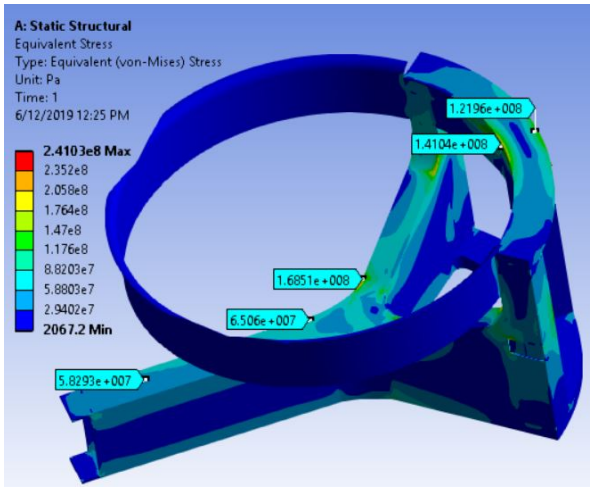
Comparing the results obtained from the MATLAB tool and the finite element simulation in Ansys, the MATLAB tool provides an accurate estimation of the results for load cases 1 and 2, which are symmetrical load cases. However, it becomes unreliable for asymmetrical loading, because of the assumption of slender beams. The beam joints are assumed by the frame approximation to be nodes with negligible area, which is not representative for the design. In loading cases 3 and 4, which are asymmetrical, stress due to torsion is therefore significantly overestimated in the arch elements (C-D, D-E, E-F, F-G, CP-DP, DP-EP, EP-FP, FP-DP). Only the results for members A-B, B-C, B-G, C-CP, E-EP and G-GP were considered in the asymmetrical loading cases. The maximum stress in an arch structure occurs when the load is placed on the crown, corresponding to the symmetric loading cases. The results used for the symmetrical loading cases were used entirely. This way, the tool was used to estimate the maximum stresses occurring in each section of the approximation during the identified critical loading cases, which allowed for the elements to be sized appropriately. The design is successfully verified with Ansys.

Table 12.4: MATLAB tool numerical results, maximum von mises stress in each member

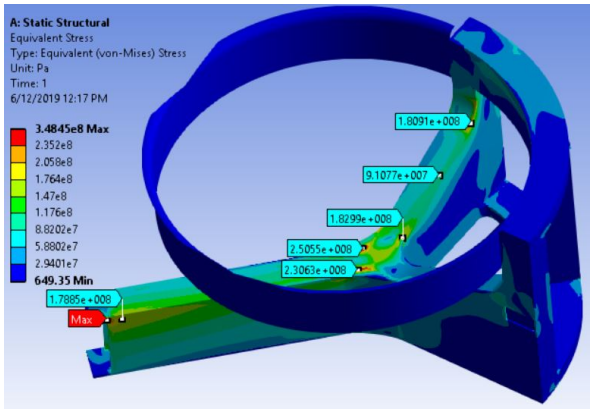
	Loading case 1 σ_v	**Loading case 2 σ_v**	**Loading case 3 σ_v**	**Loading case 4 σ_v**
A-B	7.19058e+06	4.85806e+07	1.19246e+08	8.11129e+07
B-C	5.01427e+07	1.52245e+08	1.32014e+08	5.57885e+07
B-G	5.01427e+07	1.52245e+08	1.22464e+08	2.52390e+07
C-D	1.62022e+08	2.76441e+07	3.35576e+08	1.63391e+08
D-E	1.74880e+08	3.51412e+07	3.25409e+08	1.64255e+08
E-F	1.74880e+08	3.51412e+07	3.01712e+08	7.10101e+07
F-G	1.62022e+08	2.76441e+07	2.89167e+08	6.75050e+07
CP-DP	8.79359e+07	1.34792e+08	4.51811e+08	3.92709e+07
DP-EP	8.98918e+07	1.62530e+08	4.47295e+08	4.10252e+07
EP-FP	8.98918e+07	1.62530e+08	2.28357e+08	4.10149e+07
FP-GP	8.79359e+07	1.34792e+08	2.19213e+08	3.90350e+07
C-CP	2.62735e+07	2.03741e+08	1.88642e+08	1.82281e+07
E-EP	2.25460e+07	2.83613e+07	7.54554e+07	2.33189e+07
G-GP	2.62735e+07	2.03741e+08	1.14897e+08	2.41313e+07



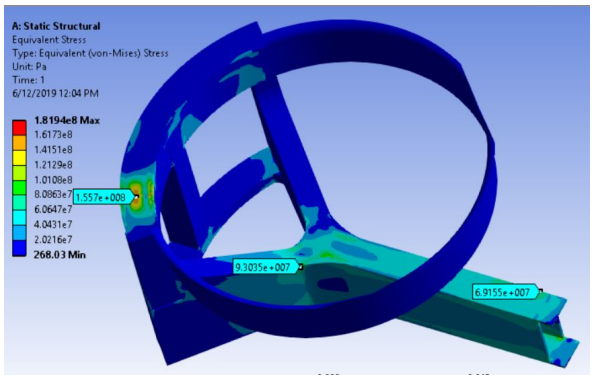
(a) Analysis results, 300 N normal to surface force applied at node E



(b) Analysis results, 300 N normal to surface force applied at node EP



(c) Analysis results, 300 N normal to surface force applied next to node CP, on top arch



(d) Analysis results, 150 N vertical force applied next to node CP, op top arch

Figure 12.20: Ansys analysis results

12.2.2. PROPELLER DESIGN VERIFICATION

Hand calculations performed on the propeller blades can be verified by performing finite element analysis in Ansys for the same loading cases. Figure 12.21 displays the results. The maximum stress predicted with the simple hand calculation is 78.1 MPa, and the maximum occuring stress in the FEM simulation is 73.5 MPa.

Both values are well under the tensile strength of the material, of 310 MPa. Therefore, the propeller design is verified.

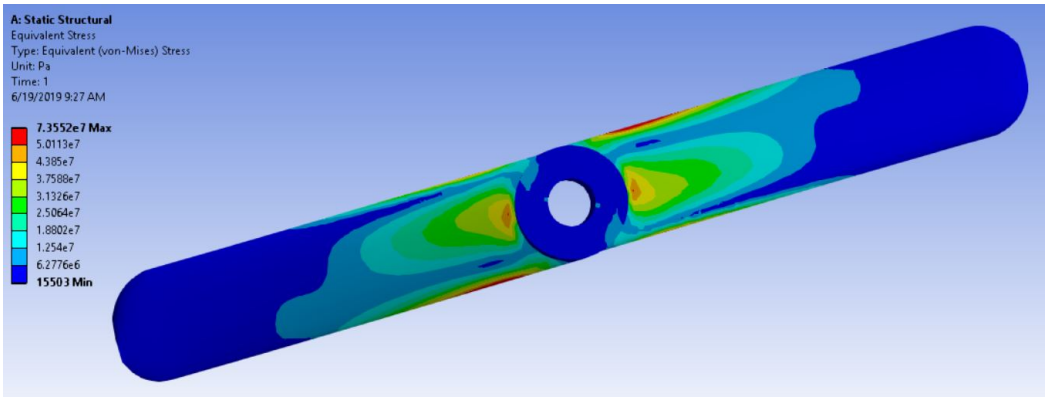


Figure 12.21: Von Mises stress occuring in propeller

12.2.3. Frame Design Validation

The structural design can be validated by performing two kinds of tests: compression tests on 3D printed drone arm specimens, and drop tests of the entire assembled structure with foam. For the compression tests, one of the drone arms can be manufactured, clamped at the electronic compartment end and compressed at the duct end with a press with the design loads. For the drop test, the entire structure can be manufactured, leaving out the electronics from the assembly to reduce the cost of the test, and dropped repeatedly from the equivalent maximum estimated potential energy. It is necessary that the printing technology is selective laser sintering in order to get relevant test results. As no such printer is available to the team at the moment, the test would have to be performed post-DSE. Validating tests are required to eliminate uncertainties related to material properties associated with the manufacturing procedure. In case the results are not satisfactory, further design iterations are necessary for the structure subsystem, possibly even reconsidering the manufacturing procedure.

12.2.4. Propeller Design Validation

The propeller design can also be validated with a laboratory test in which it is rotated to the specified maximum angular velocity. As injection molding is a highly predictable manufacturing procedure, and the occurring stress is not high relative to the strength of the material, it is likely that the result of a practical test will be positive.

12.3. Aerodynamics & Propulsion

Verification of the propeller performance predicted by JBLADE is done using a CFD simulation in ANSYS Fluent. The domain is constructed by enclosing a CAD model of the propeller, which was made in CATIA, by a fluid disk with a radius of 65 mm and a height of 20 mm, which in turn is enclosed in a larger cylindrical fluid with a radius of 0.4 m and a height of 0.8 m.

The domain is meshed in ANSYS Mesher. The default element size is 20 mm with a growth rate of 1.2, mesh defeature size of 0.1 mm and a minimal curvature size of 0.2 mm. Furthermore, the faces near leading, trailing and tip edges of the propeller are refined using an element size of 6×10^{-2} mm. This results in a mesh of approx. 2 400 000 tetrahedral elements with a minimal and average orthogonality of 0.23 and 0.77 respectively.

In ANSYS Fluent, a steady state, realizable k-ϵ turbulence model is used with scalable wall functions [57]. A Multiple Moving Reference Frame (MMRF) is used to rotate the inner disk (with propeller geometry) at a rate of 11 000 RPM [1]. The circular faces of the outer cylinder are set as pressure outlets, while the side surface and propeller are set as no-slip wall boundaries. The SIMPLE algorithm [58] is used with second-order spatial discretizations.

[1] Due to the limited timeslot, only the approximate RPM needed to hover is analyzed

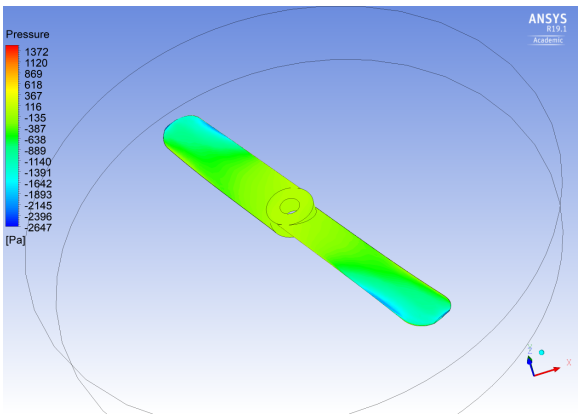


Figure 12.22: Propeller top surface pressure contour

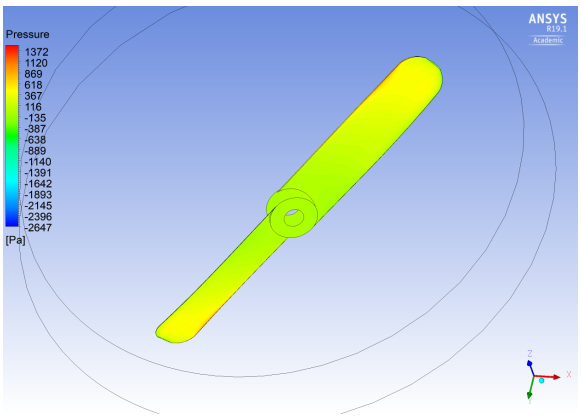


Figure 12.23: Propeller bottom surface pressure contour

The pressure contours on the top and bottom surfaces of the propeller can be seen in figure 12.22 and figure 12.23. As expected, the gauge pressure is positive on the top surface and negative on the bottom surface, and the pressure difference is greater near the blade tips. A total force (pressure and viscous forces) of 0.8158 N is found in axial direction, while the total torque around the rotation axis is 6.605 mN m, which equates to a power of 7.6 W. Normalizing these values according to equation (8.2) and equation (8.3) results in a C_{T_0} of 0.1981 and a C_{P_0} of 0.1008. The FOM is evaluated at 0.6982 using equation (8.4), which is 1.8 % lower than the predicted value of 0.711 from JBLADE. Both the C_{T_0} and C_{P_0} values obtained from JBLADE differ significantly, but the FOM is very close. A real life test should be performed to validate these results, similar to the ones performed in [51], but again this is not within the scope of this project.

12.4. Payload and Electronics

In this section the payload and electronics subsystem design is validated. Verification was deemed not suitable for payload and electronics, as no software of analytic models have been used in developing the payload and electronics subsystem. Instead, several tests that either are performed or are to be performed in the future are described as validation. The results are stated and linked to both the requirements and the risks.

12.4.1. Validation

This subsection presents the validation tests that were either executed or that are proposed for future design phases. These tests cover the validation of the fuel tank illumination and the validation of the green mask detection algorithm.

12.4.1.1. Illumination Validation

The required illumination of 100 lm/ft^2 is validated by illuminating the inside of a wing with a LED light. The LED source that was used is a simple smartphone flashlight. Via measurements it was found that this flashlight outputs approximately 60 lm. The test video was shot at a distance of approximately 1 m illuminating an area of 1 m × 1 m, which is similar to the maximum distance of the drone cameras to the tank structure. An example of the green masks to be detected by the tank is shown in figure 12.24. An image of the illuminated tank structure is shown in figure 12.25.

Figure 12.24: Green masks of dye during inspection [6]

Figure 12.25: Test image of similar wing [6]

As can be seen in the test results, the tank is illuminated well enough to spot all individual elements in the fuel tank. Therefore, this test adds confidence to the fact that green spots in the tank can be seen by the cameras with this amount of illumination. This test is not very accurate due to the inaccuracy in the brightness of the phone light. For this reason, a safety factor of 2 was applied to find the required illumination. Based on this, the required value of 120 lm/ft^2 was determined and proper LED sources were selected accordingly.

12.4.1.2. Detection algorithm validation

Validation of the Simulink color detection algorithm was performed by running it via a smartphone camera on a paper sheet with green areas on it. The result is visualized in figure 12.26. The algorithm was run on a printed sheet with green marks with RGB code 0, 120 and 0. The module was set to detect this color with a margin in HSV of 0.3, 0.22 and 0.32, respectively and return a red pixel for each detected green pixel. As can be seen, the module detects the preset green color as intended. At some points a few pixels are not detected as for example in the lower left corner of the big square, but when moving the camera over the image, all green pixels are highlighted at some instance.

This test was performed at a height of approximately 30 cm. This distance is similar to the distance of the drone camera inside the fuel tank. As this test shows that all four objects in figure 12.26 have been clearly identified, it is validated that the drone could use this algorithm to detect green dye at similar distance. As the exact fuel tank conditions differ from the test conditions, the thresholds for detection should be adjusted for optimal results. Therefore, when testing the algorithm further in a real tank, the margins for HSV should be iterated over to find the optimal value where only the green dye is detected. For actual inspection, it would be better to set the thresholds to slightly higher values than optimal, as it is better to accidentally inspect a spot that is not actually an anomaly than to miss an actual crack in the tank. In this way, risk **C6** is mitigated, as the certainty of detecting an anomaly is increased.

Figure 12.26: Simulink color detection algorithm run on green paper

12.5. Grounding

This section will elaborate on how the grounding solutions explained in section 9.4 can be verified and validated. As this section primarily describes physical tests, it mainly focuses on validation. For grounding, no numerical or analytic models are used, so no verification is performed in this area. A flowchart visualizing all the tests related to grounding can be found in figure 12.27. This figure shows the main steps to be taken to validate the selected grounding solution. First, general static charge tests will be performed, in addition to tests for the redacted solution. If it turns out that only this is not sufficient to prevent sparks, the addition of an anti static agent will be tested. Finally, if it turns out that this also does not gives the desired results, other options are looked into. The following sections will elaborate on all individual tests that either are performed

or should be performed in the future to ensure safe operation of ARTIS.

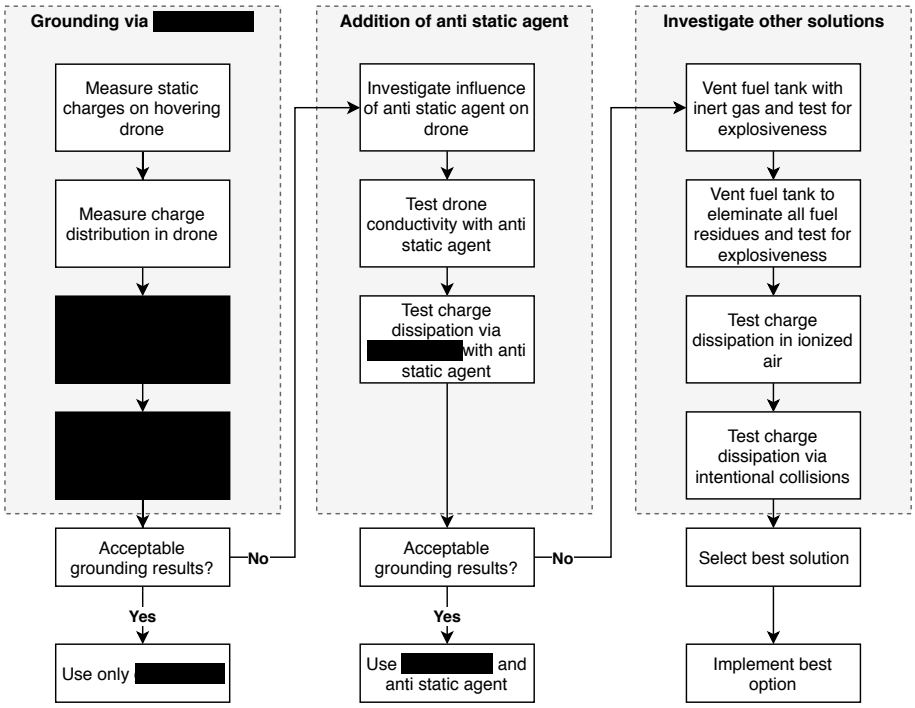


Figure 12.27: Verification and validation of the grounding

12.5.1. ELECTROSTATIC CHARGE TEST

Creating sparks within the environment in which ARTIS is operating must in all cases be avoided (**AMD-NF-CNST-Sfty-02 (S,D1)**). In order to get a first order approximation on the static charge build-up due to hovering of a drone the following test was performed.

For the test the following equipment was used; an electrostatic field meter, Zero volt ionizer, non-conductive landing pad, tape measure and a DJI Phantom 4 drone. The environment in which the test was performed had a low level of humidity and a temperature of 20 °C. This means that the actual static discharge test functions as a worst case scenario test. The relation between humidity and static discharge is explained in more depth in section 9.4.1. A positive result would thus mean even better things for the real scenario in which ARTIS will operate.

The test was performed as follows: the hand held device was connected to the grounding pins of a wall socket and pointed at the wall socket to see if a field of 0 kV/m was present. This is done to verify if the device and so the wall socket is grounded properly. Then the field at the heart of one of the drones propellers was measured at a fixed distance of 0.1 m to see if there was any electrostatic charge present. If a field existed the zero volt ionizer was pointed towards that propeller and charge was removed until it reached a value close to 0 kV/m for the electric field. Next up the drone was put into hover mode at an approximate height of 1 m so some ground interference was introduced to simulate the slightly unstable hover. After 10 minutes the drone is landed on the non-conductive landing pad. This 10 minutes of flight time is related to **AMD-F-P&P-Endu (S,Y)**. Finally the electric field was measured, again at a distance of 0.1 m to have consistent results.

Performing the test resulted in a zero measurement of −0.065 kV/m and a final value of 0.210 kV/m. During the measurements the human body heavily interacted with the electric field, changing the measurement by half a magnitude. Using this observation a quantitative conclusion is made. The static electric field generated by hovering with a DJI Phantom 4 (Larger size than ARTIS) for 10 minutes is around the order of magnitude of the electric field around a human body. This result is not conclusive enough to actually state that no sparks will be created.

Further recommendations and improvements for this test are as follows. In the first place the environment must be as simplistic as possible and surrounding objects must be grounded, including all persons within

the room. Secondly a rigid tripod shall be used for the measurement device, as well as landing spot markings for the legs to always ensure the same measurement position and distance. Thirdly a thermometer, humidity indicator and climate control should be used to better regulate the flight environment. Lastly one might introduce flying in a more compact environment to see if the wall induction might introduce more electrostatic charge.

12.5.1.1. Charge Distribution

The assumption made previously in section 9.4.3, regarding the lack of electrostatic charge buildup outside of the ducts and propellers will have to be validated before operation of the drone in the hazardous environment can begin. Even though this is assumption is supported by the fact that ARTIS does not fly very fast and should avoid any collisions, which would mean the triboelectric effect could be negligible. This verification can be done by testing the distribution of static charge in a drone that has been hovering for a certain amount of time and comparing with a drone that has been flying around slowly for the same amount of time.

If the drone, outside of the ducts and propellers, does not hold any significant static charge this assumption can be considered confirmed. However, in the case that there is a significant charge found on the drone, any solutions based upon this assumption will have to be considered not enough.

12.5.1.2. Testing the redacted solution

Several tests have also been proposed for the solution for which a patent is currently being considered. This has thus been left out of the report since it revealed the nature of this solution too much.

12.5.1.3. Testing the anti static agent

In case it is found that the selected solution is not sufficient, anti static agent is applied on the drone as shown in figure 12.27. The purpose of this agent is to make the drone more conductive so that any built up charge can be more easily dissipated. Testing the drone with the agent applied will be done via three steps:

1. Test the effect of the anti static agent on the drone. The anti static agent is a spray that will be applied to the drone before flight. Research should be done on how the agent interacts with the drone. The drone is designed to be air and liquid tight, but spray vapours might still hinder for example the performance of the motors. The influence of the spray can be investigated by spraying the drone with the agent and then flying it. Afterwards, the drone should be disassembled to check if any parts got damaged or affected in any way.
2. Test how the anti static agent influences the conductivity of the drone. One should apply the agent to the drone and perform measurements on how the agent affected the conductivity of the drone. Using an electrostatic charge meter, one should investigate if all parts on the drone carry the same amount of charge, meaning that the charge is homogeneously distributed.

12.5.1.4. Other grounding tests

If it is found that both the redacted solution and the anti static agent do not ground the drone as desired, other options are considered. These options are visualized on the right side of figure 12.27 and described in section 9.4.2. These solutions include venting the tank with inert gasses, venting the fuel residues out of the tank, ionizing the air inside the tank and dissipating the charge via intentional collisions. All of these options will be further investigated and the best of the aforementioned options will be implemented in the drone accordingly.

13
OPERATIONS AND LOGISTICS

This chapter will elaborate on the operations and logistics of ARTIS. Firstly section 13.1 contains a RAMS analysis. Then section 13.2 will go into the operational environment, the actual operations for the customer and finally the logistics.

13.1. RAMS ANALYSIS

This section is dedicated to the discussion of the reliability, availability, maintainability and safety characteristics of the ARTIS project. Risks have already been discussed in chapter 6, considered throughout the design and will be once again be discussed in chapter 16. This section will therefore focus on other RAMS characteristics.

13.1.1. RELIABILITY

The reliability analysis relates to the ability of ARTIS to operate continuously and to a satisfactory level. Note that this section is speculative, and an objective truth about the reliability of ARTIS can only be determined once tests have been performed and ARTIS has been in use for a substantial time period.

It is assumed that the failure rate over the lifetime of ARTIS shall be distributed in a similar way to the bathtub curve shown in figure 13.1. The failure rate will probably be high in the start phase of the usage and the end phase when the product wears out. During the remaining useful life period, the failure rate should be approximately constant. Standard commercial drones (non-autonomous quadcopters) have been estimated to fail at a rate of 1 per 1000 hours [158]. Nevertheless, this would be an optimistic estimate for ARTIS given its autonomous nature and the rough environment it must operate in. This failure rate should thus be updated in order to be an applicable estimate for ARTIS.

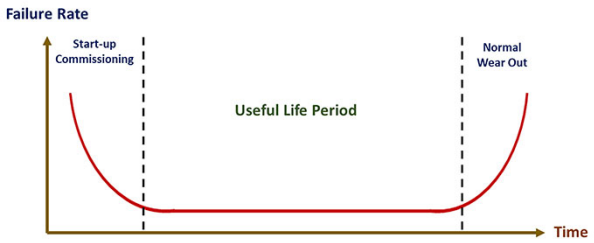


Figure 13.1: Failure rate bathtub curve

Firstly, hardware components are analyzed. Table 13.1 lists the reliability of the main hardware components used in ARTIS. The reliability of the VPU was estimated based on information found about CPU reliability [159] rounded down to one decimal point given that the VPU that will be used is new and not released to the public yet, it is thus expected to have lower reliability since its still in an early-life phase. The battery reliability was estimated based on its life time (see figure 9.8). All remaining values were based on literature sources [158–161]. From this data it can be seen that most hardware components have a high reliability and should not cause the the failure rate to increase from 1/1000h.

Table 13.1: Reliability of components

Component	Reliability
CPU	0.99996 [159]
VPU	0.9
IMU	0.99996 [159]
Battery	0.9967
Motors	0.99996 [160]
Cameras	0.97 [161]

In order to investigate other possible causes for failure apart from hardware reliability, a high level fault tree was composed as seen in figure 13.2. The tree includes major events that can cause the system to fail and thus highlights the driving components for ARTIS' reliability.

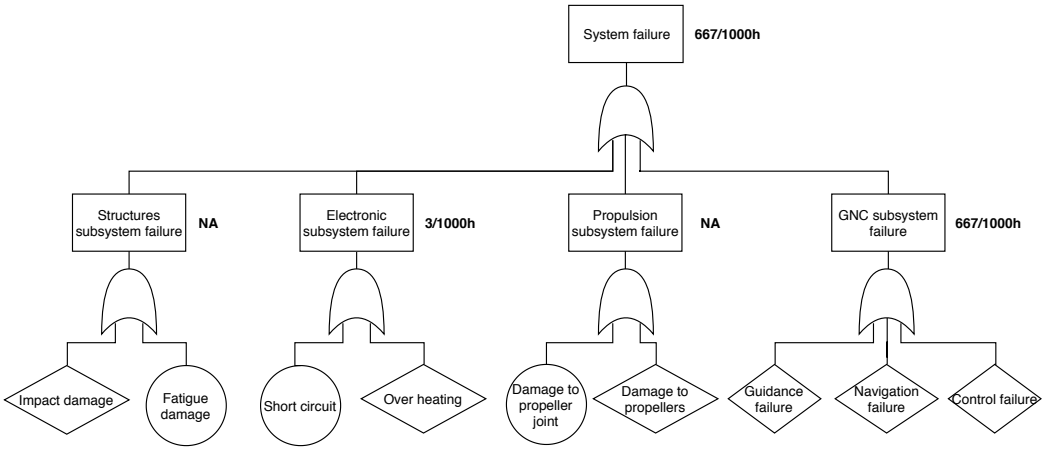


Figure 13.2: High level fault tree for unsuccessful inspection

The structural failure events are dependent on the structural frame of ARTIS. Without testing, it is not suitable to make predictions on the failure rate for structural systems. For electronic subsystem failure, an upper limit for failure is taken from literature corresponding to a failure rate of 3/1000h. The upper limit was taken given the infant stage of the design. This failure rate is based on big data information on electronic systems from the US Navy, British Telecom and Bell Telephone [162]. Given this information, the failure rate is brought up from 1/1000h to 3/1000h. The propulsion systems failure, if considered independently from hardware failure, can only result from structural damage to joints or to the propellers themselves. Similar to the structural subsystem failure, this is challenging to predict at this stage of the design. The final failure events are those relating to the GN&C subsystem. It is suspected that these events are those which will result in a higher failure rate. Autonomous flight in a confined environment with low error margins is uncommon and innovative, thus the technology is not fully developed and the system proposed at this stage may need to be updated and corrected during the growth phase of ARTIS' lifetime. A preliminary estimate based on a study of monocular SLAM autonomous flight results in a time to failure of 90 minutes (equivalent to approximately 667 failures per 1000h). In this study the algorithms used were ORB-SLAM and ORBSLAMM [163]. The final failure rate during ARTIS' useful life period is thus estimated to be 1 failure per 90 minutes. The failure rates can also be seen in figure 13.2.

13.1.2. MAINTAINABILITY

Maintenance operations are key for the continuous, safe and optimal performance of the ARTIS. Therefore it is crucial to outline maintenance procedures for the ARTIS' lifecycle. To this end several maintenance polices must be determined, including the description of regular and periodic checks as well as corrective and preventive measures.

For each deployment of ARTIS, damage checks should be performed before and after operations. The damage checks include visually inspecting the structure, motors, propellers, battery and SD cards (an outline for visual inspection can be seen in table 13.2). In case the inspector suspects there is an anomaly, the drone can not be deployed into operation and should undergo an unscheduled thorough inspection as outlined in table 13.3. If the visual inspection is passed before operation, a short flight test should be performed to determine if ARTIS is behaving nominally. This test only needs to be performed before deployment into the fuel tank. Once again if suspicious features are observed, ARTIS should undergo a more thorough inspection and can not be used in operations. If both pre-operation tests are passed ARTIS can perform the mission. After the mission is complete, only the visual inspection needs to be performed.

When a thorough inspection is performed, any defect parts should be replaced. The downtime due to replacement depends on the availability of spare parts and the time it takes to order required parts. The thorough inspection outlined in table 13.3 should also be performed periodically independently of the outcome of visual inspections. It is suggested that a thorough inspection is performed in between 10 and 20 inspections. This period is a preliminary suggestion and should be updated as a result the ARTIS testing period. A

Table 13.2: Visual inspection outline

Visual inspection action	What to look for
Structure	Cracks, especially in the joints, foam condition and any other surface damage.
Motors	Condition of shaft and copper coils
Propellers	Structural damage e.g. chipped edges
Battery	Bloated battery, damaged battery connectors
SD cards	Worn out contacts, structural damage and storage space
Aluminum cover	large ripped portions and scratches
Communication	check data relay from drone to outboard monitoring unit
Total estimated time	**<10min**

Table 13.3: Thorough inspection outline

Thorough inspection action	Estimated time
Check electronics compartment cables and hardware	40min
Check motors individual for full RPM range	20min
Measure battery charge	5min
Inspect structure for cracks and fatigue.	10min
Check aluminum cover for damage	5min
Check air and liquid tight property	40min
Perform flight test	10min
Check communications system	5min
Total inspection time	**2h 15min**

second periodic action to consider is the replacement of the battery due to degradation. The battery should be replaced after 300 to 500 inspections as per the degradation cycle discussed in section 9.3.1.1.

It should be noted that these maintenance activities aid in mitigating various risks discussed in chapter 6 by preventing unfit use of ARTIS. The likelihood of risks **A2, A5, A6, C1, C3, E1, P2, S1 and S3** is reduced by the implementation of the discussed preventive maintenance inspections.

13.1.3. AVAILABILITY

Availability deals with the ability of ARTIS to be readily available for use when required. In order to minimize downtime and satisfy customers, ARTIS should be consistently readily available to use. To this end, availability characteristics and policies must be determined.

Following on from maintenance inspection on ARTIS, there may arise the need to replace certain components. If all new parts would need to be ordered when required, the downtime for repair would be high. Therefore a set of spare parts should always be available with ARTIS. The list of spare parts that should always be available along with ARTIS can be seen in table 13.4.
Once the spare parts reach a quantity threshold of 2, the parts should be ordered in order to maintain the listed quantities of spare parts. This policy would improve the availability of the drone as it would make most of the common parts readily available for repair. If a part that is not included in the spare part list is required it would have to be ordered. In this case the repair downtime would increase due to the delivery time. The time would vary depending on the part supplier.

Table 13.4: Spare part list

Part	Quantity
Propellers	8
Batteries	3
SD cards	8
Cameras	4
LED lights	8

In order to maximize availability, it is suggested that at least two drones are present at the MRO facility where the fuel tank inspections are performed. It is also suggested that certain personnel are trained to work with ARTIS, in order to perform thorough inspections and repairs.

13.1.4. Safety

Safety functions and policies are critical considerations for the ARTIS project. As risks have already been dealt with, and will be further discussed in chapter 16, this section is limited to the outlining of safety policies in order to both address possible risk consequences and improve the overall operational safety.

The following list outline suggested safety procedures and considerations that should be followed when operating ARTIS:

- While ARTIS is performing the fuel tank inspection, all personnel should not be in contact with the wing structure.
- All fuel inside the fuel tanks should be drained before the deployment of ARTIS (small puddles that are not removed with normal draining activities are acceptable).
- There should only be one drone operating inside a fuel tank compartment at any point in time.
- The aircraft should be grounded at all times during the inspection.
- No exposed electronics should be present on ARTIS when it is deployed.
- All onside personnel working on the aircraft should be aware of the use of ARTIS.
- There should always be two people operating ARTIS.
- ARTIS should be handled with care and always be transported safely.
- ARTIS should never be deployed under circumstances that are suspected to be unsafe.
- ARTIS should not be deployed when atmospheric conditions are not within acceptable limits (e.g. extremely high temperature, high pressure etc.)

13.2. Operations and Logistics Analysis

ARTIS is designed for inspecting the fuel tank of the C-130. First the inspection environment will be explained to show where ARTIS will operate. Then a breakdown of the operations is done. Within this breakdown a diagram will show the tasks done by either ARTIS or the operational personal from unpacking of ARTIS until packing ARTIS again for storage. After the operations, the logistics behind ARTIS from ordering up to discarding will be examined.

13.2.1. Operational Environment

As ARTIS is mainly designed to be operative inside the fuel tank of a C-130 aircraft, a more detailed description of the working environment is provided. According to the maintenance document of the C-130 and personal contact with Lockheed Martin cooperation the fuel tank consists of multiple bays [14, 164]. Each wing half has three separate fuel bays all having elliptical entrances that are located on top of the wing. Inside these fuel bays fueling pipelines cross both horizontally and vertically, obstructing the open flight space. Besides the fueling system itself, other obstacles and geometrical features are present within the tank. Along the top and bottom panel stringers are present in spanwise direction. These stringers have a height of approximately 60 mm, which constrains the allowable flight altitudes. Moreover, ribs consisting of triangular truss structures are present in spanwise direction with a constant placement pitch. A simple view of the fuel tank bottom with ribs (flat plate ribs, ribs with holes and truss structured ribs) is presented in figure 13.3. All these features plus the before mentioned fueling system pipes within the fuel tank create a complicated flight environment. As ARTIS flies autonomously, the GN&C was designed according to this intricate environment, as explained in chapter 10. Besides the geometrical features that are always be present, the environment also changes dynamically. As ARTIS will enter the fuel tank after every four flights fuel residues might be present, even after draining the tank. Moreover, during flight, cables and or pipes might come loose, changing the environment. These dynamic changes are taken into account within the mitigation of risk **C5** as described in section 10.3. Even though ARTIS is designed for the C-130 fuel tank environment, other airplanes have similar fuel tank configurations, e.g. Lockheed Martin's C-5 aircraft Martin [165]. This will improve the market position as explained in chapter 4.

13.2.2. Operations Breakdown

Now that the environment is described, the operations for performing such inspection will be elaborated on. The overall operations are shown in a flowchart diagram figure 13.4. This diagram consists of four main operation activities. First (1.0) is the preparation and initialization of ARTIS. Within this process the customer

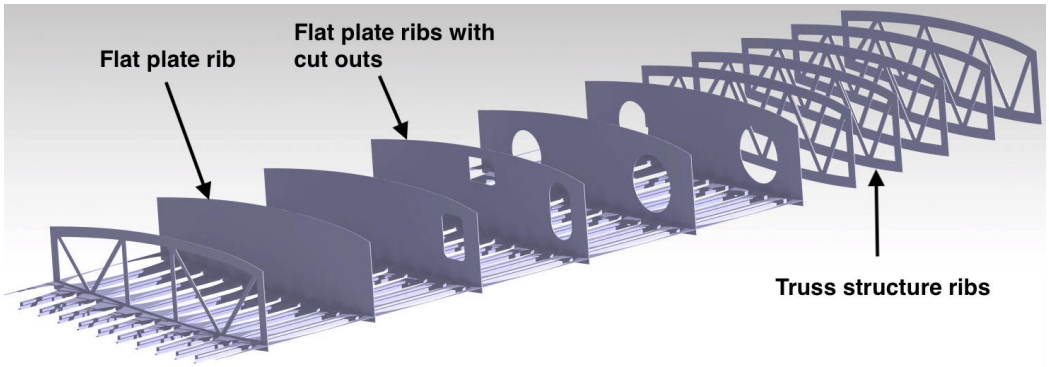


Figure 13.3: Open view of the fuel tank wing section

shall check ARTIS for defects and make sure that the battery and SD cards are ready for use. If defects are detected a maintenance overhaul is required. The procedure for this is explained in section 13.1.2. Then, before deploying ARTIS into the fuel tank, the SD cards inserted in ARTIS will be switched on via the capacitive switch on the motherboard, and then the battery charge is checked. Following the initialization of ARTIS, a hover test will be executed to observe the flying capabilities. During this test the user can check the status of ARTIS on a monitor and see if no problem flags are sent and ARTIS can perform a stable hover.

Secondly (2.0) ARTIS will be deployed inside the tank. When inside, the user can send the start inspection command to ARTIS, which will then start the inspection. During the entire inspection the user should monitor ARTIS to see if no problems arise. For the inspection ARTIS will follow a path determined by the guidance algorithm described in section 10.4 based on the key points set in section 11.2.2. If no problems arise it will inspect the fuel tank and save the video and images. ARTIS will keep the operator up to date by sending flags and its position to the outboard monitoring unit. For more detail about communication during the inspection an explanation with accompanying diagrams are found in section 9.1.2 and section 9.1.3 accompanied by figure 9.2 and figure 9.3. As explained in section 13.2.1 the tank consists of multiple bays. When ARTIS is done inspecting one bay, the operator will check if the battery has enough charge to inspect another bay. This will decrease the likelihood of running out of battery when inside, i.e the likelihood of risk **P3**.

Thirdly (3.0) different problems might arise during the inspection. The severity of these unexpected situations are explained in section 13.1.1 with an accompanying fault tree figure 13.2. When a problem arises ARTIS will notify the user by switching a flag via the transmitter to the outboard monitoring unit. As the environment is dynamic, the avoidance of unexpected obstacles will lead to going off-track, creating a sub-path for going back to the normal procedure. Another problem that might arise due to the dynamic environment is getting stuck. There are three possible options for this, of which all comprise different approaches to decrease the severity of risk **C1**. First the environment might change after ARTIS passed by blocking the way back or the navigation fails in identifying a hole within a rib. This will make ARTIS activate safe mode leading to a controlled landing. Secondly, it might hit something and as a result may not be able to fly away. This will make ARTIS switch off the engines so no further damage will be done. Thirdly, an object might block the path so that no further inspection is possible. This will make ARTIS return to the exit. Then, during the inspection a component might fail. Depending on the severity ARTIS will either go to the exit or activate safe mode. Activating safe mode is done to mitigate risk **A6**. A problem might not only arise concerning ARTIS, but an outside emergency may happen. The operator will then send a return command to ARTIS and retrieve ARTIS. Lastly, as mentioned in section 11.2.2 the total inspection time is longer than one battery charge lasts. When a battery charge of below 10% is reached the operator will be notified and ARTIS will return for charging. Charging before deployment for inspection also helps mitigate risk **P3**.

To retrieve ARTIS the official method of entering a fuel tank needs to be performed as explained in [164]. One should start looking for ARTIS if no sign of life is noted for 15 minutes or a flag has been sent that ARTIS landed within the tank. The 15 minutes is based on flight time battery life of 10 minutes and 40 seconds, this makes sure that ARTIS has ran out of power and that the longer taking official procedure is not started when

there might still be possibility for ARTIS to return on its own.

Lastly (4.0) is for the operations when ARTIS is finished with the inspection it will locate the exit and land as it would during an emergency. The operator then switches ARTIS off. As before the inspection started a visual check is done after which the SD cards are removed. The battery will be charged and the SD cards will be emptied onto the outboard monitoring unit. The operator then runs the green primer algorithm on the video feed and images and afterwards checks the results. During the reviewing of the video and images data the operator will check for false positives which do not lead to further action, moreover it will check for positives which will lead to further action. This post data processing mitigates the risks **A4** and **C6**. If data is corrupted the operator will notice this during the data processing and resend ARTIS back in to capture the corrupted parts. This mitigates risk **C3**. In the end ARTIS will be switched off and stored inside a case, including the charger and SD cards.

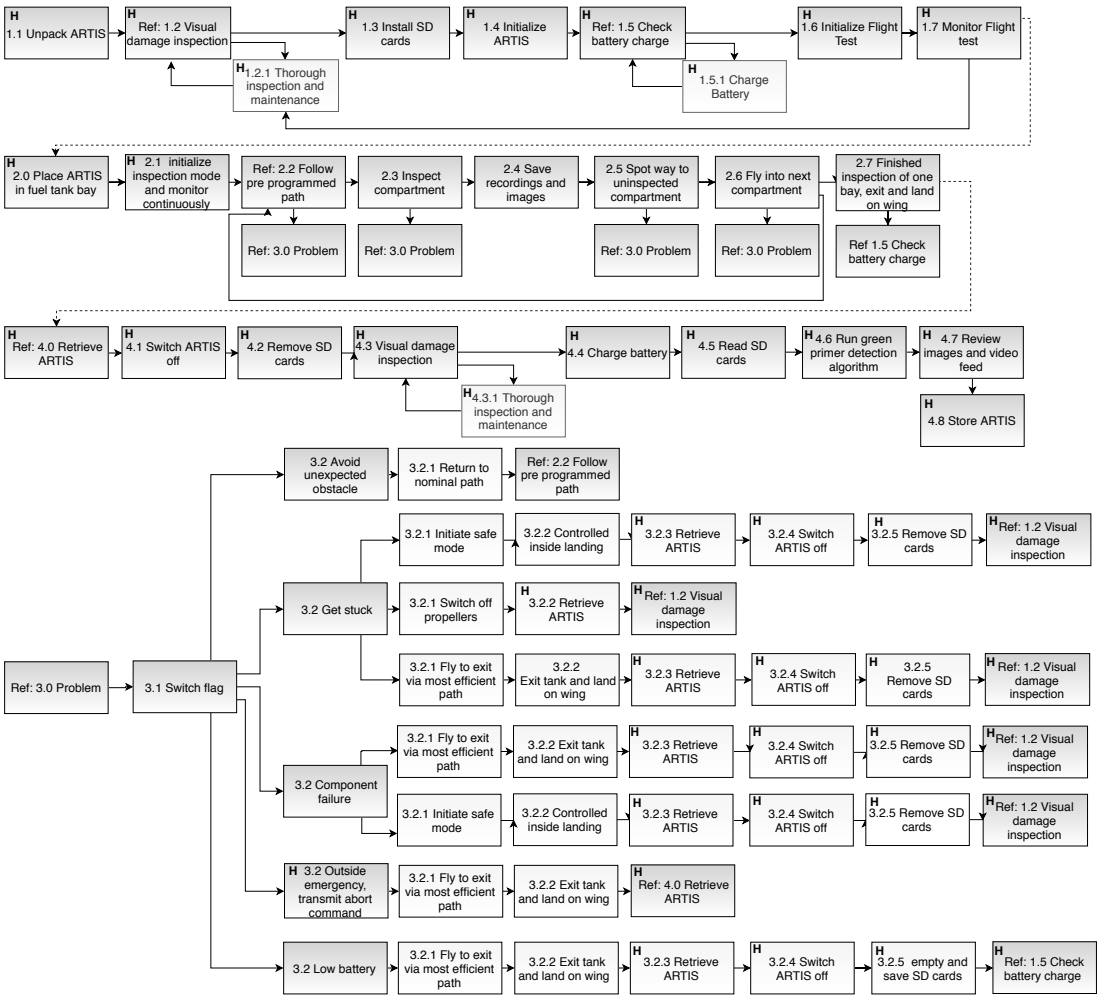


Figure 13.4: Operations diagram for inspection of a fuel tank by ARTIS

Now all the operations have been described an approximation on the total operation time is made. The operations done will be based on figure 13.4. The estimation will be made under the assumption that no other problems arise and charging is not required after every bay. Moreover the inspection is done with one drone. If a problem arises it will add additional time to the final time stated. The actual inspection path time explained in section 11.2.2, resulted in a time of 27 minutes and 19 sec. With 1 battery charge ARTIS can fly for 10 minutes and 40 sec and so charging is required. This will both be done during the inspection and after the full inspection to make sure the battery is charged before storing ARTIS. The retrieving and placement of ARTIS, connecting the charger and performing the hover test is included within inspection preparation. For running the green primer algorithm and reviewing the results a first order time estimate

was made. This estimate is based on the total inspection time which is in correlation with the amount of video and image data. Then before deploying and storing ARTIS a visual inspection is performed for which the time was determined in section 13.1.2. Lastly storing ARTIS and making sure all parts are included will approximately take 5 minutes. Adding all these operational tasks summarized in table 13.5, comes down to a total operational time for one inspection of 235 minutes and 55 sec.

Table 13.5: Total operational time for one inspection

	Time [min:sec]	**Amount**	**Total time [min:sec]**
Visual inspection	10:00	2	20:00
Inspection preparation	5:00	6	30:00
Inspection (1 charge)	10:40	2.56	27:19
Battery charging (full charge)	60:00	2.56	93:36
Green primer Algorithm	30:00	1	30:00
Image and video review	30:00	1	30:00
Inspection ending	5:00	1	5:00
Total operation time			**235:55**

As the estimation above is based the operational use of only one drone it can be noticeably improved. One improvement is reached when multiple drones are used sequentially, so there is no waiting time for in between charging. An even larger improvement is made when multiple drones are used simultaneously. When multiple drones are used simultaneously a stronger outboard monitoring unit must be used to handle all the communication links. This is preferred over using multiple outboard monitoring units to keep all data on one screen which is easier to monitor for the operator. Both off these improvement will save time on charging but more visual inspection are needed to check all the drones used. In the end it will be up to the user and the available resources to optimize the personal operations.

13.2.3. Logistics Breakdown

Besides operations the logistics is another important aspect. A customer can order ARTIS. When a customer orders a drone they shall receive ARTIS from the stock or when a larger order is placed ARTIS will be produced on demand. The total Cost of ARTIS and additional parts will be determined in section 14.1. Before shipment ARTIS will be tested and than marked as ready for use. As ARTIS is shipped fully assembled (SD cards not inserted), the package also includes a charger, storage case, manual and spare parts. The amount and specific spare parts were described within section 13.1.3. For the overall operations the procedure is shown in section 13.2.2, this will also be included inside the manual. When the customer needs additional spare parts he can make a request and the spare parts will be delivered against certain costs. The main operations and maintenance of ARTIS are described in section 13.2.2. At a certain moment ARTIS might be irreparable due to a crash or it reaching end of life. For disposal extra attention should be noted for the battery as this is the most harmful component for the environment. For the other components a specific protocol will be made and included within the manual. All the recycling and disposal is done with the purpose of leaving the smallest possible carbon footprint.

14
FINANCIAL ANALYSIS

The total cost of producing, assembling and internationally distributing a maintenance drone for the Lockheed Martin C-130 must be established to assess potential savings compared to the current expenses. As was outlined in chapter 4, with approximately 26 000 C-130 fuel tank inspections being performed per year, Lockheed Martin incurs a total cost of $42.9 (i.e. €38.30) millions per year for this specific inspection purpose. Aiming to reduce their overhead expenses, the financial analysis presented in this section will outline the revenues the team would make during the first year of its commercial activity, as well as the benefits to its first contractor, Lockheed Martin.

Throughout the financial analysis, the first year of sales (anno. 2020) is the centre of attention. Years following will be considered as later stages and will be discussed qualitatively. A conversion rate from USD to EU of 0.89 will be maintained throughout the section, based on an average of Google Finance's conversion rates in the month of June 2019.

14.1. COST BREAKDOWN STRUCTURE

To come up with an efficient estimate, a Cost Break-down Structure (CBS) is set up. This AND tree identifies all elements involved in the production, assembly and international distribution of the ARTIS. The total costs will be segmented into *system costs* (with a €2500.00 **AMD-NF-CNST-Cost-01** requirement presented in [6] and *supporting costs*, in [6] being set at €5000 per drone by **AMD-NF-CNST-Cost-02**). Initially focusing on the latter expenses, figure 14.1 shows how the tree is further segmented into shipping & delivery costs, maintenance & support, research & development, the sales & marketing team and the management board. The supporting costs have thus been structured under an all-inclusive framework. For the first-level categories, the percentages shown below refer to the part on the total ARTIS cost, and will be elaborated on in the rest of this section's discussion. The value being shown for the supporting costs (i.e. €8806.7, above the mentioned requirement of €5000.00) is to be intended for the overarching supporting structure lying at the base of the first year of operations of a for-profit institution capable of producing, assembling and internationally distributing a maintenance drone for the Lockheed Martin C-130 fuel tank. Hence, it shall not be compared to the mentioned requirement.

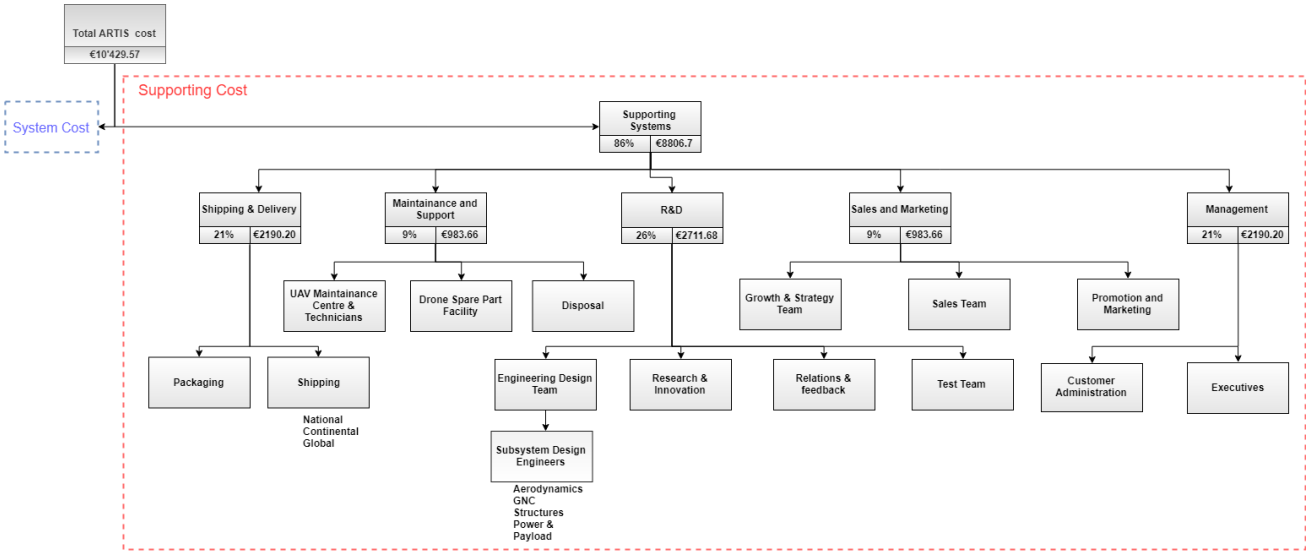


Figure 14.1: Cost breakdown of ARTIS supporting costs

The focus of the present discussion will be firstly on the CBS structure itself, followed by the approach to the set-up.

For delivery costs involved in the post-preliminary system design stages, it is expected that the first office, production lab and assembly facility lie within Delft and its vicinity. It is worth mentioning as the drone distribution is then performed mainly overseas (i.e. to the main customer Lockheed Martin), however later possibly to more local companies (see KLM and Airbus). On the long-term, the company aims to move production to the United States to further strengthen the contacts with the customer Lockheed Martin and the US commercial aviation sector, due to its growth mentioned in chapter 4. For this, a first cash inflow is however required, thus this is expected to happen only after the first year of operation.

Throughout the ARTIS life-cycle, maintenance and repair activities shall be performed to ensure a smooth running of its functional operations. For logistic purposes, the repair centres should be located closely to the 13 Lockheed Martin MROCs around the world, with a more in-depth overview of those being explained in [17].

A key component of the ARTIS costs will be reserved for all purposes related to research and development, where most of the current team efforts and budget is spent. The importance of cutting-edge solutions call for a further division in 4 separate sub-teams, those being the engineering design team, the research & innovation team, mainly hunting for novel concepts to propose to the former team, a group of professionals dealing with client relations and feedback, and finally a testing team, mainly focusing on autonomous flight testing, material and structural strength testing, and electronic charge testing.

Within sales and marketing, 3 sub-teams will be formed. One for marketing and promotional events, one for focusing on strategies for growth and one for direct sales. Regarding promotion and marketing, two sub units will be formed, dealing with direct marketing efforts, whether in the use of social media, ads and email marketing, to name a few, and on possible ARTIS prototype projects. These projects will be held in the form of exclusive reveal events, in which interested parties are cordially invited to hear about the newest solutions and innovative prototypes.

Finally, expenses covering the team management shall be covered, and include the administrative costs.

Under the total system costs, on the left side of the presented CBS, a visual display is offered in section 14.1.1. The production and assembly section can be further segmented in fixed and variable costs. Furthermore, as explained in chapter 8, the materials required for the ARTIS can be either produced in-house via additive manufacturing, or bought via a third party provider.

14.1.1. Cost Analysis

Having given a brief explanation of the total costs breakdown, attention will now be turned to quantitative cost estimations.

The approach used to come up with the total ARTIS cost aims at making as few assumptions as possible, and at having as close an estimate to a real-case scenario as feasible. The only premises being made are the ones concerning the 6 categories of individual contributions to the total cost, those being production and assembly, shipping and delivery, maintenance and support, research and development, sales and marketing, and management. A qualitative assessment was first set in place, with 4 categories being low, medium, medium-high and high contribution to the total cost. Those were then assigned numerical values as 3, 5, 7 and 9 respectively, as shown in table 14.1. The percentages reported below have been rounded.

Table 14.1: Categories percentage contribution to total ARTIS cost

Category	Contribution to Total Cost	Value	% Total Cost
Production & Assembly	Medium	5	14
Shipping & Delivery	Medium-High	7	21
Maintenance & Support	Low	3	9
Research & Development	High	9	26
Sales & Marketing	Low	3	9
Management	Medium-High	7	21
Total		34	100

The qualitative values were assigned according to engineering judgment and looking into open-source corporate cost structures. The values were in line with aircraft division of budget [106]. Scaling the obtained values resulted in the relative percentages shown in the right-most column of the table above.

Of course, the explanation thus far does not stand-alone to justify the cost contributions. Hence, with the purpose of verifying the results, a research study concerning corporate cost breakdowns was performed. The range of spend is, of course, varying across industries and companies, but some guidelines confirming the estimates proposed thus far were found. More concerned with software providers than actual hardware manufacturers, high-tech companies tend to spend 23 % of total revenues on their R&D team [166], close to the value reported in table 14.1.

The next educated guess shown in table 14.1 that was verified is the sales percentage of the total system cost. According to research commissioned by the US Small Business Administration, sales and marketing budgets of thriving businesses should be around 7 to 8 percent, thus perfectly fitting the value obtained with the qualitative assessment [167]. Finally, an estimate of the required management spend confirmed an average between 15 % and 30 % of gross revenues on payroll, according to a 2012 survey by the National Small Business Association [168]. Using the proposed qualitative assessment, backed by a research study, it is believed the cost percentages distribution is a solid first-level estimate.

Having discussed the high-level cost distribution among which the total system is segmented, attention shall be turned to the actual estimation of the individual costs. Computing these and knowing their percentage contribution to the total will then allow assessing the ARTIS cost.

Before proceeding with the cost estimates, a small digression to assess the number of drones being produced per year is required. The market analysis in chapter 4 concluded that 26000 C-130 fuel tank inspections are performed per year. Hence, with the ARTIS inspection time around 3 hours, 2 inspections (to be conservative) shall be performed per working day. With an average of 261 working days per year, 522 C-130 fuel tank inspections are needed per year. With each ARTIS being sold with a 1-year warranty agreement, $\frac{26000}{522} \approx 50$ drones are required to be produced, per year, to keep up with the current Lockheed Martin schedule standards.

Now turning the attention to the cost estimates, as shown in the left-side of figure 14.2, production and assembly costs can be further segmented into fixed, being paid independent of ARTIS sales, and variable costs, those being a function of the quantity of ARTIS drones being produced.
The fixed costs include the drone production labs, to be originally established in Delft, the Netherlands, aiming at overseas expansions, lab technicians for assembly and the cost of a 3D printer. The cost values reported in CBS are on a per-drone basis. Starting with the 3D production, the Vit Natural Robotics SLS printer [169] perfectly fits the size and weight requirements of the ARTIS and shall be bought as a one time expense. Assuming a 5 years usage, a total cash outlay of €2000 is then obtained. With an estimated 50 ARTIS being produced per year, the 3D printer cost per drone results in €40, as shown in the CBS.

As far as the payroll for lab technicians for assembly goes, a $21.50 average rate is expected [170], which, converted to euros and considering an average of 261 8-hours working days per year, leads to €39943.44 as yearly expense. Considering the explained 50 drones per year production rate, assembly costs per drone amount to €798.87 as shown in the cost breakdown structure.

The last item to be added to the ARTIS fixed costs per year is a drone production lab. It was decided to rent office space in the team original location, Delft, in the Netherlands. The chosen office is part of a start-up incubator, YES!Delft, further strengthening the commercial viability of the ARTIS. Besides basic rent (€168.96 per m^2), yearly service costs amount to €71.28 per m^2, leading to a total of €7207.2 cost per year for a 30 m^2 space in the incubator offices location [171]. The space applies only to office spacing for the first year activities, and are subject to drastic changes once the company will gain revenues. Dividing this by the number of drones per year, the drone production lab expenses amount to €144.14, then summing up the fixed costs (i.e. 3D printer, assembly and drone production lab) leads to €983.01 per drone.

Figure 14.2 reports the expenses the team will have to incur for the ARTIS material production in the lower part of the tree. Those have been further divided into in-house production, i.e. being 3D printed, or bought

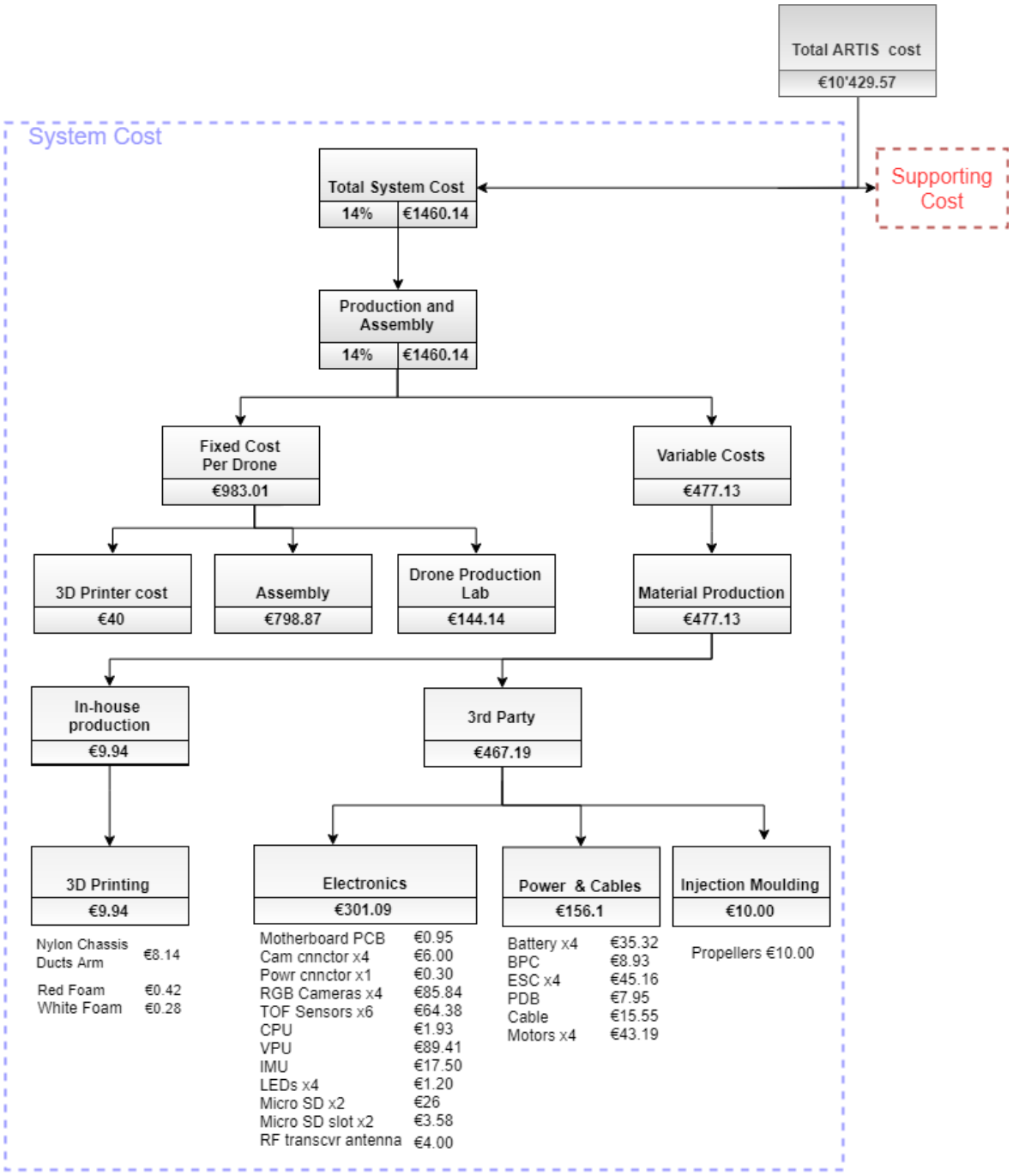


Figure 14.2: Cost breakdown structure for production and assembly costs of the ARTIS

via a third party source. Table 8.1 overviewed the material weights needed. A carbon powder filled nylon for the chassis and ducts arms will be ordered from Guangdong Silver Age Sci.& Tech. [172], at the price of $110.00 per kg. Given the weight reported in table 8.1, €88.14 worth of carbon powder filled nylon shall be ordered per drone for the chassis and structural layout. Following the same line of reasoning, with 0.012 kg and 0.011 kg of LD and MD foam needed, €0.418 and €0.383 are required for high impact polystyrene (HIPS) dissolvable filament[173].

As far as items ordered from a third-party source go, electronics components with their respective costs were presented in table 9.3 in chapter 9 and power hardware in table 9.6. Those values are also reported in the cost breakdown structure for production and assembly in figure 14.2. The propellers will be injection moulded by an external provider using a long carbon fiber reinforced nylon thermoplastic composite, impregnating the fiber with the matrix resin to improve performance [25]. Once several items are injection moulded, the effective cost per part is lower than the 3D printed cost per part [174]. Hence, being conservative due to the high volume rates required for the ARTIS throughout the year, the propeller cost was set up as the 3D printing one, at €10.

Having arranged all the necessary materials resources, the production will then proceed in-house, supported by the Vit Natural Robotics SLS printer.

Addition of fixed and variable costs yields a total system cost of €1460.14, clearly below the €2500.00 **AMD-NF-CNST-Cost-01** requirement presented in [6]. figure 14.2 shows how producing and assembling the drone amounts to 14.7 % of the total ARTIS costs. Back-engineering the calculation, the total first year cost of setting up and running a for-profit institution capable of producing, assembling and internationally distributing a maintenance drone for the Lockheed Martin C-130 fuel tank amounts to €10 429.57. Of course, it could be argued that the costs shown in figure 14.2 and figure 14.1 are extremely low to run an established company, but it is believed sufficient for a start-up. During the first year of activity, government subsidies would have to be arranged, alongside potential offers from venture capitalists to survive in a competitive business environment.

As argued at the beginning of this section, a drastic expense drop would be experienced by Lockheed Martin in utilizing the ARTIS solution.

14.2. Return on Investment and Operational Profits

To find out whether or not ARTIS is a profitable investment, this section will calculate the Return of Investment for both the customer Lockheed Martin and the production team of ARTIS, which will be refered to as the company. In order to do this, the following equations are consulted.

For Lockheed Martin, the return of investment is as follows.

$$RoI_{Lockheed2020} = \frac{Current\ Value\ by\ Investment - Cost\ of\ Investment}{Cost\ of\ Investment} = \frac{Saved\ Cost\ by\ investing}{Cost\ of\ Investment} \tag{14.1}$$

In order to calculate the above parameters, the following steps will be taken to find meaningful numbers for this equation.

First, the market price for the product is established. In the first year, ARTIS is planned to be sold for €166 000. Market research from market analysis shows expectancy of selling 50 drones in the first year to Lockheed Martin, leading up to a total income of 50 × 166 000 = €8.3 M. As all of this income is expected to arrive from Lockheed Martin, this value is the cost of investment.
From chapter 4 follows that as their current cost of inspection is calculated to be €38.30 M. Subtracting the cost of investment of €8.3 M, this investment implies a saved cost of €30 M.
Thus, for the customer Lockheed Martin, the following return of investment is expected.

$$RoI_{Lockheed2020} = \frac{€38.3\,M - €8.3\,M}{€8.3\,M} = 361\,\% = 3.6:1 \tag{14.2}$$

For Lockheed Martin, the return of investment is 3.61:1, which is considered an attractive opportunity for the customer.

For the company, the following parameters are required.

$$RoI_{Company2020} = \frac{Current\ Value\ of\ Investment - Cost\ of\ Investment\ Systemcost}{Cost\ of\ Investment} \tag{14.3}$$

For the companies first year, €0.75 M is expected to be required to produce, assemble, and support the operations of the drone. This value is therefore equal to the *cost of investment*. As a way of verifying that this is a sufficient value, 14 % of €0.75 M is planned to go to production and assembly, equating €105 000 per 50 drones and thus €2100 per drone. This is by a margin of 46% than the required €1434 , thus validating that the expected income suffices.
The €8.3 M which Lockheed Invests can be considered as the current value of investment.

$$RoI_{Company2020} = \frac{€8.3\,\text{M} - €0.75\,\text{M}}{€0.75\,\text{M}} = 1007\,\% = 10:1 \tag{14.4}$$

For the company, the return of investment is high for the first year as computed in equation (14.4). This is due to this being a small start-up, and in the initial phase, the team agrees on low wages. Moreover, research and development is supported by assistance from TU Delft. Although the return of investment is high, the company does anticipate a lot of cost to unforeseen issues like software updating, hardware repair, and unexpected manufacturing resource limitations, especially in the first year. This is taken into account and mitigated in the chapter 6, in which therefore the percentage of R&D and tech support is set to 9 %, which is approximately €0.75 M. The design and engineering costs for the year prior to assembly and production will be considered the "zeroth" year, and are discarded in this chapter.

The above plan of expenditure implies that €7.55 M is left to reinvest for the year after. This budget could be used to expand facilities, increase R&D efforts, expand team sizes, or to invest more in promotion and overseas growth, all resulting in an increased drone order and production. It is expected that during reinvestment, the percentages indicated in the CBS will roughly stay the same.

The given analysis is performed with a market volume of 50 drones for the first year. This is based upon calculations in section 14.1.1, and is based upon two important assumptions. Firstly, that Lockheed Martin is our sole customer in the first year of sales, and secondly, the drones only operate inside one of the 13 Maintenance, Repair, Overhaul Centers (MROC). From the market analysis in chapter 4, it is expected that the market volume will grow upon opening up to commercial aviation later years. As it is estimated 300 000 fuel tanks per year are expected. the result is 11 times as many drones sold. This estimate is however on the low side, as there are far more than 13 locations, as well as companies to sell the drone to. The company is aware that this estimate is conservative.

In the second year, it will come to light how the drone performs throughout the year. After approximately five hundred runs, ARTIS's battery is expected to reach its end-of-life. This means that after 500 runs, which equates to roughly one year of inspection, all other components should have proven to be fail-proof and fully functional. If so, the market volume can be increased with confidence, and the company would expand its service to outside of the MROC's. Granted all compondnts work as expected,

Lastly, one must remain critical on the results. The return of investment formula does not take into account compound effects, so over the years the error would accumulate. Also, the conversion rate between USD and EU is assumed constant, while for large sums, this might influence the outcome.

15
SUSTAINABLE DEVELOPMENT STRATEGY

This chapter will detail the approach to sustainability within this project. This will be done in several parts, starting with the general considerations in section 15.1, this also includes health effects. Section 15.2 will detail the materials and manufacturing techniques used in the design, along with their environmental impact. Section 15.3 will then be used to give an overall sustainability score to the design.

15.1. GENERAL CONSIDERATIONS

The main contribution of this project to sustainability is the elimination of the health hazards for the personnel inspecting the tank. The fumes in the JP8 jet fuel are shown to be highly toxic, having adverse effects on liver function [34]. Furthermore, JP8 can also enter the body through the skin [35, 36]. This means that mere gas masks are not enough to sufficiently protect one from the harmful effects. These effects do not only include the aforementioned physical effects but also include neurological degradation, leading to depression and anxiety [37, 38]. The use of ARTIS will cut down the exposure of Lockheed personnel to JP8 and thus greatly improve the quality of life for those tasked with inspecting the fuel tank.

Another contribution is the reduction in time and money spent on tank inspections. This is because the tank inspections currently require several people, as well as constant venting during inspection and venting beforehand. Together with trained personnel. Using ARTIS instead means venting is not required, so there is time saved due to not venting.

15.2. ENVIRONMENTAL IMPACT ANALYSIS

The first step in evaluating the overall sustainability of a product and establishing effective sustainability strategies is to identify the environmental impact a product will have during its lifetime. A useful tool in achieving this is the UNEP MET (Material usage, Energy usage, Toxic emissions) matrix, which provides insight in the environmental characteristics based on the lifecycle stages of a product. As the name implies the MET matrix assesses the environmental profile based on the material usage, energy usage, and toxic emissions, which are represented as separate columns in the matrix, whereas the sustainability performance of the distinct lifecycle stages is enclosed in the rows of the matrix [39]. The exact lifecycle stages that are used in this endeavor are determined based on the relevance of certain lifecycle stages with respect to the product lifetime[1], and are presented below:

1. Selection of materials and components with low environmental impact
2. Optimization of manufacturing procedures
3. Diminishing impact during product utilization
4. Optimization of product durability
5. Decrease of impact of end-of-life system

Since all the materials, energy usage, and toxic emissions are known, the MET matrix can be constructed. The final MET matrix for ARTIS is shown in figure 15.1. This MET matrix can then be used as a tool in establishing the EcoDesign strategy wheel, which will be done in the next section.

[1] It is illogical and ineffective, for example, to include distribution as a lifecycle stage for a product that is manufactured at the location where it is supposed to be used. Distribution in this light can therefore be considered irrelevant with respect to the product lifetime and is therefore not included in the sustainability assessment. The distribution of the components is included in the materials stage.

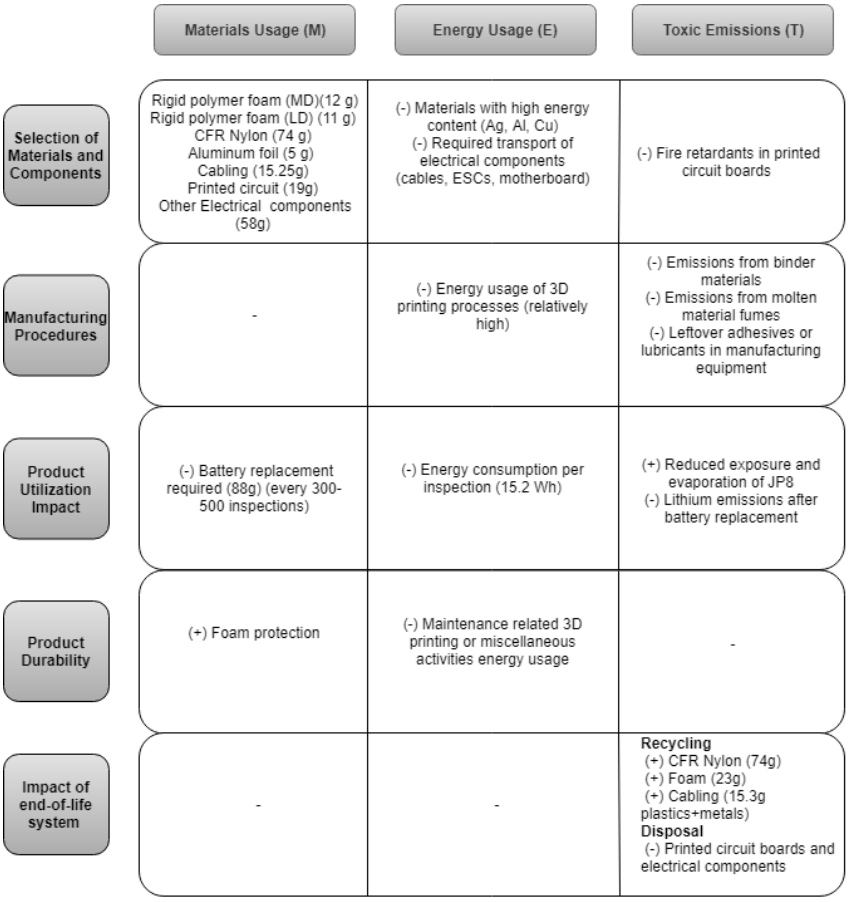


Figure 15.1: MET matrix for the environmental impact analysis of ARTIS

15.3. EcoDesign Strategy Evaluation and Sustainability Scoring

The establishment of the complete environmental profile of ARTIS as displayed in figure 15.1 enables the overall sustainability of ARTIS to be assessed and subsequent strategies that aim to improve the sustainability characteristics to be formulated. In the scope of the UNEP sustainable engineering approach, the so-called EcoDesign strategy wheel is the standard tool to assess sustainability and identify possible sustainability strategies [39]. The EcoDesign strategy wheel visualizes the sustainability score for each lifecyle stage defined in section 15.2 by means of an ordinal scale that runs from one to five, with a higher scoring indicating better lifecycle stage sustainability performance. The EcoDesign strategy wheel therefore provides a clear indication of which lifecycle stages need improvement in terms of sustainability, after which an effective sustainability development strategy can be devised [39].

Evidently, the scoring levels of the ordinal scale that is used for the sustainability assessment of the lifecycle stages need to be defined before a meaningful strategy wheel can be constructed. The definitions of the scoring levels for each of the previously defined lifecycle stages are shown below:

- Selection of materials and components with low environmental impact

Level 1 - Materials used are non-renewable, non-recyclable and highly toxic. Production processes for the raw materials require high amount of energy and have significant toxic emissions.

Level 2 - Materials used are non-renewable, but recyclable and not very toxic. Production processes for the raw materials require high amount of energy and have some toxic emissions.

Level 3 - Materials used are non-renewable, but recyclable and not toxic. Production processes for the raw materials require high amount of energy and have low toxic emissions.

Level 4 - Materials used are renewable, recyclable and not toxic. Production processes for the raw materials require only a moderate amount of energy and have no toxic emissions.

Level 5 - Materials used are renewable recyclable and not toxic. Production processes for the raw materials require only a small amount of energy and have no toxic emissions.

- Optimization of manufacturing procedures

Level 1 - Inefficient distribution of production steps, leading to a large number of production steps, each consuming a large amount of energy. A large amount of non-recyclable waste is produced.
Level 2 - Fairly inefficient distribution of production steps, leading to a moderately large number of production steps, each consuming a moderate amount of energy. A large amount of non-recyclable waste is produced.
Level 3 - Decent distribution of production steps, leading to a moderate number of production steps, each consuming a moderate amount of energy. The amount of non-recyclable waste produced is not very large.
Level 4 - Efficient use of production steps, leading to a fairly small number of production steps, each consuming a small amount of energy. The amount of non-recyclable waste produced is fairly small.
Level 5 - Efficient use of production steps, leading to a very small number of production steps, each consuming a tiny amount of energy. The amount of non-recyclable waste produced is negligible.

- Diminishing impact during product utilization

Level 1 - High energy consumption, with all the used energy originating from energy sources that are non-sustainable. Energy is wasted to an unacceptable extent. A great amount of consumables is needed during operation, with all consumables being non-sustainable. The number of consumables that is wasted is unacceptable.
Level 2 - High energy consumption, where the majority of the energy is derived from non-sustainable sources and only a minority originates from sustainable energy sources. A significant amount of energy is wasted. A large amount of non-sustainable consumables and a small amount of sustainable consumables is required. The number of consumables that is wasted is significant.
Level 3 - Average energy consumption, where the majority of the energy is derived from non-sustainable sources and only a minority originates from sustainable energy sources. Average amount of energy waste. Average number of consumables is needed, where the amount of non-sustainable and sustainable consumables needed is equal. An average number of consumables is wasted.
Level 4 - Average energy consumption, where the majority of the energy is derived from sustainable sources and only a minority originates from non-sustainable energy sources. Low amount of energy waste. The majority of consumables needed is sustainable. A small amount of consumables is wasted.
Level 5 - Low energy consumption, with all the used energy originating from energy sources that are sustainable. A small number of consumables is needed, where all the consumables are sustainable. A small amount of consumables is wasted.

- Optimization of product durability

Level 1 - ARTIS is highly unreliable and has low durability. Repairing parts, as well as performing maintenance, is impossible. ARTIS is not intuitive to use.
Level 2 - ARTIS is unreliable and not very durable. Repairing parts, as well as performing maintenance, is difficult. ARTIS is not very intuitive to use.
Level 3 - ARTIS is fairly reliable and does not break very easily. Repairing parts, as well as performing maintenance, is not difficult. ARTIS is also intuitive to use.
Level 4 - ARTIS is reliable and does not break easily. Repairing parts, as well as performing maintenance, is easy. ARTIS is also very intuitive in its use.
Level 5 - ARTIS is highly reliable and does not break under normal use. Repairing parts, as well as performing maintenance, is very easy and does not require any training or instructions. ARTIS is also very intuitive in its use.

- Optimization of end-of-life system

Level 1 - Refurbishment is not possible, nor is reusing the product. Recycling is impossible and a significant amount of harmful fumes are introduced in the environment upon incineration.
Level 2 - Refurbishment is not possible, nor is reusing the product. Recycling is possible but incineration introduces a significant amount of harmful fumes in the environment.
Level 3 - Refurbishment is possible, but the product cannot be reused. Recycling is possible and incineration introduces a small amount of harmful fumes in the environment.
Level 4 - Refurbishment as well as reusing the product is possible. Recycling is possible and incineration introduces a small amount of harmful fumes in the environment.
Level 5 - Refurbishment as well as reusing the product is possible. Recycling is possible and incineration introduces no harmful fumes in the environment.

At the beginning of the design process, it was additionally decided to establish minimal requirements for the sustainability performance of the lifecycle stages. These minimal sustainability requirements were set to act as a benchmark for the identification of lifecycle stages that are insufficiently sustainable. The minimal sustainability requirements can be expressed in terms of the above defined ordinal scale for sustainability scoring and then be displayed in a strategy wheel, which is shown in figure 15.2 together with the final score which will be elaborated upon below. It was decided to set the minimal sustainability requirements to the levels shown in the aforementioned figure for the respective lifecycles for the following reasons:

1. Selection of materials and components with low environmental impact
    - Since the drone has to operate within a dangerous environment, it is more important that the selected materials do not deteriorate within the environment than that they are sustainable in their production. Materials incapable of operating within such an environment would be less sustainable due to them having to be replaced often. A Level 3 minimum sustainability requirement was set for that reason.
2. Optimization of manufacturing procedures
    - It is projected that the drone will be produced using additive manufacturing, where ARTIS in its entirety is likely to be printed in one piece. This makes concurrent production of components challenging, leading to a relatively complex production and manufacturing steps. Moreover, depending on the type of additive manufacturing procedure that is used, the energy usage might be considerable [175]. As such, it was decided to set a minimal sustainability requirement of 2 for the manufacturing lifecycle stage.
3. Diminishing impact during product utilization
    - ARTIS is projected to not produce any emissions during operation, and has been designed with maximizing flight efficiency in mind through e.g. increasing the propeller diameter to as large as possible within the given size constraints. It is expected, however, that the battery ought to be replaced after a certain amount of time to maintain optimal performance. Evidently, replacing the battery requires careful handling of the harmful components it contains in terms of sustainability. This therefore offsets the benefits of the respectable flight performance, and since no other user requirements were found that specifically relate to the sustainability of the operational phase, a minimal sustainability score of 3 is set.
4. Optimization of product durability
    - The user requirements that have been provided do not disclose any details on the durability or related notions. Considering the manufacturing procedures that have been outlined, it is assumed that the modularity will be low and maintenance will involve some degree of difficulty. Accordingly, a minimal sustainability score of 2 was set.
5. Decrease of impact of end-of-life system
    - Similar to the previous lifecycle stages, no user requirements were provided that touch upon the disposal or incineration at the end-of-life. However, since the drone is likely to include numerous

plastics that allow for recycling, but nevertheless release some degree of harmful fumes upon incineration, it was decided to set the minimal sustainability score to 3.

After the minimum sustainability levels have been defined, each level needs to be assigned an actual score to be filled in in a wheel, done in figure 15.2. The actual scoring for ARTIS is disclosed below:

1. Selection of materials and components with low environmental impact
    - A score of 3 was found - The biggest advantage for this stage is the fact that there is a very low amount of toxic materials in ARTIS. The entire structure at the very least is completely non-toxic. There is however a high likelihood of there being some toxic materials in the printed circuit boards. Furthermore, the pre-manufactured electrical components and cables require transports which also leads to emissions. Finally the metals in the cables and circuitry require a relatively high amount of energy to be made. Moreover, the selected CFR Nylon is 3D-printable as opposed to e.g. aluminum, which is more sustainable and in part the reason why CFR Nylon was selected over aluminum.
2. Optimization of manufacturing procedures
    - A score of 3 was found - The biggest advantage for this stage is the low waste of the selected 3D printing technique - SLS - relative to traditional, substractive manufacturing methods. Moreover, SLS requires no additional support structures, and thus leaves less waste compared to other 3D printing techniques, and enables the production of thin-walled structures [175]. This however comes with the disadvantage of a somewhat higher energy usage. For this reason it was decided to stay conservative.
3. Diminishing impact during product utilization
    - A score of 3 was found - The most important positive aspect for this stage is the advantage to the health of the personnel working on inspecting the tank as described in section 15.1. Also described in section 15.1 is the benefits of not needing to vent the tank. Inspecting the tank still costs energy but significantly less than venting. There is of course the disadvantage of the battery replacement, but the battery has a relatively high lifetime (300 to 500 charge cycles as described in section 9.3.1.1). The battery is considered to be largely outweighed by the health benefits, but is definitely not negligible. Looking at the battery usage for all C-130 inspections (26 000 per year), an approximate of 200 batteries per year was found.
4. Optimization of product durability
    - A score of 3 was found - The biggest contribution to the durability is the foam, it does not only protect the tank from damage but also enhances the durability of ARTIS by protecting its structure. It is fairly difficult however to replace the battery, which lowers the score for this stage. This is paired with a low degree of modularity.
5. Decrease of impact of end-of-life system
    - A score of 4 was found - The structure is fully recyclable, the metals in the cabling are also fairly easy to recycle. The only disadvantage is the recyclablity of the electrical components. It is possible but not very easy [176].

In accordance with the above attributed scores, a final design strategy wheel can be created, which is shown in figure 15.2. Figure 15.2 also displays the minimal sustainability requirements, and therefore demonstrates that all the minimal sustainability requirements have been met. It is therefore concluded that the product lifetime of ARTIS contributes sufficiently to sustainability in light of the five described lifecycles.

15.4. Future Sustainability Prospects

The discussion regarding sustainability at this stage has not yet considered the improvements that can still be made in further design iterations. Nevertheless, several lifecycle stages have been identified in which

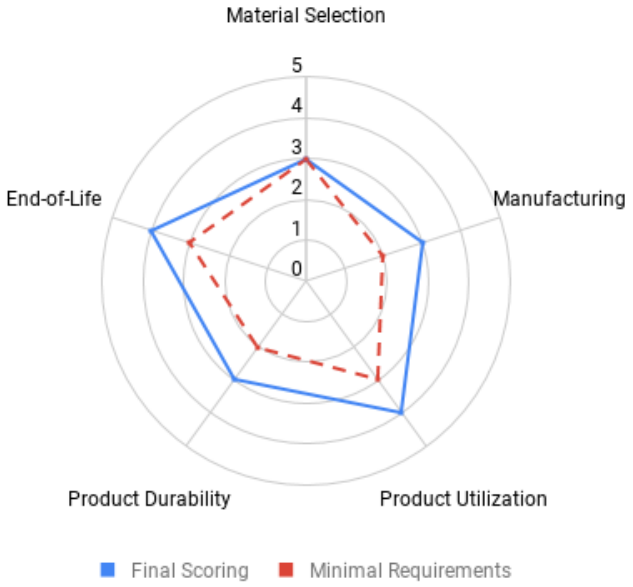


Figure 15.2: Final sustainability scoring for ARTIS versus the minimal sustainability requirements

steps can still be taken to improve their sustainability score. Most notably, in increased level of modularity in future versions has the prospect of making the product utilization phase more sustainable. Partly through making it easier to tear ARTIS apart, having smaller parts which are easier to recycle. Strides can additionally be made in terms of sustainability for the product utilization phase by establishing an NFC transfer for the captured video data inside the fuel tank. The current proposed method would require a significant amount of available data storage space on the processing platform to facilitate the storage of all the captured data. Depending on how long the data is kept, a significant amount of this stored data could turn out to be waste. This problem can be circumvented by streaming the video feed from inside the fuel tank in real-time to the human operator, as it would generally require less data storage space on the processing platform.

Currently CFR Nylon is used, since this is stronger than regular nylon which is needed as described in section 8.3. This nylon is however not easy to recycle due to its high carbon content. It is however definitely possible [177], hence the meeting of our minimal sustainability requirements. Regardless, it would be more optimal if another material with the same strength was found to allow for a more sustainable design.

16

Risks Compliance and Management

Risks severely affecting the safety of performing autonomous fuel tank inspections arise when performing the design of a complex system such as ARTIS. The present chapter will overview the mitigation techniques being proposed thus far, alongside a post-mitigation risk map, as well as further considerations to be implemented in future design iterations.

16.1. Risk Mitigation Approaches

Throughout the design process, the risks discussed in chapter 6 have been considered and mitigated. Having consistently and integrally considered risks, ARTIS has been designed with inherit safety in mind. In this section the result of the mitigation of risks is summarized so as to have a concise overview of how the risks have been managed. Those risks that have not been mitigated or have not been dealt with to a satisfactory level, will be further discussed in the ensuing section.

A1. Errors with systems integration

This risk has not been mitigated during the design stage, therefore it shall be dealt with in the upcoming section 16.2.

A2. Explosion induced by system malfunctioning

This risk has been dealt with on three occasions. Firstly in section 9.3.1.2 where the likelihood has been mitigated by the use of a battery protection circuit. In section 9.4 it has been approached by tackling static charges and thus the likelihood has been mitigated. Finally maintenance practices as discussed in section 13.1.2 further prevent such anomalous events. As a result of the combined features, the likelihood has been mitigated two levels down to low likelihood.

A3. Induced sparks during operations

Four approaches have been proposed to mitigate the mentioned risk, firstly in section 8.1.1 and section 8.2 by using brushless electromotors and in section 9.3.2.3 by including all wiring internally within the structural frames. Finally in section 9.4 via the grounding subsystem, thus lowering the likelihood to low.

A4. Broken parts due to collisions remain undetected inside fuel tank

The mitigation of this risk was based on post-data processing activities analyzing any potentially suspicious behaviour, overviewed in section 13.2.2, further lowering its likelihood from medium-low to low.

A5. Communication failure

Different suggestions have been made regarding possible communication failures. In section 9.2.2.1 the use of two SD slots for on board data storage was proposed, whereas section 13.1.2 and section 13.2.2 dealt with preventive maintenance inspections. The likelihood of the mentioned risk is thus lowered from medium to low.

A6. Damage to fuel tank structure due to collisions

The addition of a duct and protecting structure around each propeller to protect the blade tips was explained in section 8.3, thus reducing the risk of damaging the fuel tank. Furthermore, maintenance activities and safe mode approaches were also discussed, in section 13.1.2 and section 13.2.2, respectively. From high likelihood, as shown in figure 6.1, the risk is lowered to medium-low, as seen in figure 16.1.

C1. Vehicle gets stuck inside fuel tank

The risk has been dealt with via maintenance and operational activities in section 13.1.2 and section 13.2.2, with the severity being reduced to medium.

C2. Loss of stability and control

Extensive tuning of the flight controller, presented in section 10.5.3, allows for a relatively fast response for attitude corrections and minimization of collision with the walls of the fuel tank while updating position

and altitude. The loss of stability and control, being a major point of concern, will be further investigated in section 16.2, with extensive explanations of so-far neglected aerodynamic interference due to wall proximity.

C3. Data corruption during on-board processing

To mitigate this risk, a controlled landing mitigation feature was mentioned in section 10.3.1, bringing its severity to a medium level. Furthermore, the maintenance activities overviewed in section 13.1.2 bring the likelihood from medium to medium-low.

C4. Inaccurate position knowledge

Section 10.3.1 assessed the impact of adequate calibration of cameras, IMU and ToF sensors to mitigate this likelihood of the risk. Furthermore, computing position and orientation information with the IMU when computer vision is not sufficient and applying the possibility of controlled landings with the ToF sensors mitigate the severity. In conclusion the severity is lowered two levels to medium-low and the likelihood to medium.

C5. Inaccurate obstacle avoidance and path planning

The integration of flight distance sensors (FDS) within the navigation framework (see section 10.3.1) and the addition of margins to the obstacle coordinates as explained in section 10.4.3 substantially reduce the risk likelihood from medium-low to low.

C6. Inaccurate detection of fuel tank damage/corrosion

Validation of the Simulink color detection algorithm as explained in section 12.4.1.2 and post-data processing activities (see section 13.2.2) help mitigate the risk. Nevertheless, this risk has not been discussed to a satisfactory level, therefore further approaches to mitigate the risk will be discussed in the following section 16.2.

E1. Electronics subsystem failure

The use of a battery protection circuit significantly reduces the severity of electronics subsystem failure as outlined in section 9.3.1.2, alongside preventive maintenance inspections activities (section 13.1.2 and section 13.2.2), thus bringing the risk severity to a medium level.

P1. Power required exceeds available power

This risk has been considered mainly when selecting the motors in section 8.2. Since the motors consume 90.7% of the available power they are the most crucial component when managing this event. An appropriate choice of motor in section 8.2 reduces the likelihood that the motor will cause the realization of risk P1. The risk is thus mitigated one level to medium-low likelihood.

P2. Propulsion sub-system failure

This risk has not been mitigated to a satisfactory level during the design stage, therefore it shall be dealt with in the upcoming section 16.2.

P3. Insufficient range and endurance during flight operations

Chapter 13 and section 13.1.2 mitigate this risk by taking into account the charging of the batteries before inspection and the checking of battery conditions before and during the inspections. The likelihood is reduced to low and the severity is also mitigated to medium since as discussed in chapter 13 ARTIS will end the inspection once the battery level is too low.

PL1. Image quality not sufficient for analysis

Risk PL1 has been mitigated in section 9.2.1.1 by the choice of a suitable camera with a high resolution and by making sure that the constraints for sufficient resolution are determined and complied with. Furthermore, the use of four cameras ensures redundancy such that in the case certain cameras are not producing the required quality, the others can serve as replacements. Finally, operational flags and watchdogs prevent the use of poor image quality use for inspection. The severity of this risk is thus lowered to medium.

PL2. Payload mass or power exceeds set budget

This risk was mainly approached by the use of a mass and power estimation tool and by the choice of electronic components capable of meeting the requirements with a margin on performance. In section 9.2.1.4

it is mentioned how some of the electronic components require low power and thus help mitigate the likelihood of this risk to medium.

PL3. Payload subsystem failure upon impact

The structural elements of ARTIS were designed in section 8.4 with payload safety considerations in mind and by keeping the cameras protected as well as the electronics compartment and other sensors, thus mitigating the likelihood of this risk to low. The controlled landing feature mentioned in chapter 10 and chapter 13 lowers the consequences of an impact to medium severity.

S1. Undetected manufacturing flaws

Risk S1 has been mitigated by the choice of manufacturing process and operational and maintenance procedures. In section 8.3 the choice of 3D printing is discussed as beneficial tool for replicable manufacturing thus avoiding anomalous flaws. Finally, in section 8.4 and section 13.1.2 the risk is approached through preventive inspection and maintenance practices thus also mitigating the likelihood of occurrence. The risk has thus been lowered one level to low likelihood.

S2. Unexpected manufacturing resource limitations

As a manufacturing related risk, the mitigation of risk S2 has been discussed in section 8.4. The likelihood of this risk has been reduced by the choice of 3D printing as the main manufacturing technology, a readily available and easily acquired tool for manufacturing. Furthermore in chapter 14 the severity is mitigated by taking into account possible limitations and delays and thus being prepared for such events. The risk is therefore mitigated to medium likelihood and low severity.

S3. Structural sub-system failure upon impact

This risk has been dealt with mainly in section 8.4 where the likelihood of collisions has been determined and minimized. Furthermore in section 13.1.2 maintenance practices prevent ARTIS from being deployed with a fragile structure thus also mitigating the likelihood of the risk. Finally the thorough verification and validation of the structural frame in chapter 12 increases the confidence in the design. The likelihood of this risk is reduced by one level to medium-low. Furthermore, the structural design being made of few individual items and the foam keeping elements together in case of collision, the severity decreases from high to medium-high. As future concerns, actual physical tests should be performed to completely validate the performance of the structure.

Post mitigation risk map

Some preventing measures have been proposed to decrease the severity and likelihood of occurrence of the risks. The post-mitigation map shown in figure 16.1 suggest the presence of risks that shall be more carefully dealt with (those in the yellow and red areas). As of now, many approaches to risk mitigation have been outlined throughout the ARTIS design explanation. Yet, for instance, the severity of risk **A2** (i.e.explosion induced by system malfunctioning) has no immediate solutions, as explosion would imply catastrophic failure of the mission and severe damages to the entire operating environment. Section 16.2 proposes strategies and mitigation techniques the team would implement in late stages of the design for those risks that have not been mitigated so far (seen in orange in figure 16.1. These risks have not been mitigated as there was no immediate solution during the design stage and no feasible design implementations at this early stage of the project were manifested.

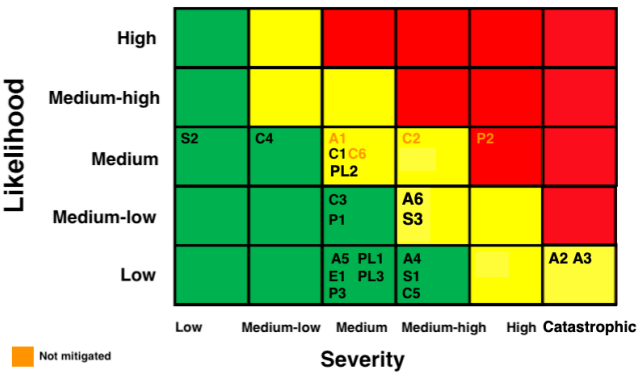


Figure 16.1: Post-mitigation risk map

16.2. Further Considerations for Risk Mitigation

Some risks have been left unmitigated at the current state of the design, either due to the complexity of the risk or the fact the risk was only identified at a late stage, which made formulating an appropriate risk mitigation strategy in the remaining scope of the project difficult. This section presents a few risks that have been left unmitigated in an unsatisfactory manner, as well as possible strategies in which these risks can be mitigated in future design iterations.

A1. Errors with system integration

Assembly and integration during a second design iteration will consist of a model validation approach and of structural integrity verification activity. The former will be composed of 4 stages, starting with a simulation of the ARTIS inspection in a pure host PC. Having satisfied the requirements of such phase, testing aimed at verifying software integration into the ARTIS onboard computer will be performed. The third stage consists of the integration of the electronics and power hardware for interface testing which, if satisfactory, will be followed by the final stage, a full functional integration of the ARTIS subsystems' software and hardware.

To verify the mechanical integration of components, the ARTIS assembly shall be reliable and practical. An early indication of the systems structural integrity will be performed using an available shaker at the faculty of Mechanical, Maritime and Materials Engineering at the Delft University of Technology. A prototype ARTIS version shall be used as input, so as to imitate vibrations loads.

C2. Loss of stability and control

As explained in section 8.1, it was found that the stability of the drone is hampered considerably in the vicinity of fuel tank walls. Hence, the current discussion aims at mitigating the loss of stability and control due to wall proximity aerodynamic interference. This behavior, which was confirmed by a test that comprised flying an existing drone into a fuel tank, demonstrated that the drone has the tendency to be sucked towards the wall, which may result in a wall collision .

The suggested approach to mitigating the risk of an aerodynamic interference induced loss of stability (see risk C2) close to walls is to expand the current linear control system by gain scheduling in order to provide better response characteristics to the nonlinear behavior in the proximity of the walls. Constructing an appropriate gain schedule first and foremost requires the identification of the scheduling variables that different gain sets have to be designed for. A switch that is implemented inside the control loop is then responsible for selecting a particular set of gains based on the values of the scheduling variables, and a simplified control loop that demonstrates this principle which should be implemented for the six PID controllers is displayed in figure 16.2 [7, 178].

Logically, the scheduling variables that ought to be designed for should relate the position of the rotors with respect to a set of walls. Due to the limited size of the confined wing section compared to the size of ARTIS, three scenarios that stipulate a particular wall configuration proposed by [179] were deemed appropriate:

- *In Channel Effect (IChE)*. Two walls are present, with one being downstream of the rotors and one being upstream of the rotors [179].
- *In Low Corner Effect (ILoCE)*. Two walls are present, with one being downstream of the rotors and one being perpendicular to the plane of the rotors [179].
- *In Upper Corner Effect (IUpCE)*. Two walls are present, with one being upstream of the rotors and one being perpendicular to the plane of the rotors [179]

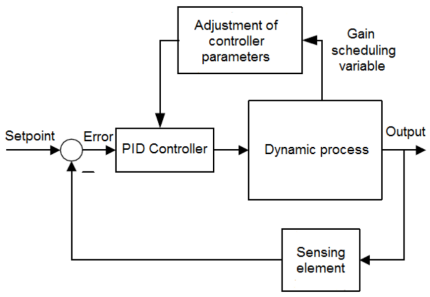


Figure 16.2: Gain scheduled enhanced control loop [7]

The resulting flow disturbances from the three above mentioned scenarios are in turn a function of the ratio of the distance between the wall and the rotor, perpendicular to the rotor plane (h/R), and the distance between the wall and the rotor, parallel to the rotor plane (d/R) [179]. The ratios h/R and d/R can as such

be used as scheduling variables, where the magnitudes of these variables first determine which of the three wall configuration should be chosen[1]. The appropriate set of gains for a wall configuration can then be determined using a gain scheduling surface in terms of h/R and d/R. Such a gain scheduling surface that can e.g. be used to select a set of gains for the ILoCE of IUpCE scenario is shown in figure 16.3, where ideally the transient between two sets of gains should be smooth to prevent aggressive transient responses [178].

P2. Propulsion sub-system failure

There are two main causes for propulsion subsystem failure: Damage to the propellers or to the motor/propeller joints and motor failure. As discussed in section 13.1.1 the reliability of electric motors is relatively high, especially since ARTIS uses brushless motors, and damage to the propellers has been mitigated by the use of ducts and other protective structural features (see section 8.4). Nevertheless, one of the most crucial elements that can fail is the propeller joint which after various operational cycles will suffer from fatigue. Therefore the structural design of this joint should be further investigated and tested in order to determine the failure rate and reliability of the propulsion subsystem. It should also be outlined that during inspection of the drone before deployment, special attention should be given to the condition of propeller integration, and the propellers should be replaced if needed.

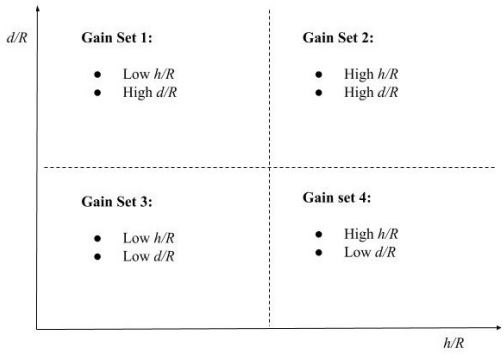


Figure 16.3: Exemplary gain scheduling surface for ensuring proper control and stability in the vicinity of a set of walls (ILoCE or IUpCE)

C6. Inaccurate detection of fuel tank damage/corrosion

As mentioned in section 9.2.1.2, color detection is currently done off board with a simple algorithm that compares HSV values for all pixels in an image. This approach can be improved via machine learning solutions, in the form of neural networks trained on detection datasets. Also, the thresholds for detection can be set relatively high so that ARTIS will not miss a detection easily. This comes at the cost of having more false detections, but as a human operator will always look at the footage to check for actual green masks at the locations where the algorithm detects something, false detections are not considered a big problem.

Missed Detection Due to Path Deviations

This risk had not been identified in chapter 6 and was instead identified when reviewing the design stage. Thus, it is managed in this section.

Currently ARTIS performs the inspection on a pre-planned path, ensuring the entire tank is inspected, as outlined in section 11.2.2. However, if ARTIS were to encounter an obstacle while traversing this path, it will have to deviate. This deviation might lead to ARTIS not inspecting a part of the tank. Furthermore, sudden inputs to the control subsystem could lead to a blurred inspection, incapable of capturing corrosion marks. In future design iterations, the ARTIS will be equipped with mechanisms capable of sending red flags if a surface were to be left not inspected. At this stage in the design, the ARTIS would just avoid the obstacle and continue following the preset path.

[1] It can e.g. be argued that IUpCE should be used when, for example, h/R is greater than half the height of the confined section while $d/R > 0.05$, while IChE can for example be used when $d/R > 0.05$, regardless of the value of h/R.

17
REQUIREMENTS COMPLIANCE

This chapter will look back to the requirements stated in chapter 5 and check whether they are met or not. The requirements compliance table is presented in table 17.1. This table states the subsystem requirements stated in section 5.2, the category 1 and 2 driving requirements from [6] and the driving requirements from [6]. These requirements were chosen as they are the most important for this mission and all play a vital role for the missions success. The remaining requirements stated for this mission, labelled as 'other' in [6], are not analyzed in this chapter for brevity sake and because they are less important for mission success.

The left most column of table 17.1 shows the requirements to be analyzed. The other columns show the categories of compliance: Fully met, partially met, not met and to be tested/investigated. A requirement is labelled 'fully met' if that requirement is adhered to via measures explained in this report. An example of this is **AMD-NF-CNST-Mass-01 (S,Y)**: *The drone's total mass shall not exceed* 300 g. The ARTIS was sized and designed to meet this requirement and as a result, the final ARTIS mass turned out to be 285 g. This thus is a clear example of a requirement that is fully met by designing the drone specifically for that requirement.

Only one requirement is placed in the 'partially met' category. This category is used for requirements that changed a bit during the design phase. The originally stated requirement is thus not exactly met, but as this requirement changed during the design phase it is not placed in the 'not met' category. This requirements will be stated below, including an explanation as to why it is not met.

- **AMD-NF-LFCC-Bat-02**: *The drone shall be able to notify the operator if the battery charge level gets below* 5 %. Initially, 5 % battery charge was assumed a sufficient charge level to notify an outside operator. This would correspond to 32 seconds of flight time, which was considered to low to safely return from every location in the tank. Therefore, the drone now notifies an operator when the battery level drops below 10 %, which translates to 64 seconds of flight time (or approximately 12.8 m. This is sufficient to fly through all holes to the closest exit if necessary with reasonable margin. Also, if the design were to change to a flight time of 10 minutes, 10% battery charge would still give a flight time of 60 s, which is still sufficient to fly out of the tank.

In the ARTIS design, no requirements were considered to be completely unmet. All set requirements were either met, partially met, or to be investigated. This last category will now be further explained.

Lastly, nine requirements are placed in the 'to be tested/investigated' category. These requirements are likely met as throughout this report several measures have been taken to comply with them, but no guarantee on meeting them is assured. An example of this is **AMD-NF-CNST-Sfty-02 (S,D1)**: *The drone shall not create sparks during inspection.* In section 9.4, the design process of eliminating charge in the drone is explained. Still, one cannot say with 100 % certainty that ARTIS will never create a spark. For that reason, several tests are proposed in figure 12.27 to increase to confidence on complying with this requirement. Only after extensive testing and investigating this requirement can be moved to 'fully met'. Another example is **AMD-NF-CNST-Cost-02 (S,Y)**: *The supporting systems cost of the drone shall not exceed* €5000 p*er drone.* This requirement is again likely to be met, but as no actual drone operations have taken place, one cannot yet be fully confident that the operational costs will stay below €5000 per drone at all times. Further investigation into this aspect is required.

Overall, it can be seen in table 17.1 that most requirements are fully met. Only one requirement is partially met as the this requirement was changed during the design phase. A few requirements should still be further tested or investigated to ensure compliance. The table shows that the ARTIS design followed the strategy of designing based on the requirements well. If after testing it is determined that all blue requirements can be moved to the green 'fully met' category, the ARTIS meets all requirements that are considered important for the mission success.

Table 17.1: Requirements compliance table

Requirement	Fully met	Partially met	Not met	To be tested/investigated
General requirements:				
AMD-NF-CNST-Sfty-03 (D1)	✓			
AMD-NF-CNST-Sfty-02 (S,D1)				✓
AMD-F-ST&M-Join-03 (D1)	✓			
AMD-NF-CNST-Regu-02 (D1, K)	✓			
AMD-F-CNST-Material-01 (D1)	✓			
AMD-NF-CNST-Size-02 (S,D2)	✓			
AMD-NF-CNST-Size-01 (S,D2)	✓			
AMD-NF-CNST-Sfty-04 (S,Y)				✓
AMD-NF-CNST-Sfty-05 (S,Y)				✓
AMD-F-C&T-Grst-01 (Y)				✓
AMD-F-C&T-Grst -02 (Y)	✓			
AMD-NF-CNST-Cost-03 (Y)				✓
AMD-NF-CNST-Cost-01 (S,Y)	✓			
AMD-NF-CNST-Cost-02 (S,Y)				✓
AMD-NF-CNST-Mass-01 (S,Y)	✓			
Aerodynamic requirements:				
AMD-F-AERO-Ent-01 (D2)				✓
AMD-F-AERO-Thr-01	✓			
AMD-F-AERO-Duc-01	✓			
AMD-F-AERO-Thr-02	✓			
Structural requirements:				
AMD-NF-CNST-Sfty-01 (S,D1)	✓			
AMD-NF-CNST-Sfty-06 (S,D1)	✓			
AMD-NF-CNST-Sfty-07 (S,D1)	✓			
AMD-F-ST&M-Join-01 (D1)	✓			
AMD-F-ST&M-Imp-01 (S,Y)	✓			
Electronics and power requirements:				
AMD-F-P&P-Endu (S,Y)	✓			
AMD-NF-LFCC-Dura-05				✓
AMD-NF-LFCC-Bat-05	✓			
AMD-NF-LFCC-Bat-02		✓		
AMD-NF-LFCC-Bat-03				✓
Payload requirements:				
AMD-F-PAYL-Cam-01 (D2)	✓			
AMD-F-PAYL-Cam-02 (S,Y)	✓			
AMD-F-PAYL-Cam-03 (Y)	✓			
AMD-F-PAYL-Ill-01 (Y)	✓			
GNC requirements:				
AMD-F-GN&C-Navi-02 (D2)	✓			
AMD-F-GN&C-Navi-03 (Y)	✓			
AMD-F-GN&C-Cont-05 (Y)	✓			
AMD-F-GN&C-Guid-05 (Y)	✓			
AMD-F-GN&C-Guid-03 (Y)	✓			
AMD-F-GN&C-Navi-01 (Y)	✓			
AMD-F-GN&C-Guid-02 (Y)	✓			
AMD-F-GN&C-Guid-01 (Y)	✓			

18

Conclusion and Further Research

Human inspection of aircraft fuel tanks today is a wearying and error-prone manual task. A cost of €38.3 millions per year is estimated for Lockheed Martin, who commissioned an autonomous inspection system, which is presented in this report. By this mean, DSE Group 05 at the Delft University of Technology aims to prove that there is a more efficient and reliable way to perform hazardous inspections. The ARTIS is an unmanned aerial vehicle capable of autonomously inspecting the inside of empty fuel tanks in aircraft wings, expecting to save €30 millions to Lockheed Martin in its first year of operation.

Characterized by a total mass of 285 g, and requiring 85 W of power for its operations, mostly due to the demands of the 4 motors, the ARTIS is capable of flying for 10 minutes and 40 seconds before battery charging must occur. Considering all necessary operational tasks, one full inspection is expected to last around 236 minutes including data handling and battery charging, which is substantially lower than the downtime required for venting of hazardous gases in the tanks when regarding a human inspection.

As far as the ARTIS materials are concerned, the duct, struts and the compartment exposed to direct airflow from the propeller will be covered in a 0.016 mm aluminum layer, making the parts conductive, aiding static electricity dissipation. Also, polymer foam will be placed around the ducts for protection. Moreover, the structural frame and propeller shall be made out of CFR Nylon.

The ARTIS is also equipped with 4 cameras, providing stereovision and 180° view of the tank, an IMU, and 6 times of flight sensors to measure the distance to surrounding walls. For the autonomous flight, the IMU measurements will be integrated with the computer vision algorithms, aimed at minimizing the error with the ground truth, resulting in position estimates. Furthermore, the stereo images generate a depth map leading to information about grid occupancy and obstacle locations. Having detected obstacles, the A^* path planning algorithm will provide the shortest collision-free path from start to the target location. The required manoeuvres, proper flight control and stability will be ensured by virtue of a cascaded controller consisting of six PID controllers. Furthermore, all GNC algorithms will be performed on board, with video data being stored on a micro SD chip for later retrieval.

Thorough verification and validation activities have been performed to assess the correctness of the developed programs and models. In particular finite element analysis in Ansys successfully verified the design, whereas a full 3D simulation environment served as a validation framework for the guidance, navigation and control programs. During the entire design process risk mitigation was accounted for. As most important aspect was setting up a further testing plan for mitigating the risk of inducing sparks during operations **A3**. Further design iterations will aim at continuously improving the performance of the ARTIS design, which, at the present stage, is believed to be a reliable for fuel tank inspections.

Project Development and Further Research

To ensure the design process would progress smoothly, system engineering methods were used throughout all design stages. An example of this is the mass and power iteration tool that was used to update the whole design if one parameter were to change. In addition, team meetings and small talks were used to notify the team when design iterations occured. In this way, the team always was fully informed, leading to a coherent design.

Now, this section will elaborate on the design & development logic after this report. These tasks can be performed by the client, Lockheed Martin, if they would like to pursue this project and actually use it for inspecting their aircraft. Running in parallel to all steps described below, a different team can work on Research & Development to further improve the drone and research new innovations that are applicable to the design. An example is further designing the addition of slots for the SD cards so that they are easily retrievable. Further aspects that need to be further developed include the outside computer, the human interface of the mission and the inclusion of more advanced electronics. Also, the scalability of the system to other

mission needs can be analyzed, for example using the drone to inspect other aircraft fuel tanks or for other inspection procedures.

With the detailed design of ARTIS finished, the first logical step is to build a prototype. All required materials, manufacturing tools and off-the-shelf parts should be acquired, after which the manufacturing of new parts can be done. With all parts ready, they can be assembled into subassemblies, which are after system verification assembled into the prototype.

Once the prototype is constructed, it can be subjected to testing in order to validate the predicted behavior. Five different categories of tests should be performed. First, propulsion tests allow to check if the propulsion system provides enough thrust for hover and control. Second, with static charge tests it can be checked if the risk of spark generation is mitigated sufficiently in order to operate safely in an explosive environment. This also includes testing of any applied explosion proofing, such as the anti static agent. Third, an endurance test should check if the drone can fly long enough to perform the inspection efficiently. Fourth, the GNC system should be tested to observe the ability of the drone to perform an inspection autonomously. Fifth and last, structural tests should be performed to inspect possible structural damage during collisions, both on the drone and the environment. For each of these test categories, the testing procedures should be determined strictly, after which the testing environment can be setup and the necessary testing equipment can be acquired. When all the previous steps are completed, the tests for each category can be performed. The test results can then be analyzed and compared to the predicted behavior and requirements. Any possible design flaws, such as insufficient thrust, catastrophic structural failure or poor endurance, will become apparent at this stage.

If the design flaws that are uncovered during the prototype tests are deemed detrimental to the performance of the drone, the design will have to be reiterated. To solve these flaws, one can either improve the design choices that are currently implemented, select different design options or a combination of both. After the design iteration, a new prototype can be built and tested, with the same procedures described above. This iteration process can be repeated as often as necessary.

Once the design has been iterated enough, a test in a real operating environment (i.e. fuel tank with fumes) can be performed in order to validate the product. Again, a clear testing procedure needs to be determined before performing the actual test. With this test the overall product performance is validated. Afterwards this validated product can be sent to the authorities to get regulation certification.

With the final product validated, a market plan can be set up and executed. As discussed in the market analysis, the drone could potentially be sold to other aircraft manufacturing and servicing companies. The most important aspect is the yearly budget, which comprises projecting several incomes and expenses: production cost per unit, production volume, transport costs, turn-over per unit, return on investment, etc.

Once the validation of the product is done and the yearly budget is set, production will be the following step. During the entire production process the management will keep close contact to monitor this stage of the project. First the different parts that will be custom made are identified and specific part tolerances are set, keeping in mind the manufacturing process used for each specific part. Then after identification several manufacturers will be selected and contacted. These will produce the parts, check if the quality is coherent with the tolerance set and the correct quantity is ready for sending. Finally the parts that will be sent to the location where the assembly will take place.

Before actual assembly can start a manual must be set-up to guarantee the quality and integrity of the final product. Moreover an assembly working environment must be arranged and the needed tools must acquired. Then when all the parts arrive at the location assembly can start and the final product named ARTIS is ready for software implementation. Afterwards it can be sent to the customers.

After the drone has been in operation for one month, a quarterly evaluation can be performed, where input from customers can be gathered. This feedback can be used to update the design and be relayed to the R&D team.

All the design & development steps provided above are summarized in figure 18.1. The most important blocks are colored and explained in more detail on the bottom of this figure. The Gantt chart describing the project continuation including time scale is found in the page after.

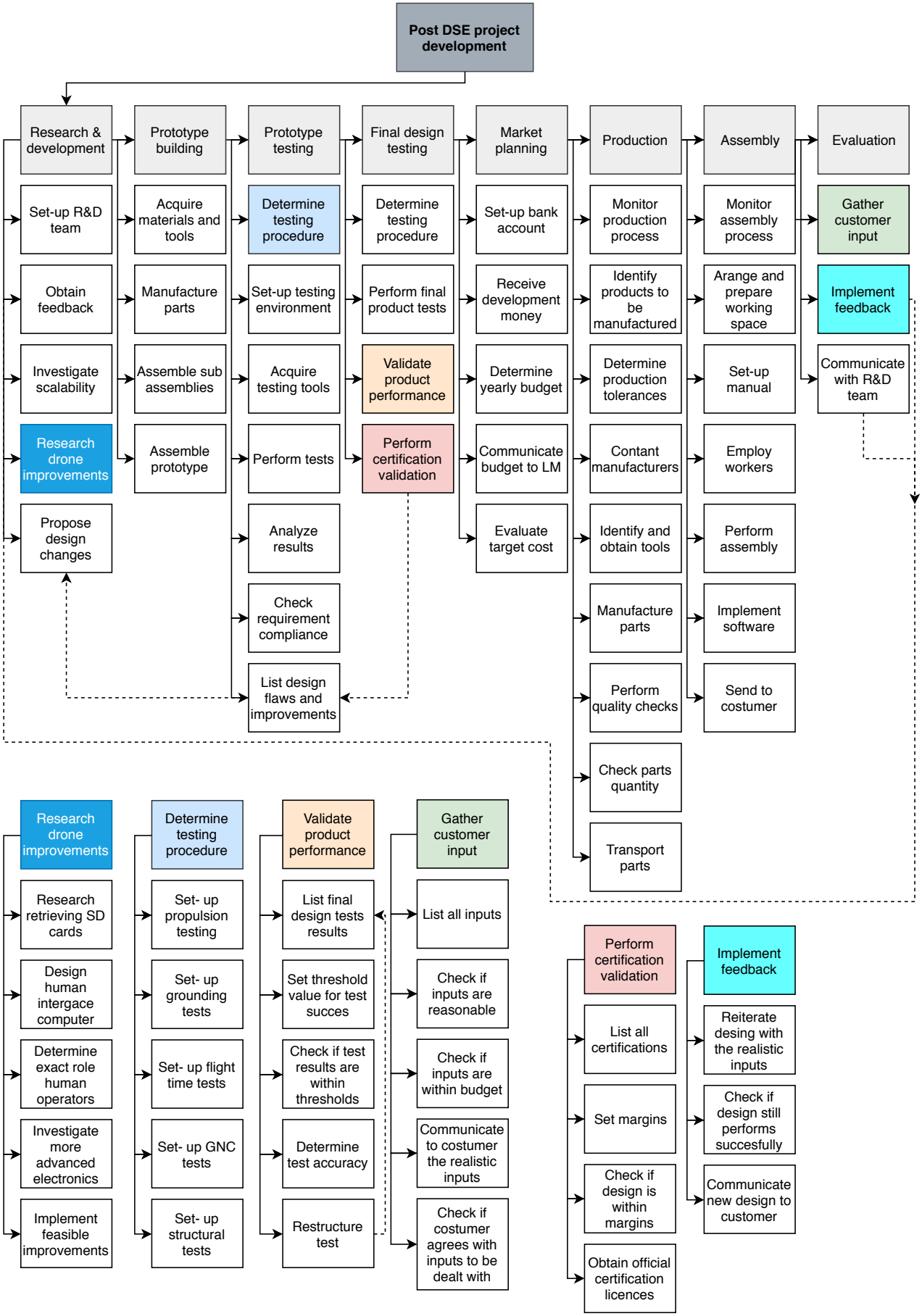
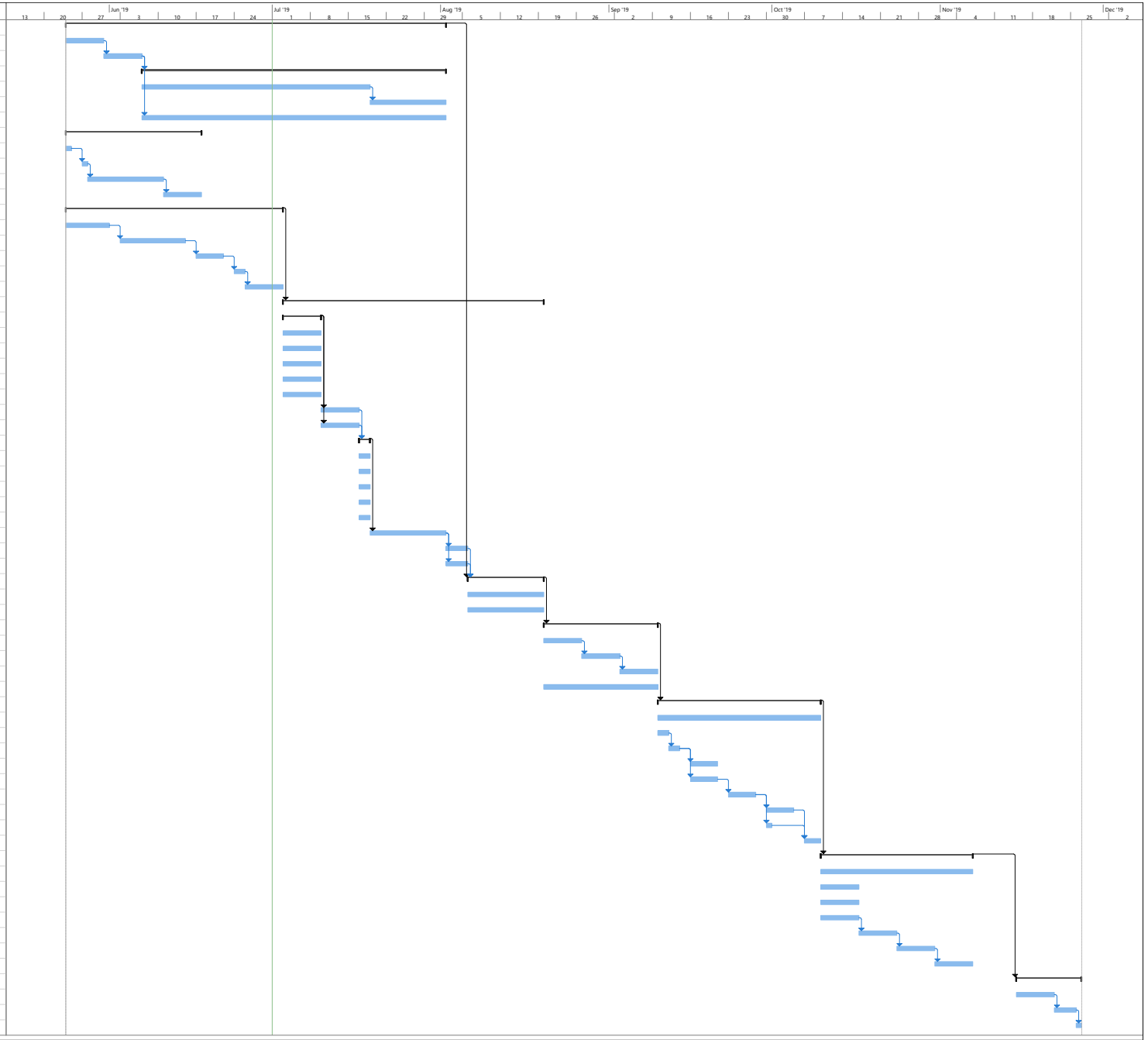


Figure 18.1: Project development logic block diagram

ID	Task Name	Duration	Start	Finish	Predecessors
1	**1 Research and Development**	**50 days**	**Fri 24-5-19**	**Thu 1-8-19**	
2	1.1 Set up R&D team	5 days	Fri 24-5-19	Thu 30-5-19	
3	1.2 Gather customer feedback	5 days	Fri 31-5-19	Thu 6-6-19	2
4	**1.3 Research scalability**	**40 days**	**Fri 7-6-19**	**Thu 1-8-19**	**3**
5	1.3.1 Expand on market analysis in commercial sector	30 days	Fri 7-6-19	Thu 18-7-19	
6	1.3.2 Research amount of products to build in coming year	10 days	Fri 19-7-19	Thu 1-8-19	5
7	1.4 Research improvements for the drone	40 days	Fri 7-6-19	Thu 1-8-19	3
8	**2 Market Planning and execution**	**17 days**	**Fri 24-5-19**	**Mon 17-6-19**	
9	2.1 Set up bank account	1 day	Fri 24-5-19	Fri 24-5-19	
10	2.2 Receive development cost from LM	1 day	Mon 27-5-19	Mon 27-5-19	9
11	2.3 Set up yearly budget	10 days	Tue 28-5-19	Mon 10-6-19	10
12	2.4 Evaluate target cost and compare with current budget	5 days	Tue 11-6-19	Mon 17-6-19	11
13	**3 Prototype Building**	**28 days**	**Fri 24-5-19**	**Tue 2-7-19**	
14	3.1 Acquire materials, manufacturing tools and off-the-shelf parts	6 days	Fri 24-5-19	Fri 31-5-19	
15	3.2 Manufacture custom parts	10 days	Mon 3-6-19	Fri 14-6-19	14
16	3.3 Assemble parts into subassemblies	5 days	Mon 17-6-19	Fri 21-6-19	15
17	3.4 Assemble subassemblies into prototype	2 days	Mon 24-6-19	Tue 25-6-19	16
18	3.5 Implement on-board software	5 days	Wed 26-6-19	Tue 2-7-19	17
19	**4 Prototype Testing**	**34 days**	**Wed 3-7-19**	**Mon 19-8-19**	**13**
20	**4.1 Determine prototype testing procedure**	**5 days**	**Wed 3-7-19**	**Tue 9-7-19**	
21	4.1.1 Determine propulsion test procedure	5 days	Wed 3-7-19	Tue 9-7-19	
22	4.1.2 Determine static charge test procedure	5 days	Wed 3-7-19	Tue 9-7-19	
23	4.1.3 Determine endurance test procedure	5 days	Wed 3-7-19	Tue 9-7-19	
24	4.1.4 Determine GNC test procedure	5 days	Wed 3-7-19	Tue 9-7-19	
25	4.1.5 Determine structural test procedure	5 days	Wed 3-7-19	Tue 9-7-19	
26	4.2 Setup testing environment	5 days	Wed 10-7-19	Tue 16-7-19	20
27	4.3 Acquire testing equipment	5 days	Wed 10-7-19	Tue 16-7-19	20
28	**4.4 Perform prototype tests**	**2 days**	**Wed 17-7-19**	**Thu 18-7-19**	**27;26**
29	4.4.1 Perform propulsion tests	2 days	Wed 17-7-19	Thu 18-7-19	
30	4.4.2 Perform static charge tests	2 days	Wed 17-7-19	Thu 18-7-19	
31	4.4.3 Perform endurance tests	2 days	Wed 17-7-19	Thu 18-7-19	
32	4.4.4 Perform GNC tests	2 days	Wed 17-7-19	Thu 18-7-19	
33	4.4.5 Perform structural tests	2 days	Wed 17-7-19	Thu 18-7-19	
34	4.5 Analyze and report test results	10 days	Fri 19-7-19	Thu 1-8-19	28
35	4.6 Check requirement compliance with test results	2 days	Fri 2-8-19	Mon 5-8-19	34
36	4.7 List design flaws	2 days	Fri 2-8-19	Mon 5-8-19	34
37	**4.8 Update design**	**10 days**	**Tue 6-8-19**	**Mon 19-8-19**	**36;35;1**
38	4.8.1 Improve current design choices	10 days	Tue 6-8-19	Mon 19-8-19	
39	4.8.2 Implement new design options	10 days	Tue 6-8-19	Mon 19-8-19	
40	**5 Final product testing and validation**	**15 days**	**Tue 20-8-19**	**Mon 9-9-19**	**37**
41	5.1 Determine final product test procedures	5 days	Tue 20-8-19	Mon 26-8-19	
42	5.2 Perform final product tests	5 days	Tue 27-8-19	Mon 2-9-19	41
43	5.3 Analayze and report final test results	5 days	Tue 3-9-19	Mon 9-9-19	42
44	5.4 Perform regulation certification	15 days	Tue 20-8-19	Mon 9-9-19	
45	**6 Production**	**22 days**	**Tue 10-9-19**	**Wed 9-10-19**	**40**
46	6.1 Monitor the production process	22 days	Tue 10-9-19	Wed 9-10-19	
47	6.2 Identify parts to be manufactured	2 days	Tue 10-9-19	Wed 11-9-19	
48	6.3 Establish production tolerances	2 days	Thu 12-9-19	Fri 13-9-19	47
49	6.4 Establish contact with manufacturers	5 days	Mon 16-9-19	Fri 20-9-19	48
50	6.5 Identify tools needed to manufacture parts	5 days	Mon 16-9-19	Fri 20-9-19	48
51	6.6 Manufacture parts	5 days	Mon 23-9-19	Fri 27-9-19	50
52	6.7 Check quality of manufactured parts	5 days	Mon 30-9-19	Fri 4-10-19	51
53	6.8 Check quantity of parts	1 day	Mon 30-9-19	Mon 30-9-19	51
54	6.9 Transport parts to assembly workspace	3 days	Mon 7-10-19	Wed 9-10-19	52;53
55	**7 Assembly**	**20 days**	**Thu 10-10-19**	**Wed 6-11-19**	**45**
56	7.1 Monitor assembly process	20 days	Thu 10-10-19	Wed 6-11-19	
57	7.2 Arrange and prepare working space for manufacturing team	5 days	Thu 10-10-19	Wed 16-10-19	
58	7.3 Set-up manual for workers	5 days	Thu 10-10-19	Wed 16-10-19	
59	7.4 Employ manufacturing workers	5 days	Thu 10-10-19	Wed 16-10-19	
60	7.5 Perform assembly	5 days	Thu 17-10-19	Wed 23-10-19	59
61	7.6 Implement software	5 days	Thu 24-10-19	Wed 30-10-19	60
62	7.7 Send to customers	5 days	Thu 31-10-19	Wed 6-11-19	61
63	**8 Quarterly evaluation**	**8 days**	**Fri 15-11-19**	**Tue 26-11-19**	**55**
64	8.1 Gather input from customers	5 days	Fri 15-11-19	Thu 21-11-19	
65	8.2 Implement feedback; update design	2 days	Fri 22-11-19	Mon 25-11-19	64
66	8.3 Communicate with R&D team	1 day	Tue 26-11-19	Tue 26-11-19	65



Project: Post-DSE
Date: Mon 1-7-19

Task | Split | Milestone | Summary | Project Summary | Inactive Task | Inactive Milestone | Inactive Summary | Manual Task | Duration-only | Manual Summary Rollup | Manual Summary | Start-only | Finish-only | External Tasks | External Milestone | Deadline | Progress | Manual Progress

BIBLIOGRAPHY

[1] T. Jardin, S. Prothin, and C. G. Magaña, *Aerodynamic performance of a hovering microrotor in confined environment,* Journal of the American Helicopter Society **62**, 1 (2017).

[2] M.Kok, J. D. Hol, and T. B. Schon, *Foundations and trends in signal processing,* AGARD , 1,153 (2017).

[3] DronesStaff, *The physics of multirotor drone flight,* (2019), accessed: 03-06-2019.

[4] Mathworks, *Drone simulation and control,* (2019), accessed: 06-2019.

[5] S. White, *Graphics pipeline,* (2018), accessed: 24-06-2019.

[6] Group 05, *Lockheed Martin Aircraft Maintenance Drone. AE3200 Design Synthesis Exercise: Midterm Report,* Tech. Rep. (TU Delft, Delft, 2019).

[7] H. Sarhan, *A software based scheduling of pid controller,* International Journal of Instrumentation and Control Systems (2014).

[8] Battery Space, *Product specification model 553562-10c,* https://www.batteryspace.com/prod-specs/4542.pdf (2019), accessed: 07-06-2019.

[9] Rijksdienst voor Ondernemend Nederland, *Dutch robotics for inspection and maintenance,* https://www.rvo.nl/sites/default/files/2017/05/roboasia-dutch-robotics-for-inspection-and-maintenance.pdf (Accessed: 24-05-2019).

[10] Bureau of Transportation Statistics, U.S. Department of Transportation, *Rail profile,* https://www.bts.gov/content/rail-profile (2016).

[11] D. Stewart, *Mro europe 2018. forecast & key trends,* https://www.oliverwyman.com/content/dam/oliver-wyman/v2/publications/2018/october/Europe-MRO-2018-Trends-&-Forecast-preso.pdf (Accessed: 24-05-2019).

[12] Lockheed Martin Corporation, *C-130j super hercules,* https://www.lockheedmartin.com/content/dam/lockheed-martin/aero/documents/C-130J/MG180389_C-130Brochure_NewPurchase_Final_Web.pdf (Accessed: 24-05-2019).

[13] Bureau of Labor Statistics, *Repair and maintenance: Naics 811,* https://www.bls.gov/iag/tgs/iag811.htm (2019), accessed: 03-06-2019.

[14] C. Colaw, Personal communication (2019).

[15] A. Shannon, *Evaluation & insights of commercial aircraft maintenance programs,* http://aircraftmonitor.com/uploads/1/5/9/9/15993320/basics_of_aircraft_maintenance_programs_for_financiers___v1.pdf (2010).

[16] Flight Ascend Consultancy, *Global aviation sector overview,* http://www.aviationnews-online.com/wp-content/video/Korea-2017/slides/d2-02.pdf (Accessed: 24-05-2019).

[17] Group 05, *Lockheed Martin Aircraft Maintenance Drone. AE3200 Design Synthesis Exercise: Baseline Report,* Tech. Rep. (TU Delft, Delft, 2019).

[18] E. Gill, S. Mestry, F. Oliviero, and W. Verhagen, *Systems engineering & aerospace design, lecture notes,* TU Delft (2019).

[19] J. B. Brandt, R. W. Deters, G. K. Ananda, and M. S. Selig, *Uiuc propeller database,* https://m-selig.ae.illinois.edu/props/propDB.html (Accessed: 09-05-2019).

[20] *Airfoil tools,* http://airfoiltools.com/ (2019), accessed: 06-06-2019.

[21] GetFPV, *Rotorx rx1404 3600kv v2 motor,* (2019), accessed: 18-06-2019.

[22] Cambridge university engineering department, *Materials data book,* (2013).

[23] Good Fellow, *Polyamide carbon fibre reinforced material information,* http://www.goodfellow.com/E/Polyamide-Nylon-6-6-30-Carbon-Fibre-Reinforced-Polymer.html (2019).

[24] Boedeker Plastics, Inc., *Static control plastics,* https://www.boedeker.com/Industries/Industry-Landing/Static-Control-Materials (2019).

[25] PlastiComp, *Long fiber reinforced thermoplastic composites,* https://www.plasticomp.com/complet-lcf50-pa66/ (2018).

[26] T. Ghomian and S. Mehraeen, *Survey of energy scavenging for wearable and implantable devices,* ScienceDirect **178**, 33 (2019).

[27] NA, *Direct methods in visual odometry,* http://wavelab.uwaterloo.ca/slam/2017-SLAM/Lecture14-Direct_visual_inertial_odometry_and_SLAM/slides.pdf (Accessed: 06-06-2019).

[28] Yousif, K., Bab-Hadiashar, A., Hoseinnezhad, R., *An Overview to Visual Odometry and Visual SLAM: Applications to Mobile Robotics,* Tech. Rep. (School of Aerospace, Mechanical and Manufacturing Engineering, RMIT University, Melbourne, 2015).

[29] V. Usenko, L. Stumberg, A. Pangercic, and D. Cremers, *Real-time trajectory replanning for mavs using uniform b-splines and a 3d circular buffer,* https://arxiv.org/pdf/1703.01416.pdf (Accessed: 11-06-2019).

[30] P. Vance Vandoren, *To pid or not to pid,* (2017), accessed: 03-06-2019.

[31] C. Cömert and C. Kasnakoğlu, *Comparing and developing pid and sliding mode controllers for quadrotor,* International Journal of Mechanical Engineering and Robotics Research **6** (2017).

[32] A. S. McCormack and K. R. Godfrey, *Rule-based autotuning based on frequency domain identification,* IEEE Transactions on Control Systems Technology (1998).

[33] OpenCV, *Optical flow,* https://docs.opencv.org/3.4/d4/dee/tutorial_optical_flow.html (2019).

[34] D. Harris, D. Sakiestewa, R. Robledo, and M. Witten, *Short-term exposure to jp-8 jet fuel results in long-term immunotoxicity.* Toxicol Ind Health. **13**, 559 (1997).

[35] H. Raymond, S. Clifford, G. Gary, and H. Terence, *Human exposure to the jet fuel, jp-8,* Aviation, Space, and Environmental Medicine **75**, 49 (2004).

[36] E. Chao, L. Gibson, and A. Lenna, *Dermal Exposure to Jet Fuel (JP-8) in US Air Force Personnel,* Annals of Work Exposures and Health **49**, 639 (2005), http://oup.prod.sis.lan/annweh/article-pdf/49/7/639/346477/mei021.pdf .

[37] G. Struwe, B. Knave, and P. Mindus, *Neuropsychiatric symptoms in workers occupationally exposed to jet fuel — a combined epidemiological and casuistic study,* Acta Psychiatrica Scandinavica **67** (1983).

[38] G. Struwe, B. Knave, and P. Mindus, *Neurasthenic symptoms in workers occupationally exposed to jet fuel,* Acta Psychiatrica Scandinavica **67** (1979).

[39] H. Brezet and C. van Hemel, *Ecodesign: A Promising Approach to Sustainable Production and Consumption*, United Nations publication (United Nations Environment Programme, Industry and Environment, Cleaner Production, 1997).

[40] S. Stramigioli, *Inspection robotics and leo,* http://www.leorobotics.nl/sites/leorobotics.nl/files/files/LEO%20Robotics%20-%20Inspection%20robotics.pdf (Accessed: 24-05-2019).

[41] L. Martin, *Innovation and disruption for q&ms (future), folder approved for public use,* Available upon request (2018).

[42] GlobeNewswire, *Rising demand from naval sector set to support growth of global remotely operated vehicle (rov) industry,* https://www.globenewswire.com/news-release/2019/04/09/1799804/0/en/Remotely-Operated-Vehicle-ROV-Market-Revenue-USD-3-12-Bn-by-2023-Rising-Demand-from-Naval-Sector-Set-to-Support-Growth-of-Global-Remotely-Operated-Vehicle-ROV-Industry.html (Accessed: 11-06-2019).

[43] T. Cooper, I. Reagan, C. Porter, and C. Precourt, *Global fleet & mro market forecast commentary. 2019 - 2029,* (Accessed: 24-05-2019).

[44] K. Michael, *Mro industry outlook,* https://www.aeromontreal.ca/download/fca8adddfff353/01-MRO+Industry+OutlookKevin+Michaels.pdf (Accessed: 25-04-2019).

[45] T. Cooper, I. Reagan, C. Porter, and C. Precourt, *Global fleet & mro market forecast commentary. 2018 - 2028,* https://www.oliverwyman.com/content/dam/oliver-wyman/v2/publications/2018/January/2018-2028_Global_Fleet_MRO_Market_Forecast_Commentary_Public_Final_web.pdf (Accessed: 24-05-2019).

[46] H. Canaday, *Mro regulatory challenges anticipated in 2017,* https://www.mro-network.com/manuals-repair/mro-regulatory-challenges-anticipated-2017 (Accessed: 24-05-2019).

[47] A. Little, *Aviation mros,* https://www.adlittle.com/sites/default/files/viewpoints/ADL_The_future_of_the_MRO_industry.pdf (Accessed: 24-05-2019).

[48] Lockheed Martin Corporation, *Lockheed martin reports first quarter 2019 results,* https://news.lockheedmartin.com/2019-04-23-Lockheed-Martin-Reports-First-Quarter-2019-Results (Accessed: 24-05-2019).

[49] S. Prince, *Computer vision: models, learning and inference* (Cambridge University Press, 2012).

[50] W. Johnson, *Helicopter Theory*, Dover Books on Aeronautical Engineering Series (Dover Publications, 1994).

[51] R. W. Deters and M. S. Selig, *Static testing of micro propellers,* Researchgate (2008), 10.2514/6.2008-6246.

[52] R. Deters, G. Ananda, and M. Selig, *Reynolds number effects on the performance of small-scale propellers,* (2014).

[53] Engineering ToolBox, *International standard atmosphere,* https://www.engineeringtoolbox.com/international-standard-atmosphere-d_985.html (Accessed: 06-06-2019).

[54] M. Drela, *Xfoil,* https://web.mit.edu/drela/Public/web/xfoil/ (2013), accessed: 07-06-2019.

[55] M. Joao, *Jblade: a propeller design and analysis software,* https://sites.google.com/site/joaomorgado23/Home (2014), accessed: 10-05-2019.

[56] L. Pereira and Jason, *Hover and wind-tunnel testing of shrouded rotors for improved micro air vehicle design,* Researchgate (2008).

[57] D. Wilcox, *Turbulence Modelling for CFD* (DCW Industries, Inc., 2007).

[58] J. H. Ferizger and M. Perić, *Computational Methods for Fluid Dynamics* (Springer, Berlin, Heidelberg, 2002).

[59] M. Mueller, *ecalc motor database,* https://www.ecalc.ch/xcoptercalc.php (2019).

[60] The Engineering ToolBox, *Typical properties of engineering materials like steel, plastics, ceramics and composites,* https://www.engineeringtoolbox.com/engineering-materials-properties-d_1225.html (2019).

[61] A. Kassimali, *Matrix analysis of structures,* Cengage Learning (2010).

[62] FriendlyElec, *13.2mp mipi camera module,* (2019), accessed: 18-06-2019.

[63] KIWI, *Raspberry pi camera board v2 - 8mp,* (2019), accessed: 18-06-2019.

[64] FriendlyElec, *5mp camera module for nanopi duo2,* (2019), accessed: 18-06-2019.

[65] Pimoroni, *Standard – camera module for raspberry pi zero,* (2019), accessed: 18-06-2019.

[66] GreenIce, *High voltage led,* (2019), accessed: 14-06-2019.

[67] C. de Wagter, Personal communication (2019).

[68] Xsens, *Mti 1-series,* (2018), accessed: 05-06-2019.

[69] Movidius, an Intel company, *Enhanced visual intelligence at the network edge,* (2017), accessed: 11-06-2019.

[70] Amazon, *Microsd slot,* (2019), accessed: 14-06-2019.

[71] Texas Insturments, *Cc2500 2.4-ghz rf transceiver,* (2019), accessed: 17-06-2019.

[72] Distrelec Group AG, *PCB Terminal Blocks*, Tech. Rep. (RND Connect, Grabenstrasse 6, 8606 Nänikon, Switzerland, 2018).

[73] Molex, *FFC / FPC Board Connector*, Tech. Rep. (Farnell, 2014).

[74] Hobby Electronica, *Pcb cost,* (2019), accessed: 14-06-2019.

[75] Aliexpress, *Matrix-cam1320,* (2019), accessed: 14-06-2019.

[76] Pololu, *Time of flight vl53l1x,* (2019), accessed: 14-06-2019.

[77] NXP, *Cortex m4 nxp,* (2019), accessed: 14-06-2019.

[78] E. Brown, *Intel myriad x vpu,* (2018), accessed: 14-06-2019.

[79] XSens, *Xsense mti-1 imu,* (2019), accessed: 14-06-2019.

[80] DataIO, *64gb microsd sandisk,* (2019), accessed: 14-06-2019.

[81] M. A. P. Mahmud, N. Huda, S. H. Farjana, and C. LangOrcID, *Comparative life cycle environmental impact analysis of lithium-ion and nickel-metal hydride batteries,* MDPI (2019).

[82] Battery Space, *Battery space 1s battery protection circuits,* https://www.batteryspace.com/pcbfor37vpack.aspx (2019), accessed: 03-06-2019.

[83] The FVT team, *Fvt littlebee,* http://fvt-littlebee.com/ (Accessed: 05-06-2019).

[84] Amazon, *Matek pdb,* https://www.amazon.co.uk/Distirbution-Support-Current-Quadcopter-LITEBEE/dp/B071477C2B?ref_=fsclp_pl_dp_5 (2019), accessed: 04-06-2019.

[85] PowerStream Technology, *Wire gauge and current limits including skin depth and strength,* (2019), accessed: 06-06-2019.

[86] KabelTronik, *Li5y ptfe stranded wire 600v,* (2019), accessed: 06-06-2019.

[87] Molex, *Premo-Flex Cable Jumpers*, Tech. Rep. (Mouser Electronics, 2018).

[88] CIRRIS Systems, *Resistance and resistivity,* (2016), accessed: 11-06-2019.

[89] CIRRIS Systems, *Temperature coefficient of copper,* (2019), accessed: 11-06-2019.

[90] AA Portable Power Corp, *Protection circuit module (pcb) for 14.8v li-ion battery pack (4 cells with 7a limit) with fuel gauge socket,* (2019), accessed: 14-06-2019.

[91] Speed Drones, *Pdb cost,* (2019), accessed: 14-06-2019.

[92] Mouser, *Cabling cost,* (2019), accessed: 14-06-2019.

[93] K. L. Kaiser, *Electrostatic Discharge* (CRC Press, 2006).

[94] J. Kuchta and J. Murphy, *Fire hazard evaluation of thickened fuel.* AGARD (1971).

[95] I. Limited, *Product data sheet stadis 450,* https://www.fmv.se/FTP/Drivmedel%202018/SV/datablad/M0729-465015_Stadis_450PDS.pdf (2011).

[96] Innospec Limited, *Safety data sheet stadis 450,* http://www.tanker-enemy.com/PDF/Stadis_450_sds_eng.pdf (2006).

[97] A. McCartnet, *Static electricity & relative humidity,* https://pdfs.semanticscholar.org/45d9/2525e308c595ff012183c6601ac11d0a899f.pdf (2012).

[98] J. M. Choi and T. W. Kim, *Humidity sensor using an air capacitor,* Transactions on Electrical and Electronic Materials **14**, 182 (2013).

[99] J. Schmitigal and J. Tebbe, *JP-8 and other military fuels*, Tech. Rep. (RDECOM, 2011).

[100] W. S. Chow and W. L. Tham, *Effects of antistatic agent on the mechanical, morphological and antistatic properties of polypropylene/organo-montmorillonite nanocomposites,* eXPRESS Polymer Letters **3** (2009).

[101] M. Meadows, *Air ionization,* https://blog.gotopac.com/2013/04/17/air-ionization-how-it-works/ (2013), accessed: 11-06-2019.

[102] Simco Ion, *Frequency asked question,* https://technology-ionization.simco-ion.com/resources/faqs (2019), accessed: 11-06-2019.

[103] C. Styles, *Costs of nitrogen gas,* https://puritygas.ca/nitrogen-gas-costs/ (2017), accessed: 11-06-2019.

[104] J. Pike, *C-130 hercules,* https://fas.org/man/dod-101/sys/ac/c-130.htm (2000), accessed: 11-06-2019.

[105] NXP Semiconductors N.V. 2, *Lpc54018jxm/lpc54s018jxm,* (2019), accessed: 05-06-2019.

[106] R. Hamann and M. van Tooren, *Systems engineering & technical management techniques,* p. 109/214: https://www.investopedia.com/terms/r/returnoninvestment.asp (2006).

[107] E. Jakob, K. Vladen, and C. Daneil, *Direct Sparse Odometry*, Tech. Rep. (Technical University Munich & Intel Labs, 2016).

[108] P. Sang, L. Kyoung, L. Sang, *A Line Feature Matching Technique Based on an Eigenvector Approach*, Tech. Rep. (School of Electrical Engineering, Seoul National University, Seoul, Korea, 1999).

[109] M. Yanxia, C. Qixing, *A Method of Line Matching Based on Feature Points*, Tech. Rep. (Chongqing Normal University, China, 2012).

[110] K. Li, J. Yao, X. Lu, L. Li, and Z. Zhang, *Hierarchical line matching based on line–junction–line structure descriptor and local homography estimation,* Neurocomputing **184**, 207 (2016).

[111] C. Beall, *Stereo visual odometry,* http://www.cs.cmu.edu/~kaess/vslam_cvpr14/media/VSLAM-Tutorial-CVPR14-A12-StereoVO.pdf (2014), accessed: 11-06-2019.

[112] P. Bronislav, Z. Pavel, and C. Martin, *Absolute Pose Estimation from Line Correspondences using Direct Linear Transformation*, Tech. Rep. (Brno University of Technology, Brno, Czech Republic, 2017).

[113] OpenCV, *Camera calibration and 3d reconstruction,* (2019), accessed: 12-06-2019.

[114] X. Chi, Z. Lilian, C. Li, and K. Reinhard, *Pose estimation from line correspondences: A complete analysis and a series of solutions,* Transactions on Pattern Analysis and Machine Intelligence **39**, 1209 (2016).

[115] T. Lacey., *Tutorial: The Kalman Filter* (MIT, 2018).

[116] J. Chouinard, *Imaging basics – calculating exposure time for machine vision cameras,* https://www.1stvision.com/machine-vision-solutions/2018/06/industrial-camera-exposure-time.html (2018).

[117] T. Sieberth, R. Wackrow, and J. H. Chandler, *Influence of blur on feature matching and a geometric approach for photogrammetric deblurring,* The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences **XL-3** (2014).

[118] C. Goerzen, Z. Kong, and B. Mettler, *A survey of motion planning algorithms from the perspective of autonomous uav guidance,* Journal of Intelligent and Robotic Systems , 65 (2009).

[119] V. Kunchev, L. Jain, V. Ivancevic, and A. Finn, *Path Planning and Obstacle Avoidance for Autonomous Mobile Robots: A Review*, Tech. Rep. (School of Electrical and Information Engineering, Univeristy of South Australia, Mawson Lakes SA 5095, Australia, 2008).

[120] E. Vitolo, C. Mahulea, and M. Kloetzer, *Path-planning in Discretized Environments with Optimized Waypoints Computation*, Tech. Rep. (International Conference on Emerging Technologies and Factory Automation, 2018).

[121] P. Broz, *Path planning in combined 3D grid and graph environment*, Tech. Rep. (Department of Computer Science and Engineering University of West Bohemia, Pilsen, Czech Republic, 2006).

[122] M. Radmanesh, M. Kumar, and M. Sarim, *Overview of path planning and obstacle avoidance algorithms for UAVs: a comparative study*, Tech. Rep. 2 (Department of Mechanical and Materials Engineering, University of Cincinnati, Cincinnati, OH 45221, USA, 2018).

[123] M. Pivtoraiko, A. Knepper, and A. Kelly, *Differentially Constrained Mobile Robot Motion Planning in State Lattices*, Tech. Rep. (Robotics Institute Carnegie Mellon University, Pittsburgh, Pennsylvania 15213, 2009).

[124] H. Gould, J. Tobochnik, and W. Christian, *An Introduction to Computer Simulation Methods Applications to Physical System*, Tech. Rep. (CreateSpace Independent Publishing Platform, 2019).

[125] D. Wagner, D. Schultes, P. Sanders, and D. Delling, *Engineering Route Planning Algorithms*, Tech. Rep. (Algorithmics of Large and Complex Networks: Design, Analysis, and Simulation, 2012).

[126] S. Al-Arif, A. Ferdous, and S. Nijami, *Comparative Study of Different Path Planning Algorithms: A Water based Rescue System*, Tech. Rep. (Department of Electrical and Electronic Engineering, Islamic University of Technology, Boardbazar, Gazipur – 1704, Bangladesh., 2012).

[127] Donald Bren School of Information and Computer Sciences, *Informed search algorithms,* https://www.ics.uci.edu/~welling/teaching/271fall09/InfSearch271f09.pdf (Accessed: 12-06-2019).

[128] W. Zeng and R. Church, *Finding shortest paths on real road networks: the case for a^*,* (2009), accessed: 12-06-2019.

[129] A. Guruji, H. Agarwal, and D. Parsediya, *Time-efficient a^* algorithm for robot path planning,* https://core.ac.uk/download/pdf/82517222.pdf (Accessed: 12-06-2019).

[130] S. Mussmann and A. See, *Graph search algorithms,* (2019), accessed: 12-06-2019.

[131] A. Felner, *Position paper: Dijkstra's algorithm versus uniform cost search or a case against dijkstra's algorithm,* https://www.aaai.org/ocs/index.php/SOCS/SOCS11/paper/viewFile/4017/4357 (Accessed: 12-06-2019).

[132] D. Ferguson, M. Likhachev, and A. Stentz, *A guide to heuristic-based path planning,* https://www.cs.cmu.edu/~maxim/files/hsplanguide_icaps05ws.pdf (Accessed: 12-06-2019).

[133] A. Patel, *Heuristics from amit's thoughts on pathfinding,* (2019), accessed: 13-06-2019.

[134] T. Goto, T. Kosaka, and H. Noborio, *On the heuristics of a^* or a algorithm in its and robot path-planning,* https://pdfs.semanticscholar.org/ffd2/1dabf306f1698faf39a9c688eeb1e0887708.pdf (2011), accessed: 12-06-2019.

[135] T. Abiy, H. Pang, and B. Tiliksew, *a^* search,* https://brilliant.org/wiki/a-star-search/ (Accessed: 12-06-2019).

[136] F. Duchon, A. Babinec, M. Kajan, P. Beno, M. Florek, T. Fico, and L. Jurisica, *Path planning with modified a star algorithm for a mobile robot,* (2014), accessed: 01-06-2019.

[137] K. Khantanapoka and K. Chinnasarn, *Pathfinding of 2d & 3d game real-time strategy with depth direction a^* algorithm for multi-layer,* (2009), accessed: 02-06-2019.

[138] D. Griffiths, *An Introduction to Matlab* (Department of Mathematics, The University of Dundee, 2012).

[139] A. computer science portal for geeks, *Analysis of algorithms, big-o analysis,* https://www.geeksforgeeks.org/analysis-algorithms-big-o-analysis/ (2019).

[140] S. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach* (Prentice Hall, 2003).

[141] H. Ozark, *Branching factor,* http://ozark.hendrix.edu/~ferrer/courses/335/f11/lectures/effective-branching.html (2019).

[142] Y. Chao and W. Hongxia, *Developed dijkstra shortest path search algorithm and simulation,* https://ieeexplore.ieee.

org/stamp/stamp.jsp?tp=&arnumber=5541129 (Accessed: 11-06-2019).
[143] Stackoverflow., *Difference and advantages between dijkstra & a star,* https://stackoverflow.com/questions/13031462/difference-and-advantages-between-dijkstra-a-star (Accessed: 11-06-2019).
[144] D. Nau, *Informed search algorithms,* (Accessed: 11-06-2019).
[145] M. Yamamoto, M. Iwamura, and A. Mohri, *Quasi-time-optimal motion planning of mobile platforms in the presence of obstacles,* https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=774046 (1999).
[146] J. Kim, *Discrete approximations to continuous shortest-path: Application to minimum-risk path planning for groups of uavs,* https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=1272863 (Accessed: 24-05-2019).
[147] K. Judd and T. McLain, *Spline based path planningfor unmanned air vehicles,* https://arc.aiaa.org/doi/pdf/10.2514/6.2001-4238 (Accessed: 28-06-2019).
[148] Mathworks, *Quadcopter project package,* (2019), accessed: 03-06-2019.
[149] P. Pounds, *Design, Construction and Control of a Large Quadrotor Micro Air Vehicle*, Ph.D. thesis, Australian National University (2007).
[150] D. Q. Chu, *Incremental non-linear dynamic inversion,* (Accessed: 06-06-2019).
[151] M. Drela, *Performance of propellers,* Massachusetts Institute of Technology (2018).
[152] Wikipedia, *Runge–kutta methods — Wikipedia, the free encyclopedia,* (2019), [Online; accessed 24-June-2019].
[153] Wikipedia, *Phong reflection model — Wikipedia, the free encyclopedia,* (2019), [Online; accessed 24-June-2019].
[154] K. Sun, *Msckf vio,* (2019), accessed on 24-06-2019.
[155] MathWorks, *Debug a matlab program,* https://nl.mathworks.com/help/matlab/matlab_prog/debugging-process-and-features.html (Accessed: 13-06-2019).
[156] MathWorks, *Write script-based unit tests,* https://nl.mathworks.com/help/matlab/matlab_prog/write-script-based-unit-tests.html (Accessed: 13-06-2019).
[157] E. Mooij, Z. Papp, and W. van der Wal, *Ae3211-ii simulation, verification and validation lecture notes,* TU Delft, the Netherlands (February 2019).
[158] P. Enrico, L. Fabio, and C. Lorenzo, *Reliability and maintenance analysis of unmanned aerial vehicles,* Reliability Degradation, Preventive and Corrective Maintenance of UAV Systems. (2018).
[159] S. Fred, *The case of drones,* (2019), accessed: 17-06-2019.
[160] C. ran Tang, *Reliability of an electric motor system*, Ph.D. thesis, University of Johannesburg (2005).
[161] F. Josh and S. Austin, *Digital Camera Failure Rates: Panasonic most reliable manufacturer of Point-and-shoot Cameras*, Tech. Rep. (SquareTrade, US, 2010).
[162] XPpower, *Reliability in electronics,* https://www.xppower.com/Portals/0/pdfs/Reliability.pdf (2008).
[163] D. Hayyan, L. C. S. Aznul, and, and M. A. Mohammed, *Slamm: Visual monocular slam with continuous mapping using multiple maps,* Plos one (2018), https://doi.org/10.1371/journal.pone.0195878.t003.
[164] Air Force Institute for Environment, Safety and Occupational Health Risk Analysis., *C-130 Confined Space Technical Guidance Document.*, Tech. Rep. (Health and Safety Devision, 2513 Kennedy Circle, 2002).
[165] Air Mobility Command, *C-5 Handbook.*, Tech. Rep. (-, 2012).
[166] S. Abdullah, *Successful saas companies spend 23% of revenue on r&d,* https://medium.com/@sammyabdullah/successful-saas-companies-spend-23-of-revenue-on-r-d-3602e9dc2de (2018).
[167] G. Boykin, *What percentage of gross revenue should be used for marketing & advertising?* https://smallbusiness.chron.com/percentage-gross-revenue-should-used-marketing-advertising-55928.html (2019).
[168] G. Ferguson, *What percentage of the budget should be spent on payroll?* https://smallbusiness.chron.com/percentage-budget-should-spent-payroll-64862.html (2012).
[169] D. Natives, *The different sls 3d printers on the market,* https://www.3dnatives.com/en/different-sls-3d-printers-220320184/ (2018).
[170] PayScale, *Average mechanical lab technician salary,* (2019), accessed: 18-06-2019.
[171] YES!Delft, *Rental rates office space,* (20184), accessed: 18-06-2019.
[172] Guandong Silver Age Sci.& Tech., *Good rigidity, low density laser sintering technology nylon pa12 powder carbon fiber/pa composite powder,* https://www.alibaba.com/product-detail/Good-rigidity-low-density-Laser-Sintering_60514337154.html (2019).
[173] MatterHackers, *High impact polystyrene (hips) dissolvable filament,* https://www.matterhackers.com/store/3d-printer-filament/hips-175mm-1kg (2019).
[174] Slant 3D Blog, *The price of 3d printing vs injection molding,* http://www.slant3d.com/slant3d-blog/the-price-of-3d-printing-vs-injection-molding (Accessed: 17-06-2019).
[175] J.Sinke, *New developments - additive manufacturing. ae3211-ii lecture notes,* TU Delft (2019).
[176] Interco, *Recycling printed circuit boards pcb,* (2019), accessed: 13-06-2019.
[177] F. Knappich, M. Klotz, M. Schlummer, J. Wölling, and A. Mäurer, *Recycling process for carbon fiber reinforced plastics with polyamide 6, polyurethane and epoxy matrix by gentle solvent treatment,* Waste Management **85**, 73 (2019).
[178] Mathworks, *Control systems in practice,* (2019), accessed: 06-2019.
[179] S. Prothin, C. F. Escudero, T. Jardin, and N. Doué, *Archean: Aerodynamics of rotors in confined environments,* International Micro-Air Vehicles Conference **10** (2018).