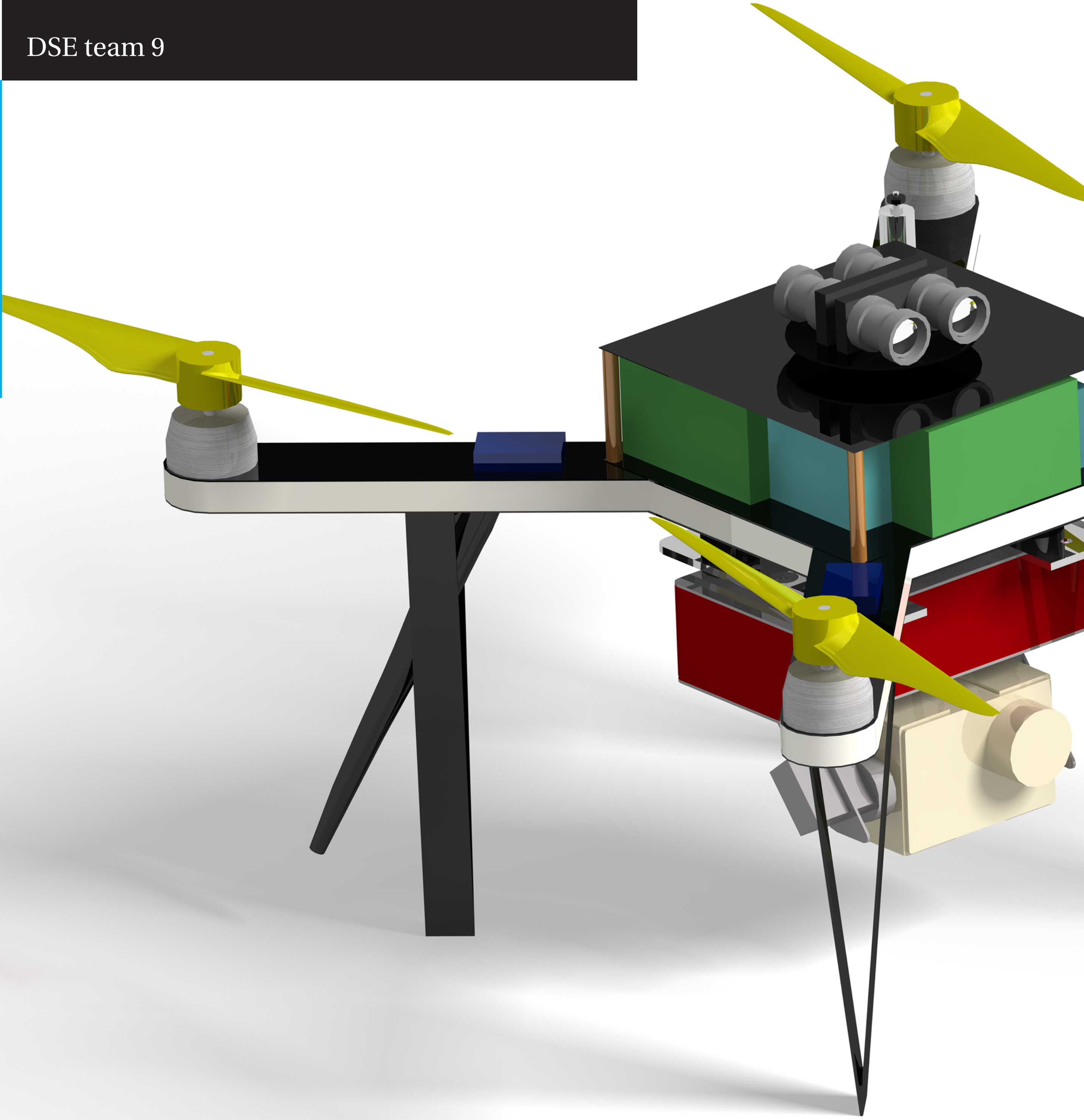
DSE Spring 2014

IMAV 2014 Single-UAV operations

Final report

DSE team 9

Delft University of Technology



Delft University of Technology

Challenge the future

IMAV 2014
Single-UAV operations

Final report

by

4150740	I. K. Ashraf
4090675	M. C. Butijn
4146557	Z. Liu
4139461	M. P. R. van Moorselaar
4168070	M. Pfahler
4152859	M. Scherff
4185692	M. Siddiquee
4147111	S. Unni
4142179	J. van Wensveen
4169883	P. L. Wyzen

Project duration:	April 22, 2014 – July 3, 2014	
Tutor:	Dr. Ir. E. van Kampen	Delft University of Technology
Coaches:	Q. Guan	Delft University of Technology
	D. Mehta	Delft University of Technology

TU Delft
Delft University of Technology

Acknowledgements

Group 9 of the 2014 spring DSE would like to thank their tutor, supervisor and coach Dr. Ir. Erik-Jan van Kampen, Qingbao Guan and Dhruv Mehta for their support and feedback on the obtained results. Their help has been very valuable for the group to obtain the final design. We would also like to thank the faculty of Aerospace Engineering of the Delft University of Technology and the OSSAs for the resources and facilities provided to perform this DSE assignment. All these people and institutes have contributed towards a successful end of the DSE and the corresponding final report.

Preface

This report is the final report written by Team 9 in the Spring Design Synthesis Exercise of the Faculty of Aerospace Engineering at the Delft University of Technology.

All work conducted in this report is based upon the preceding project plan, baseline report and mid term report. The first one presented the planning and general organisation of the project, while the baseline report presented the requirements and design options for the overall mission objective of "impressing the jury at the IMAV competition 2014". In the mid term report different concepts were developed, from which eventually one design was chosen. In this report that design choice is finalised. It can however be read on its own, as it presents a fully worked out design.

All the different aspects and subsystems of the final design are looked into, divided over the different chapters. The reader interested in only one part of the design is therefor directed to the chapter in which this aspect is elaborated upon.

Contents

LIST OF FIGURES

LIST OF TABLES

List of abbreviations

Abbreviation	**Meaning**
AC	Alternating Current
AoV	Angle of View
c.g.	centre of gravity
COTS	Commercially available Off The Shelf
CSI	Camera Serial Input
DC	Direct Current
DSE	Design Synthesis Exercise
DSLR	Digital Single Lens Reflex
ESC	Electronic Speed Controller
EIRP	Equivalent Isotropic Radiated Power
FCS	Flight Control System
FFBD	Functional flow block diagram
FBD	Functional breakdown structure
FPV	First Person View
FSO	Free Space Optical communication
GCS	Ground Control System
GNC	Guidance Navigation Control
GPS	Global Positioning System
GPU	Graphics Processing Unit
GS	Ground Station
GUI	Graphical User Interface
IMAV	International Micro Air Vehicle Conference and Competition
IMU	Inertial Measurement Unit
IP	Internet Protocol
LIDAR	LIght Detection And Ranging
MAV	Micro Aerial Vehicle
MEMS	Micro-Electro-Mechanical Systems
MMOI	Mass Moment Of Inertia
QAM	Quadrature Amplitude Modulation
Req	Required / Requirement
RGB	Red Green Blue
RPM	Revolution Per Minute
RPS	Revolution Per Second
Rx	Reception
SLAM	Simultaneous Localisation And Mapping
STANAG	Standardization Agreement
TBD	To Be Determined
Tx	Transmission
UAS	Unmanned Aerial System
UAV	Unmanned Aerial Vehicle
UHF	Ultra High Frequency
VDECS	Vibration Damped Electronics and Camera Structure
VHF	Very High Frequency
WLAN	Wireless Local Area Network
WPAN	Wireless Personal Area Network

List of symbols

Table 1: Upper-case Latin symbols

Symbol	Description	Unit
A	Area	[m^2]
B	Bandwidth	[Hz]
C	Coefficient	[-]
C	Discharge rate	[C]
DR	Data/Downlink rate	[bs]
D	Diameter	[m]
D	Drag	[N]
E	Energy	[J]
E	Young's modulus	[Pa]
F	Force/ load	[N]
G	(Budget communication) Gain	[dB]
I	Current	[A]
I	Mass moment of inertia	[kgm^2]
I	Area moment of inertia	[m^4]
J	Advance ratio	[-]
Kv	Voltage constant	[rpm/V]
L	Budget communication loss	[dB]
L	Length	[m]
L	Lift	[N]
L	Loss	[dB]
N	Noise	[dB]
P	Power	[W]
Re	Reynolds number	[-]
S	Surface area	[m^2]
T	Temperature	[K]
T	Thrust	[N]
V	Flight speed	[m/s]
V	Voltage	[V]
W	Weight	[N]

Table 2: Lower-case Latin symbols

Symbol	Description	Unit
c	Chord length	[m]
c	Speed of light	[m/s]
d	Characteristic length	[m]
d	Propagation path length (slant range)	[m]
d	Distance	[m]
f	Frequency	[Hz]
h	(Flying) Height	[m]
m	Mass	[kg]
n	Rotational speed	[rad/s]
r	Arm-length	[m]
r	Rotor radius	[m]
t	Thickness	[m]
u	Inflow/Outflow Speed	[m/s]

Table 3: Subscripts

Symbol	Description
0	Axial inflow
a	Advance
a	Propagation and polarization
b	Bit
d	Downlink
d	Drag
e	Outflow/propwash
l	Lift
l	Line
r	Radial inflow
ra	Receiver antenna
s	Space
s	System noice
ta	Transmitter antenna
u	Uplink
o	Other
p	Pointing
x	Horizontal
y	Vertical
prop	Propeller

Table 4: Greek symbols

Symbol	Description	Unit
α	Angle of attack	[°]
δ	Deflection	[m]
η	Efficiency	[-]
μ	Dynamic viscosity	[kg/m s]
Ω	Rotational velocity	[rad/s]
ρ	Air density	[kg/m^3]
θ	Tilt angle	[°]
θ	Pitch angle	[°]
ϕ	Roll angle	[°]
ψ	Yaw angle	[°]

1 INTRODUCTION

The International Micro Air Vehicle Conference and Competition (IMAV) combines a scientific conference with a technological competition involving Micro Air Vehicles (MAVs). It is organized with the intention to stimulate the focus on research that can be used for real life scenarios while allowing the various research groups from around the world to share their knowledge. The 2014 competition consists of a single mission that combines both outdoor and indoor mission elements with a focus on the following tasks: surveillance, object recognition, endurance, and multi-MAV operations. During this mission a jury will judge the design of each team based on a number of different criteria. Examples of these criteria are the level of autonomy, overall performance and the dimensions of the MAV.

The goal of this Design Synthesis Exercise (DSE) project is to design a single Unmanned Aerial Vehicle (UAV) system that will compete in the IMAV 2014 competition. This year the mission is to create a map of a small village hit by a major natural disaster, observe buildings and search for survivors inside houses. The design should be able to perform this mission, given a number of requirements and restrictions. It should also comply with the safety rules set by the organizers of the event and Dutch laws. However, the most important goal that the design has to fulfill is to impress the jury of the IMAV competition. Note that the list of requirements can be found in Appendix A. Throughout the report there will be referred to these requirements in bold.

This report marks the end of the detailed design phase and with that the end of the DSE. In this report the final design of the UAV is presented. First a market analysis is performed in Chapter 2. Next the way the UAS is operated is explained in Chapter 3.

In the chapters after that the design of the different subsystems and parts of the UAV will be shown in detail. First the propulsion and aerodynamics in Chapter 4, followed by the structural design, as well as the manufacturing process in Chapter 5. Next the sensors are described in Chapter 6 with the GNC systems explained after that in Chapter 7. The communication subsystem is shown in Chapter 8. Finally the on-board power system is shown in Chapter 9. After this the ground system design is elaborated upon in Chapter 10. Finally Chapter 11 wraps up the whole design.

After the design has been fully shown, Chapter 12 shows the procedures taken to verify and validate the programs used for designing the UAV. After this a risk assessment is shown in Chapter 13. Next the RAMS(Reliability, Availability, Maintainability and Safety) characteristics are shown in Chapter 14, followed by the project design and development logic needed for the post-DSE phases of the project, see Chapter 15. Chapter 16 is the last chapter, it discusses sustainability aspects of the UAV. Eventually everything is concluded in the conclusion, Chapter 17.

2

MARKET ANALYSIS

Fully autonomous and micro-scaled aerial vehicles are increasing in popularity. Already, more than 1 million hours are flown annually in the United States alone [10]. Currently the global market revenue is worth over 5 billion euros and is expected to grow over 6 billion euros in 2018 [11]. Though the market is rapidly developing, it faces quite some challenges. First the laws and regulations have to be updated for UAVs and MAVs. Secondly the market will have to increase its accessibility, availability, pricing and sustainability. Finally since new MAV applications are still being identified on a regular basis, a lot of technological developments can be expected as well.

The last point is not only an obstruction, but also a possibility for engineers to capture a non-existing market share. A MAV that is capable of performing the tasks as specified by the IMAV 2014 challenge will severely increase its added value. This is mainly because the use of UAVs for humanitarian, disaster response, search and rescue, and other life-saving operations have entertained little acknowledgement [10].

This chapter is meant to identify typical hardware, software and overall system costs for the discussed UAV design. This cost estimation will be established - as well as is possible now - in Section 2.2. However, before looking at specific UAV related costs, Section 2.1 classifies existing surveillance systems and briefly discusses them.

2.1. EXISTING SYSTEMS

- Human surveillance and exploration (no fixed cost, salary): not always preferred for ethical reasons, but the most available solution.

- Satellites or aircraft (up to millions of euros): only provide global top-view information and is limited for indoor surveillance, although infra-red is still an option sometimes. Though expensive, they can usually be used for a wide variety of other applications as well. Examples are the P-3B Orion aircraft or the KH-4B Corona satellite.

- Ground robotics (varying from thousands to millions of euros): can go indoors and perform simple tasks, but are relatively slow and have difficulty with most obstacles and stairs. Existing robots are for example: DRAGON RUNNER (about 125 thousand euros) [12], Foster Miller TALON (about 50 thousand euros) [13], Atlas (unknown amount of million euros).

- UAVs (varying from thousands to millions of euros): still an area in high development, especially for combined operations. Faster and more agile than ground robots, but with increased challenges in power management. Existing surveillance UAVs include the Trimble UX5 and the sensFly: eBee, both costing tens of thousands of euros and capable of mapping outdoor areas, including depth analysis [14]. Some military drones (predator, Reaper, Global hawk) cost up to hundreds of millions of euros, but have wider applications [15].

2.2. COST ESTIMATION

An integral objective of a market analysis is to establish a target cost for the product in question. This is done primarily by investigating the costs of similar products already on the market, evaluating the services they provide and assessing how the performance of the designed product compares to its competition. Due to the fact that the UAV industry is rapidly developing with innovation constantly occurring, it can be challenging to find products with the same features as the MAV being designed for the IMAV competition. Table 2.1 contains some UAVs that perform more or less similar functions to the MAV that is to be designed, which could serve as a price benchmark.

Next to simple hardware setups for a general UAV, Table 2.3 provides prices of essential hardware and software components for an autonomous MAV. With these two tables a target price can be set up. A reasonable price range for the hardware would be between 1,000 and 7,000, based on Table 2.1. A reasonable price range for essential UAV software would be in the range of 1,000 to 25,000, based on Table 2.3.

Note that previously mentioned prices are not the prices for the final design presented in this report. They were used before the design process started to make an allocation for the budget and to get a feel for the prices the team would be working with. The final costs of the UAV are presented in Section 11.5.

Table 2.1: Price and features of similar products on the market [5] [6]

Name	Skybotix CoaX Autonomous UAV Micro Helicopter Drone	Xtreme 2.0 Gen II	MicroPilot MP- Vision UAV Glider
Price [€]	3,616	2,711	6,870
Features	• COAX BASIC • Gumstix board- Overo Earth and 2 Gb micro SD card • Camera- Targus micro Webcam • Wifi module • Micro SD card reader	• Naza V2 controller • Motors and ESCs • V1-XR Booms • Gimbal and Landing Gear	• Airframe • MP2128P Autopilot • 2.4 GHz data link • Ublox GPS receiver • HORIZON ground control software • CropCam image software • Lithium polymer batteries (x4)

Table 2.2: Price and features of similar products on the market (continued)[5] [6]

Name	RTX-X1FPV Quadcopter UAV	Turbo Ace X830-D Drone RTF
Price [€]	1,810	1,000
Features	• Built in Camera Mount for FPV Camera and Go-Pro Series of Cameras • Highly modifiable, for custom additions • Composite Construction • Modular frame for fast repair • V1-XR telescopic boom design so you can run 10” to 14” Props • Designed to be flown outside in wind conditions • Plenty of room for payload • Designed to be flown FPV but designed to be easily configured for aerial photography application	• High payload capacity • Video camera and lens mounting capabilities • Crash and impact resistant • 23-30 minutes flight time • Foldable

Table 2.3: Overview of essential software components for any autonomous MAV [7]

#	Name	Advantages	Disadvantages	Cost indication [€]	Sources
Communication protocols (software)					
1	STANAG 4586	NATO standard	Limited support, expensive	3,900	[16]
2	MAVLink	Open Source, community support	Used by hobbyist	Free	[17]
IMU/Autopilot (hardware plus software)					
1	ArduPilot	Open source, community support, simulation support	Not STANAG compliant, requires a lot of coding	175	[18] [19]
2	PX4	Open source, community support	Not STANAG compliant, requires a lot of coding	200	[20] [21]
3	Papparazzi	Open source, community support	Requires extensive coding	180	[22] [23]
4	Piccolo	STANAG compliant, high level processing, simulation software, includes communication system	Expensive	4k - 6k (estimate)	[24]
5	MicroPilot	STANAG compliant, high level processing, simulation software, 24 grams, 10 x 4 cm, includes communication system	Expensive	2k - 6k	[25]
Radio transceiver (hardware)					
1	DigiXTEND	Long range, communication support	Incompatible with ArduPilot and PX4	130	[26]
2	3DR	Small and lightweight, open source, frequency hopping, community support	Limited range, low power	75	[27]
3	Digi XBEE pro	Small, lightweight, affordable, community support	Limited range, depreciated by model	95	[28]

3
UAS OPERATIONS

This chapter highlights the operations of the UAS as well as the logistics to support it. The mission plan follows these sections and the function analysis ends this chapter.

3.1. Logistics

The UAS will be carried in two backpacks. One person is needed to carry the system. Being independent from a motorised mean of transport improves the deployment of the UAS in remote areas.

The operator should make sure that from the ground station there is a clear line of sight to the UAV. Otherwise the operator should opt for a high spot to deploy his ground station. The ground station will be investigated in more details in Chapter 10.

The backpack comprises all the maintenance tools needed to repair the UAV in case of failure. Spare parts are also provided.

The operator does not need extra resources. The GS runs with batteries and so does the UAV. A sufficient amount of batteries is provided since it is assumed that no electricity is available on the deployment site. An improvement of the system in the future will be the implementation of a green energy battery charger such as a portable windmill or a solar panel. But for now, the cost budget does not allow for this option.

In terms of human resources , one person is sufficient to carry and deploy the drone but for the ease of deployment the second operator will be of help. To operate the UAS it is required by Dutch law to have an operator and an observer (**Req-sys-10**).

3.2. Operations

The UAS will be able to perform missions which include mapping the area and detecting blocked roads, searching house by house for survivors, mapping the inside of a house and observing a building. All these tasks have to be performed with a certain level of autonomy. This level of autonomy has a large influence on the way the UAV is operated.

After the technical checks the operator switches the quadcopter on. It initiates itself and waits the input command from the operator. As soon as the mapping area and resolution have been determined and the take-off clearance given, the drone takes-off and flies over the area. The map is sent live to the GS computer where the operators can analyse the map and communicate the location of the eventual blocked roads to the rescuers.

The next mission is to check house by house if there are any survivors. The UAV is autonomous, therefore it is able to avoid obstacles and find its way through and out of the building. The operator receives continuously information about the status of the UAV on his screen. Moreover an audio and video live feed is available during all the operations except the mapping phase. The operator is able to control (redefine) the mission from the GS at any time.

The live feed is useful for the next mission which consists of the observation of a building that represents a potential hazard to the emergency services.

The last portion of the mission is a safe and precise landing close to the GS where the operator can collect the memory devices to analyse the high definition imagery. The user can replace the empty battery. The whole mission is performed in 30 minutes. So is the battery life.

3.3. Mission plan

This section focuses on the mission plan of the competition. The goal is to impress the jury: no optimisation is done to achieve the highest score possible. The location of the missions is given in Fig. 3.1. In this section the four main legs of the mission will be explained.

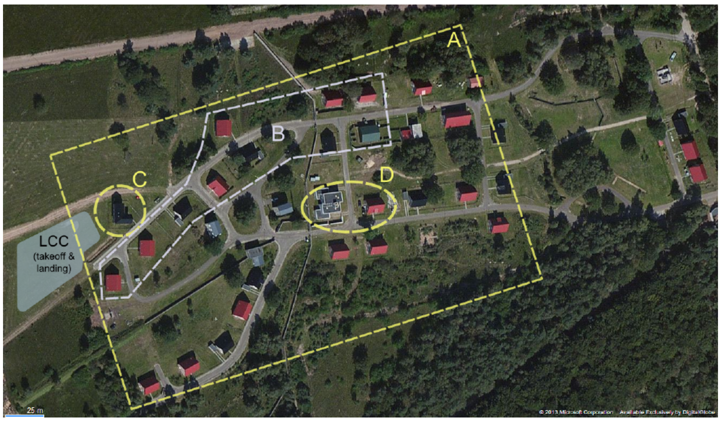

A
B
C
D
LCC
(takeoff &
landing)
25 m
© 2013 Microsoft Corporation Available Exclusively by DigitalGlobe

Figure 3.1: Location of the missions

Leg 1: mission A The drone will fly at an optimum altitude of 64 m to fulfil the resolution C and blurring requirements [29]. The spiral pattern used to map the area is explained in Chapter 6. Assuming a climbing speed of 6 m/s (Chapter 4) and a mission time of 306 seconds, the first leg takes 327 seconds.

Leg 2: mission D After leg 1, the drone is located at the centre of the map. This is the location of the building used for mission D. First it consists in landing on a roof: 45 seconds are spent for the landing and the take-off from the yellow cross.

The second part of mission D is the observation of the building. The UAV hovers in front of a board where digits are displayed. The time spent in front of this board is the time remaining: 30 minutes minus the time for all other legs. In summary 214 seconds are allocated for leg 2.

Leg 3: mission B Now comes the most time consuming mission: the house-by-house scan for survivors. At the end of leg 2 the UAV is still located in the centre of the area. To go from its initial point to the last house to be checked (the ninth), it will fly over a distance of 340 m with an assumed speed of 14 m/s. If a house is accessible, the UAV has to enter to look for survivors. It is assumed that the MAV spends 90 seconds per house on average. The total mission duration is estimated to be equal to 13 minutes 54 seconds.

Leg 4: mission C The last leg will take place close to the landing site. The mission consists in a room by room scan. The drone is able to identify and locate different items in the house. Assuming that there are seven rooms in the house and that 45 seconds are spent per room, it will take 5 minutes and 22 seconds to perform the mission. Note that the house is located at 100 m from the last house of mission B and therefore, a travel time of 8 seconds has to be added.

Summary The mission planning time distribution is provided in Table 3.1. Note that one minute is given to land and take-off. The pre-mission operations are not taken into account in the table. The time available is equal to 1800 seconds.

Table 3.1: Mission planning time distribution

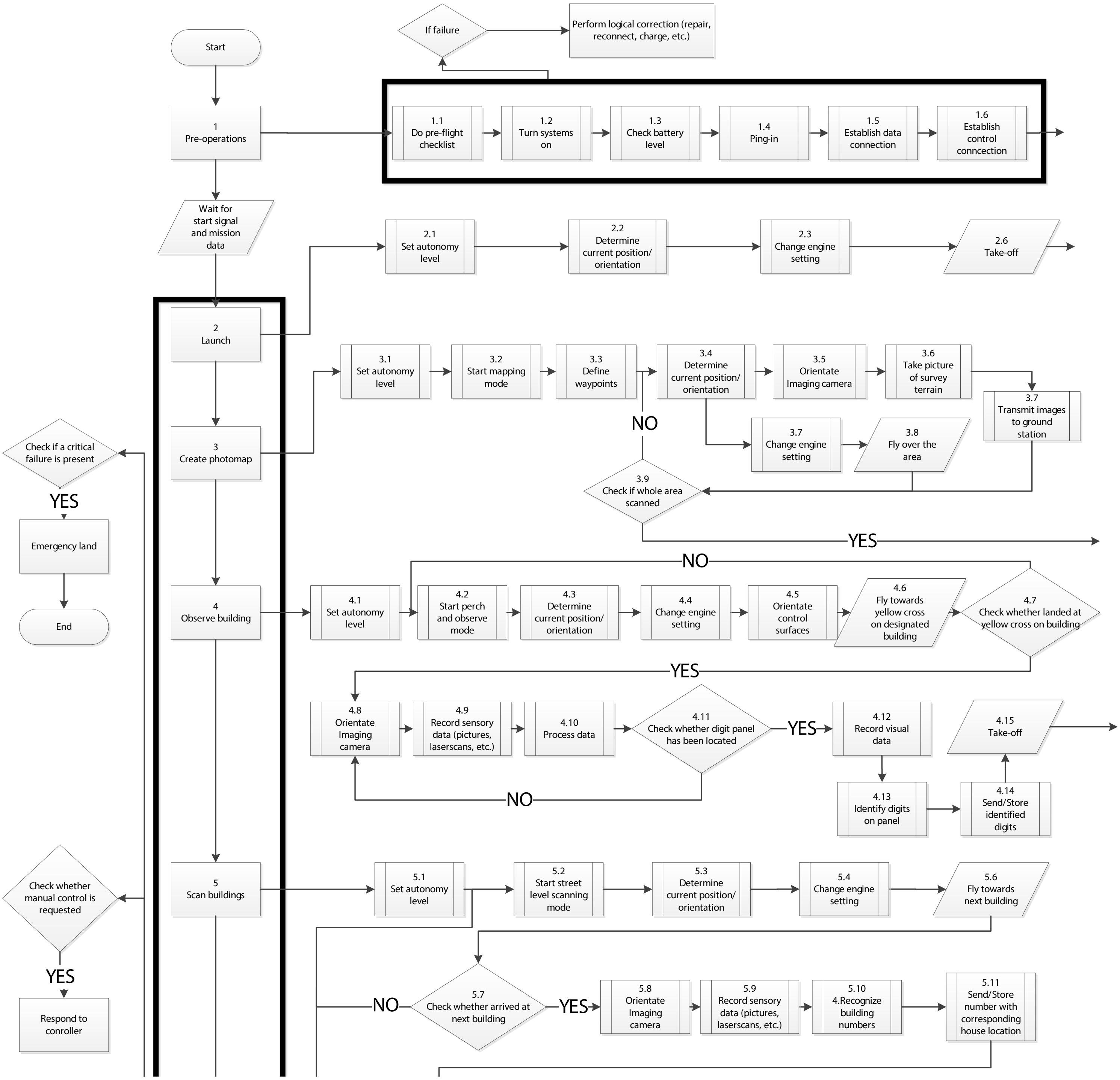
	Cruising speed [m/s]	Distance/ Height [m]	Time [s]
Take-off			30
Mission A	6	38	327
Mission D	14	50	214
Mission B	14	340	834
Mission C	14	100	322
Landing			30
Total			1757

3.4. Functional analysis

The functional analysis consists of two parts used to illustrate the different functions that the UAV system should perform: the FFBD, Functional Flow Block Diagram, and the FBS, Functional Breakdown Structure. The functional analysis is performed for all the subtasks of the IMAV mission predefined.

3.4.1. Functional flow block diagram

Figure 3.2 represents the logical flow of functions that the UAV system will have to perform.


Start
If failure
Perform logical correction (repair, reconnect, charge, etc.)
1 Pre-operations
1.1 Do pre-flight checklist
1.2 Turn systems on
1.3 Check battery level
1.4 Ping-in
1.5 Establish data connection
1.6 Establish control conncection
Wait for start signal and mission data
2 Launch
2.1 Set autonomy level
2.2 Determine current position/ orientation
2.3 Change engine setting
2.6 Take-off
3 Create photomap
3.1 Set autonomy level
3.2 Start mapping mode
3.3 Define waypoints
3.4 Determine current position/ orientation
3.5 Orientate Imaging camera
3.6 Take picture of survey terrain
3.7 Transmit images to ground station
3.7 Change engine setting
3.8 Fly over the area
NO
3.9 Check if whole area scanned
YES
Check if a critical failure is present
YES
Emergency land
End
4 Observe building
NO
4.1 Set autonomy level
4.2 Start perch and observe mode
4.3 Determine current position/ orientation
4.4 Change engine setting
4.5 Orientate control surfaces
4.6 Fly towards yellow cross on designated building
4.7 Check whether landed at yellow cross on building
YES
4.8 Orientate Imaging camera
4.9 Record sensory data (pictures, laserscans, etc.)
4.10 Process data
4.11 Check whether digit panel has been located
YES
4.12 Record visual data
4.15 Take-off
NO
4.13 Identify digits on panel
4.14 Send/Store identified digits
Check whether manual control is requested
YES
Respond to conroller
5 Scan buildings
5.1 Set autonomy level
5.2 Start street level scanning mode
5.3 Determine current position/ orientation
5.4 Change engine setting
5.6 Fly towards next building
NO
5.7 Check whether arrived at next building
YES
5.8 Orientate Imaging camera
5.9 Record sensory data (pictures, laserscans, etc.)
5.10 4.Recognize building numbers
5.11 Send/Store number with corresponding house location

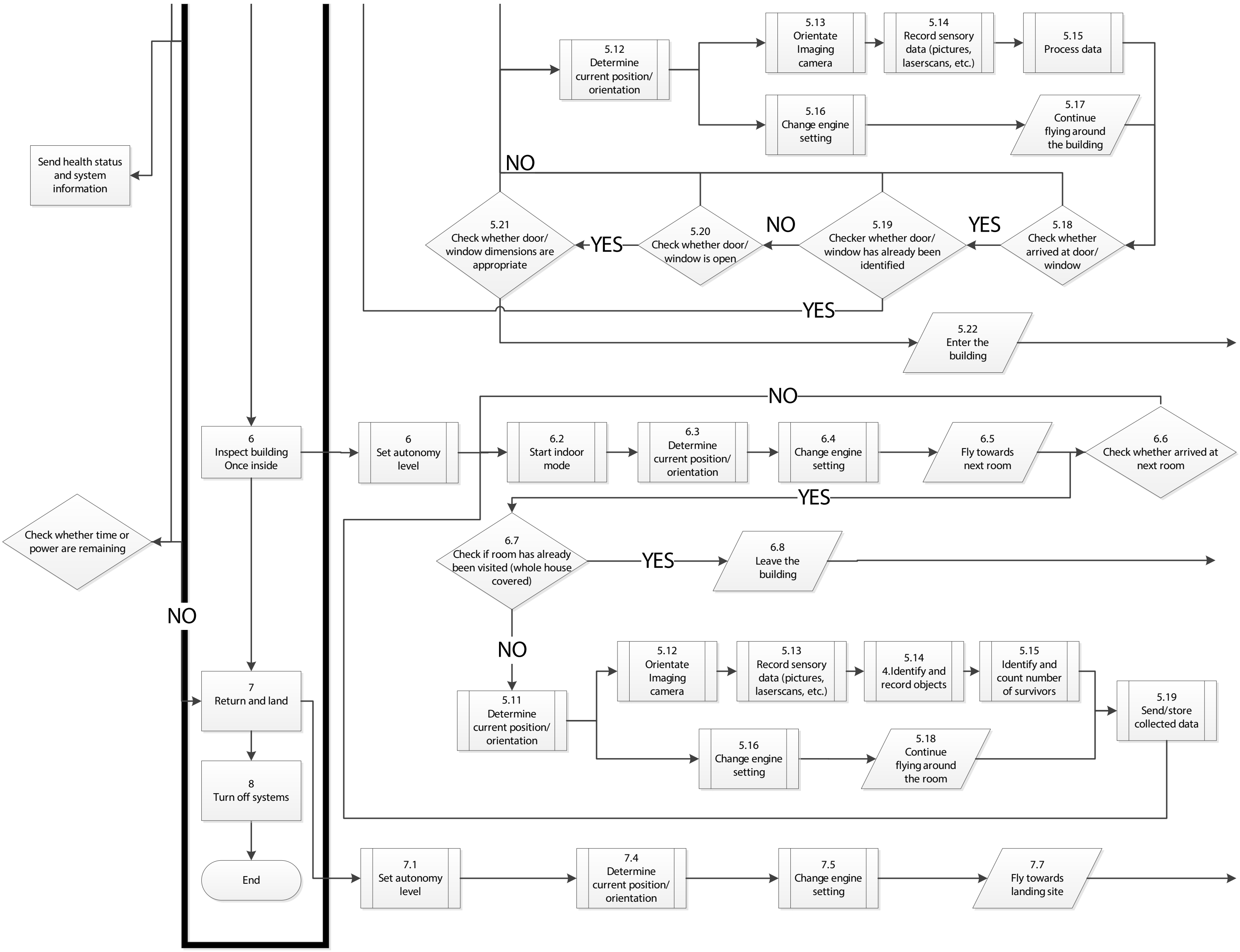

5.12 Determine current position/ orientation
5.13 Orientate Imaging camera
5.14 Record sensory data (pictures, laserscans, etc.)
5.15 Process data
5.16 Change engine setting
5.17 Continue flying around the building
Send health status and system information
NO
5.21 Check whether door/ window dimensions are appropriate
YES
5.20 Check whether door/ window is open
NO
5.19 Checker whether door/ window has already been identified
YES
5.18 Check whether arrived at door/ window
YES
5.22 Enter the building
NO
6 Inspect building Once inside
6 Set autonomy level
6.2 Start indoor mode
6.3 Determine current position/ orientation
6.4 Change engine setting
6.5 Fly towards next room
6.6 Check whether arrived at next room
YES
Check whether time or power are remaining
6.7 Check if room has already been visited (whole house covered)
YES
6.8 Leave the building
NO
NO
5.11 Determine current position/ orientation
5.12 Orientate Imaging camera
5.13 Record sensory data (pictures, laserscans, etc.)
5.14 4.Identify and record objects
5.15 Identify and count number of survivors
5.19 Send/store collected data
5.16 Change engine setting
5.18 Continue flying around the room
7 Return and land
8 Turn off systems
End
7.1 Set autonomy level
7.4 Determine current position/ orientation
7.5 Change engine setting
7.7 Fly towards landing site

Figure 3.2: Functional flow block diagram of the MAV system

TIME-INDEPENDENT FUNCTIONS

Since the diagram represents the logical order of functions, the function blocks are time-dependent. However, the system also has to perform time-independent functions. These are illustrated by non numbered blocks on the left and the top of Fig. 3.2. The time-independent functions are related to flight worthiness, safety, controllability, system status and communications.

TIME-DEPENDENT FUNCTIONS

The remaining blocks are numbered in a chronological order and illustrate the time-dependent functions.

The first functions to be performed take place before the actual flying activities (1). Those are similar to common aircraft: preflight checks are held. Examples of inspection elements are the structural, electric devices, rotors and the sensors.

After the vehicle is launched in the air (2), its first mission element is to create a map of the area (3). The MAV has to fly over the area while orientating the observation sensor. If it does not take one big picture of the area, the different pictures will have to be assembled. Once the map is created (3.12), the MAV system needs to indicate which roads are blocked (3.13).

The next mission element consists of scanning buildings from the outside(4). The MAV should also be able to detect openings to enter the building. Once inside (5), it will inspect every accessible room of the building looking for survivors (5.15). It will then report the number of survivors and eventually provide a picture to facilitate the SAR operations.

The last mission element is to land on a yellow cross located on a flat roof. As soon as it has landed, the

MAV has to observe the surroundings: read digit numbers off a board on a building across the street (6.13). After the digit sequence is recorded or sent, the MAV has to take-off from the roof and land where it originally took off (7).

The drone requires additional general functions as well to perform the missions successfully. Examples are changing its autonomy level, motor settings and attitude.

3.4.2. Functional breakdown structure

The required functions can now be grouped per subsystem in FBSs. These trees are presented in Fig. 3.3. Two AND trees were created; one for the general functions and one for the mission specific functions. The first tree shows the general functions to be performed. These are to process the data, fly, control, provide power, navigate, communicate and to provide guidance. The second tree is related to the mission functions. An identification letter is assigned to each of the different functions for traceability in later phases.

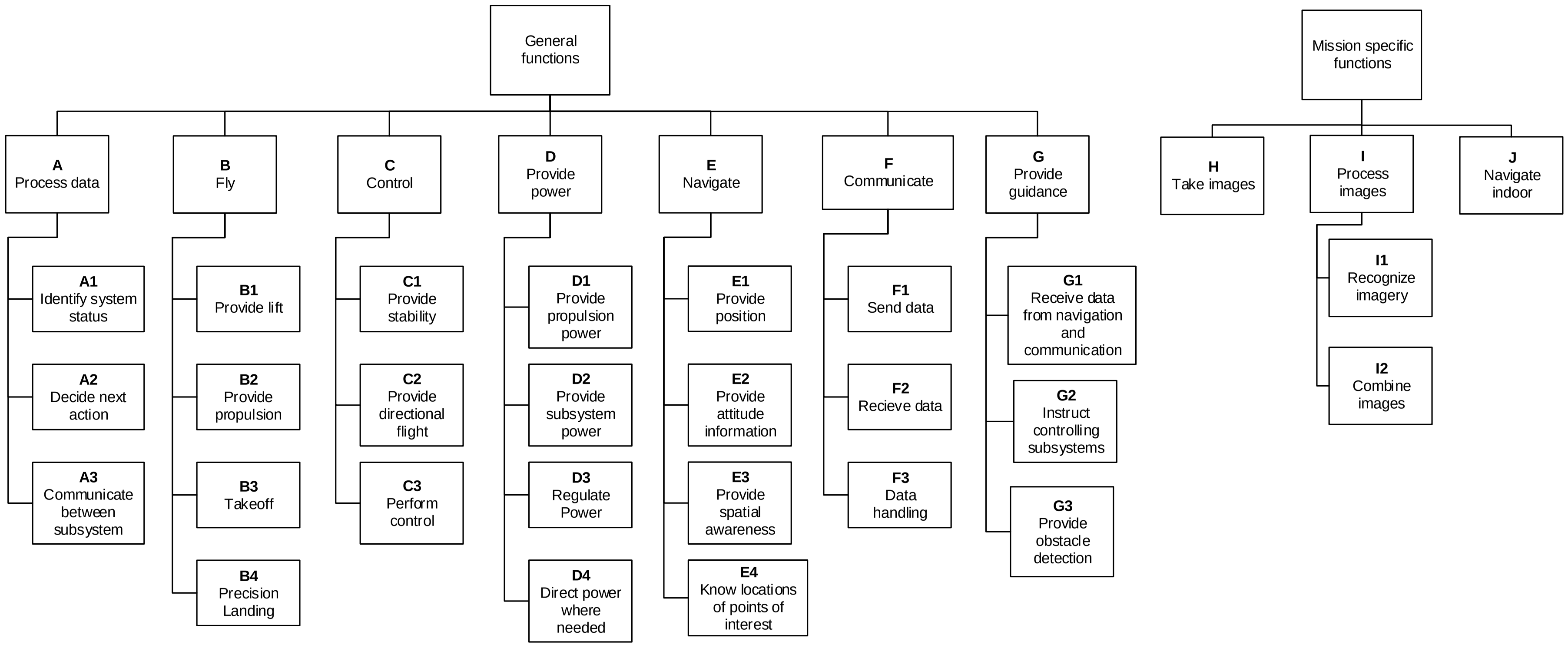

General functions
A Process data
A1 Identify system status
A2 Decide next action
A3 Communicate between subsystem
B Fly
B1 Provide lift
B2 Provide propulsion
B3 Takeoff
B4 Precision Landing
C Control
C1 Provide stability
C2 Provide directional flight
C3 Perform control
D Provide power
D1 Provide propulsion power
D2 Provide subsystem power
D3 Regulate Power
D4 Direct power where needed
E Navigate
E1 Provide position
E2 Provide attitude information
E3 Provide spatial awareness
E4 Know locations of points of interest
F Communicate
F1 Send data
F2 Recieve data
F3 Data handling
G Provide guidance
G1 Receive data from navigation and communication
G2 Instruct controlling subsystems
G3 Provide obstacle detection
Mission specific functions
H Take images
I Process images
I1 Recognize imagery
I2 Combine images
J Navigate indoor

Figure 3.3: Functional breakdown structures of the MAV system

4

Propulsion & aerodynamics

This chapter discusses the design of the propulsion subsystem, together with its (aerodynamic) performance. Section 4.1 discusses the general lay-out/configuration of the quadcopter. In Section 4.2 various propeller airfoils are proposed and analysed in detail. Hereafter, Section 4.3 continues with sizing the propellers, including a chord and twist/pitch distribution along the span. Section 4.4 discusses the selection of an appropriate motor. Finally, a discussion of the results is presented in Section 4.6.

4.1. Configuration

Having a quadcopter design still leaves a lot of room in designing for a specific configuration. In effect, determining what propeller mechanism should be used can be quite an extensive investigation in and of itself. To that end, conventional fixed pitch propellers, variable pitch propellers, variable tilt-rotors and fixed tilt-rotors were evaluated. Advantages of the variable pitch and variable tilt-rotor are mainly a higher degree of control and better manoeuvring performance. Given the conditions under which one would expect the drone in question to perform its missions, a general design philosophy centred around robustness of the drone was adopted. Therefore, after some preliminary analysis, it was determined that the moving parts involved with variable pitch propellers and tilt-rotors conflicted with the aforementioned design philosophy. These parts would decrease the reliability of the system and would also make repairing and maintaining the drone far more of a challenge. Therefore, only a trade-off between the conventional fixed pitch propeller and a fixed tilt-rotor has been made.

4.1.1. The tilt-rotor

Before proceeding, some insight into the fixed tilt-rotor is required. This design mainly aims to be more fail-safe. Its working principle can best be described by looking at Fig. 4.1. Consider for example an engine failure, represented by the cross in the figure. In order to maintain stable flight, the UAV must recognise this with its gyroscopes (Chapter 6) and shut down the opposing motor to remain level. Now, a quadcopter with a conventional configuration will start spinning in the shown plane due to the torques created by the motors (note that opposing motors always need to spin in the same direction to make yaw control possible with all engines operational). However, the tilted-rotor has small thrust components oriented such that they cancel the motor torque.

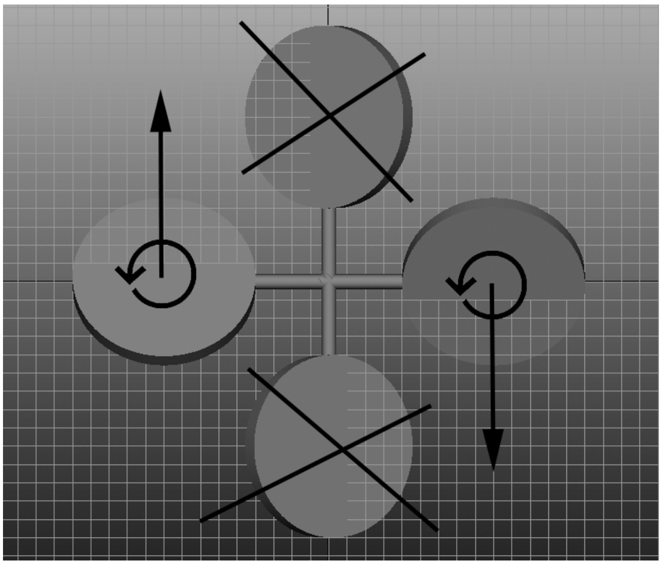


Figure 4.1: Torque cancellation of a fixed tilt-rotor quadcopter concept

An estimation of the magnitude of the tilt angle can be calculated by using data from preliminary sizing estimations made earlier by the team and Eqs. (4.1) to (4.3). The moment arm of the x-component of the total thrust force to the middle of the quadcopter, r, is equal to 0.25 m. The power per motor, P, equals half (two rotors operational) of the total original power requirement, being 60 W. The vertical thrust component per rotor, T_y, equals a half of the original vehicle weight, being 13.2/2 N. In the equations, the horizontal thrust component, T_x, counteracts the motor torque over the arm, r. Other than that, the equations should be self-explanatory.

$$\tau = \frac{P}{\Omega} \tag{4.1}$$

$$T_x = \frac{\tau}{r} \tag{4.2}$$

$$\theta = \tan^{-1}(\frac{T_x}{T_y}) \tag{4.3}$$

Assuming two extreme numbers of the motor rotational velocity, Ω, an estimate of the minimum and maximum tilt angle can be obtained. The tilt angle, θ, equals 4.0 or 1.0 degrees for an rpm of 5,000 (524 rad/s) and 20,000 (2,094 rad/s) respectively. Note that the higher the rotational speed, the lower the torque.

4.1.2. TRADE-OFF

In terms of the trade-off, five criteria were considered, which are discussed in the following paragraphs. The weighted sum method was used, with weights ranging from 1 to 3, and scores from 1 to 5; 5 being the best and 1 the least. Ultimately, the propeller with the highest weighted sum will be chosen. The trade-off table can be found in Table 4.1.

Table 4.1: Quadcopter configuration trade-off

		Score		Weighted Score	
Criterion	**Weight**	**Tilted**	**Conventional**	**Tilted**	**Conventional**
Efficiency	2	4	5	8	10
Robustness	3	3	5	9	15
Manufacturability	1	2	5	2	5
Failsafe	2	5	4	10	8
Manoeuvrability	2	5	4	10	8
			Total	39	46
			Normlised	0.458824	0.541176

Efficiency

The first criterion evaluated was efficiency. While somewhat self-explanatory, this criterion essentially refers to the amount of lift or thrust being dissipated, thereby giving rise to higher power requirements and potentially a greater mass. Since efficiency affects so many aspects of flight performance, it receives a relatively high weight. Since the tilted-rotor is by definition mounted at an angle, part of the thrust is not used for providing lift. However, since the required tilt angle is relatively small, the thrust dissipation is small as well. Therefore, the tilted-rotor scores only 1 point less, compared to the conventional lay-out.

Robustness

Robustness refers to the UAV's ability to withstand impact and cope with physical damage. Since this is an integral part of the group's design philosophy, it is given the highest possible weight. As per the structures department, the conventional fixed pitch quadcopter can be designed such that it is much more robust, while the general functionality of the tilted-rotor is easily jeopardized with long-term use. Small angular offsets would already nullify the tilt-rotor's general working principle.

Manufacturability

Manufacturability is a criterion worth considering, yet as long as it can be done at a reasonable cost with the required precision, it is not of paramount importance. The conventional fixed pitch rotor is renowned for its ease of production while the tilt-rotor depends greatly on precision of production, thereby making the process far more complicated and thus costly.

Failsafe

Failsafe refers to the UAV's ability to retain a certain performance level in the event of an engine failure or damage or loss of a propeller. Specifically speaking, the UAV should be able to fly to a desired location and then land safely, with one engine down. This is a rather desirable feature, especially since the system may be costly and might have important information on board that needs to be retrieved. In case of an engine failure, a conventional fixed pitch propeller can make use of certain software so as to land itself safely, however it has little to no control over where it lands. This could be problematic if the UAV is for instance flying over water. The fixed tilt-rotor on the other hand retains full control and can therefore be guided more precisely towards a specific landing location.

Manoeuvrability

The fifth criterion is manoeuvrability, which is another advantage of the tilted-rotor design. For this criterion it is assumed that both the tilted-rotor and the conventional design are equipped with the same electric motors. Taking the conventional design as a reference, the tilted rotor design is more stable in one direction and less stable in the other flight direction. Since the UAV has full yaw control, it can always fly in the "less stable" direction, while being less susceptible to for example gusts in the other direction. This is particularly useful for flying through narrow spaces such as doors and windows. As with efficiency, the difference in scoring is minimal due to the small tilt angle.

4.1.3. Sensitivity analysis and discussion

As seen in Table 4.1, the conventional configuration is found to be optimal with respect to the above mentioned criteria. It has a normalised score of 54 % as opposed to the tilt rotor which has a score of 46 %. At first glance this difference may not seem significant enough for any conclusions to be drawn. Therefore a sensitivity analysis is carried out in order to lend greater credibility to the trade-off process. The sensitivity analysis consists of modifying the weights and scores of certain criteria such that it has a negative impact on the winning option. This allows one to see whether small changes in inputs could in fact influence the outcome of the trade-off. In terms of weights, increasing the weight of the failsafe or manoeuvrability criteria results in the conventional configuration having a 7 percentage point advantage over the tilt. In addition, decreasing the weight of the robustness criterion also results in the conventional having 6 percentage points more than the tilt. In terms of scores, the most extreme scenario considered was decreasing the score of the conventional configuration for robustness from a 5 to a 3. Even in this instance the conventional configuration was the winner, albeit by a difference of 1 percentage point. Therefore, the conventional configuration was chosen for the design.

4.2. Propeller airfoil selection

In terms of designing the propellers, a good starting point is to investigate appropriate airfoils. While there are several airfoil characteristics that can be interesting to investigate, the most critical were found to be the optimal lift over drag coefficient, $\frac{C_l}{C_d}_{opt}$, the lift coefficient C_l at the optimal point and the thickness to chord ratio t/c. These three criteria were considered during the trade-off between airfoils.

When performing the trade-off, the weighted sum method was used. Weights were allotted from 1 to 3, with 3 being the most important criterion. The $\frac{C_l}{C_d}_{opt}$ was found to be the most important criterion as it relates directly to the efficiency with which the propeller can function. Given that it has a strong influence on the performance of the rotors and on the entire UAV, this criterion was given a weight of 3. The lift coefficient C_l at this optimal point relates directly the surface area of the propellers. The higher the lift coefficient, the smaller the required propeller surface, thereby leading to a reduction in mass. Needless to say, this is beneficial for the system. Finally, the thickness, t/c also relates to the mass of the propeller, as a thinner airfoil uses less material and therefore has a smaller mass. Since both the C_l and t/c mostly relate to mass in this context, they are both assigned a weight of 1. They are deemed to be of less importance than the $\frac{C_l}{C_d}_{opt}$ which is why they carry only a third of the weight.

During the course of preliminary research, four main airfoil categories were identified, namely the ARA-D, Eppler, Onera and MH series. Although other categories do exist, these four constitute the primary focus of this investigation. The above mentioned data was collected for eleven different airfoils at a Reynolds number of $1{\cdot}10^5$ and was acquired using Javafoil [30] and an airfoil database [31]. The data is presented in Table 4.2. Chapter 12 can be referred to for a validation of Javafoil and the data in [31].

The fact that there are quantitative values available for each of the criterion under consideration makes this trade-off fairly straight forward. First, the values obtained for a certain criterion are normalised by expressing each value as a function of the highest possible value. Second, these normalised values are multiplied by the assigned weights. Subsequently these normalised, weighted values are summed. It should be noted however that the thickness score should be subtracted from the total as having a higher maximum thickness is actually detrimental to the UAVs performance. Once this is done, the final scores are normalised with respect to the sum of all the scores. This provides a good overview of how much a specific airfoil wins by. All this data is presented in Table 4.2.

Table 4.2: Airfoil characteristics and trade-off

	Criterion			Weighted and Normalised			Total	Normalised Total
Airfoil	**C_l/C_d[−]**	**C_l[−]**	**Max t/c [%]**	**C_l/C_d[−]**	**C_l[−]**	**Max t/c [%]**		
arad10-il	44.74	0.85	10	2.10	1.13	0.84	2.39	6.94
arad6-il	63.89	1.15	6	3.00	1.53	0.50	4.03	11.68
e850-il	38.67	0.58	8	1.82	0.77	0.67	1.92	5.55
e853-il	60.00	1.08	11.1	2.82	1.44	0.93	3.32	9.63
e854-il	55.91	1.23	13.4	2.63	1.64	1.13	3.14	9.10
hor04-il	47.27	0.52	4.1	2.22	0.69	0.34	2.57	7.44
hor07-il	57.89	1.1	7.2	2.72	1.47	0.61	3.58	10.38
hor12-il	57.69	1.5	11.9	2.71	2.00	1.00	3.71	10.75
mh114-il	58.33	1.75	13	2.74	2.33	1.09	3.98	11.53
mh121-il	47.78	0.86	8.8	2.24	1.15	0.74	2.65	7.68
mh116-il	51.25	1.23	9.9	2.41	1.64	0.83	3.21	9.32

Therefore it can be concluded that the ARAD6-il is the best airfoil as per the trade-off. The airfoil profile, $C_l - \alpha$ curve and lift-drag polar are shown in Figs. 4.2, 4.3 and 4.4 respectively.

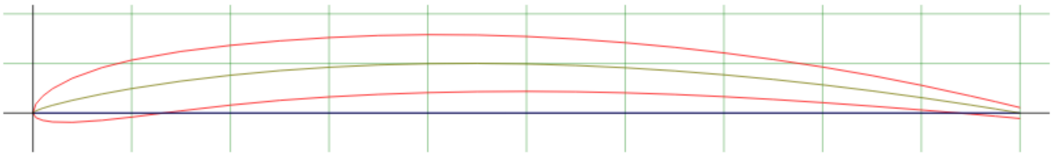


Figure 4.2: ARAD6-il Airfoil Profile

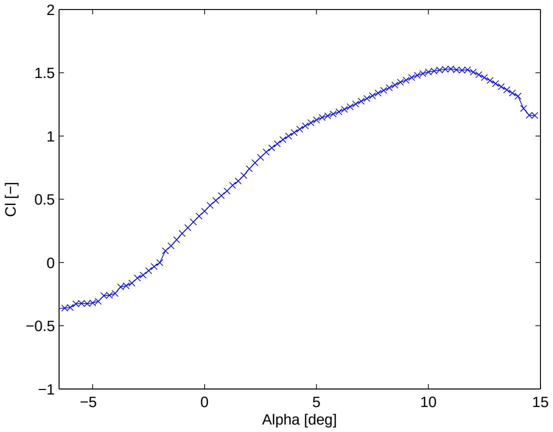


Figure 4.3: $C_l - \alpha$ curve ARAD6-il

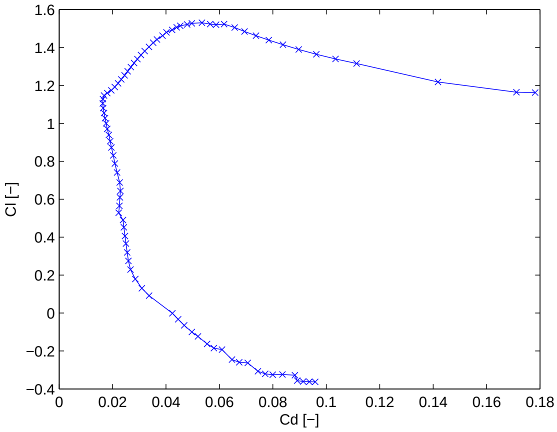


Figure 4.4: Lift drag polar ARAD6-il

4.2.1. Sensitivity analysis

As can be seen, ARAD6-il scores the highest with 12.3 % of the total score, with the second best HOR07-il obtaining 10.7 %. While this difference seems rather small, it should be noted that there are eleven airfoils involved and therefore as a percentage it is difficult to have very big gaps between airfoils. In performing a sensitivity analysis, the weights of the criterion were changed one at a time to gauge whether ARAD6-il still emerged victorious from the trade-off. The weight of $\frac{C_l}{C_d}_{opt}$ was lowered from 3 to 2, C_l was increased from 1 to 2 and t/c was also increased from 1 to 2. In all three instances ARAD6-il was found to be the best, however in the case of the C_l weight being increased, the ARAD6-il won by 0.2 %. Although this is not a very significant margin, given that the ARAD6-il consistently scores the highest, it can be considered to be the best airfoil.

4.3. Propeller design

The main performance driving parts of the quadcopter have to be its propellers. Since every vehicle is unique, propellers can best be custom-made. Moreover, this proves to be more sustainable in the long run in terms of efficiency. This section elaborates on the design steps that are required for a proper propeller sizing.

4.3.1. Design point

With the airfoil selected, the actual propeller can be designed. This propeller will be sized for optimal efficiency at hovering, since most of the thrust force is used for generating lift in any flight condition. Later on, in Section 4.5, this will actually be proven to be the case, making this a valid assumption. The total required thrust for hovering is in principal equal to the weight of the vehicle. This weight was iteratively found to equal 14.1 N (1.44 kg). For the first iteration, the team used data from a previously made preliminary design. The weight of this design was 13.2 N (1.35 kg).

Now, the thrust per motor can ideally be found by dividing the total required thrust by the number of motors (four). However, the mounting supports for the motors will always block part of the flow and thus induce some additional drag. The required increase in thrust should thus be calculated with the basic drag formula, as given in Eq. (4.4). Most of the parameters in the formula were iteratively determined. For example, the exposed surface area, S, will be larger for a larger propeller radius. For a propeller radius of 0.1 m (see Section 4.3.3) and a beam width of 0.03 m (see Chapter 5), the surface area equals 0.003 m^2. Similarly, V was found to be 6.9 m/s. C_D was found from [32] and equals about 1.2 for high aspect ratio beams. This gives a total drag of 0.10 N and brings the total required thrust per propeller to 3.63 N. Other parts of the body are hardly exposed in the flow and have not been taken into account, since the drag of the (fully exposed) beams is already minimal.

$$C_D = \frac{D}{0.5\rho V^2 S} \tag{4.4}$$

4.3.2. Design methods

Several propeller design methods exist:

- **Glauert's Blade Element Method**: the propeller is cut down into small sections of equal surface area, which can be sized separately in terms of chord length and pitch angle. This method assumes idealised wing theory and does not take any 3D effects such as tip vortices into account. Therefore, it is best suited for preliminary sizing only.

- **Prandtl's Lifting Line Theory**: an extension of the Blade Element Method which analyses each section as a theoretical airfoil, represented by a vortex. This methods gives a first order approximation of 3D effects, but only works well for high aspect ratio wings and propellers [33].
- **Extension of classical methods**: a whole range of methods that apply correction factors and other improvements to the classical element and vortex methods [33]. Often the propeller is not only divided in span-wise sections, but also in chord-wise sections. The various theories give high agreement with experimental data (see also Chapter 12), but can only be solved numerically.

4.3.3. Propeller sizing

Due to the limited accuracy of traditional Blade Element and Lifting Line methods, it has been decided to use a program that makes use of more advanced theories. The program that has been opted for is QProp, which is fully validated in Chapter 12. In addition to the full airfoil characteristics, it takes the following parameters as input:

- Required thrust for design point, 3.63 N, see above.
- Axial inflow velocity, u_0, which is the air velocity infinitely far ahead of the propeller. For hovering this is 0.0 m/s, assuming no thermals or columns of moving air.
- Number of rotor blades.
- Rotor radius, by definition equal for all blades.
- Rotational velocity in rpm for the design point.

The first two parameters are fixed for a specific design point, while the other three parameters have to be chosen based on an iteration with the output variables, as listed below:

- Chord length and pitch angle per radial section: these values define the geometry of the propeller. Values are optimised for optimal theoretical efficiency, but must be checked for practicality. Especially the chord length might for example get smaller than a millimeter or larger than the rotor radius.
- Actual thrust, which equals the input thrust during design. However, rotating the propeller at a different rotational velocity than that used for the design point, outputs a different thrust.
- Required shaft power (and torque): the main design parameter, which must be as low as possible.
- Propeller efficiency, which QProp defines as in Eq. (4.5). This value is not of use for hover calculations, since $u_0 = 0.0$ which makes the efficiency go to 0.0 as well. u_e is the axial propwash velocity.
- Other sectional parameters: operating lift, drag and swirl coefficient, Reynolds number, propwash velocity and Mach number.

$$\eta_{prop} = \frac{2}{1 + \frac{u_e}{u_0}} \tag{4.5}$$

The iterative process for sizing the propeller can be defined with the following steps:

1. Set all known values for the design point: airfoil characteristics, thrust and inflow velocity.
2. Set the rotor radius.
3. Set the rotational velocity and change until lowest value of shaft power is found.
4. Check if chord and pitch angle distributions are feasible and iterate by changing the number of blades (more blades, smaller chord lengths).
5. If required, start at step 2 again.

A few remarks about the process can be made. First of all, larger rotor radii are always more efficient in theory, but have an unrealistic chord distribution after some point. As such the final radius was found to be 0.10 m. The rotational velocity for hovering was found to be 9,150 rpm for hovering, requiring a minimum of 31 W of shaft power. The total number of blades per propeller is 2.

4.4. Motor selection

The motors should provide the required power at the selected rotational velocity, while operating at an as high as possible efficiency. Due to the wide range of available electric motors (**Req-flight-6**), use was made of a motor catalogue [34]. Though extensive, the data in this catalogue was by no means assumed to be correct and has always been checked with the actual manufacturers data.

From Section 4.3, the required shaft power for hover was found to be 31 W. Since this is the design point, the motor should run at optimal efficiency. However, the minimum power required is assumed to be at least 80 W. This value was found with QProp, by systematically increasing the rotational velocity of the propeller till a total (static) thrust-to-weight ratio of 1.75 was reached. A ratio of 1.75 was iteratively found to be sufficient for most flight manoeuvres (Chapter 7) and in terms of forward flight velocity.

Another important sizing parameter is the voltage constant. The voltage constant, shortly Kv, refers to the increase of rotational velocity (rpm) per increase in electric potential (Volt). The minimum value can be set by knowing the required rotational velocity for hover (9,150 rpm, see Section 4.3) and the battery voltage (± 11.4 V, see Section 9.1); this equals about 800 RPM/V. The maximum value cannot be estimated as being the rotational velocity required for a thrust-to-weight ratio of 1.75 divided by the battery voltage. This is because in a non-ideal scenario, the rotational velocity is also dependent on the actual power consumption/current flow through the motor. The higher the current flow, the lower the actual maximum rotational velocity that can be achieved at a certain voltage. As a result, high power output is commonly only achieved with lower rotational velocities, which makes the point of maximum thrust/power consumption/rotational rate, the most critical to design for.

Inputting both the minimum power limit and voltage constant, the catalogue can compile a list, ordered based on weight (lowest weight first). Going down the list, every motor was analysed for being capable of reaching a rotational velocity of 15,000 rpm at the discussed output power of 80 W. The motor that was capable of doing this with the highest efficiency was selected; weight appeared to be a relatively unimportant criteria (spread of a few grams). The selected motor is the Turnigy Park 300-1080, which operates with the Turnigy Plush speed controller. Their dimensions and characteristics are summarised in Tables 4.3 [34] [35] and 4.4 [36]. The motors are located on top of their support beams and the speed controller is located on top of the beam, but closer to the centre.

Table 4.3: Motor specifications

	Turnigy Park 300-1080
Mass [kg]	0.025
Dimensions [mm x mm]	27 x 23
Maximum current [A]	9.0
No-load current [A]	0.2
Voltage constant [rpm/V]	1,080
Internal resistance [ω]	0.04
Average efficiency [%]	93
Cost [€]	10.90

Table 4.4: Electronic speed controller specification

	Turnigy Plush Speed Controller
Mass [kg]	0.009
Dimensions [mm x mm x mm]	27 x 17 x 6.0
Maximum continuous current [A]	10
Maximum burst current [A]	12
Assumed efficiency [%]	97
Cost [€]	6.90

4.5. Performance

After having selected all performance related components, the performance of the UAV can be analysed. Hovering (Section 4.5.1) and forward flight (4.5.2) are especially evaluated in detail. It must be noted that this section also makes use of the final lay-out and sizing of other subsystems for aerodynamic calculations. However, the final configuration has not been motivated yet, but will generally be explained in the remainder of this report.

4.5.1. Hovering

The maximum static thrust can be calculated by changing the rotational velocity of the propeller in QProp. The static thrust represents an instantaneous response of the vehicle from a hovering condition (**Req-flight-**

3), assuming zero inertia of the propellers. This data is not useful for researching exact flight performance, but can be used in prototype sizing of the control system, see Chapter 7. The static thrust curve can be found in Fig. 4.5.

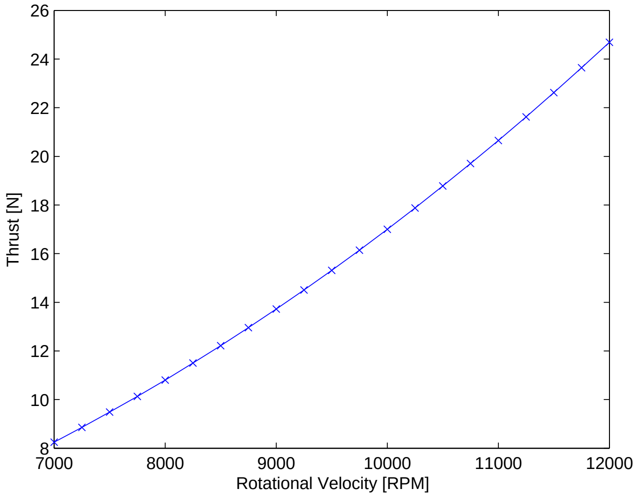


Figure 4.5: Static thrust of the quadcopter as a function of rotor rotational velocity

Another relevant curve that can be plotted for hovering flight is related to the so called ground effect. This effect causes an increase in lift when flying close above the ground, which can be calculated with Eq. (4.6). This equation showed compliance with flight tests [37]. In this equation, T_{hover}, is the required thrust for hovering as calculated in Section 4.3. r refers to the rotor radius and h is the flying height. Note that this height is at minimum 0.18 m, due to the structural height of the vehicle. As can be seen in Fig. 4.6, the increased lift due to the ground effect is negligible.

$$L = \frac{T_{hover}}{1 - \frac{r^2}{16h^2}} \tag{4.6}$$

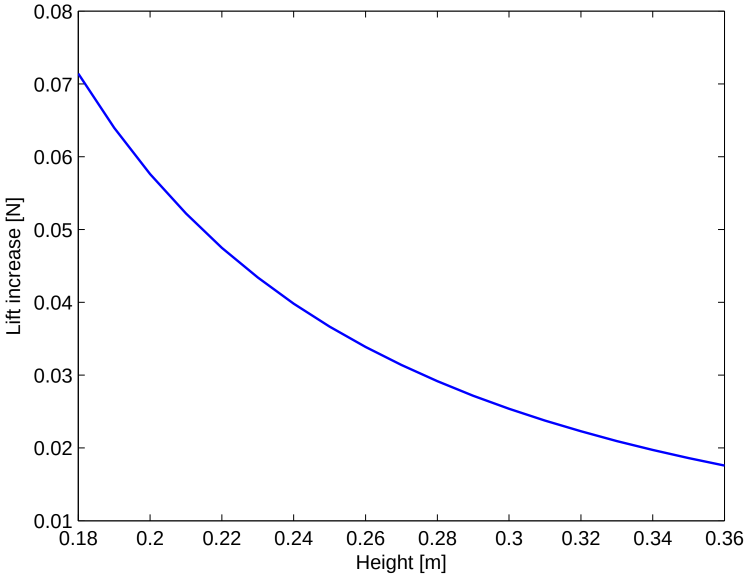


Figure 4.6: Lift increase per quadcopter propeller due to the influence of the ground effect

4.5.2. Forward flight

More interesting to know is if the exact correlation between flight velocity, power consumption, rotational velocity of the propellers, total drag and angle of attack can be found. To this end a program called Solidworks has been used, see Chapter 12. Where QProp can only analyse performance in one dimension, Solidworks can analyze in three. Moreover, QProp is only suitable for propellers and cannot analyse the drag of the rest of the structure. Therefore, the entire propeller has been modelled in Solidworks, together with the vehicle structure, payload and wiring, see Fig 4.7 and 4.8.

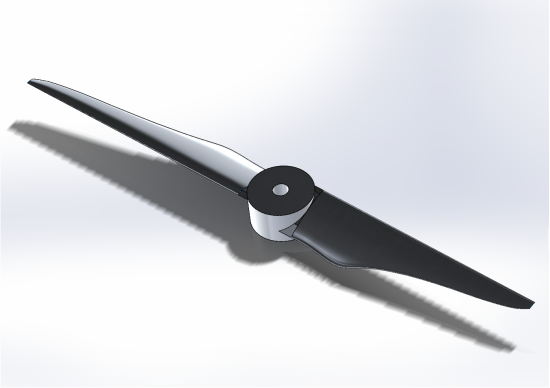


Figure 4.7: Visual representation of the designed propeller in Solidworks

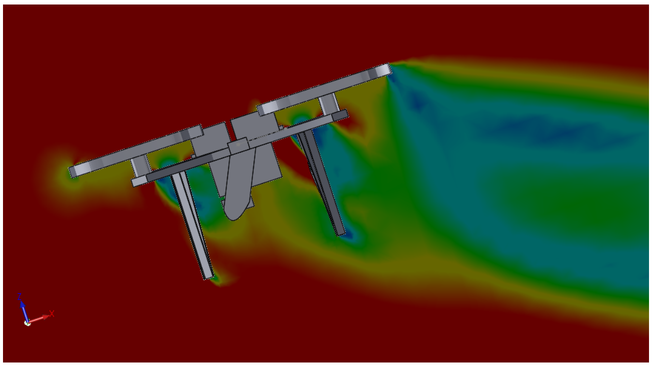


Figure 4.8: Representation of the quadcopter in Solidworks

First of all, the propeller was analysed separately to confirm compliance with QProp. However, the results of the Solidworks compiler would never converge within a reasonable amount of time, but rather takes hours to complete with the desired precision. Apparently, analysing a rotating object requires quite some computational power. Therefore, it has been decided to model the propellers as circular disks with equal radius and thickness, producing a uniform axial prop-wash velocity. Although this outflow velocity changes as a function of radial distance, the spread was usually found to be low, which makes this a valid assumption. Moreover, the only structural parts in the wake of the propeller are their mounting beams, which make up only a minor part of the drag.

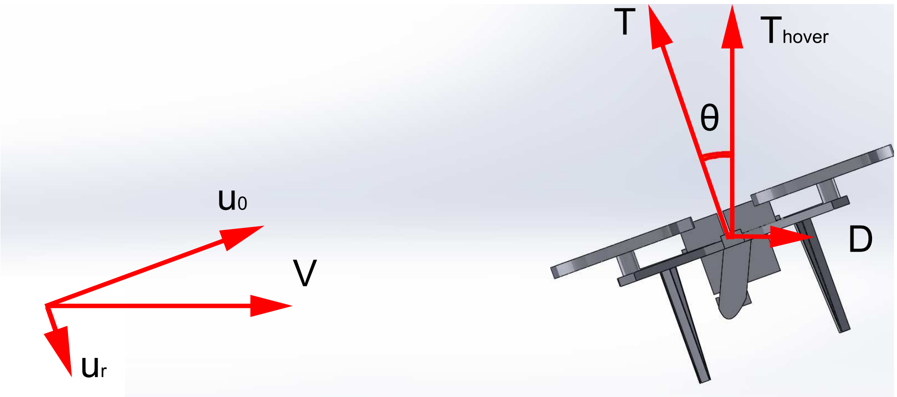


Figure 4.9: Relevant forces and velocities acting on the quadcopter in forward flight

The following iterative process was used to find the correlation between the desired parameters, please refer to Fig. 4.9 for a derivation of the equations:

1. Set the rotational velocity and (re)estimate a corresponding axial inflow velocity, u_0, for the propeller.
2. Compute the thrust with QProp, based on these input values.
3. Calculate the corresponding drag with Eq. (4.7) and (4.8).
4. Calculate the radial inflow velocity with Eq. (4.9).
5. Obtain the axial propwash velocity from QProp.
6. Input the results from step 4 and 5 in Solidworks, together with the assumed axial inflow velocity.
7. Compare the output drag from Solidworks with (four times) the drag calculated in step 3.
8. Reiterate with a higher inflow velocity if the drag from Solidworks is lower and vice versa.

$$\theta = \cos^{-1}\left(\frac{T_{hover}}{T}\right) \tag{4.7}$$

$$D = T\sin(\theta) \tag{4.8}$$

$$u_r = \frac{u_0}{\tan(\theta)} \tag{4.9}$$

Based on the discussed iterations, two main diagrams can be drawn, relating the most important flight parameters. The diagrams can be found in Fig. 4.10 and 4.11, thrust can also be easily derived from these diagrams. Note that it is not claimed that flight speeds up to 30 m/s can actually be attained.

The flight speed for maximum efficiency in terms of range can be found by taking the tangent line to the power-vs-flight speed graph from the point (0,0). By dividing the battery capacity of 120 Wh (Chapter 9.1) by the required power at the point of intersection, a flight time of 0.625 hours can be found. At a flight speed of 20 m/s, this results in a range of 45 km. In reality this number will be lower, since the given battery capacity will be lower at a higher discharge rate, as discussed in Chapter 9.1.

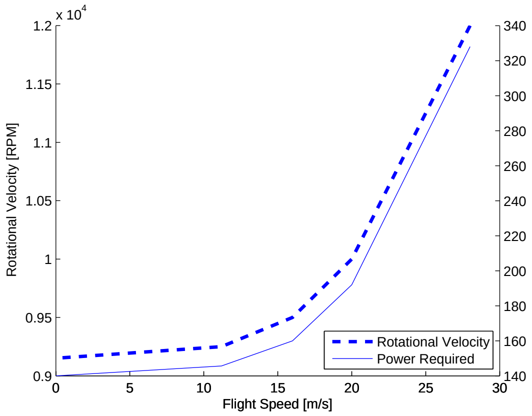


Figure 4.10: Rotor rotational speed and motor power consumption of the quadcopter as a function of forward flight speed

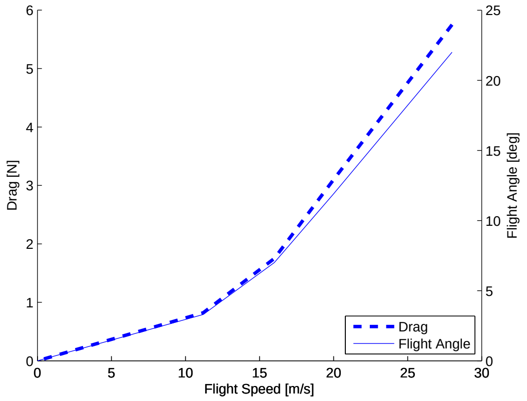


Figure 4.11: Aerodynamic drag and flight angle of the quadcopter as a function of forward flight speed

Another interesting parameter that can be derived from the curve is the drag coefficient of the vehicle, which can be calculated with Eq. (4.10). In this equation, the drag can be represented as a function of airspeed, see Eq. (4.11). This relation was derived from Fig. 4.11. The reference surface, S, is proposed to be calculated with Eq. (4.12). Here, r_{max} is the maximum dimension of the UAV, being 0.61 m. Filling in the equations gives a C_D value of 0.011.

$$C_D = \frac{D}{0.5\rho V^2 S} \tag{4.10}$$

$$D = 0.0076V^2 \tag{4.11}$$

$$S = \pi r_{max}^2 \tag{4.12}$$

A similar method as discussed above can be used to generate curves relating to vertical flight speed (**Req-flight-2** and **Req-flight-4**). The graphs can be seen in Fig. 4.12 and 4.13.

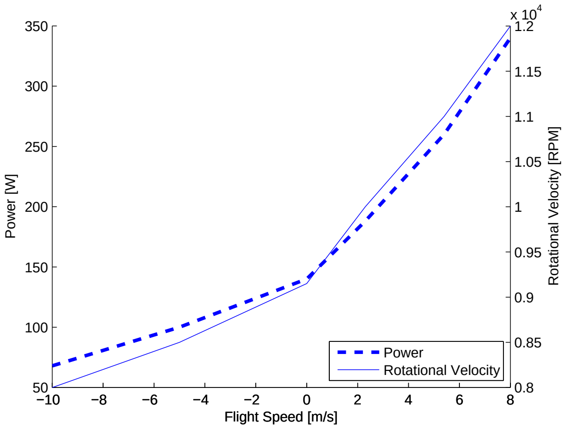


Figure 4.12: Rotor rotational speed and motor power consumption of the quadcopter as a function of vertical flight speed

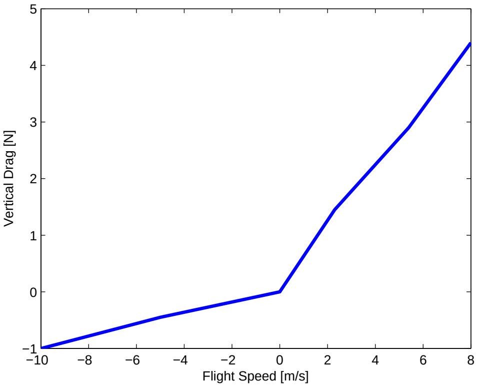


Figure 4.13: Aerodynamic drag of the quadcopter as a function of vertical flight speed

4.6. Discussion and recommendations

Throughout this chapter the propulsion system has been designed and analysed. However, a few recommendations for future work can be made to further extend this research:

- An optimal airfoil has been selected based on selection of representative propeller airfoils. However, literally thousands of airfoils are around, from which a few might still better suit the design. It is also possible to create a custom-made airfoil. Future research might look into this to just give this tiny increase in performance gain.
- The structural department did not find time to analyse the material selection of the propellers. Therefore, their masses in Section 11.5 have been estimated using data from 5.4 in Chapter 5. In the future it is important to analyse the loads on the propellers and select a material that can cope with these.
- In terms of performance, backward and sideways flight can be analysed in the same manner as forward flight. However, due to the extreme computational time involved with the calculations, this has not been done at this stage. Analysis can also be performed on inclined climb or descending flight paths.
- Also, although the performance proved to be quite good already, the structure might be optimised for aerodynamics. This allows for greater flight speeds, but also more efficient flight at lower flight speeds. So far, the structure has mostly been optimised for load-carrying requirements, since these are dominant.
- A third elaboration on performance analysis can be made in terms of angular accelerations of the vehicle. At this stage, estimates were very hard to make due to the combined influence of propeller inertia, structure inertia and motor acceleration rates. Especially the last parameter can only be obtained from experimental data.
- A whole range of parameters and characteristics can be analysed in depth by performing static, windtunnel or even full-scale tests. Experimental verification should eventually be part of every design.

5 Structure, materials & manufacturing

In this chapter the structural design of the quadcopter will be explained. First the general layout is designed. After this the calculation methods used for designing will be explained, followed by the final structural design. In the end information will be given on the manufacturing of the quadcopter.

5.1. Structural design choices

In this section the choices made to come to a final structural design will be explained. First the overall layout will be decided upon, followed by the different parts of the structure. Finally the materials to be used will be decided on.

5.1.1. General layout

The general layout of the structural design follows from the preliminary design made for the Mid-Term Report. This design was based upon the X-CSM quadcopter frame [38], a quadcopter design optimised to be as lightweight as possible. For the final design it has been decided to follow the general shape of the X-CSM frame, with a central structure and four beams supporting the engines. Looking from the top, the structure is fully symmetrical, which is beneficial in terms of controllability.

With the general shape down, there are still many options available on how to achieve this shape. These options can be divided into two separate categories. The first category is a design in which the centre structure and the beams are made out of one piece. The second category has the beams and centre structure separable, with for example for separate beams attached to a central structure. A trade-off needs to be made to be able to decide between these two options. The final summary of this trade-off is given in Table 5.1, with all considerations given below. The maximum score each design could get per criteria was three, with the lowest being one.

The first and most important criteria is repairability, how easy is it to repair the structure in case of damage to it. This criteria is given the highest weight because of usability and sustainability reasons. When deploying the quadcopter in an actual emergency situation it is important to be able to use the quadcopter as much as possible, as a quick response could save lives. Also in terms of sustainability it is important to have a structure that can be repaired with the least amount of waste. When the structure can be repaired instead of being replaced resources are saved. The design consisting of one piece scores low on this criteria. That is because when for example one beam is snapped off due to a crash, the whole structure needs to be replaced. This costs a lot of resources but especially a lot of time as all the subsystems have to be transferred to the new body. If the same thing happens with a design having separable beams the broken beam is taken off, a new one is attached to the centre plate and the engine with the electronic speed controller (ESC) is reattached. This costs a lot less time and resources.

The second criteria is weight, which is deemed a bit less important than the repairability, as it is assumed that the differences between the different designs are not as high. It is however more important than the transportability as weight is one of the main design drivers. In this case the design consisting of one piece is better, as no extra fastening materials are needed. This is the case for the other design, where the different parts need to be attached to each other.

The last criteria is transportability, how easy is it to transport the quadcopter to the location where the mission needs to be performed. In the case of the separable pieces design the quadcopter can be easily taken apart so it can be transported easily when a large enough bag is unavailable. However as it is assumed that a large enough bag is available this criteria is given a low weight. Also because the separable design still has to be taken apart it is not given the full score of three.

Summing everything up in Table 5.1 the separable design wins. From this it was decided to use a design consisting of separate parts.

Table 5.1: Summary trade-off structural layout

Criteria	Weight	Score one piece	Score seperate pieces
Repairability	6	1	3
Weight	4	3	1
Transportability	2	1	2
Total		20	26

5.1.2. Beams

With the decision on the general overview of the design fixed the subcomponents can be designed. The beams to which the engines are attached will be designed first. A requirement on the beams is that they should be separable from the centre structure. From market research two main options were found on how to make the structure of the beams. The first one uses tubes and the second one sandwich panels. Both designs will be analysed after which a trade-off will be made.

First the design using tubes will be analysed. This design is based on the frames as made by Syndrones [39] and Steadidrone [40]. The main load carrying structure in this case is a tube. The tube is attached to the engine on one end and to the centre plates on the other end. The tubes are attached to the centre plate using clamps. These clamps are two rectangular shapes, with half a circle taken out. The tube can be placed in between two of these structures, after which they are clamped together. This will then keep the tube in place. The engines are attached to the tubes in a similar manner. Per tube four clamps are needed, two for the centre plate and two for the engine placement. Using two clamps on each end will reduce the bending loads on each clamp. The centre structure will in this case consist of two plates, one above and one below the clamps.

The second design is based upon the X-CSM [38], which has sandwich panels since beams. Also many of the drones as designed by AscTec use sandwich panels [41]. The engines can be attached to the sandwich beams easily as they are flat on the top and bottom, meaning that the engines can be bolted to the beam. The beams can also be connected to the centre structure using bolts. The sandwich panel ends will be shaped in such a way that they make up for a quarter of the centre structure shape. That way they can easily be bolted to two thin plates on the top and bottom, allowing for a stiff sandwich like centre structure.

A trade-off between the two designs will be made on the based on weight, the ease of manufacturing and the ease of adding attachments to the beam. A summary of the trade-off can be found in Table 5.2.

The highest weight is given to the weight of the beams, as the beams are a large part of the structure so they will influence the total weight of the quadcopter a lot. The sandwich panel like structure scores the highest on this criteria. That is because not much extra material is needed to attach the panel to the centre plate, or the engines to the panel. For the design using tubes this is not the case, as extra clamps are needed to attach the beams. Also because of the clamps the tubes cannot have a too small a radius, as enough surface is needed for the clamping. This will result in extra weight as compared to the sandwich panel.

The next criteria is the ease of manufacturing the beams and attaching them to the centre plate. In this case it is assumed that the tubes and panels can be bought in the correct dimensions. The clamps to attach the tubes to the centre plate will probably have to be manufactured in house. They can be cast, which is very easy after the mould has been made the first time. For the sandwich panels only holes need to be drilled for screwing, or they have to be glued. This means that the sandwich panels are a bit easier to manufacture when compared to the tubes.

The last criteria is the ease of attaching anything to the beam, such as engines and feet. For the sandwich panels this is quite easy, as anything can be bolted onto it. For the tube structure it is also not hard to attach extra parts. That is because when the clamps have been manufactured they can be used to add everything to the beams. For both methods however additional actions have to be undertaken to add attachments to the beam. For this reason both designs do not get the highest score.

From the trade-off it is found that the sandwich panels are the best choice for the beam design of the quadcopter. This can also be seen in Table 5.2.

Table 5.2: Summary trade-off beam design

Criteria	Weight	Tubes	Sandwich
Weight	6	1	3
Ease of manufacturing	4	3	2
Ease of attachments	2	2	2
Total		22	30

5.1.3. Centre structure

With the beams designed the centre structure can be designed. As already mentioned when designing the overall layout in Section 5.1.1 symmetry is of importance. This also counts for the centre structure. From this it follows that two shapes are available, a square and a circle. Other shapes, such as a hexagon are also possible, but these can be regarded as a less round circle, being in between the square and the circle. For the centre it is important that the surface is maximised for a maximum dimension. For a given maximum dimension, the square has the highest surface area. Therefore it has been decided to go with the square centre structure. To allow for proper stiffness a plate will be added to the top and bottom of the core structure, made up from the sandwich panels as explained in Section 5.1.2.

To accommodate for the batteries one extra plate will be added above the centre plate. These plates will have the same shape as the bottom plate and will be suspended on spacers. The amount of extra plates, or payload bays, depends on the size and amount of payload that needs to be added.

5.1.4. Feet

The final structural parts that need to be designed are the feet. The feet will be located under the beams, as there is enough space there. In that way there is also place left for the camera underneath the central structure. As the quadcopter has four beams, it is decided to go with four feet as well.

For the feet three different designs were considered. The first design uses tube like structures, attached to the beams using small plates mounted to the tubes. The second and third design use two plates, bolted underneath the beam and bended down. The difference between the second and third design however is that in the second design the beams run straight down and for the third design they are bolted together at the bottom, forming a V-shape.

These three different designs are compared on four criteria. These criteria are grip on the ground, weight, ease of attachment to the beam and stiffness. Again a summary can be found in Table 5.3.

First the grip on the ground is compared. The V-shaped beam scores highest on this criteria, because of its slightly more pointy feet. This will give a higher pressure on the ground, and it allows the feed to find grip quicker on a rough surface. On rocks for example, the feet can wiggle between the rocks. This is not possible with the tubes or straight down beams. Both the straight beams and the tubes however are not bad at providing grip and as such score the same.

The next criteria is the weight of the feet. The tubes score the worst in this aspect, as they require a lot more material to form the whole tube. This is also due to the fact that they have to be attached to small plates, which then connect to the beams. Because of this the tube radius has to be quite large, adding weight. The other two designs use the same amount of material and only the configuration is different. Because of this they both score the same. They are also expected to be a lot lighter than the tube structure.

The second to last criteria is the ease of attachment of the feet to the beams. As mentioned before the tubes need extra material to be attached to the beams, making it harder to attach the feet. This is also because more fastening materials are needed. The other two designs can be bolted underneath the beams, bending the plates down. Bending can however be a problem, depending on the materials used. So because of this both designs do not get the highest score.

The final criteria is stiffness. Again the tube structure scores the lowest. Although the tubes are probably the stiffest, the stiffness is decreased because of the extra small plates that need to be used to attach the tubes to the beams. The straight beams score the highest in this case, as their moment of inertia is constant and always as high as the highest from the V-shaped beam. That is because the Steiner terms do not decrease, as is the case with the V-shaped beam.

Following from this trade-off, summarised in Table 5.3, the V-shaped beams are the best choice. This is however with a very small margin. Looking at the market, such as the Syndrone frames [39], it can be found that a lot of drones use V-shaped feet. Due to this it was chosen to go with the outcome of the trade-off.

Table 5.3: Summary trade-off feet design

Criterea	Weight	Tubes	V shaped	Straight down
Grip	4	2	3	2
Weight	3	1	3	3
Ease of attachment	2	1	2	2
Stiffness	1	1	2	3
Total		14	27	24

5.1.5. Material selection

As mentioned before, weight is of great importance in the design of the quadcopter. Therefore this is also of great importance for the material selection. The structure however also needs to be stiff. The optimal material would therefore have a low density and a high Youngs modulus. Because of this it was chosen to go with carbon fibre reinforced plastic as the main material, because of its low density and high stiffness.

The sandwich core will be made of ROHACELL 31 IG, a very lightweight foam. The core does not have to carry a lot of loads, but is mainly meant for spacing the outer panels. Therefore it was decided to go with a very light material, without a very high strength. In Table 5.4 a summary of the material properties can be found.

Table 5.4: Material properties [8] [9]

Material	Density [kg/m^3]	Youngs modulus [Pa]
CFRP	1600	$70 \cdot 10^9$
ROHACELL 31 IG	32	$70 \cdot 10^6$

5.1.6. Fastening methods

The last design choice that needs to be made is on the fastening method, how are the different structural components going to be attached to each other. In this two options have been defined, bolting and the use of adhesives. Both of these methods will be graded on two criteria, repairability and weight.

The first criteria looked at is repairability. This is where the bolts easily win, as they can be unscrewed to remove a broken piece of structure. When everything is stuck together using adhesives this is not the case and a lot of time will be needed to remove the broken structure and the leftover adhesives. Also a lot more time is needed to clean the surface and let the adhesives dry.

When looking at weight on the other hand, the adhesives easily win as not much extra material is added. Weight however is given a lower weight than repairability as it influences the usage of the quadcopter a lot less. Also because a separable structure had been chosen for its repairability it was deemed that a repairable fastening method would be important in that case.

As repairability is considered more important than the weight, the bolts have been chosen. To save weight nylon bolts will be used. Probably these will also break before the rest of the structure upon impact, which might save some of the larger more expensive parts as they are separated from the structure.

5.2. Calculation methods

In this section the calculation methods used for designing the quadcopter will be explained. First the calculation methods for for bending will be shown, followed by calculations for the centre plate. After this the calculation methods for the impact loading will be shown. The chapter is ended with the mass moment of inertia calculation.

5.2.1. Bending

In this section the calculation methods used for beams in bending will be explained. A large part of the loads on the quadcopter are bending loads, for example the load of the engines on the end of the beams. In this section it will be explained how the bending stresses and deflections are determined. First the general formulas will be explained, after which the bending calculations will be shown for the beams and the feet.

Bending in general

When looking at the bending of a beam two quantities are the most important, the deflection of the beam and the maximum stress in the beam. When designing a small vehicle, such as the quadcopter under consideration, the loads are not as high as when for example designing a full scale aircraft. For that reason it has been decided to set a maximum deflection when designing the beams, rather than reaching the yield stress at ultimate load for example.

The maximum deflection of a beam loaded at the tip can be calculated using Eq. (5.1) [42]. In this equation δ_{max} is the maximum deflection, occurring at the tip. The force at the tip is F and L is the length of the beam. The moment of inertia around the bending axis is I and E is the Young's modulus of the material used.

$$\delta_{\text{max}} = \frac{FL^3}{3EI} \tag{5.1}$$

$$I = \frac{FL^3}{3E\delta} \tag{5.2}$$

As said before, the beams will be designed for deflection. Equation (5.1) can therefore be rewritten into Eq. (5.2) to compute the required moment of inertia to achieve this maximum deflection when a length and material have been chosen for the beam.

With the moment of inertia known the stress in the beam can be calculated. This is done using the flexure formula, Eq. (5.3). In this formula M is the internal moment in the beam at the part under consideration, y the distance from the neutral axis and I the moment of inertia. As the maximum stress is the most interesting the Eq. (5.3) can be rewritten to Eq. (5.4). Here c is the maximum distance from the neutral axis. Equation (5.4) assumes that there is only a tip load on the beam.

$$\sigma = \frac{My}{I} \tag{5.3}$$

$$\sigma_{\text{max}} = \frac{FLc}{I} \tag{5.4}$$

Area Moment of inertia

In the previous section the moment of inertia was calculated as a requirement from the maximum allowed deflection. In this section the general formulas for calculating the area moment of inertia will be explained. These will then be adapted for use in the design of the beams and feet in the following sections.

In the design of the quadcopter only rectangular shapes that have a plane of symmetry are used. Because of this only two general formulas to calculate the moment of inertia are needed, Eq. (5.5) and (5.6) [42]. Here h is the height of the rectangle, w the width and r the distance of the centroid of the rectangle to the axis around which the moment of inertia is calculated. This axis is the neutral axis of the beam when calculating the moment of inertia for bending.

$$I_{xx} = \frac{1}{12}wh^3 + whr^2 \tag{5.5}$$

$$I_{yy} = \frac{1}{12}w^3h + whr^2 \tag{5.6}$$

Adaptions for beam calculations

The beams on the quadcopter will be made from sandwich panels, which implies that they will be made from different materials. Equations (5.1) and (5.3) however are only valid for beams made out of one material. To apply these equations to the sandwich panels used, one of the materials has to be transformed into the other [42]. This is done by comparing the Young's moduli of the core and panel material. In this case it has been decided to transform the core material into the plate material. First the ratio between the two Young's moduli is calculated using Eq. (5.7). Next the width of the core is multiplied with this ratio n, Eq. (5.8). Upon doing this the sandwich panel is transformed into an I-beam of equivalent strength fully made from the plate material. The reference frame used for the calculations is defined as follows: the origin is located at the centroid of the tip of the beam, the x-axis is pointing upward, the y axis is following the length of the beam and finally the z-axis is completed by following the right hand rule.

$$n = \frac{E_{\text{core}}}{E_{\text{plate}}} \tag{5.7}$$

$$w_{\text{core,transformed}} = n w_{\text{core}} \tag{5.8}$$

Using this transformed beam the stresses can be calculated. For the core material however, the stresses have to be transformed back to the actual core. This is done by multiplying the stress in the core as calculated using Eq. (5.3) with the ratio n.

Adaptions for feet calculations

Equations (5.1) and (5.3) are not only restricted to beams made out of one material, but also with constant cross-section. With the chosen feet design, the cross-section is not constant. Because of this a different calculation method had to be found. Different methods were tried to come up with a solution. In this section it will be explained how an approximation of the deflection is found.

First an axis system for the feet calculations will be defined. The y-axis is pointing towards the centre of the vehicle, the z-axis upwards and the x-axis completes the right handed axis system. The origin of the axis system is placed in the middle of the cross-section of the beam, all the way at the bottom.

The constantly varying cross-section is not a problem for calculating the moment of inertia around the x-axis. The two beams are of constant size and the location does not change with respect to the x-axis. The calculation method for the moment of inertia as given in Section 5.2.1. Because of this the deflection around the x-axis can also be calculated using the methods from Section 5.2.1.

For the deflection around the y-axis however adaptations are needed. That is because there is a continuously varying moment of inertia around that axis. The moment of inertia is a function of z, as can be seen in Eq. (5.9). In this formula t is the thickness of the feet, w the width of a feet and b the width of a beam. Because the moment of inertia is a function of z, the bending cannot be calculated using Eq. (5.1). Another option is to use Eq. (5.10), which also uses the moment as a function of z, Eq. (5.11), to calculate the deformation of the beam.

$$I_{yy}(z) = 2\left(\frac{1}{12}t^3 w + tw\frac{b^2}{4h^2}z^2\right) \tag{5.9}$$

$$E\frac{d^2 v}{dz^2} = \frac{M(z)}{I(z)} \tag{5.10}$$

$$M(z) = Fz \tag{5.11}$$

It was tried to solve Eq. (5.10) by integrating it twice, Eqs. (5.12) and (5.13), and using the boundary condition that at z equals h the equation must equal zero. This however did not result in a analytical solution. After this it was decided to divide the foot up in sections, which have a constant cross-section. Then the moment of inertia of this part was calculated, along with the moment acting on it. Then the stresses were calculated and using the Young's modulus of the material the strain and elongation were calculated. This was also done for the elongation when bending around the x-axis. These values were then compared to get an estimation of the deflection. If the orders of magnitude are the same, the deflection around the y axis is assumed to be about the same.

$$E\frac{dv}{dx} = \frac{2Fh^2\ln(z)}{b^2tw} + C_1 \tag{5.12}$$

$$Ev = \frac{2Fh^2z\ln(z)}{B^2tw} - \frac{2Fh^2z}{b^2tw} + C_1z + C_2 \tag{5.13}$$

5.2.2. Centre plate

The arm has a square on one side of the arm, on the other side the motor is attached. By placing 4 arms together, the squares form the centre plate. An additional thin sheet will be placed on top and below these for squares to provide attachment.

There will be screws going through the top and bottom sheets for each arm. The centre square is assumed to be clamped, this means that the moment will not further increase after the first screw. As the centre plate has the same properties as the arm and the moment is not increasing, it can then be assumed that the stress will also be the same. As long as the arm is not failing, the centre plate will certainly not fail.

The centre plate needs to be checked for deflection. The deflections are mainly caused due to the moment from the arms and the loads on the plate by the payload systems. The moments can be decomposed into components which coincides with the axis directions of the centre. As the moments will be the same on both sides of the plate, it can be seen as a clamped plate with the applied moment on one side and the reaction moment on the other side. The situation is sketched in Fig. 5.1.

The deflections are calculated by superimposing the deflections along the x and y axes. The deflections along those axes can be calculated by integrating the moment twice and determining the boundary conditions.

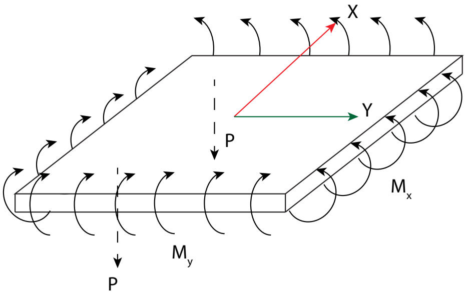


Figure 5.1: Forces and moments on the centre plate

There is a standard case for a clamped beam on which a moment is exerted as found in Appendix C of Mechanics of Materials [42]. The equation is stated in Eq. (5.14), where x is the distance to the clamped location, M the moment applied on the free end, E the Young's modulus and I the area moment of inertia of the cross-section. For the plate case the x ranges from -$\frac{\text{width}}{2}$ to $\frac{\text{width}}{2}$.

$$v = \frac{Mx^2}{2EI} \tag{5.14}$$

The point loads can be modelled on a beam that is supported on both ends. There is a standard case for this as well, but it only covers the first part of the beam to the location where the point load is applied. It might be possible to invert the beam and apply the same equation to the other part of the beam, but implementation wise it turned out not to give reliable outcomes. Thus another approach was taken, integrating the moment twice. In this way it could be made sure that there is no need to flip coordinates, however there will still be 2 equations, one for each part of the beam.

Integrating the moment to get the deflection of a point load

First of all a free body diagram is made, as seen in Fig. 5.2, secondly the moment as a function of x will be determined for both parts of the beams, thirdly the moment equations will be integrated and finally the coefficients of integration will be found using boundary conditions.

1) Free body diagram

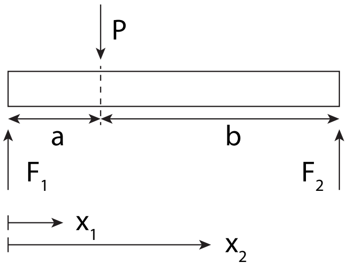


Figure 5.2: Free body diagram of a point load applied on the beam

2) Reaction forces and moments

$$F_1 = \frac{Pb}{a+b} \tag{5.15}$$

$$F_2 = \frac{Pa}{a+b} \tag{5.16}$$

$$M_1 = F_1 x_1 \qquad 0 \leq x \leq a \tag{5.17}$$

$$M_2 = P \cdot a - F_2 x_2 \qquad a \leq x \leq a+b \tag{5.18}$$

3) Integrate

$$EI\frac{d^2 v_1}{dx_1^2} = F_1 x_1 \qquad EI\frac{d^2 v_2}{dx_2^2} = Pa - F_2 x_2 \tag{5.19}$$

$$EI\frac{dv_1}{dx_1} = \frac{F_1}{2}x_1^2 + C_1 \qquad EI\frac{dv_2}{dx_2} = Pax_2 - \frac{F_2}{2}x_2^2 + C_3 \tag{5.20}$$

$$EIv_1 = \frac{F_1}{6}x_1^3 + C_1 x + C_2 \qquad EIv_2 = \frac{Pa}{2}x_2^2 - \frac{F_2}{6}x_2^3 + C_3 x + C_4 \tag{5.21}$$

4) **determine the coefficients**

At $x = 0$ the deflection v is 0, this means that C_2 is also 0, as can be observed from Eq. (5.21). The other coefficients are not 0 and correspond to the following boundary conditions: at $x = a + b$ the deflection v is 0, at $x_1 = x_2 = a$ the deflection angle of both parts should equal $\frac{dv_1}{dx_1} = \frac{dv_2}{dx_2}$ and at $x_1 = x_2 = a$ the deflections should equal $v_1 = v_2$. The coefficients C_1,C_3andC_4 can then be found by solving the following set of equations, written out in matrix format:

$$\begin{bmatrix} 1 & -1 & 0 \\ 0 & a+b & 1 \\ -a & a & 1 \end{bmatrix} \begin{bmatrix} C_1 \\ C_3 \\ C_4 \end{bmatrix} = \begin{bmatrix} Pa^2 - \frac{F_2}{2}a^2 - \frac{F_1}{2}a^2 \\ -\frac{Pa}{2}(a+b)^2 + \frac{F_2}{6}(a+b)^3 \\ \frac{F_1}{6}a^3 - \frac{Pa}{2}a^2 + \frac{F_2}{6}a^3 \end{bmatrix} \tag{5.22}$$

Now it is possible to place any amount of loads every where on the plate to calculate the corresponding deflection.

5.2.3. Impact

The MAV might be dropped accidentally, this should not lead to failure of the structure. It is assumed that the MAV is dropped from half a meter and that it lands with its full weight on one arm. The arm is modelled as a spring and the ground as a rigid body. All energy is assumed to be transformed into potential energy in the spring. This will lead to a very conservative result as in reality the MAV will have lost a lot of energy already due to drag, spinning and producing sound. This method is taken from Mechanics of Materials [42].

The method deals with a falling rigid block hitting a structure with stiffness k, as shown in Fig. 5.3. The block with mass m is positioned a distance h above the structure modelled as a spring with stiffness k. The block will fall down a distance $h + \Delta_{max}$.

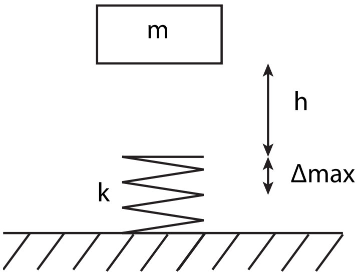


Figure 5.3: Mass falling on a spring with assigned parameters

An expression can be found for Δ_{max} by equating the potential energy U_e and the internal energy U_i of the spring:

$$U_e = U_i$$

$$mg(h + \Delta_{max}) = \frac{1}{2}k\Delta_{max}^2$$

$$\Delta_{max}^2 - 2\frac{mg}{k}\Delta_{max} - 2\frac{mg}{k}h = 0$$

The determinant is:

$$4\left(\frac{mg}{k}\right)^2 + 4 \cdot 2 \cdot \frac{mg}{k}h$$

Then the maximum contraction will be:

$$\Delta_{max} = \frac{mg}{k} + \sqrt{\left(\frac{mg}{k}\right)^2 + 2\frac{mg}{k}h} \tag{5.23}$$

Where only the positive part of the root has been taken. The force on the structure can then be calculated by multiplying this contraction with the spring constant. Finally the stress can be calculated by dividing the force by the cross-sectional area.

The arm of the MAV is not a spring, thus it needs to be modelled as a spring. The cross section of the arm is shown in Fig. 5.4.

The force needed to give a beam a displacement in axial direction is:

$$F = \frac{EA}{L}\delta$$

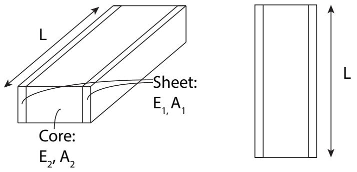


Figure 5.4: Beam with assigned parameters

Where F is the force applied, E the Young's modulus, A the cross-sectional area, L the length of the beam and δ the displacement. Thus $\frac{EA}{L}$ can be seen as the spring constant. As the arm of the MAV is composed of 2 materials the resultant spring constant will be :

$$k = \frac{E_1 A_1}{L} + \frac{E_2 A_2}{L}$$

5.2.4. Mass moment of inertia

The mass moment of inertia (MMOI) of an object depends on the size and mass. The MMOI is particularly important for manoeuvrability, as the angular acceleration is directly related to it. In this section the approach to calculate the MMOI of the MAV will be discussed. The approach consist of splitting the structure into simple shapes such as rectangular bars or cylinders. The MMOI of each separate element will then be transformed into the body axis system. The body axis convention is shown in Fig. 5.5. Then the MMOI will be corrected to be the MMOI around the centre of gravity.

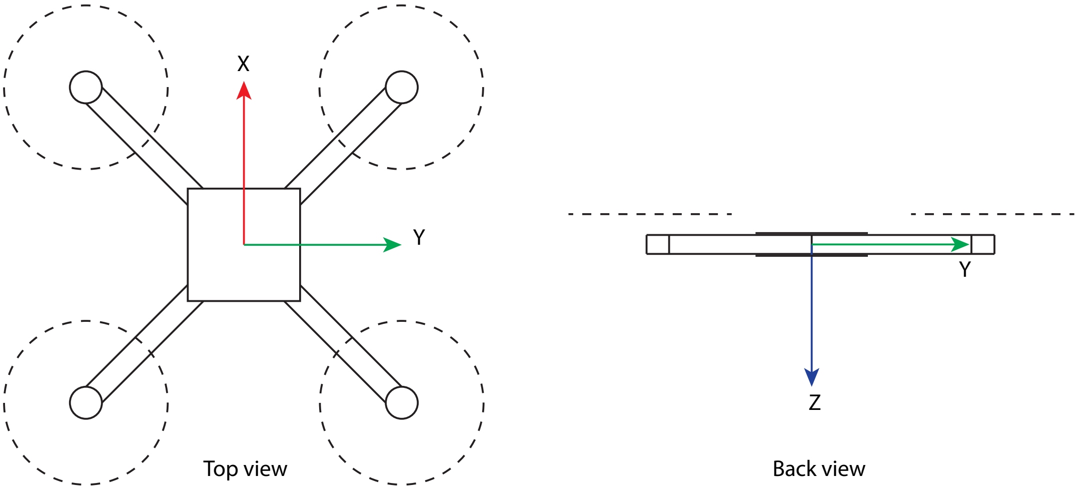


Figure 5.5: Reference frame for the MAV

The MMOI of most of the elements such as the centre plates and the motors can be found using standard cases and the parallel-axis theorem, as discussed in Engineering Mechanics Dynamics fifth edition in SI units by A. Bedford and W. Fowler [43]. A summary of the parallel-axis theorem is given in Eq. (5.24). Where m is the mass in kg, d the distance in meters and the ' in the subscript stands for the reference frame having its own centre of gravity as origin.

$$\begin{aligned} I_{xx} &= I_{x'x'} + (d_y^2 + d_z^2)m \quad & I_{xy} &= I_{x'y'} + d_x d_y m \\ I_{yy} &= I_{y'y'} + (d_x^2 + d_z^2)m \quad & I_{yz} &= I_{y'z'} + d_y d_z m \\ I_{zz} &= I_{z'z'} + (d_x^2 + d_y^2)m \quad & I_{zx} &= I_{z'x'} + d_z d_x m \end{aligned} \tag{5.24}$$

The arms which connects the motor with the centre structure does not have its axes aligned with the body reference system. Thus, an arm will be used to demonstrate the process of calculating the MMOI.

The lower right arm has been taken, as depicted in Fig. 5.6. As can be seen the arm consist of a square part which is exactly one fourth of the area of the centre, a panel and a cylindrical platform with the same thickness as the panel. For the ease of calculations the parts that are covered twice are not removed and parts that are not covered at all will not be added. For example, the square centre panel and the long panel are assumed to touch and not to collide into each other. The variables used to name the dimensions per element are shown in Fig. 5.7.

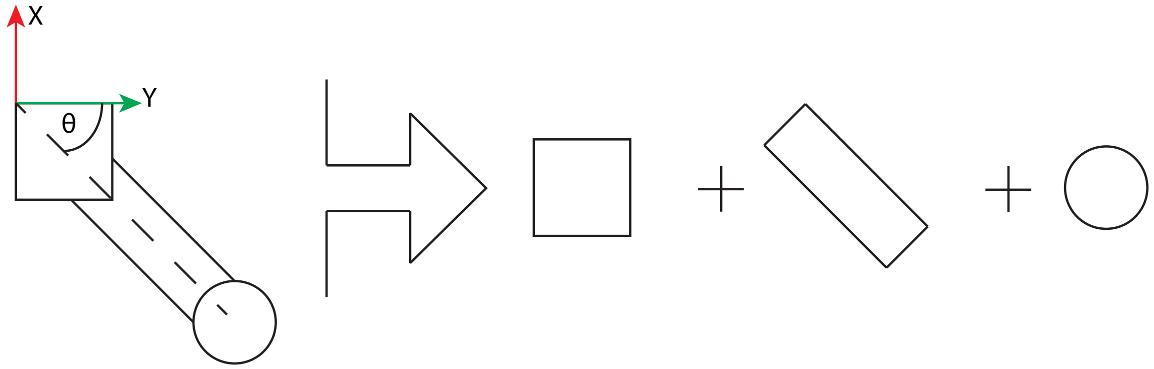


Figure 5.6: Arm of the MAV

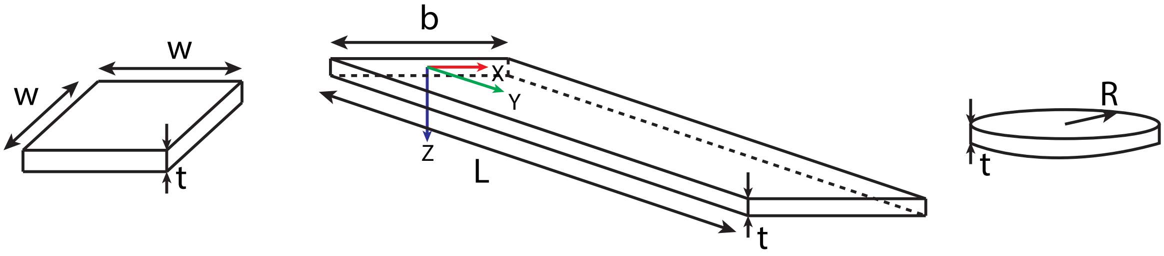


Figure 5.7: Arm of the MAV including the dimensional parameters

The square block structure can be seen as a rectangular prism for which a standard case around its centre of gravity is available. By applying the Eq. (5.24), the MMOI become can be calculated around the body axes. The equations are shown in Eq. (5.25).

$$\begin{aligned} I_{xx} &= \frac{1}{12}m\left(t^2+w^2\right)+m\left(\frac{1}{4}w^2\right)=\frac{1}{12}mt^2+\frac{1}{4}mw^2 & I_{xy} &= \frac{1}{4}mw^2 \\ I_{yy} &= \frac{1}{12}m\left(t^2+w^2\right)+m\left(\frac{1}{4}w^2\right)=\frac{1}{12}mt^2+\frac{1}{4}mw^2 & I_{yz} &= 0 \\ I_{zz} &= \frac{1}{12}m\left(w^2+w^2\right)+m\left(\frac{1}{4}w^2+\frac{1}{4}w^2\right)=\frac{2}{3}mw^2 & I_{zx} &= 0 \end{aligned} \tag{5.25}$$

The rectangular plate will first be analysed in its own reference frame and then transformed to the body frame. The distance to move is not just $0.5\cdot L$, but there needs to be an additional length added to it in order to make the origins coincide. The distance is estimated to be $\sqrt{2}\cdot w$. Using the same method, the MMOI's were found to be:

$$\begin{aligned} I_{xx} &= \frac{1}{12}mt^2+\frac{1}{3}mL^2+2\cdot mw^2 & I_{xy} &= 0 \\ I_{yy} &= \frac{1}{12}m\left(t^2+b^2\right) & I_{yz} &= 0 \\ I_{zz} &= \frac{1}{12}mb^2+\frac{1}{3}mL^2+2\cdot mw^2 & I_{zx} &= 0 \end{aligned} \tag{5.26}$$

From Engineering mechanics dynamics [43, p.550], it can be seen that the MMOI around an arbitrary axis described by the unit vector *e* can be calculated as given in Eq. (5.27).

$$I_e = I_{xx}e_x^2+I_{yy}e_y^2+I_{zz}e_z^2-2I_{xy}e_xe_y-2I_{yz}e_ye_x-2I_{zx}e_ze_x \tag{5.27}$$

For the rectangular plate problem the unit vectors will correspond to the x and y axis of the body axis as seen from the local frame, the z axis is already pointing in the same direction.

In a similar fashion the MMOI for the cylinder can be determined. In this case the distance to be added to let the x and y axes coincide are $L\sin\theta + w$ and $L\cos\theta + w$ respectively. The final result is given in Eq. (5.28). The same procedure is followed for the other subsystems.

$$
\begin{aligned}
I_{xx} &= \frac{1}{3}mt^2 + \frac{1}{4}mR^2 + 2\cdot mw^2 + mL^2\cos^2\theta & I_{xy} &= m\left((L\sin\theta + w)(L\cos\theta + w)\right) \\
I_{yy} &= \frac{1}{3}mt^2 + \frac{1}{4}mR^2 + 2\cdot mw^2 + mL^2\sin^2\theta & I_{yz} &= 0 \\
I_{zz} &= \frac{1}{2}mR^2 + mL^2 + 2\cdot mw^2 & I_{zx} &= 0
\end{aligned}
\tag{5.28}
$$

As the mass and the location of each shape is known, the centre of gravity (cg) of the whole structure can be calculated using:

$$
x_{cg} = \frac{\sum x\cdot m}{\sum m} \qquad y_{cg} = \frac{\sum y\cdot m}{\sum m} \qquad z_{cg} = \frac{\sum z\cdot m}{\sum m}
\tag{5.29}
$$

Then the MMOI around the centre of gravity will be:

$$
\begin{aligned}
I_{xx,cg} &= I_{xx} - (y_{cg}^2 + z_{cg}^2)m \\
I_{yy,cg} &= I_{yy} - (x_{cg}^2 + z_{cg}^2)m \\
I_{zz,cg} &= I_{zz} - (x_{cg}^2 + y_{cg}^2)m
\end{aligned}
\tag{5.30}
$$

5.3. Final design

In this section the final design of the structure of the quadcopter will be shown. All the structural sizes will be given, as well as the forces and deflections used to come to the final design.

5.3.1. Beams

The first structural part that was designed are the beams. They are sized for a maximum deflection of 1 mm at a tip load of 50 N. This amounts to a load of two times its own weight. The length was first fixed at 12 cm, to allow for enough clearance between the propeller and the centre structure. A plate thickness of 0.5 mm was set as a first estimate. Using MATLAB an optimum design was calculated, minimising the weight of the beams, by setting the width and core thickness.

This design was then tested for impact, dropping the whole quadcopter straight down on one beam from 0.5 m. With the beam sized for minimum weight the stresses in the beam became too high, resulting in a failed beam upon impact. The beam was then manually resized for impact, after which the deflection was calculated. This resulted in the beam as shown in Table 5.5, with a 1 cm core thickness made out of ROHACELL and a 0.5 mm plate thickness, made from carbon fibre reinforced plastic. The design was then again tested on yielding, but the stresses developed were not large enough to yield the material. The deflections were also under the minimum set at the beginning. The mass reported includes only the mass of the beam sticking out, the rest of the mass is included with the centre plate.

Table 5.5: Final beam design

Lenght	12 cm
Width	3 cm
Plate thickness	0.5 mm
Core thickness	1 cm
δ_x	0.55 mm
δ_y	0.13 mm
Mass 1 beam	6.9 g

5.3.2. Centre plate

The centre square is designed to have sides of 12 cm long, allowing for enough subsystem space. The centre plates were given a thickness of 0.25 mm, the thinnest sheet thickness that can be bought. These plates are bolted onto the middle parts of the beam, as described in Section 5.1.2.

5.3.3. Feet

With the beams and centre plate designed, the last major structural part to be discussed are the feet. The feet are 15 cm high, allowing for enough ground clearance for the electronics box and camera that will be placed

Table 5.6: Final centre plate design

Width	12 cm
Plate thickness	0.25 mm
Δ	0.085 mm
Mass	39.2 g

Table 5.7: Final feet design

Height	15 cm
Width	1.5 cm
Plate thickness	1 mm
Δ	0.24 mm
Mass	4.3 g

underneath the quadcopter. The feet were then given a thickness of 1 mm and a width of 1.5 cm. A load of 100 N was then placed on the end of the beam in all three directions, checking for deflections and maximum stresses. A higher load was taken than for the beams to allow for a rough landing. The maximum stresses were far under the yield stress of the material, proving the beam design.

5.3.4. Payload bay

Only one payload bay is needed to support the LIDAR system on top of the centre structure and to cover the batteries placed underneath it. As the LIDAR system is not heavy it was chosen to go with a 0.25 mm thin sheet, supported over the batteries by spacers. This gives a payload bay mass of 5.8 g.

5.3.5. Bolts

As the width of the beams and feet is not very high, it is not possible to use very large bolts. However, using very small bolts might result in a bolt breaking long before the structure, causing unnecessary structural failure. Due to this it was decided to go with M4 bolts, having a diameter of 4 mm. A total of 35 bolts and nuts have been taken into account for the weight. Six are needed to attach a beam and its feet to the centre plate, another four are needed for the payload bay and another four are needed to attach the electronics box. It was chosen to take a few screws to many, to take into account any longer screws that might be necessary.

5.3.6. Total

All of this results in a total structural mass of 136 grams and a diagonal, from engine axis to engine axis, of 41 cm. Comparing this design to designs already existing it is a very reasonable design. The X-CSM [38] has a diagonal of 32 cm and a mass of 75 g. This is quite a bit lighter for the size difference, however the X-CSM does not have large feet such as the design made. This is an explanation for the large weight difference. Comparing the design to the Asctec Hummingbird [44], with a empty weight of 350 g and a engine to engine diagonal distance of 76 cm, the design is on the light side. However, it is not fully known what Asctec places under empty weight, this might already include some of the on-board systems such as engines. However even then the size and weight of the design are quite reasonable when compared to quadcopter structures that are already flying.

The last thing that needs to be checked is the compliance with the requirements set for the structure. The requirements can be found in Appendix A. Requirement **Req-struc-1** is complied with, as the system is designed for these loads. **Req-struc-2** is not fully complied with as the structure has been designed with sustainability and repairability in mind. **Req-struc-3** is not complied with as the structure is a bit heavier. This requirement was set with the X-CSM in mind, which does not have feet and is a bit smaller. This explains the difference. **Req-struc-4** is passed easily, calculations on the beams and feet have shown that the stress in the parts is at respectively 10 and 30 percent of the maximum stress, so no sign of failure. **Req-struc-5** is also passed as the beams are designed with this requirement in mind. The last two requirements, **Req-struc-6 and 7** are also passed, with more information given in Chapter 6.

A picture of the final layout of the design is shown in Fig. 5.8. The batteries are placed on top of the centre structure, with the LIDAR system on top of the payload bay. Underneath is the Vibration Damped Electronics and Camera Structure (VDECS) attached to it. The electronic speed controllers are located on the beams, near the centre plate, that engines are located on the very end of the beam. The antenna's are located inside two of the feet.

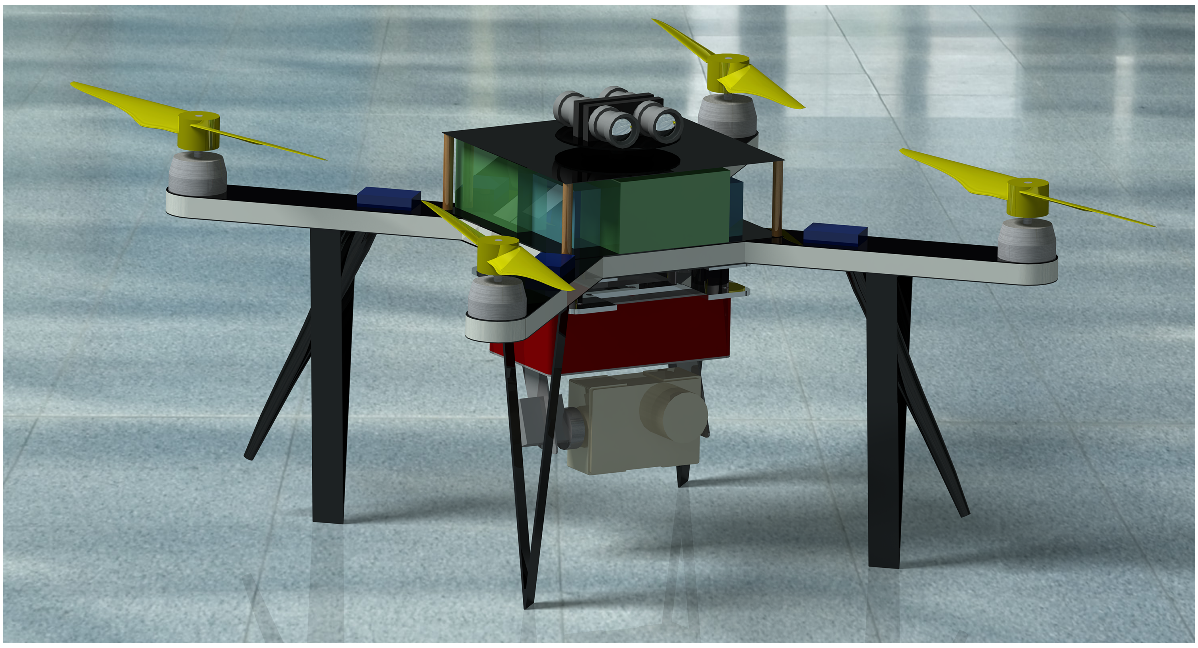


Figure 5.8: Render of the final layout of the quadcopter

5.4. Manufacturing

In this section the manufacturing process of the components of the frame will be explained. After that the integration of the components will be discussed. The framework can be divided up in the following components: the arm which connects motor to the centre plate, the feet, the top and bottom sheets to attach the arms on in the centre and the plateau that will be placed on top of the centre plate.

First the materials used are listed:

- 0.25 mm carbon fibre sheet
- 0.5 mm carbon fibre sheet
- 1 mm carbon fibre sheet
- 1 cm ROHACELL plate
- nylon bolts/ threaded rods
- plastic spacers

The carbon fibre sheets were found at Easy Composites [45].

5.4.1. Processing the materials

Before going into detail, the ways to process the materials and the corresponding safety aspects need to be discussed. The carbon fibre sheets and the ROHACELL plates need to be cut to the right shape. The cutting tutorial by EasyComposites has been used as a guide [46].

The ROHACELL plate can be cut using a hot wire, the cutting method will provide a smooth cut. During cutting the only risk is the hot wire.

The carbon fibre sheets can be cut using a jigsaw with an abrasive blade or a metal cutting blade (which has small teeth). Smaller teeth will give better results as the splintering of the sheet will be less, so the abrasive blade would give the best results. As the sheet will shatter on the cutting edge, it is wise to leave some space between the actual sketched form and the location of the cut. Afterwards the rough edge can be sandpapered to have a smooth finishing. To make the sheet waterproof, the edge can be gone over with epoxy to seal it.

While processing the carbon fibre sheets lots of dust and shattered parts will come off the plate, also toxic gases may come off from the epoxy. This poses a potential risk for the health. The small particles can cause irritations on the hands, eyes and lungs. As no high speed machinery is used, the spread of the particles will be limited, so simple protective measures can be taken. Surgery gloves can be used to protect the hands, safety goggles can be used to protect the eyes and a mouth cap can be used to prevent small particles and the toxic gases entering the lungs. However, if there is some dust getting on the arms it is best to wash it down using cold water, warm water will open the pores which will worsen the irritation. It is important to keep the working place ventilated and remove the dust with a vacuum cleaner, or other means of suction, continuously.

5.4.2. Manufacturing the parts

The arms are made of a sandwich panel structure composed of a top and bottom carbon fibre sheet of 0.5 mm and a core of 1 mm ROHACELL. The method of first cutting the sheets and core separately has been chosen over assembling the panel first and then cutting out the shape. Wire cutting is good for the ROHACELL but not for the carbon fibre sheet, vice versa for sawing. Then the sheet is attached to the core using an epoxy resin. ROHACELL is very durable and can withstand nearly all commercial adhesives [47].

The feet are made by cutting a strip of 1 mm carbon fibre sheet and heating it locally to be able to bend the sheet into the V-shape. A little bit of spring back of the sheet is not really a problem, as the V-shape is held together with bolts.

The sheets used in the centre and the plateau will just be cut out from the 0.25 mm carbon fibre sheet.

5.4.3. Assembling individual parts

Assembling of the individual parts is done using the nylon bolts. The 4 arms can be put together to form a big square in the middle, then a sheet is put on top and below it. Each arm is attached to the sheets using 4 bolts. Then 4 spacers can be placed at the corners on top of the centre part; a threaded rod will be put through each of the spacers. This rod can replace the bolt to fasten the arm and since it passes through the centre plate, the VDECS can be fastened to it. The feet will be bolted onto midpoint of each arm

5.5. Recommendations

With the limited time available during the DSE it is not possible to analyse all the structural parts in full detail. Due to this several recommendations can be made for further analysis. The first analysis that needs to be performed is a vibrational analysis. Performing a vibrational analysis on the structure is not as straight forward as it may seem. Modelling the beams as clamped beams allows for a quick calculation of the eigenfrequencies. However, the beams are bolted to the centre plate, most likely allowing for a bit of extra motion, completely changing the eigenfrequencies. It is also unknown at which frequency the engines vibrate for example, which also needs a more in-depth analysis. The next part that needs more analysis are the bolts and the holes in the plates where the bolts are put trough. In these locations stress concentrations will occur. Although the maximum stress will not be reached with a stress concentration factor of 2, more analysis is needed to know the stresses exactly. It is therefore recommended to make a model with the exact properties and to perform a Finite Element Method analysis on it, to gain a more in-depth look into the stress distribution over the structure.

6
Sensors and processing unit

This chapter will discuss the sensor, CPU, imaging and LIDAR systems. First an overview of the sensors used on general MAVs will be given. In the design process, the primary sensors are chosen first. This is required to determine on specific processors. The sensor and the CPU selections will be explained in the second and third section. The next step is to perform the camera selection for both mapping and object detection; this is explained in the fourth chapter. After that the LIDAR system, required for indoor navigation and mapping, will be designed. Finally the Vibration Damped Electronics and Camera Structure (VDECS) will be designed and the data handling block diagram is shown.

6.1. Sensors

This section will explain in more detail about the sensor system on the MAV. A general overview is given on the types of sensors followed by the definition of the sensor system in the context of this design. After that the requirements on the sensor system are described. The section concludes by summarizing the final design decisions made regarding the sensor subsystem.

6.1.1. General overview

An autonomous MAV has a range of sensors required for attitude determination and autonomous actions. Some sensors are vital to flight operations of an UAV in any generic condition, regardless of its autonomy level. These are the sensors in the Inertial Measurement Unit (IMU), which estimate the acceleration, angular velocity and orientation [48]. This is done with accelerometers, gyroscopes and a magnetometer. When navigational and positional estimation capabilities are required, a Global Positioning System (GPS) receiver can be used [49]. More accurate and stable autonomous control can be achieved by adding specific sensors. The measurement data of multiple (different) sensors can then be combined and processed through a Kalman filter. This could yield an improved final result on for example the state of the MAV. Sensors can also be added to allow the UAV to accomplish a specific task. Examples of this are indoor navigation and object tracking. Due to the small size and low cost of most modern Micro-Electro-Mechanical System (MEMS) sensors, the three primary points to consider when during sensor selection are:

- **Accuracy**
  A more accurate sensor will need less filtering and fewer measurements in order to estimate the correct state of the UAV.
- **Update rate**
  A higher update rate means more data, meaning higher rate of attitude correction to attain better control and stability properties.
- **Interfacing options**
  A sensor needs to interface with the on-board computer efficiently so that its readings can be used.

After creating a preliminary design of the MAV in the conceptual phase, an idea of the sensor system required was obtained. The following list mentions the types of sensors that are considered for the UAV. The listed sensors have been divided into two categories, one category listing the vital sensors for the IMAV 2014 mission and another category listing the auxiliary sensors that are used on the system to improve the quality of the state estimates and provide other types of functionalities.

- Primary sensors, has to be in the system for any kind of autonomous flight
  - Accelerometer
  - Gyroscope
  - Barometer
  - GPS
  - Magnetometers
- Secondary sensors, included in the system to perform mission specific tasks or improve operations
  - Infra-red sensors for attitude estimation
  - Gas sensor to detect the presence of harmful gases
  - Bluetooth/GSM sensors to detect phones of survivors in a disaster situation
  - Ultrasonic sensors to aid landing and/or avoid objects
  - Microphone to detect sounds in the environment
  - IR range finders for indoor navigation and mapping (generation of point clouds)
  - Camera for RGB vision to the operator and visual object detection

Definition of sensor system: Sum of all the hardware and their interfacing that provides the UAV with sensory information such as acceleration, angular rotation rates, altitude, objects in proximity etcetra. The imaging sensors (the cameras) and the laser range finding sensors have been excluded from the sensor system because they are considered as systems by themselves due to their complexity and mission relevance. For this design all the sensory hardware except for the cameras and the laser range finder are defined as part of the sensor system.

6.1.2. Requirement definition

The primary requirements on the accuracy of the sensor system were derived from the Parrot AR Drone [50]. Other IMUs and sensors studied in the works of various researchers were used to approximate the requirements on the sensors [50–53]. A lot of the papers use the Hummingbird quadrotor from Ascending Technologies. The primary reason given is the ability of the Hummingbird to send control updates at a rate of 1 kHz to the motors [52]. However that is not a requirement on the sensor, but the CPU. Most of the autonomous quadrotors studied had an update of frequency of around 50 Hz from the IMU and the gyroscopes (such as the MT9-B IMI [54] as used in [55]), therefore a value of 35 Hz was set as the update frequency requirement on the IMU and the gyroscope. The update requirement from the magnetometer and the barometer are lower since these sensors are used to correct the error in the attitude estimation of the UAV caused by the sensor drift in the IMU and the gyroscope. The required update rate for the barometer and the magnetometer were set at greater than 10 Hz. Finally the update rate on the GPS was set at a minimum of 2 Hz. This is how **Req-pl-sens-1** was defined.

Req-pl-sens-2 was defined because this was thought be a useful function for a SAR UAS; **Req-pl-sens-3** is required for indoor navigation and collision avoidance. Finally **Req-pl-sens-4** is defined to give the UAV some sense of how high above ground the UAV really is; this is required to make sure there are sensors and control inputs making sure it does not crash on the ground and also to aid the autonomous landing and take off of the UAV.

6.1.3. Final sensor choices

There were three primary options for the design of the UAV sensor system. The first and most design intensive method is to design a complete sensor solution based entirely out of individually selected MEMS chips for each required sensor, interfacing them all through an integrated circuit maybe using a breadboard, and interfacing the breadboard with the selected CPU. The second option would be to use breadboard modules designed with various sensors on-board for specific applications for the design. The sensor modules will interface with the main computer to send it attitude data. The third and the easiest option is to buy and use a complete sensor solution package, one with all the required sensors and microcontrollers for flight and processing the attitude information from the sensors built in. These boards sometimes even have autopilot applications that come with them.

Due to the budget and time constraint on the design, it was decided to go for the last option. An integrated board combining the maximum number of desired sensors will be chosen for. The expensive prices of full autopilot solution excludes that as an option; on the other hand the lack of programmability or processing power of many of the cheaper full autopilot solution excluded them as choices for this design. The final choice was between the NAVIO integrated sensor board and the AeroVero board from Gumstix. Due to the dependencies of these sensor on the CPU selected, the main sensor board selection was halted until a CPU selection was made. This selection is done in a such way as to fulfil **Req-pl-sens-1**.

The other sensors required for the operation of the UAV were ultrasonic sensors for object detection, gas sensor for sensing flammable gasses, some kind of range finding system for indoor navigation and mapping and a camera for various applications. Due to the importance of the indoor mapping and the imagining, these two functionalities were defined as separate subsystems (the LIDAR system and the imaging subsystem). A gas sensor (The MQ-2 Flammable Gas and Smoke Sensor [56]) has been selected to fulfil **Req-pl-sens-2**. Three ultrasonic sensors were also selected (Maxbotix LV-EZ0 [57]), two for object sensing in the flight direction and one under the UAV for distance to ground; this fulfils **Req-pl-sens-3** and **Req-pl-sens-4**.

6.2. Processor

In this section hardware will be selected for the processing tasks that will be performed on-board. First a literature survey will be conducted to get an indication for the computational power required. Based on this survey requirements will be set for the processor. Then the possible solutions will be presented resulting from a market analysis. From these the final components for the MAV will be selected.

6.2.1. Literature survey

A lot of research is currently performed around the autonomy of MAVs. A great part of this research is focussed on Simultaneous Localisation And Mapping (SLAM) and computer vision. These processes are computational heavy and thus require hardware with a sufficient processor. However, it is hard to calculate exactly how much computational power is required for a certain process or algorithm since this is dependent on various factors. Examples of these are the required process or algorithm speed, required process or algorithm accuracy, coding quality and hardware optimisation for that type of calculations. When reading articles to find methods for autonomous flight for Section 7.3.1 and 7.3.2, some of these papers presented experimental results together with an overview of the hardware used. To get an indication of how much processing power is required for a certain process an overview will be made of the hardware used. Based on this overview a minimum amount of required CPU power will be determined. Table 6.1 presents the results of the literature survey.

From Table 6.1 it can be seen that the autopilots requires the least amount of computational power compared to the other processes. In Section 7 it was explained the control system will use a 1 KHz update frequency. The Navio is an autopilot shield designed for the Raspberry PI and makes use of that processor. For the 1 kHz update frequency it uses around 20 % of the 700 MHz. A logical explanation for this could be that the hardware of the Raspberry is not specifically optimised for autopilot processes. NAVIO is an autopilot aimed for general MAVs, mainly of hobbyists. The same goes for PIXHAWK, which uses a 168 MHz processor, but can only go up to a 400 Hz update frequency. The Lisa series consists of (very) small and lightweight autopilot boards designed around the same 0.072 MHz CPU processor family. The Lisa/S was developed in collaboration with TU-Delft in the Netherlands and is smallest fully capable autopilot available on the market [69]. It has a dimension of 20 by 20 by 5 mm and weights as little as 2.8 grams. Both researchers and hobbyists make use of them because of their sizes. Although there were some indications of a 400 Hz maximum update frequency, no hard information was found on this property. When the hardware and the control process is highly optimized it is able to run a 1 KHz update frequency on a 60 MHz CPU. This is the case of the AscTec Autopilot. This high quality system is delivered with high performance multirotors focussed on research [41]. From Table 6.1 it can be seen that the three SLAM processes have been performed on a 1.6 GHz CPU. In fact they all used the same quadcopter and hardware. This quadcopter, the AscTec Pelican, is specifically designed for researchers [70]. It houses the AscTec Autopilot by default and can be expanded with a more powerful processing unit. These come in the form of a the AscTec Atomboard or the AscTec Mastermind. The Autopilot handles the flight controls and sensors, while the expansion can be used for user designed process. In this case they use the Atomboard for the SLAM process. It houses a Intel Atom Z530 processor running at 1.6 GHz with 1 GB DDR2 RAM. The vision based obstacle avoidance and navigation process was performed on a 600 MHz processor, which is less than the 1.6 GHz processor used for SLAM.

6.2.2. Requirements

A processor needs to be connected to other system components. This is done through a board on which the processor is directly connected, like a motherboard, breakout-board or expansion-board. Designing these is complex and currently the group has insufficient knowledge to do this. Therefore it is considered outside the scope of the project and only processors which have such processor boards[1], that meet the requirements, will be considered. The most critical processes are the SLAM ones and it is shown they can run on a 1.6 GHz processor. Since they have been performed on a Intel Atom Z530 processor, requirements have been made on the CPU together with more general requirements. These requirements can be found Appendix A.

[1] With a processor board is meant: motherboard, breakout-board or expansion-board.

Table 6.1: Overview computational power used during experimental testing

	Processor speed [GHz]	CPU usage [%]	Update frequency [Hz]	Image resolution [px]
Autopilots (including sensor management)				
NAVIO [58] [59]	0.700	50	3000	
	0.700	20	1000	
	0.700	5	200	
PIXHAWK [60] [61]	0.168	NA	400	
Lisa series [62]	0.072	NA	NA	
AscTec AutoPilot [63] [64]	0.060	NA	1000	
Visual SLAM				
M. Achtelik et al. [65] [66]	2x 1.86	60 (1 core)	30	752×480
	1.600	100	20	752×480
Laser SLAM				
A. Bachrach [3]	1.600	NA	40	
Visual SLAM with obstacle avoidance and navigation				
S. Weiss et al. [67]	1.600	NA	5	752×480
Obstacle avoidance and navigation				
K. Prashanth et al. [68]	0.600	NA	0.275	720x576
			0.549	360x266
			1.099	180x144

6.2.3. Market analysis

A market analysis is performed to find feasible possibilities for the on-board computer. Only options that have both enough computational power and can be combined with an integrated sensor board are considered. The results of this market analysis are presented in Table 6.2.

It should be noted that the Gumstix DuoVero with Aerocore extension-board does not require an additional integrated sensor board. It is designed specifically for MAVs and the sensors are integrated on the processor board already. Taking the **Req-pl-proc** requirements as defined in Appendix A into consideration, only two options remain. The AscTec Atomboard is too expensive, while the Mastermind is also too heavy. Therefore only the Gumstix Duovero/Aerocore and the Hummingboard with Navio autopilot shield will be considered as viable options. Both processors run the Linux operating system by default. This is beneficial for running programs/code besides the autopilot. Both MATLAB and Phyton support Linux. Additionally a lot of open source programs/algorithms are written for Linux. For example the iSAM library[2], used in the SLAM process as will be explained in Section 7.3.2, is written for Linux. Also EBLearn[3], which is used in the number recognition algorithm as explained in Section 7.7.2, is mainly designed for Linux. Both the CPU and the GPU of both processors is more powerful than the ones from the Intel Atom Z530 processor [71]. However, since the core speed is lower (1 GHz versus 1.6 GHz) it could be possible that a single core is not sufficient for the computational heavy programs. In that case the programs should be written such that they use two or more cores. This should be tested and verified during the programming phase.

6.2.4. Final components

To decide on the processor a trade-off will be performed between the Gumstix Duovero/Aerocore and the Hummingboard with Navio autopilot shield. An overview of the features is given in Table 6.3 and the final trade-off scores are given in Table 6.4. Weights of 1 and 2 were assigned and scores between 1 and 3, with 3 being the highest possible.

In the trade-off, the weight and dimensions were given a lower weight since both boards are designed to be used on MAVs. The computational power and ram were given a higher weight since these directly influence how much algorithms can be run on the MAV. They are also important for image processing and are features that could be of high importance for future work as well. The power consumption was given a lower weight since there will be no substantial difference and it is a relatively low amount on the total MAV power budget. The sensors were also given a low weight since any missing sensor/lower performance sensor can be added/replaced using an external sensor. The interfaces were deemed important again since this is one of the limiting factors on what can be connected to the processor. Although now everything has been designed to connect, maybe in future work additional/different interfaces are required. From the total scores of the trade-off and the substantial difference between them it can be seen that the Hummingboard with Navio autopilot shield won. Also changing a weight or score of an entry does not influence this result. Therefore the Hummingboard with Navio autopilot shield will be used as processor in the MAV.

6.3. Imaging subsystem

A camera is required to perform mission element A [29]; other applications that it may serve is the streaming of a live feed to the ground station for the benefit of the operator and also object detection to aid the navigation of the aircraft. Firstly the imaging subsystem is defined in as much detail as possible. However, the defining requirements for the camera selection comes from mission element A. Thus before one or more cameras are decided upon, a rough analysis is carried out on mission element A; the following text presents this analysis. After that a discussion is made into the types of cameras available in the market that can fulfil the requirement and the camera chosen to perform the mission. Following this a final selection is made on the imaging subsystem.

Imaging subsystem definition: The imaging subsystem is defined as all the hardware and software elements in the UAS that primarily accommodate the acquisition and processing of images and videos. Due to the multi-subsystem interaction of the CPU, the sensors and the CPU are defined as separate subsystems and are not part of the imaging subsystem. All the on board cameras and the on board and off board image processing software and algorithms are considered part of the imaging subsystem.

6.3.1. Imaging subsystem requirements

This part of the report defines the requirements on the imaging subsystem of the UAV. Many of the requirements in the following list have not been fully defined (mentioned as TBD); this chapter will attempt to define

[2] http://people.csail.mit.edu/kaess/isam/

[3] http://eblearn.sourceforge.net/

Table 6.2: Overview processors resulting from market analysis

	AscTec Atomboard	AscTec Mastermind	Gumstix DuoVero/Aerocore	Hummingboard
Processor	Intel Atom Z530 (1.6 Ghz)	Intel Atom D510 (2 x 1.66 GHz), Intel Core 2Duo SL9400 (2 x 1.86 GHz) Intel Core i7-3517UE (2 x 1.7 GHz) Intel Core i7-3612QE (4 x 2.1 GHz)	TI OMAP4430 - A9 Dual core (2 x 1 Ghz)	Cortex-A9 Freescale i.MX6Q (4 x 1 GHz)
Ram	1 GB DDR2	4 GB DDR3	1 GB DDR2	2 GB SDR
Interfaces	MicroSD 7x mini USB 2.0 2x UART 1x mini-PCI-Express 1x LVDS	MicroSD 1x VGA 5x USB 2.0 2x USB 3.0 3x FireWire 6x UART 1x GBit Ethernet 1x GPIO 1x Cfast 1x SATA 1x mSATA/Mini-PCI-Express (shared) slot 1x mini-PCI-Express	MicroSD 1x SPI 1x USB 2.0 (up to 4 devices) 2x I2C 3x UART 2x ADC 8x PWM output 1x 100Mbit Ethernet	MicroSD 1x MIPI CSI input 1x Combined MIPI CSI and DSI connector 1x LVDS 1x HDMI (video and audio) 1x 3.5 mm Jack (audio) 1x Coax SPDF 1x Infrared Receiver 1x mini PCI-E connector 1x mSata connector 2x USB 2.0 1x 1 Gbit Ethernet WiFi IEEE 802.11 a/b/g/n Bluetooth 2.1
Delivered with	Bootable 8 GB microSD with Ubuntu Linux	Bootable 60 GB mSATA SSD with Ubuntu Linux	-	-
Optional	WLAN module	WLAN module	W2CBW0015 WiFi and Bluetooth module U-Blox NEO-7M GPS Receiver	5 MP camera module
Required	LVDS display	-	-	-
Integrated sensor board	AscTec Autopilot	AscTec Autopilot	Gumstix Aerocore	Navio
Weight [g]	105	280-325	40	55
Cost [€]	1119	1799 - 2399	235	120

Table 6.3: Feature overview of the Gumstix Duovero/Aerocore and the Hummingboard with Navio autopilot shield

	Gumstix DuoVero/Aerocore	Hummingboard/Navio
Weight [g]	40	75
Cost [€]	235	210
Computational power	ARM A-9 Duocore 1GHz	ARM A-9 Quadcore 1 GHz
Ram [GB]	1	2
Dimensions [mm]	96.5 × 50.1 × 10.0	85.60 x 56 x 30
Power consumption [W]	7	7
Sensors	MS5611 barometric pressure sensor	MS5611 barometric pressure sensor
	MS5611 temperature sensor	MS5611 temperature sensor
	MPU9250 3-axis gyroscope (MPU-6500 chip)	MPU9250 3-axis gyroscope (MPU-6500 chip)
	MPU9250 3-axis accelerometer (MPU-6500 chip)	MPU9250 3-axis accelerometer (MPU-6500 chip)
	MPU9250 3-axis magnetometer (AK8963 chip)	MPU9250 3-axis magnetometer (AK8963 chip)
	U-Blox NEO-7M GPS Receiver	u-blox NEO6T GPS module
Interfaces	1x MicroSD	1x microSD
	1x SPI	1x MIPI CSI input
	1x USB 2.0 (up to 4 devices)	1x Combined MIPI CSI and DSI connector
	2x I2C	1x LVDS
	3x UART	1x HDMI (video and audio)
	2x ADC	1x 3.5 Jack (audio)
	8x PWM output	1x Coax SPDIF
	1x 100Mbit Ethernet	4x ADS1115 16-bit ADC
		1x Infrared Receiver
		1x mini PCI-E connector
		1x mSata connector
		1x I2C connector
		2x USB 2.0
		1x UART connector
		1x SPI connector
		13x PWM output
		1x 1000Mbit Ethernet
		WiFi IEEE 802.11 a/b/g/n
		Bluetooth 2.1

Table 6.4: Overview of the processor trade-off weights and scores

		Gumstix DuoVero/Aerocore		Hummingboard/Navio	
	Weight	Score	Weighted score	Score	Weighted score
Weight	1	3	3	2	2
Dimensions	1	3	3	1	1
Computational power	2	1	2	3	6
Ram	2	1	2	3	6
Power consumption	1	2	2	2	2
Sensors	1	2	2	2	2
Interfaces	2	1	2	3	6
Total	**10**		**16**		**25**

those requirements and link them to the list of requirement by looking at the functionalities that the camera is supposed to perform.

Intended functionalities of the imaging subsystem:

1. The imaging subsystem will provide a means to survey the terrain and make a map for SAR operations; it should provide at least a ground resolution of 11 cm (resolution level A) as dictated in the IMAV documents. **Req-pl-imag-1**

2. The imaging subsystem will have an imaging camera with a range of AoV, wthat will allow the UAV to perform mission A (the terrain survey) without blurring, between an altitude of 20 to 280 m. The lower limit of 20 m is set to avoid trees and the 280 m upper limit comes from the safety rules. **Req-pl-imag-2**, **Req-pl-imag-3**

3. The imaging subsystem will be able to interface with the UAV computer for on-board real time image processing; it should be able to send images to an on board processing unit or off board computer for image processing. The image has to be sent to the computer at a rate to allow the system to process the image, recognize objects and react to the objects fast enough to avoid it from colliding with them during flight. As a fail safe it will be able to send at least the mapping footage and live first person view to the ground station. **Req-pl-imag-4**

4. The imaging subsystem will have a pointing accuracy of ± TBD ° for object detection and recognition and a jitter of less than ± TBD ° to prevent blurring. **Req-pl-imag-5**

5. The imaging subsystem will at least be able to detect road blockages in the terrain map for mission A, detect objects such as windows and obstacles to aid the UAV control for mission B and C, and recognize numbers for mission B and D. **Req-pl-imag-6**

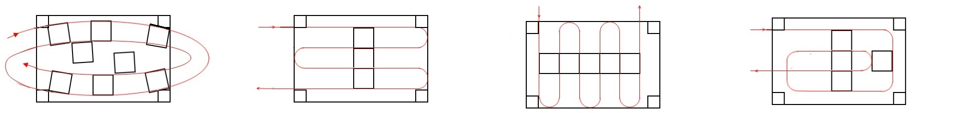
6. The imaging subsystem will have a camera with tilt freedom so that it maybe oriented to the ground or towards the front; the servo mechanism should have a TBD angular rate. **Req-pl-imag-7**

7. The imaging subsystem will have a mass of lower than 150 grams. **Req-pl-imag-8**

8. The imaging subsystem will be able to complete mission A within TBD minutes. **Req-pl-imag-9**

IMAGING SUBSYSTEM REQUIREMENTS FOR MAPPING

Mission A analysis: To obtain the visual map for mission element A, the aircraft has to fly a flightpath over area A as defined in the IMAV2014 guidelines [29]. Various patterns are possible for this flightpath; a number of these patterns are considered (see Fig. 6.1) and analysed. The IMAV 2014 competition site was used as the standard for the ground to be mapped. The two assumptions made for the analysis of the flight paths are starting at the top left edge with a 25 m offset (the small black squares in the corners in Fig. 6.1) and having a path spacing of about 40 m (the big black squares between parts of the flight path in Fig. 6.1) for all the patterns. The length of each path is calculated graphically and other characteristics observed; this has been tabulated in Table 6.5. Based on these an optimized flight path for the IMAV mission is selected. Moreover, due to various advantages and disadvantages of each flight path it was decided to either give the user the ability to define the type of flight path (s)he wants the UAV to fly for mapping based on the specific situation; for better autonomy the on board computer should be given the intelligence to optimize its flight path from the options available based on the specific situation.



(a) Inward elliptical spiral (b) Horizontal parallel (c) Vertical parallel (d) Inward rectangular spiral

Figure 6.1: Types of flight paths considered

Table 6.5: Analysis of flight patterns

Flight Pattern	Length [m]	90° turns	Remarks
Inward elliptical spiral (Fig. 6.1a)	1554.5	-	Gets image of the perimeter first. Long path; complex manoeuvre; Path gap requirement not met.
Horizontal parallel (Fig. 6.1b)	1066.8	6	Short. Greater crosswind interference due to long path.
Vertical parallel (Fig. 6.1c)	1028.7	10	Shortest route; less crosswind interference due to shorter paths. A lot of turns; exits far from centre; edge distance requirement not met
Inward rectangular spiral (Fig. 6.1d)	1059.2	6	Short; ends in the centre. More possible interference from crosswind.

After the analysis it was decided to use the inward rectangular spiral pattern (Fig. 6.1d) for the first iteration of the IMAV mission. This was done because it had a flight which is approximately 50 metres bigger than the smallest path and also because it is advantageous to have the UAV in the centre of the mission location after it is done surveying the terrain. It should be stated here that the analysis on the flight path was very rough with distance measurements carried out by physically measuring the paths on a scaled down map; more accurate analysis should be carried out on the path optimisation for mission A.

Camera requirements: From the above analysis a number of specific required performance qualities of the camera were defined; these have been discussed below.

Resolution requirement: With the selected flight path, the individual image dimensions can be calculated assuming some kind of overlap between each of the captured image and a flight path. Fig. 6.2 shows the assumed overlap of the images during the area mapping mission; from this the number of image widths that will be equivalent to the area width can calculated. For the benefit of the calculation the edge images are considered to have 60 % of their width not overlapping; images between the edge images are considered to have 80 % of their width not overlapping. Furthermore, the number of images in the North-South direction has been defined by the selected flight plan to be four. Thus 2.8w should be equivalent to the width of the area to be surveyed (which was approximated to be 173 m). This results in a ground image width (the dimensions on the ground that each image will span) of about 61.8 m. By assuming a wide photograph and aligning the wider side of the image with the North-South direction of the terrain (to maximize resolution), the ground image breadth is approximated to be 30.9 m.

Now that the ground image dimensions have been approximated, the pixel requirement for the images can be derived for each of the resolution levels as defined in the IMAV documents. This is done by first calculating the ground pixel length for each of the resolution levels and multiplying that by the ground image dimensions. For a detail level of D it was found that the camera should be 477.5 megapixel and for detail level C the requirement is a 19.1 megapixel camera. The results of the analysis have been tabulated in Table 6.6.

A python script was written that calculates the required ground image dimension and mission A (see Fig. 6.3) path based on the resolution of the camera and the desired resolution of the user. This file gives other useful outputs such as the required altitude to perform the mission based on the AoV the camera (which is given as an input) and the motion blur on the sensor to approximate the blurring. This script is later used to verify that requirements **Req-pl-imag-1**, **Req-pl-imag-2** and **Req-pl-imag-3** are met. It is assumed that fulling **Req-pl-imag-1** will automatically fulfil **Req-pl-imag-6-1** as it is assumed roadblocks will be bigger than 10 cm and the ground station will have object detection programs. Furthermore, the requirement on

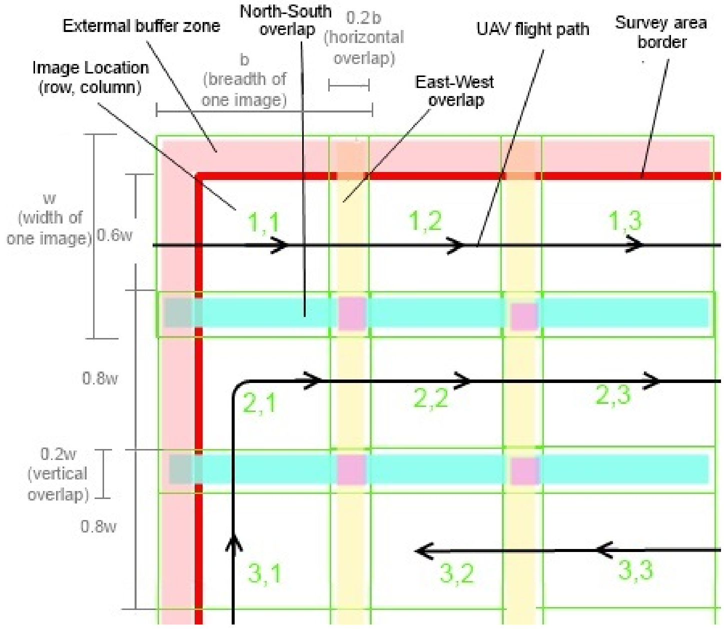


Figure 6.2: Inward rectangular spiral

Table 6.6: Resolution requirement calculation results

Area dimensions [m]	
North-South	267
East-West	173

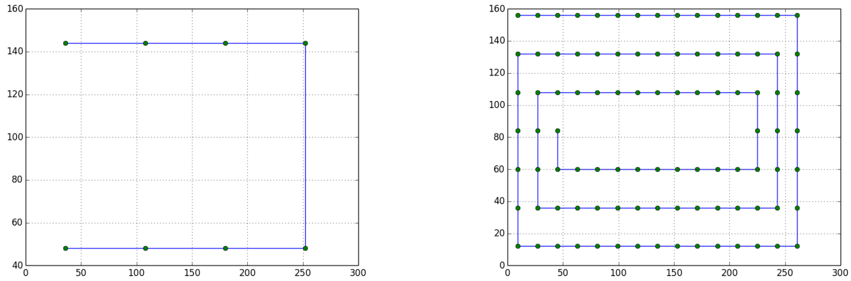
Ground image [m]	
Width	61.8
Length	30.9

Number of images	
North-South	4
East-West	12

Required	
North-South	4
East-West	12

Required Image Resolutions [MP]	
A [0.118 m per pixel]	0.1353
B [0.04 m per pixel]	1.194
C [0.01 m per pixel]	19.1
D [0.002 m per pixel]	477.4

the pointing accuracy is derived for the specific mission after camera selection; using the arctangent of the height and half the ground overlap, a minimum pointing accuracy (**Req-pl-imag-5-1**) can be found for which it will be confirmed that at least half of the defined overlap will actually occur.



(a) Mission path and imaging points at resolution level B **(b)** Mission path and imaging points at resolution level C

Figure 6.3: The path and imaging points as generated by the MisAPlan.py script for two resolution levels; 4000x3000 pixel image dimensions; mission area from IMAV2014 competition

Motion blur: The shutter speed and focal length of the camera along with the flight velocity and altitude of the aircraft will determine how blurry the images will be. The shutter speed is the duration for which the shutter remains allowing light to fall on the imaging sensors. The focal length of the camera determines the point behind the lens where the rays from a point from the world converges on the imaging sensor; this gives an indication of the zooming and focus capacities of the camera among other things. If rays emerging from a point on the ground fall on two consecutive pixels on the sensor inside the camera, the image will appear blurred. Thus the motion of the aircraft might lead to blurring if the shutter speed of the camera is too slow or the focal length too high since this would mean the point moves more on the sensor; on the other hand if the aircraft flies too fast or too low there would also be more blurring for similar reasons. The basic requirement is the motion of a ray from the point on the ground must move less than one pixel in the sensor for there to be no blurring. This is calculated using Eq. (6.1) [72]; in the equation, b is the displaced distance of a ray from the ground on the sensor, f is the focal length of the lens, t is the shutter speed, H is the height of the aircraft and V the velocity of the aircraft. Since the flight altitude for mission A will depend on the selected camera this equation cannot be used to put an explicit requirement on the camera, however after selection it can be used to estimate the amount of blurring. Furthermore, the blurring requirement also puts a limit on the acceptable jitter of the aircraft. The camera jitter should be at least less than half the ground resolution to prevent blurring at the imaging altitude. The blurring and jitter calculations are incorporated into the MisAPlan.py file and the requirement derived and added to the requirement list.

$$b = \frac{f}{H} V t \tag{6.1}$$

Lighting requirements: Due to the high shutter speed required for minimizing the motion blur on the images and the small aperture for a farther focus, it is possible that the sensor will not receive enough illumination to make readable images. The four factors that influence the brightness of an image are the luminance of the imaging subject, the shutter speed, the ISO sensitivity and the aperture width. The shutter speed has already been discussed; the aperture width is the diameter of an opening in front of the sensor controlled by a mechanism. This mechanism controls the amount of light getting into the sensor. Finally, the ISO sensitivity determines the amplification of the light that falls on the imaging sensor. Since the aperture and the shutter speeds are minimized for the performance of mission A and the environmental luminance is not controllable, it was decided to use a high ISO setting to compensate; this would result in a noisier image.

The relationship between aperture width, shutter speed, ISO speed and luminescence has been described by Sidney F. Ray in chapter 19 (Camera exposure determination) of the book The Manual of photography [73]; this can be seen in Eq. (6.2). In the equation N stands for the relative aperture, t the shutter speed, L the average scene luminescence, S the ISO speed and K the reflected light meter calibration constant. If there is sufficient time after camera selection, an analysis will be carried out using Eq. (6.2) to verify the performance of the imaging subsystem. This analysis is related to **Req-pl-imag-9**.

$$\frac{N^2}{t} = \frac{LS}{K} \tag{6.2}$$

Imaging subsystem requirements for object detection and avoidance

Besides the resolution requirement for mission A there are resolution requirements for object detection and avoidance on the imaging subsystem. Most of these criteria were defined after the first iteration. A number of assumptions were made to determine the remaining time to react to objects; the assumptions about the velocity and acceleration were made by leaving margins from the performance limits of the UAV. For the UAV to detect objects and avoid them for mission B and C the outdoor velocity is assumed to be 3.5 m/s, the indoor to be 1 m/s and a maximum deceleration to be 1 m/s^2; and for the outdoor element objects 10 cm long should be identifiable 20 m away and for the indoor navigation, objects 1 cm long should be identifiable 5 m away.

The requirement **Req-pl-imag-6** is defined as degrees of the camera FoV per pixel of the image dimension; it is calculated from the above assumption using Eq. (6.3) (Fig. 6.4). In the equation l is the object height; one pixel is assumed to be half its length to allow for its detection thus l/2 is used; d is the distance of the object from the camera; θ is the required angle subtended at the camera sensor by one pixel. The degree of FoV of the camera per pixel for a specific camera can be directly found by dividing its FoV with the pixel length of its image. It is found that the out door requirement is 0.14 °/pix and the indoor requirement is 0.11 °/pix. Half of this angle can be considered the required minimum jitter (**Req-pl-imag-5-2**), thus 0.05 ° .

$$\theta = \arctan(\frac{l}{2d}) \tag{6.3}$$

6.3.2. Camera selection

This part of the report describes some camera systems and describes the design steps taken to select a camera. A trade-off is carried out between the different types of cameras available to make a selection.

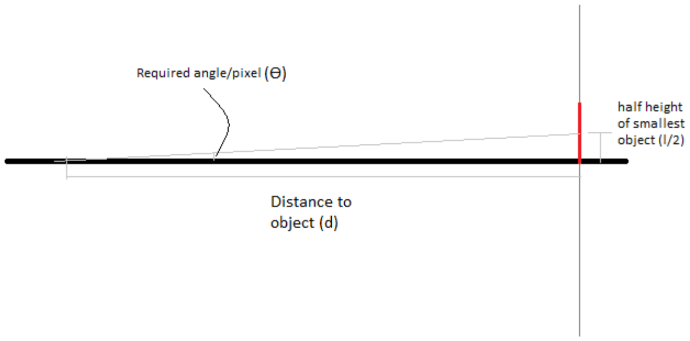


Figure 6.4: Figure explaining the calculation of the required °/pix from the camera

Types of cameras: Following the resolution requirement the camera selection was carried out. A wide variety of cameras were explored as options for the system. The size, interfacing, relative mass, data transmission over RCA (audio video cable) and image quality were analysed qualitatively and tabulated in Table 6.7. The size is relevant for the analysis as it influences drag The camera should be controllable by the on board computer; by controllable it is meant that the UAS should be able to switch modes in the camera, change focus, ISO settings, change between high resolution photography to video mode and vice versa, turn the camera on and off etcetra. The ability of a camera to easily send video feeds over RCA cables would be useful since this would mean the camera can be directly connected with an FPV system; the direct connection would take decreasing the processing requirements of the on-board computer as the main imaging and video data bypass it. Finally the image quality, meaning the resolution of images and videos of the cameras, is a very relevant property for the UAV design; the resolution determines how well the UAV can perform a variety of missions, along with how much information is available to the UAS for image based object detection.

During the design, the camera control was found to be a big challenge. Although the IP and CSI cameras are easily interfaceable and controllable by the main computer, it is a great challenge for the digital and action cameras. These cameras have closed source firmware which cannot be programmed and require memory and processing intensive applications to run on a desktop computer. Applications released from the user community such as the CHDK [74] provides some options for interfacibility with the digital cameras; however this kind of hacks into the cameras and CHDK is only compatible with certain Canon cameras. However, if a cheap and easily implementable high resolution imaging solution is needed, an interfacing option into these higher resolution cameras needs to be found or developed.

In the trade-off interfacing and relative mass were given the greatest weights due to the importance of a light system for the UAV and the need for easy interfacing for camera control and image processing. The next greatest weight is assigned to the image quality due to the importance of high resolution images for the mapping and object detection at longer ranges; transmission and size are given the least weight, because size is important for drag and subsystem positioning while the transmission can be controlled by imposing a lower frame rate on the live feed and decreasing the bandwidth requirement for mission A. A brief description of the types of cameras are presented below. The design of the initial imaging subsystem (that is camera and mounting) had a constraint of 200 g for the first iteration; this value was later reduced to 100 g (only for main imaging camera) due to the separate mass budgeting of the camera support structure (VDECS).

Digicams Digital cameras were explored since they are capable of taking high resolution images which can be useful and they maybe directly interfaced with an FPV system to send information over RCA connectors; however their drawback is a greater mass and interfacing difficulties. The digital cameras studied were the Sony DSC-QX100 [75] and the Canon Powershot ELPH 150 IS [76].

CSI Smaller CSI cameras that can directly interface with the CPU were considered; they are often lightweight and compact and are useful for on board image processing due to their fast data communication rate with the CPU. Its major drawbacks are the fact that it is more processor intensive to stream their videos to the ground station and their limited image resolutions. The CSI cameras studied were Raspberry Pi camera module [77] and the VCC-F32FV19CL [78].

IP Cameras that interface with the main computer using the Internet Protocol were also looked into; they can also communicate directly with the main computer over WLAN or LAN which is slower than the CSI camera. They are heavier and bulkier than CSI cameras and their resolutions are limited. Their defining advantage is the market availability of these products, their robust casing and the possibility of having a ready made camera with tilt and yaw freedom. The IP cameras studied were Arecont 2115v1 [79] and Axis M10 [80].

Action These are light and compact cameras capable of high resolution photography; although their resolutions are lower than that of digital cameras. Also like the digital cameras they cannot directly interface with the main computer, but they can interface with an FPV system to send image information over RCA. Due to the overall performance of these cameras over the criteria analysed, they are a good option for the terrain mapping mission. The action cameras studied were GoPro Hero3+ [81] and the Sony HDR AS15 [82].

Table 6.7: Trade-off on the type of camera

	Weight	Digicams	CSI	IP	Action
Size	0.15	0.15	0.6	0.3	0.45
Interfacing	0.25	0.25	1	0.75	0.5
Relative mass	0.25	0.25	0.75	0.5	1
Image quality	0.2	0.8	0.2	0.2	0.6
Transmission	0.15	0.45	0.15	0.15	0.6
Total	1	1.9	2.7	1.9	3.15
Normalized		0.196	0.279	0.196	0.326

Trade-off results: The weighted sum method was used to carry out the trade-off. First the weighting for each criteria was determined based on previously presented argument as a percentage. Each criterion was scored between one and four qualitatively; then they were multiplied by the weight for the criteria and all the scores summed to get the final score for the camera.

As the the trade-off shows, actions cameras have a distinct advantage over the other types of cameras. Note that the CSI camera was the second option in the trade-off; due to its interfacing capabilities it was later added as a second camera in the design. In the first iteration of the design only actions cameras, IP cameras and digicams were studied. The result of this study was a choice between the Sony DSC-QX100, the Arecont 2115v1 and the GoPro Hero3+; the Hero3+ was selected from these three, since research into the camera revealed that it has a BUS which can be interfaced with to control the camera and get the imaging data [83]. The Hero3+ BUS is interfaced with a microcontroller which converts the video data to the RCA format for interfacing with the FPV communication system. The camera can be controlled by influencing the inputs on the other ports of the BUS. The details of how the camera will exactly interface with the FPV system and the CPU have not been defined in this design.

A sensitivity study was carried out on the trade-off. Two things were checked, first the effect of swapping the weights around and the second of swapping the scores around. Re-ordering the weights, that is giving the current highest weights the least and the least weights the highest values, it was found that the action camera score was relatively unaffected but its closest competitor (the CSI camera) increases by 2%, so it wins by a smaller margin. It was found that decreasing the score of the Hero3+ in relative mass and transmission criteria would lower the margin between the action camera and the CSI camera to 8 %. On the other hand, increasing the score of the CSI camera in its worst performing criteria (image quality and trasmission) lowers this gap to 10 %. Bigger changes of scores lead to the CSI camera wining the trade-off for most cases.

In the second iteration of the design a camera system capable of interfacing with the main computer through the CSI port was also added to give the UAV on board object detection abilities; the Raspberry Pi camera module was selected for this purpose due to its light weight and compact form and the high pixel quality of its images.

Other details: Due to requirement **Req-pl-imag-7** the a camera in the UAV will have to have tilt freedom; another detail of the imaging subsystem design is the interfacing of the main camera with the UAS. This part of the report describes the design decisions regarding the other details of the imaging subsystem. The tilt freedom is obtained by using a servo in the camera mount giving it tilt freedom. This requirement has been translated as a requirement on the structure (VDECS) through **Req-pl-struct-5**. The other requirement for the system is interfacing capability with the on-board computer. As discussed before this proved to be a challenge, and it was even considered to not use a high megapixel camera due to this lacking. With more study into cameras it was found that some of these cameras offer interfaces for popular operating systems to interface with them. Some were even hacked by the user community to get more functionalities and control on the hardware. An online source [83] provided interfacing guidelines with the GoPro Hero3+; using this option one can control the modes of the camera and directly get the video output feed of the camera. This would require the programming of a EEPROM (Electrically Erasable Programmable Read Only Memory) microcontroller that will aid the interfacing with the camera. The source [83] only states the steps of getting the video feed out in detail; however they also mentions the output pin configuration of the interface and there is an active GoPro Hero camer user community who are trying to interface their to get the most out it. It is believed that these sources will provide sufficient knowledge to interface with the camera and control it. An important note here is that if it not possible to control the camera using this method, a reselection of the imaging camera will need to undertaken to find a camera that can be interfaced. Since a microcontroller circuit board will be used to interface with the camera, it was considered to use this microcontroller in conjunction with a gyroscope MEMS or a magnetometer MEMS to stabilize yaw angle of the camera. This would be done by creating a PID controller that takes the desired orientation of the camera as input, and uses the MEMS readings to give the servo a desired orientation.

It was decided to perform mission A with detail level C. If done with a cruise velocity of 6 m/s, the mission can be finished in 306 seconds if performed at an altitude of approximately 63.3 m with a horizontal camera AoV of about 65 degrees. It should be noted here that although this will meet the requirement of resolution, it will set harder requirements on the jitter and pointing accuracy of the camera on the structural subsystem.

Final design and requirements compliance

The final design of the imaging subsystem will consists of the main GoPro Hero3+ main imaging camera and the Raspberry Pi camera module; the minimum required °/pix has been fulfilled by both the on board cameras; the minimum vertical °/pix of the Hero3+ is 0.016 and for the Raspberry Pi camera module is 0.013. Structural requirements were passed on as requirements on the camera and electronics structure (VDECS) design. Software requirements relevant to imaging were discussed; certain software capable of stitching images and doing image processing were demonstrated in Section 7.7.1. It was shown that Hugin can be a realistic option for image stitching with certain optimisations; it was also shown that Monteverdi can do feature recognition to a realistic enough degree to carry out some kind of road block detection from the high resolution maps.

The following list discusses the compliance or non-compliance of the current design parameters with the requirements defined for the imaging subsystem:

Req-pl-imag-1 This requirement is fulfilled by the GoPro with an imaging resolution of 12 MP and image resolution of 4000 x 3000 pixels, mission requirement C will be performed.

Req-pl-imag-2 The imaging altitude is 63.3 m.

Req-pl-imag-3 The horizontal AoV of the GoPro was used as an input to calculate the height; a horizontal AoV of 64.6 was used [84].

Req-pl-imag-4 A BUS hack for the Hero3+ as learned in [83] will be improvised to feed the camera data through an FPV and also control the camera mode and other camera properties. This will require programming of the microcontroller in greater detail; this has not been discussed or verified in the project; However and independent communications system capable of sending interfacing directly with the camera through a RCA (also known as the Audio Video cable) cable has been designed.

Req-pl-imag-5 These requirements are translated as requirements on the camera structure (VDECS) through **Req-pl-struct-6** and **Req-pl-struct-5**.

Req-pl-imag-6 **Req-pl-imag-6-1** has been satisfied by the resolution of the mapping and the image processing; **Req-pl-imag-6-2,6-3** are defined from requirements on the outdoor and indoor navigational requirements on the imaging subsystem; **Req-pl-imag-6-4** is based on the number recognition requirement.

Req-pl-imag-7 This requirement is translated as a requirement on the camera structure (VDECS) through **Req-pl-struct-5**.

Req-pl-imag-8 This requirement is met by the Hero3+ as its mass is 74 g.

Req-pl-imag-9 It has not been verified if the requirement on luminescence has been met, however the Hero3+ has a maximum ISO sensitivity of 6400 and a way to control exposure; manipulating those qualities can be used to get the required luminescence required for bright images.

Req-pl-imag-10 It has been calculated that with a cruise velocity of 6 m/s it will take a little over 5 minutes to finish the mission, verifying that this requirement has been met.

6.4. Indoor navigation

This section will explain more about the indoor navigation required for certain mission elements. First some general information about the method will be explained. Next the hardware on the MAV will be presented. The algorithms used for indoor flight are explained in 7.3.2, as part of the GNC section.

6.4.1. General method

The MAV should be able to search inside buildings for survivors and other possible objects of interest. To perform autonomous indoor navigation the MAV needs to know where it is located inside the building and where it (still) needs to go. This is done by flying through the building while creating a map of the building interior and determining where the MAV is inside the building and/or on the map. This process is also known as Simultaneous Locating and Mapping (SLAM). Besides using the map for MAV navigation it will also be very useful to emergency services. With the map and possible locations of interest marked they can take quick and effective action. To perform SLAM one should be able to detect objects and their distance. There are three main options to do so for SLAM: laser, vision and sonar based detection. Laser and vision based SLAM is mainly used on MAVs and ground robots, while sonar (but recently also vision) based SLAM is more common for underwater vehicles [85]. Laser systems are attractive because they are accurate active sensors. They can provide up to 360 ° object detection and do not require visual reference features. However they are relatively heavy and expensive. On the other hand vision systems are passive and less acurate. The computational cost is considerably high and good visual features can be difficult to extract and match. Cameras however can be very cheap and lightweight and additionally used for other features. Currently vision based SLAM is mainly preferred over laser based as the latter has a restricted perception distance and is still heavy for MAVs [67]. Ultimately the MAV system designed here will be used for search and rescue missions. It will serve as a tool for emergency services to locate survivors and to create (detailed) maps of areas and buildings. Because lives of both casualties and emergency workers are at stake, they should be able to work as safe and efficient as possible. It is therefore important that the maps can be made consistently, while they are accurate and of high quality. Since laser systems provide the highest measurement accuracy and have a better performance in environments lacking visual features, this method is chosen to perform the object detection.

6.4.2. Laser scanner

Since the laser scanner will be mounted on the MAV weight is an important factor and because of the limited budget also the costs are important. But as mentioned before, laser scanners are relatively heavy and expensive. Laser scanners come in 2D and 3D versions, but only 2D is required to create a map (floor plan) of a building. The lightest of-the-shelf scanner found on the internet was the RPLIDAR by Robopeak [86]. This 2D Laser Imaging, Detection and Ranging (LIDAR) system has a weight of 200 gram with a price of about € 294. It scans 360 ° with an angular resolution of 1 °. The distance resolution is 0.2 cm but the range is limited to 6 meters. Especially the weight of this scanner is substantial compared to the rest of the MAV components. To see if there is an opportunity to go lighter and cheaper it is decided to come up with an own design.

When searching for LIDAR systems the kick-starter project LIDAR-Lite was found on Dragon Innovation [87]. This project currently raised their fund and established the company PulsedLight. *"PulsedLight targets the need for very compact optical distance measurement sensors in cost-sensitive markets such as robotics or for applications requiring increased range or size reduction. Implemented as a single-chip processing solution, PulsedLight enables a new class of optical distance measurement sensors that exceed the performance of current solutions at a substantially lower cost [1]."* LIDAR-Lite is a very small laser distance measurement sensor using inexpensive, off-the-shelf, electro-optical components. The thing that makes LIDAR-Lite so special is that it is capable of measuring out to 50 meters, weights 12 grams and costs only € 58. This is made possible because they use a new flight time determination method. The sensor houses removable or interchangeable optics to meet specific range and/or beam angle requirements. The standard accuracy 1 %, but the developers indicated that the accuracy can be increased by using the built in external clock reference. When connected to the Hummingboard, with a clock accuracy of 50 ppm [88], the measurement accuracy would increase to $5{\cdot}10^{-3}$ %. It has an I^{2}C interface and can perform measurements on command. It can measure up to 10 Hz at a range of 50 m. The maximum frequency is up to 100 Hz, at which it the maximum range becomes 25 m. A picture of a LIDAR-Lite prototype can be seen in figure 6.5. Because it is a very light, cheap and long range measurement sensor it suits the needs of the MAV design. Because this sensor is currently one of the first to achieve this, it will be used in the design.

However, this sensor only scans in a straight beam and does not rotate. Therefore the sensor will have to be placed on a rotating platform to get a scan in multiple directions. The easiest solution is to put the LIDAR-Lite system on a spinning platform, powered by an small motor. This platform will have to be placed on top of the MAV, such that is will have a clear all-round view. A strong and lightweight option is a thin carbon fibre reinforced composite plate to mount the LIDAR-Lite on. Since this sensor is very light, two sensors can be put on the platform in opposite direction. This would benefit in a symmetric weight distribution and doubles the amount of measurements over a time period which is beneficial for the angular resolution in case there is a time constraint on a 360 ° scan. The wired power and signal connection will have to be transferred to LIDAR-Lites on the platform. A wireless connection is likely to interfere with the other systems on the MAV. Therefore a slip ring will be used. Additionally the slip ring can be used as the shaft to rotate the platform with and is

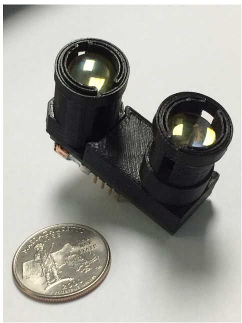
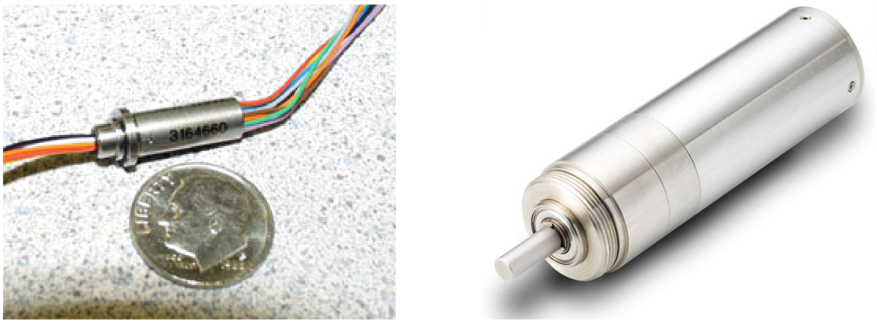


Figure 6.5: LIDAR-Lite prototype [1]

therefore considered the easiest and cheapest solution. As an off-the-shelf product the all purpose slip ring capsule model 2040-00 from Electro-Miniatures Corporation has been chosen [89]. It is made out of stainless steel such that it is strong enough to carry the platform. It houses up to 10 rings at 1 A maximum. This should be sufficient since the laser sensor requires a maximum of 100 mA and has up to 5 connections. The slip ring is rated at a maximum of 225 RPM, which translates into 3.75 RPS limits is the maximum frequency the platform can spin at. The slip ring can be seen in figure 6.6a.



(a) Slip ring model 2040-00, example part for the laser scanner. [89]

(b) PMI mini metal gearmotor, example part for the laser scanner. [90]

Figure 6.6: Pictures of the example components for the laser scanner.

The selection of the motor to drive the platform is important as well. The motor has to be small and light weight but still provide sufficient torque. However small motors have a very high RPM rate with little torque. Therefore a gearbox should be added to lower the RPM and to increase the torque. The company PMI offers a series of miniature brushless DC motors with gear-heads that can be screwed on them. This enables the customer to select any engine with compatible gear-heads to meet the required RPM and torque. The gearbox and the platform will be connected by a small rubber band, which can have a gear ratio over the two shafts as well. This system is proposed to use as driving unit, as it is modular and can be used to provide a wide RPM range, in case a different distance system or other equipment would be used. An example of an item combination is a size 5 Model B0504-050 with a 25 gear ratio gearbox [90]. With an additional gear ratio between the gearbox and platform of 6 this would result in a no-load RPS of 4.24. By adjusting the input voltage this can be tweaked to meet a required RPS up to 3.75.

To generate the point cloud, the location of the measured point with respect to the MAV should be known as well. This requires two variables: the attitude of the MAV and the direction of the scan. The attitude is known through the on-board sensors. The direction of the scan is still unknown. To determine this angle, an cheap and lightweight (less than € 1 and 1 g) IR reflective sensor is suggested [91]. This sensor measures how much infra-red radiation is reflected. By placing a white dot on the black plate which is rotating, the sensor will sent a signal when the dot passes by. From this the RPS and the orientation at that moment can be determined. By integrating the RPS and adding the known orientation at a time instance, the orientation of the platform is known at any time instance. By correcting the orientation and range with the attitude angles an accurate 2D point cloud can be created. A impression of the proposed system can be seen in Fig. 6.7.

An overview of the component weigh and cost is given in Table 6.8.

From Table 6.8 it can be seen that the laser scanner weighs 61.9 grams while it will cost around € 153. It is able to perform a 360 ° scan with a resolution of 1.8 ° at 0.5 RPS. The cost and weight are substantially lower compared to the RPLIDAR system which weighs 200 grams with a price of € 294. Meanwhile the designed

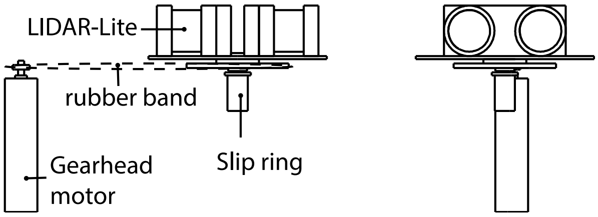


Figure 6.7: Impression of the proposed LIDAR system.

Table 6.8: Weight and cost overview laser scanner components

Item	Weight [g]	Cost [€]
LIDAR-Lite	2 x 12.0	2 x 53.00
Slip ring	2.4	*25.00
Platform plate	6.0	0.00[4]
Motor	24.0	20.00*
Wiring	*5.0	*2.00
IR sensor	0.5	0.40
Total	**61.9**	**147.40**

system is still able to come up with the measurement points within seconds to perform SLAM operations. Hence it is decided to go with the designed laser scanner for the SLAM process. However, once it has been produced it should be tested and verified to work the SLAM algorithm described in 7.3.2. This will also be noted in the recommendations.

For indoor SLAM a high angular resolution is suggested. By measuring with a frequency of 100 Hz at 0.5 RPS, the angle between measurements will be 1.8 °. A point cloud of 360 points per second will be produced. The maximum range will be 25 meters in any direction, hence a scan diameter of 50 meters.

6.5. Payload mounting structure

This section describes the structural design of the mounting platform for the electronics box and the camera; it is called the Vibration Damped Electronics and Camera Structure (VDECS). The whole structure consists of three parts; the vibration dampener, the electronics box and the camera mount. Due to the positioning constraints on the subsystems and the space needed for the battery stack, it was decided to put the camera and the electronics bay under the UAV. Inputs from the other subsystems, such as selected camera and electronics, allowed for the sizing of the structural elements.

The main structural calculations carried out were on the stress bearing capacity. More detailed analysis about the load distribution inside the structure, deflections of key elements and detailed vibrational analysis needs to be carried out in later phases.

6.5.1. Design process

The requirements for the VDECS are listed in the requirements section Appendix A. The assumptions and initial inputs, design procedure and results for the two iterations of the design have been described in below. The formula used to find the camera drop based on its span wise location can be seen in Eq. (6.4); in this equation d is the required camera distance from the center plate in the z direction, x is the spanwise camera position, θ is the pitch angle of the UAV and AoV is the smallest angle of view of the camera. The formula used to calculate the required thickness is (6.5); in this formula FoS is the factor of safety, σ_{ult} is the ultimate stress of HMCF, m is the mass of the system (camera or camera + electronics), g is the gravitational acceleration, D is the maximum drag acting on the UAV, r is the distance between the camera centroid and the bottom of the centre panel and l is the length of the connection at the root of the payload mounting. A safety factor of 3 is used during the calculations; even with the high safety factor, the required thickness calculated was , thus a value of 1 mm was used. The mass was approximated by calculating the area from the CATIA models and multiplying it with the thickness of the structural element.

The servo was selected in the first iteration; for this a number of COTS servos were analysed and a short list created by the required torque for controlling the camera. Besides the torque, minimizing the dimensions and mass and the possession of metal gear were used to further filter out the list of possibilities. The required torque was calculated by assuming a force of 5 N acting at the tip of the camera; this results in a required torque of 1.7 kgcm from the servo for control (since te arm is about 25 mm). It was also decided to use a servo with metal gears for durability. A final selection was made for the Futaba S3156 [92] servo due to its torque capacity of 2 kgcm, lightweight design of 9.3 g and its usage of metal gears.

$$d = x \tan\theta + \frac{AoV}{2} \tag{6.4}$$

$$t = \frac{FoS}{\sigma_{\text{ult}}}\left(\frac{mg}{l} + \frac{Dr(0.5l)}{\frac{1}{12}l^3}\right) \tag{6.5}$$

The following text gives an outline of the assumptions, assumed values, design procedure and results of the two iterations.

First iteration:

First the assumptions and the assumed values are listed; than the outline of the design procedure given and finally the results are summed.

Assumptions

- Only camera is part of the payload.
- Only the camera mass and the maximum forward accelerations are considered in structural sizing.
- The camera should have tilt freedom.
- The main camera should have forward vision when cruising.
- The drag acts though the centroid of the camera.
- The material used for the HMCF (high modulus carbon fiber, Ultimate strength of 150 Mpa).
- No vibration damping considered.

Assumed Values

- Thickness = 2 mm
- Maximum drag force on Payload = 3 N
- Minimum forward vertical field of view of camera 25° (known)
- Camera mass = 75 g
- Cruise $\theta = 3°$
- UAV midpoint 25 cm from nose.
- The thickness of all the components will be assumed to be 2 mm.
- Camera mounted within 3 cm of the UAV midpoint).
- 2 mounting arms of 25 mm cross sectional length.

Outline of design procedure

1. First the camera drop from the bottom of the UAV was approximated by con-straining the requirement of having a good forward view with a forward displacement of 2 cm from the UAV midpoint;
2. Using this drop angle the maximum stress near the root are calculated and are checked against the failure stresses of the material. The bending stress due to the moment and the axial stress due to the weight of the components are accounted for.
3. A servo is selected for the tilting of the main camera.
4. This information is used to come up with an initial CATIA design for the structure.

Results

- A field of 28 degrees was opted for (Fig. 6.8).
- The Futaba S3156 servo was selected.
- Camera spanwise position 23 cm from rotor tip.
- Camera height 8 cm under the bottom plate.
- Required thickness 0.05 mm.
- Total structural mass 37 g

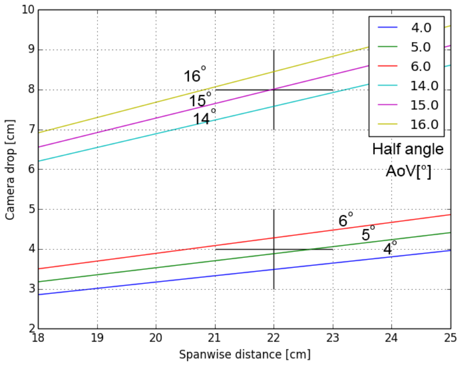


Figure 6.8: Camera vertical position against horizontal position

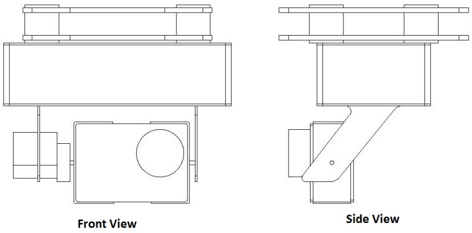


Figure 6.9: Sketch of the VDECS

SECOND ITERATION:

The second iteration was carried out with more information and a change in the configuration. This part first lists the changes to the assumptions, than the values used in the calculations and finally the results.

Changes to assumption

- Camera, internal electronics and the electronics box as are all defined as payload.
- The camera mass, the internal electronics, the electronics box and the maximum forward accelerations are considered in structural sizing.
- The electronics box and the camera are assumed as point masses in the camera c.g. location and the total drag is assumed to act through there as well
- The thickness of all the components will be assumed to be 1 mm.
- Vibration damping structures were considered;.

Used values

- Thickness of electronics box and camera supporting arms is 1 mm.
- Dimensions of the internal electronics for electronics box 8.6 cm x 5.4 cm x 3 cm.
- Maximum drag force on Payload = 5.75 N.
- Camera mass = 75 g
- Electronics mass = 65 g
- UAV midpoint 25 cm from nose.
- Camera mounted 2 cm in front of the UAV c.g.

Results

- A field of 28 degrees (Fig. 6.8).
- The Futaba S3156 servo was selected.
- Camera spanwise position 23 cm from rotor tip.
- Camera height 8 cm under the bottom plate.
- Required thickness 0.1 mm.
- total structural mass 82 g

The vibration damping structure consists of two carbon fibre plates one of which is rigidly connected to the UAV through screws and the other to the payload through screws. These two plates are connected using 4 rubber dampeners each with a carrying capacity of 120 g [93].

6.6. Recommendations and Conclusions

This sections describes the further work that needs to be done and some conclusions about the sensor, the on-board computer, the camera and the LIDAR systems.

6.6.1. Recommendations

The further work that needs to be performed to get a fully detailed design are discusses here. Firstly, a deeper and more physical experimentation based derivation of the requirement on the accuracy and the update of the sensors should be carried out. A great of work remains in fleshing out all the interfacing of the UAS; as it stands the details of communication between the different subsystems have only been described at a superficial level. The kind of connection made and the protocols used to send and receive data should be defined and checked to verify that the system does have the required interfacing capacities. The BUS hack of the Hero3+ also needs to be studied further; knowledge about electronics needs to be acquired to understand the details of programming the EEPROM used to interface with the camera. After sufficient knowledge has been gained, the hack needs to be validated by demonstrating that it is actually possible to get the live video stream out and control the desired camera settings. The VDECS needs to be optimised and the interfacing of the structural components fully defined; the attachment for the subcomponents in the VDECS has not been part of this design. A thorough study on the application of other types of sensors, such as IR attitude sensors, needs to be done.

The software for image stitching, terrain object detection needs to be either optimised or development from scratch using SimpleCV or OpenCV. The same goes for the optimization or the programming of the object detection and avoidance algorithms.

6.6.2. Conclusion

The final sensor choices were the NAVIO integrated sensor board for aerial applications, three Maxbotix LV-EZ0 sensors for proximity detection and the one MQ-2 gas sensor. The HummingBoard single board computer will act as the main computer for the UAV. The imaging system will consist of one higher resolution GoPro Hero3+ camera and a lower resolution Raspberry Pi camera module for on-board object detection. Finally a custom designed LIDAR system for point cloud generation. Figure 6.10 is a very top level sketch of the data flow in the UAS. For more detailed diagrams refer to the software or the hardware block diagrams (Fig. 11.2 and 11.3).

6.7. Data handling block diagram

Figure 6.10 shows the data handling block diagram between the main components of the UAS sytem.

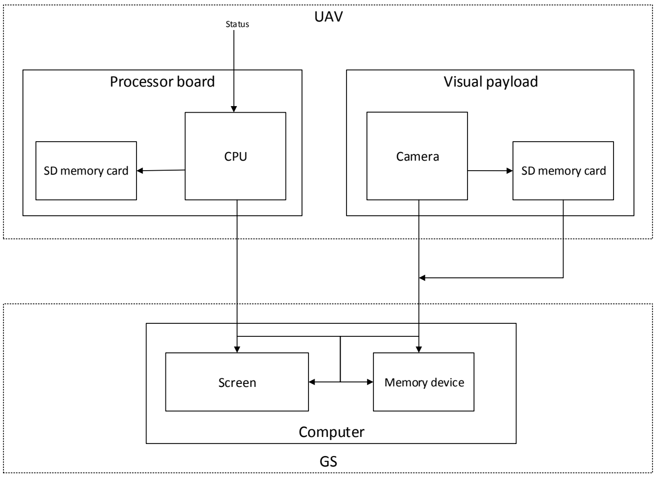


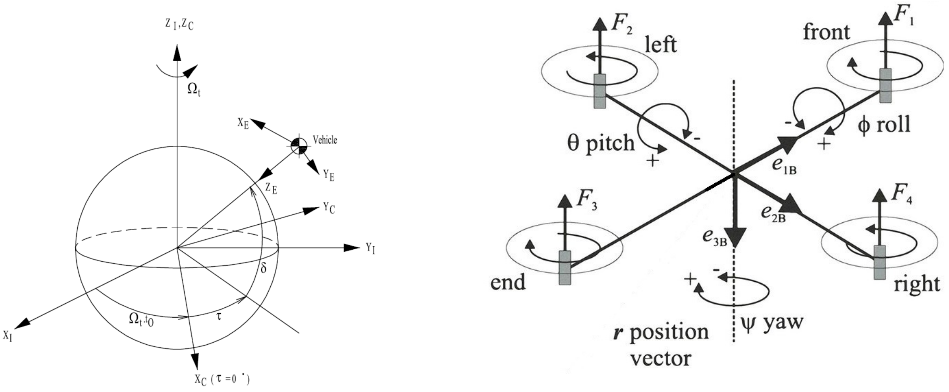
Figure 6.10: Data handling block diagram; This diagram shows how the data flows within the UAS

7 GNC SYSTEMS

Some of the sensors introduced in the last chapter are brought to life in this chapter, by using different kinds of software. The GNC, standing for Guidance, Navigation and Control, systems and their related on-board systems are introduced here. As part of the navigation system, a state-estimation software is developed and an outdoor navigation system based on GPS measurements is introduced. SLAM algorithms used for indoor navigation and guidance, and object avoidance software are presented in the guidance section, where also optimization of the guidance system is discussed. Control and simulation of the model are treated in the last section of this chapter. The first half of the section focuses on the control part and the equations of motion, while the remainder is dedicated to the simulation of the model.

7.1. REFERENCE FRAMES

Throughout this chapter, it is of utmost importance to differentiate the three reference systems that are used. Therefore, before delving into navigation, guidance and control the three frames are introduced: body-fixed reference, North-East-down (NED) also called vehicle carried reference frame and the Earth-centred, Earth-fixed frame (F_c). In Fig. 7.1 both vehicle carried frames are shown.



(a) Inertial (NED) frame [2]

(b) Body fixed frame [94]

Figure 7.1: Vehicle carried reference frames used in this document

The body-fixed reference frame is, as implied by its name, fixed to the frame of the quadcopter. Its corresponding axes are defined in Fig. 7.1b; e_{1_B}, e_{2_B} and e_{3_B} indicate the body-fixed x, y and z axes. Fig. 7.1a represents the North-East-down frame, with its x-axis pointing North, the y-axis pointing orthogonal to that towards East, and the z-axis completes the right-handed system, pointing towards the centre of the Earth. In order to use both systems they need to be related using the transformation matrix R given below [2]. R represents the transformation matrix form NED to body frame.

$$R = \begin{bmatrix} \cos(\theta)\cos(\psi) & \cos(\theta)\sin(\psi) & -\sin(\theta) \\ \sin(\phi)\sin(\theta)\cos(\psi) - \cos(\phi)\sin(\psi) & \sin(\phi)\sin(\theta)\sin(\psi) + \cos(\phi)\cos(\psi) & \sin(\phi)\cos(\theta) \\ \cos(\phi)\sin(\theta)\cos(\psi) + \sin(\phi)\sin(\psi) & \cos(\phi)\sin(\theta)\sin(\psi) - \sin(\phi)\cos(\psi) & \cos(\phi)\cos(\theta) \end{bmatrix} \tag{7.1}$$

In the R matrix the roll angle is represented by ϕ, θ represents the pitch angle and ψ represents the yaw angle. Since the body fixed frame is fixed to the quadcopter, ϕ, θ and ψ are the angles between the x-, y- and z-axes of the two reference frames, respectively. Finally, the navigation and guidance system, and possibly the pilot, need to be aware of the quadrotors position. Therefore, the Earth-centred frame (indicated by F_c) is considered. This inertial frame's origin coincides with the centre of the Earth as illustrated by Fig. 7.2. Its x-axis passes through the zero meridian and the equator. Normal to the x-axis, also in the equatorial plane the y-axis is located and the z-axis completes the right-handed system, passing through the international reference pole.

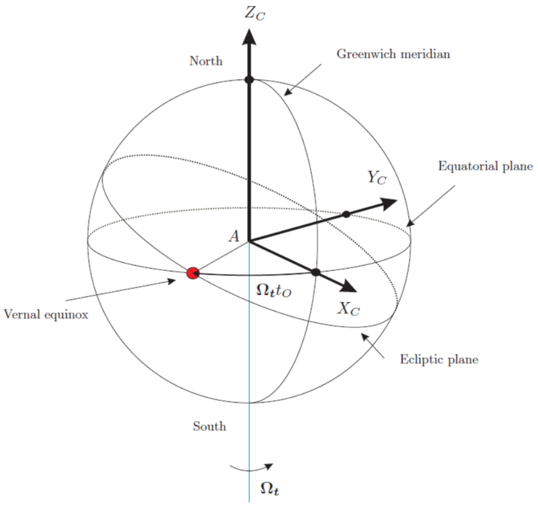


Figure 7.2: Inertial Earth-centred, Earth-fixed reference frame [2]

For the purpose of this transformation, it is assumed that the Earth is not rotating. For that reason, the rotation is not accounted for in the transformation matrix. This assumption is valid, since the flight velocity is low, and the mission duration is short [2]. The transformation matrix from F_c to NED is given by Eq. (7.2).

$$T = \begin{bmatrix} -\sin(\varphi)\cos(\lambda) & -\sin(\varphi)\sin(\lambda) & \cos(\varphi) \\ -\sin(\lambda) & \cos(\lambda) & 0 \\ -\cos(\varphi)\cos(\lambda) & -\cos(\varphi)\sin(\lambda) & -\sin(\varphi) \end{bmatrix} \tag{7.2}$$

λ represents the longitude and φ the latitude of the vehicle as explained in the following subsection. The inverse of matrix T gives the transformation from NED to F_c. Using the transformations given by Eqs. (7.1) and (7.2) the state estimate can be represented in an inertial frame.

7.2. Navigation

In this section the navigation system of the MAV is treated. Due to the operational requirements, navigation is required in both, indoor and outdoor environments. While outdoor navigation is fairly straightforward, where GPS satellites are in reach, indoor navigation requires a whole set of hardware and software. For that reason, this section is subdivided into two groups, first discussing outdoor then indoor navigation. Prior to that, the state-estimation methods used aboard the quadrotor are presented.

7.2.1. State-estimation

An essential part of guidance, navigation and control is the state-estimation of the aircraft. All estimates (the state-variables are discussed in Section 7.4) are based upon measurements from sensors such as the IMU and GPS. To improve the quality of the state-estimate, a Kalman-Filter is implemented. This filter brings two advantages as compared to only using IMU or GPS signals: high update frequency of the state-estimate provided by the IMU and a time independent error in the estimate given by the GPS. Even though there are more advanced, non-linear Kalman-Filters, due to time constraints only a linear Kalman-Filter is developed for this mission. There are two independent inputs required by the Kalman-Filter and an initial error estimate for each input. Furthermore, an initial state-estimate is required. Based on that information, a state-estimate is made by modelling the dynamics of the system. In case of an integrated IMU/GPS system, the quadrotors position is estimated by integrating the accelerations coming from the accelerometers twice. The here presented equations are taken from [95].

$$x_{est} = \int_0^t \int_0^t a\mathrm{dtdt} = \frac{1}{2}at^2 + vt + x \tag{7.3}$$

Where a represents the measured acceleration, t the time since start of measurement and v and x are the velocity and position respectively. In matrix form and discrete time steps dt this becomes:

$$\tilde{X}_{est} = \begin{bmatrix} x_{est,t} \\ v_{est,t} \end{bmatrix} = \begin{bmatrix} 1 & dt \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_{est,t-1} \\ v_{est,t-1} \end{bmatrix} + \begin{bmatrix} \frac{dt^2}{2} \\ dt \end{bmatrix} \begin{bmatrix} a \end{bmatrix} = AX_{est,t-1} + Ba \tag{7.4}$$

Next, the covariance matrix is estimated. This is only dependent on the time growing error in the acceleration signal.

$$\tilde{\Sigma}_t = A\Sigma_{t-1}A^T + E_x \tag{7.5}$$

With $E_x = \begin{bmatrix} \sigma_p^2 & \sigma_{p,v} \\ \sigma_{v,p} & \sigma_v^2 \end{bmatrix}$ where $\sigma_p = dt^2/2$ represents the standard deviation of the position, $\sigma_v = dt$ the standard deviation of the velocity and $\sigma_{p,v} = \sigma_{v,p} = dt^3/2$ is the product of the deviation of position and velocity. Therefore, $E_x = \begin{bmatrix} dt^4/4 & dt^3/2 \\ dt^3/2 & dt^2 \end{bmatrix}$. Now, the Kalman gain is calculated, which takes the errors of both position estimates into account, and depending on that weighs one of the estimates higher. Mathematically, the Kalman gain is calculated as depicted in Eq. (7.6).

$$K = \tilde{\Sigma}_t C^T \left(C\Sigma_t C^T + E_z\right) \tag{7.6}$$

In Eq. (7.6) C is a row vector $\begin{bmatrix} 1 & 0 \end{bmatrix}$ and E_z is the measurement error of the GPS squared - $E_z = \sigma_{GPS}^2$. As final two steps, the Kalman state-prediction is made and the covariance matrix is updated.

$$X_{est,t} = \tilde{X}_{est,t} + K\left(z_t - C\tilde{X}_{est,t}\right) \tag{7.7}$$

$$\Sigma_t = (I - KC)\,\tilde{\Sigma}_t \tag{7.8}$$

In the above equations, I is a 2x2 identity matrix and z_t the measured GPS location. Based on equations (7.4) to (7.8) a linear Kalman-Filter is developed. Therefore, a GPS signal with an error of 4 m (error of actual sensor) and acceleration measurements with an error of 20 Gs. The latter is extremely exaggerated for verification and visualization purposes. Both signals are assumed to be normally distributed with a standard deviation equal to the respective error. For the simulation, a free falling object is assumed, for which the equation $x = \frac{1}{2}at^2 + vt + x_{initial}$ holds. The simulation is shown in Fig. 7.3.

Fig. 7.3 illustrates the power of the Kalman-Filter. The estimated position almost entirely overlaps with the actual position.

Similarly to position, the Euler angles have to be estimated, as the data provided by the gyroscopes has a time dependent error similar to that of the accelerometers, and additional uncertainties due to structural vibrations. However, there is no external estimate available for the attitude, that can be used as a secondary input to the Kalman-Filter. The second input has therefore to be obtained indirectly by determining the attitude based on velocity. For roll and pitch this approach is fairly straight forward, as a certain velocity in x or y direction (in the body-fixed frame) is resulting from a certain pitch or roll angle and total thrust. Yaw angle estimation must be done separately, using data from the guidance system. Since the quadrotor is designed such that the camera is pointing in flight direction, the yaw angle can be determined from flight direction.

This part has not been mathematically developed yet, as the guidance system is not covered yet and the simulation only provides valuable data for hovering conditions. Furthermore the state-space model does not contain any accurate estimates of aerodynamic properties such as drag other than for forward flight, as no Computational Fluid Dynamics (CFD) analysis has been performed on all motions of the quadcopter. This results in pitch and roll angles for a certain velocity not being accurate.

7.2.2. Outdoor navigation

On the hardware side of the navigation system, there are Global Positioning System (GPS) sensors, as well as the Inertial Measurement Unit (IMU) which consists of three axis accelerometers - mounted along the body axis - and three axis gyroscopes, which measure the rotational speed of the quadrotor about each of its body axes. Since GPS determines the quadcopters position in terms of longitude and latitude, conversion to x, y and z (Cartesian) coordinates in the inertial frame is necessary.

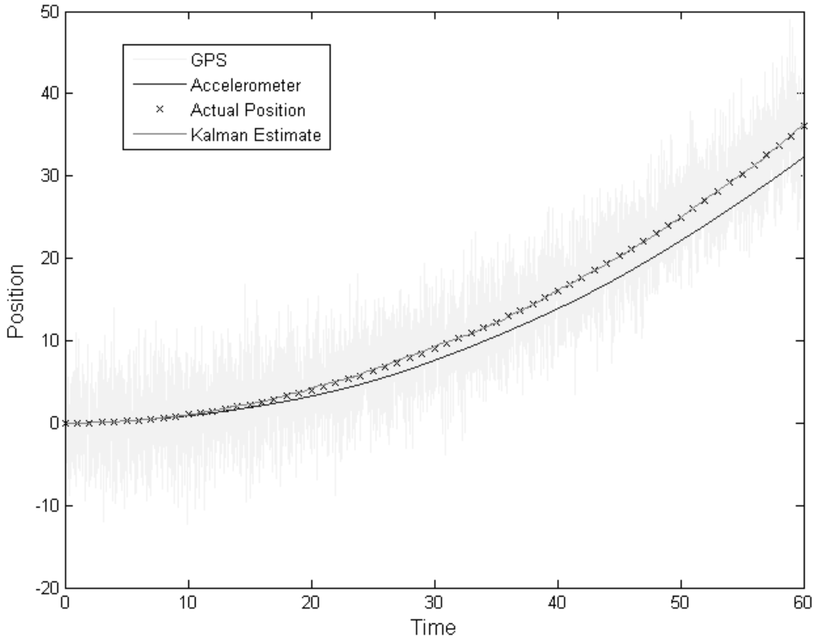


Figure 7.3: Simulation of the developed Kalman-Filter

$$x = R\cos(\varphi)\cos(\lambda) \tag{7.9}$$
$$y = R\cos(\varphi)\sin(\lambda) \tag{7.10}$$
$$z = R\sin(\varphi) \tag{7.11}$$

R is the radius of the Earth. φ stands for latitude - angle between equatorial plane and North (or South) position of the vehicle. Similarly, λ represents the longitudinal location (angle between East (or West)) of the vehicle. The origin of this Cartesian coordinate system is the Earth's centre, corresponding to the Earth-centred, Earth-fixed frame.

These coordinates are, however, in the inertial frame, whilst the accelerometers measure any excitations in the body frame. Therefore, the position obtained from integrating acceleration measurements are all in the body frame, and need to be converted to the NED-frame using Eqs. (7.13) and (7.14) and then to the inertial frame using Eq. (7.2).

$$X^b = RX^{NED} \tag{7.12}$$
$$X^{NED} = R^{-1}X^b \tag{7.13}$$
$$X^{F_c} = T^{-1}X^{NED} \tag{7.14}$$

The superscripts b and NED indicate the Cartesian coordinates in body and inertial frame, respectively. R^{-1} is the inverse of the transformation matrix given in Eq. (7.1). All angles required for the transformation matrix are obtained by integrating the measurements from the gyroscopes.

7.3. Guidance

The guidance system is responsible for the quadrotor's trajectory and its velocity. Furthermore, this system controls take-off and ascend as well as descend and landing. Another essential part of the guidance system is the object avoidance algorithm, which is discussed in more detail in Section 7.3.1. Due to constraints on time the development of this system is left for the future. This section focuses on the description of the guidance system, and its interaction with navigation and control of the quadrotor in more detail than described in Section 7.7.

Planning the velocity and trajectory requires knowledge about the current state of the MAV and the final state it should attain. Information on the final state is delivered by the operator in real emergency situations, for the IMAV itself the navigation system is going to deliver it. The vehicle's state in-between the initial and final state are determined in such a way that energy use is optimised, so that sustainability is also becoming a part of the control system. Since the efficiency of the quadrotor is directly related to its flight velocity, the cost function Eq. (7.15) is to be minimized [96].

$$\Phi = \frac{1}{T}\int_0^T \left(P_1\dot{x}^2 + P_2\dot{y}^2 + P_3\dot{z}^2\right)^{\frac{1}{2}} dt \tag{7.15}$$

Where $\dot{x}$, $\dot{y}$ and $\dot{z}$ are the velocities in x, y and z direction in the body frame, and P_i are weighing factors.

However, the guidance system also has to be aware of the dynamics of the system, its environment and potential constraint. Constraints specifically applying to the IMAV 2014 are maximum altitude of 950 ft and the mission area, within which it has to remain at any time. Depending on these, and the maximum allowable velocity, the guidance system can plan both trajectory and flight velocity.

The trajectory is going to be a set of 3-dimensional way points - set in the inertial frame - along which the MAV flies. Based on the current state of the vehicle and the location of the next way point, the guidance system sends commands to the controller, which contain change in velocity ($\Delta\dot{x}$, $\Delta\dot{y}$ and $\Delta\dot{z}$). A change in heading ($\Delta\psi$) results from the constraint camera movement, which sends a live stream to the ground station. Therefore, the MAV has to change its yaw angle to have to camera point in flight direction. Throughout the path and velocity planning, the guidance system is working in the inertial F_c frame. However, all commands given to the engines from the controller result in motions in the body frame. For that reason, the velocity increments are converted to the body frame, and then forwarded to the controller.

7.3.1. Object detection & avoidance

Object detection and avoidance plays a crucial role in the guidance system. The difficulty in the design of this system arises from bypassing not only static objects such as buildings or trees but also moving object as birds or other UAVs.

Low altitude outdoor flight

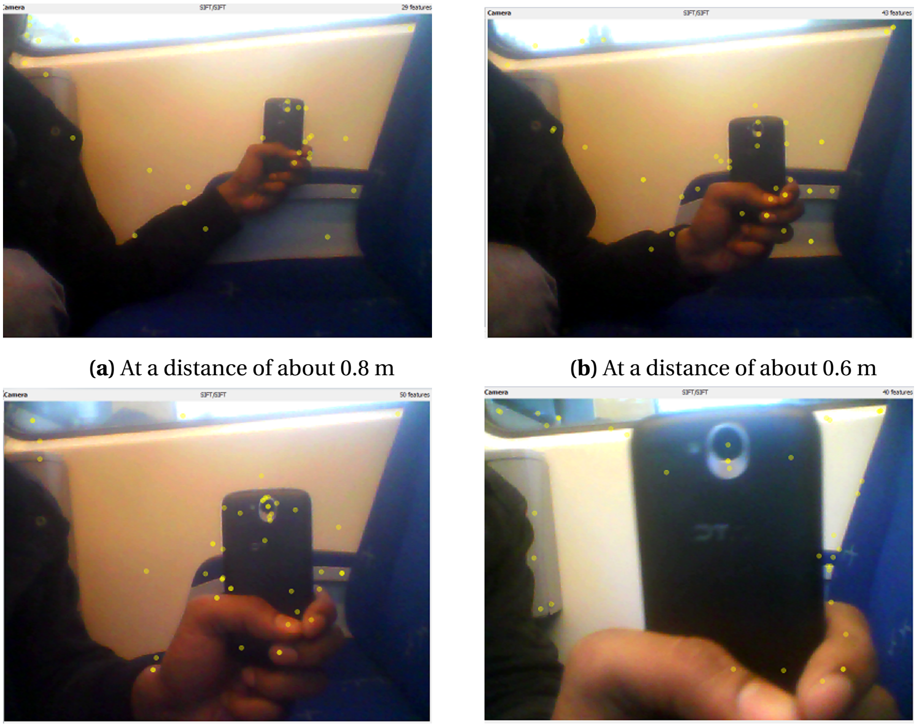
The requirements as defined for the camera selection in Section 6.3.1, also apply to the algorithm selection. Due to the definition of these requirements after the first iteration, a data flow similar to the one presented in the conclusion is assumed for this design stage of the imaging subsystem. Thus for object detection and avoidance, the on board processing time as well as the off board processing time (accounting for the communication lag) is also assumed. It is assumed that the on board and off board systems will run similar image processing software; only that the on board computer will be doing image processing to a lesser depth and using better quality images, while the off board (ground station) will be doing more intensive analysis on images of lower resolution. A 0.5 second processing time is assumed for both on board and off board image processing based on application of fast MOPS and SIFT as demonstrated by the work of Jeong-Oog Lee et al in [97]. In another paper, not published in a peer reviewed journal but available online by Alvarez et al, it was demonstrated that a modified Parrot AR Drone could detect obstacles and plot a safe path based on just monocular object detection system with the update frequency of 1 hz and the IMU data [98]. Therefore a processing time of 0.5 seconds is considered.

A required update rate of 2 Hz is defined based on the findings of the paper [97]. In that paper a 2 Hz update rate is sustained in simulation using a Intel Core 2 Quad 2.4 GHz CPU. With the assumed acceleration and velocities the time left for object recognition and actuation is 2.2 seconds outdoor and 4 seconds indoor. Since these are idealized calculations a safety factor of two is imposed on the requirement for the time between detection and actuation; therefore a value of 1.1 second is used in the requirement **Req-pl-imag-6-3**.

The verification of the required update frequency is carried out using the Find-Object program, previously discussed in the camera selection section. The software was used to run object detection using the SIFT algorithm over a distance of approximately one metre on a mobile phone (Fig. 7.4). Also as mentioned in the previous many of the parameters for object detection can be tuned with the program to get the desired result. By keeping the number of features at 50 it was found the system described in [99] can easily sustain an update rate of 2 hz for object detection. It can also be seen in Fig. 7.4 that the program successfully tracks the outline of the mobile phone as it is moved from a distance of 0.8 m to about 0.2 m in front of the camera. This is believed to have demonstrated the fulfilment of the update rate requirement from the imaging system as defined **Req-pl-imag-6**.

7.3.2. Indoor flight

While the hardware for the SLAM process is designed, the software is still missing. That part will be covered in this section. Over the past years there has been a lot of research in developing accurate SLAM algorithms,



(a) At a distance of about 0.8 m

(b) At a distance of about 0.6 m

(c) At a distance of about 0.4 m

(d) At a disntace of about 0.2 m

Figure 7.4: Fast-Object program tracking the contour of a mobile phone at an update rate of 2 hz; the SIFT algorithm was used for feature detection and description and the RANSAC algorithm was used to generate homographies

like [100] and [101]. However these algorithms had a main focus on ground based vehicles. Recently there have been results of SLAM on MAVs: [102] and [103]. However, the first one tested their algorithm on ground robots, while the second made assumptions on the environment making it less suitable for unknown environments. [104] developed a fully autonomous quadrotor that relies only on on-board sensors for stable control without requiring prior maps or assumptions of the environment. Two years later they came with extended article about this algorithm, presented in [3]. This algorithm will be implemented on the MAV for indoor navigation. Their quadcopter system is capable of autonomous flight in unstructured and unknown GPS-denied environments. Developing this capability required careful engineering of a complex system that leverages existing algorithms to balance the trade-offs imposed by GPS-denied flight. They also describe experimental assessments of first using on-board sensors to estimate the vehicle's position and secondly using the same sensor data to build a map of the environment around the vehicle. They noted that their 1.6 GHz Atom processor was powerful enough to run the on-board algorithm. Running it on the Hummingboard should therefore be no problem. This indoor navigation process has been chosen because of the following reasons:

- **Optimised algorithms**
  For the various element processes, either multiple algorithms have been tested and the one with the best results used or well known algorithms have been used and improved for optimal performance.

- **Extensively tested**
  The researchers have performed over 100 flight hours with their MAV to test and improve the algorithms.

- **Won competition with similar characteristics**
  The system presented was used in the winning entry as Team MIT-Ascending Technologies in the 2009 International Aerial Robotics competition (IARC). In this competition the MAV had to explore and navigate autonomously through the interior of a power plant set-up by artificial walls. The main goal was to find potential visual targets, in the form of a gauge panel printed on a paper sheet. This is highly similar to the IMAV competition where the MAV has to explore and navigate autonomously through buildings to find visual targets in the form of survivors.

- **Modular design**

The presented indoor navigation process has been designed and built in a modular fashion. This means that the separate algorithms can be replaced/improved easily, when better performing ones come available.

- **Works in urban outdoor environments**
  The process has been tested outdoors in an urban environment. It was able to ignore the non-matching map scans from the trees, while locking onto the vertical walls of the buildings. This feature could be useful for future work.

The following adaptations should be made to the process as presented by [3] to make it more suitable for the MAV competing in the IMAV challenge:

- Use the downward oriented ultrasonic sensor for height measurements rather than deflecting some laser scanner beams down with a right-angled mirror. The rest of the height estimation process can remain the same.

System overview

Bachrach et al. designed a 3-level sensing and control hierarchy which can be seen in Fig. 7.5 grouped by colour. The green and yellow layer run on-board the MAV in real time. The green layer, or the on-board controller stabilises the attitude at a rate of 1 kHz making use of the IMU, as described in section 7. The other process can control the vehicle by interacting with this control loop.

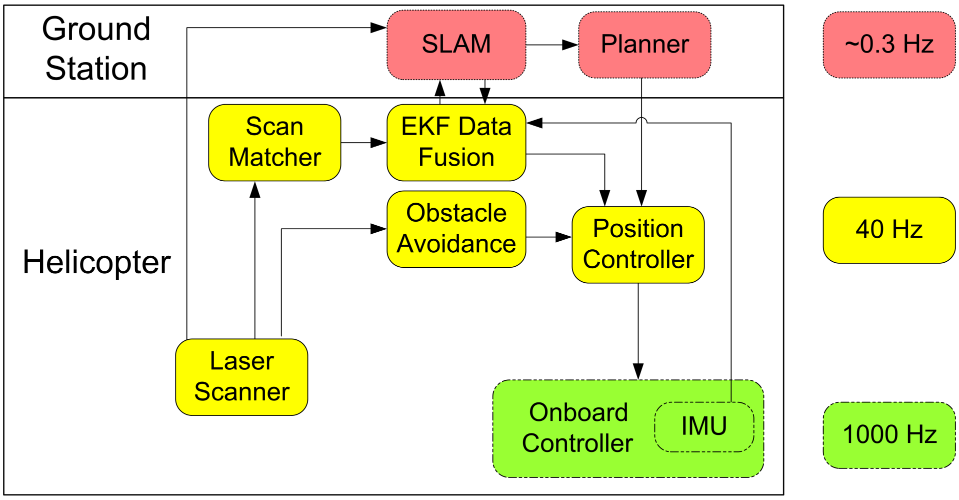


Figure 7.5: 3-level sensing and control hierarchy for indoor navigation [3]

The yellow layer performs low-level obstacle avoidance. A fast and high resolution laser scan matching algorithm is used to determine the vehicles relative motion. Together with the IMU measurements this is fed into an extended Kalman filter-based data fusion module which provides an accurate, high frequency vehicle state estimate. These enable the position controller to position the MAV. An obstacle avoidance module will hold a minimum distance between the MAV and observed obstacles.

The red modules, running on the groundstation, provides the high-level mapping and planning functionalities. The state estimates from the date fusion filter are fed into the SLAM algorithm together with the laser scans. This creates a global map, which ensures globally consistent state estimates. Bachrach et al. state that it takes 1-2 seconds to subjoin the newly received scans, and it is therefore not part of the real-time feedback control loops at the lower levels. However, the real-time state estimates are kept globally consistent by providing the delayed correction signals to the data fusion module. Finally, with the planning and exploration module the vehicle can plan paths within the map generated by the SLAM module. This will guide the vehicle towards unexplored regions or other desired locations.

The system modules from the hierarchical software system shown in Fig. 7.5 are built as a set of independent processes which communicate using the Lightweight Communications and Marshalling (LCM) library[1] [105].

Algorithm optimisation for the IMAV competition

This section presents options to optimise the indoor navigation process specifically for the IMAV competition.

[1] https://code.google.com/p/lcm/

- **Set maximum object length in the height determination algorithm**
  The height determination algorithm makes use of the measured ground distance. When flying over an object, like furniture, this measurement gives a false reading of the distance to the real ground. To account for this, it contains a part that will reject these outliers over a certain flight distance. This distance can be set to (maximum) furniture lengths that are common inside buildings.

- **Set optimum percentage to add contours to local map**
  Contours are added to the local map when the fraction of points in an incoming scan that are given a high likelihood in the current map drops below a threshold[2]. The threshold should be set high enough that incoming scans are still able to be matched accurately, but low enough that scans are not added too often. They experimentally determined that a threshold of 75 % gave a good trade-off, adding a scan to the map approximately every 2 seconds during normal operation. However, the buildings of the IMAV competition are a lot smaller compared to the MIT campus building it was tested in. Therefore this threshold should be redetermined for an optimum value for this different kind of buildings.

Recommendations

In this section some indoor navigation specific recommendations will be done for future work.

- **Test the laser scanner with the proposed indoor navigation process**
  Since the indoor navigation process proposed is tested with a high quality off-the-shelf laser scanner, it should be tested with the self-designed laser scanner. This because the latter has a lower update frequency for a full scan.

- **Sensor integration**
  Since the MAV has also cameras on-board, a possibility is to include this in the SLAM process. This would require a monocular vision based SLAM algorithm, which can then be integrated with e laser scanner SLAM data. One of the main benefits of this system would be that visual information can be added to the map. This could be beneficial for emergency service workers when they have to navigate through the building.

- **Window recognition**
  The laser beams of the laser scanner will scan through windows rather deflecting back. This means that windows are not "seen" as an object, with potential accidents as a result. This is something that will have to be looked into. A possible option could be to use the ultrasonic sensors before flying through a window, to test if there is glass.

- **Exit route**
  The SLAM and planning processes are performed on the ground station. However it is possible that the link between the MAV and the ground station is lost. Therefore its recommended that with the path to unexplored areas also an exit path is provided. This way the MAV can leave the building and fly back to predetermined GPS coordinates in case of the connection is lost. Please note that this is not allowed for the IMAV competition in which is has to land immediately in case of a connection loss. This is therefore a feature for beyond the competition.

7.4. Control

Quadrotors are inherently unstable, making an on- or off-board autopilot a necessity to stabilize the vehicle and maintain a desired flight condition. The autopilot is the location where the data related to the attitude and position measured by different sensors and resulting state-estimations come together. This information is then processed and commands are returned to each of the four engines. The main part of the required calculations is done using an on-board computer. Only more accurate calculations on the sate of the MAV based on vision based information are done off-board, and fed back to the MAV with a time delay to correct for errors in the attitude estimate.

7.4.1. Operational decisions

Before delving into controller, guidance and navigation design, some decisions are to be made on the desired flight behaviour of the aerial vehicle.

As a first top level requirement to the control systems comes the heading angle. Due to its design, a quadrotor can fly in any direction without having to yaw. For this design however, a live stream is send to the

[2]

ground station. The camera used for that is pointing in flight direction and therefore requires the quadcopter to fly with this camera pointing in flight direction.

The next design decision is of particular importance for restricted areas where the quadrotor is allowed to fly, such as the IMAV competition. For mission element A - mapping the area - the aircraft has to fly multiple times over the area to capture everything with an adequate resolution (see Sec. 3.3). Turning manoeuvres, however, are to be made on a very confined space, as even the 'corners' of the area have to be mapped but at the same time the vehicle cannot fly outside this regulated airspace. For that reason, and the fact that the camera used for mapping can only rotate along the vehicles pitch axis, the quadrotor will have to come to a complete stop at the end of the area and accelerate again in the opposite direction as illustrated for purposes of clarity only by Fig. 7.6.

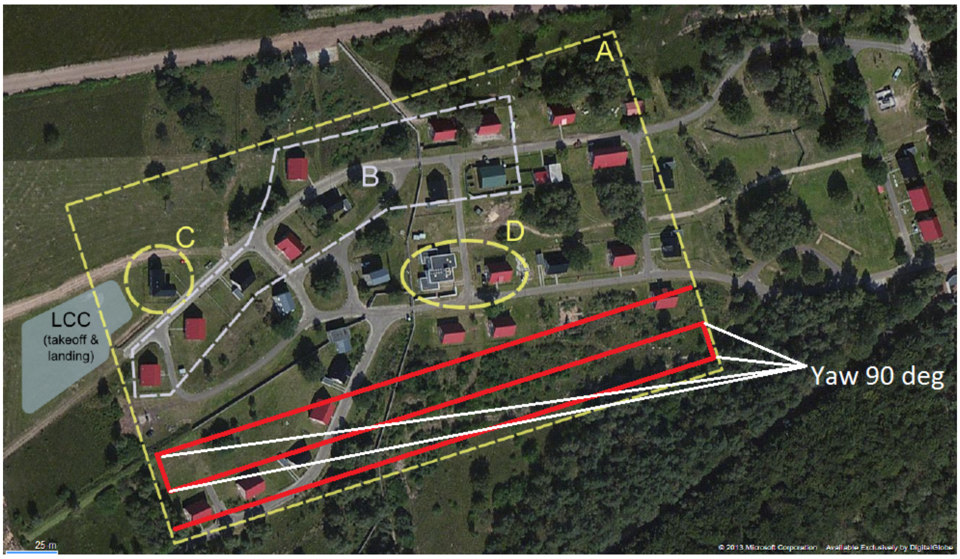

A
B
C
D
LCC
(takeoff & landing)
Yaw 90 deg
25 m
© 2013 Microsoft Corporation Available Exclusively by DigitalGlobe

Figure 7.6: Turning manoeuvres during mission element A

This is done despite a round turn initiated by combined roll and pitch motion being more efficient, for the sole purpose of not exiting the yellow box.

Important to note on fig. 7.6 is that it does not represent the exact path the quadrotor follows during mission element A. This is only to illustrate the turning manoeuvres.

7.4.2. Manoeuvres

Before developing a controller for any vehicle, it is important to determine what kind of motions are to be actively controlled. In case of a quadcopter, six degrees of freedom have to be managed. This section describes how all motions are controlled.

Pitch and roll

Because the MAV has to be sufficiently agile, especially inside buildings, pitch and roll must be adequately controllable. The behaviour of these angles is the most important motion to look at. This is because these angles cause large movements, because the thrust direction is changed making the quad-copter accelerate. When a pitch command comes in, the fourth engine must generate more thrust, whereas the second engine less thrust. For a roll command, the first engine should immediately generate more thrust, and the third less. Later it will be explained why a feedback loop in the control scheme is necessary.

Height

As the MAV should map the area, it must be able to reach a certain altitude, and stay there. In order to do so, the height must be controllable to a certain degree of precession and within a response time. When the operator commands the MAV to go up, all four engines must increase their RPM (the absolute value) and thereby the thrust. When the mass of the MAV is known, the equations of motion can be established by using Newton's second law of motion: force equals mass times acceleration. On top of that the current energy level in the battery also has influence on the RPM adjustments of the rotor. Therefore a controller must be made that converts a combination of current charge level in the battery and command sequence, into a change in voltage levels towards the engines.

Yaw angle

Since the body x-direction is the direction of flight, and the camera always points in this direction, the yaw motion must be controllable. This is done by making use of the physical law of conservation of angular momentum. If there is a command to perform a yaw motion, then all engines have to increase their rotational velocities in the same direction. That means in the absolute sense that two turn faster, and the other two turn slower. As explained before, two rotors that oppose each other in fixed position spin clockwise and the two others spin counter-clockwise. If all rotors' angular velocities are increased in the same direction, that means that all of them spin faster counter-clockwis. The increased angular momentum resulting from the rotors must be compensated according to this law. This compensation is a clockwise motion of the UAV. When this yaw manoeuvre is performed, there will not be an increase in lift since the cumulative RPM, and therefore the thrust, stays the same.The drone will not experience a change in pitch or roll angle either. Two opposing rotors generate equal thrust, hence no moment about the either the x or the y axis. Only a bending moment will be present which will be exerted on both axes.

Control allocation

Now that the use of three different types of angular motions is explained, a problem arises when multiple commands are being handled at the same time. A decision must be made resulting from a possible event which may take place when certain angular velocities during a manoeuvre had been reached. When a roll or pitch motions is performed, then the maximum RPM value that the autopilot could command is 80 % of the total maximum RPM that the engine can deliver. In that way there always is some room left when other commands come in simultaneously, thereby allowing for controllability in such cases as well.

7.4.3. Equations of motion & simulation

This section discusses the controller design and begins with the derivation of the linear equations of motion. A simulation is then built using these equations of motion to determine and visualize the behaviour of the system to different inputs. Based on that, the gains are tuned preliminarily.

Before developing the controller itself, the authors decided upon using Euler angles rather than quaternions or other state representations despite the disadvantages linked to Euler angles. There are two reasons for that decision: the knowledge about Euler angle representation of the authors exceeds their knowledge about quaternions and secondly, it is assumed that the aircraft does not fly any aggressive or aerobatic manoeuvres for which singularities occur in the Euler angle representation. Furthermore, literature available on Euler angles is more accessible making it easier to gather the required knowledge for the controller design.

Equations of motion & state-space representation

From literature, the full Equations Of Motion (EOM) of a quadrotor are obtained [106]:

$$\ddot{x} = -\frac{\sin(\theta)\cos(\phi)}{m}u_1 \tag{7.16}$$

$$\ddot{y} = \frac{\sin(\theta)}{m}u_1 \tag{7.17}$$

$$\ddot{z} = -\frac{\cos(\theta)\cos(\phi)}{m}u_1 + g \tag{7.18}$$

$$\ddot{\phi} = -\dot{\psi}\dot{\theta}\cos(\phi) + \frac{\cos(\psi)}{I_{xx}}u_2 - \frac{\sin(\psi)}{I_{xx}}u_3 + \frac{I_{yy}-I_{zz}}{I_{xx}}\left(\dot{\psi} - \dot{\theta}\sin(\phi)\right)\dot{\theta}\cos(\phi) \tag{7.19}$$

$$\ddot{\theta} = \frac{\dot{\psi}\dot{\phi}}{\cos(\phi)} + \dot{\phi}\dot{\theta}\tan(\phi) + \frac{\sin(\psi)}{\cos(\phi)I_{xx}}u_2 + \frac{\cos(\psi)}{\cos(\phi)I_{yy}}u_3 - \frac{I_{yy}-I_{zz}}{I_{xx}}\left(\dot{\psi} - \dot{\theta}\sin(\phi)\right)\frac{\dot{\phi}}{\cos(\phi)} \tag{7.20}$$

$$\ddot{\psi} = \dot{\phi}\dot{\psi}\,tan(\phi) + \frac{\dot{\phi}\dot{\theta}}{\cos(\phi)} + \frac{\sin(\psi)\tan(\phi)}{I_{xx}}u_2 + \frac{\cos(\psi)\tan(\phi)}{I_{yy}}u_3 + \frac{1}{I_{zz}}u_4 - \frac{I_{yy}-I_{zz}}{I_{xx}}\left(\dot{\psi} - \dot{\theta}\sin(\phi)\right)\dot{\phi}\tan(\phi) \tag{7.21}$$

Equations (7.16) to (7.21) are now linearised. The mathematics are simpler and more advanced for linearised systems, and tuning gains is significantly less time consuming and less complicated for linearised systems. Therefore the EOM are linearised about the hover conditions and rewritten in state-space format of form:

$$\dot{x} = Ax + Bu \qquad \rightarrow \Delta\dot{x} = A\Delta\dot{x} + B\Delta\dot{u} \tag{7.22}$$

$$y = Cx + Du \qquad \rightarrow \Delta y = C\Delta x + D\Delta u \tag{7.23}$$

With A being the state matrix, B the input matrix, C representing the output matrix and D is the throughput matrix. The vectors x and u are the state vector and input vector respectively. Δ represents the deviation from the initial state in the linearised system. An example is $\Delta x = x - x_0$ with x_0 being the initial condition about which the system is linearised, and x the current state of the vehicle. The state-variables are taken as proposed by [94].

$$x^T = \begin{bmatrix}\Delta x & \Delta y & \Delta z & \Delta\phi & \Delta\theta & \Delta\psi & \Delta\dot{x} & \Delta\dot{y} & \Delta\dot{z} & \Delta\dot{\phi} & \Delta\dot{\theta} & \Delta\dot{\psi}\end{bmatrix} \tag{7.24}$$

$$\begin{bmatrix}\Delta\dot{x}\\ \Delta\dot{y}\\ \Delta\dot{z}\\ \Delta\dot{\phi}\\ \Delta\dot{\theta}\\ \Delta\dot{\psi}\\ \Delta\ddot{x}\\ \Delta\ddot{y}\\ \Delta\ddot{z}\\ \Delta\ddot{\phi}\\ \Delta\ddot{\theta}\\ \Delta\ddot{\psi}\end{bmatrix} = \begin{bmatrix}0&0&0&0&0&0&1&0&0&0&0&0\\ 0&0&0&0&0&0&0&1&0&0&0&0\\ 0&0&0&0&0&0&0&0&1&0&0&0\\ 0&0&0&0&0&0&0&0&0&1&0&0\\ 0&0&0&0&0&0&0&0&0&0&1&0\\ 0&0&0&0&0&0&0&0&0&0&0&1\\ 0&0&0&0&\frac{u_{1_0}}{m}&0&0&0&0&0&0&0\\ 0&0&0&\frac{u_{1_0}}{m}&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&0&0&0&0\end{bmatrix} \begin{bmatrix}\Delta x\\ \Delta y\\ \Delta z\\ \Delta\phi\\ \Delta\theta\\ \Delta\psi\\ \Delta\dot{x}\\ \Delta\dot{y}\\ \Delta\dot{z}\\ \Delta\dot{\phi}\\ \Delta\dot{\theta}\\ \Delta\dot{\psi}\end{bmatrix} + \begin{bmatrix}0&0&0&0\\ 0&0&0&0\\ 0&0&0&0\\ 0&0&0&0\\ 0&0&0&0\\ 0&0&0&0\\ 0&0&0&0\\ 0&0&0&0\\ \frac{1}{m}&0&0&0\\ 0&\frac{cos(\psi_0)}{I_{xx}}&-\frac{sin(\psi_0)}{I_{xx}}&0\\ 0&\frac{sin(\psi_0)}{I_{xx}}&\frac{cos(\psi_0)}{I_{yy}}&0\\ 0&0&0&\frac{1}{I_{zz}}\end{bmatrix} \begin{bmatrix}\Delta u_1\\ \Delta u_2\\ \Delta u_3\\ \Delta u_4\end{bmatrix} \tag{7.25}$$

$$\begin{bmatrix}\Delta x\\ \Delta y\\ \Delta z\\ \Delta\phi\\ \Delta\theta\\ \Delta\psi\\ \Delta\dot{x}\\ \Delta\dot{y}\\ \Delta\dot{z}\\ \Delta\dot{\phi}\\ \Delta\dot{\theta}\\ \Delta\dot{\psi}\end{bmatrix} = \begin{bmatrix}1&0&0&0&0&0&0&0&0&0&0&0\\ 0&1&0&0&0&0&0&0&0&0&0&0\\ 0&0&1&0&0&0&0&0&0&0&0&0\\ 0&0&0&1&0&0&0&0&0&0&0&0\\ 0&0&0&0&1&0&0&0&0&0&0&0\\ 0&0&0&0&0&1&0&0&0&0&0&0\\ 0&0&0&0&0&0&1&0&0&0&0&0\\ 0&0&0&0&0&0&0&1&0&0&0&0\\ 0&0&0&0&0&0&0&0&1&0&0&0\\ 0&0&0&0&0&0&0&0&0&1&0&0\\ 0&0&0&0&0&0&0&0&0&0&1&0\\ 0&0&0&0&0&0&0&0&0&0&0&1\end{bmatrix} \begin{bmatrix}\Delta x\\ \Delta y\\ \Delta z\\ \Delta\phi\\ \Delta\theta\\ \Delta\psi\\ \Delta\dot{x}\\ \Delta\dot{y}\\ \Delta\dot{z}\\ \Delta\dot{\phi}\\ \Delta\dot{\theta}\\ \Delta\dot{\psi}\end{bmatrix} + \begin{bmatrix}0&0&0&0\\ 0&0&0&0\\ 0&0&0&0\\ 0&0&0&0\\ 0&0&0&0\\ 0&0&0&0\\ 0&0&0&0\\ 0&0&0&0\\ 0&0&0&0\\ 0&0&0&0\\ 0&0&0&0\\ 0&0&0&0\end{bmatrix} \begin{bmatrix}\Delta u_1\\ \Delta u_2\\ \Delta u_3\\ \Delta u_4\end{bmatrix} \tag{7.26}$$

For the linear equations of motion the initial conditions presented in Table 7.1 are substituted.

Table 7.1: Initial state-variables in hover

State-variable	x_0	y_0	z_0	ϕ_0	θ_0	ψ_0	$\dot{x}_0$	$\dot{y}_0$	$\dot{z}_0$	$\dot{\phi}_0$	$\dot{\theta}_0$	$\dot{\psi}_0$
Initial value	0	0	0	0	0	0	0	0	0	0	0	0

By definition, all derivatives of position and angles with respect to time are zero during hovering. The initial position x_0, y_0 and z_0 are chosen to be zero for reasons of convenience. The same counts for the initial yaw angle ψ_0. In state-space system equations (7.25) and (7.26), the input variables u_1 through u_4 are chosen to represent total thrust, yaw moment, pitch moment and yaw moment. Referring to Fig. 7.1b, the input vector becomes:

$$\begin{aligned}
u_1 &= F_1 + F_2 + F_3 + F_4 && \rightarrow u_{1_0} = mg\\
u_2 &= \ell\,(F_2 - F_4) && \rightarrow u_{2_0} = 0\\
u_3 &= \ell\,(F_1 - F_3) && \rightarrow u_{3_0} = 0\\
u_4 &= Q_1 + Q_2 + Q_3 + Q_4 && \rightarrow u_{3_0} = 0
\end{aligned}$$

ℓ represents the distance between the centre of the engine and the centre of gravity. This distance corresponds to 20.5 cm, assuming that the centre of mass coincides with the axis of symmetry when looking at the quadrotor from top. The initial value u_{1_0} equals the mass of the quadrotor m multiplied by the gravitational acceleration g. F_i are the thrusts created by engines 1 through 4 as depicted in Fig. 7.1b. The torque created

Table 7.2: Initial input variables

Input variable	u_{1_0}	u_{2_0}	u_{3_0}	u_{4_0}
Initial value	mg	0	0	0

by every individual engine adding to the yawing moment u_4 is symbolised by Q_i. Deviations of the input variables from the initial value are, just as the fluctuations of the state-variables, denoted by a Δ.

For the controller that is to be implemented in the quadrotor itself, the inputs given by the autopilot are not going to be forces and moments, but rather changes in voltage that increase or decrease the rotational velocity of the propellers, and therewith thrust and resulting moments. The thrust vs. RPM curve based on which the controller will change the angular speed of the engine to induce a certain motion is shown in Fig. 4.5 on page 18. The input variables chosen here result from the information that was available at the time the state-space system was created. There was no data available yet on the relation between thrust and angular velocity of the rotors.

7.5. Simulations in MATLAB & Simulink

In order to get an idea of the dynamic behaviour of the quadcopter, a numerical tool must be created that simulates rotational rates, angles, velocities and position. This program can be made in MATLAB directly or in Simulink. The simulation was first created in MATLAB and for visual purposes later implemented in Simulink. The team also had to make a decision between creating a single or multiple controllers: one for pitch and roll motions, one for altitude, and one for yaw motion. Although this approach seems more work, this strategy is the easiest option. This has to do with the interaction between functions, and what signal is on what line. However, in both MATLAB and in Simulink, only one controller manipulating all degrees of freedom is used. Many sources that have been found in literature are using only one controller for all motions.

7.5.1. Control scheme in Simulink

Next to putting transfer functions in MATLAB, another numerical method that can be used for control related simulation is Simulink. This program is based on MATLAB, and enables a user to connect blocks that contain certain functions. All these blocks rely on a time dependent input signal. Not only functions, but also features such as signal generators, and summation blocks can be implemented to design a scheme that represents a fully working controller. Because of its visual convenience, all command signals are handled in the same controller. Not all line that are depicted represent a voltage, some lines mean a physical motion. The next sections explains some more about these functions, how are connected and why. Once a proper controller had been designed, it is implemented on the hardware of the CPU of the MAV. From that moment on, signals from the sensors are handled properly by converting them into engine voltage adjustments

EOM block

To start the building of the total system in Simulink, a block representing equations of motion must be placed. This function models the dynamical behaviour of the MAV. Quantities of mass moment of Inertia, mass, thrust and radius are necessary for the model. The overall application of these functions has been explained previously. The matrices that are used for this program are presented by Eqs. (7.25) and (7.26) in Section 7.4. As such, it was decided that the input vector for this system contains the total thrust, roll moment, pitch moment and yaw moment. As said before, the state vector in this case is chosen to be the same as the output vector. This will provide the benefit that the equation of motion system has to made only once. However, one has to make sure that the right signals are fed back to the right position.

Feedback loops

In control theory it states that in order to compare a desired value to an actual or measured value, a feedback path must be used on which the actual signal is located. Feedback paths go from the output of the EOM-block towards a summation block, connecting a command to an difference, or error value. Since only three or four values must be fed back to each input value, on these feedback paths matrices are placed. These matrices must make sure that the right I/O size is fed back, and also the right signal to the right command input line. These matrices therefore have three or four rows, and twelve columns and are sparse matrices (that is consist of zeros mostly). Furthermore these matrices contain elements that indicate the linear relation between the true physical quantity, and the voltage that is fed back to the system. In other words, on these feedback paths measurement units are placed, and these measuring instruments convert an actual movement (linear or angular) into a voltage, containing information about quantities of interest. However, since detailed information about the sensors is not yet available, it is assumed that these sensors are perfect. Thus a value of one is put on each appropriate location in such a matrix. For a velocity measurement, the GPS receiver is used. This measurement is done by using the Doppler-effect. When the MAV makes a movement, temporarily an altered frequency is detected, and this is changed into a velocity increment.

Gain tuning and test signal

What is described above is a still a method to control the MAV, and not an tangible controller. The latter can be produced, only if the plan is fully worked out, and therefore first some properties concerning the desired behaviour must be found. The system described above could not work without proper gain tuning. This is process where a physical constant must be found, to make sure that the response will be as desired. First the open system must be tuned. This is done by first putting a gain factor in front of the signal. This gain is shown as an typical triangular amplifier block and at this point contains the value one. Then, a certain test signal is fed to the open loop, which is wired to the system via a manual switch. After running the simulation, the plots are exported to the MATLAB workspace. This output vector (which is n by 12) is plotted, and as such the user can determine whether the observed results are satisfying or not. If the latter is not the case, then the earlier established gain allows for adjustments, that will cause a better response in the figure. This procedure is repeated until the right graphs are obtained. Once this is done, the switch that connected the test signal is opened. Now that the (first) open system is tuned, the closed loop is the following system to tune. For each separate input signal a gain is needed that converts the input linearly. After each loop the switch is opened, and the next gain is tuned. For the hardware, that means that these gains represent electrical resistances that are to be put in the actual controller.

7.5.2. Control scheme

The functional diagram that has been made in Simulink is shown in Fig. 7.7. In this figure, a total view of the above described system can be seen.

7.6. Results

After the tuning of the gains is completed, all kinds of signals can be applied to the system. All plots are done in MATLAB, and the results of the simulation can be observed in this section. In Figs. 7.8 to 7.9, x and z position inputs graphs can be seen.

Since the motion in the y direction is similar to the motion in x direction, the progress for this motion is the same as well.

Furthermore an angle command can be requested from the system as well by giving it a step input. The results for pitch and yaw can be seen in the next section.

The roll angle response to a step input is similar to the pitch response behaviour likewise.

7.7. Integration of guidance, navigation & control

Autonomously flying robots are heavily dependent on guidance and navigation algorithms that determine the MAV's current state using an integrated IMU/GPS system as part of the outdoor navigation system. In indoor environments where GPS signals are unavailable, a laser range finder provides data to compensate for the time drift of the IMU. Knowing the state of the system, the guidance system can calculate the trajectory and velocity required to reach a target position. The trajectory is a set of way points that the MAV has to follow in order to reach its destination. Therefore, the guidance system feeds a yaw angle ψ and desired velocity increments in x, y and z direction to the controller. The controller then uses both the state estimate and the target velocities as well as the yaw angle and outputs signals to each individual engine which then in return increase or decrease their rotational velocity, inducing the desired motion. The integration is illustrated on a high level by Fig. 7.12.

7.7.1. Image stitching and terrain map feature recognition

A number of image processing software were looked into for stitching the images to make a terrain map and other software to carry out image processing for object avoidance and detection. Since the on board computer and the ground station both use Linux as their OS, a restriction on only Linux compatible software was imposed on the imaging subsystem (**Req-pl-imag-11**). Due to the availability of open source software available for Linux they can be readily implemented and modified for prototype development. A google search revealed two options for Linux; a free open source image stitching tool called Hugin [107] and a paid tool called Autopano Pro by Kolor [108].

The Hugin software was downloaded and tested on laptop running a 64 bit Windows 7 with six gigabytes of RAM, an integrated Intel 4000 GPU and a core i7 2.30 Ghz processor [99]. About 11 MB of JPEG format image

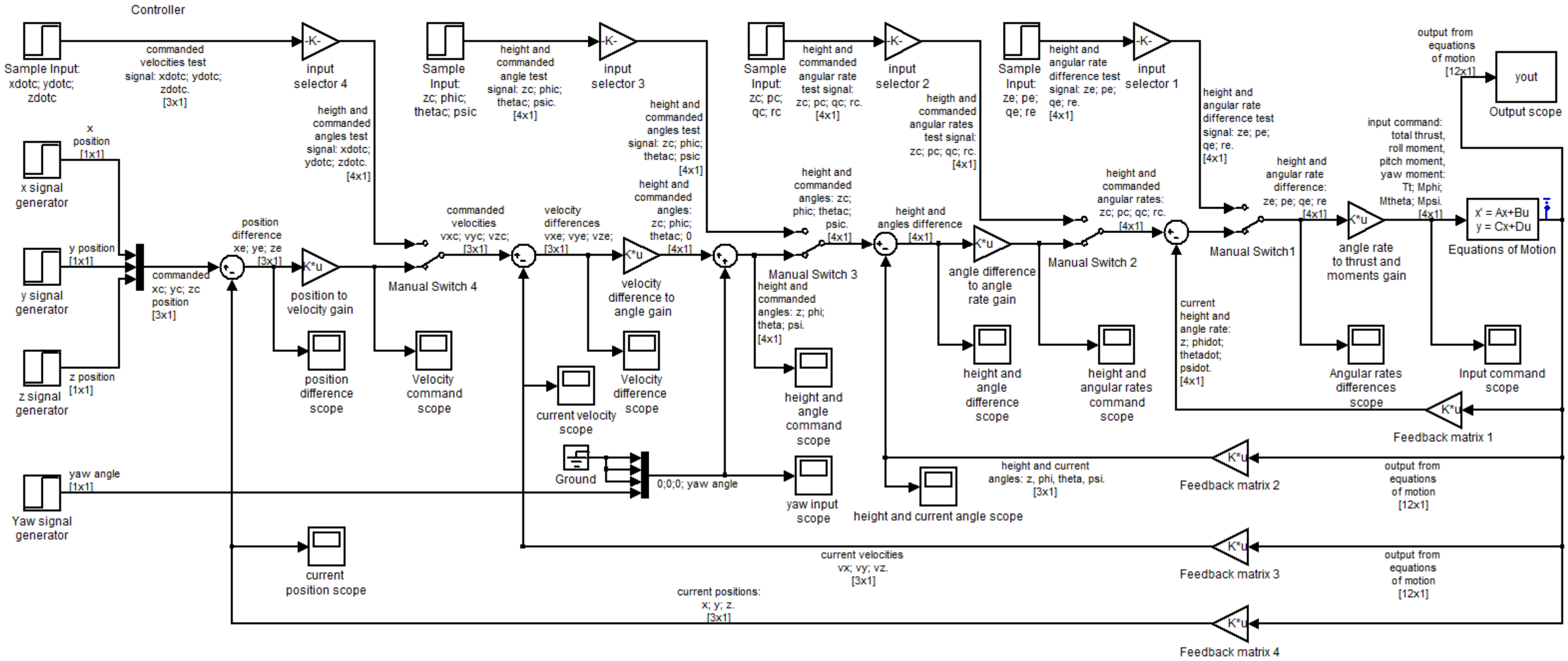

Controller
Sample Input: xdotc; ydotc; zdotc
commanded velocities test signal: xdotc; ydotc; zdotc. [3x1]
-K-
input selector 4
heigth and commanded angles test signal: xdotc; ydotc; zdotc. [4x1]
Sample Input: zc; phic; thetac; psic
height and commanded angle test signal: zc; phic; thetac; psic. [4x1]
input selector 3
height and commanded angles test signal: zc; phic; thetac; psic [4x1]
Sample Input: zc; pc; qc; rc
height and commanded angular rate test signal: zc; pc; qc; rc. [4x1]
input selector 2
heigth and commanded angular rates test signal: zc; pc; qc; rc. [4x1]
Sample Input: ze; pe; qe; re
height and angular rate difference test signal: ze; pe; qe; re. [4x1]
input selector 1
height and angular rate difference test signal: ze; pe; qe; re. [4x1]
output from equations of motion [12x1]
yout
Output scope
x position [1x1]
x signal generator
y position [1x1]
y signal generator
z position [1x1]
z signal generator
yaw angle [1x1]
Yaw signal generator
commanded xc; yc; zc position [3x1]
position difference xe; ye; ze [3x1]
K*u
position to velocity gain
position difference scope
Manual Switch 4
Velocity command scope
commanded velocities vxc; vyc; vzc; [3x1]
current velocity scope
Ground
velocity differences vxe; vye; vze; [3x1]
velocity difference to angle gain
Velocity difference scope
0;0;0; yaw angle
height and commanded angles: zc; phic; thetac; 0 [4x1]
Manual Switch 3
height and commanded angles: z; phi; theta; psi. [4x1]
height and angle command scope
yaw input scope
height and commanded angles: zc; phic; thetac; psic. [4x1]
height and angles difference [4x1]
angle difference to angle rate gain
height and angle difference scope
height and current angle scope
height and current angles: z, phi, theta, psi. [3x1]
Manual Switch 2
height and commanded angular rates: zc; pc; qc; rc. [4x1]
height and angular rates command scope
current height and angle rate: z; phidot; thetadot; psidot. [4x1]
Manual Switch1
height and angular rate difference: ze; pe; qe; re [4x1]
angle rate to thrust and moments gain
Angular rates differences scope
input command: total thrust, roll moment, pitch moment, yaw moment: Tt; Mphi; Mtheta; Mpsi. [4x1]
x' = Ax+Bu
y = Cx+Du
Equations of Motion
Input command scope
Feedback matrix 1
Feedback matrix 2
output from equations of motion [12x1]
Feedback matrix 3
output from equations of motion [12x1]
current velocities vx; vy; vz. [3x1]
current position scope
current positions: x; y; z. [3x1]
Feedback matrix 4

Figure 7.7: Control scheme

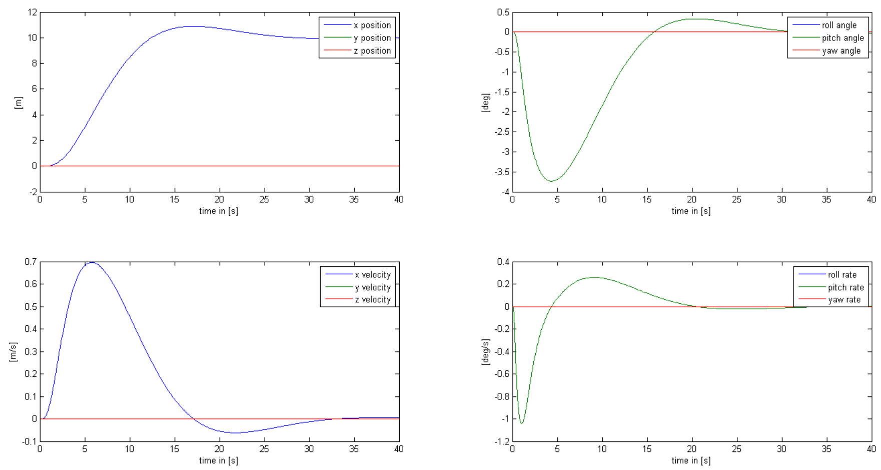


Figure 7.8: Input in the x direction

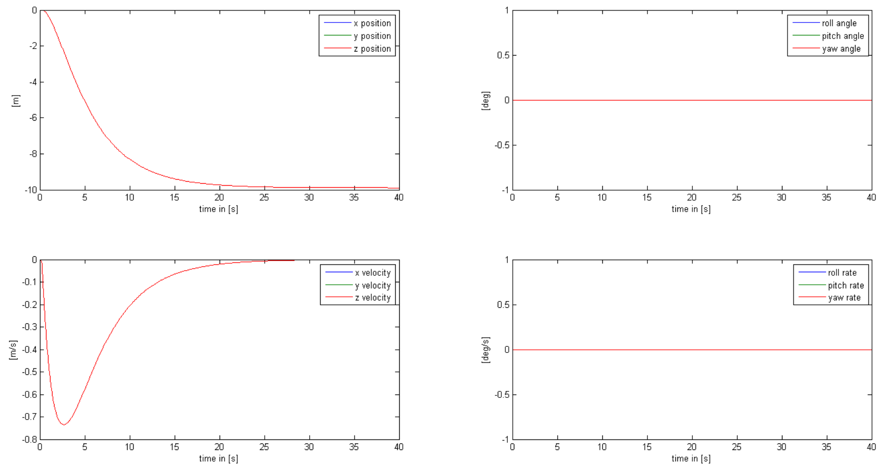


Figure 7.9: Input in the z direction

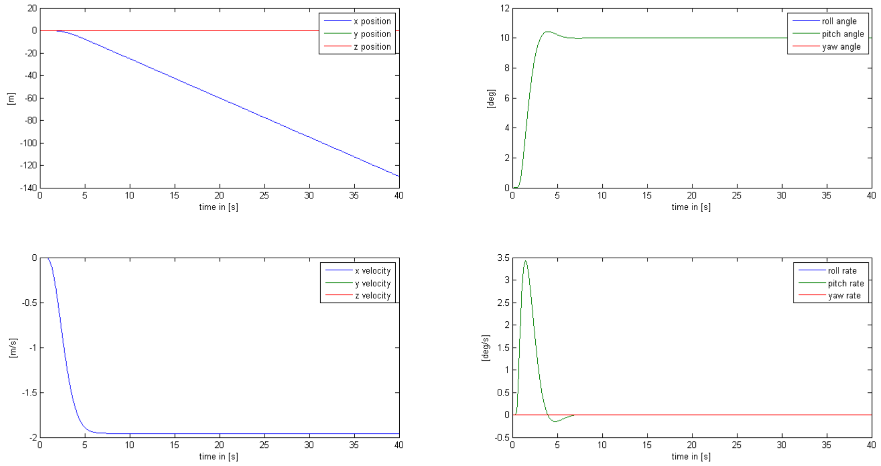


Figure 7.10: Input in the pitch angle

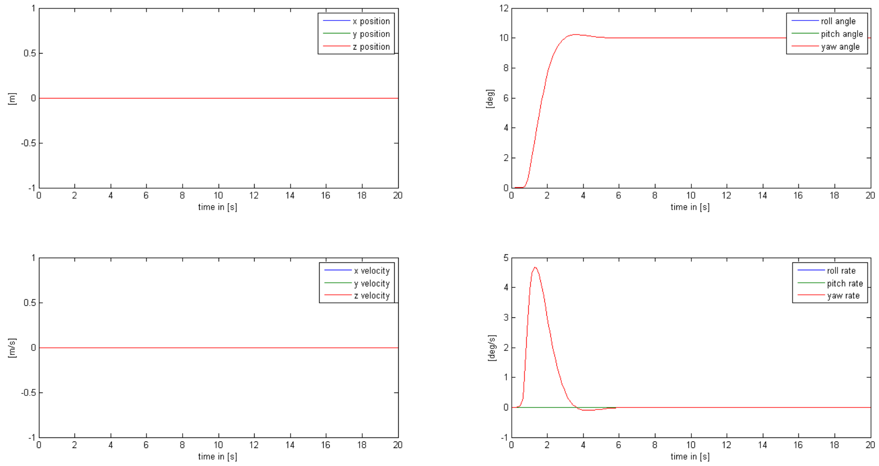


Figure 7.11: Input in the yaw angle

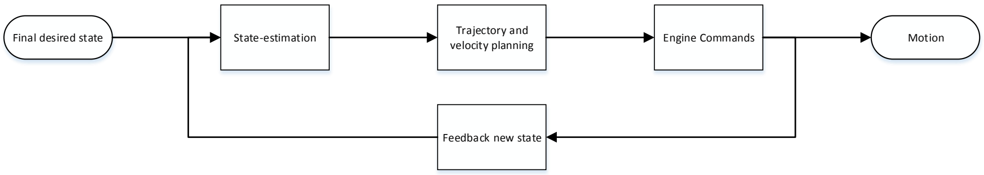
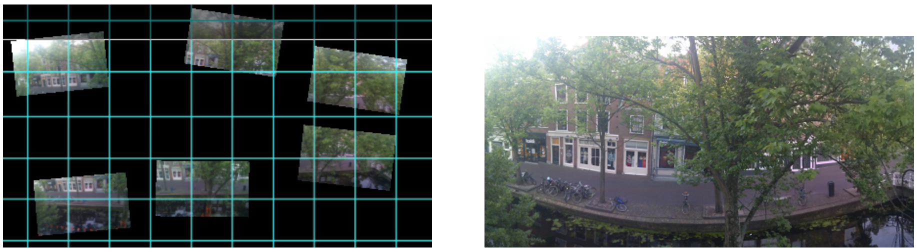


Figure 7.12: High level flowchart of the guidance, navigation and control system

were stitched together in approximately 15 seconds (Fig. 7.13). The data consisted of six images each of size approximately 1.8 MB and of resolution 2592 x 1552. It is assumed, for the sake of a time approximation, the ground system consisting of Intel Core i3 processor and similar GPU and RAM will be able to merge six images of 12 MP in twice the time; hence it will require 30 seconds to merge 6 images, which is equal to five seconds per image. Performing the mission at image level C requires 96 images; this translates to 8 minutes of processing time required for stitching; this means **Req-pl-imag-10** will not be fully satisfied since more than 6 minutes will be required by the system to finish generating the map for mission A. Since the software is open source the time required for image stitching can be improved by using the UAV attitude as inputs. The image stitching software will also have the ground map from openstreetview to help align and orient images. This will hopefully reduce the time required for image stitching and help meet the requirement.

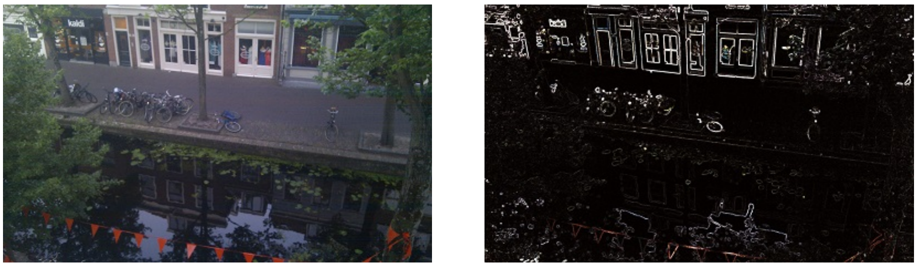


(a) 6 different images **(b)** Stitched panorama generated by Hugin

Figure 7.13: Merging images using Hugin

Besides stitching, it is also desired for the software to be able to detect the features on the terrain map. Identifying road blocks are a defined requirement (**Req-pl-imag-6-1**), but the ground map can be processed to extract a lot more information out of it to improve the autonomy of the whole UAS. To help perform mission B better the GPS cordinates of the boundaries of the houses would be useful, this information can be extracted by feature recognition software. Generating a road map from the terrain map can also be useful for autonomous operation; the recognition of other features such as fires, and measuring the degree of damage to property can be useful information for SAR operations. The software used for feature detection and GPS coordinate estimation can also be used for object detection and avoidance for missions B, C and D.

The software that were looked into for image processing are OpenCV [109], Monterverdi [110] (see Fig. 7.14 for edge extraction by Monteverdi) which uses the Orefo Toolbox, Find-Object [111], SimpleCV [112] and GemIdent [113]. OpenCV, Monteverdi, SimpleCV and GemIdent are capable of image processing; of these four, Monteverdi and GemIdent have GUI besides being open source. GemIdent is software based on Java that can quickly identify objects and patters from images; this software can be used for road block identification, besides a number of other types of object detection. Since OpenCV and the SimpleCV are modules which need to be used for coding they were not tested but their applications and capabilities were studied for possible application into the UAS. OpenCV and SimpleCV are both capable of almost all the functions of Monteverdi and even have tools for video processing. Find-Object is a purely video processing software with a GUI that gives feature points output from a video file or a video stream; it has the ability to implement a number of algorithms for object detection and definition (SIFT, SURF, Dense or Fast algorithm) which can be selected by the user. The information from this program can also be used to estimate the ground velocity of the UAV by using optical flow algorithms; this however is not currently available in Find-Object. This software is discussed more in the outdoor navigation section, where it is used to verify the requirements on the imaging system from the outdoor missions. Monterverdi carried out edge extraction from a 5 MP JPEG image in less than a second on the same Intel i7 laptop [99].



(a) Original image **(b)** Edge extracted image

Figure 7.14: Edge extraction by Monteverdi

The software for image processing has not been designed in this report, however with more time a custom tailored software using SimpleCV or OpenCV will be possible. Some of the discussed software were tested to measure and demonstrate the amount of time required to process the images. It was found that approximately 5 seconds would be required to stitch each image, meaning a total of 8 minutes for the whole of mission A. Further it was demonstrated that edge extraction can be performed in under a second on a 5 MP image. It is believed with software specifically tailored for the mission these times can be reduced. For an overview of how the imaging data handled by the UAS refer to Fig. 6.10 .

7.7.2. Low altitude image feature recognition

Image feature recognition is the process of recognising certain features in images. An example of this is number recognition, which is required to perform mission B and D autonomously. For mission B (element 4) building numbers should be recognised, while for mission D a sequence of digits should be reported. Another example is the detection of the landing zone, which is marked by a black and white marker. When looking for a number recognition algorithm, it turned out that the algorithm is more generally applicable to any feature. This section will explain how the feature recognition process will be performed and which algorithm will be used.

General process

Although the mission parts which require feature recognition have a different setting (flying versus stationary), the general process to recognise features will be the same. The first step in the process is to capture an image which contains the feature. This will be done by extracting frames from the live feeds. This can be done for either camera, depending on which has the best view on the feature. There is also a minimum feature resolution requirement imposed by the algorithm which should be taken into account when the camera is selected. The next step is to process the image for the feature that is being searched for. Once the feature has been identified it can be processed further. This means for example assigning the house number on the map, or allocating the GPS coordinates the the location of the feature. Off-board image processing is preferred since it will save computational load and therefore also power of the MAV. This can be done when the feature is not required for immediate navigation. Examples of this are house number recognition and locating the yellow cross roof to assign GPS coordinates. However when the feature is required for navigational purposes, the image will not be sent to the groundstation because of possible delays. In this case it will be directly processed on-board the MAV. Examples of this are landing in the middle of the marked landing zone or landing next to the yellow cross. Here the MAV will position itself based on the camera vision. Since the MAV has less processing power than the groundstation, it could be possible that the resolution has to be lowered before processing. However this is something that will have to be experimentally tested.

Feature recognition

Character recognition in images is a subject that has been researched quite extensively in the last half century. The main focus has been on images resulting from scanned books and documents [114] [115] and handwritten characters [116] [117]. Only recently the focus shifted to natural scenes in photographs, partially because of Google. They are interested in such algorithms to improve their Google Maps services. A new benchmarking database (SVNH database[3]) and number recognition algorithm was developed by [118], supported by Google. This algorithm had an accuracy of 90.6 %. [119] then reached a state-of-the-art accuracy of 95.10 % on the SVHN dataset, which is a 48 % error improvement, using Convolutional Neural Networks (ConvNets). The source code and a tutorial are available at *eblearn.sf.net*. Also the system will be used by emergency services with lives at stake. Therefore the accuracy should be as high as possible. Because of this it was decided to implement the ConvNets algorithm in the design for number recognition. However ConvNets are hierarchical feature learning neural networks. Their structure is biologically inspired and they can automatically learn a unique set of features optimized for a given task. Therefore they can be used to recognise any feature, something that [119] tested as well. They tested with Pedestrians detection and Traffic Signs classification, while [120] applied ConvNets to facial recognition. [121] showed the superiority of ConvNets in a traffic sign classification challenge. Hence the ConvNets algorithm will be used as general feature recognition algorithm.

Algorithm optimisation for the IMAV competition

The ConvNet algorithms are so called learning algorithms. This means that they first need to learn to recognise features, before they can actually do so. Hence the algorithms can be optimised for the IMAV competition by learning them features that are specific for the competition. In this case that will be the digit board, the on A4 printed house numbers, the cardboard survivors, the landing zone mark and the yellow cross marker. By learning the algorithm with only images of these items the accuracy should be optimised for the competition. However note that for outside the competition the learning process should be as general as possible to be able to operate accurately in as many environments as possible.

7.8. Conclusions & recommendations

In conclusion, development of the navigation and control system was rather successful. However, more information from the aerodynamics and propulsion department is necessary to simulate the dynamic response more accurately, and to tune the gains such, that they can be implemented in the final system. There is also work to be done on verification and validation on all parts developed here, except for the Kalman-Filter which has been verified previously in this chapter.

A major part that still has to be done is linearising the EOM about different flight conditions, such that the gains can be tuned for these conditions, providing better flight performance throughout the entire mission. Additional recommendation for the indoor navigation system are listed below.

Requirements

Five main requirements have been identified that specifically apply to the guidance, navigation and control systems:

1. Navigation accuracy of 20 cm
2. Determine current state within 0.01 s
3. Bypass outdoor objects with 1 m distance
4. Bypass indoor object with 20 cm distance
5. Maximum power of sensors and computer of 5 W

The first requirement is verified by simulating a trajectory over 30 minutes. Therefore, a constant acceleration of 0.02 m/s^2 is taken, and the actual measurement errors of GPS and accelerometers implemented in the Kalman-Filter. The deviations of the Kalman estimates form the actual position after 30 minutes shall then not be larger than 20 cm. The final values are critical since the error in the acceleration will be largest right

[3] http://ufldl.stanford.edu/housenumbers/

before the mission is over. Several runs (necessary due to the Gaussian distribution of the error) of the simulation showed that after 30 minutes a deviation of about 30 cm from the actual position is obtained. These values are to be regarded with care though, as in the real flight vibrations and accelerations induced by gusts will dramatically increase the error in the readings of the accelerometers. Nevertheless, the quadrotor also has laser range finder on-board, which increase the navigation accuracy tremendously when indoors. Also, for mission element D, the quadrotor will have to land atop a building. Once landed, the bias in the state estimate can be reset.

Despite laser range finder and the option to reset the estimation bias, further investigation is required before requirement one is verified.

The speed at which the current state is determined depends on two main factors: clock-speed of the CPU and update frequency of IMU and GPS. Since the IMU operates at 200Hz and CPU operates at 1000 MHz, this requirement is verified. Yet, this is to be validated once the system is set up. These values here are not be mistaken with the frequency at which the controller adjusts engine setting. That control frequency equals 1000 Hz

Requirements three and four cannot be verified as of now, since there is no guidance system developed yet.

Finally, the maximum power consumption of this system is only dependent on the power usage of the on-board system lies at 5 W according to the manufacturer. Therefore, this requirement is verified.

8

COMMUNICATION

The communication subsystem's primary objective is to provide a stable data link between the UAV and the user. Designing this subsystem should not be underrated. A failure can lead to the untimely end of the mission. This chapter focusses on the design of the communication subsystem. First all suitable means of communication are covered. A trade-off is then be done and finally, the sizing of the selected technology is handled.

8.1. DATA PACKETIZATION

A lot of information will be transferred from the transmitter to the receiver. First the different types of data need to be determined and then data packets will be created. The references [122], [123] are used as background knowledge in this section.

8.1.1. DATA TYPE

In this section, all data types are listed in two categories: air to ground and ground to air.

- Air to ground
  - High resolution pictures (mission A)
  - LIDAR point cloud
  - Live feed (video and/or audio)
  - Altitude
  - Latitude and longitude
  - Velocity
  - Battery status
  - Control signals
- Ground to air
  - Flight commands
  - Mission controls

8.1.2. DATA PACKETS

Data types are grouped together into three packets depending on their respective volume and update frequency: status packet, imagery packet and command packet.

STATUS PACKET

The status packet contains the location and behaviour parameters of the UAV. The packet contains its altitude, longitude, latitude, velocity, control signal, battery and payload status.

Longitude and latitude require 28 bits each if a six decimal accuracy is given. Since the ceiling of the UAV is 950 ft, 16 bits are used for the altitude and another 16 for the velocity. 7 bits are used to define the battery status. The LIDAR measurement points are also transmitted via this link. That is 39 bits with a frequency of 360 Hz.

In total one set of information of the status packet (every item) represents 162 bits. If the update frequency of each parameter is 10 Hz (except for the battery status which updates at 1 Hz and the LIDAR), that means that 14.85 kb/s will be sent downstream (1.86 B/s).

Imagery packet

The imagery packet completes the downstream volume of the communication module. It consists of the map of the selected area and the live feed.

The HD pictures will weigh at most 2.15 GB and the data rate will depend on the transmission time. If the transmission duration is five minutes, the data rate will be equal to 55.2 Mb/s.

Another feature of the imagery packet is the live feed. For the video five low resolution frames are sent per second, 640x480 pixels. That represents 23.43 Mb/s of data. Regarding the live audio feed it was decided to define one sample by 16 bits. The usual sampling frequency is equal to 8,000 Hz.

Since there is only one camera connected to the transmitter, when the camera is mapping it will not transmit the live feed. The maximum data rate is 55.2 Mb/s. Depending on the quality of the link, this data rate may not be reached. In that case a memory card functions as buffer.

Command & control packet

The command packet contains the input from the pilot. It is the only link between the ground and the UAV. From the competition regulations, the package consists of two kinds of data: the mission inputs and the remote control inputs.

The mission inputs are commands given through the ground station: take-off, clear to land, abort mission, hover where you are, map the selected area. A combination of 8 bits are used for each of those commands. For the selection of the area, four coordinate pairs (220 bits) are added to the combination. Also, the payload is controllable: the camera and the LIDAR may be oriented from the ground station. Two angles are required from -90 to 90°, that is 16 bits. The LIDAR needs a HD map which is thus sent to the CPU.

Because of **Req-cont-3**, it should be possible to send control inputs to the UAV. There are six degrees of freedom to control (**Req-cont-4**) and the remote controller will have its own controller. The outputs are engine control commands. Despite it being difficult to find references on frequency bands used by RC controllers, it was decided to chose a 1 kHz frequency which is sufficient for real-time control of the UAV.

Summary

Table 8.1 summarises the packet sizes.

8.2. Communication technologies

A literature study needs to be performed to acquire some knowledge about the different means of communications. Five categories can be identified: cellular network communication, free-space optical communication (FSO), wireless local area network (WLAN), wireless personal area network (WPAN) and other radio frequencies. All types use electromagnetic waves to propagate through air. At the end of this section a table summarizes the characteristics of each type of communication.

8.2.1. Cellular network

For years engineers have been developing mobile internet. Nowadays the fourth generation is commonly used (4G). The third generation (3G) is also still in use. They make use of the mobile communication network to access the internet.

The differences between 3G and 4G are mainly about the bandwidth and the centre frequencies. The 4G technology is characterized by a larger data rate and by its frequency ranging from 2 to 8 GHz instead of 1.8 to 2.5 GHz for the 3G technology.

Their range is theoretically unlimited but in remote areas, where mobile network is sparse, problems may be encountered. Another problem is the susceptibility to saturated networks. In case of a disaster, relatives might be looking for news about people living in the disaster struck area and the network might not be able to provide enough bandwidth to allow for an optimal data link.

The cost for the use of technology varies with the desired data rate. They are set by mobile network operator such as Vodafone, KPN amongst others.

8.2.2. Free-space optical communication

Optical communication is a well known technology. Fast communication links between continents have been established by implementing fibre optics cables. However, nowadays wireless communication is very accessible and lighter than having to carry cables, Free-Space Optical (FSO) is discussed here.

The FSO consists of a transmitter and reciever as for all means of communication. The transmitter is a laser beam generator which has to be accurately pointed to the reciever. While most of the modulation schemes used are QAM, PSK or FSK, modulating laser communication is preferably done using the OOK scheme [124]. For this modulation scheme the wavelength and phase are not relevant. Only power is a relevant parameter for the transmission of data and that makes it robust against weather or any source of noise.

Table 8.1: Overview of the different data packets

Item	Volume [b]	Frequency [Hz]	Max data rate	
Status packet				
Longitude	28	10		
Latitude	28	10		
Altitude	16	10		
Velocity	16	10		
Battery status	7	1		
Control signals	28	10		
Payload status	39	360		
			15,207	b/s
			1,901	B/s
Command packet				
Take-off	8			
Landing clearance	8			
Abort mission	8			
Hover	8			
Payload control	16			
Map the area	228			
Building map LIDAR	12,800,000	0.5		
			6	Mb/s
			781	kB/s
Control packet				
Engine 1 setting	16	1,000		
Engine 2 setting	16	1,000		
Engine 3 setting	16	1,000		
Engine 4 setting	16	1,000		
Camera control	16	1,000		
			80,000	b/s
			1,000	B/s
Imagery packet				
Map	$17.2 \cdot 1024^3$	0.0056		
Live feed video	$24.5 \cdot 1024^3$	1		
Live feed audio	16	8,000		
			55.2	Mb/s
			7	MB/s

Even though FSO is a fast way to communicate, 1 Gb/s has been reached experimentally [125], there are two drawbacks. The first is that communication is only possible in Line Of Sight (LoS) which means that communicating is impossible whenever the UAV enters a building or is behind another object. The second is the high required pointing accuracy. The pointing accuracy required is in the order of the micro radians [126].

8.2.3. Wireless local area network

To the general public, WLAN is the most known type of wireless communication because it is often used to create a home-, company or university-network [122]. Different protocols exist with different characteristics, eg data rate and range. The power required however stays constant among the different protocols, in the order of a tenth of a Watt. The most widely used protocols are 802.11b/g/n [122].

The data rate varies from 11 Mb/s to 248 Mb/s [127] while the maximum outdoor range varies from 100 m to 150 m and indoor from 35 m to 70 m [128].

8.2.4. Wireless personal area network

The definition of WPAN wrongly implies that its range is smaller than WLAN. However, higher ranges can be achieved, up to 1 500 m [128]. Two WPAN devices may be suitable regarding the data rate and the range: Bluetooth and ZigBee.

Long range Bluetooth system can transmit up to 1000 m [129] and can even be extended to 2000 m. Its data rate reaches 3 Mbs.

Regarding XBee, its data rate is only around 250 kbs and its range up to 1 500 m [128].

Their centre frequency is 2.4 GHz although some types of the XBee can operate around 800 MHz, which increases the range of transmission.

8.2.5. Radio frequency

In this report radio frequency (RF) means the other types of communication which use radio waves. Its rate of oscillation ranges from 3 KHz to 300 GHz. It can be seen that the frequency of the WPAN devices (2.4 GHz) are included in this range.

Relay points allow for a longer range compared to direct communication. Theoretically, the range of RF is unlimited. The data rate is limited by the bandwidth of the RF used. The lower the frequency, the less sensitive the signal becomes to disturbances at the cost of being able to carry less information.

The only challenge is to comply with EU regulations regarding the frequency spectrum and the transmission power. Further studies are done later in this chapter.

8.2.6. Overview

This section gives an overview of all characteristics of the above-mentioned communication means. Some cells are left empty due to a lack of information or a lack of relevance. The effect of the walls on the Bluetooth signals for example are not fully known.

Table 8.2: Characteristics of different wireless mean of communication

	DR_d [Mb/s]	**d_{indoor} [m]**	**$d_{outdoor}$ [m]**	**B [MHz]**	**f [GHz]**	**Cost [€]**
Cellular network						
3G	3	∞	∞	5 - 20	1.8 - 2.5	+/- 500
4G	12	∞	∞	5 - 20	2 - 8	+/- 500
FSO						
Laser	> 1,000	NA	∞	25	+/- 10^6	200 - 1,000
Radio						
VHF		∞	∞		0.03 - 0.3	
UHF		∞	∞		0.3 - 3	
Satellite	Limited	∞	∞	Limited	0.5 - 1 / 12 - 18	> 70,000 / an
WLAN						
802.11b	11	35	110	22	2.412 - 2.484	+/- 50
802.11g	54	35	110	20	2.412 - 2.484	+/- 50
802.11n	248	70	160	20-40	2.412 - 5	+/- 50
WPAN						
Bluetooth 2.0	3		1,000	1	2.4	150
Bluetooth 2.1	3		2,000	1	2.4	+/- 30
XBee	0.250		1,600	0.3 - 0.6 / 2	0.868 - 0.915 / 2.4	+/- 35

8.3. Trade-off

Trading-off is a key action in a design process. This is the way to base a choice on something else than feelings. In the section four criteria are weighted and then used to compared the different means of communication. Those four criteria are cost, bandwidth, data rate and range.

8.3.1. Criteria weighting & volume estimation

Due to the low amount of criteria, it is decided to use Saaty's method to weigh the criteria. The method consists of giving grades from one to three to each criterion. This is summarized in Table 8.3.

Table 8.3: Criteria weights determination with Saaty's method for communication trade-off

	Cost	**Bandwidth**	**Data rate**	**Range**
Cost	1	3	2	2
Bandwidth	0.33	1	2	0.5
Data rate	0.5	1	1	1
Range	0.5	2	1	1
Sum	2.33	7	6	4.5
Normalised	0.12	0.35	0.30	0.23

Although some criteria have a low weight, each type of communication should still comply with the set of requirements given in Appendix A. The most important ones at this stage are **Req-coms-1** and **Req-coms-2**. **Req-coms-3** and **Req-coms-4** are dependent on payload characteristics. This will be subject to iteration.

8.3.2. Results

The main characteristics of the UAS and the data link have been defined. Now the different means of communication may be compared.

Table 8.4: Trade-off between means of communication

	Weight	Cellular network	FSO	Radio	WLAN	WPAN
Cost	0.12	4	2	3	2	4
Bandwidth	0.35	3	4	3	4	1
Data rate	0.3	2	5	4	3	1
Range	0.23	5	3	5	1	2
		3.27	**3.84**	**3.76**	2.78	1.58

The laser communication and the radio frequency turn out to be the most suitable choices despite laser communication being unserviceable indoors. It is therefore decided to prioritise RF. Designing a working FSO communication system is time consuming and requires lots of knowledge. Nevertheless, FSO presents one big advantage: its data rate. Furthermore, laser communication would also be impressive to the jury.

That is why much effort is invested in the design of the RF system and if time is remaining, the laser communication system is designed. Otherwise, a recommendation is made to do further research about FSO technologies for UAV air to ground data link in the future.

Note that a sensitivity analysis was assessed. If the bandwidth criterion was removed, the data rate and range criteria would have similar weight and the weight of the cost would be half of theirs. The RF system would be the clear winner of the trade-off and the FSO would have the same amount of points as the 3G-4G.

8.4. Frequency

Section 8.4 focusses on the frequency spectrum of the communication. The trade-off has been done and it shows that the RF communication type is the most suitable for the UAS. First information from RC hobbyist community is collected and then a choice is made. This part of the chapter also assesses the compliance of the used frequency with Dutch and European regulations.

8.4.1. Frequency spectrum

Since years spectrum spread systems have been studied and developed [130, 131]. Direct sequence spectrum spread (DSSS) frequency for remote controls have to be seriously considered. It is a robust modulation technique used to prevent interference and jamming. The DSS technology makes it impossible to interfere with the frequency [130]. It is comparable to a car changing lane as soon as its lane is blocked. This technique is mainly used for 2.4 GHz frequency (WiFi, cordless phones, RC vehicles) but also for other frequencies. A similar technique is the frequency hopping spread spectrum (FHSS) where the frequency is constantly hopping in the spectrum. Different frequencies will then be used depending data packet.

Overview of amateur frequency spectrum

The different frequencies spectrum for video transmission are here highlighted [132–134].

Table 8.5: Frequency spectrum used for amateur RC air vehicle

	+	-
5.8 GHz	Small antenna size	High obstacle sensitivity
	No interference with GPS	Low market availability
2.4 GHz	High market availability	Obstacle sensitivity
	No interference with GPS	Moderate penetrating ability
	DSSS	Crowded frequency
1.3 GHz	Barely used frequency	Risk of GPS interference
- video only	Low obstacle sensitivity	Not small antenna
433 MHz	Barely used frequency	Large antenna
- data only	Low obstacle sensitivity	
	No interference with GPS	
35 MHz	Dedicated frequency for RC in NL	Large antenna
- data only	Very low obstacle sensitivity	
	No interference with GPS	

Frequency for data and video transmission should be carefully selected. Indeed, they may not be the same except if the 2.4 GHz DSSS frequency is selected. Note that in the US other frequencies are available such as 900 MHz and 72 MHz [133]. It may be interesting to have modular transmitter and antenna on the UAV to switch to a frequency which is more suitable in the US. Potential clients are situated in the USA, which makes it interesting for the post-competition phase: they have money, and many disasters have struck the States during the past years.

Trade-off

Different criteria have to be taken into account for the comparison between the frequencies. Performing a trade-off as it has been done previously in this chapter revealed to be difficult since the requirements for the video and data transmission are the same, but each of them has to use a different frequency (except from 2.4 GHz).

It was then decided to proceed by elimination. From the 35 MHz frequency the range is more than sufficient, but its antenna has to be too large. The same reasoning applies to 433 MHz; those frequencies are over-designed for the mission. Those frequencies may still be considered for the post-competition phase: when the UAV will have to operate in remote regions. Finally the 5.8 GHz frequency is too sensitive to obstacles (walls for example) to be considered as a good choice.

Resulting from this elimination, it arises that the most suitable frequencies are 1.2-1.3 GHz and 2.4 GHz.

To be sure that operating those frequencies are not forbidden by law, the Dutch and European regulations are checked. From 1.2 to 1.3 GHz, all channels (1 kHz) are dedicated for amateur radio [135]. Note that if the 1.2 GHz frequency is used, a low pass filter is required to avoid coupling with the 2.4 GHz band. It is the same counts for frequencies ranging from 2.32 to 2.45 GHz. **Req-cont-1** is fulfilled.

Req-cont-1 is also met: the frequency 2.4 GHz DSS uses different channels of the spectrum.

8.5. Link budget analysis & antenna selection

An important analysis to assess the effectiveness of the transmission link is the link budget analysis. The spreadsheets are given in Appendix C. This section is divided in four parts: antenna selection, gains, losses and results.

8.5.1. Antenna selection

Different types of antennas are available on the market. They mostly depend on the frequency and the desired gain. Table 8.6 summarizes the characteristics of five main types of antenna: cloverleaf, whip, loop, helical and parabolic. Commercial platforms are used to get an idea of the characteristics (hobbyking.com and hobbywireless.com). All of them are anisotropic except the cloverleaf antenna which is almost isotropic. Some types are not considered because of their great mass such as the Yagi antenna.

Table 8.6: Characteristics of different types of antenna

Type	Maximum gain [dB]	Frequency [Hz]	Height [mm]
Clover leaf	7	2.4 - 2.5	32
Clover leaf	5	1.08 - 1.38	59
Whip	3	2.4 - 2.5	137
Whip	11	2.4	330
Loop	5 - 15	2.4	100
Helical	7.94	2.4	100
Parabolic	0.97	2.4	30

The helical and parabolic antenna gain has been calculated with the equations provided in reference [136]. The diameter of the helical antenna is assumed to be 3 cm and its length 10 cm. Regarding the parabolic antenna a diameter of 6 cm is considered. The maximum gains are given in Table 8.6.

Parabolic and helical antenna are quickly taken away from the trade-off. A high pointing accuracy is required to achieve the maximum gain. Due to the limited weight budget a movable antenna is not an option but may be considered for further improvements of the product. The helical antenna offers indeed high gains. The loop antenna is wrapped around the central piece of the UAV. Interference with the different magnetic fields of the other subsystems is the biggest reason why the loop antenna was put aside. Since the attitude of the MAV varies constantly, having a isotropic antenna is beneficial for the simplicity of the communication system. The cloverleaf is chosen after the first iteration process. Unfortunately, the structure group could not allocate a suitable location for the cloverleaf antennas since two hemispheres are difficult to integrate into the structure. A pair of whip antennas are therefore considered after the second iteration.

For the ground station a whip antenna is also selected. The operator can manually change the orientation of the antenna to achieve the best receiving gain. Whip antenna was also chosen because of its simplicity, transportability, robustness and its high gain.

8.5.2. Gains

The only gains of the link budget are the antennas. The receiving 30 cm whip antennas has a maximum gain of 11 dB while the shorter on board antennas have a gain of 7 dB.

8.5.3. Losses

Five losses are considered: transmission line loss, antenna pointing loss, free space loss, propagation and polarization loss and other losses.

Computing the transmission line loss turned out to be a difficult task. Reference [136] is used to evaluate this loss to be equal to 1 dB.

The pointing loss of the receiving and transmitting antenna is almost null since the pair of whip antennas is quasi-isotropic and the GS whip antenna is manually oriented to achieve an optimal gain.

The free space loss is computed with Eq. (8.1) where d is the maximum range in km (**Req-coms-2**) and f is the frequency in GHz. The loss equals 92.1 dB.

$$L_s = 92.5 + 20\log_{10} d + 20\log_{10} f \tag{8.1}$$

From the same reference used before [136] the propagation loss is estimated to be 0.5 dB.

The UAV shall be able to communicate outdoors. Therefore the communication link has to cross walls. The wall attenuation is dependent on the frequency of the wave, the geometry (thickness) and material of the wall. A wall made out of concrete with a thickness of 30 cm has a attenuation of 35 [137] to 50 dB [138] for the 2.4 GHz frequency. Note that the losses experienced by lower frequencies are smaller since they have better penetrability [139, 140]. If the walls are composed of wood, the attenuation can be measured to be 50 dB [138] (independent on the thickness). The effects of the wave penetration angle is not studied in this report because of the lack of knowledge about the deployment site. But it can be said that the bigger the penetration angle becomes, the bigger becomes the loss [141]. If the wave goes through a wall with windows, the loss is decreased by 5 to 7 dB depending on the type of window [141].

Frequencies below 10 GHz are almost not affected by weather (rain, fog, blizzard) [142, 143].

8.5.4. Results

The link budget is performed for the 2.4 GHz frequency. The 1.3 GHz frequency will have a greater margin because it is less subject to the different losses. The results are available in Table C.1.

The maximum required energy per bit to noise density E_b/N_0 for all the different types of modulation types is 7.61 dB. This figure is set to establish the margin. After having added the losses and gains to the equivalent isotropically radiated power, the actual E_b/N_0 is equal to 61 dB in line of sight communication. Note that the noise temperature is assumed to be 135 K [136] and the download rate is equal to 25 Mbs. The margin is enough to have an efficient link: 54 dB.

In case of indoor communication, the received signal becomes very weak if the other losses L_o are set to -50 dB (that represents the attenuation experienced after going through a wood wall without windows): 4 dB. If the data rate is decreased to 1 kbs (status packet only), the margin is 48 dB.

8.6. Communication architecture

Now the communication type, the operating frequencies and the type of antenna have been defined, the final architecture of the communication subsystem is illustrated in this section.

8.6.1. UAV antenna location

After the first iteration some structural restrictions forced a reconsideration of the antenna type. The selection changed from the cloverleaf to whip antenna. Most of the time the UAV is higher than the ground station. That implies that the antenna is to be located at the bottom of the vehicle.

The radiation pattern of a whip antenna is characterized by a torus with small inner radius. That means that in the axis of the antenna, the gain is minimum or null. To insure an optimal coverage, it is decided to install a pair of antennas. Each of them having a different inclination.

To operate more than one transmitter (or frequency) simultaneously an antenna diplexer is used [144]. Therefore two antennas are sufficient to provide an acceptable gain.

8.6.2. UAV Tx-Rx

On-board a 2.4 GHz 250 kbs DSSS transceiver insurea the status and command link between CPU and the GS [145]. Its characteristics are summarized in the communication budget table.

Besides the status packet the GS computer receives the imagery packet (live feed or map). This data does not go through the CPU. A 1.3 GHz transmitter is used and connected to the antennas.

Since a Turnigy radio controller is used, it is decided to use a receiver from the same manufacturer for the control packet.

8.6.3. Communication block diagram

In Fig. 8.1 the overview of the communication architecture is shown.

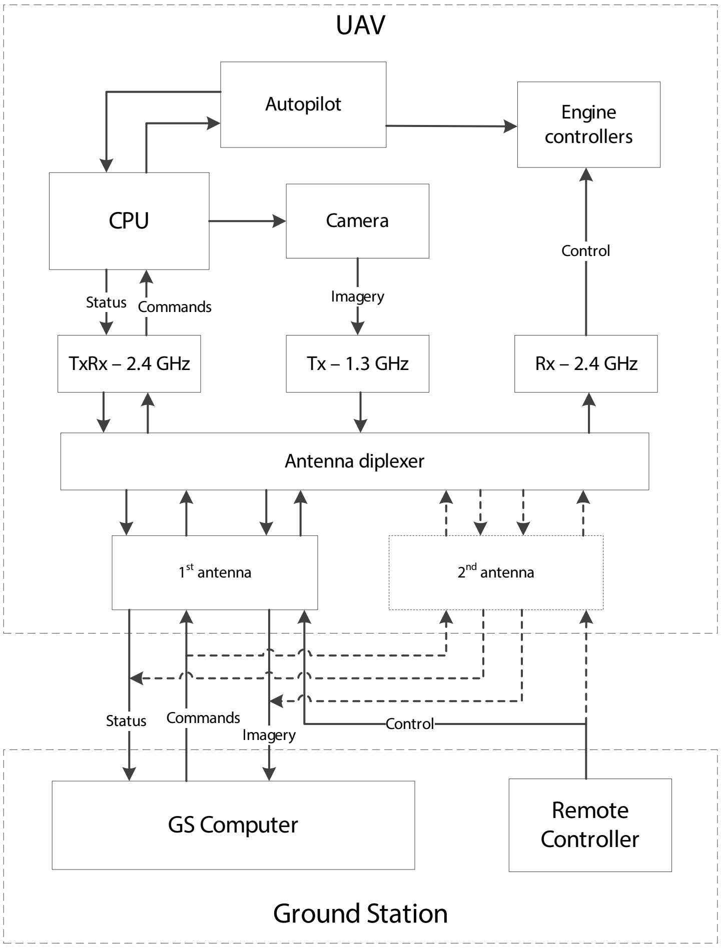


Figure 8.1: Communication block diagram

The dotted antenna block and arrows represent an alternative. It is an 'or gate'. If the system is able to detect that the range is not achieved with one antenna, it tries with the other one.

8.7. Communication budget

Different components are mounted on the UAV to establish an efficient communication link. To guarantee that the mass, cost and power budget are not overshot, a summary is provided in Table 8.7.

Table 8.7: Communication power, mass and cost budgets

	f [GHz]	I [mAh]	V [V]	P [W]	m [g]	Cost [€]
Tx	1.2 - 1.3	450	12	5.4	35	100
TxRx	2.4	200	3.3	0.66	5	20
Rx	2.4	50	4.5	0.225	18	6
Antenna	-	-	-	-	20	25
Diplexer	-	-	-	-	5	-
Total				6.2	103	176

8.8. Recommendations

The communication subsystem has been designed. Some aspects have not been covered though, because of the time restrictions. This section highlights the different challenges that may be addressed as future work.

Already mentioned in this chapter, the laser communication is studied further to be implemented to the UAV. Its very high data rate and its robustness against jamming are its main advantages. Some ideas came up during the design process such as a transmitter mirror system to reflect the laser beam coming from the GS. By doing so, on-board power is saved.

Another solution which is to be considered for the next UAV versions is the movable anisotropic antenna. Higher gain is achieved and the mission range is extended. This improvement has a cost: increase in mass due to movable systems, increase in power consumption and increase in cost.

The time restrictions of the project did not allow to cover some aspects of the communication. It is recommended to go through those before going any further with the product. Those are the bit error analysis, the overflow phenomena but also the design of a micro antenna diplexer.

9 ON-BOARD POWER

The power subsystem provides electrical current at the required voltage to the right location at the right time. Since different subsystems have different power requirements, the battery should be as universal as possible. Battery selection and design is discussed in Section 9.1. The on-board computer regulates the power distribution through a set of converters and regulators to the various system components. This power architecture is discussed in more detail in Section 9.2

9.1. BATTERY SELECTION

The batteries should provide enough power for the vehicle to perform all missions in half an hour of time. Since batteries take up a big percentage of the weight of the vehicle, it is desired to select batteries with a high energy density. Currently, these are of the type Lithium-ion [146, 147]. These batteries also have a long cycle and shelf life and do not suffer from memory effects [148], which makes them a sustainable choice. Disadvantage of these batteries is ageing; in other words: the total available capacity decreases when the total number of discharge cycles increases [146]. However, with the current developments, this main disadvantage has been overcome at the expense of an higher cost [147]. The battery cells that will be used, have a maximum energy density of 212 Wh/kg and can be found in [147]. Throughout this section, please note that the sizing was part of an iterative process and that some data comes from later chapters.

The batteries need to have a minimum capacity of 85.5 Wh, which can be computed by adding up the power requirements of the various subsystems, multiplied with the mission time of half an hour (**Req-flight-5**). The required power of all non-propulsive subsystems was found to equal 23 W in total (please refer to all the individual subsystem chapters). The total power consumption per motor for flight speeds up to 14 m/s is about 37 W, as can be found in Section 4.5. This flight speed is the maximum that is allowed to fly with at the competition, since the total momentum of the vehicle may not exceed 20 kg m/s (**req-sys-2**). With a vehicle weight of 14.1 N (1.44 kg), this gives the presented flight speed. The battery will be sized for the power requirement for this maximum speed, although many parts of the competition comprise hovering and slower flight. This might seem as over-designing the battery, but it's rather a safety factor, since the power required for hovering is still as much as 35 W (Section 4.5). Moreover, fast accelerations might also lift the average power consumption to the 37 W.

Summing the required power for four motors and the other subsystems gives a total nominal power consumption of 171 W. This corresponds with the discussed capacity of 85.5 Wh for the mission time of half an hour. Though this capacity is the minimum required in theory, more cells need to be on-board for practical reasons:

- **Aging** Although this has become less of an issue, it still has to be taken into account. Therefore, the total capacity has been taken as the value corresponding to the capacity after 365 discharge cycles and equals 6.0 Wh per cell [147]. This allows the batteries to be able to deliver the power for the complete mission after a year of daily use, for high sustainability. Hereafter, the batteries can still be used, only the flight time will slowly reduce with each discharge cycle.

- **Voltage** The nominal operating voltage also influences the total number of cells, since the desired voltage must equal an integer multiple of the number of batteries connected in series. The total nominal operating voltage has been chosen to be 11.4 V, which corresponds to three packs composed of individual cells providing 3.8 V. Lower operating voltages were found to be insufficient to provide enough power to the motors. Higher voltages would require (more or bigger) voltage transformers (additional mass), to provide the other subsystems with the appropriate voltage.

- **Discharge rate** The higher the discharge rate the lower the total available capacity. From [147] it can be seen that the total capacity of the battery lowers by about 10 % for a discharge rate of 2 C. More losses than this should definitely not occur. The discharge rate is also influenced by the temperature; room temperature is assumed to be representative for the sizing.

- **Safety factor** A safety factor of one extra cell per battery pack for all factors that have not been taken into account or possibly even forgotten. Moreover, this allows for a few extra minutes of flight time to test and setup the vehicle before the start of the mission.

Now, the total number of cells can be calculated with an iterative process:

1. Derive the capacity of an individual cell from [147], assuming 365 discharge cycles and an arbitrary discharge rate.

2. Calculate the total number of cells required by dividing the required capacity of 85.5 Wh by the capacity of an individual cell.

3. Round the total number of cells to an integer multiple of 3, due to the voltage requirement.

4. Calculate the discharge rate by dividing the required power by the total battery capacity and return to step 1.

5. Add one extra cell per battery pack as a safety factor.

It has been found that the total number of cells per battery pack equals 7. This makes one battery pack weigh 0.226 kg. The total battery weight for the three battery-packs in series can than be estimated to weigh about 0.700 kg, including a few grams of weight for wires and connectors. The nominal discharge rate was found to be 1.4 C, which corresponds with a loss of about 5 % in battery capacity. The final available battery capacity was found to be 120 Wh, which is 5.7 Wh per cell. The batteries will be located in the middle of the quadcopter, on top of the main structure.

9.2. Power distribution

The power management and distribution subsystem is comprised of three primary subgroups, namely conversion, regulation and distribution. All these subgroups work in unison to power the various other subsystems and payloads of the UAV such that they may operate effectively (**Req-pwr-1**). As a brief overview, the power conversion unit converts the voltage to the desired amount, the regulation system maintains it at that level and the distribution unit enables the power to reach the relevant part.

Conversion

The power conversion unit is tasked with providing the appropriate voltage to the various subsystems. Given that several components operate at different voltages, this is a very important task. In effect, delivering power at too high a voltage could result in the component over heating, getting damaged or possibly even exploding, while supplying power at too low a voltage will inhibit the subsystem from performing optimally. For instance, the propeller motors operate at 11.4 V while the communication receiver operates at 4.5 V, thereby illustrating the importance of using a converter. In this case a DC-DC converter needs to be used. Both linear and switch mode converters are considered. Although switch mode converters are more efficient, they also create radio frequency noise which could eventually interfere with other systems [149]. For that reason, linear converters shall be used.

Regulation

The regulation unit is designed to automatically stabilise the DC voltage used by the subsystems and payloads. It does so by either using a simple feed-forward design or by using negative feedback control loops. Active regulators will be used as they are more efficient, primarily due to the fact that they do not dump the excess current not needed by the load. Active regulators can be divided into several classes, namely linear series regulators, switching regulators and silicon controlled rectifiers [150]. Linear regulators shall be used as they provide clean power with little noise and relatively low losses at an acceptable level of complexity.

Distribution

The power distribution unit is comprised primarily of switches and wiring. Its function is to safely transport power to the various subsystems and payloads. Switches are basic devices that provide fault protection and turn loads on and off. Therefore, they help in terms of both safety as well as execution of inputs. Electric wires make use of insulated conductors in order to transport power at a certain voltage, which must remain within certain temperature range for safety reasons. The loads on the UAV wires will not be very significant and therefore standard grade EU wires can be used.

Power block diagram

Figure 9.1 illustrates the above mentioned interactions in the form of a block diagram. The left side shows all the power management and distribution features while the right depicts all the subsystems and payloads that need to be powered by the UAV battery.

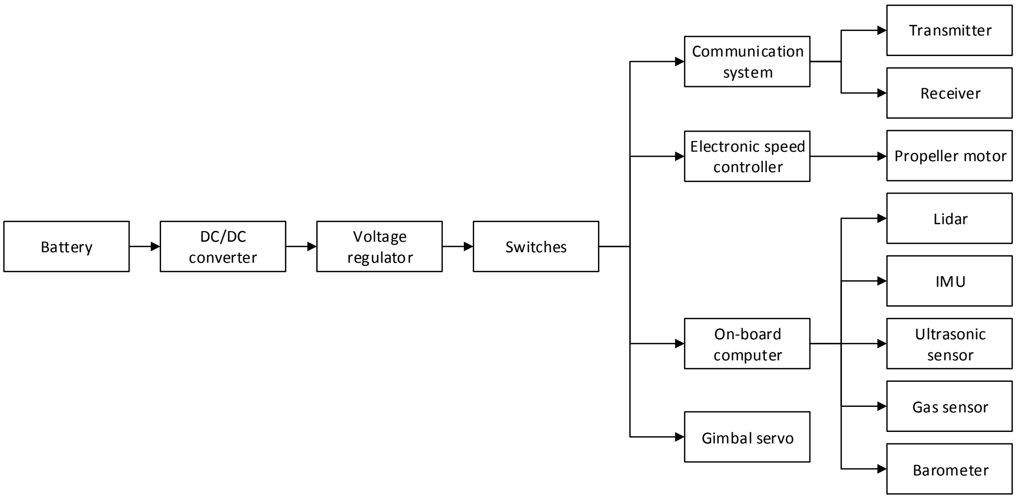


Figure 9.1: Power block diagram

9.3. Recommendations

With the exception of the battery, several components of the power management and distribution subsystem can be evaluated in further detail.

- **DC-DC converter**: As already mentioned, a switch mode controller is the more efficient choice, but due to time restrictions this option was not considered. It would have an impact on other components due to interference and the increased complexities meant that a more detailed analysis would have been required. Moving forward it would be worthwhile to investigate this option in more depth and deal with the interference.
- **Switching regulators**: As with the DC-DC converter, a switching regulator is more efficient but was neglected as there was not enough time to fully evaluate and implement the more complex option.
- **Sustainability**: Although sustainability is already implemented to a certain extent in the power system, in a world where renewable energy sources are becoming increasingly viable options, it would be interesting to investigate some form of renewable based charging system for the UAV battery.

10
GROUND STATION

One of the three major segments of an Unmanned Aerial System (UAS) is the ground station, with the other two being the UAV and the communication link. Therefore, despite the level of autonomy of the MAV, special attention is given to the realisation of the ground station. The considered ground segment will act as a mission planning centre, mission support station and mission control station. Before designing such a crucial segment, a literature study is done and sufficient guidelines are found from [4], [151],[152],[153], [154], [155] and [156] to design the ground segment.

GCS system has its own requirements, constraints and certifications. Furthermore, interactions of the ground system with the air vehicle and communication link impose further requirements on GCS system. All these requirements and interactions are need to be considered in the design of GCS to have a successful realisation UAS. During the design all the requirements were met and they explained in coming sections .

This chapter starts with a section elaborating obtained information from the literature. Following that is the design and the design rationale of the ground station. Cost and mass budget of the different element are listed in the subsequent section. And finally the chapter ends with conclusion and recommendation.

10.1. LITERATURE STUDY AND INVESTIGATION

In order to design the GCS system, firstly literature study was carried out. According to [4, 151, 156] ground system acts as the “hive" of an UAS. GCS manages the deployment, flight and recovery of the aerial system [4]. It receives and processes the data from the internal sensors of the flight systems and from the payload; controls the operation of the payload and provides the interface between the MAV and the operators [4, 151–156].

In [4, 151] hardware architecture of an GCS was explained and several examples were revealed. It was seen that ground system varies from a system of four to five trucks to a small back pack. In literature [152–156], various type of GCS software development and execution have been explained. It was noticed that all the authors prefer having modular architecture over integrated architecture for the software.

It was found in [4, 151] that, for long term application and durability, the ground system needs to have an architecture based on “open", “interoperable" and “common" philosophies. One such architecture is given in Fig. 10.1. Considering full openness and interoperability, it is possible to add, remove or replace one or more elements of the architecture which is highly desirable for long run application and adaptability of the ground station to different systems (commonality).

Following literature study, hands on experience on existing ground system software and software development platform was obtained. Options that have been looked into are QGroundControl and the Paparazzi ground system which are famous, open source and existing ground station software. Microsoft.NET and labVIEW was played with to get experience on GUI development platforms. Outcome from this experimentation are the realisation of genericness of existing platforms and capability of development platform.

10.2. GROUND STATION DESIGN AND RATIONALE

The ground control station (GCS) system had been designed along with the aerial system. Due to time, training and resource constraints it is decided to limit the design to customising a ground system using off-the-shelf hardware rather than developing. For the software, several existing open and closed source Graphical User Interfaces (GUI) and several GUI development platforms have been surveyed and tested. A model of the GUI is presented in the forthcoming section.

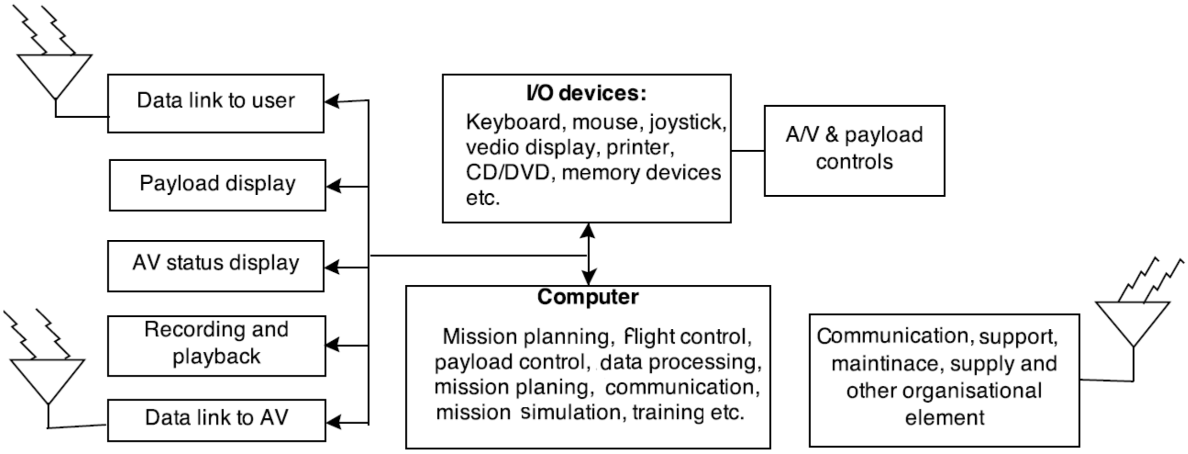

Data link to user
Payload display
AV status display
Recording and playback
Data link to AV
I/O devices:
Keyboard, mouse, joystick, vedio display, printer, CD/DVD, memory devices etc.
A/V & payload controls
Computer
Mission planning, flight control, payload control, data processing, mission planing, communication, mission simulation, training etc.
Communication, support, maintinace, supply and other organisational element

Figure 10.1: Ground system architecture (adapted from [4])

Two iterations were performed in the design procedure. First iteration was solely focused on achieving maximum performance and in the last one was optimising for budget. Because of the page limit, only the final result is being reported. It is worthwhile to note that the difference in these iterations is the mass of the ground station system which decreased and the cost which increased.

10.2.1. Overview

The GCS system serves as the human interface of the UAV and provides additional services for the mission control . The UAS will be accompanied by two people who will act as an operator , observer and ground crew . Additionaly the ground segment houses spare parts , such as extra propellers, fasteners, wires, connectors and other parts to service and replace faulty parts of the UAV. This is required to continue the mission after a failure. Moreover to secure the contingencies, the ground station also provides two check-lists. Maintaining the check-lists will reduce the chances of failure.

Finally the GCS has a high performance laptop acting as the server, central data storage, control unit, mission planning centre, payload data processing and observation unit and the simulation environment. The laptop comes with a custom made screen visor which ensures the readability of the screen at direct sunlight .

10.2.2. Ground station crew

According to requirement **Req-sys-10**, It is required to have personnel of two at the ground station at any time. One acts as the operator and the other as the observer. To use this human resource effectively, it is decided to train both the people to operate the UAS using the ground station computer and the RC controller . Both of them shall have their unique authorising identification and password to access the ground station which satisfies **Req-GS-3**. As both of the personnel will be trained, they can switch their role if desired. Furthermore, both of them will be trained to run maintenance on the MAV and the ground station.

10.2.3. Check-list

One of the functions of the ground station is to manage the deployment and the flight and to retrieve the MAV. Since the MAV is a quad copter which is small compared to other existing MAVs, it can take-off and land in any position of space relieving the ground station of the launching and landing procedure. Regardless of the system, it is desirable to have a system which is safe for the mission and the people around the place it is operating. Therefore it is decided to equip the ground control station with two check-lists which will ensure safe operations. The check-list is provided in Appendix D. It can be seen that all the subsystems which have the probability to fail are checked prior to take-off. There is also a small check-list for the post landing phase which ensures all systems are properly shut down after the mission execution. Performing the check-list and maintaining it during the whole lifetime of the UAS will increase the lifetime of the system and contribute to the sustainable philosophy.

10.2.4. Maintenance equipment

It may happen that the MAV has to perform an emergency landing landing due to hazardous external events. Therefore it is critical to the ground system to carry some spare parts of the MAV, the ground system itself and some repair tools. Taking these parts and tools increase the likelihood of mission success. The following list includes all the spare parts and the tools that will be present in the ground station to support the mission.

Table 10.1: Electronic screw driver trade off

Criteria	**Weight**	Makita TW100DWE	Bosch GSR Mx2Drive	Dewalt DCF895M2
Weight	4	7	7	2
Cost	4	5	8	2
Form factor	2	8	9	7
Total		64	78	30

1. 1 set of 4 legs
2. 1 set of 4 beams
3. 1 set of 4 propellers
4. 1 set of 4 motors
5. 2 × 3 battery packs
6. 2 antennas
7. 1 set of fasteners (100 pieces)
8. spare wires and connectors
9. 1 GSR Mx2Drive Professional electric screwdriver
10. 1 small set standard multi functional tool kit
11. 1 industrial level duct tape

It is highly unlikely that all four beams, motors, legs or propellers will break simultaneously. From the experience of the quad copter community it is most likely that only a couple or one breaks due to high impact or crashing. It is wise to take a couple of legs, beams and propellers but to add redundancy it is decided to take one whole set of these components. Taking redundant beams and legs does not add on cost extensively because the parts are made from sheets bought off the shelf, from these sheets there will be enough left to produce the spare parts. Furthermore their small dimensions and mass allows one to transport them easily while not increase the mass of the GCS noticeably.

Fasteners, bolts, wires and connectors are smallest components of UAV assembly and it is common that they break down. If the broken fasteners or wires are not fixed, it would lead to failure of UAV. In order to have a fail safe system, sets of fasteners, bolts, wires and connectors are included in the ground station.

UAV carries a pack of three for its operation. The battery has been optimised for the mission operation. Therefore it is likely that one set of battery will discharge completely if the mission is extended or if the quad copter does high power demanding manoeuvres. It was identified that there is a requirement of carrying at least one redundant set of quad copter battery but it is decided to take two sets batteries for redundancy.

The antennae are one of the exposed systems of an UAS. They may be damaged during the operations such as during landing. Therefore reasoning similar to batteries is also true for antennas, hence two antennas were taken in the ground station.

The ground system shall house an electric screw driver in order to screw and unscrew the subsystems as fast as possible. In order to select the screw driver, firstly a market survey was done. First criteria of market survey was high longevity as **Req-GS-26** demands atleast two years of functionality. From a quick market analysis it was found that Makita, Bosch and Dewalt provides the most longevity and reliable appliances. Three small electric screw drivers were selected, one from each company and then one of them was selected by having a trade off. Table 10.1 the selected screw drivers and their relative scores. Criteria of weight and form factor was chosen as it determines the transportability and cost was selected to have a budget optimised buying. GSR Mx2Drive Professional electric screwdriver won the trade off because of its lower mass, smaller size and lower cost.

A multi-functional tool kit and some industrial level duct tape are taken on board to cover exposed wires and resolve small structural issues.

10.2.5. Manual control

It is mandatory that the system can switch between automatic and manual flight mode (**Req-sys-7**). In order to meet this requirement it is decided to take an RC controller/transmitter. It is chosen to take a separate RC controller for manual control over manual control by computer because of safety reasons. This manual control system is a separate module, thus the operator will not lose the control over the MAV in case of failure of the ground system computer. If the ground control computer fails or if the on-board FCS does not work as expected or if it is instructed to enter manual flight mode, the operator can switch to the manual flight mode.

To select the RC controller, a market survey was done first. From the market three options were obtained that could meet be used to control the quad copter. The chosen options are the Aurora 9, the Futaba 9C, and the Turnigy 9XR PRO. RC controller was selected by using method of elimination. All three controller have similar functionality but it was found that the Turnigy 9XR PRO Radio Transmitter cheaper in price, have better support from the developers and user community, have better interface and more available than the others. Therefore Turnigy 9XR PRO was selected as the manual control.

10.2.6. Ground system computer

It is required for the ground station to have at least one computer. This is because the ground system needs to have a VGA connection to show the laptop screen to the jury, to analyse images, to control the payload, to program the FCS, to navigate the MAV, to analyse flight data, to store flight and payload data, to debug the FCS, to provide flight simulations and to plan the mission. There are several direct requirements on the computer hardware such as being smaller than 22,000 cm^3, weighing less than 15 kg, having a VGA connection, being readable in direct sunlight and complying with the RoHS (**Req-GS-1, Req-GS-2, Req-GS-4,Req-GS-24, Req-GS-28**). There are also implied requirements on the ground system from its software; such as being able to run the GUI, to process the images, to stitch the images and to store data.

Upon analysing the requirements, the following categories meet the hardware requirements:

1. rugged outdoor laptop.
2. personal laptop enclosed in rugged case with a shade to allow visibility in sunlight.
3. customised net top computer enclosed in a rugged case with a shade to allow visibility in sunlight.
4. rugged outdoor hand held tablet computer.
5. high brightness tablet in rugged case.

In order to select one option, it was decided to reflect on detailed requirements that can be obtained from the requirements stated earlier. To run or develop the ground station software, there are some key hardware specifications such as hard drive capacity, Random Access Memory (RAM) size, processor speed and system architecture. Examples of software that may need to be run on the ground system computer include, Linux, C++, Python, Matlab, QGroundControl, Paparazzi ground system, Microsoft.NET and other standard graphics development programs. For smooth execution of all these software, a suitable system configuration was sought and it is found that any computer with the following minimum hardware specifications will be sufficient:

- Any Intel or AMD x86 processor.
- Hard disk space of 100 GB.
- 2 GB of RAM.
- Intel HD Graphics 4,000 or similar.

Upon getting these system configuration it can be said that available hand held tablet computers will not suffice. Therefore they are eliminated from the design options. Now it is known that all three remaining options come with the required specifications, therefore it is decided to obtain a budget optimised system. First of all the most expensive choice is eliminated, which is a rugged outdoor laptop. Nominally rugged laptops are more expensive than the whole budget of the UAS. Next a market analysis is done for regular laptops and net-top computers. Provided that both will require a rugged case and a shade, only the system cost can be the driving force. It is found that a regular laptop comes with a substantially lower price and better performance. As a result the laptop is selected as ground station computer. The chosen laptop is the ASUS X551CA-SX175H which has better support and less chances of failure than its competitors. The system configuration of the laptop is:

- Intel Core i3-3217U, x86 processor, 1.8 MHz .
- Disk space of 500 GB.
- 6 GB of RAM.
- Intel HD Graphics 4,000.

For redundancy one extra battery for the ground system computer is also taken on board. For protection and visibility in direct sunlight, one rugged case is taken from the market and one sunlight visor is designed which is explained in section 10.2.7.

10.2.7. Transportation

Transportation of the whole UAS is one of the mission segments. During the conceptual phase of the design, it was decided that the system needs to be highly mobile so that it can be carried to different places within the shortest amount of time. Because of that it has been decided that all systems need to be designed such that they can fit into a back pack, which leads to requirements **Req-GS-1** and **Req-GS-2**. The bag pack would be such that it does not cause any damage to the structure during transportation and should have high longevity.

One large high volume tactical bag has been chosen to transport the MAV and the maintenance equipment. The chosen bag is the Rush 72 Backpack from [157]. It conforms to the requirements of transportation and ergonomics. From user experience it is known that it can be used in heavy field duty for more than 2 years. To provide protection against impact, vibrations and components hitting each other during moving, a foam tray has been designed. The foam trays is depicted in Appendix B.

To transport the ground station computer the Pelican 1495CC2 case is chosen. The rationale behind this case is that it is dust and water proof, has a physical lock for extra protection and has a foam padding to protect against unpredicted loads. The check lists and laptop visor will be carried in this case.

The laptop visor is used to be able to see in direct sunlight. After doing a market analysis, the obtained conclusion is to manufacture it in house. The visor is made of paper board and is painted black. The drawing of it is given in Appendix B.

10.2.8. Human machine interface

The human machine interface is the most important part of the GCS system. It is the medium by which the MAV is prepared, controlled and managed. Most of the requirements for the GCS is for the interface. Requirement **Req-GS-3** and requirements **Req-GS-5** to **Req-GS-21** is for the human machine interface.

As mentioned in section 10.1, several ground station software and software development platform have been looked into. It was seen that these interfaces are generic and can account for various systems. To be able to run these software, the user needs experience, therefore they have been discarded from the design options. In order to make the GCS GUI user intuitive and user friendly, it has been decided that the interface for the considered system should be more application based. Therefore designing a dedicated user interface is the best option.

Software design itself is a design procedure, which requires knowledge, expertise and experience. As a result the software was not created, rather a guideline for software design has been made which can be followed to make the user interface. These guideline will support and accelerate the development of GCS software.

Software development

To make the graphical user interface(GUI) budget optimised, it is decided to use open source programs and libraries for its development. For example, the GUI can be developed using Qt library as it allows an easy development by having its signal and slot connection method and easy integration with other C++ libraries [152]; video image treatment can be based on OpenCV, a library from Intel [152]; map integration can be done using Marble library [152]. Finally interprocess integration can be solved using YARP library [152]. The coding for the software shall be done in C# because of its ability to integrate with these libraries. Furthermore there is a large base of C# programmer support on-line.

Software architecture

The software architecture shall have a server/client model. This choice is done to have a modular design and hence give the architecture opportunity to be ‘open", “interoperable" and “common". The server and the clients can be run on same computer similar to the literature [154] and [155].

Server application

The server shall be working when the main application will be started and have minimum user interface with regular user. Full access to server shall require expertise from the developers. It is responsible for communicating with the MAV, maintaining mission data and making the data available for the clients. The server will also record the data for future analysis. The server is as well responsible to send command data to UAV.

Additionally the server shall request to the user to input authentication when it is turned on. In the considered design server is the computer, running the GCS software and connected to data link system. Thus server meets the requirements **Req-GS-3**, **Req-GS-7**, **Req-GS-8**.

Control client

Primary application of control client is to control and monitor the MAV flight. Its purpose is to give the user control and plan the mission and provide feedback to the user about the progress of the mission. Its one of the main feature shall be ability to work both off flight and during flight. During autonomous flight, the control client allows the user to change the flight dynamically using simple point and clicks. It shall have several integrated buttons such as take-off, land, climb, rotate 180 ° and toggle between automatic and manual flight. In the screen, control client shall show real time map where user shall select an area for surveillance and control client would make optimum way points for flight. Control client shall be showing UAV real time position and altitude. It shall also show important sensor values such as heading, flight speed, battery status of the UAV and GCS, flight duration and IMU reading in a virtual instrumentation panel. This kind of platform satisfies the requirements **Req-GS-5, Req-GS-6, Req-GS-9, Req-GS-10, Req-GS-11, Req-GS-12, Req-GS-13, Req-GS-14, Req-GS-20**.

Debug client

After that comes the debug client. It is required for the FCS developer to follow the flight control laws. It needs to have gauges, graphs and detail on the UAV's performance. Using this client, the user will be able to fix and change the control gain and control methodology of the MAV.

Simulation client

Following is the simulation client. It is a stand alone client which the user shall be using to see how the system reacts to a set of mission criteria. Using this client user can do hardware in the loop test without even flying the MAV. It can be also used to train the user. This

Sensor observation client

The sensor observation client shall be a omnipresent client while MAV is flying. This client constantly estimates and observes the status of each sensor. If there is something wrong with the sensor values it gives a visual and audio notifications to the user. It shall ask the user to take appropriate actions. If the appropriate action is not taken, this client shall toggle the emergency landing command in command client.

Payload client

The most important client is the payload client. For outdoor missions, the payload client will be showing the recently taken pictures, any interesting object identified in that frame, the location in the map where the MAV is, its predicted and actual heading. Another thing the payload client shall be doing is streaming the live video from the mission site. For indoor missions, the payload client shall be showing images from the Lidar system, showing the last image and interesting objects. Identified object shall be listed in a list as well. If the object shape is not in system memory, payload client shall ask the user to identify it manually. The payload client will make the data accessible to the server in order to store the data.

The image processing and stitching client takes images from designated locations, processes them, stitches them and then stores this processed data in another designated location from where the user can retrieve it. The SLAM and path planning client will create the full map of the LIDAR measurements of the MAV and plans a path to points of interest.

Client interaction

All clients shall be independent and stand alone. All the clients shall be connected to server application and clients can only interact with each other through server.

10.3. Mass and cost budget

The mass and cost budget of all the physical elements of the ground system are listed in Table 10.2. All the sources of information is given in the table as well.

10.4. Conclusion and recommendation

The ground control station system has been designed, keeping in mind the whole mission scenario. For mission support check lists have been made, redundant components of the MAV and GS have also been taken in ground system. GCS includes high performance and high storage computer on board. The GCS will be transported in one tactical bag and protective case and there will two people with ground system at all time. The capability of GCS software has been modelled and discussed.

Table 10.2: Source, mass and cost budget of the ground control station system

Criteria	Item	mass [kg]	cost [€]	source
Bag	Rush 72 Back pack	2.20	170	5.11 Tactical [157]
	Pelican 1495CC1	3.00	155	Pelican [158]
Ground station computer	ASUS X551CA-SX175H	2.20	485	Tweakers [159]
	Extra 1 battery	0.30	40	Amazon
Electric screw driver	GSR Mx2Drive Professional	0.50	100	Kieskeurig [160]
RC controller	Turnigy 9XR PRO	1.20	75	Hobbyking [161]
Duct tape	3M Heavy Duty Duct Tape	0.12	7	3M [162]
Tool kit	Fixa 17 piece tool kit	1.00	5	Ikea [163]
Fasteners	Fasteners and Bolt	0.08	-	
Spare parts of UAV	4 Spare Beam	0.03	-	UAV manufacture
	4 Spare Propellers	0.02	-	
	4 Spare legs	0.02	-	
	4 spare motor	0.10	100	
	2 Battery	1.40	90	
	Wiring	0.03	5	
	2 set of Antennae	0.04	10	
Ground station Rx/Tx	Receiver and transmitter	0.02	80	
Foam padding	Foam padding for protection	0.20	25	
	Total	**12.46**	**1347**	

Recommendation from the ground system is to buy the components that has been enlisted, assemble the ground system and lastly the software needs to be designed, tested and implemented as mentioned in the discussion.

11

System engineering

Throughout projects such as this one, systems engineering is the essential connection between management and engineering. This chapter comprises the mass and cost budgets for every department and presents the interfaces of the different components. In this chapter the link between the different departments is made.

11.1. Systems architecture

To begin with, a high level systems architecture is created. It gives a rough overview of how hardware components are connected and how software interacts with these. Fig. 11.1 shows the architecture. The rectangular blocks represent hardware, while the round blocks stand for software. A more detailed version of the system architecture can be seen in the hardware and software block diagrams (Fig. 11.2 and 11.3). These figures attempt to give an overview of the hardware and software present in the UAS; it further explains the information that flows between the important hardware and software blocks.

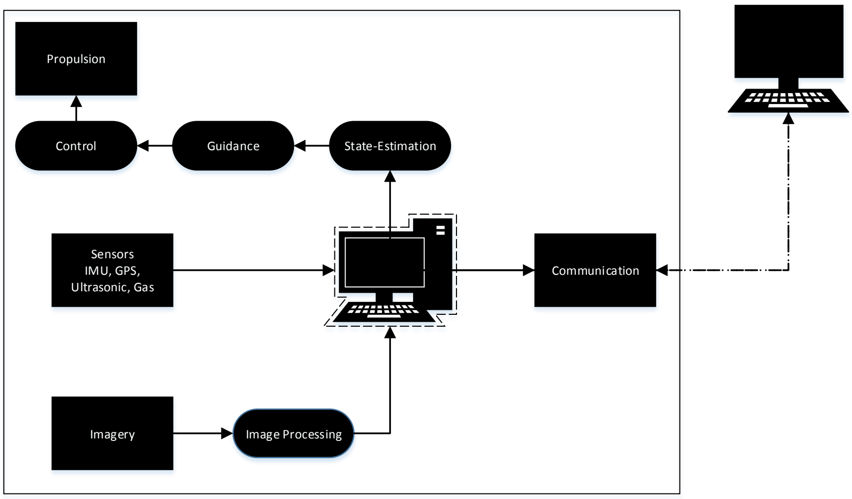


Figure 11.1: High level systems architecture of the entire system

11.2. Departments

The development of the quadrotor system is divided into ten categories, each of which handles a different aspect of the design. Five independent departments are working on one or two categories, which are tabulated in Table 11.1.

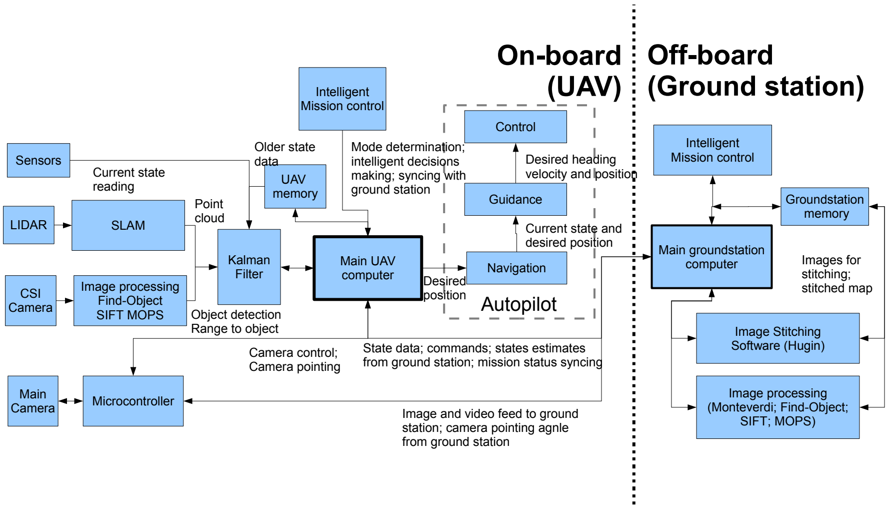

On-board (UAV)
Off-board (Ground station)
Intelligent Mission control
Control
Sensors
Current state reading
Older state data
UAV memory
Mode determination; intelligent decisions making; syncing with ground station
Desired heading velocity and position
Guidance
LIDAR
SLAM
Point cloud
Kalman Filter
Main UAV computer
Current state and desired position
Navigation
Desired position
Autopilot
CSI Camera
Image processing Find-Object SIFT MOPS
Object detection Range to object
Camera control; Camera pointing
State data; commands; states estimates from ground station; mission status syncing
Main Camera
Microcontroller
Image and video feed to ground station; camera pointing agnle from ground station
Intelligent Mission control
Groundstation memory
Main groundstation computer
Images for stitching; stitched map
Image Stitching Software (Hugin)
Image processing (Monteverdi; Find-Object; SIFT; MOPS)

Figure 11.2: Software block diagram of the entire system

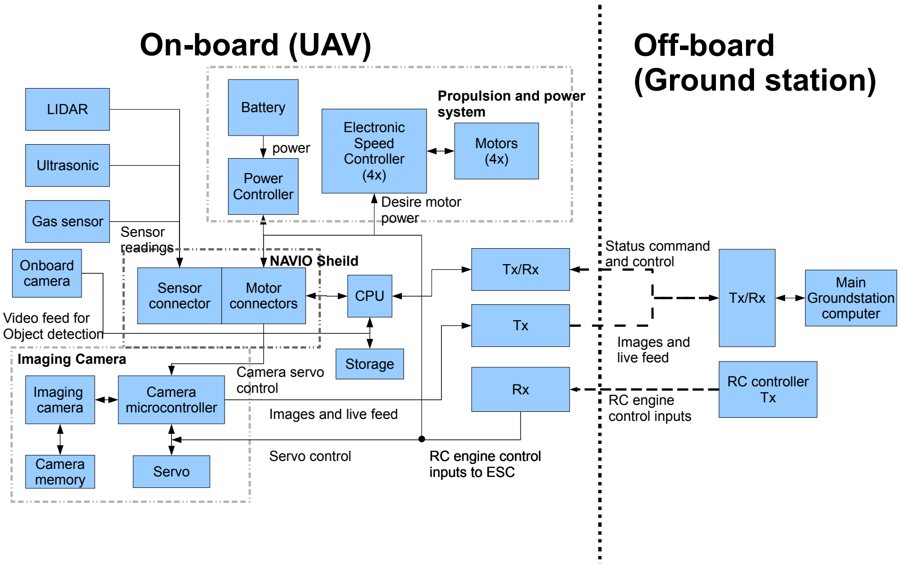


Figure 11.3: Hardware block diagram of the entire system

Table 11.1: Work distribution among departments

Department	Task	Description
1	Structures and materials	Integration of all on board systems, structural design of MAV, material selection, landing gear design
2	Propulsion and aerodynamics	Propeller design, engine selection, aerodynamic performance analysis, battery sizing
3	Communication and ground station	Determination of communication link budget, design of on-board and off-board part of communication system, design of manual controller for quadrotor, ground station design, transportation and maintenance
4	Control and stability	Simulation of quadrotor motion, controller design and tuning, navigation and guidance system development, system stabilisation, stability analysis
5	Imagery and sensors	Mapping, visual navigation, camera, GPS, IMU, CPU, laser range finder (among others) selection, gimbal design for camera, damping and mounting of camera and electronics box

11.3. Interfaces

In this section physical as well as non-physical interfaces are presented. Physical interfaces are mostly driven by structural integrity and physical connection between components, eg. wiring. Non-physical interfaces describe the flow of information/signals between different components. The following describes the interfaces and relations that every engineering department had with other departments.

- Structures and materials
  - Propulsion: Forces, torques, size and placement of engines, controllers and wiring
  - Aerodynamics: Size of propellers, aerodynamic forces and moments
  - Communication: Size and placement of antenna and on-board communication hardware
  - Ground station: -
  - Control and stability: Maximum in-flight accelerations acting on vehicle
  - Imagery: Camera mounting and weight of camera, wiring with other systems
  - Sensors: Placement, weight and size of electronics box, wiring of engines and other systems outside the electronics box
- Propulsion
  - Structures and Materials: Maximum size of engines, engine mounting
  - Aerodynamics: Size of propellers, torques and angular speeds of engines, efficiency of propellers
  - Communication: -
  - Ground station: -
  - Control and stability: Required response speed of engines
  - Imagery: -
  - Sensors: -
- Aerodynamics
  - Structures and Materials: Structural layout
  - Propulsion: Power and angular speed range of engine
  - Communication: -
  - Ground station: -
  - Control and stability: Required forces for manoeuvring
  - Imagery: Frontal area of camera and gimbal system
  - Sensors: -
- Communication
  - Structures and Materials: Room for antenna and on-board communication system
  - Propulsion: -
  - Aerodynamics: -
  - Ground station: Communication links - frequency, link budget
  - Control and stability: Required link budget for telemetry data, link for manual control
  - Imagery: Quality/size of live stream, size of pictures for visual navigation and object detection
  - Sensors: Link budget for telemetry data
- Ground station
  - Structures and Materials:
  - Propulsion: -
  - Aerodynamics: -
  - Communication: Communication links - frequency, link budget
  - Control and stability: Required link budget for telemetry data, link for manual control

  - IMAGERY: Required computing power for off-board SLAM algorithm, storage capability for video and photographic footage
  - SENSORS: -

- CONTROL AND STABILITY
  - STRUCTURES AND MATERIALS: Mass moments of inertia
  - PROPULSION: Engine specifications, thrust vs. angular velocity curve, torque vs. angular velocity curve
  - AERODYNAMICS: Drag coefficients of the UAV
  - COMMUNICATION AND GROUND STATION: Feedback for error estimation in state-estimation from time delayed SLAM algorithm
  - IMAGERY: Pointing direction of camera
  - SENSORS: Data from accelerometers, gyroscopes, laser range finder and GPS

- IMAGERY
  - STRUCTURES AND MATERIALS: Location for camera mounting, type of mounting
  - PROPULSION: -
  - AERODYNAMICS: -
  - COMMUNICATION: Available link budget for images and video stream
  - GROUND STATION: Available processing power for off-board SLAM algorithm
  - CONTROL AND STABILITY: Flight velocity for mapping
  - SENSORS: -

- SENSORS
  - STRUCTURES AND MATERIALS: Available space for electronics box
  - PROPULSION: -
  - AERODYNAMICS: -
  - COMMUNICATION: Link budget for telemetry data
  - GROUND STATION: -
  - CONTROL AND STABILITY: Update frequency of telemetry data, required accuracy of sensors
  - IMAGERY: -

The aforegoing list does not contain all information that is required from every sub-discipline by others. It is only intended to provide the reader with an overview of the communication among the disciplines in order to design a fully functioning system.

11.4. BUDGETS

Every department has a mass and cost budget assigned to them, which is based upon the conceptual design.

Table 11.2: Mass and cost budget for every department

Department	Mass budget [kg]	Cost budget [€]
1	0.150	300
2	0.9	200
3	13	3,000
4	0	0
5	0.1	1200
Total	14.15	4,700
Quadrotor	1.5	1,750

Values presented in Table 11.2 contain all contingencies. Of the 1.5 kg assigned to the quadrotor 50 g are incorporated as contingency. Similarly, the total cost budget of the total system includes € 500 for unforeseen additional expenses. There is not cost and mass assigned to the control and stability department, since the autopilot hardware is included in mass and costs for sensors and imagery department. The 11 kg assigned to department three includes both on-board and off-board systems. Throughout the design process the above shown values had to be adjusted several times. Nevertheless, the total mass of the quadrotor remained constant, as an increase in weight of one component was compensated for with a decrease in weight of another component. The largest adjustment in mass and cost budget had to be made to the ground station, which turned out to be 7 kg heavier than initially assumed.

11.5. Results

This section displays the final results after the detailed design phase. Only the results obtained after the second iteration are listed in Table 11.3.

Table 11.3: Final mass and cost estimation

Group	Subsystem	Mass [kg]	Cost [€]
Prop/Aero	Engine	0.1	43.60
	Propeller	0.01	32
	Batteries	0.7	45
	Controllers	0.036	27.60
Control/Stab	Autopilot/On-board Computer	0.075	210
	Laser Range Finder	0.05	300
Structure	Structural materials (incl. bonding materials)	0.136	275
Imagery	Camera HQ	0.074	170
	Gimbal	0.01	15
	Protection	0.08	200
	Camera LQ	0.003	22
Sensors	Ultrasonic	0.0135	120
	Gas	0.002	7
	Wires	0.05	
Communication/Ground system	Onboard link	0.103	176
	Ground station link	12.46	1347
	Operation		
	Total On-board	1.4425	1,643.20
	Total Off-board	12.46	1,347
	Total	13.9025	2,990.20

All masses and costs are well within limits and as of now the contingencies are not required. Nevertheless, future development can cause dramatic changes to any of the subsystems. For that reason, it is recommended to further update all budgets and contingencies as the final product is further designed in the future. The configuration of the MAV can be found in Appendix B.

11.6. Checking requirements

The requirements listed in Appendix A need to be checked against the actual design. This is summarised in the compliance matrix as seen in Table 11.4. The requirements that are met will have a tick.

Table 11.4: Compliance matrix for the requirements list

Requirement	met	Requirement	met	Requirement	met	Requirement	met
Req-sys-1	✓	Req-pwr-1	✓	Req-GS-21	✓	Req-pl-imag-5	
Req-sys-2	✓	Req-comms-1	✓	Req-GS-22	✓	Req-pl-imag-5-1	
Req-sys-3	✓	Req-comms-2	✓	Req-GS-23	✓	Req-pl-imag-5-2	
Req-sys-4	✓	Req-comms-3	✓	Req-GS-24	✓	Req-pl-imag-6	
Req-sys-5	✓	Req-comms-4	✓	Req-GS-25	✓	Req-pl-imag-6-1	✓
Req-sys-6	✓	Req-struct-1	✓	Req-GS-26	✓	Req-pl-imag-6-2	✓
Req-sys-7	✓	Req-struct-2	✓	Req-pl-sens-1	✓	Req-pl-imag-6-3	✓
Req-sys-8	✓	Req-struct-3		Req-pl-sens-1-1	✓	Req-pl-imag-6-4	✓
Req-sys-9	✓	Req-struct-4	✓	Req-pl-sens-1-2	✓	Req-pl-imag-7	✓
Req-sys-10	✓	Req-struct-5	✓	Req-pl-sens-1-3	✓	Req-pl-imag-8	✓
Req-flight-1	✓	Req-struct-6	✓	Req-pl-sens-1-4	✓	Req-pl-imag-9	✓
Req-flight-2	✓	Req-struct-7	✓	Req-pl-sens-1-5	✓	Req-pl-imag-10	✓
Req-flight-3	✓	Req-GS-1	✓	Req-pl-sens-2	✓	Req-pl-imag-11	✓
Req-flight-4	✓	Req-GS-2	✓	Req-pl-sens-3	✓	Req-pl-SLAM-1	✓
Req-flight-5	✓	Req-GS-3	✓	Req-pl-sens-4	✓	Req-pl-SLAM-2	✓
Req-flight-6	✓	Req-GS-4	✓	Req-pl-proc-1	✓	Req-pl-SLAM-3	✓
Req-cont-1	✓	Req-GS-5	✓	Req-pl-proc-1-1	✓	Req-pl-SLAM-3-1	✓
Req-cont-2	✓	Req-GS-6	✓	Req-pl-proc-1-2	✓	Req-pl-SLAM-3-2	✓
Req-cont-3	✓	Req-GS-7	✓	Req-pl-proc-1-3	✓	Req-pl-SLAM-4	✓
Req-cont-4	✓	Req-GS-8	✓	Req-pl-proc-2	✓	Req-pl-SLAM-5	✓
Req-nav-1	✓	Req-GS-9	✓	Req-pl-proc-2-1	✓	Req-pl-struct-1	✓
Req-nav-2	✓	Req-GS-10	✓	Req-pl-proc-2-2	✓	Req-pl-struct-2	✓
Req-nav-2-1	✓	Req-GS-11	✓	Req-pl-proc-3	✓	Req-pl-struct-3	✓
Req-nav-2-2	✓	Req-GS-12	✓	Req-pl-proc-4	✓	Req-pl-struct-4	✓
Req-nav-3	✓	Req-GS-13	✓	Req-pl-proc-5	✓	Req-pl-struct-5	✓
Req-nav-4	✓	Req-GS-14	✓	Req-pl-imag-1	✓	Req-pl-struct-6	
Req-nav-4-1	✓	Req-GS-15	✓	Req-pl-imag-2	✓	Req-pl-struct-7	✓
Req-nav-4-2	✓	Req-GS-16	✓	Req-pl-imag-3	✓		
Req-nav-5	✓	Req-GS-17	✓	Req-pl-imag-4			
Req-nav-5-1	✓	Req-GS-18	✓	Req-pl-imag-4-1	✓		
Req-nav-6	✓	Req-GS-19	✓	Req-pl-imag-4-2			
Req-nav-7	✓	Req-GS-20	✓	Req-pl-imag-4-3	✓		

12

Verification & validation procedures

12.1. Propulsion and aerodynamics

In designing the UAV, various departments made use of different programs in order to perform more in depth analysis of the system. Some programs were made by the respective departments while others used already existing products. Regardless, keeping good scientific practice in mind, all programs are verified and/or validated in order to lend greater credibility to the obtained results.

12.1.1. Javafoil validation

In order to validate Javafoil, a classic airfoil is analysed using Javafoil, experimental data from a laminar wind tunnel, data from the NACA 824 report and XFLR5. Using the data from all four sources, the lift-drag polar and the $C_l - \alpha$ curves are plotted so as to easily compare the results. The results are presented in Figs. 12.1 and 12.2. It should be noted that Fig. 12.2 does not have experimental wind tunnel data.

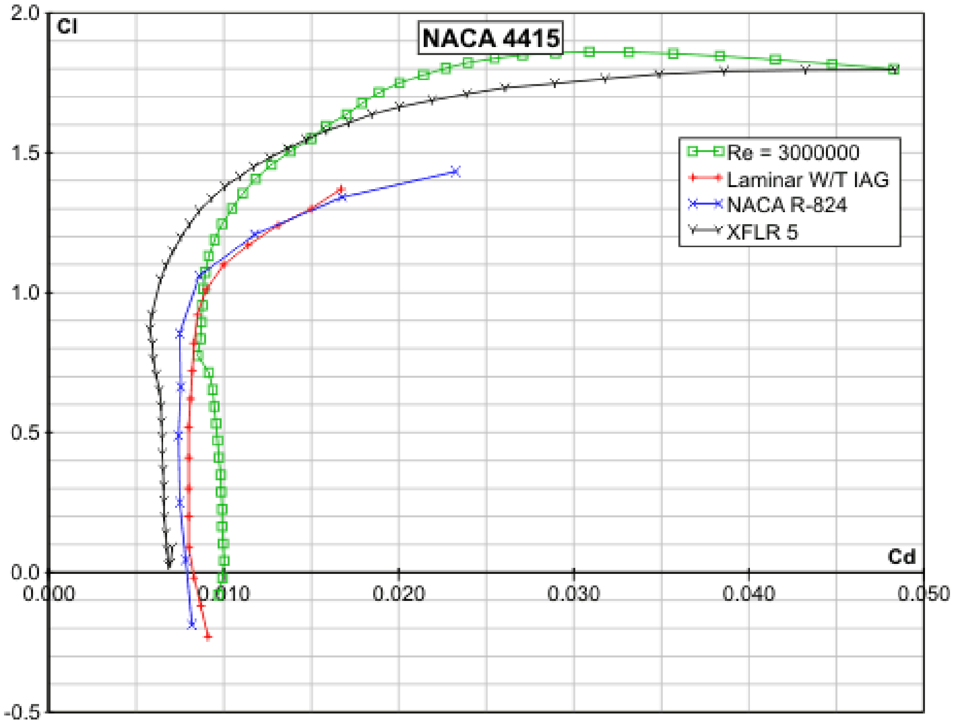


Figure 12.1: Lift-drag polar

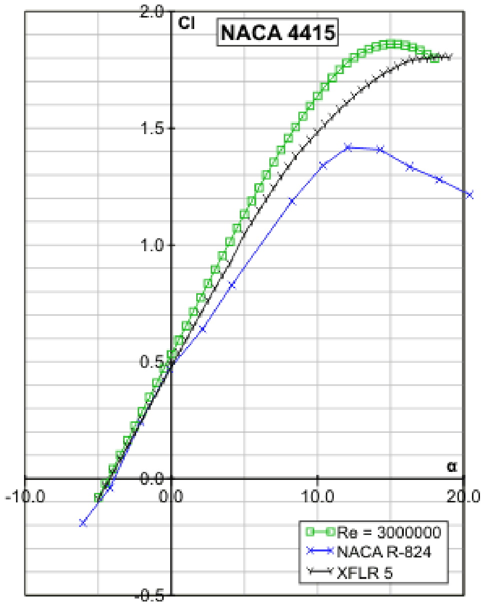


Figure 12.2: $C_l - \alpha$ curve

Thus, when comparing the Javafoil results to data from the NACA report, it can be seen that the maximum lift coefficient is over-estimated by Javafoil. The lift gradient from Fig. 12.2 is also over estimated as boundary layer effects are not accurately modelled. Javafoil also finds the drag coefficient to be higher than the wind tunnel and NACA report data. Despite these discrepancies, it is safe to use Javafoil as an airfoil analysis tool as the magnitude of the errors is reasonable.

12.1.2. QProp validation

QProp is an MIT developed analysis program used for predicting the performance of propeller-motor combinations. It makes use of an extension of the classical blade-element/vortex formulation, developed originally by Betz [164], Goldstein [165], and Theodorsen [166], and reformulated by Larrabee [167]. The extensions include:

- Radially-varying self-induction velocity which gives consistency with the heavily-loaded actuator disk limit.
- Perfect consistency of the analysis and design formulations.
- Solution of the overall system by a global Newton method, which includes the self-induction effects and powerplant model.
- Formulation and implementation of the Maximum Total Power (MTP) design condition for windmills.

In order to validate the program and subsequently legitimise the results obtained, two propellers are analysed using QProp and compared with experimental data [168] and [169]. Efficiency vs advance ratio plots are created for both propellers and the extent to which they differ from the experimental data is recorded. Efficiency is defined as given in Eq. (12.1), where u_e and u_0 refer to the propwash velocity and axial inflow velocity respectively. Advance ratio is defined as the ratio between the distance the propeller moves forward through the fluid during one revolution, and the diameter of the propeller, as characterised by Eq. (12.2). V_a is the speed of advance, n is the propeller's rotational speed and D is the propeller diameter. As can be seen, only the diameter is fixed at 0.2 m, and therefore for a given advance ratio, an infinite combination of V_a and n is possible. However, based on the experimental data being used, it was found that V_a was kept constant and only the rotational velocity was varied in order to change the advance ratio. Therefore, the same is done when analysing the propellers through QProp. Since the airfoils used by the propellers are stated, the relevant airfoil characteristics can be easily extracted using an airfoil database [31] and Javaprop [169].

$$\eta_{prop} = \frac{2}{1+\frac{u_e}{u_0}} \tag{12.1}$$

$$J = \frac{V_a}{nD} \tag{12.2}$$

Figures 12.3 and 12.4 illustrate the experimental and analytical results for the two propellers under consideration. As can be seen, the results are extremely similar. In Fig. 12.3, the largest error in efficiency is of 3 percentage points, an error that is essentially negligible. In the case of Fig. 12.4, the largest error is of 6 percentage points, a number that is somewhat more significant. While this is not ideal, these discrepancies can be explained to a certain extent by the fact that experimental data acquired from a wind tunnel is not perfectly accurate either. In effect, phenomena such as flow interaction with the wall, interference from the measurement equipment, measurement noise and gas compression error can all give rise to incorrect results. Since the offsets between the analytical and experimental data are small, and can be explained by several reasons, QProp can be considered a reliable design tool.

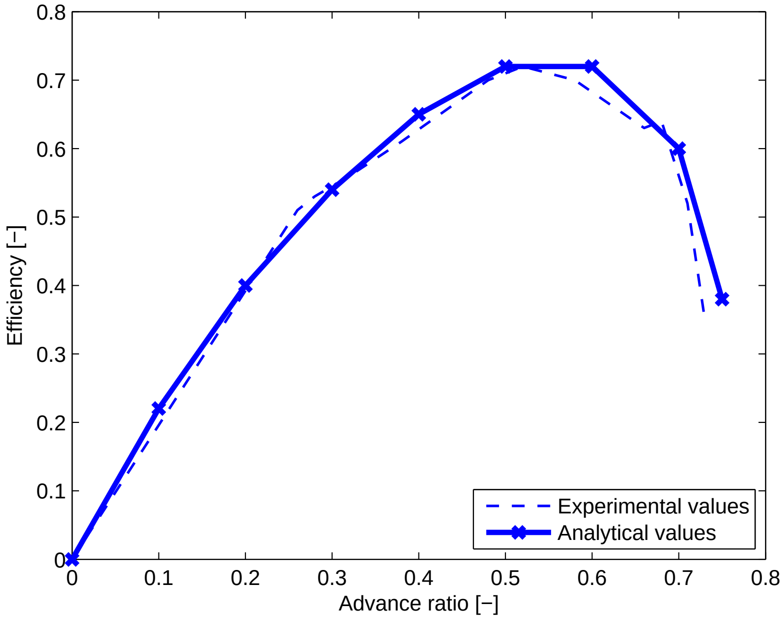


Figure 12.3: Efficiency vs advance ratio propeller 1

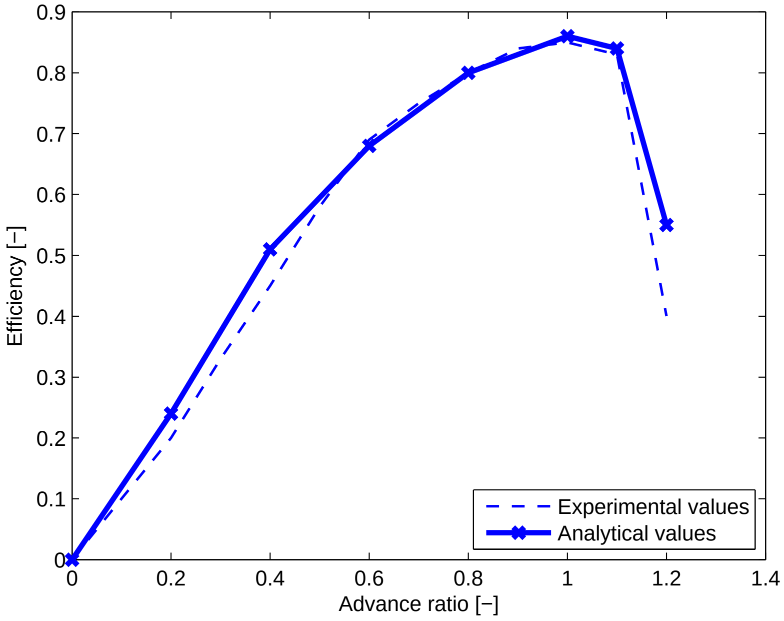


Figure 12.4: Efficiency vs advance ratio propeller 2

12.1.3. Solidworks validation

Another program used by the propulsion and aerodynamics department is Solidworks. Given that this program is commonly used in industry and costs several thousand euros per annum of use, it can already be concluded that the results obtained are of a high quality. However, in order to demonstrate that the program was used correctly and lend greater credibility to the results, the program is validated.

Solidworks makes use of a modified k-ε two-equation turbulence model designed to accurately simulate a wide range of turbulence scenarios in unison with immersed boundary Cartesian meshing techniques [170]. The meshing techniques allow for accurate flow field resolutions even with low cell mesh densities. In addition to turbulence modelling, Solidworks also simulates fluid boundary layer effects. Ordinarily, solving Navier-Stokes equations with a two-equation k-ε requires a very fine computational mesh, which is why a ”wall function´´ approach is adopted. This approach makes use of the fluid-wall frictional resistances and heat fluxes to calculate boundary conditions for solving the Navier-Stokes equations.

In order to validate the program, a 50 mm sphere is placed in a flow of varying speeds, simulating conditions for different Reynolds numbers. The drag coefficient of the sphere is then computed and compared with experimentally obtained data, shown in Fig. 12.5 [171]. In order to find the corresponding Reynolds number, Eq. (12.3) is used. Here Re is the Reynolds number, found to be 10, 10,000 and 100,000 for velocities of 0.003 m/s, 3 m/s and 30 m/s respectively. ρ is the air density, taken to be 1.225 kg/m^3, d the characteristic length taken to be 0.05 m, and μ the dynamic viscosity, taken to be 1.79·10^{-5} kg/ms.

$$Re = \frac{\rho V d}{\mu} \tag{12.3}$$

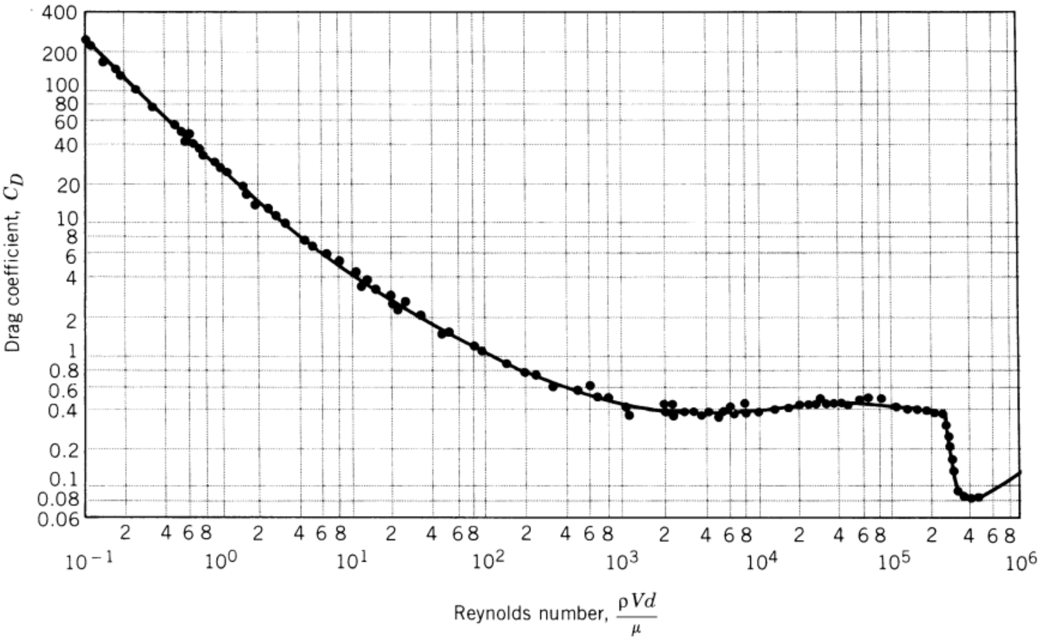


Figure 12.5: Sphere drag coefficient versus Reynolds number

The results obtained from the Solidworks simulation as well as the percentage errors from the experimental values are summarised in Table 12.1. As can be seen, the largest error is only 2.5 %, an offset that can be considered negligible. Furthermore, since the quadcopter body drag during forward flight is estimated using essentially the same flow simulation approach, the results obtained from that process can also be considered accurate.

Table 12.1: Comparison of experimental and analytical sphere drag

Re[-]	Experimental C_D [-]	Analytical C_D [-]	Percentage Error [%]
10	4	4.1	2.4
10,000	0.39	0.40	2.5
100,000	0.41	0.42	2.4

12.2. MMOI TOOL

A tool has been written to calculate the MMOI of the MAV. The tool consists of two classes which contain standard shapes (a cylinder and a box). They can be translated and rotated and subsequently for this new position the inertias around the x,y and z axes can be calculated. There are also a few wrapper classes that contain a number of the standard shapes, such as the arm of the MAV connecting the centre plate with the motor. A wrapper class is a class that does not define a specific shape itself, but contains other shape classes to compose a new component. For the wrapper classes there is an additional wrapper class that represents the MAV, this class incorporates all parts.

12.2.1. CYLINDER AND RECTANGULAR BOX CLASS

The steps taken to verify the cylinder and rectangular box class is by comparing the MMOI to simple cases worked out by hand. The cases are as follows:

- The centre of gravity coincides with the origin
- The cylinder is rotated 90 degrees about the y axis, Ixx and Izz should be switched around
- Move the cylinder to x=1, by means of the parallel axis theorem Izz and Iyy will be increased by 1
- Rotate the cylinder by 90 degrees about the y axis and move it to x = 1. The MMOI around x and z are switched first and then 1 is added to Iyy and the new Izz
- For the rectangular class an additional test can be done, namely a cube should have the same MMOI around the 3 axes

The tests have been conducted and all the tests mentioned were passed.

12.2.2. WRAPPER CLASSES

The wrapper classes gives the MMOI of the whole part, that is done by calculating all the MMOIs of the individual parts and adding them. It has to be checked that the values are added correctly. This has been checked to be the case.

12.3. GNC AND SIMULATION

Here, the verification and validation the software developed in this chapter are discussed.

12.3.1. KALMAN-FILTER

Verification and validation of the Kalman-Filter is solely based on Fig. 7.3 and on comparison to results obtained from the MATLAB code developed by [172]. Looking at Fig. 7.3 shows no unexpected result. The GPS signal is time independent, whilst the integrated position estimation has a time dependent error. The Kalman estimated position is fairly close to the actual position and varying the error in both, accelerometer reading and GPS position, revealed that the Kalman estimate remains closer to the input with a lower error. Due to spatial constraints only Fig. 7.3 is shown in this document. [172] its Kalman-Filter shows very similar results to the ones obtained in this chapter. Even though the validation of the Kalman-Filter was not performed in more depth, the software is assumed to be flawless.

12.3.2. SIMULATION

The simulation is verified in a similar manner as the Kalman-Filter. The results shown in the Figs. 7.8 to 7.11 are just as expected and similar to that presented in [173]. However, further validation and verification work is necessary on the model, as due to time constraints no full validation process could be performed.

13

TECHNICAL RISK ASSESSMENT

Technical risk assessment is important during the early stages of the design process. It enables the systems engineers to detect bottle necks early enough to develop mitigating actions. At this stage of the design process this analysis is performed on a high level, as many details of the final design are still unknown. In this section, first the systems are identified and assigned their respective consequence of failure and likelihood of failure. Next, a risk map is created showing the risk in tabular form.

13.1. ELABORATION OF RISK

Every system is assigned a likelihood, describing the probability of this group failing to design it with sufficient quality, and a consequence, which portrays the effect it will have on achieving the mission objective. Numbers from 1-10 are given, with 1 being highly unlikely and/or has a very little consequence and 10 representing highly likely and a severe consequence. The product of likelihood and consequence gives the risk associated with developing a system. The higher they score on risk, the more attention has to be paid to that part during the design process. The scores are shown in table 13.1.

Flight This part contains everything required for flight: Propulsion system, Navigation system, Control system,Guidance system, Attitude determination system and Communication system.

- PROPULSION SYSTEM The propulsion system is essential for the final product. Without it the MAV is not capable of perform any mission tasks, unless a glider is build. Nevertheless, the impact of failing to design a proper propulsion system mission is severe, as most mission tasks cannot be performed at all, or not sufficiently. Despite this great impact on achieving the mission objective, propulsion systems are fairly well known and much literature is available. To further minimise risk, off the shelf systems are available, making it very unlikely to fail in implementing a sufficiently working system.
- NAVIGATION SYSTEM Designing a navigation system is fairly complex. The associated risk is therefore high. However, in case this group does not manage to properly design this system, manual navigation can be used which decreases the consequence on achieving the mission objective. A lot of navigation systems are on the market though, and implementing one of these minimises the risk and reduces time required to design this system.
- CONTROL SYSTEM This system is essential to the final product, as without it neither autonomous flight nor manually controlled flight is possible. Designing such a system is - due to the complexity of MAV dynamics - a difficult task. As for throttle control systems, there is much literature available on how to control MAVs. Additionally, control systems are available for these small aircraft, which can directly be implemented into the system, which minimises the likelihood of failure.
- GUIDANCE SYSTEM Also the guidance system is complex. It is therefore difficult to properly design such a system. The consequence of a fail is high, since the MAV needs to be guided to reach its destination without crashing. However, the guidance system can be replaced by manual guidance, which makes the system not essential. To impress the jury at the competition it is an advantage to have an autonomous system. Again, there are autopilots available for MAVs, minimising the probability of failing.
- ATTITUDE DETERMINATION Determining the attitude is essential, especially for control. This becomes more important when designing a rotorcraft which is inherently unstable. Failing to determine the

attitude makes it impossible to control the aircraft, especially for the parts of the mission when the aircraft is out of sight and manual control is problematic. However, acceleration sensors, gyroscopes, etc. are available. Mitigation of the risk of failing because of the attitude system is therefore very small.

- COMMUNICATION SYSTEM Communication with the ground station is important in both autonomous and manual flight. Failing to design a stable communication link between ground station and MAV ultimately leads to disqualification from the competition. As for previous systems, off the shelf systems are available which can be implemented, which mitigates the risk.

Mission Here, the functions and components required to fly the mission are treated. These include the camera, mapping, detection of objects/people/road blockages and storage/sending of data and observation.

- CAMERA For mapping and detecting objects any form of camera is essential. Therefore failing to implement one in the MAV has a great consequence on achieving the mission objective. However, plenty of small camera systems are available that are suitable to be implemented for this mission.
- MAPPING Mapping is one part of the entire mission, and requires software and a certain flight path. Depending on camera angle and resolution the flight path can be selected to capture pictures which are then, using software, combined to form a map. Developing a software that is capable of creating a to scale map of the area is a difficult task and associated with a high likelihood of failure. Its consequence is not that severe, as the main mission objective can be achieved without completing this task. The risk can be mitigated by planning sufficient time for the software development.
- DETECTION OF OBJECTS/PEOPLE/ROAD BLOCKAGES Again, this is mainly a software problem that has to be solved, if the mission is to be completed autonomously. In that case, the likelihood of failing is very high. The consequences of this fail are not that high. To mitigate failing either a lot of time developing a recognition software is to be scheduled or this task is performed manually.
- STORAGE/SENDING OF DATA Developing a system capable of storing or sending data does comes with very little probability of failing. There are plenty of systems available to store data. In case the data is sent, this becomes part of the communication system. Even though the risk of failing is very low, if neither sending nor storing data works sufficiently well, this mission task cannot be performed as well and achieving the overall mission goal is unlikely, as none of the mission tasks can be completed.
- OBSERVATION Observation of a building is moderately complex and failure would not lead to failing at impressing the jury. Again, the level of autonomy determines the probability of failing. Therefore, depending on the level of autonomy that is chosen, more time for software development has to be scheduled.

Structure Structure comprises all objectives that have to be fulfilled by the MAV's structure. Those are: Integrity of all components, strength and protection of subsystems.

- STRUCTURAL INTEGRITY This is a very essential function the structure has to complete. If components do not fit it is very likely that competing is not possible. However, proper documentation of the design and using CAD models minimise this risk.
- STRENGTH The strengths of the material and the load carrying capacity of the entire structure is of major importance. Failing to provide the necessary strength ultimately leads to exclusion from the competition. Therefore, mitigating the risk of failure is important and can be achieved by means of FEM analyses of the structural components.
- PROTECTION OF SUBSYSTEMS Protecting the sub systems is of great importance, especially when colliding with objects. Failing at this task has therefore a moderately high impact on the mission objective. Using CAD models provides an easy way to check whether all systems are stored securely within a protecting structure.

Power System The main tasks to be performed by the power system are energy storage and power distribution to all the subsystems.

- ENERGY STORAGE Energy storage is inevitable to perform the mission objectives. Its consequence is therefore rated very high. Storage devices are however available, making it not very likely to fail at implementing a suitable system. The risk of under-designing this system can be mitigated by constantly updating the power budget.

- POWER DISTRIBUTION Power distribution is almost as important as energy storage. If too little power is provided to a system, it will not perform as anticipated, providing too much power will decrease range and endurance and can lead to overheating systems. Its consequence is therefore fairly high. The likelihood of failing in designing a working power distribution system is moderate, due to its complexity. The design has to be tested and redesigned if necessary to not fail at this task.

13.2. RISK MAP

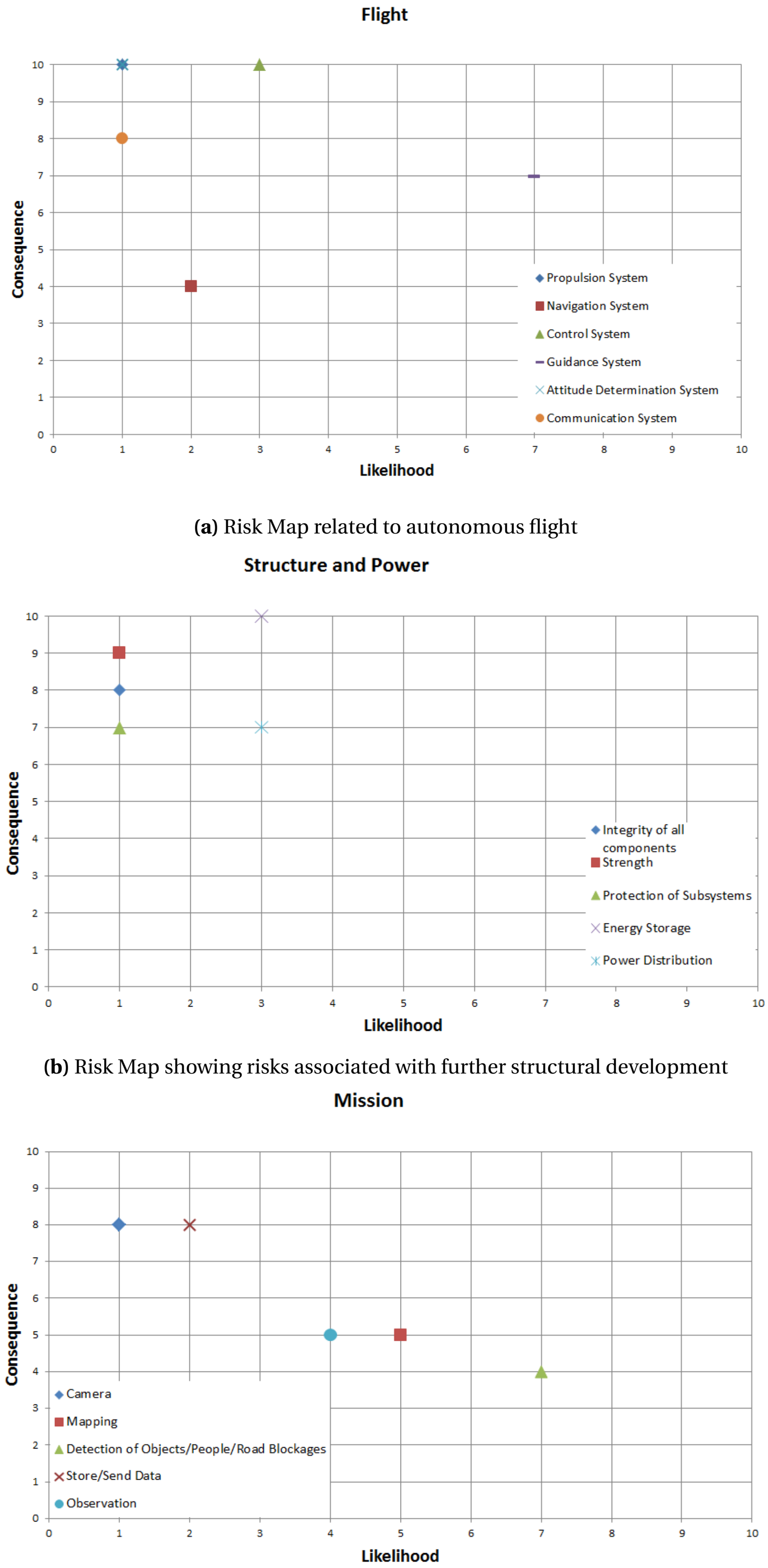
Using the elaboration in the previous subsection, a risk map in tabular form (Tab. 13.1) and as illustration (Fig. 13.1) are created. In the risk map numbers are assigned to likelihood and consequence. The second column from the left of Tab. 13.1 shows the likelihood of failing in designing a certain system before any mitigation actions. Similarly, in Fig. 13.1 the risk is displayed by the location of a system within the graph. The risk increases moving from bottom left (no risk) to top right (very high risk).

The risk map is updated, and the shows now the risks associated with finalizing the design of different subsystems. Since non of the systems is completed and tested yet, all have a remaining risk. Comparing this risk map to versions displayed in prior reports reveals that on average the risks decreased, and symbols shown in Figs. 13.1 moved towards the origin. Additionally, the tabulated form of Figs. 13.1 is given by Tab. 13.1.

Table 13.1: Risks associated with development of subsystems

Fail of Component Development - Quadrotor	Likelihood	Consequence	Risk
Flight			
Propulsion System	1	10	10
Navigation System	2	4	8
Control System	3	10	30
Guidance System	7	7	49
Attitude Determination System	1	10	10
Communication System	1	8	8
Mission			
Camera	1	8	8
Mapping	5	5	25
Detection of Objects	7	4	28
Store/Send Data	2	8	16
Observation	4	5	20
Structural			
Integrity of all components	1	8	8
Strength	1	9	9
Protection of Subsystems	1	7	7
Power System			
Energy Storage	3	10	30
Power Distribution	3	7	21

The risk associated with further developing the quadrotor system decreased significantly due to great technical progress made by this group during the past weeks. The two bottlenecks for future development identified by the risk map are the guidance and object detection systems. Those have not yet been developed in much detail and pose threats on the final design. In particular the guidance system, being of outstanding importance for an autonomously navigation UAV, requires research and developments in much more depth.



(a) Risk Map related to autonomous flight

(b) Risk Map showing risks associated with further structural development

(c) Risk Map related to mission specific tasks

Figure 13.1: Risk Maps with top right being high risk, centre part corresponding to medium risk and bottom left to low risk

14 RAMS CHARACTERISTICS

Now the MAV has been designed it is of great importance to know how it behaves from the user's perspective. The user would like to know about the Reliability, Availability, Maintainability and Safety (RAMS) of the product he just bought. First the reliability aspects will be discussed, then the availability and maintainability and finally the safety.

14.1. Reliability

The reliability of the whole system is dependent on the reliability of the weakest component. As the structure is designed for a maximum deflection and not for the maximum stress, the stresses during normal flight conditions will be much lower than the failure stress. The structure is estimated to have 99 % reliability. The electronics however are more prone to failure as there are a lot of small components; the reliability of the electronics is assumed to be 95 %. The batteries will have the lowest reliability, as it is very easy to charge the batteries in a wrong way, which shortens the working time of the battery. The batteries are estimated to have a reliability of 85 %.

14.2. Availability

The availability concerns the up time of the MAV. As the MAV is modular and spare parts are taken with the ground station, it is very easy to replace a malfunctioning part. This will result in a 99 % availability. However the repair intervals for the structure will be lower as it is simple, while the electronics will need to be checked more often.

14.3. Maintainability

The design is modular, so if one part breaks it can be replaced easily. The loads on the structure will be low if the maximum stress is compared to the stresses caused due to the loads. Thus during normal operations it is very unlikely that there will be some damage. So it is sufficient to check the structure once every few months. However it strongly depends on the environment in which the MAV is flying and also the manoeuvres done.

14.4. Safety

As the MAV is unmanned, no safety measures need to be taken to ensure safety of people on board. So the effects on people and environment in the neighbourhood needs to be accounted for.

The propellers have sharp edges and can be a potential danger. The MAV is equipped with sensors and algorithms to avoid objects. But it is not likely that colliding with the propellers would kill someone, as the propellers are relatively small and light. Furthermore hitting one of the blades will destabilise the MAV such that the MAV will need to stabilise itself again and thus will not continue drilling into the victim.

The batteries are also a source of danger. Although the batteries are safe most of the time, it is not unthinkable that the batteries will be overheated and catch fire.

The MAV itself is a potential danger. It weighs 1.4 kg and if it falls down and crash into someone the injuries may range from little headache to being slammed unconscious.

15

Project Design & Development logic

The project design & development logic, PD&D, shows the activities to be performed in the post-DSE phases of the project. In this case this is up until the IMAV 2014 competition, with a possible extension depending on the performance of the quadcopter at the competition. The quadcopter might need more work, or it might be ready to improve the design for production and use in an actual emergency environment.

In this chapter only the first phase is taken into account, up until the competition. Only the system itself is taken into account, all the logistics surrounding the competition are not taken into account. The work is divided into several workpackages, finishing the design, production, programming and the competition. Workpackage 2 and 3, production and programming can be done simultaneously. A diagram showing the different phases can be found in Fig. 15.1.

In the first workpackage the design as made during the DSE needs to be finished. As is also mentioned in Chapter 5 on structures, a vibrational analysis still needs to be performed, which will probably require a lot of time. After this the detailed attachment of the payload needs to be done. Also a more detailed design of the interfacing of the subsystems is needed, together with a more detailed power system. At the same time two other tasks need to be performed, finishing the autopilot design, which will require a lot of tuning. The other task is to perform a more detailed analysis on the communication, as mentioned in Chapter 8. Finally more detailed manufacturing drawings will need to be made.

After workpackage 1, the second and third worckpackage can be started. The second workpackage revolves around building the actual product. All the materials and payload have to be ordered, afterwhich manufacturing can start. Not all parts however can be ordered, such as the propellors. They have to be manufactured from scratch. So it is also neccesary to order raw materials. The same goes for the sandwich panels, which can probably not be bought at once. Time must me spend to create the sandwhich structure. At the same time the ground station can be produced. When all the structural parts and payload is attached the basic functions can be tested, such as starting the engines. While work is being done on the production, another part of the team can simultaniously work on the programming of the autopilot and payload. The object detection alogorithms for example need to be programmed. After all the programming and manufacturing is done the quadcopter can be put through its paces, testing all its functions.

The last workpackage is the competition itself. The quadcopter will perform at the competition, hopefully impressing the jury in the proces, as that is what it was designed to do.

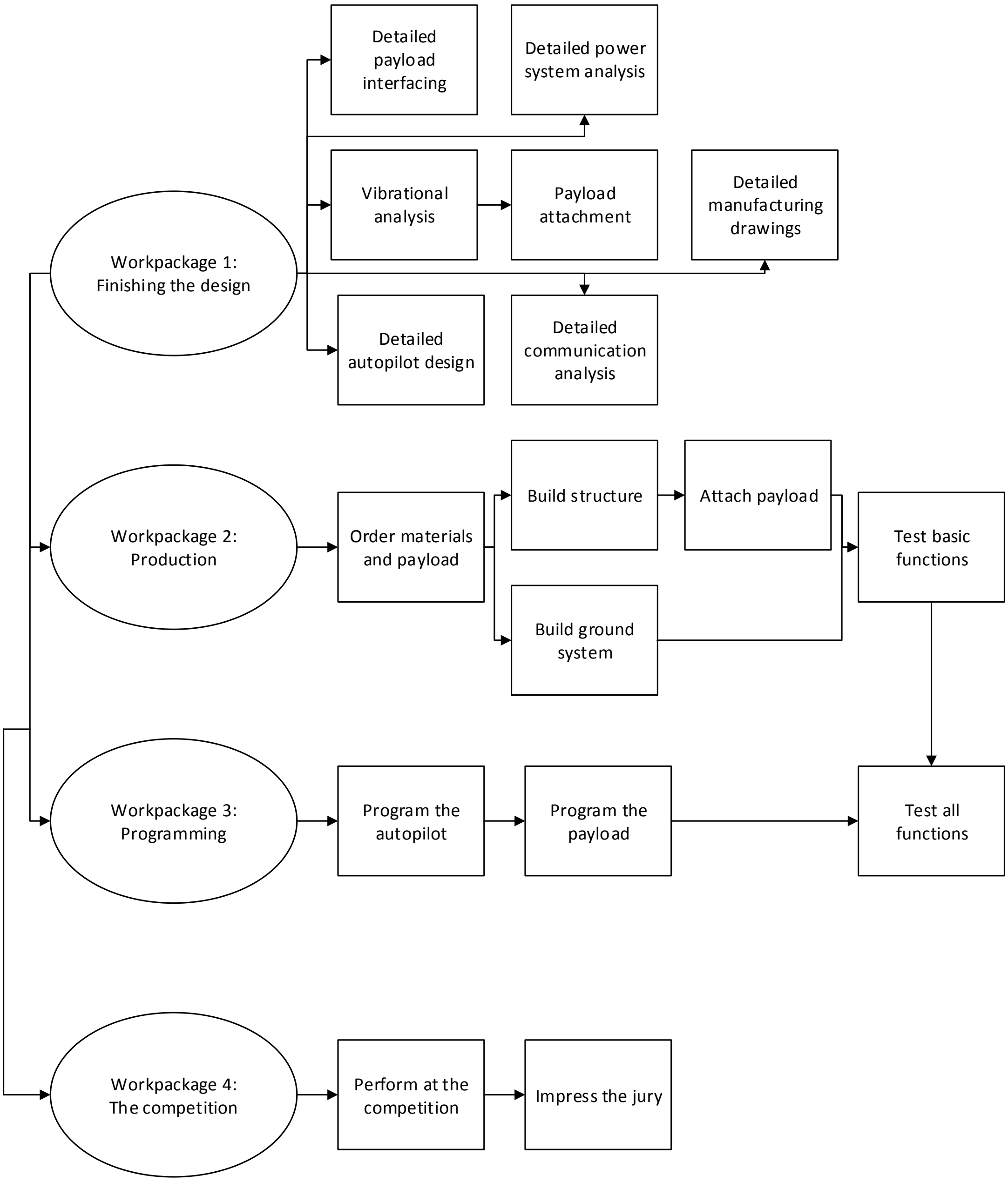

Detailed payload interfacing
Detailed power system analysis
Vibrational analysis
Payload attachment
Detailed manufacturing drawings
Workpackage 1: Finishing the design
Detailed autopilot design
Detailed communication analysis
Build structure
Attach payload
Workpackage 2: Production
Order materials and payload
Test basic functions
Build ground system
Workpackage 3: Programming
Program the autopilot
Program the payload
Test all functions
Workpackage 4: The competition
Perform at the competition
Impress the jury

Figure 15.1: Project design & development logic diagram

16

SUSTAINABILITY

Sustainability has grown into an important issue for research and development. Products and designs are required to be more sustainable, overcoming global challenges such as resource depletion, damage to ecosystems, environmental pollution and rapid population growth. The purpose of sustainability is to ensure a better quality of life for current generations without jeopardizing the standards of living of future generations.

Considering the mission that the UAV has to perform, it is apparent that great steps are already being taken towards a more sustainable future. Instead of using a full sized helicopter to scan the disaster struck area, followed by manned missions to the houses, possibly using motorised vehicles, only a small UAV is used. The UAV will use much less resources when fulfilling its mission. When emergency rescuers eventually have to go towards survivors they can go there quickly and effectively, as they already know the location. This saves a lot of resources. So looking at the overall mission, the use of an UAV for emergency response is already a sustainable one. The UAS can also be transported to the area of deployment in two backpacks, eliminating the need for large transport vehicles.

Not only is the mission of the UAV sustainable, but sustainability has been taken into account even in terms of the design of the UAV. Although sustainability is not an issue for the competition, it is incorporated into the design. When designing the UAV, emphasis was laid on efficiency, repairability and durability.

Efficiency was one of the main design drivers for the selection of the propulsion system. The higher the efficiency of the UAV, the longer it can perform its mission. However, it will also be able to perform more tasks while using less resources. Also because the UAV is powered by an electric propulsion system, no fossil fuels are used to power the UAV. From a sustainable point of view this is very beneficial.

During the structural design of the UAV a lot of attention was given to the repairability of the structure. Allowing broken parts of the structure to be replaced without the need to replace anything else saves a lot of resources. This is again beneficial for the sustainability of the system, as less material is needed in case of an accident. In terms of operations this is also very beneficial of course, as the UAV can be used again a lot quicker.

The last part in which sustainability has been taken into account is the durability of the system. A durable system can last for a longer time, eliminating the need for replacement, which would cost more resources. The UAV is designed to sustain loads higher than it would meet during normal flight. Furthermore, the ground station is also designed with durability in mind.

In conclusion, it can be said that the UAV is a sustainable system, even though it was not designed to be as sustainable as possible. This is mainly due to its mission, but also due to its design being efficient, repairable and durable.

17

Conclusion

The purpose of this report is to outline the design of a new MAV that is capable of exploring an unknown and possibly hazardous area. More precise, it is designed to compute in the IMAV 2014 competition, which simulates such a scenario. The main features of this design are outlined below. Moreover, as motivated throughout the report, the system is expected to be highly reliable, available, maintainable, sustainable and safe.

- Conventional configuration: propellers on top, opposing propellers rotating in same direction, close to symmetric configuration, c.g. in the middle of the craft. The maximum dimension of the vehicle is 61 cm and its total mass 1.44 kg.
- 20 cm diameter custom-made propellers: optimized for providing a hovering thrust of 3.62 N, consuming only 31 W of power. They are operating at a rotational rate of 9 000 to 11 000 rpm on average.
- Turnigy Park 300-1080 motor and 10 Amp. Plush speed controller. Together providing sufficient power at a high efficiency (± 90 %) during all flight stages.
- Maximum thrust-to-weight ratio of 1.75, optimum cruise speed of 20 m/s and climb speed of 6 m/s. Maximum speeds possibly up to 28 and 8 m/s respectively, limited by motor overheating.
- A structure weighing only 136 grams, able to sustain loads of aggressive flight maneuvers, dropping and rough handling. It is also designed to allow for easy maintainability and disassembling for storage.
- On-board sensors providing system-state, acceleration and orientation information. Also included are various observational sensors, from which a “LIDAR system” and a light-weight 12 Megapixel Hero3+ camera are most notable. This camera can achieve a centimeter per pixel resolution at an altitude of over 60 m, without blurring the recordings at a 6 m/s flight speed.
- A communication subsystem for control, data (image) transfer and system state indication. It has a maximum downlink rate of 25 Mbs and a link margin of 50 dB in full sight, allowing high amounts of losses.
- A high-end *Hummingboard processor* with a *Navio autopilot shield* will be used on-board, to guide control, sensory and communication signals. This 1 Ghz quad-core board with 2 GB of RAM provides plenty of jackets and connectors, while still weighing as less as 55 gram.
- Vehicle specific autopilot software, providing fast control and path optimization algorithms. Moreover, algorithms for visual object recognition have been selected and analyzed for their performance.
- Three battery packs, weighing 700 grams in total, providing 120 Wh of capacity. This is enough to fly more than 30 minutes, even after a year of daily use. An extensive power distribution system assures proper operation.
- A ground system with a mass of 12,5 kg, to be able to be carried in a backpack. It not only provides systems for guidance, control and monitoring, but is also equipped with plenty of spare-parts and useful tools.
- Total cost of less than € 3 000,-. From this budget, off-board systems cost up to € 1 400,- and on-board systems about € 1 650,-.

BIBLIOGRAPHY

[1] *Pulsedlight,* http://pulsedlight3d.com/.

[2] J.A.Mulder, W. van Staveren, J. van der Vaart, E. de Weerdt, C. de Visser, A. in 't Veld, and E. Mooij, *Flight dynamics,* (2013).

[3] A. Bachrach, A. de Winter, H. Ruijie, G. Hemann, S. Prentice, and N. Roy, *Range - robust autonomous navigation in gps-denied environments,* , 1096 ().

[4] P. Fahlstrom, *Mission planning and control station,* in *Introduction to UAV Systems*, edited by P. Fahlstrom (Wiley, New York, 2012).

[5] Robotshop, *Unmanned aerial vehicle (uav),* http://www.robotshop.com/en/unmanned-aerial-vehicles-uav.html.

[6] Turboace, *X830 quadcopter,* http://www.turboace.com/turbo-ace-x830-d-f-quadcopter.aspx.

[7] K. Kanzelmeyer, *Uav communication protocal and technology matrices,* (2012) http://prezi.com/jg2nrzloq2gv/uav-communication-protocol-technology-matrices/.

[8] P. Composites, *Mechanical properties of carbon fibre composite materials, fibre / epoxy resin ($120^c ircc$ cure),* (2014), http://www.performance-composites.com/carbonfibre/mechanicalproperties_2.asp.

[9] E. Industries, *Rohacell ig/ig-f properties,* (2014), http://www.rohacell.com/sites/dc/Downloadcenter/Evonik/Product/ROHACELL/product-information/ROHACELL%20IG_IG-F%20Product%20Information.pdf.

[10] A. industries association, *Unmanned aircraft systems: Perceptions and potential,* (2013), http://www.aia-aerospace.org/assets/AIA_UAS_Report_small.pdf.

[11] marketsandmarkets.com, *Unmanned aerial vehicle (uav) market (2013 - 2018),* http://www.marketsandmarkets.com/Market-Reports/unmanned-aerial-vehicles-uav-market-662.html (2013).

[12] P. Murray, *Us military to request up to 5 000 new robots that fit in your rucksack,* http://www.robotreviews.com/news/2009/11/20/dragon-runner-throwbots-to-join-irobot-packbots-in-afghanistan (2011).

[13] J. Baglole, *Talon - multipurpose robot,* http://usmilitary.about.com/od/weapons/a/talonrobot.htm (2014).

[14] J. Lewinski, *Ces 2014: Consumer drones evolve beyond toys, weapons,* http://www.craveonline.com/lifestyle/articles/628899-ces-2014-consumer-drones-evolve-beyond-toys-weapons (2014).

[15] M. Southworth, *The us drone: A fact sheet,* http://cpc.grijalva.house.gov/uploads/Drones_Fact_Sheet_FCNL1.pdf (2012).

[16] U. Europe, *Stanag 4586 vsm c++ sdk,* http://www.uas-europe.se/index.php/products/stanag-4586-vsm-c-sdk (2013).

[17] QGROUNDCONTROL, *Mavlink micro air vehicle communication protocol,* http://qgroundcontrol.org/mavlink/start (2014).

[18] APM, *Apm autopilot suite,* http://ardupilot.com/ (2013).

[19] 3drobotics, *Apm 2.6 set,* http://store.3drobotics.com/products/apm-2-6-kit-1 (2013).

[20] Pixhawk, *Px4 autopilot,* https://pixhawk.org/start (2014).

[21] 3drobotics, *3dr pixhawk,* https://store.3drobotics.com/products/3dr-pixhawk (2013).

[22] PaparazziUAV, *Welcome to paparazzi,* http://wiki.paparazziuav.org/wiki/Main_Page (2014).

[23] PPZUAV, *Hardware store,* https://www.ppzuav.com/nl/ (2014).

[24] CloudCapTech, *Piccolo autopilots - the standard in uas systems,* http://www.cloudcaptech.com/piccolo_system.shtm (2013).

[25] MicroPilot, *Mp2x28 series autopilots,* http://www.micropilot.com/products-mp2028-autopilots.htm (2013).

[26] Digi, *Xtend,* http://www.digi.com/products/wireless-wired-embedded-solutions/zigbee-rf-modules/point-multipoint-rfmodules/xtend-module (2013).

[27] 3drobotics, *3dr radio set,* https://store.3drobotics.com/products/3dr-radio (2013).

[28] Digi, *Xbee 802.15.4,* http://www.digi.com/products/wireless-wired-embedded-solutions/zigbee-rf-modules/point-multipoint-rfmodules/xbee-series1-module (2013).

[29] *IMAV 2014 Competition Rules, version 1.3* (2014), http://www.imavs.org/2014/documents/IMAV2014CompetitionRules_V1_3.pdf.

[30] M. Hepperle, *Javafoil,* (2014), http://www.mh-aerotools.de/airfoils/javafoil.htm.

[31] *airfoiltools.com,* http://www.airfoiltools.com.

[32] B. Montgomerie, *Drag coefficient distribution on a wing at 90 degrees to the wind*, Tech. Rep. (ECN, 1996) http://www.ecn.nl/docs/library/report/1995/c95061.pdf.

[33] R. W. K. Brenden P. Epps, *Unified rotor lifting line theory,* Journal of Ship Research **57** (2013), JOSR.57.4.110040.

[34] *xcoptercalc,* http://ecalc.ch/xcoptercalc.php?ecalc&lang=en.

[35] *Hobbyking.com,* http://www.hobbyking.com/hobbyking/store/__19033__turnigy_park300_brushless_outrunner_1080kv.html.

[36] *Hobbyking.com,* http://www.hobbyking.com/hobbyking/store/__4204__turnigy_plush_10amp_9gram_speed_controller.html.

[37] W. B. I.C. Cheeseman, *The effect of the ground on a helicopter rotor in forward flight*, Tech. Rep. 3021 (Aeronautical research council, 1957) http://naca.central.cranfield.ac.uk/reports/arc/rm/3021.pdf.

[38] *X-3d x-bl quadrocopter, general quadrocopter, x-ufo discussion part 3,* (2009), http://www.rcgroups.com/forums/showthread.php?t=950768.

[39] Syndrones, *Syndrones multi rotor systems,* (2013), http://syndrones.com/.

[40] Steadidrone, *Steadidrone - the ultimate in ready-to-fly advanced multirotor systems!* (2014), http://www.steadidrone.com/.

[41] A. Technologies, *Manufacturer and innovator of micro-uavs - ascending technologies,* (2014), http://www.asctec.de/.

[42] R. Hibbeler, *Mechanics of materials*, 8th ed. (Prentice Hall, 2011).

[43] A. Bedford and W. Fowler, *Engineering Mechanics Dynamics*, 5th ed. (Prentice Hall, 2008).

[44] A. Technologies, *The asctec hummingbird for flight dynamics & swarming,* (2014), http://www.asctec.de/downloads/datasheets/AscTec-Hummingbird_Safetydatasheet.pdf.

[45] E. Composites, *Carbon fibre sheet, pre-preg, double sided high gloss - easy composites,* (2014), http://www.easycomposites.co.uk/Products/Pre-Preg-Carbon-Fibre-Sheet.aspx.

[46] E. Composites, *How to cut carbon fibre sheet and carbon fiber parts (technique & safety),* (2014), https://www.youtube.com/watch?v=EWzN6GZeEzs.

[47] E. Industries, *Rohacell® - high-performance structural foam cores,* (2014), `http://www.rohacell.com/product/rohacell/en/Pages/default.aspx`.

[48] S. Gupte, P. I. T. Mohandas, and J. M. Conrad, *A survey of quadrotor unmanned aerial vehicles,* , 1 ().

[49] J. Wendel, O. Meister, C. Schlaile, and G. F. Trommer, *An integrated gps/mems-imu navigation system for an autonomous helicopter,* .

[50] P.-J. Bristeau, F. Callou, D. Vissière, and N. Petit, *The navigation and control technology inside the ar. drone micro uav,* .

[51] T. Templeton, D. H. Shim, C. Geyer, and S. S. Sastry, *Autonomous vision-based landing and terrain mapping using an mpc-controlled unmanned rotorcraft,* .

[52] S. Grzonka, *Mapping, State Estimation, and Navigation for Quadrotors and Human-Worn Sensor Systems*, Ph.D. thesis, Albert-Ludwigs-University of Freiburg (2011).

[53] S. Gupte, P. I. T. Mohandas, and J. M. Conrad, *A survey of quadrotor unmanned aerial vehicles,* , 1 ().

[54] *Motion tracker technical documentation mt9-b and mt6-b,* (2014), `http://mrl.isr.uc.pt/people/lgm/datasets/manuals/MT9-B%20Technical%20Documentation.pdf`.

[55] S. Bouabdallah, P. Murrieri, and R. Siegwart, *Design and control of an indoor micro quadrotor,* **5**, 4393.

[56] *Technical data mq2 gas sensor,* (2009), `http://www.hobbytronics.co.uk/datasheets/MQ2-sensor.pdf`.

[57] *Techinical data mq-2 gas sensor,* (2009), `http://www.hobbytronics.co.uk/datasheets/LV-MaxSonar-EZ0-Datasheet.pdf`.

[58] *Navio: Autopilot for raspberry pi,* (), `http://www.emlid.com/`.

[59] *High rate imu on raspberry pi with navio,* (), `http://www.emlid.com/high-rate-imu-on-raspberry-pi-with-navio/#more-274`.

[60] 3DRobotics, *3dr pixhawk,* `https://store.3drobotics.com/products/3dr-pixhawk`.

[61] *Px4 autopilot,* `http://pixhawk.org/platforms/multicopters/esc_refresh_rate`.

[62] Paparazzi, *Lisa,* `http://wiki.paparazziuav.org/wiki/Lisa`.

[63] *Asctec hummingbird with autopilot user's manual,* (), `http://www.asctec.de/downloads/manuals/AscTec-Autopilot-Manual-v10.pdf`.

[64] *Nxp (founded by philips) lpc2146,* (), `http://www.keil.com/dd/chip/3934.htm`.

[65] M. Achtelik, S. Weiss, and R. Siegwart, *Onboard imu and monocular vision based control for mavs in unknown in- and outdoor environments,* , 3056 (), `http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=5980343`.

[66] M. Achtelik, S. Weiss, S. Lynen, M. Chli, and R. Siegwart, eds., *Vision-based MAV Navigation: Implementation Challenges Towards a Usable System in Real-Life Scenarios* (RSS Organizing Committee 2012) `http://margaritachli.com/papers/RSS2012paper.pdf`.

[67] S. Weiss, D. Scaramuzza, and R. Siegwart, *Monocular-slam–based navigation for autonomous micro helicopters in gps-denied environments,* **28**, 854–874, dOI: 10.1002/rob.20412.

[68] K. Prashanth, P. Shankpal, B. Nagaraja, G. Kadambi, and S. Shankapal, *Real time obstacle avoidance and navigation of a quad-rotor mav using ical flow algorithms,* **12**, 31, `http://msruas.ac.in/pdf_files/sastechJournals/May2013/5.pdf`.

[69] *Lisa/s,* `http://1bitsquared.com/products/lisa-s`.

[70] *Asctec pelican,* (), `http://www.asctec.de/uav-applications/research/products/asctec-pelican/`.

[71] M. Larabel, *Freescale's i.mx6 soc smacks the old intel atom z530,* (2014), `http://www.phoronix.com/scan.php?page=news_item&px=MTU4MjU`.

[72] J. P. MILLS, I. NEWTON, and R. W. GRAHAM, *Aerial photography for survey purposes with a high resolution small format digital camera,* .

[73] R. E. J. ad Sidney F. Ray, G. G. Attridge, and N. R. Axford, *The Manual of Photography: Photographic and Digital Imaging*, 9th ed. (Focal Press, 2000).

[74] C. user community, *What is chdk?* (2013), http://chdk.wikia.com/wiki/CHDK.

[75] Sony, *Basic specifications,* (2013), http://www.sony.net/Products/di/en-gb/products/ec8t/#specifications.

[76] Canon, *Powershot elph 150 is blue,* (2013), http://www.usa.canon.com/cusa/consumer/products/cameras/digital_cameras/powershot_elph_150_is?selectedName=Specifications.

[77] Adafruit, *Raspberry pi camera board,* (2013), http://www.adafruit.com/products/1367.

[78] P. . T. corp, *Cis vcc-f32fv19cl,* (2013), http://www.phase1tech.com/Cameras/CIS-VCC-F32FV19CL.

[79] A. Vision, *Av2115v1,* (2013), http://www.arecontvision.com/product/MegaVideo+Compact+Series/AV2115v1#Description.

[80] A. Communication, *Axis m10 network camera series,* (2013), http://www.axis.com/files/datasheet/ds_m10_55974_en_hi.pdf.

[81] GoPro, *Hero3+ black edition,* (2013), http://gopro.com/cameras/hd-hero3-black-edition#technical-specs.

[82] Sony, *As15 action cam,* (2013), http://www.sony.co.uk/electronics/actioncam/hdr-as15.

[83] B. S. Ltd., *Rough, but handy hack; accessing the gopro hero3 black bus for component video and audio,* (2013), http://www.benthicscience.com/beta/products/custom-builds/gopro-hero3-bus-access.

[84] GoPro, *Hero3+ black edition field of view (fov) information,* (2013), http://gopro.com/support/articles/hero3-field-of-view-fov-information.

[85] J. Aulinas, Y. Petillot, J. Salvi, and X. Lladó, *The slam problem: a survey,* , 363http://www.researchgate.net/publication/221045411_The_SLAM_problem_a_survey/file/79e4151236ba1087e3.pdf.

[86] *Robopeak rplidar,* http://www.robopeak.com/blog/?cat=8.

[87] *Lidar-lite by pulsedlight,* http://www.dragoninnovation.com/projects/32-lidar-lite-by-pulsedlight.

[88] *I.MX 6Dual/6Quad Applications Processors for Consumer Products* (2014), http://cache.freescale.com/files/32bit/doc/data_sheet/IMX6DQCEC.pdf?fasp=1&WT_TYPE=Data%20Sheets&WT_VENDOR=FREESCALE&WT_FILE_FORMAT=pdf&WT_ASSET=Documentation&Parent_nodeId=1337637154535695831062&Parent_pageType=product.

[89] *Sub-miniature all purpose slip ring capsule model 2040-00,* http://www.electro-miniatures.com/mod2040_00.html.

[90] *Miniature Brushless DC & Specialty AC Motors* (2001), http://www.dprg.org/uploads/escapinfo.pdf.

[91] *Reflective optical sensor tcrt5000,* http://www.dipmicro.com/store/index.php?act=viewProd&productCode=TCRT5000.

[92] Futaba, *Futaba s3156 micro digital hi-torque metal gear servo,* (2013), http://www.gpdealera.com/cgi-bin/wgainf100p.pgm?I=FUTM0656.

[93] goodluckbuy.com, *http://www.goodluckbuy.com/d17d7l21-gimbal-anti-vibration-rubber-ball-120g-loading-red-4pcs-pack.html,* (2013), http://www.goodluckbuy.com/d17d7l21-gimbal-anti-vibration-rubber-ball-120g-loading-red-4pcs-pack.html.

[94] H. Bolandi, M. Rezaei, R. Mohsenipour, H. Nemati, and S. M. Smailzadeh, *Attitude control of a quadrotor with optimized pid controller,* **4**, 35654 (2013), 10.4236/ica.2013.43040, `http://file.scirp.org/Html/12-7900231_35654.htm`.

[95] L. Kleeman, *Understanding and applying kalman filtering,* {`http://www.cs.cmu.edu/~motionplanning/papers/sbp_papers/integrated3/kleeman_kalman_basics.pdf`}.

[96] I. D. Cowling, J. F. Whidborne, and A. K. Cooke, *Optimal trajectory planning and lqr contorl for a quadrotor uav,* .

[97] J.-O. Lee, K.-H. Lee, S.-H. Park, S.-G. Im, and J. Park, *Obstacle avoidance for small uavs using monocular vision,* .

[98] H. Alvarez, L. M. Paz, J. Sturm, and D. Cremers, *Collision avoidance for quadrotors with a monocular camera.* (2013), `https://vision.in.tum.de/_media/spezial/bib/alvarez14iser.pdf`.

[99] Asus, *A55vd,* (2013), `http://www.asus.com/my/Notebooks_Ultrabooks/A55VD/specifications/`.

[100] J. Guivant, E. Nebot, and S. Baiker, *Autonomous navigation and map building using laser range sensors in outdoor applications,* Journal of Robotic Systems **17**, 3817 (2000).

[101] V. Nguyen, A. Harati, A. Martinelli, R. Siegwart, and N. Tomatis, *Orthogonal slam: a step toward lightweight indoor autonomous navigation,* , 5007 (), `http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=4059215`.

[102] S. Grzonka, G. Grisetti, and W. Burgard, *Towards a navigation system for autonomous indoor flying,* , 2878 `http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=5152446`.

[103] V. Nguyen, A. Harati, A. Martinelli, R. Siegwart, and N. Tomatis, *Orthogonal slam: a step toward lightweight indoor autonomous navigation,* , 217 (), autonomous Flight in Unknown Indoor Environments.

[104] A. Bachrach, R. He, and N. Roy, *Autonomous flight in unknown indoor environments,* **1**, 217 (), `http://multi-science.metapress.com/content/80586kml376k2711/fulltext.pdf`.

[105] *Lightweight Communications and Marshalling for Low-Latency Interprocess Communication,* `http://dspace.mit.edu/handle/1721.1/46708`.

[106] E.Balasubramanian and R.Vasantharaj, *Dynamic modeling and control of quad rotor,* **5** (2013), `http://www.enggjournals.com/ijet/docs/IJET13-05-01-013.pdf`.

[107] B. Postle and O. H. Authors, *Hugin - panorama photo stitcher,* (2013), `http://hugin.sourceforge.net/`.

[108] Kolor, *Hugin - panorama photo stitcher,* (2013), `http://www.kolor.com/panorama-software-autopano-pro.html`.

[109] OpenCV, *Opencv (open source computer vision,* (2013), `http://opencv.org/`.

[110] F. S. A. (CNES), *Monteverdi 1&2,* (2013), `http://www.orfeo-toolbox.org/otb/monteverdi.html`.

[111] M. Labbé, *Find object,* (2013), `https://code.google.com/p/find-object/`.

[112] I. Sight Machine, *Computer vision platform using python,* (2013), `http://simplecv.org/`.

[113] B. of Trustees of Leland Stanford Junior University, *Introducing gemident,* (2013), `http://www.gemident.com/`.

[114] G. Nagy, *At the frontiers of ocr,* **80**, 1093 , `http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=156472`.

[115] S. Mori, C. Suen, and K. Yamamoto, *Historical review of ocr research and development,* **80**, 1029 .

[116] R. Plamondon and S. N. Srihari, *Online and off-line handwriting recognition: a comprehensive survey,* **22**, 63 , `http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=824821`.

[117] N. Arica and F. T. Yarman-Vural, *An overview of character recognition focused on off-line handwriting,* **31**, 216 , http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=941845.

[118] Y. Netzer, T. Wang, A. Coates, A. Bissacco, B. Wu, and A. Ng, *Reading digits in natural images with unsupervised feature learning,* http://static.googleusercontent.com/media/research.google.com/nl//pubs/archive/37648.pdf.

[119] P. Sermanet, S. Chintala, and Y. LeCun, *Convolutional neural networks applied to house numbers digit classification,* , 3288 http://ieeexplore.ieee.org/xpl/login.jsp?tp=&arnumber=6460867.

[120] S. Lawrence, C. Giles, A. Tsoi, and A. Back, *Face recognition: a convolutional neural-network approach,* **8**, 98 (), http://ieeexplore.ieee.org/xpl/login.jsp?tp=&arnumber=554195.

[121] S. Lawrence, C. Giles, A. Tsoi, and A. Back, *Traffic sign recognition with multi-scale convolutional networks,* , 2809 (), http://ieeexplore.ieee.org/xpl/login.jsp?tp=&arnumber=6033589.

[122] S. van Berkel and X. van Kooten, *Ground station user interface and data link,* .

[123] P. Van Mieghem, *Data Communications Networking* (Techne Press, 2006).

[124] H. Henniger and O. Wilfert, *An introduction to free-space optical communications,* **19**, 203.

[125] T. Garlington, J. Babbitt, and G. Long, *Analysis of free space communication as a transmission technology,* .

[126] A. Harris, J. Sluss, and H. Refai, *Alignment and tracking of free-space optical communication actions link to uav,* .

[127] A. Nawaz, *802.11 (wireless lan),* (NASA - Exploration Systems Mission Directorate, 2008) http://spacese.spacegrant.org/uploads/Margins%20Module/16.%20Margins_Module_V1.0.ppt.

[128] Y.-L. Theng and H. Duh, *Ubiquitous Computing,* edited by Hershey (Information Science Reference) Chap. 8, p. 117.

[129] *Bluemagnet,* (2009), http://www.bluemagnet.com/xrange2000.html.

[130] J. Proakis, *Interference suppression in spread spectrum systems,* , 259.

[131] E. Coyle and J. Lehnert, *Packet radio and the factory of the future,* , 208.

[132] E. Mikowski, *Fpv video and rc control frequencies,* (2013), http://www.fpvforme.com/fpv-video-frequency/.

[133] C. Anderson, *Some tips on picking frequencies,* (2012), http://diydrones.com/profiles/blogs/some-tips-on-picking-frequencies.

[134] D. Windestal, *Fpv starting guide - fpv setup,* (2009), http://rcexplorer.se/educational/2009/09/fpv-starting-guide/.

[135] A. telecom, *National frequency register,* (2012), http://appl.at-ez.nl/nfr/main_nfr_uk.html.

[136] E. Gill, *Aerospace design and systems engineering elements ii - communication,* (Space Systems Engineering).

[137] T. Schafer, J. Maurer, and W. Wiesbeck, *Measurement and simulation of radio wave propagation in hospitals,* , 792.

[138] P. Ah-Rantala, L. Ukkonen, L. Sydanheimo, M. Keskilammi, and M. Kivikoski, *Different kinds of walls and their effect on the attenuation of radiowaves indoors,* , 1020 ().

[139] P. Ah-Rantala, L. Ukkonen, L. Sydanheimo, M. Keskilammi, and M. Kivikoski, *Indoor propagation comparison between 2.45ghz and 433mhz transmissions,* , 792 ().

[140] R. Gahleitner and E. Bonek, *Radio wave penetration into urban buildings in small cells and microcells,* , 887.

[141] J. Horikoshi, K. Tanaka, and T. Morinaga, *1.2 ghz band wave propagation measurements in concrete building for indoor radio communications,* **VT-35**, 887.

[142] B. Harden, J. Norbury, and W. White, *Estimation of attenuation by rain on terrestrial radio links in the uk at frequencies from 10 to 100ghz,* **2**, 97.

[143] T. Manabe and T. Yoshida, *Rain attenuation characteristics on radio links,* , 77.

[144] I. Poole, *Antenna rf diplexer tutorial,* (2013), http://www.radio-electronics.com/info/antennas/diplexer/antenna-rf-diplexer.php.

[145] *2.4GHz DSSS Radio Module with Integrated Power Amplifier*, Artaflex Inc.

[146] M. A. J.M. Tarascon, *Issues and challenges facing rechargeable lithium batteries,* Nature **414** (2001).

[147] *Panasonic lithium ion battery,* http://www.panasonic.com/industrial/includes/pdf/UF495255STD.pdf.

[148] Y. Yu, C.-H. Chen, J.-L. Shui, and S. Xie, *Nickel-foam-supported reticular coo–li2o composite anode materials for lithium ion batteries,* Angewandte Chemie International Edition **44**, 7085 (2005), http://dx.doi.org/10.1002/anie.200501905.

[149] K. Inou, H. Matsumura, and H. Saito, *Dc/dc converter,* (1987), uS Patent 4,680,688.

[150] V. Blasko and V. Kaura, *A new mathematical model and control of a three-phase ac-dc voltage source converter,* Power Electronics, IEEE Transactions on **12**, 116 (1997).

[151] G. Dagi and S. Marieke, *The mature unmanned aircraft ground control station (muagcs),* in *AirTec 2012, 6th International UAV conference*, AirTec (AirTec, airtec GmbH & Co. KG, Postfach/P.O. Box 700164, D-60596 Frankfurt / Main, Germany, 2012).

[152] D. Perez, I. Maza, F. Caballero, D. Scarlatti, E. Casado, and A. Ollero, *A ground control station for a multi-uav surveillance system,* Journal of Intelligent & Robotic Systems **69**, 119 (2013).

[153] M. Battipede, M. Vazzola, and P. Gili, *Mobile ground station for unmanned elettra-twin-flyer airtship,* pp. 33–1 to 33–10.

[154] D. Lewis, J. McBride, Mark, Mikkelson, and H. Nguyen, *Design of an Autpnomous Unmanned Aerial Reconnaissance Vehicle and Ground Control Station*, Tech. Rep. (Virginia Commonwealth University, 2005).

[155] M. Battle, J. Elmore, S. Lanctot, M. Stuart, J. Philips, M. Elkins, O. Youssef, and B. Young, *Autonomous Arial Vehicle and Ground Control System*, Tech. Rep. (Virginia Commonwealth University, 2012).

[156] Y. Hong, J. Fang, and Y. Tao, *Ground control station development for autonomous uav,* in *Intelligent Robotics and Applications*, Lecture Notes in Computer Science, Vol. 5315, edited by C. Xiong, H. Liu, Y. Huang, and Y. Xiong (Springer Berlin Heidelberg, 2008) pp. 36–44.

[157] 511tactical, *5.11 tactical rush 72 backpack,* (2014), http://www.511tactical.com/rush-72-backpack.html.

[158] Pelican, *1495cc2,* (2014), http://www.pelican.com/cases_detail.php?Case=1495CC2.

[159] Tweakers, *Asus x551ca-sx175h,* (2014), http://tweakers.net/pricewatch/384961/asus-x551ca-sx175h/specificaties/.

[160] Kieskeurig, *Bosch gsr mx2drive professional,* (2014), http://www.kieskeurig.nl/schroefboormachine/bosch/gsr_mx2drive_professional_1,3li-ion2_accus/prijzen/afhalen/1148042.

[161] Hobbyking, *Turnigy 9xr pro,* (2014), http://www.hobbyking.com/hobbyking/store/__51441__Turnigy_9XR_PRO_Radio_Transmitter_Mode_1_without_module_.htmll.

[162] 3M, *3m heavy duty duct tape 3939,* (2014), http://solutions.3m.com/wps/portal/3M/en_US/Adhesives/Tapes/Products/~/3M-Duct-Tape?N=4294924369+5472497&rt=rud.

[163] IKEA, *Fixa 17-piece tool kitk,* (2014), http://www.ikea.com/us/en/catalog/products/00169254/1.

[164] A. Betz, *Airscrews with minimum energy loss*, Tech. Rep. (Kaiser Wilhelm Institute for Flow Research, 1919).

[165] S. Goldstein, *On the vortex theory of screw propellers*, Tech. Rep. (Royal Society, 1929).

[166] T. Theodorsen, *Theory of Propellers* (McGraw-Hill, 1948).

[167] M. OL, C. Zeune, and M. Logan, *Minimum induced loss windmills and propellers,* Journal of Wind Engineering and Industrial Aerodynamics **15**, 1 (1983).

[168] M. OL, C. Zeune, and M. Logan, *Analytical-experimental comparison for small electric unmanned air vehicle propellers,* AIAA **46**, 1 (2008).

[169] T. Theodorsen, G. W. Stickle, and M. Brevoort, *Characteristics of six propellers including the high-speed range*, Tech. Rep. (Langley field, 1937).

[170] D. systemes, *Enhanced turbulence modelling in solidworks flow simulation*, Tech. Rep. (Dassault systemes, 2011).

[171] B. Munson, D. Young, and T. Okiishi, *Fundamentals of Fluid Mechanics* (Wiley, 1990).

[172] B. Dave, *Kalman filter,* (2013), `http://studentdavestutorials.weebly.com/kalman-filter-with-matlab-code.html`.

[173] C. Balas, *Modelling and Linear Control of a Quadrotor*, Master's thesis, Cranfield University (2007).

A

Requirements list

The requirements are listed and numbered for clearer overview and easier documentation.

Req-sys-1 The system shall have a mass smaller than 5 kg.
Req-sys-2 The system shall have a maximum momentum smaller than 20 Ns.
Req-sys-3 The system shall have a maximum dimension less than 150 cm.
Req-sys-4 The name and address of a team member shall be identifiable on the system.
Req-sys-5 The mission shall be able to be aborted at all times.
Req-sys-6 The system shall be able to perform an emergency landing at all times.
Req-sys-7 The system shall be able to switch between automatic and manual flight mode.
Req-sys-8 The system shall not cost more than € 5,000.
Req-sys-9 The system shall comply with the Dutch law.
Req-sys-10 The system shall always be companied by two personnel (operator and observer).

Req-flight-1 The system shall not fly at an altitude higher than 950 ft.
Req-flight-2 The system shall be able to take-off vertically.
Req-flight-3 The system shall have a hover capability.
Req-flight-4 The system shall be able to land vertically.
Req-flight-5 The system shall be able to fly for 30 minutes.
Req-flight-6 The system shall use an electrical propulsion system.

Req-cont-1 The system shall use a control frequency within the allowable bandwidth.
Req-cont-2 The system shall be controllable over at least two different frequencies.
Req-cont-3 The system shall have the option to be manually controlled at all times.
Req-cont-4 The system shall be 3-axis controllable.

Req-nav-1 The system shall transmit its coordinates 10 times per second.
Req-nav-2 The system shall be able to navigate both indoors and outdoors.
Req-nav-2-1 The system shall have an outdoor navigation accuracy of 0.2 m.
Req-nav-2-2 The system shall have an indoor navigation accuracy of 0.2 m.
Req-nav-3 The system shall be able to determine its current state within 0.01 s.
Req-nav-4 The system shall be able to avoid obstacles both indoors and outdoors.
Req-nav-4-1 The system shall be able to bypass outdoor obstacles with a minimum distance of 1 m.
Req-nav-4-2 The system shall be able to bypass indoor obstacles with a minimum distance of 0.2 m.
Req-nav-5 The system shall be able to perform pathway calculations.
Req-nav-5-1 The system pathway calculations shall be able to stay within the allowed flight zone.
Req-nav-6 The system shall be able to map the inside of a house.
Req-nav-7 The GNC subsystem shall require at most 5 W.

Req-pwr-1 The power subsystem shall be able to manage the power distribution as required by the other subsystems.

Req-comms-1 The system shall be able to communicate both indoors and outdoors.

Req-comms-2 The system shall be able to have stable connection at a range of 400 m.
Req-comms-3 The system shall have a maximum download speed of 25 MB/s.
Req-comms-4 The system shall have a maximum upload speed of 0.002 MB/s.

Req-struct-1 All components shall have a maximum deflection of 1 mm at loads twice the MAV weight.
Req-struct-2 The materials used shall be chosen with sustainability in mind.
Req-struct-3 The structure shall not have a mass larger than 0.1 kg.
Req-struct-4 The structure shall be able to sustain maximum loads of 200 N.
Req-struct-5 The structure shall survive a drop from 0.5 m.
Req-struct-6 The structure shall accommodate all subsystems.
Req-struct-7 The structure shall protect the subsystems.

Req-GS-1 The ground system shall have a volume no more than 60,000 cm^3.
Req-GS-2 The ground system shall have a mass no more than 15 kg.
Req-GS-3 The ground system shall be only accessible by authorised personnel.
Req-GS-4 The ground system shall have a VGA connection.
Req-GS-5 The ground system shall be able to plan the path according to current mission off line with in 20 mins.
Req-GS-6 The ground system shall be able to reorientate the mission at any time.
Req-GS-7 The ground system shall accept downlink attitude information data from UAV.
Req-GS-8 The ground system shall have physical memory to store flight and payload data.
Req-GS-9 The ground system shall have a virtual instrument panel.
Req-GS-10 The ground system shall show the position of UAV from the data received from the aerial system.
Req-GS-11 The ground system shall show the attitude of UAV from the data received from the aerial system.
Req-GS-12 The ground system shall show the battery status of UAV.
Req-GS-13 The ground system shall show battery status of ground system.
Req-GS-14 The ground system shall show the flight duration.
Req-GS-15 The ground system shall process images received from UAV.
Req-GS-16 The ground system shall stitch images when required.
Req-GS-17 The ground system shall show all faults in UAV and GS.
Req-GS-18 The ground system shall give error alerts visually and audibly.
Req-GS-19 The ground system shall have simulation environment.
Req-GS-20 The ground system shall have an interactive map.
Req-GS-21 The ground system shall not clutter the GUI of user screen.
Req-GS-22 The ground system screen shall be readable in direct sunlight.
Req-GS-23 The ground system shall have spare structural and electronic parts .
Req-GS-24 The ground system shall have tools to maintain or repair the UAV.
Req-GS-25 The ground system shall comply with environmental standard RoHS.
Req-GS-26 The ground system shall be functional no less than two years.

Req-pl-sens-1 The sensor system shall be able to sense and communicate data to the CPU.
Req-pl-sens-1-1 The sensor system shall be able to sense and communicate to the CPU the acceleration estimates of the UAV on the 3 axis with a precision of up to ± 50 mg at an update rate of at least 35 Hz.
Req-pl-sens-1-2 The sensor system shall be able to sense and communicate to the CPU the angular rotation of the UAV about all 3 axis with within a range of 2,000°/s at an update rate of at least 35 Hz.
Req-pl-sens-1-3 The sensor system shall be able to sense and communicate to the CPU the angular attitude of the UAV in all 3 axis with a precision of 6° at an update rate of at least 10 Hz.
Req-pl-sens-1-4 The sensor system shall be able to sense and communicate to the CPU the GPS coordinates of the UAV with an accuracy of at least 5 m and at a rate of at least 2 Hz.
Req-pl-sens-1-5 The sensor system shall be able to sense and communicate to the CPU the altitude of the UAV with an accuracy of 10 cm.
Req-pl-sens-2 The sensor system shall be able to detect dangerous explosive gases.
Req-pl-sens-3 The sensor system shall be able to detect obstruction within a 120 degrees up to a range of 3 m and with an update of frequency of at least 5 Hz.
Req-pl-sens-4 The sensor system shall sense and communicate to the CPU the real distance to the floor ground from the UAV from an altitude of 3 m with an accuracy 5 cm.

Req-pl-proc-1 A processor board, which meets the set sub-requirements, shall be supplied with or available for the processor.
Req-pl-proc-1-1 The processor board shall have the interfaces required for power supply, integrated sensor

board, additional sensors, data storage, motor control, communication and remaining general purpose connectors.
Req-pl-proc-1-2 The processor board shall be compatible with an integrated sensor board that meets the requirements on sensors.
Req-pl-proc-1-3 The processor board shall have mounting holes to attach it to the structure.
Req-pl-proc-2 The processor shall be equivalent or better than the Intel Atom Z530 processor.
Req-pl-proc-2-1 The processor shall contain all the functions required.
Req-pl-proc-2-1 The processor shall contain a GPU for image processing.
Req-pl-proc-3 The on-board computer[1] shall have a minimum of 1 GB of RAM memory.
Req-pl-proc-4 The on-board computer shall not weigh more than 150 g.
Req-pl-proc-5 The on-board computer shall not cost more than € 500.

Req-pl-imag-1 The imaging subsystem shall have a ground resolution of at least 118.8 mm.
Req-pl-imag-2 The imaging altitude for mission A shall be between 20 and 280 m.
Req-pl-imag-3 The mapping camera lens shall have a range of AoV allowing the UAV to perform the terrain survey without blurring.
Req-pl-imag-4 The imaging subsystem shall be able to interface with the UAS.
Req-pl-imag-4-1 The imaging subsystem shall be able to interface with the UAS allowing the mapping camera to send mapping images and live feed to the ground station.
Req-pl-imag-4-2 The imaging subsystem shall be able to interface with the UAS allowing the UAS to send instructions about the camera operation mode and its tilt to the imaging subsystem.
Req-pl-imag-4-3 The imaging subsystem shall be able to interface with the UAS allowing the imaging subsystem to communicate the visual information to the UAS.
Req-pl-imag-5-1 The mapping camera shall have a pointing accuracy of less than 3.6 °.
Req-pl-imag-5-2 The mapping camera shall have a jitter of less than 0.004 ° to prevent blurring.
Req-pl-imag-6-1 The imaging subsystem shall be able to identify road blockages.
Req-pl-imag-6-2 The imaging subsystem shall have a camera which has a 0.11 °/pix FoV to pixel ratio.
Req-pl-imag-6-3 The imaging subsystem shall have an update frequency of at least 2 Hz.
Req-pl-imag-6-4 The imaging subsystem shall be able to make images of a 3 mm per pixel resolution at a distance of 3 meters for feature recognition.
Req-pl-imag-7 The imaging subsystem shall be able to make images in its flight direction as well as in the NADIR direction.
Req-pl-imag-8 The imaging subsystem shall have a mass of less than 150 grams.
Req-pl-imag-9 The imaging subsystem shall provide images of sufficient brightness.
Req-pl-imag-10 The imaging subsystem shall be able to perform mission element A within 6 minutes.
Req-pl-imag-11 The imaging subsystem shall use Linux compatible software.

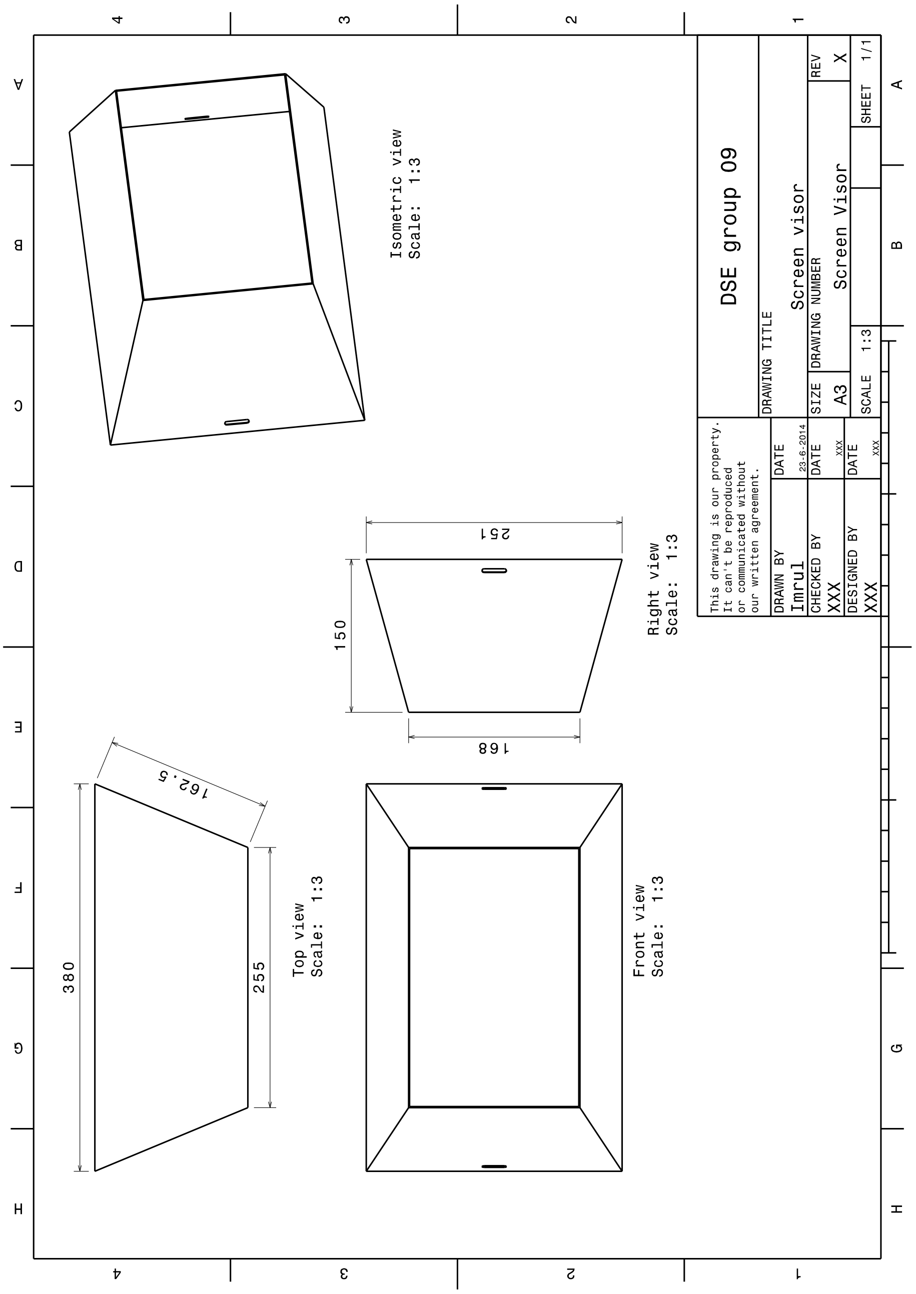
Req-pl-SLAM-1 The SLAM system shall be able to create a map.
Req-pl-SLAM-2 The SLAM system shall be able determine the MAVs location on the map.
Req-pl-SLAM-3 The SLAM system shall provide at least 180 ° range data.
Req-pl-SLAM-3-1 The SLAM system shall provide this data with a resolution of at least 12 measurement points.
Req-pl-SLAM-3-2 The SLAM system shall provide this data with a frequency of at least 0.5 Hz.
Req-pl-SLAM-4 The SLAM system shall have a range of at least 10 m.
Req-pl-SLAM-5 The SLAM system shall have an accuracy of at least 10 cm.

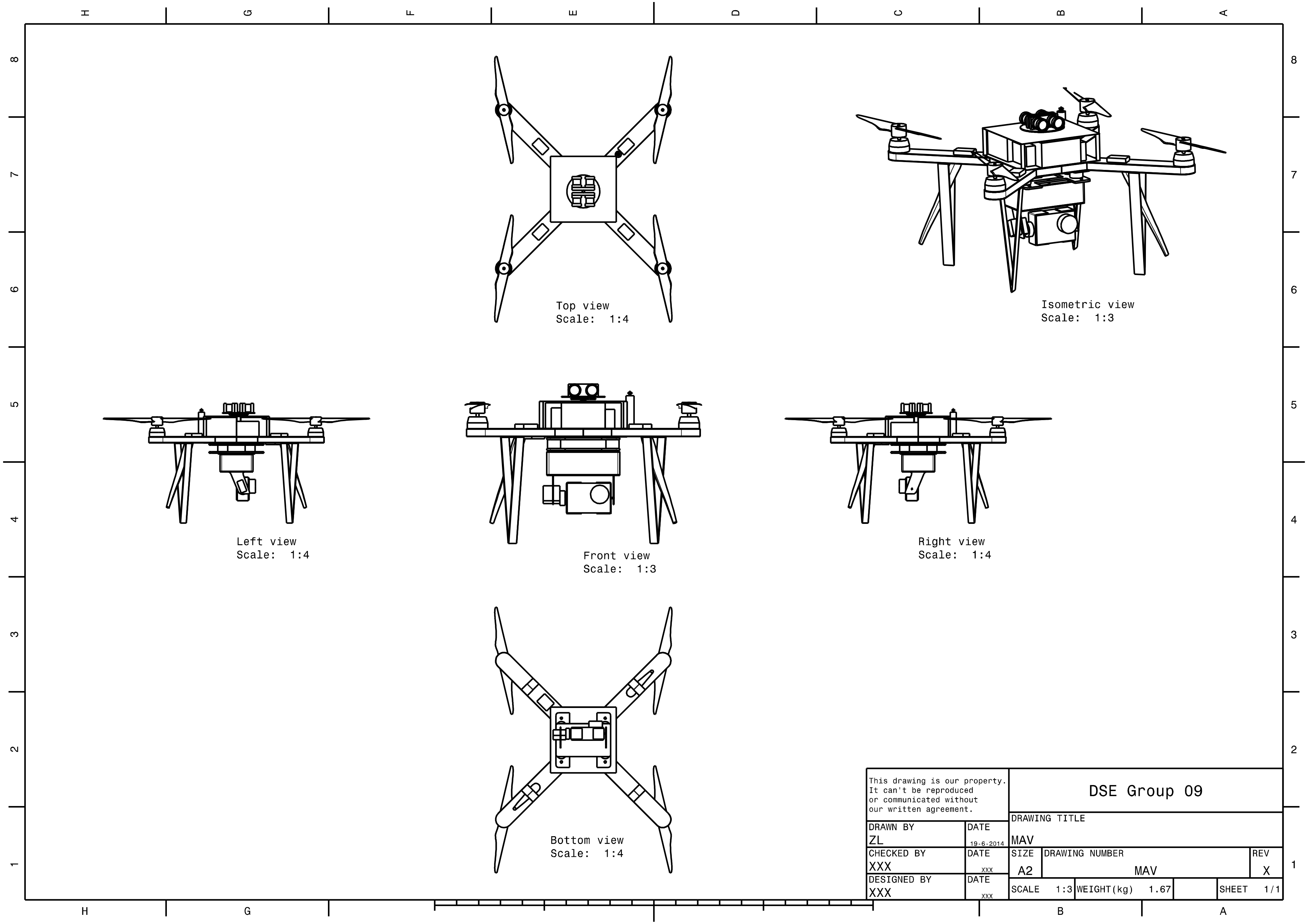
Req-pl-struct-1 The mounting structure shall provide an enclosed housing for all the on board electronics.
Req-pl-struct-2 The mounting structure shall accommodate the main camera as well as the on-board camera.
Req-pl-struct-3 The mounting structure shall provide the imaging camera with a forward FoV of greater than 20 °.
Req-pl-struct-4 The structure shall bear all the loads of the payload.
Req-pl-struct-5 The structure shall be able to provide tilting capability to the payload with a pointing accuracy of less than 3.6 °.
Req-pl-struct-6 The payload structure shall have jitter of less than 0.004 °.
Req-pl-struct-7 The payload structure shall be modular.

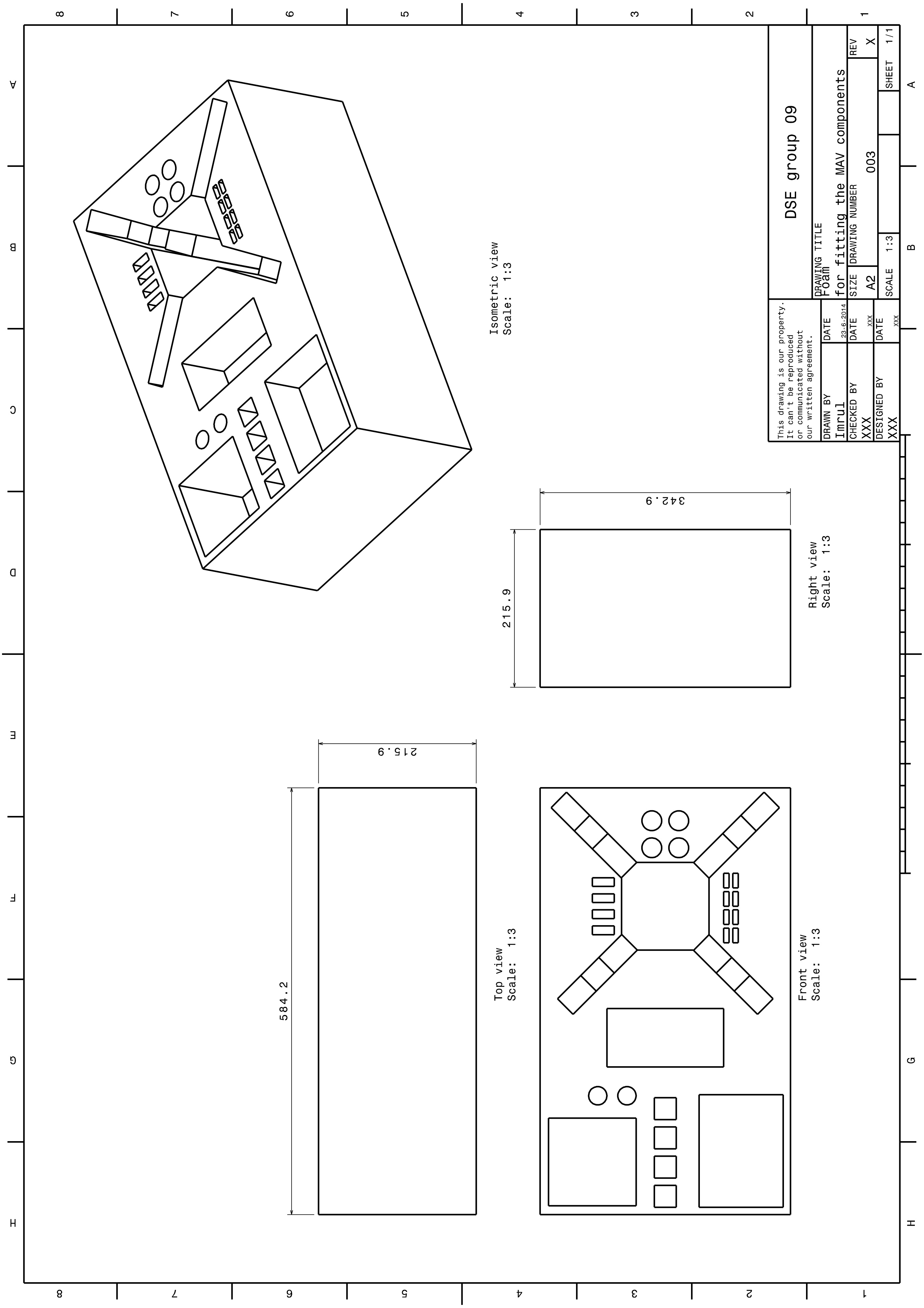
[1] With on-board computer is meant the combination of processor and processor board

B

TECHNICAL DRAWINGS




Top view
Scale: 1:4
Isometric view
Scale: 1:3
Left view
Scale: 1:4
Front view
Scale: 1:3
Right view
Scale: 1:4
Bottom view
Scale: 1:4
This drawing is our property.
It can't be reproduced
or communicated without
our written agreement.
DSE Group 09
DRAWING TITLE
MAV
DRAWN BY
ZL
DATE
19-6-2014
CHECKED BY
XXX
DATE
XXX
DESIGNED BY
XXX
DATE
XXX
SIZE
A2
DRAWING NUMBER
MAV
REV
X
SCALE 1:3
WEIGHT(kg) 1.67
SHEET 1/1
A
B
C
D
E
F
G
H
1
2
3
4
5
6
7
8


Isometric view
Scale: 1:3
Right view
Scale: 1:3
342.9
215.9
Top view
Scale: 1:3
215.9
584.2
Front view
Scale: 1:3
DSE group 09
DRAWING TITLE
Foam
for fitting the MAV components
SIZE
A2
DRAWING NUMBER
003
REV
X
SCALE
1:3
SHEET
1/1
This drawing is our property.
It can't be reproduced
or communicated without
our written agreement.
DRAWN BY
Imrul
DATE
23-6-2014
CHECKED BY
XXX
DATE
XXX
DESIGNED BY
XXX
DATE
XXX

C

LINK BUDGET SPREADSHEET

Table C.1: Link budget between UAV and ground station outdoor

OUTDOOR				
Symbol	**Description**	**I/O**	**Value**	**Unit**
UAV data				
f_d	Frequency downlink	I	2.4	GHz
c	Speed of light	I	0.3	Gm/s
P	Transmit power	I	1	W
P	Transmit power	O	0.00	dBW
L_l	Transmitter line loss	I	-1	dB
G_{ta}	Peak transmit antenna gain	I	7	dBi
L_{tp}	Transmit antenna pointing loss	I	0	dB
EIRP	Equivalent isotropic radiated power	O	6.00	dBW
Propagation path				
d	Propagation path length (slant range)	I	400	m
L_s	Space loss	O	-92.10	dB
L_a	Propagation and polarization loss	I	-0.5	dB
Ground station parameters				
f_u	Frequency uplink	I	2.401	GHz
G_{ra}	Peak receive antenna gain	I	11	dBi
L_{rp}	Receive antenna pointing loss	I	0	dB
DR	Downlink rate	I	25	Mbs
T_s	System noise temperature	I	135	K
Req E_b/N_0	Required energy per bit to noise density	I	7.61	dB
E_b/N_0	Energy per bit to noise density	O	57.77	dB
Margin	Margin	O	50	dB

Table C.2: Link budget between UAV and ground station indoor

	INDOOR			
Symbol	**Description**	**I/O**	**Value**	**Unit**
UAV data				
f_d	Frequency downlink	I	2.4	GHz
c	Speed of light	I	0.3	Gm/s
P	Transmit power	I	1	W
P	Transmit power	O	0.00	dBW
L_l	Transmitter line loss	I	-1	dB
G_{ta}	Peak transmit antenna gain	I	7	dBi
L_{tp}	Transmit antenna pointing loss	I	0	dB
EIRP	Equivalent isotropic radiated power	O	6.00	dBW
Propagation path				
d	Propagation path length (slant range)	I	400	m
L_s	Space loss	O	-92.10	dB
L_o	Other loss	I	-35.50	dB
L_a	Propagation and polarization loss	I	-0.5	dB
Ground station parameters				
f_u	Frequency uplink	I	2.401	GHz
G_{ra}	Peak receive antenna gain	I	11	dBi
L_{ra}	Receive antenna pointing loss	I	0	dB
DR	Downlink rate	I	25	Mbs
T_s	System noise temperature	I	135	K
Req E_b/N_0	Required energy per bit to noise density	I	7.61	dB
E_b/N_0	Energy per bit to noise density	O	22.27	dB
Margin	Margin	O	15	dB

D

Check-list

Quad operation check-list

Operator Name :
Observer Name :
Date : / /
Time : : hr
Flight Number :

No.	Action	$\surd/\times$	Remarks
Pre-flight Assembly			
1	Assemble quadcopter	□	
2	Perform pre-flight check-list	□	
Flight			
3	Review safety procedure	□	
4	Field operations procedure completed	□	
5	FCS ON	□	
6	Payload ON	□	
7	TX ON	□	
8	RX ON	□	
9	Propeller functioning in auto mode	□	
10	Propeller functioning in manual mode	□	
11	Radio range check	□	
12	TX in range check in low power mode	□	
13	Transmitter OFF	□	
14	All equipment OFF	□	
If not the first flight, following steps needs to be taken			
15	Install payload and ready for flight	□	
16	Verify RX	□	
17	Verify FCS	□	
18	Check battery	□	
19	Verify transmitter frequency	□	
20	Verify ground system software	□	
21	Verify plugs and wires secure	□	
22	Full power check	□	
23	Final propeller check	□	
24	Final payload check	□	
25	Verify GCS on and functioning	□	
26	Verify FCS on and functioning	□	
27	Verify manual control ready	□	
28	Verify data logging is on	□	
29	Verify timer running	□	
Post flight and shut down			
30	Throttle cut OFF	□	
31	Engine OFF	□	
32	Payload OFFF	□	
33	Receiver Off	□	
34	Transmitter OFF	□	
35	FCS OFF	□	
36	Note timer value and complete logbook entry for flight	□	

Quad Pre-flight check-list

Operator Name :
Observer Name :
Date : / /
Time : : hr
Flight Number :

No. Action

Ground station

1 GCS ON
2 GCS GUI visible
3 Check and record power of GCS
.... % .... mins remaining
4 All GCS client open and running
5 Image stitching software running
6 Image processing software working
7 Maintenance material out and ready
8 RC controller functional
9 Check and record power of RC controller
.... % .... mins remaining

Aerial system

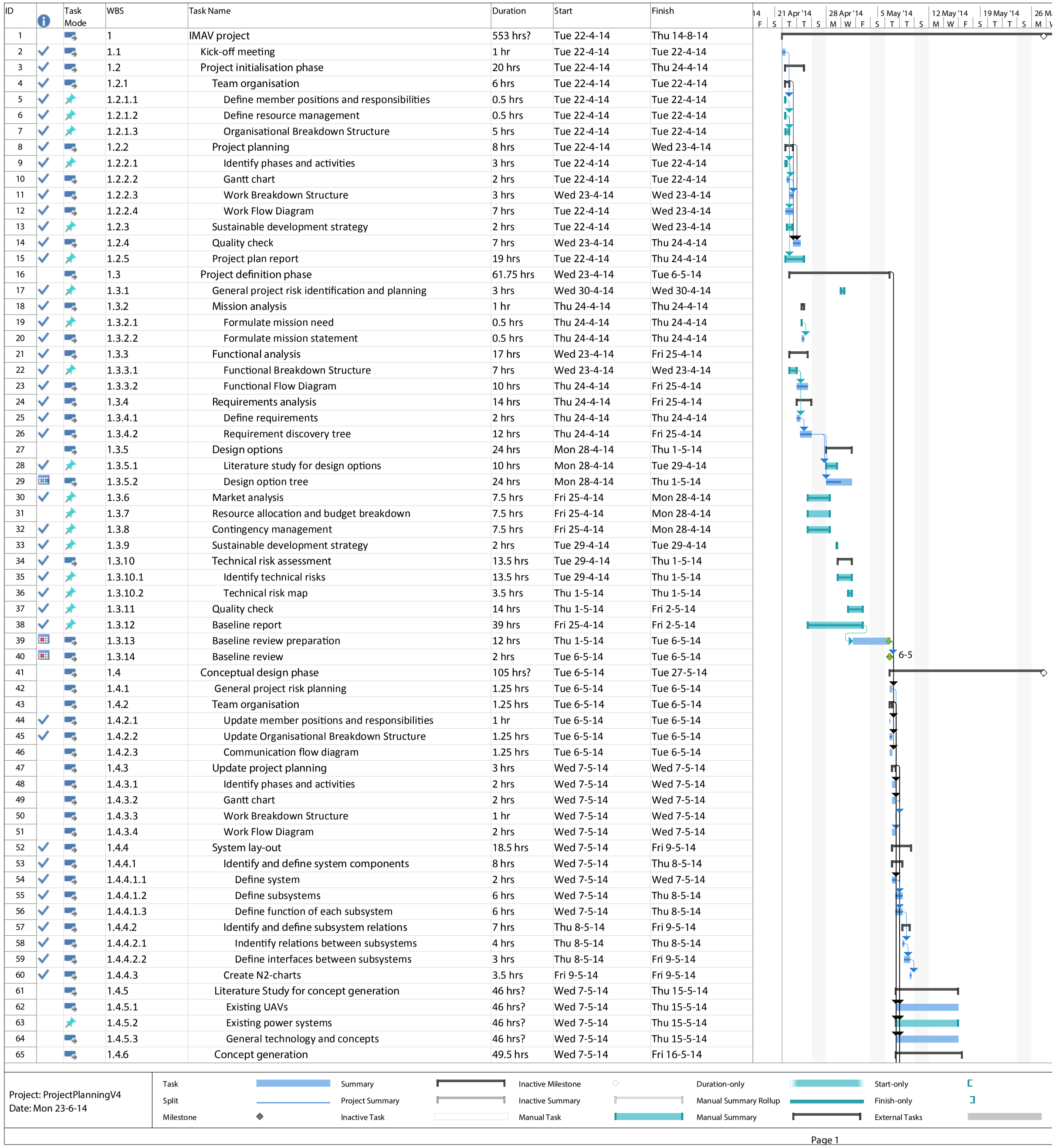
10 Check and record battery voltage
.... V [12 V at pre-flight]
11 Propeller connection (tighten)
12 Motor clamp (tighten)
13 Landing gear bolts (tighten)
14 Payload bay tighten with central plate
15 FCS connection secured
16 Payload connection secured
17 Electronic box connection secured
18 FSC ON
19 Internal camera ON
20 Main camera ON
21 Navio ON
22 Gas sensor ON
23 Pressure sensor ON
24 Antenna properly installed
25 Connectivity secured
26 UAV and GCS data link established
27 Transmitter ON
28 Receiver ON
29 Transmitter ON
30 Radio range check

Take-off

31 Observer ready
32 Pilot ready
33 Request for take-off accepted

E

Gantt chart

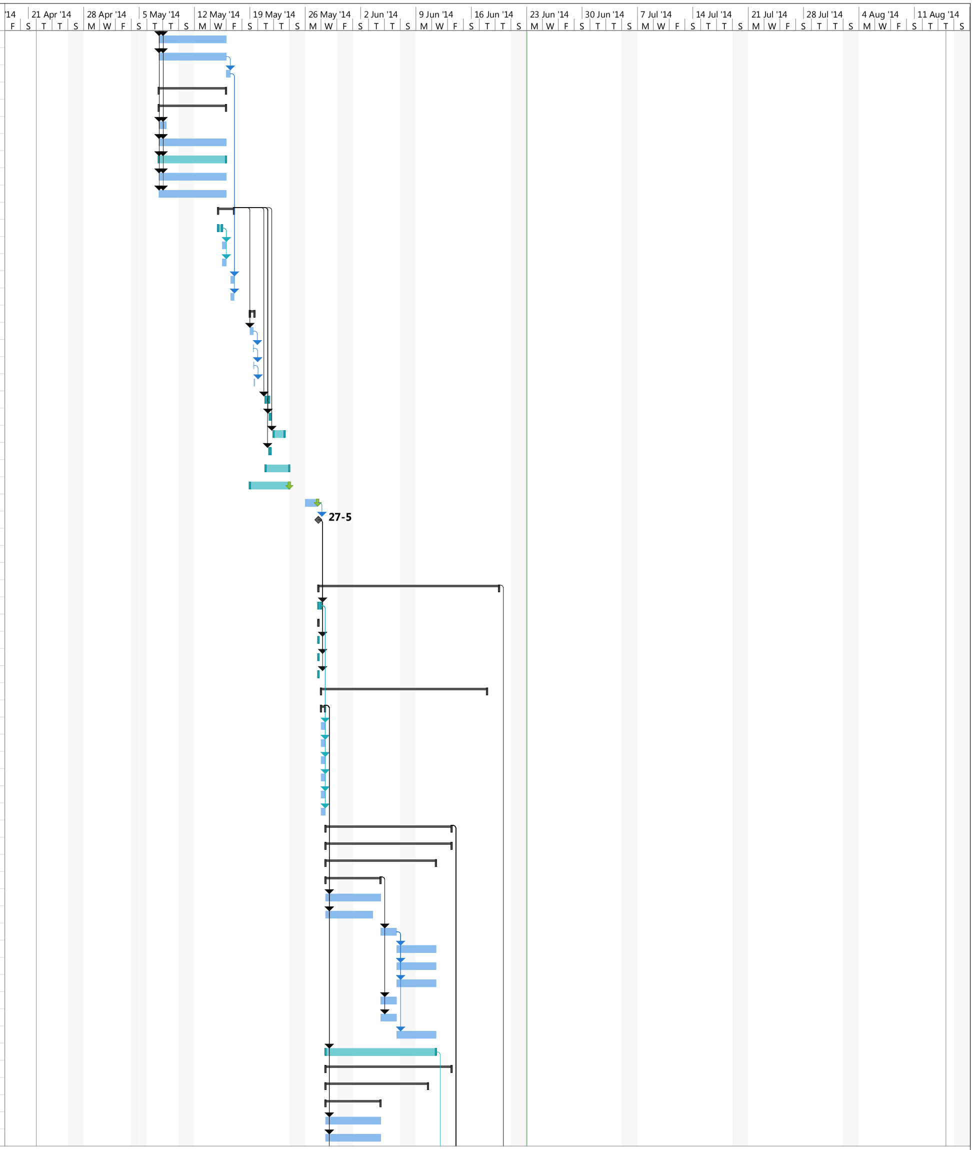


ID	ℹ	Task Mode	WBS	Task Name	Duration	Start	Finish
1		➡	1	IMAV project	553 hrs?	Tue 22-4-14	Thu 14-8-14
2	✓	➡	1.1	Kick-off meeting	1 hr	Tue 22-4-14	Tue 22-4-14
3	✓	➡	1.2	Project initialisation phase	20 hrs	Tue 22-4-14	Thu 24-4-14
4	✓	➡	1.2.1	Team organisation	6 hrs	Tue 22-4-14	Tue 22-4-14
5	✓	📌	1.2.1.1	Define member positions and responsibilities	0.5 hrs	Tue 22-4-14	Tue 22-4-14
6	✓	📌	1.2.1.2	Define resource management	0.5 hrs	Tue 22-4-14	Tue 22-4-14
7	✓	📌	1.2.1.3	Organisational Breakdown Structure	5 hrs	Tue 22-4-14	Tue 22-4-14
8	✓	➡	1.2.2	Project planning	8 hrs	Tue 22-4-14	Wed 23-4-14
9	✓	📌	1.2.2.1	Identify phases and activities	3 hrs	Tue 22-4-14	Tue 22-4-14
10	✓	➡	1.2.2.2	Gantt chart	2 hrs	Tue 22-4-14	Tue 22-4-14
11	✓	➡	1.2.2.3	Work Breakdown Structure	3 hrs	Wed 23-4-14	Wed 23-4-14
12	✓	➡	1.2.2.4	Work Flow Diagram	7 hrs	Tue 22-4-14	Wed 23-4-14
13	✓	📌	1.2.3	Sustainable development strategy	2 hrs	Tue 22-4-14	Wed 23-4-14
14	✓	➡	1.2.4	Quality check	7 hrs	Wed 23-4-14	Thu 24-4-14
15	✓	📌	1.2.5	Project plan report	19 hrs	Tue 22-4-14	Thu 24-4-14
16		➡	1.3	Project definition phase	61.75 hrs	Wed 23-4-14	Tue 6-5-14
17	✓	📌	1.3.1	General project risk identification and planning	3 hrs	Wed 30-4-14	Wed 30-4-14
18	✓	➡	1.3.2	Mission analysis	1 hr	Thu 24-4-14	Thu 24-4-14
19	✓	📌	1.3.2.1	Formulate mission need	0.5 hrs	Thu 24-4-14	Thu 24-4-14
20	✓	➡	1.3.2.2	Formulate mission statement	0.5 hrs	Thu 24-4-14	Thu 24-4-14
21	✓	➡	1.3.3	Functional analysis	17 hrs	Wed 23-4-14	Fri 25-4-14
22	✓	📌	1.3.3.1	Functional Breakdown Structure	7 hrs	Wed 23-4-14	Wed 23-4-14
23	✓	➡	1.3.3.2	Functional Flow Diagram	10 hrs	Thu 24-4-14	Fri 25-4-14
24	✓	➡	1.3.4	Requirements analysis	14 hrs	Thu 24-4-14	Fri 25-4-14
25	✓	➡	1.3.4.1	Define requirements	2 hrs	Thu 24-4-14	Thu 24-4-14
26	✓	➡	1.3.4.2	Requirement discovery tree	12 hrs	Thu 24-4-14	Fri 25-4-14
27		➡	1.3.5	Design options	24 hrs	Mon 28-4-14	Thu 1-5-14
28	✓	📌	1.3.5.1	Literature study for design options	10 hrs	Mon 28-4-14	Tue 29-4-14
29	📅	➡	1.3.5.2	Design option tree	24 hrs	Mon 28-4-14	Thu 1-5-14
30	✓	📌	1.3.6	Market analysis	7.5 hrs	Fri 25-4-14	Mon 28-4-14
31		📌	1.3.7	Resource allocation and budget breakdown	7.5 hrs	Fri 25-4-14	Mon 28-4-14
32	✓	📌	1.3.8	Contingency management	7.5 hrs	Fri 25-4-14	Mon 28-4-14
33	✓	📌	1.3.9	Sustainable development strategy	2 hrs	Tue 29-4-14	Tue 29-4-14
34	✓	➡	1.3.10	Technical risk assessment	13.5 hrs	Tue 29-4-14	Thu 1-5-14
35	✓	📌	1.3.10.1	Identify technical risks	13.5 hrs	Tue 29-4-14	Thu 1-5-14
36	✓	📌	1.3.10.2	Technical risk map	3.5 hrs	Thu 1-5-14	Thu 1-5-14
37	✓	📌	1.3.11	Quality check	14 hrs	Thu 1-5-14	Fri 2-5-14
38	✓	📌	1.3.12	Baseline report	39 hrs	Fri 25-4-14	Fri 2-5-14
39	📅	➡	1.3.13	Baseline review preparation	12 hrs	Thu 1-5-14	Tue 6-5-14
40	📅	➡	1.3.14	Baseline review	2 hrs	Tue 6-5-14	Tue 6-5-14
41		➡	1.4	Conceptual design phase	105 hrs?	Tue 6-5-14	Tue 27-5-14
42		➡	1.4.1	General project risk planning	1.25 hrs	Tue 6-5-14	Tue 6-5-14
43		➡	1.4.2	Team organisation	1.25 hrs	Tue 6-5-14	Tue 6-5-14
44	✓	➡	1.4.2.1	Update member positions and responsibilities	1 hr	Tue 6-5-14	Tue 6-5-14
45	✓	➡	1.4.2.2	Update Organisational Breakdown Structure	1.25 hrs	Tue 6-5-14	Tue 6-5-14
46		➡	1.4.2.3	Communication flow diagram	1.25 hrs	Tue 6-5-14	Tue 6-5-14
47		➡	1.4.3	Update project planning	3 hrs	Wed 7-5-14	Wed 7-5-14
48		➡	1.4.3.1	Identify phases and activities	2 hrs	Wed 7-5-14	Wed 7-5-14
49		➡	1.4.3.2	Gantt chart	2 hrs	Wed 7-5-14	Wed 7-5-14
50		➡	1.4.3.3	Work Breakdown Structure	1 hr	Wed 7-5-14	Wed 7-5-14
51		➡	1.4.3.4	Work Flow Diagram	2 hrs	Wed 7-5-14	Wed 7-5-14
52	✓	➡	1.4.4	System lay-out	18.5 hrs	Wed 7-5-14	Fri 9-5-14
53	✓	➡	1.4.4.1	Identify and define system components	8 hrs	Wed 7-5-14	Thu 8-5-14
54	✓	➡	1.4.4.1.1	Define system	2 hrs	Wed 7-5-14	Wed 7-5-14
55	✓	➡	1.4.4.1.2	Define subsystems	6 hrs	Wed 7-5-14	Thu 8-5-14
56	✓	➡	1.4.4.1.3	Define function of each subsystem	6 hrs	Wed 7-5-14	Thu 8-5-14
57	✓	➡	1.4.4.2	Identify and define subsystem relations	7 hrs	Thu 8-5-14	Fri 9-5-14
58	✓	➡	1.4.4.2.1	Indentify relations between subsystems	4 hrs	Thu 8-5-14	Thu 8-5-14
59	✓	➡	1.4.4.2.2	Define interfaces between subsystems	3 hrs	Thu 8-5-14	Fri 9-5-14
60	✓	➡	1.4.4.3	Create N2-charts	3.5 hrs	Fri 9-5-14	Fri 9-5-14
61		➡	1.4.5	Literature Study for concept generation	46 hrs?	Wed 7-5-14	Thu 15-5-14
62		➡	1.4.5.1	Existing UAVs	46 hrs?	Wed 7-5-14	Thu 15-5-14
63		📌	1.4.5.2	Existing power systems	46 hrs?	Wed 7-5-14	Thu 15-5-14
64		➡	1.4.5.3	General technology and concepts	46 hrs?	Wed 7-5-14	Thu 15-5-14
65		➡	1.4.6	Concept generation	49.5 hrs	Wed 7-5-14	Fri 16-5-14

Project: ProjectPlanningV4
Date: Mon 23-6-14

Page 1

ID		Task Mode	WBS	Task Name	Duration	Start	Finish
66			1.4.6.1	Update and reflect on design option tree	46 hrs	Wed 7-5-14	Thu 15-5-14
67			1.4.6.2	Generate concepts	46 hrs	Wed 7-5-14	Thu 15-5-14
68			1.4.6.3	Reflect on concepts	3.5 hrs	Fri 16-5-14	Fri 16-5-14
69			**1.4.7**	**Concept analysis**	**46 hrs**	**Wed 7-5-14**	**Thu 15-5-14**
70			**1.4.7.1**	**Basic characteristics analysis**	**46 hrs**	**Wed 7-5-14**	**Thu 15-5-14**
71			1.4.7.1.1	Aircraft subsystem characteristics estimate	5 hrs	Wed 7-5-14	Thu 8-5-14
72			1.4.7.1.2	Aerodynamic characteristics estimate	46 hrs	Wed 7-5-14	Thu 15-5-14
73			1.4.7.1.3	Structural characteristics estimate	46 hrs	Wed 7-5-14	Thu 15-5-14
74			1.4.7.1.4	Stability and control characteristics	46 hrs	Wed 7-5-14	Thu 15-5-14
75			1.4.7.2	Maintenance	46 hrs	Wed 7-5-14	Thu 15-5-14
76			**1.4.8**	**Design concept trade-off**	**14 hrs**	**Thu 15-5-14**	**Fri 16-5-14**
77			1.4.8.1	Trade-off method, rationale and organisation	3 hrs	Thu 15-5-14	Thu 15-5-14
78			1.4.8.2	Trade-off criteria	4 hrs	Thu 15-5-14	Thu 15-5-14
79			1.4.8.3	Weight factor criteria	4 hrs	Thu 15-5-14	Thu 15-5-14
80			1.4.8.4	Decision on final concept	3.5 hrs	Fri 16-5-14	Fri 16-5-14
81			1.4.8.5	Trade-off summary	3.5 hrs	Fri 16-5-14	Fri 16-5-14
82			**1.4.9**	**Risk Assesment**	**4 hrs**	**Mon 19-5-14**	**Mon 19-5-14**
83			1.4.9.1	Risk identification	1 hr	Mon 19-5-14	Mon 19-5-14
84			1.4.9.2	Risk assessment	1 hr	Mon 19-5-14	Mon 19-5-14
85			1.4.9.3	Risk analysis	1 hr	Mon 19-5-14	Mon 19-5-14
86			1.4.9.4	Risk handling	1 hr	Mon 19-5-14	Mon 19-5-14
87	✓		1.4.10	Operations and logistic concept	2 hrs	Wed 21-5-14	Wed 21-5-14
88			1.4.11	Update sustainable development strategy	3 hrs	Wed 21-5-14	Wed 21-5-14
89			1.4.12	Sensitivity analysis	8 hrs	Thu 22-5-14	Fri 23-5-14
90	✓		1.4.13	Verification and Validation procedures	3 hrs	Wed 21-5-14	Wed 21-5-14
91			1.4.14	Quality check	21 hrs	Wed 21-5-14	Fri 23-5-14
92			1.4.15	Mid-term report	35 hrs	Mon 19-5-14	Fri 23-5-14
93			1.4.16	Mid-term review preparation	10.75 hrs	Mon 26-5-14	Tue 27-5-14
94			1.4.17	Mid-term review	2 hrs	Tue 27-5-14	Tue 27-5-14
95			~~1.4.18~~				
96			~~1.4.19~~				
97			~~1.4.20~~				
98			**1.5**	**Detailed design phase**	**95.75 hrs?**	**Tue 27-5-14**	**Thu 19-6-14**
99			1.5.1	General project risk planning	1.25 hrs	Tue 27-5-14	Tue 27-5-14
100			**1.5.2**	**Team organisation**	**1 hr**	**Tue 27-5-14**	**Tue 27-5-14**
101			1.5.2.1	Update member positions and responsibilities	1 hr	Tue 27-5-14	Tue 27-5-14
102			1.5.2.2	Update Organisational Breakdown Structure	1 hr	Tue 27-5-14	Tue 27-5-14
103			1.5.2.3	Update Communication Flow Diagram	1 hr	Tue 27-5-14	Tue 27-5-14
104			**1.5.3**	**Creating detailed design**	**84 hrs?**	**Wed 28-5-14**	**Tue 17-6-14**
105			**1.5.3.1**	**Resource allocation and budget breakdown**	**3.5 hrs**	**Wed 28-5-14**	**Wed 28-5-14**
106			1.5.3.1.1	Mission element time budget	3.5 hrs	Wed 28-5-14	Wed 28-5-14
107			1.5.3.1.2	Sub-system power budget	3.5 hrs	Wed 28-5-14	Wed 28-5-14
108			1.5.3.1.3	Sub-system weight budget	3.5 hrs	Wed 28-5-14	Wed 28-5-14
109			1.5.3.1.4	Sub-system cost budget	3.5 hrs	Wed 28-5-14	Wed 28-5-14
110			1.5.3.1.5	Sub-system CPU budget	3.5 hrs	Wed 28-5-14	Wed 28-5-14
111			1.5.3.1.6	Sub-system communication budget	3.5 hrs	Wed 28-5-14	Wed 28-5-14
112			**1.5.3.2**	**Designing system**	**63 hrs?**	**Wed 28-5-14**	**Fri 13-6-14**
113			**1.5.3.2.1**	**MAV**	**63 hrs**	**Wed 28-5-14**	**Fri 13-6-14**
114			**1.5.3.2.1.1**	**Subsystems**	**49 hrs**	**Wed 28-5-14**	**Wed 11-6-14**
115			**1.5.3.2.1.1.1**	**Observation**	**21 hrs**	**Wed 28-5-14**	**Wed 4-6-14**
116			1.5.3.2.1.1.1.1	SLAM system	21 hrs	Wed 28-5-14	Wed 4-6-14
117			1.5.3.2.1.1.1.2	Image pointing and stabilisation	14 hrs	Wed 28-5-14	Tue 3-6-14
118			1.5.3.2.1.1.2	Attitude determination	14 hrs	Wed 4-6-14	Fri 6-6-14
119			1.5.3.2.1.1.3	Communication system	14 hrs	Fri 6-6-14	Wed 11-6-14
120			1.5.3.2.1.1.4	Data processing	14 hrs	Fri 6-6-14	Wed 11-6-14
121			1.5.3.2.1.1.5	Data storage	14 hrs	Fri 6-6-14	Wed 11-6-14
122			1.5.3.2.1.1.6	Guidance system	14 hrs	Wed 4-6-14	Fri 6-6-14
123			1.5.3.2.1.1.7	Position determination	14 hrs	Wed 4-6-14	Fri 6-6-14
124			1.5.3.2.1.1.8	Power system	14 hrs	Fri 6-6-14	Wed 11-6-14
125			1.5.3.2.1.1.9	CATIA Models	49 hrs	Wed 28-5-14	Wed 11-6-14
126			**1.5.3.2.1.2**	**Propulsion and aerodynamics**	**63 hrs**	**Wed 28-5-14**	**Fri 13-6-14**
127			**1.5.3.2.1.2.1**	**Design**	**42 hrs**	**Wed 28-5-14**	**Tue 10-6-14**
128			**1.5.3.2.1.2.1.1**	**Electronic propulsion system**	**21 hrs**	**Wed 28-5-14**	**Wed 4-6-14**
129			1.5.3.2.1.2.1.1.1	Electronic Speed Controller	21 hrs	Wed 28-5-14	Wed 4-6-14
130			1.5.3.2.1.2.1.1.2	Motor	21 hrs	Wed 28-5-14	Wed 4-6-14

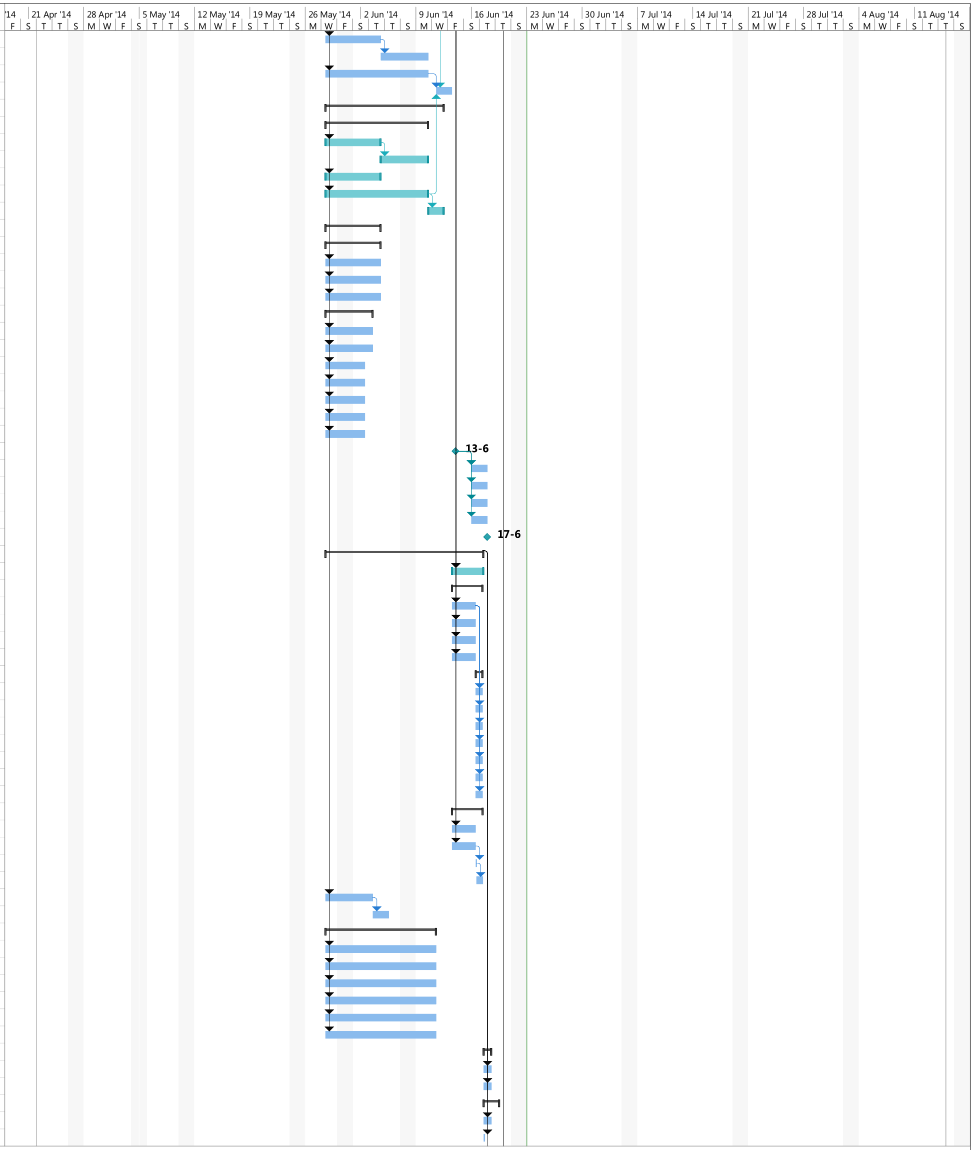


Project: ProjectPlanningV4
Date: Mon 23-6-14

Task		Summary		Inactive Milestone	◇	Duration-only		Start-only	[	External Milestone	◆	Manual Progress	
Split		Project Summary		Inactive Summary		Manual Summary Rollup		Finish-only	]	Deadline	⬇		
Milestone	◆	Inactive Task		Manual Task		Manual Summary		External Tasks		Progress			

Page 2

ID	Task Mode	WBS	Task Name	Duration	Start	Finish
131	⇒	1.5.3.2.1.2.1.2	Propeller aerodynamics	21 hrs	Wed 28-5-14	Wed 4-6-14
132	⇒	1.5.3.2.1.2.1.3	Structure aerodynamics	21 hrs	Wed 4-6-14	Tue 10-6-14
133	⇒	1.5.3.2.1.2.1.4	CATIA Model	42 hrs	Wed 28-5-14	Tue 10-6-14
134	⇒	1.5.3.2.1.2.2	CFD analysis	14 hrs	Wed 11-6-14	Fri 13-6-14
135	⇒	**1.5.3.2.1.3**	**Structural**	**56 hrs**	**Wed 28-5-14**	**Thu 12-6-14**
136	⇒	**1.5.3.2.1.3.1**	**Design**	**42 hrs**	**Wed 28-5-14**	**Tue 10-6-14**
137	📌	1.5.3.2.1.3.1.1	Boom design	21 hrs	Wed 28-5-14	Wed 4-6-14
138	📌	1.5.3.2.1.3.1.2	Core design	21 hrs	Wed 4-6-14	Tue 10-6-14
139	📌	1.5.3.2.1.3.1.3	Propeller design	21 hrs	Wed 28-5-14	Wed 4-6-14
140	📌	1.5.3.2.1.3.1.4	CATIA model	42 hrs	Wed 28-5-14	Tue 10-6-14
141	📌	1.5.3.2.1.3.2	FEM analysis	14 hrs	Tue 10-6-14	Thu 12-6-14
142	⇒	**1.5.3.2.1.4**	**Stability and control**	**21 hrs**	**Wed 28-5-14**	**Wed 4-6-14**
143	⇒	**1.5.3.2.1.4.1**	**Design**	**21 hrs**	**Wed 28-5-14**	**Wed 4-6-14**
144	⇒	1.5.3.2.1.4.1.1	Autopilot	21 hrs	Wed 28-5-14	Wed 4-6-14
145	⇒	1.5.3.2.1.4.1.2	Control system	21 hrs	Wed 28-5-14	Wed 4-6-14
146	⇒	1.5.3.2.1.4.2	Flight simulation analysis	21 hrs	Wed 28-5-14	Wed 4-6-14
147	⇒	**1.5.3.2.2**	**Ground station, operations and logistics**	**14 hrs?**	**Wed 28-5-14**	**Tue 3-6-14**
148	⇒	1.5.3.2.2.1	Hardware	14 hrs	Wed 28-5-14	Tue 3-6-14
149	⇒	1.5.3.2.2.2	Software	14 hrs	Wed 28-5-14	Tue 3-6-14
150	⇒	1.5.3.2.2.3	Transportation	7 hrs?	Wed 28-5-14	Mon 2-6-14
151	⇒	1.5.3.2.2.4	Setting up systems	7 hrs?	Wed 28-5-14	Mon 2-6-14
152	⇒	1.5.3.2.2.5	Pre-flight checks	7 hrs?	Wed 28-5-14	Mon 2-6-14
153	⇒	1.5.3.2.2.6	During flight	7 hrs?	Wed 28-5-14	Mon 2-6-14
154	⇒	1.5.3.2.2.7	After mission	7 hrs?	Wed 28-5-14	Mon 2-6-14
155	📌	1.5.3.3	First iteration finish date	0 hrs	Fri 13-6-14	Fri 13-6-14
156	⇒	1.5.3.4	Subsystems	14 hrs	Mon 16-6-14	Tue 17-6-14
157	⇒	1.5.3.5	Propulsion and aerodynamics	14 hrs	Mon 16-6-14	Tue 17-6-14
158	⇒	1.5.3.6	Structural	14 hrs	Mon 16-6-14	Tue 17-6-14
159	⇒	1.5.3.7	Stability and control	14 hrs	Mon 16-6-14	Tue 17-6-14
160	📌	1.5.3.8	Second iteration finish date	0 hrs	Tue 17-6-14	Tue 17-6-14
161	⇒	**1.5.4**	**Analyse system**	**77 hrs**	**Wed 28-5-14**	**Tue 17-6-14**
162	📌	1.5.4.1	Sensitivity analysis	14 hrs	Fri 13-6-14	Tue 17-6-14
163	⇒	**1.5.4.2**	**Performance analysis**	**11 hrs**	**Fri 13-6-14**	**Tue 17-6-14**
164	⇒	1.5.4.2.1	Flight profile diagram	7 hrs	Fri 13-6-14	Mon 16-6-14
165	⇒	1.5.4.2.2	Payload-range diagram	7 hrs	Fri 13-6-14	Mon 16-6-14
166	⇒	1.5.4.2.3	Climb performance	7 hrs	Fri 13-6-14	Mon 16-6-14
167	⇒	1.5.4.2.4	Agility performance	7 hrs	Fri 13-6-14	Mon 16-6-14
168	⇒	**1.5.4.2.5**	**Time analysis**	**4 hrs**	**Mon 16-6-14**	**Tue 17-6-14**
169	⇒	1.5.4.2.5.1	Transportation time	4 hrs	Mon 16-6-14	Tue 17-6-14
170	⇒	1.5.4.2.5.2	Setting up time	4 hrs	Mon 16-6-14	Tue 17-6-14
171	⇒	1.5.4.2.5.3	Mission (element) time	4 hrs	Mon 16-6-14	Tue 17-6-14
172	⇒	1.5.4.2.5.4	Data retrieval time	4 hrs	Mon 16-6-14	Tue 17-6-14
173	⇒	1.5.4.2.5.5	Total flight time	4 hrs	Mon 16-6-14	Tue 17-6-14
174	⇒	1.5.4.2.5.6	Pack time	4 hrs	Mon 16-6-14	Tue 17-6-14
175	⇒	1.5.4.2.5.7	Repair/part replacement time	4 hrs	Mon 16-6-14	Tue 17-6-14
176	⇒	**1.5.4.3**	**Technical risk assessment**	**12 hrs**	**Fri 13-6-14**	**Tue 17-6-14**
177	⇒	1.5.4.3.1	Risk identification	7 hrs	Fri 13-6-14	Mon 16-6-14
178	⇒	1.5.4.3.2	Risk assessment	7 hrs	Fri 13-6-14	Mon 16-6-14
179	⇒	1.5.4.3.3	Risk analysis	2 hrs	Mon 16-6-14	Mon 16-6-14
180	⇒	1.5.4.3.4	Risk handling	3 hrs	Mon 16-6-14	Tue 17-6-14
181	⇒	1.5.4.4	Technical performance measurement	14 hrs	Wed 28-5-14	Tue 3-6-14
182	⇒	1.5.4.5	RAMS analysis	14 hrs	Tue 3-6-14	Thu 5-6-14
183	⇒	**1.5.4.6**	**Characteristics analysis**	**49 hrs**	**Wed 28-5-14**	**Wed 11-6-14**
184	⇒	1.5.4.6.1	MAV system characteristics	49 hrs	Wed 28-5-14	Wed 11-6-14
185	⇒	1.5.4.6.2	Aerodynamic characteristics	49 hrs	Wed 28-5-14	Wed 11-6-14
186	⇒	1.5.4.6.3	Structural characteristics	49 hrs	Wed 28-5-14	Wed 11-6-14
187	⇒	1.5.4.6.4	Stability and control characteristics	49 hrs	Wed 28-5-14	Wed 11-6-14
188	⇒	1.5.4.6.5	Material characteristics	49 hrs	Wed 28-5-14	Wed 11-6-14
189	⇒	1.5.4.7	Market analysis	49 hrs	Wed 28-5-14	Wed 11-6-14
190	⇒	**1.5.5**	**Project documentation**	**7 hrs**	**Tue 17-6-14**	**Wed 18-6-14**
191	⇒	1.5.5.1	Project design & development logic	7 hrs	Tue 17-6-14	Wed 18-6-14
192	⇒	1.5.5.2	Project Gantt chart	7 hrs	Tue 17-6-14	Wed 18-6-14
193	⇒	**1.5.6**	**Product documentation**	**14 hrs**	**Tue 17-6-14**	**Thu 19-6-14**
194	⇒	1.5.6.1	Technical drawings	7 hrs	Tue 17-6-14	Wed 18-6-14
195	⇒	1.5.6.2	Functional flow diagram	3 hrs	Tue 17-6-14	Tue 17-6-14

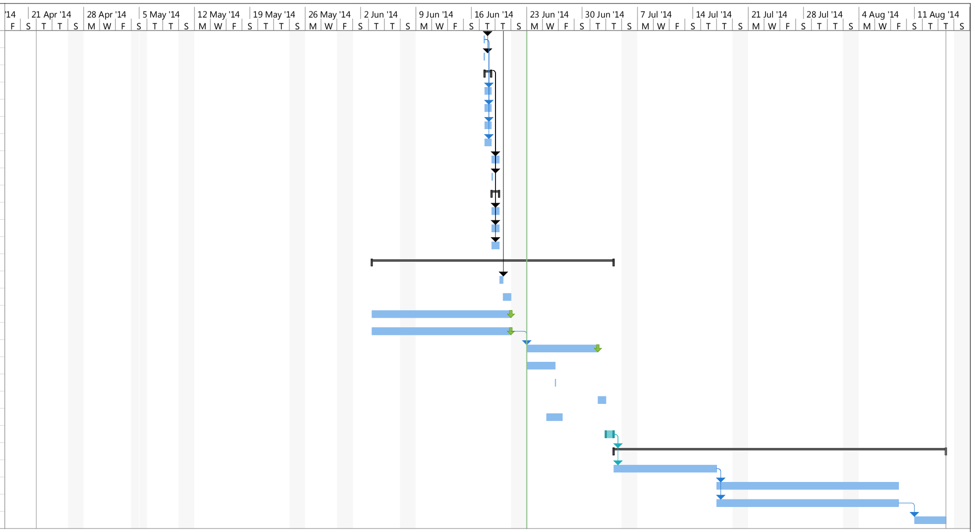


Project: ProjectPlanningV4
Date: Mon 23-6-14

Task	▬	Summary	⎴	Inactive Milestone	◇	Duration-only	▬	Start-only	[	External Milestone	◆	Manual Progress	▬
Split	┈	Project Summary	⎴	Inactive Summary	⎴	Manual Summary Rollup	▬	Finish-only	]	Deadline	⬇		
Milestone	◆	Inactive Task	▭	Manual Task	▬	Manual Summary	⎴	External Tasks	▬	Progress	▬		

Page 3

ID		Task Mode	WBS	Task Name	Duration	Start	Finish
196			1.5.6.3	Functional breakdown diagram	3 hrs	Tue 17-6-14	Tue 17-6-14
197			1.5.6.4	Cost breakdown structure	3 hrs	Tue 17-6-14	Tue 17-6-14
198			**1.5.6.5**	**Block diagrams**	**4 hrs**	**Tue 17-6-14**	**Wed 18-6-14**
199			1.5.6.5.1	Hardware block diagrams	4 hrs	Tue 17-6-14	Wed 18-6-14
200			1.5.6.5.2	Software block diagrams	4 hrs	Tue 17-6-14	Wed 18-6-14
201			1.5.6.5.3	Electrical block diagram	4 hrs	Tue 17-6-14	Wed 18-6-14
202			1.5.6.5.4	Data handling block diagram	4 hrs	Tue 17-6-14	Wed 18-6-14
203			1.5.6.6	Production plan	7 hrs	Wed 18-6-14	Thu 19-6-14
204			1.5.7	Sustainable development strategy	3 hrs	Wed 18-6-14	Wed 18-6-14
205			**1.5.8**	**Verification and validation**	**7 hrs**	**Wed 18-6-14**	**Thu 19-6-14**
206			1.5.8.1	Verification and validation procedures	7 hrs	Wed 18-6-14	Thu 19-6-14
207			1.5.8.2	Compliance matrix	7 hrs	Wed 18-6-14	Thu 19-6-14
208			1.5.8.3	Review of design analysis	7 hrs	Wed 18-6-14	Thu 19-6-14
209			**1.6**	**DSE wrap-up**	**153 hrs**	**Tue 3-6-14**	**Thu 3-7-14**
210			1.6.1	Evaluation	3.5 hrs	Thu 19-6-14	Thu 19-6-14
211			1.6.2	Jury summary	7 hrs	Fri 20-6-14	Fri 20-6-14
212			1.6.3	Quality check	90 hrs	Tue 3-6-14	Fri 20-6-14
213			1.6.4	Draft final report	90 hrs	Tue 3-6-14	Fri 20-6-14
214			1.6.5	Official final report	49 hrs	Mon 23-6-14	Tue 1-7-14
215			1.6.6	Prepare final review presentation	24.75 hrs	Mon 23-6-14	Thu 26-6-14
216			1.6.7	Final review	2 hrs	Thu 26-6-14	Thu 26-6-14
217			1.6.8	Executive summary	7 hrs	Wed 2-7-14	Wed 2-7-14
218			1.6.9	Create symposium poster	14 hrs	Wed 25-6-14	Fri 27-6-14
219			1.6.10	Symposium	7 hrs	Thu 3-7-14	Thu 3-7-14
220			**1.7**	**Post DSE**	**210 hrs**	**Fri 4-7-14**	**Thu 14-8-14**
221			1.7.1	Finishing the design	63 hrs	Fri 4-7-14	Wed 16-7-14
222			1.7.2	Production	119 hrs	Thu 17-7-14	Fri 8-8-14
223			1.7.3	Programming	119 hrs	Thu 17-7-14	Fri 8-8-14
224			1.7.4	Competition	28 hrs	Mon 11-8-14	Thu 14-8-14



Project: ProjectPlanningV4
Date: Mon 23-6-14