NTNU

Kunnskap for en bedre verden

Bacheloroppgave

IE303612 Bacheloroppgave - Data og automasjon

Drone: Dynamic positioning in relation to a given object

801, 830, 840

Totalt antall sider inkludert forsiden: 97,6 vedlegg

Innlevert Ålesund, 26.05.16

Obligatorisk egenerklæring/gruppeerklæring

Den enkelte student er selv ansvarlig for å sette seg inn i hva som er lovlige hjelpemidler, retningslinjer for bruk av disse og regler om kildebruk. Erklæringen skal bevisstgjøre studentene på deres ansvar og hvilke konsekvenser fusk kan medføre. **Manglende erklæring fritar ikke studentene fra sitt ansvar**.

Du/dere fyller ut erklæringen ved å klikke i ruten til høyre for den enkelte del 1-6:		
1.	**Jeg/vi erklærer herved at min/vår besvarelse er mitt/vårt eget arbeid, og at jeg/vi ikke har brukt andre kilder eller har mottatt annen hjelp enn det som er nevnt i besvarelsen.**	☒
2.	**Jeg/vi erklærer videre at denne besvarelsen:** • ikke har vært brukt til annen eksamen ved annen avdeling/universitet/høgskole innenlands eller utenlands. • ikke refererer til andres arbeid uten at det er oppgitt. • ikke refererer til eget tidligere arbeid uten at det er oppgitt. • har alle referansene oppgitt i litteraturlisten. • ikke er en kopi, duplikat eller avskrift av andres arbeid eller besvarelse.	☒
3.	Jeg/vi er kjent med at brudd på ovennevnte er å betrakte som fusk og kan medføre annullering av eksamen og utestengelse fra universiteter og høgskoler i Norge, jf. Universitets- og høgskoleloven §§4-7 og 4-8 og Forskrift om eksamen.	☒
4.	Jeg/vi er kjent med at alle innleverte oppgaver kan bli plagiatkontrollert i Ephorus, se Retningslinjer for elektronisk innlevering og publisering av studiepoenggivende studentoppgaver	☒
5.	Jeg/vi er kjent med at høgskolen vil behandle alle saker hvor det forligger mistanke om fusk etter NTNUs studieforskrift.	☒
6.	Jeg/vi har satt oss inn i regler og retningslinjer i bruk av kilder og referanser på biblioteket sine nettsider	☒

Publiseringsavtale

Studiepoeng: 20

Veileder: Robin T. Bye, Ottar L. Osen, Ibrahim A. Abdelhameed

Fullmakt til elektronisk publisering av oppgaven

Forfatter(ne) har opphavsrett til oppgaven. Det betyr blant annet enerett til å gjøre verket tilgjengelig for allmennheten (Åndsverkloven §2).
Alle oppgaver som fyller kriteriene vil bli registrert og publisert i Brage med forfatter(ne)s godkjennelse.
Oppgaver som er unntatt offentlighet eller båndlagt vil ikke bli publisert.

Jeg/vi gir herved NTNU i Ålesund en vederlagsfri rett til å gjøre oppgaven tilgjengelig for elektronisk publisering:	☒ja	☐nei
Er oppgaven båndlagt (konfidensiell)? (Båndleggingsavtale må fylles ut)	☐ja	☒nei
- Hvis ja: **Kan oppgaven publiseres når båndleggingsperioden er over?**	☐ja	☐nei
Er oppgaven unntatt offentlighet? (inneholder taushetsbelagt informasjon. Jfr. Offl. §13/Fvl. §13)	☐ja	☒nei

Dato: 26.05.16

BACHELOR THESIS

NTNU

CANDIDATE NO(S):				
801, 830, 840				
DATE:	SUBJECT CODE:	SUBJECT NAME:		
26.05.16	IE303612	BACHELOROPPGAVE - DATA OG AUTOMASJON		
STUDY PROGRAMME:		NO OF PAGES/APPENDIX:	BIBL. NO:	
Data og automasjon		93/6		

SUPERVISOR(S):

Robin T. Bye, Ottar L. Osen, Ibrahim A. Abdelhameed

TITLE:

Drone: Dynamic positioning in relation to a given object

Summary:
This report deals with the bachelor thesis: "Drone: Dynamic positioning in relation to a given object", given by Rolls-Royce Marine and NTNU Ålesund.

The goal of this task is to develop a framework for a drone autopilot and finding usage scenarios of drones in maritime industry. Main focus for this task is dynamic positioning in relation to an object in movement og autonomous landing and autonomous take-off.

The result is an autopilot that has three main modes. The first mode is supposed to be a simple takeoff mode, this will initialize, arm and move the drone to a given altitude. The second mode is the balloon finder mode, this will look for a target(represented by balloon) then pivot towards the target and maintain a set distance. The third mode is a simple landing mode, this will bring the drone back to its starting position while using the SIFT algorithm to recognise to find the correct landing spot which is a qr-code.

The report has the needed background theory, methods used, the results that have been achieved, and a discussion of the methods and results and at last a conclusion. In appendix you also find source code for the autopilot, meeting reports, progress reports, laws and regulation for drones by lovdata and the pre-project report

This project is handed in for evaluation and accreditation at NTNU in Aalesund.

Address
NTNU in Aalesund
Postboks 1517
N-6025 Aalesund
Norway

Org. no. 974 767 880

Internet
www.ntnu.no

Location
Larsgårdsvegen 2

Email address
postmottak@alesund.ntnu.no

Telephone
+47 70 16 12 00

Preface

This thesis is written to fulfil the requirements of a Bachelor's Degree in Computer Science and Automation for all authors. The work found in this thesis has been done from January to May of 2016. Our supervisors for this project has been Robin T. Bye, Ottar L. Osen, and Ibrahim Abdelfattah Abdelhameed from NTNU Ålesund.

In the end of 2015, Rolls-Royce, represented by Jan Marius Grimstad Sund and Dag Sverre Grønmyr, presented a project to NTNU for use as a bachelor's thesis. As the members of this group have expressed a great degree of interest in projects of this type and have experience in the fields of AI, computer vision, mechatronics, map systems, and drone use from earlier work and education, the choice of this project was obvious.

Writing this thesis has been a challenge as the project was fraught with technical difficulties, all of which have been solved or circumvented. This process has given us a great deal of experience in dealing with and planing for unexpected problems. While this project does not have a physical product, we hope our research can be of interest to later projects with a similar goal, and provide some guidance to Rolls-Royce about further development in the field of maritime drones.

The Aim of this Thesis
The aim of this thesis is to provide a foundation for Rolls-Royce's development of maritime drones and doctrine. This thesis will also be used as a potential basis for further bachelor's theses at NTNU Ålesund. We also hope this thesis will be of interest for the general drone community.

Acknowledgements

We would like to thank Randy Mackay for the code and advice he has provided in regard to the balloon finding algorithm. We would also like to thank Niklas, Kenneth and Vidar for their contributions during our physical testing.

SUMMARY

This report deals with the bachelor thesis: "Drone: Dynamic positioning in relation to a given object", given by Rolls-Royce Marine and NTNU Ålesund.

The goal of this task is to develop a framework for a drone autopilot and finding usage scenarios of drones in maritime industry. Main focus for this task is dynamic positioning in relation to an object in movement og autonomous landing and autonomous take-off.

The result is an autopilot that has three main modes. The first mode is supposed to be a simple takeoff mode, this will initialize, arm and move the drone to a given altitude. The second mode is the balloon finder mode, this will look for a target(represented by balloon) then pivot towards the target and maintain a set distance. The third mode is a simple landing mode, this will bring the drone back to its starting position while using the SIFT algorithm to recognise to find the correct landing spot which is a qr-code.

The report has the needed background theory, methods used, the results that have been achieved, and a discussion of the methods and results and at last a conclusion. In appendix you also find source code for the autopilot, meeting reports, progress reports, laws and regulation for drones by lovdata and the pre-project report.

TERMINOLOGY

Terms

Drone An unmanned vehicle, manually or autonomously controlled.

BUS or BUSS

SITL Software in the loop, simulator not requiring hardware.

Abbreviations

UAV An Unmanned Aerial Vehicle.

FoV Field of View.

LoS Line of Sight.

FD Frame differecing.

ICE Internal Combustion Engine.

GPS Global Positioning System.

PPM Pulse Position Modulation

PWM Pulse Width Modulation

SBUS Smart BUS

RPAS Remote piloted aircraft system

GPU Graphic Processing Unit

List of Figures

CONTENTS

1 INTRODUCTION

The background for this task is that Rolls-Royce wanted to explore the possibilities of usage of drones in the maritime industry. The task was given as one of the possible bachelor tasks.

Rolls-Royce Marine is a world leading supplier of ship design and systems. More than 30000 vessels in operations around the world use equipment and parts from Rolls-Royce. Rolls-Royce ha sa training and technology center situated at the Norwegian Maritime Center of Expertise in Ålesund, we were give this task from that department

Drones in commercial industry and military is a fast growing business. The civil side of the industry is expected to grow 20% economically between 2015 and 2020. Commercial drones are expected to be used in industries such as agriculture, mining, building, search and rescue, and surveying. The limits for drones are unknown and yet unexplored.
The task we have been given is compromised of two main parts, one theoretical and one practical. This project will potentially have several follow up projects completed by different groups at a later date.

1.1 Theoretical

The theoretical part is focused around the use of drones, specifically autonomous drones, in a commercial maritime setting. We are to find various uses for drones in a maritime setting and evaluate whether they would make sense to implement commercially. Recommendations are being made based on several factors affecting economic viability, among them: ease of access to the technology, ease of training personnel to use said technology, cost of use compared to traditional methods, safety of use compared to traditional methods, efficiency and task solving capabilities compared to traditional methods, and versatility of the system.

1.2 Practical

The practical part is to build a framework for an autopilot for drones with potential uses in a commercial maritime setting. This autopilot will be a test of concept to be worked on by further projects. The autopilot should be able to identify and follow a predefined target as well as find a landing pad and land there successfully.

1.3 The report

This thesis is built as a technical report. The thesis lists all relevant theory(and some less relevant) about current use of drones and autopilots. Following this is a section about equipment used and how various solutions has been implemented. Following this we have results from tests and discussions about those results in relation to information gathered from other sources. The thesis ends on a conclusion where you can find our recommendations for Rolls-Royce.

2 THEORY

2.1 Current usage of drones

Drone in this context is an unmanned vehicle, for the sake of this thesis drone is mostly used to reffer to an unmaned aerial vehicle.

Hassan L.[1] mentions a few of the bigger arguments for and against the use of UAVs in a military and commercial context. The author concludes that drones in most, if not all, applications can have several benefits; cheaper than other technical alternatives, not affected by fatigue and therefore has a higher endurance than human solutions, easier to use and more effective with less training, and highly versatile within certain physical limitations. On the other hand there are several points the author cautions against as well; removal of the human element can lead to detachment from the task at hand, lost advantages from loosing human actors in the area, laws and regulations falling behind the development and possibilities of drones and the moral questions about drone surveillance.

2.1.1 Geographical surveying / mapping

Drones has been used as a tool to map large areas. Drones are quicker, cheaper and easier to operate than larger vehicles thus saving time, manpower and resources. New technologies and techniques are enabling drones to enter roles traditionally filled by larger vehicles or manual labour.

Microdrones[2] use drones to map areas quickly and cheaply compared to doing it by hand from the ground. In this case a drone would fly a grid over an area, mapping the ground with either a camera, lidar, radar, sonar or similar equipment. Then these results will later be processed by a computer to give a 2D map or even a fully 3D rendition of the area. Given a rangefinder and accurate gps coordinates you can find the elevation of every point on this area. This can be done to give surface values or in some cases sub surface values (water).

Paul Watson[3] talks about drones used by icebreakers and cargo ships to scout for viable courses through ice fields or around icebergs. A drone would give the ship a higher vantage point as well as the option to move away from the ship finding paths that could be blocked from the ships position. Use of drones in this way complements the ships radar giving vision to areas that would be out of the ships line of sight in normal situations.

Several researchers from The US Geological Survey and Parallel Inc. wrote a paper[4] where they describe an experiment using drones for high-resolution topographical mapping. This paper is a strong example of how drones could see more widespread use after proper regulations, laws, and guidelines are established. One relevant point they make is that while this service is already filled by aircraft, doing it by drone reduces costs from almost 100,000 down to merely several thousand dollars. A full lidar scan from an aircraft will be able to cover a larger area compared to a drone in a single flight, this leads to a drone survey taking a lot longer than by aircraft, but the price tag speaks for itself. Given increases in drone technology, especially concerning range and endurance, this will only get more efficient.

Fintan Corrigan wrote an introduction to UAV mapping[5] where he describes UAV Photogrammetry and lidar mapping. He talks about several situation where use of drones will be more cost effective and efficient than established methods. Forestry Management and Planning, Flood Modelling, Pollution Modelling, Mapping and Cartography, Urban Planning, Coastline Management, Transport Planning, Oil and Gas Exploration, Quarries and Minerals (Volumetrics and Exploration), Archaeology, Cellular Network Planning.

2.1.2 Search and Rescue

In search and rescue operations the victim usually has a GPS of some sort narrowing down the required area to search. In other situations a large number of people will be required to spend huge amounts of time trawling trough a rather large area looking for tracks and traces of the missing person. With a drone or a fleet of drones you can quickly and efficiently cover a rather large area looking for discrepancies. Thermal imaging is already a highly effective way to look for people, doing so from the air increases chances of success greatly at a fraction of the cost for a chopper or other large aircraft.

In an article[6] for Ocean Hub, John Howard talks about a Spanish constructed drone designed for search and rescue. The drone is abnormally large at 80 kg with a lift capacity of up to 70 kg. The drone has a claimed flight time of more than three hours and is capable of operating in various types of challenging weather, this makes it useful for search and rescue. This drone uses thermal imaging to find people lost at sea and then transmit their gps coordinates back to the base station. The size and lift of the drone also enables it to carry a small inflatable rescue raft increasing survival rates.[38]

2.1.3 Surveillance

Drones can be used for surveillance, either autonomously looking for motion or with a controller, in areas where climate, distance, price or other factors prevent use of human surveillance. This is not a frequent use of drones currently as stationary cameras and motion sensors are already capable of completing this task with high reliability[37].

A news rapport[7] from Hellenic Shipping News talks about using drones to efficiently and discreetly take pictures of illegally moored vessels. In this situation stationary cameras are not used due to a lack of permanent sight lines and quickly shifting terrain.

2.1.4 Supply

Drones has been used to move small goods from A to B, Amazon has been working on light commercial deliveries from their warehouses and Doctors Without Borders have been experimenting with using drones to deliver critical medical supplies to isolated areas[36].

Amazon[8][9] are trying to develop a system of delivery drones to supplement their current truck deliveries. There is no hard information about how or when they will launch this concept. As it stands several conditions needs to be fulfilled regarding size, weight and range for packages delivered this way, almost every 9 out of 10 packages fit these criteria. The greatest challenges faced by this concept presently are laws and regulations as commercial use of drones in the US is still on shaky legal grounds. Other factors to be considered are the safety and privacy of people and property under the drones flight path and the safety of the drone itself.

2.1.5 Military

Drones have seen substantial amounts use among armed forces since the early 1800[20]. The ability of a drone to quickly cover large areas without setting the operator at risk has greatly increased their popularity over the last few decades. Military use of unmanned aircraft has been thrown around since just before 1850 with hot air balloons dropping bombs. During and following the first world war several attempts has been made to create guided or independent aircraft and missiles[21]. These efforts meet some limited success until 1940 when a system was developed giving the ground station knowledge about the status of the drone removing the requirement for line of sight from the operator to the drone[22].

One of the most used arguments against usage of drones in warfare is civilian targets and how it is far easier for an operator on the other side of the earth to make a mistake or accept moral risks leading to unnecessary civilian casualties. According to this article[10], several studies have concluded that unintentional loss of life in drone strikes are a lot lower than they would be with traditional forces.

2.1.6 Satellite replacement

Drones would be a natural alternative to temporary satellites as the larger versions has endurance to remain on station potentially indefinitely with the help of refueling[23]. Launching a drone comes at a fraction of the cost of a satellite and could temporarily provide crucial infrastructure to relatively large areas.

It has been theorised[11] since the late 1950 that UAVs can be used to replace satellites for atmospheric and weather research, earth and water observations, and particularly communications.

2.1.7 Agriculture

Planes has traditionally been used to dust or survey large fields and farm-able lands. With drones this could be done easier and more cheaply without a noticeable loss in efficiency.

Drones are being used for agriculture[12] in much the same way as they are used for environmental surveying. Farmers need an easy and quick way to monitor their crops and in larger fields this is traditionally done with an airplane, but drones could do the same task at a fraction of the cost with a fraction of the training for the pilot.

Drones are also being used for crop-dusting[13]. Capable of deploying any liquid on any terrain quickly and cheaply this is an option to the already widespread crop-dusting aircraft. These drones need to be of a certain size to be useful and are therefore exclusively powered by an internal combustion engine.

2.2 Drone Power/Fuel

Flight time has always been the major factor[30] in limiting where and how drones can be used. As it stands most battery driven drones have a max flight time of no more than half an hour, this is not sufficient for most commercial uses.

2.2.1 Battery

Battery is the traditional way to power small drones. While providing a limited power supply and long recharge times battery powered drones have been safer and easier to produce in the required sizes. One mentionable disadvantage with battery power are the needed charge times, while refueling a gas or hydrogen drone can be done in a matter of minutes recharging a battery could take from 40 minutes to several hours. This problem can be mitigated by having several batteries ready charging one while using the others, but there will still be severe amounts of downtime.[29]

2.2.2 Solar

One alternative to battery power that has long been pursued is solar power. Solar cells on a drone would give it potentially unlimited endurance[24] during certain conditions. The main problems with solar power are the specific conditions required for it to work effectively and the low efficiency of solar cells compared to the amount of power required to solve complicated or demanding tasks.

Alta Devices[17] has created solar cells with an efficiency of one watt per gram of weight. This grants enough power to lift lightweight drones in good conditions and have them flying potential 9 hours giving full daytime coverage. The drone they use for concept testing still has a small battery to provide extra power if conditions require, this battery charges of the cells giving somewhat increased versatility. The drone used in this example is not capable of lifting any noticeable payload beyond a camera and a companion computer. So while this drone can run surveillance on a large area for the duration of an entire day, it is not capable of solving complicated tasks requiring more or heavier equipment.

In early 2016 google tested[18] a solar powered drone. This drone is supposed to be used as a mobile/temporary hot spot for 5G wireless Internet connection. The system used could theoretically transmit data at more than 40 times the speed of current top of the line 4G systems. Solar powered drones are potentially capable of remaining airborne indefinitely assuming they get enough sunlight during the day.

2.2.3 Gasoline

Gasoline is an option to power drones. Large drones need more power than a battery can provide and as such older drones and drones over a certain weight are in the majority of cases powered by an internal combustion engine. A good internal combustion engine will outperform most other solutions, in the same price range, for sustained high power[25]. Any ICE needs to be a certain size to work, so this will not be an option for micro drones or drones where space is at a premium instead of endurance. ICEs also have the issue of sound to consider.

German inventor Holger Willeke[16] has designed a hybrid (gas-battery) powered drone capable of staying airborne for up to an hour and carry nearly 6kg payloads in addition to the base weight of the drone. A gas powered drone is easier and quicker to refuel than having to recharge the batteries normally.

2.2.4 Hydrogen

Recent experiments[14] with hydrogen fuel cells as an add-on to a standard battery powered drone has been met with success. Adding a self contained hydrogen fuel cell to the battery of a generic drone increased flight time from around 20min to over two hours.

Intelligent Energy[14] has been working towards hydrogen fuel cells to power drones giving an estimated flight time of more than two hours. This would greatly increase the versatility and potential usefullness of drones. This solution would be a fuel booster cell that could be applied to several different types and sizes of drones.

The prototype fuel cell[19] adds about 1.6 kg to the drone. Economically speaking a fuel cell like this would be considerably more expensive than a battery and the cost of hydrogen in itself would be notably higher than recharging a battery or fueling with gas.

Horizon Energy[15] Systems has constructed a drone running entirely on hydrogen. This drone was build from the ground up with hydrogen power in mind and has therefore an internal hydrogen tank, with no external secondary battery, the concept is proven but this design does not work towards increasing the range of ordinary drones. The prototype has a total weight of 5kg with a possible payload of 1kg. Fully loaded it has a flight time of around 2.5 hours, flying dry it has a potential flight time of around 4 hours.

2.3 Image Processing and Computer vision

[39] [40]
Image processing is processing by an image or a video using mathematical operations. Most image processing techniques are based of methods from signal processing. The output you get from the processing can be an image or set of parameters and/or characteristics.

Computer vision in a target analysis is normally divided into primarily parts detecting of the target and tracking the target. In this sections it will be explained a few techniques for both of them.

2.3.1 Target detection

The target detection techniques is used for detection of a moving target.

- Frame differencing algorithm: The FD algorithm uses two consecutive frames taken from the camera/video and calculates the difference between the images with a given threshold. When an object has moved from the previous frame it will be treated a difference. Therefore detected as a moving object.

- Background subtraction also known as foreground detection is method that subtracts the background from the image. Then i detects changes from the background. Used on stationary cameras to for example detect persons in a corridor and cars on a highway.

2.3.2 Target Tracking and Target Recognition

Short explanations for target tracking.

- Mean-shift is a algorithm used for tracking objects where the appearance of the object can be defined by the histogram. A tracking technique that is not limited by color only. [41]

- Feature detection is a method of calculating abstractions of image information. An image feature is not defined, but i can be for example edges, corners and blobs. A point in the image where there is a feature of a given type. The result from feature detection is often as isolated points, continous curves and connected regions in the image domain. [42]

  - SIFT-Scale invariant feature transform is a technique for feature detection and description. The SIFT detector extracts a feature description of the object. For example the extracted key-points from a training image that only show the object you want to track. Then use key-points to match them in a more complex image from a live feed, video or still image then use it to locate the object in that "space". The method i scale invariant so the object can change i form and size still be detected. [43]

- SURF-Speeded up robust features is a computer vision technique for local feature detector and descriptor. It is commonly used for tasks such as object recognition, registration, classification and 3D reconstruction. The technique is originally based of SIFT. SURF was developed for the reason that SIFT is to slow for real-time vision. In its standard version SURF is claimed to be several times faster than SIFT. SURF was presented in 2006 at the "European conference on computer vision" by Herbert Bay. The algorithm is patented if implemented into an application. [44]

- Tracking by color detection and shape is commonly used method. The technique here is at first here is using color threshold such as by hsv-color space. Then taking the thresholded image and process it contours(edges in every direction). Normally to get the shape from the processed image on will run a hough transform or a blob detector.

2.3.3 Flann based keypoint matcher

Flann based keypoint matcher is matcher implemented into opencv from a library called flann. Flann is fast approximate nearest neighbor searches. It works in high dimensional spaces. The mather contains of a library of matcher found to work the best for nearest neighbor searches. It also contains a system that chooses the best algorithm for the matching and the parameters depending on the data that runs on optimum performance. [45]

2.3.4 Threshold

Threshold can be either be done in a gray scale image or a color image to distinguish the object from the background. The result is a black and white image, where white is the object and black is the background.

2.3.5 Dilation

Dilation is a method that grows an object in a threshold image.

2.3.6 Erode

Erode is a method that shrinks the object in a threshold image.

2.4 Software

2.4.1 Visual Studio

Visual studio is an IDE for creating applications for Windows, Android and iOS. The IDE have tools for design, editors, debuggers, profillers everything built in one tool. Visual studio also has a large eco system built around it with thousands of extension. With multiple language support to code in, like C#,Visual Basic,F#,C++,HTML,Javascript,Python and more. [46]

2.4.2 CMake

CMake is an open-source tool designed for building, test and package software. CMake i used to control the software compilation process using simple platform and compiler independent configuration files, and generates native makefiles to let you use the compiler you want. [47]

2.4.3 Make

Make is a tool used to control the generation of executables and other non-source files of a program from the source files. Make is using a file called the makefile to get knowlegde of how to build the program. The makefile has a list of each non-source files and instructions of how to compute it from the other files. Make can be used to build and install program you write by using a makefile you have created. [48]

2.4.4 OpenCV

OpenCV or Open source computer vision is a computer vision library made for multiple languages. OpenCV is released under a BSD-licence. BSD licence is a free licensed software that would say that it can redistrubuted with minimal restrictions of covered software. The software is free both for academic use and commercial use. It has implementation for C++, C, Python and Java interface. It can be used on a wide variety of platforms such as Windows, Linux, Mac OS, IOS and Android. OpenCV is based on being calculation effective and have focus on interactions in real-time applications. It is written in optimized C/C++ and the library support multi-core processing. Support for hardware accelerations by opencl, cuda and opengl by using a graphics card to shorten the processing time. [49]

OpenCV contrib is a repository for OpenCV's extra modules which is made for contributed functionality. For example modules that have been well tested and do not have a stable API is things you find here. This is modules for OpenCV that are not released as a part of official

part of the distribution because it doesn't meet the requirements set by the development team in performance, stability and binary compatibility. The repository also has modules that is patented so you have pay the disclaimer if not used for academic use and therefor not in the official release since it will always be free.
[50]

GPU accelerated image processing

Gpu accelerated image processing is supported by opencv. The benefits of gpu accelerating is pure performance. Instead of computing with cpu, with gpu's you get the benefits of powerful parallel computing. The libraries supported by opencv is CUDA(nvidia gpu's), OpenCL and C++ AMP. [51]

2.4.5 Clion

Clion is a cross plattform IDE for editing and writing C and C++. [52]

2.4.6 Python

Python is a object oriented programming language and so called high level language with dynamic semantics. Its an attractive language for rapid application development because of its high level built data structures combined with dynamic bindings and dynamic typing. It is used for other task as scripting for building existing components together and make connect and talk to each other. Python is a simple and easy syntax to learn which makes emphasize on readability and for that reason it is reducing the cost of maintenance of the software. Modules and packages are supported by python which let you reuse code and gives it more modularity. Python extensive standard libraries and are avaible both as source code and binary form running on all major platforms and can freely be redistributed. [53]

2.4.7 C++

C++ is an object oriented programming language. It was developed by Bjarne Stroustrup as a phd student. The language is an extension of the normal C language. You can code C++ in "C style" or object oriented type of programming. Since it is a hybrid language it can be programmed in either way. The C++ programming language is an object oriented language for general purpose. The level of the programming language is considered to be intermediate, which means it has both low level language and high level language features. In the beginning the language was calles 'C with classes' since had everything from the C language and with classes as an extra. In 1983 it was renamed to C++. [54]

2.4.8 Sharelatex

Sharelatex is an online latex editor. Latex is a programming language that is used for typesetting technical data. The user database for latex is normally mathematicians, scientists and engineers because you can set mathematic symbols and formulas in text. Latex is often used instead of the more common text editors like word, libreoffice and so on. [55]

2.5 Flight controller

The flight controller is the brains of your drone. The flight controller reads all of the sensor data and calculates the best commands to send to your drone in order for it to fly. The different sensor data a drone use to maintain stable flight comes from accelerometer, gyroscope, compass, GPS and barometer.
[28].

2.5.1 3D Robotics

3D Robotics is company that develops innovative, flexible and reliable personal drones and UAV technology used for exploration for everyone as well as for business applications.

2.5.2 Pixhawk

PX4[27] is an independent, open-source, open-hardware project aiming at providing a high-end autopilot to the academic, hobby and industrial communities (BSD licensed) at low costs and high availability. It is a complete hardware and software platform, much like a computer, and can run multiple autopilot applications. It is supported by the Computer Vision and Geometry Lab of ETH Zurich (Swiss Federal Institute of Technology) and by the Autonomous Systems Laband the Automatic Control Laboratory as well from a number of individuals.

2.5.3 Paparazzi

Paparazzi UAV[31] (Unmanned Aerial Vehicle) is an open-source drone hardware and software project encompassing autopilot systems and ground station software for multicopters/multirotors, fixed-wing, helicopters and hybrid aircraft that was founded in 2003. Paparazzi UAV was designed with autonomous flight as the primary focus and manual flying as the secondary. From the beginning it was designed with portability in mind and the ability to control multiple aircraft within the same system. Paparazzi features a dynamic flight plan system that is defined by mission states and using way points as “variables”. This makes it easy to create very complex fully automated missions without the operators intervention. For more project information, see here.

2.5.4 Ardupilot

APM/ArduPilot[32] is an open source drone platform, able to controll a wide range of vehicles; fixed-wing aircraft, helicopters, multicopters, tiltrotors, tail sitters, boats, ground rovers. It was created in 2007 by the DIY Drones community. APM is one of the world's leading autonomous vehicle platforms. Being open source it has been developed to enable a large range of hardware and software products, including the APM/ArduPilot and Pixhawk/PX4 line of autopilot, a range of Linux based autopilot boards, and the APM:Copter, APM:Plane, APM:Rover, and SITL software projects. It relied on knowing the location of the horizon relative to the craft by measuring difference in temperature between the sky and the ground. Later the system was updated to use intenral measurement units, a combination of accelerometers, gyroscopes and magnometers.

Copter is the module of ardupilot created for controlling drones(multirotors) and helicopters autonomously.

Plane is the module of ardupilot created for controlling planes autonomously.

Rover is the module of ardupilot created for controlling cars(ground vehicles) autonomously.

2.5.5 Mission Planner

Mission planner[32] is a ground station for use with ardupilot drones of the plane, copter and rover variety. We have used Mission planner to configure our drone and controller. Mission Planner can build a flight plan for the drone to follow as well as load autonomous missions into the flight controller simply. Mission planner downloads and analyzes logs created by the autopilot to improve further missions. Mission planner also monitors the drone during the mission, granting feedback to the pilot according to what sort of telemetry hardware you have installed. According to this feedback from the drone the user can make certain changes to the mission parameters during flight. Mission planner also allows the user to take manual control over the drone on a moments notice. Manual control can then be performed normally or trough a camera mounted on the drone giving full first person control.

-Load the firmware (the software) into the autopilot (APM, PX4...) that controls your vehicle.
-Setup, configure, and tune your vehicle for optimum performance.
-Plan, save and load autonomous missions into you autopilot with simple point-and-click way-point entry on Google or other maps.
-Download and analyze mission logs created by your autopilot.
-Interface with a PC flight simulator to create a full hardware-in-the-loop UAV simulator.

With appropriate telemetry hardware you can:
-Monitor your vehicle's status while in operation.
-Record telemetry logs which contain much more information the the on-board autopilot logs.
-View and analyze the telemetry logs.
-Operate your vehicle in FPV (first person view)

2.5.6 Dronekit-Python

Dronekit-python is a library for python that enables developers to create apps for drones that runs on a companion computer. Dronekit is made to communicate on a low-latency link with ardupilot. Library is meant to enchance the autopilot with onboard apps to add greater intelligence to how the drone behaves. The onboard apps normally performs tasks that are computationally intensive or time-sensitive. Time-sensitive task can be for example computer vision, path planning or 3d modelling of the enviroment. The library can also be used in ground station apps and communicating with the drones over a higher latency radio link. [56]

2.5.7 GPS

GPS or global positioning system is a navigation system that provide location and time information in all weather conditions. The navigation system is based of satelites in space. The system works everywhere on earth where there is a clear line of sight to four or more satelites. When you have four satelites it is possible to place you in a spatially position in three dimensions and the receiver clock also gets deviation from satelite system time-reference. [57]

2.5.8 Gyro

A gyro or gyroscope is a device that determine the orientation by using Earth-gravity. [58]

2.5.9 Compass

An instrument that shows orientation. [59]

2.5.10 Accelerometer

Accelerometer is device that reacts to changes in speed or direction and measures it. It measure non-gravitational acceleration. [60]

2.5.11 Barometer

Instrument that measures the air pressure. Used in the flight controller to measure the altitude of the drone without using gps. [61]

2.5.12 SITL

SITL allows you to run ArduPilot simulations on your PC directly, without any hardware. It takes advantage of the fact that ArduPilot is a portable autopilot that can run on a very wide variety of platforms. When running in SITL the sensor data comes from a flight dynamics model in a flight simulator. ArduPilot has a wide range of vehicle simulators built in, and can interface to several external simulators. This allows ArduPilot to be tested on a very wide variety of vehicle types. For example, SITL can simulate:

- multi-rotor aircraft
- fixed wing aircraft
- ground vehicles
- camera gimbals
- antenna trackers
- a wide variety of optional sensors, such as Lidars and optical flow sensors
- Adding new simulated vehicle types or sensor types is straightforward.

A big advantage of ArduPilot on SITL is it gives you access to the full range of development tools available to desktop C++ development, such as interactive debuggers, static analyzers and dynamic analysis tools. This makes developing and testing new features in ArduPilot much simpler.
[62]

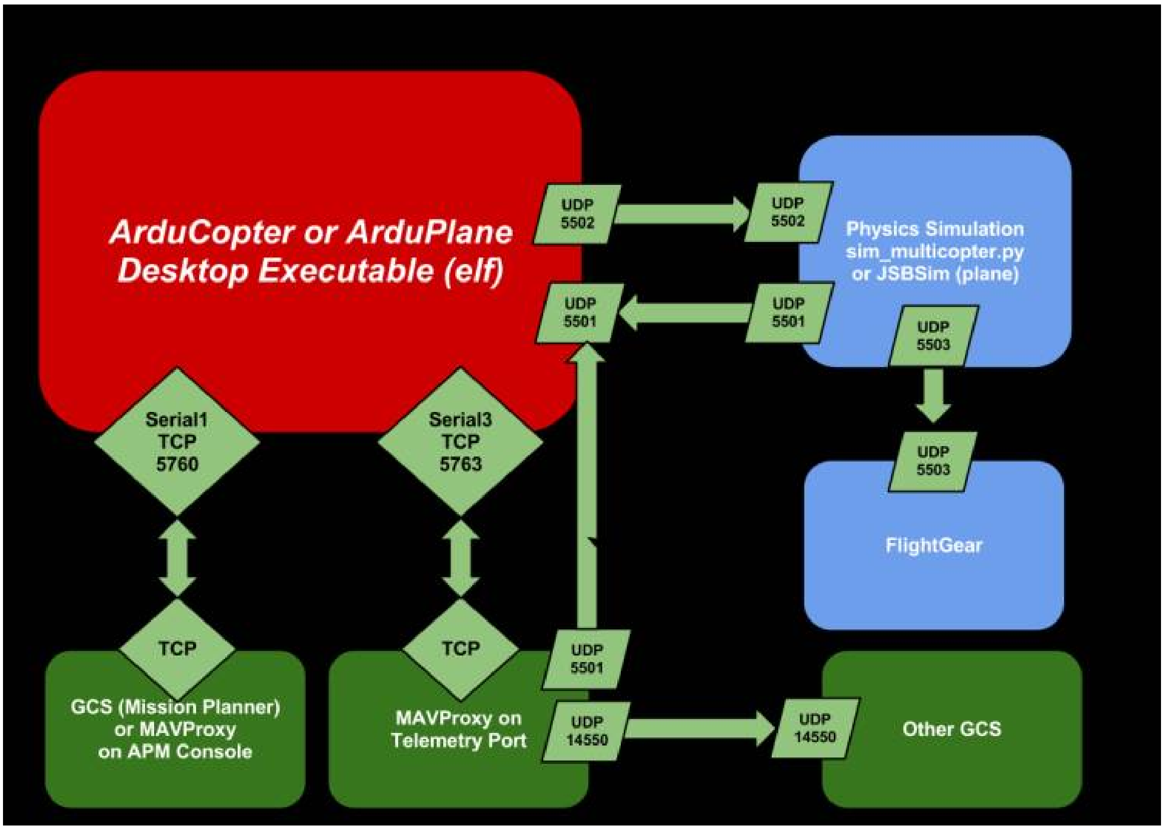

Figure 1: SITL explanation image

2.5.13 PPM

Pulse position modulation is a signal modulation used for both analog and digital signal transmissions. PPM transfers data with short pulses. Every pulse have the same amplitude and width. The one parameter that changes in the data transfer is the time between a pulse.

2.5.14 SBUS

Futaba SBUS is a bus system for R/C vehicles. Normally you have to use one cable per channel or every motor or servo you control. SBUS enables you to use one cable for up to as many channel as your transmitter has. Its simplifies the setup, removes the a lot of cable clutter on the vehicle. Meant to used on flight system on drones, planes and copters. Channels cant be mismatched when using SBUS. [63]

2.5.15 PWM

Pulse width modulation is a technique to modulate a signal. PWM is modulation can be done in two ways either by varying the frequency or the duty cycle. Duty cycle in short how long the signal stays high (1) compared to the time the signal stays low (0). PWM is normally used for controlling the power supplied to electrical devices such as motors, leds and much more. Its a way of getting analog control digitally. [64]

2.5.16 Serial UART

UART or Universal Asynchronous Receiver/Transmitter is a key component in serial communication subsystem of a computer. UART can be explained as taking the bytes of data and transfers individual bits into sending it sequential. At the receiver(destination) the second UART builds the bits together into complete bytes. [65]

2.5.17 Telemetry radio

Telemetry radio can be explained as radio used for transferring flight data such as position, speed, altitude, rpm. In general its radio communication for transferring flight data in real time. [66]

2.5.18 RTK

Real Time Kinematic (RTK)[33] satellite navigation is a technique used to enhance the precision of position data derived from satellite-based positioning systems (global navigation satellite systems, GNSS) such as GPS. It uses measurements of the phase of the signal's carrier wave, rather than the information content of the signal, and relies on a single reference station or interpolated virtual station to provide real-time corrections, providing up to centimetre-level accuracy.

2.5.19 Sonar

Sonar (SOund Navigation And Ranging)[34] is a technique that uses sound propagation to navigate, communicate with or detect objects on or under the surface of the water. Two types of technology share the name "sonar": passive sonar is essentially listening for the sounds in the environment; active sonar is emitting pulses of sounds and listening for echoes. Sonar may be used as a means of acoustic location and of measurement of the echo characteristics of "targets" in the water. Acoustic location in air was used before the introduction of radar. Sonar may also be used in air for robot navigation, and SODAR (an upward looking in-air sonar) is used for atmospheric investigations. The term sonar is also used for the equipment used to generate and receive the sound.

2.6 Gimbal

A Gimbal is a system of supports mounted to give stability and freedom of movement around a set amount of axises. The gimbal will hold a tool or some equipment of some sort, usually a sensor. In our case we use the gimbal for a camera, allowing the drone to move and act autonomously with visual input. This allows the drone to act in changing and difficult situations where simply relying on GPS co-ordinates would be impossible. In other cases a gimbal can be used to carry secondary equipment, claws, sonar, or range finders.

2.6.1 2-axis

2-axis gimbals is a gimbal that has roll and pitch.

2.6.2 3-axis

3-axis gimbals is a gimbal that has roll, pitch and yaw that gives a greater freedom i camera angles.

This project consider drones that need to maneuver in challenging and/or changing conditions. It is therefore important to ensure the drone has the possibility to extend its field of vision as far as possible. In cases where the drone is to operate autonomously it needs to "see" everything around, under, and, if possible, above itself. In most cases a line of sight skywards will not be possible as the gimbal hangs under the drone. A third axis gives stability to the camera as the drone maneuvers and stabilises relative to weather conditions.

2.7 Tuning

2.7.1 PID-controller

PID tuning and what corrections they make on the drone.

P=Proportional Relates to immediate correction. The further off it is from what you want the bigger correction on make.

I=Integral Steady state correction or over time correction. If it is not making any progress then add more corrections.

D=Derivative If the correction made on the drone are going to fast or to slow. Then slow it down to dampen the overshoot or avoid it completely.

2.8 Drone types

[70]

2.8.1 Quadcopter

[71] A quadcopter is a multirotor helicopter that is lifted and propelled by four rotors. Quadcopters are classified as rotorcraft, as opposed to fixed-wing aircraft, because their lift is generated by a set of rotors (vertically oriented propellers). Quadcopters generally use two pairs of identical fixed pitched propellers; two clockwise (CW) and two counter-clockwise (CCW). These use independent variation of the speed of each rotor to achieve control. By changing the speed of each rotor it is possible to specifically generate a desired total thrust; to locate for the centre of thrust both laterally and longitudinally; and to create a desired total torque, or turning force.

Quadcopters are the easiest to use and simplest to sett up. Quads have the hughest amount of frame options and fewer complex parts means less chance for technical failures.

2.8.2 Hexacopter

A hexacopter is a multirotor helicopter that is lifted and propelled by six rotors. Hexacoptors generaly has two sets of three rotors, clockwise and counter-clockwise. These rotors are placed alternating.

A hexacopter will generate more lift at higher stability than a quadcopter, but will therefore require more power to achieve the same amount of air time at max power. They also has a slight, but noticeable increase in yaw controll.

2.8.3 Octocopter

A octocopter is a multirotor helicopter that is lifted and propelled by eight rotors. The rotors on an octocopter alternates between clockwise and counter-clockwise.

Due to size limitations on the motors and the frame itself, eight rotors is the highest practical number of rotors on a single craft. Octocopters provide the highest amount of lift and stability among the copter drones, at the cost of the highest power consumption per time.

2.8.4 Motor redundancy

Where a quadcopter will loose control as soon as a single motor fails, larger numbers of motors grant a degree of redundancy. Hexacopters will be able to land controlled with the loss of a single motor and might make a controlled crash if two motors not adjacent to each other fail. An octocopter can potentially land with only four remaining motors active as long as they are placed in an X relative to each other. Regardless of motor number, the loss of half the drone's thrust can have a more noticeable effect as it is no longer able to support its weight, forcing the drone to the ground in an uncontrolled crash landing.

2.9 Multirotor movements

A multirotor drone maneuvers by increasing or decreasing thrust on specific motors in relation to what is needed. Movement in any direction is facilitated by tilting the drone towards the desired direction and then increasing thrust, more so on the motors furthest away from the new heading.

Drone tilt is controlled by slightly increasing thrust on the motors on one side while slightly decreasing thrust on the opposite motors. This gives the drone a high degree of control over its tilt and thus also its acceleration. This means acceleration is limited by drone weight as higher weight means more thrust must be dedicated to holding the drone airborne leaving less power for maneuvering.

Copter drones are also able to rotate. Rotors on a multirotor drone spin either clockwise or counter-clockwise, mounted alternating. Increasing the speed for all CW motors while decreasing speed on all CCW motors will spin the drone clockwise and vise-versa.

2.10 Camera

This section will explain the mechanics behind the cameras field of view.

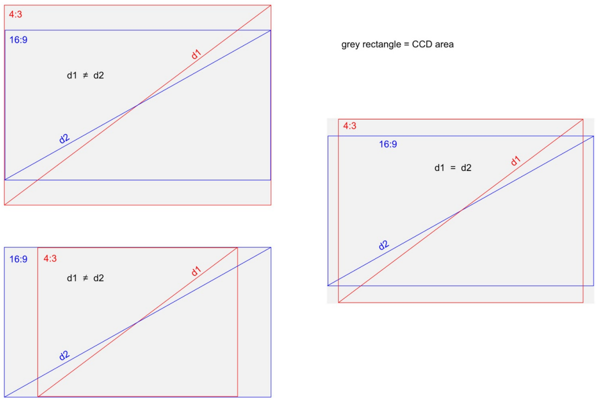

Figure 2: FOV 4/3 and 16/9 comparisons

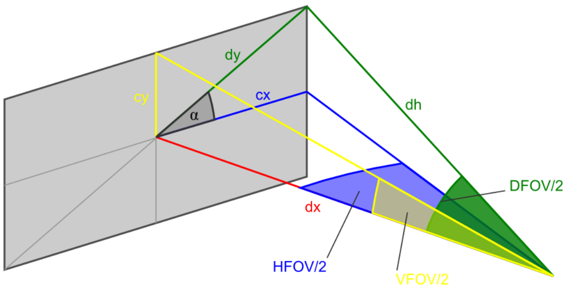

Figure 3: FOV explanation image

2.10.1 FOV calculation

[72]

$$dx = dh * cos(DFOV/2) \qquad dh = dx/cos(DFOV/2)$$
$$dy = dh * sin(DFOV/2) \qquad a = atan(9/16)$$
$$cx = dy * cos(a) \qquad cy = dy * sin(a)$$
$$HFOV/2 = atan(cx/dx) \qquad VFOV/2 = atan(cy/dx)$$
$$HFOV = 2 * atan(tan(DFOV/2) * cos(atan(9/16)))$$
$$VFOV = 2 * atan(tan(DFOV/2) * sin(atan(9/16)))$$

2.11 Safety and regulations

Norwegian laws regarding drone usage can be found in the appendix. This project adheres to them completely.

Drones used for recreation, sport or competition fall under the regulations and laws regarding use of model planes. Since this project does not fall under any of those categories we will have to follow the regulations of RPAS(Remotely Piloted Aircraft Systems). RPAS has three categories RO1, RO2, and RO3. This drone falls under the RO2/RO3 categories and as such requires a license. This licence is provided by NTNU Ålesund. The regulations and laws can be found, in their entirety, in the appendix.

[67] [68] [69]

3 MATERIALS AND METHODS

3.1 Hardware

3.1.1 DJI S1000+

The drone is a large octocopter made for aerial and cinematic photography. Its built mainly out of high strength carbon fiber to keep the weight down and maintain good stability and high structural integrity. For convenient transport the drone has fold able arms.The drone is powered by 8 motors at 500w with a combined power output at 4000w. The high power output gives a remarkable lifting capability and stability. It has a dry weight (without battery) of 4.4kg and full lift capacity up to 11kg. It is powered by a 6 cell lipo battery[26].

Figure 4: DJI S1000+ multirotor drone

3.1.2 Pixhawk PX4 flight controller

The pixhawk flight controller enables our autopilot to control the drone. The pixhawk has several inbuilt sensors including 3D accelerometer, gyroscope, and barometer. The pixhawk is also connected to an external GPS and compass. Pixhawk flight controllers are built to run ardupilot by default[27].

Figure 5: Pixhawk PX4 flight controller

3.1.3 DYS Smart 3-axis gimbal

This is the gimbal used for the project, recommended by the creators of ardupilot. Delivered with a alexmos 3-axis controller running basecam firmware. Configurable via gui on pc.

Figure 6: DYS Smart 3-axis gimbal

3.1.4 Storm32-Brushless gimbal controller

Storm32 is a common gimbal controller, made for controlling 3-axis brushless gimbals.The firmware and original controller was created by a member of the drone comunity(Olliw). Storm 32 is not a commercial product, but an ongoing effort by several hobby developers. The community maintains a wiki for the controller where they have guides and chat rooms specifically for the storm 32 system.

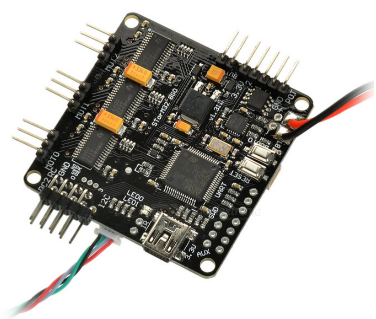

Figure 7: Storm 32 brushless controller

3.1.5 Turnigy 9XR

Turnigy 9XR is a programmable transmitter with support for up to 24 channels.

Figure 8: 9XR controller

3.1.6 FRSKY X8R and XJT 2.4GHZ

Telemetry receiver for communicating with the 9XR transmitter. Supports 8 channels via pwm and 16 channels via SBUS. XJT is the telemetry module for the transmitter communicating over 2.4ghz. The system in the XJT shift 80 channels hundred times a second to ensure a secure connections between the transmitter and the receiver.

Figure 9: FrSky XJT telemetry module

3.1.7 Logitech C920

[72] Camera used for computer vision. Max resolution of the camera is 1920x1080. The field of view of the camera in 16/9 resolution is 70.42 degrees horizontal and 43.3 degrees vertical.

Figure 10: Logitech C920 camera

3.1.8 Odroid XU4

Companion computer for the drone that runs computer vision code and sends commands to the flight controller. The autopilot will be running on the Odroid.

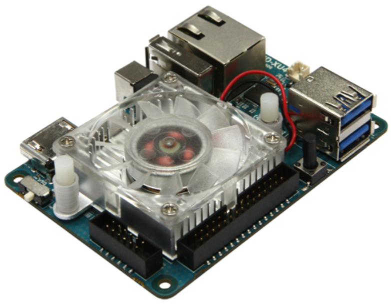

Figure 11: Odroid XU4 companion computer

3.1.9 Nvidia Jetson TX1

Jetson TX1[35] is capable of delivering the performance and power efficiency needed for the latest visual computing applications. It's built around the revolutionary NVIDIA Maxwell™ architecture with 256 CUDA cores delivering over 1 TeraFLOPs of performance. 64-bit CPUs, 4K video endcode and decode capabilities, and a camera interface capable of 1400 MPix/s make this the best system for embedded deep learning, computer vision, graphics, and GPU computing.

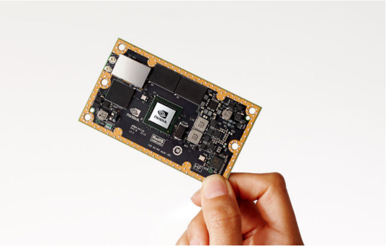

Figure 12: Nvidia Jetson TX1 Embedded System

3.1.10 Batteries

Lipo batteries for the drone. One 6cell lipo for powering the drone and two 3 cell batteries one for the gimbal and one powering the odroid and flight controller.

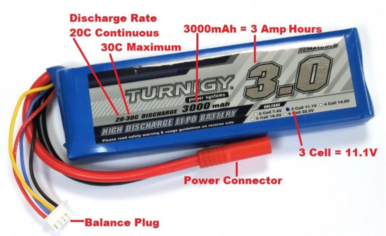

Figure 13: Example of a battery

3.2 Drone setup

The drone we use is a DJI s1000, but we use the Pixhawk autopilot flight controller. DJI and Pixhawk use a quite different connection system between the outputs on the flight controller and the inputs of the drone motors. The first issue we faced was defining motor 1, or "front right". Pixhawk M1 runs clockwise while DJI M1 runs counter clockwise. there are two different ways we could solve this problem, the most thorough would be to disassemble the individual motors, move each of them one step either direction and then reassemble. This solution would take a long time to implement and carries a substantial risk for mistakes with potentially critical results.

The other option was simply to shift forward direction one step to either side, this shifts forward direction slightly off centre but as the drone is balanced in all directions it should not affect flight characteristics under autopilot control. This does make it slightly harder to fly the drone manually as front is no longer clearly marked. Shifting direction this way also results in forward movement being slightly off in relation to the landing gear, this can potentially cause problems if landing at speed. We considered this to be an acceptable risk as the drone should not need to land with directional momentum in any potential cases during use.

Another compatibility issue between Pixhawn and DJI, perhaps the greater of the two is output/input numbering. While the outputs on a DJI controller sends signals to a corresponding DJI motor and a pixhawk controller to pixhawk motors, they do not use the same connections. DJI output 1 corresponds with DJI motor 1, but Pixhawk output 1 does not correspond to DJI motor 1. Due to this the wiring between the Pixhawk and the motors does not follow standard 1-1, 2-2 system, but a custom connection detailed here: connected motor represented by M#, connected Pixhawk output represented by P#.

This following table represents the finished connections including the shift to accommodate a "front right" clockwise motor.

DJI S1000 input –>	Pixhawk output
M1 –>	P3
M2 –>	P1
M3 –>	P5
M4 –>	P7

M5 –> P6

M6 –> P2

M7 –> P4

M8 –> P8

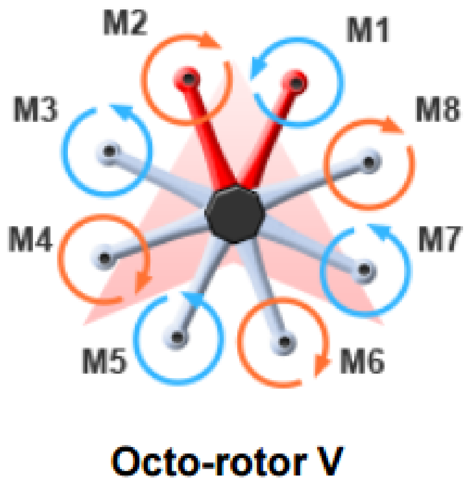

Figure 14: DJI config system

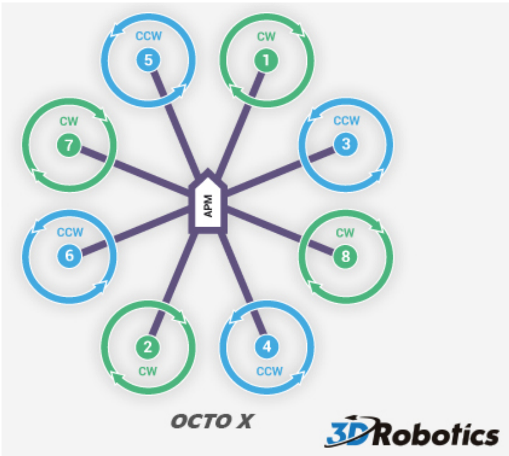

Figure 15: Pixhawk config system

In figure 14 and in 15 you can see the differences in motor numeration and rotation direction between DJI system and Pixhawk systems. Note front right motor in relation to flight direction.

3.3 Gimbal Calibration

The gimbal is mounted at the front of the drone, on a platform under the main body in line with the batteries. We are using a Logitec webcam to send frames back to the companion computer for processing. This camera is not designed for use with a gimbal so we had to insert a small mounting to make everything fit together. The camera has a relatively hard cable, this prevents the gimbal from moving to extreme points along the yaw and pitch axies. Our physical setup allows 90 degrees of yaw each direction for a total of 180 degrees forward freedom of movement and 45 degrees skywards / 80 degrees groundwards pitch. These limitations in yaw should not have a critical effect on the operations of the drone as it will shift to follow the camera, attempting to keep it faced forwards. The limitations in pitch can potentially become more of an issue during landing as the camera is unable to point directly down the drone can not land accurately directly on a QR code, this is not a problem if the QR code can be moved to the side of the landing pad or a wall next to the landing pad.

3.4 Calibrating the drone

The drone has several tools to let i know what orientation, attitude, heading, altitude, and speed it has at any given moment.

Most of these will be calibrated before flight tests to ensure all values are up to date, and any potential shift in the equipment can be compensated for. These calibrations are done trough Mission Planer. All calibrations will be done after the drone is fully constructed as the sensors are inside the flight controller, and moving the flight controller relative to the drone will then put the autopilot out of sync with the frame.
The compass is calibrated by turning the drone so it faces all the cardinal directions and up/down in a predetermined order.

The accelerometer is calibrated by turning and holding the drone around every axis allowing gravity to pull the flight controller in every cardinal direction as well as up and down in a predetermined order.

The gyro is calibrated by turning it around every axis in the same way as the accelerometer, but there is no need to follow a predetermined order or hold any positions for any length of time.

3.5 Tuning the octocopter

To get the octocopter to behave as it should a lot of tuning have to get done since the flight controller is not specific made for this frame. In the mission planner software under the tuning section both advanced and basic tuning. A badly tuned drone reacts twitchy or sluggish to inputs. The tuning is important for making the autopilot and manual modes work optimal. A well tuned drone is much easier to control when flight condition gets rougher. As this is a large drone with lifting capacity it can handle a lot of wind and not as easily interrupted by it.

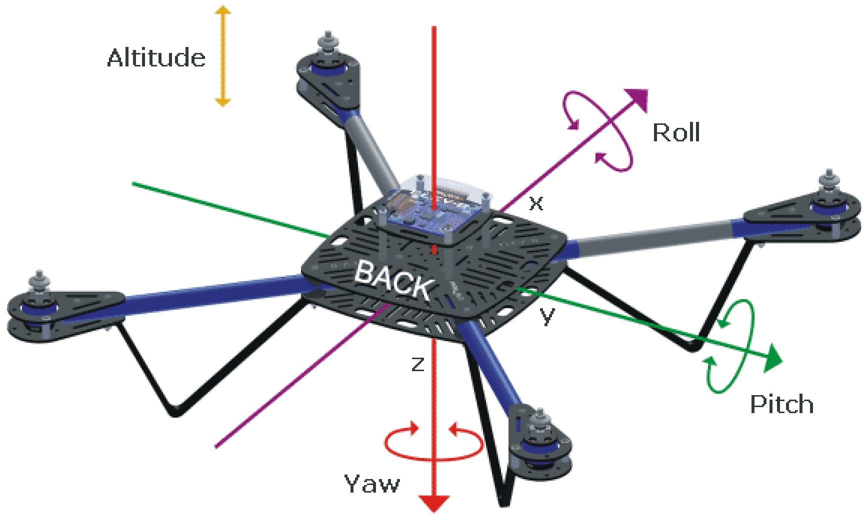

Figure 16: Axes of a drone

3.6 Tuning and how it corresponds to movement

3.6.1 Roll/Pitch tuning

Higher value gives more responsive roll/pitch inputs Too high the copter oscillate Too low the copter will be sluggish to inputs.

3.6.2 Yaw tuning

Too high copter heading will oscillate Too low copter can be unable to maintain heading.

3.6.3 Altitude tuning

-The altitude hold p converts altitude error (to low or to high) compared to desired altitude and actual altitude and converts it into desired climb rate or descent rate. Higher value means more aggressiveness in holding the altitude and too high gives a jerky throttle./par -Throttle rate converts the desired climb and descend rates into acceleration up and down./par -The throttle acceleration PID converts acceleration error into motor output./par

3.7 Calculating flight time

Calculations are based of parameters given by DJI and calculated by using matlab. The parameters given by dji are

- Weight of 9.5kg, battery of 15000mah, results in a flight time of 15min.
- Power consumption of 1500w when hovering 2meters above the ground.
- Maximum power consumption of 4000w.
- Reference flight time of 15min by above specs and 13.32min when calculated.

```
Tref=15;                        %min with 9.5kg and 1500mah battery
Tcor=Tref/13.32;                %corrections according to real data battery usage
                                %from excel chart 13.32min calculated
m=6:0.1:11;                     %kg drone weight
mref=9.5;                       %kg reference weight
C=10000:200:20000;              %mah Battery capacity
Cref=12000;                     %reference battery capacity
p=1500;                         %reference 9.5kg hover w Power
pm=(p/mref)*m;                  %power draw based of weight on drone
pmax=4000;                      %w Power
pdraw=900:100:4000;             %power draw
u=22.2;                         %V nominal voltage
i=pm/u;                         %constant current during hover for different weights
iref=p/u;                       %current for hover reference weight
Tm=((Cref/1000)./i)*60;         %based on weight
TC=((C/1000)./iref)*60;         %based on battery size
Tp=((Cref/1000)./(pdraw/u))*60; %Flight time based on power draw

TCcor=TC.*Tcor;
Tmcor=Tm.*Tcor;
```

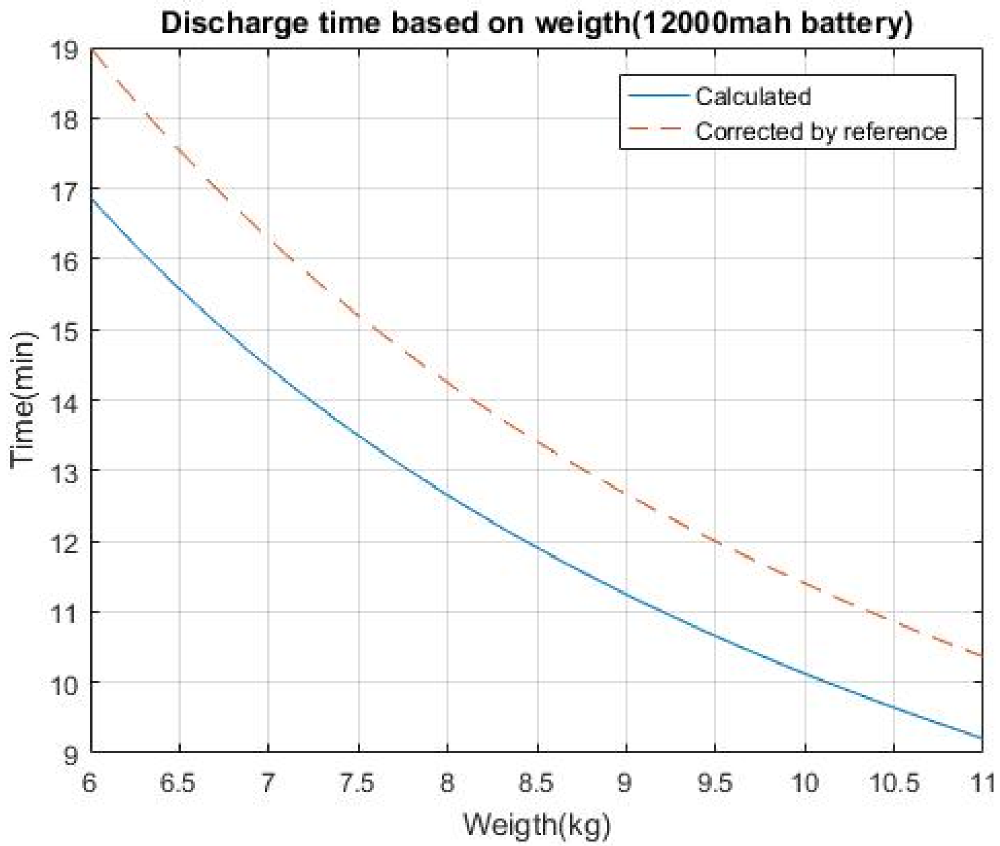

Figure 17: Graphs the battery life in relation to weight

3.8 Tracking speed calculations

```
%Loosing target from the camera
%Based on speed of target
D=10; %distance from target in m
T=0.3; %size of target in m
HFOV=70.42; %degrees
VFOV=43.3; %degrees
V=.5:0.1:5; %m/s
Hfovm=D*cosd(HFOV/2); %Horisontal fov in m
TT=Hfovm./V;  %tracked time before lost in s
```

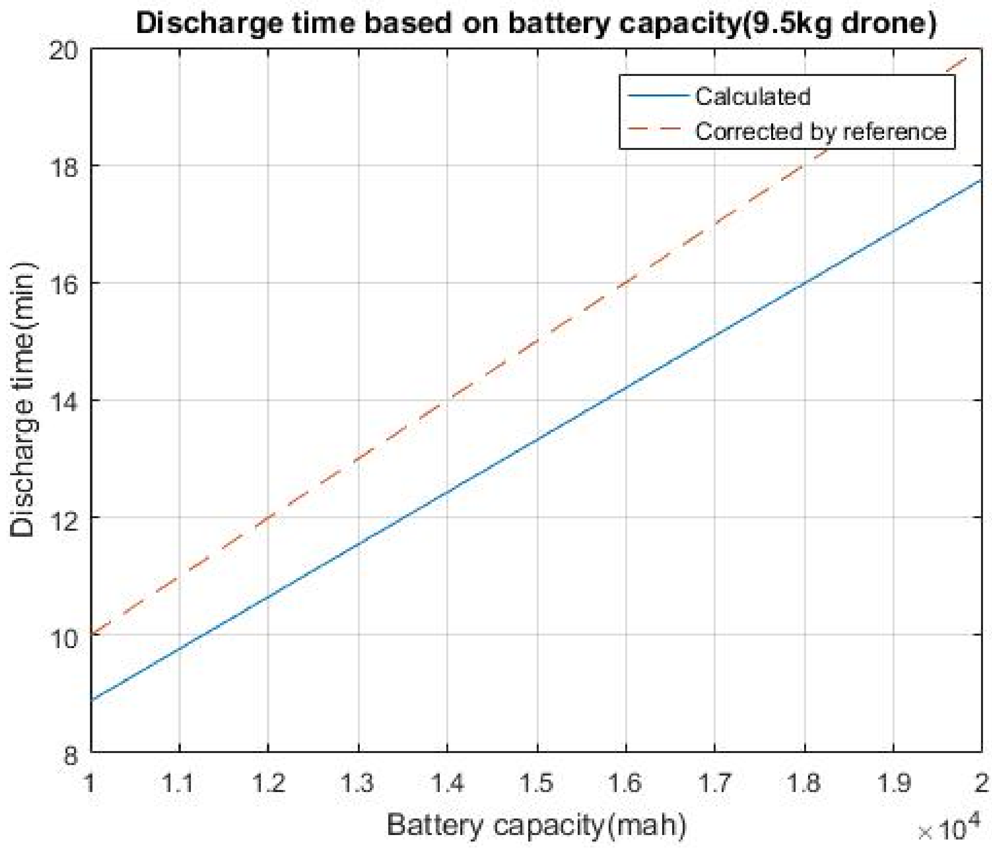

Figure 18: Graphs battery life time

Based on distance from target

```
close all; clear all;
D=1:0.3:20; %distance from target in m
T=0.3; %Target size m
HFOV=70.42; %degrees
VFOV=43.3; %degrees
V=3; %m/s
Hfovm=D.*cosd(HFOV/2); %Horisontal fov in m
TT=Hfovm./V; %tracked time before lost in s
```

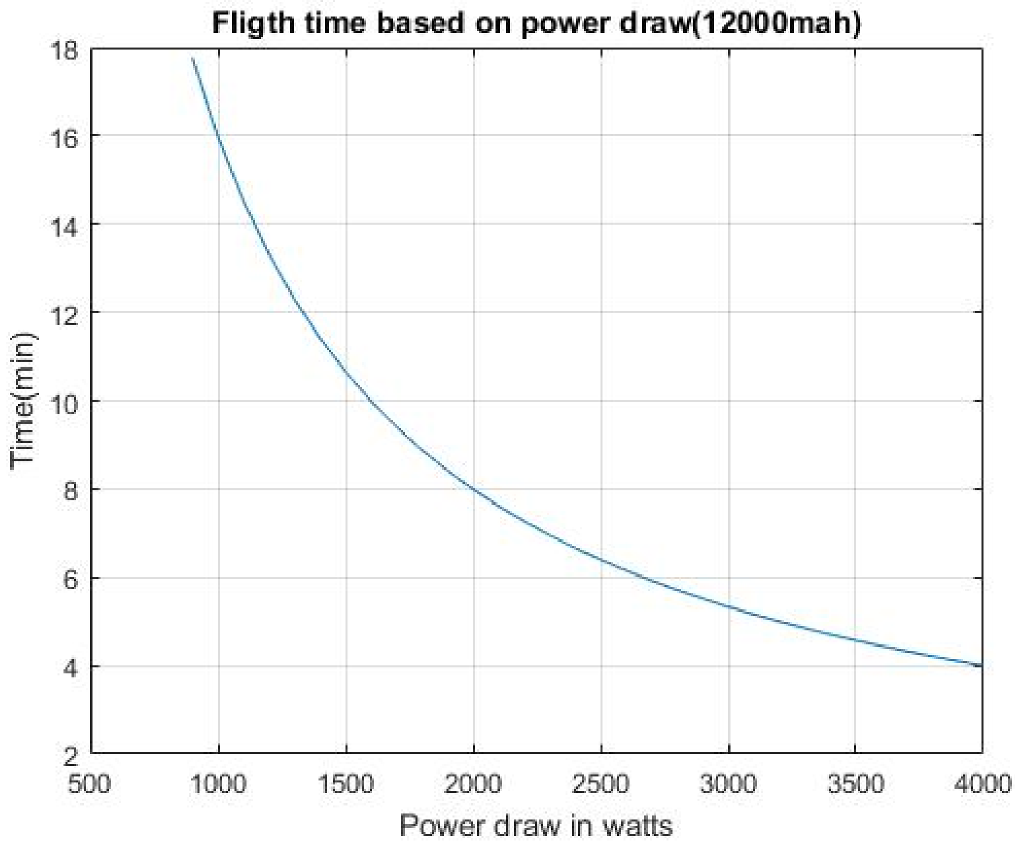

Figure 19: Graphs flight time

Distance from the target from where its in the lense based on lense

```
%distortion, 4:3 res from camera
HFOV=70.42; %degrees maximum angle of lense
VFOV=43.3; %degrees maximum angle of lense
D=20; %m distance to target middle of lense

x=D/cosd(HFOV/2);
y=D/cosd(VFOV/2);
z=0:0.4:D;

x1=0:0.3:x;
y1=0:0.3:y;
```

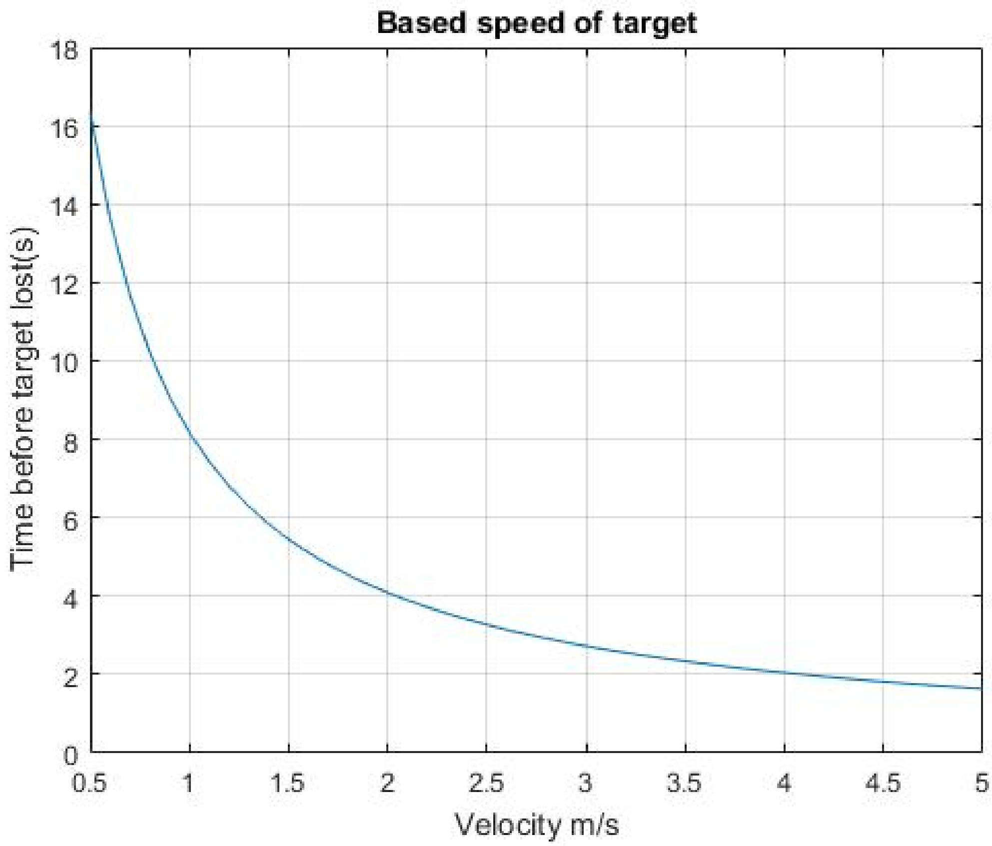

Figure 20: Tracking accuracy

```
V=0:0.1:5; %m/s
TTx=x./V;
TTy=y./V;
```

Tracking with gimbal rotation

```
%Position in frame hold the object around 25% of center of max
D=50; %distance from target in m
T=0.3; %size of target in m
HFOV=70.42; %degrees
```

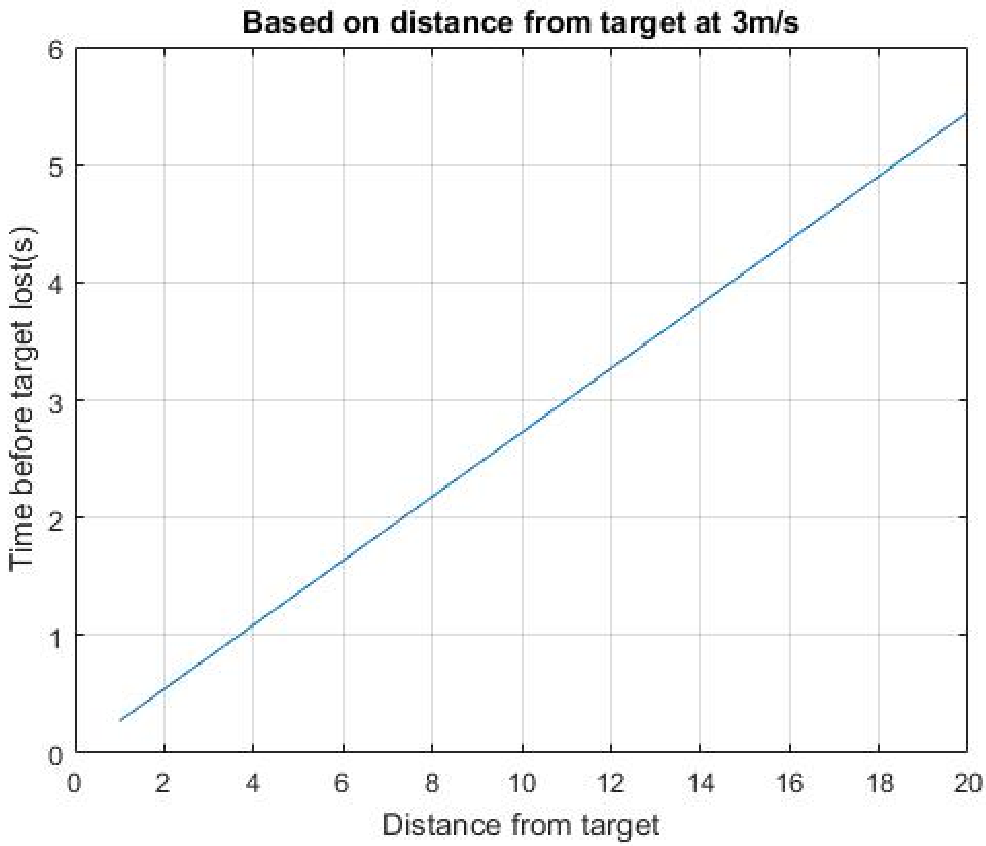

Figure 21: Target loss time

```
VFOV=43.3; %degrees
V=5; %m/s
V2=20; %degrees per second
S=0:1:10;
Hfovm=D*cosd(HFOV/2); %Horisontal fov in m
V2M=Hfovm/HFOV*V2;

V1=1:0.1:5; %m/s
VD=V1/Hfovm*HFOV; %d/s

Tl=Hfovm/2;%target location
```

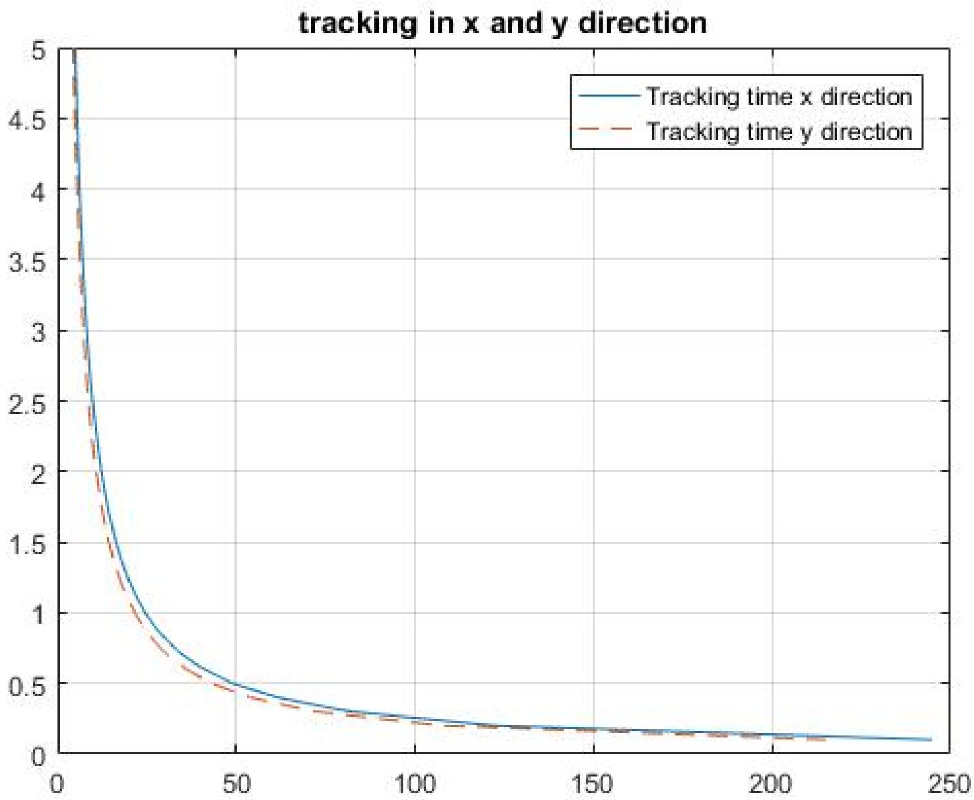

Figure 22: Tracking in XY direction

```
LIF=T1+V*S-V2M*S;  %Location in frame
Middley=Hfovm/2*ones(size(S));
```

3.9 Computer Vision

Vision based autopilot modes enables a drone to get higher intelligent autopilot.

3.9.1 Compiling opencv from source files windows/linux

Opencv has to compiled from source files to enable all needed features. Opencv contribution repository is needed since techniques for feature matching lies in this directories.

To compile opencv from the source files both the normal release repository and the contribution

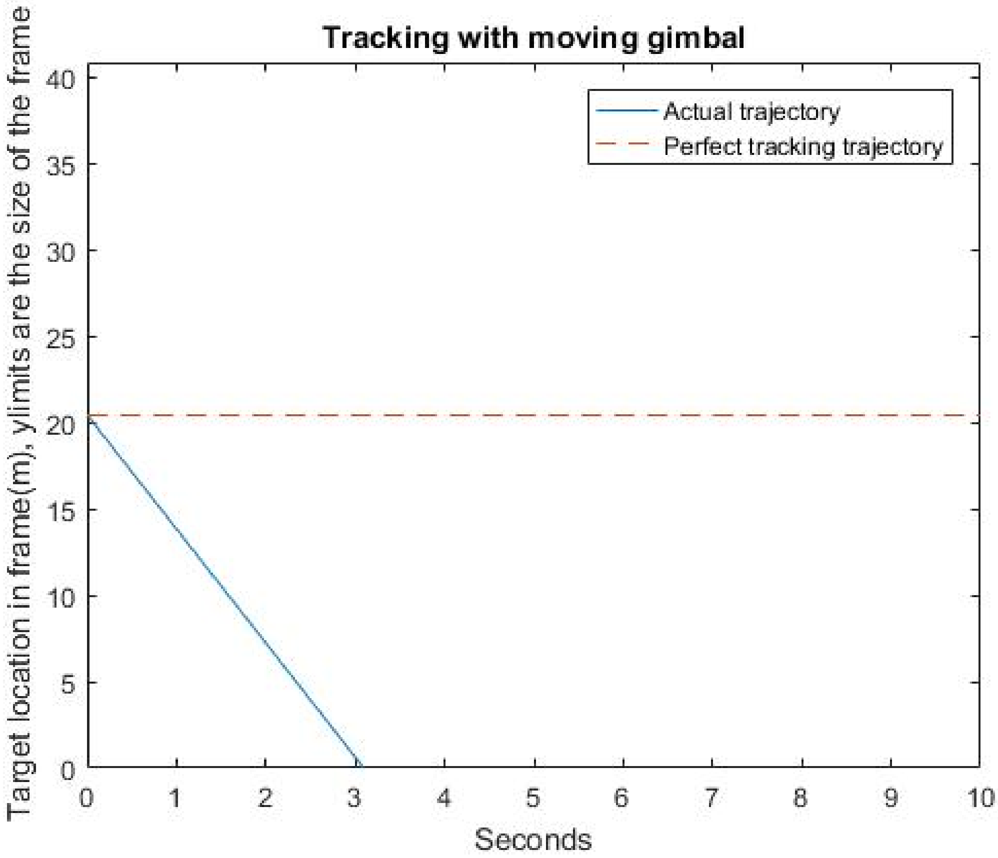

Figure 23: Tracking with moving gimbal

repository has to be downloaded from github. The software cmake are used to build the two repositories together either by running cmake in command line or the cmake-gui. When building the build directory by command line cmake has to be added to the environment variables in windows.

Making the build files in command line one has to be correct in the correct directory where you want to build the files. The files are built by using the command:

$$cmake - DOPENCV\_EXTRA\_MODULES\_PATH =< opencv\_contrib >$$

$$/modules < opencv\_source\_directory >$$

If done in linux this will be the same.

When the build is created, compiling it is the next step. Compiling in linux you need make installed and the code will be compiled by the command make -j(number of cores+1). In

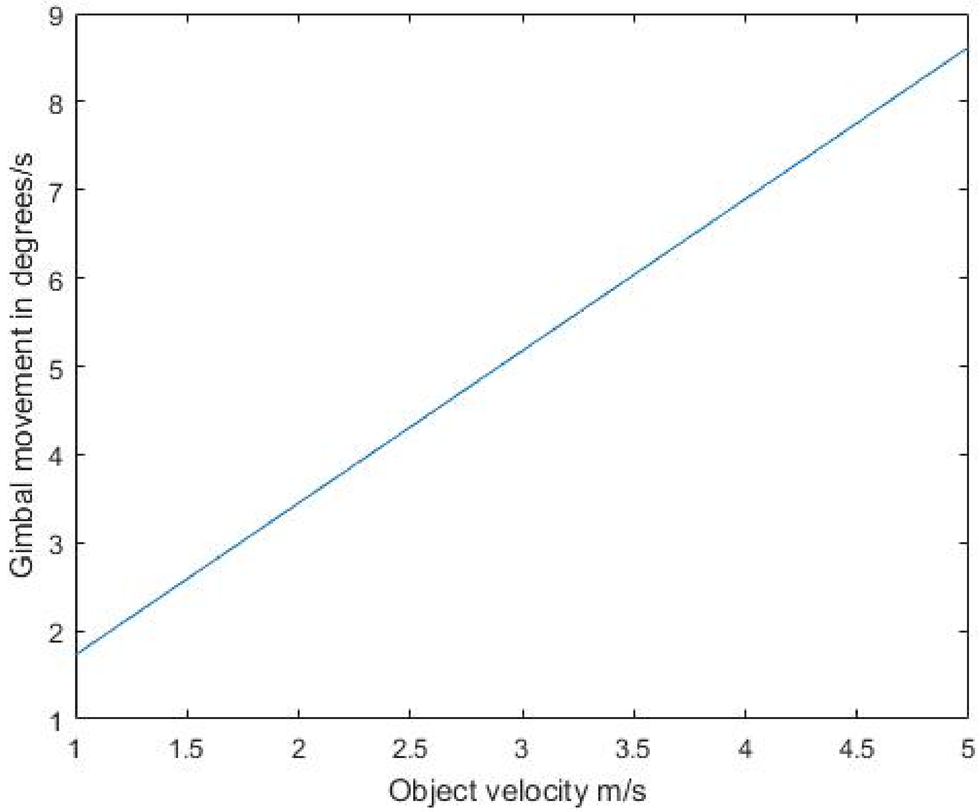

Figure 24: Required tracking speed

windows a compiler of own choice can be used. Visual studio has a built in compiler if package for c++ is installed. Here you open the main project file that created when making the build files. Then compile it by the built in compiler as a release and debug. When its compiled the files needed to run opencv in python, java and programs that can use opencv such as matlab will be created in their folders. Last step is to add opencv directory to path variables in windows.

3.9.2 Choosing a method for tracking

Computer vision is highly demanding task for any computer. The computer vision on the drone runs of the odroid xu4. When using computer vision in real-time performance is key. Since the drone is moving all the time and camera is not stationary it makes the process more complex. So algorithms such as frame differencing and background subtraction will not work as they are intended for stationary cameras. Since this a autopilot the technique for image processing must be robust and fail-safe.

3.9.3 SIFT implementation for tracking

The sift implementation starts with initializing a detector of sift type. Then it takes a training image of what you want to track and pre-process it before the camera is initialized. Detect and calculates the keypoints in the image so it doesn't need to be computed more than once.

For the live feed from the camera the keypoints will be computed for every frame. Matching the keypoints is done by flann based matcher. Then it goes trough the keypoints to check if they are good or not. The good keypoints are used to find the homography in both images. To draw they keypoints onto the final processed image a perspective transform is used so the keypoints will be correct according to placement and orientation.

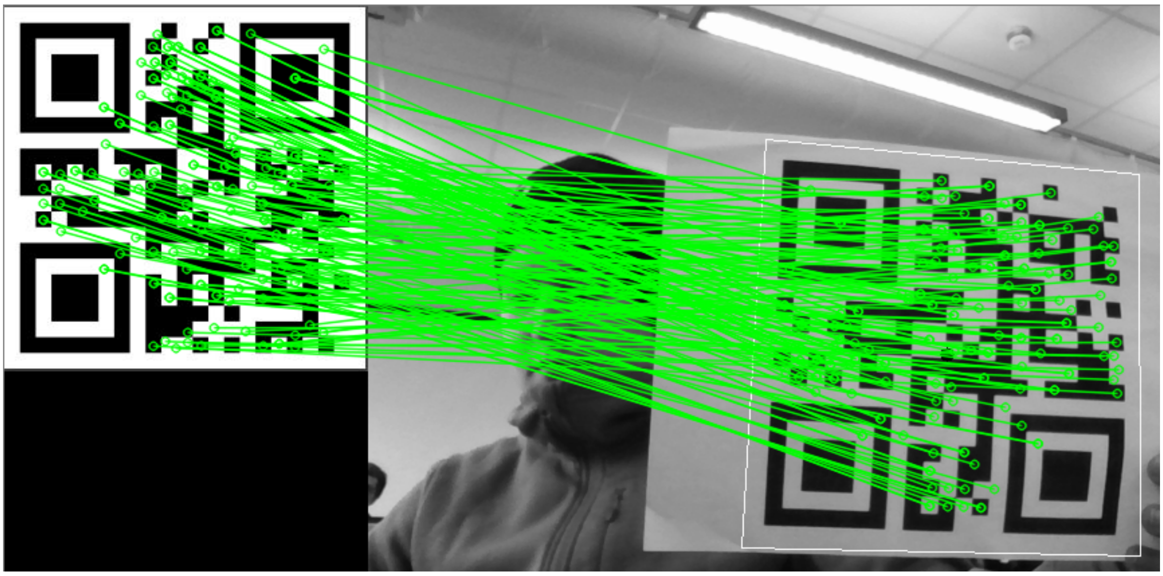

Figure 25: Matching example for the sift algorithm

3.9.4 Using color threshold and detection by blobs

The balloon detection starts with thresholding the camera image to only show the correct color. In the thresholded image the color will be represented as white and everything else as black. It then erodes the image to remove small particles and then dilates to grow it the large blobs back too its original size.

On the result of the thresholded image a blob detection is runned to detect circles. Next step is too classify the blobs so too small and too large is removed by the number of pixels. Last step is iterate threw the blobs too find the largest one. The largest one then is the target for the drone.

3.10 Autopilot Framework

One of the ideas of the project was to make an autopilot framework that was easy for others to build upon in the future. Preferably with the ability to create a 'building block' like GUI sometime in the future where a user could drag different 'modes' into a queue. The drone would then start with the first mode and move on the second mode in the queue when done with the first and continue on like this until the queue was empty.

We also wanted the ability to control the drone while waiting for the computer to process image information (as we mention in the computer vision chapter, this can take some time), as well as constantly save images to a video file regardless of image processing time.

Our solution to this is a module based program with three modules running separate processes. One module for capturing and saving images, this module will listen to a pipe to see if there are any other process requesting images from it. The second module will constantly poll images from the first module and process the images before sending the image information trough another pipe to the third module. This is of course only done if the autopilot requests processed images of any type, otherwise this process will sleep (to not use processing power). The third module is the autopilot module, that handles all logic related to the movement of the drone. We want to try to create an abstract class that all autopilot modes will be based upon. This will iteratively call 5 functions: one to update the status of the mode and check if there are any new processed images waiting, this function is common for all classes and not supposed to be overridden. The other four functions are abstract functions that we want developers to override when making new autopilot modes. The first one is to process image information (so that the other functions can act on 'fresh' information). The second logic to move the gimbal based on this new information. The third function is logic to move the drone. The fourth and last is to update the state of the autopilot mode, this function is supposed to return information on whether or not it is 'finished' or if we should continue to run more iterations of the same mode.

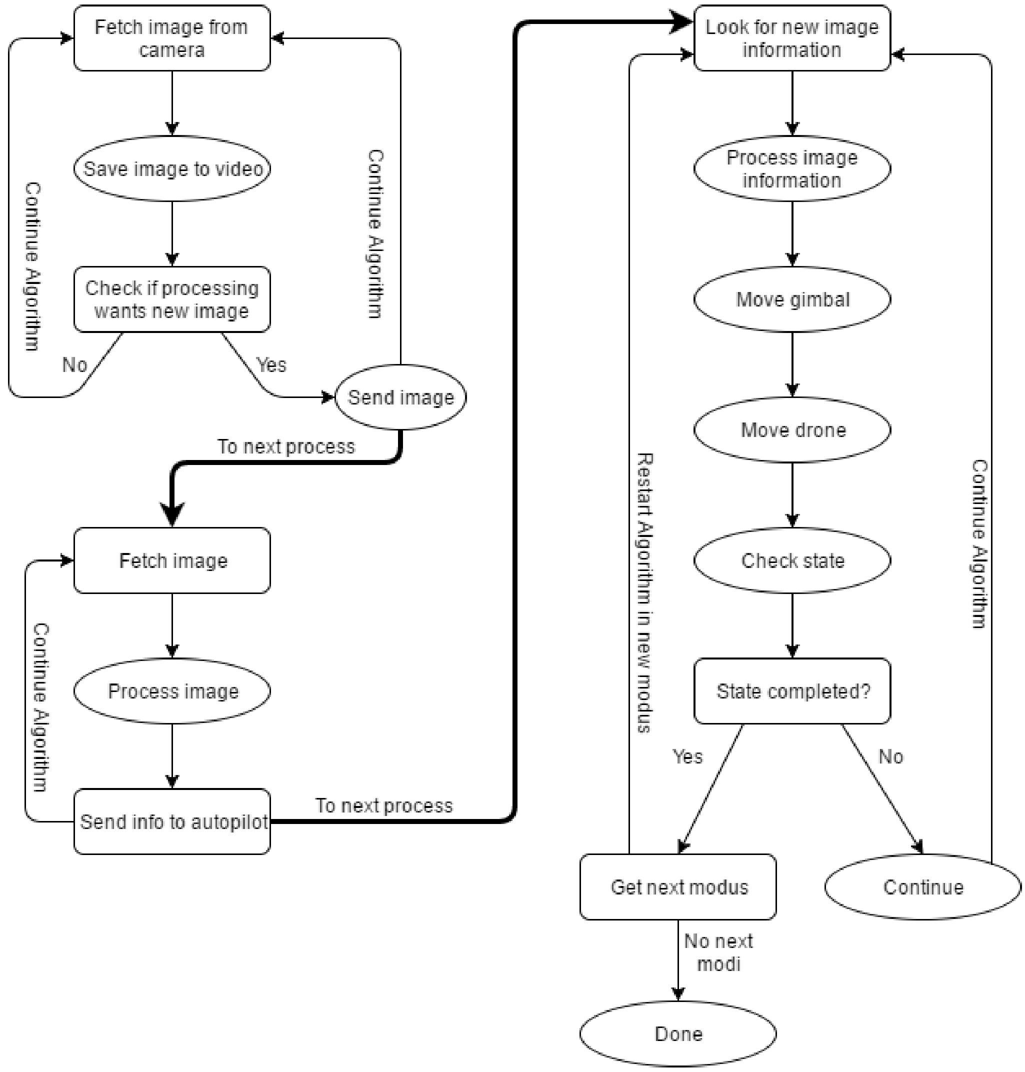

Figure 26: Logic for the three autopilot states

3.11 Autopilot logic

In this section we describe the logic for the autopilot we designed to solve the two tasks defined in the introduction of this thesis; Find and follow a predetermined target, find and land on a predetermined landing spot.

3.11.1 Autopilot modes

The group decided on making three autopilot modes for this project, where two of them uses image processing to decide the movement of the drone in relation to the target and the landing point.

The first mode is supposed to be a simple takeoff mode, this will initialize, arm and move the drone to a given altitude.

The second mode is the balloon finder mode, this will look for a target(represented by balloon) then pivot towards the target and maintain a set distance.

The third mode is a simple landing mode, this will bring the drone back to the start position where it will look for the landing pad(represented by QR code), then land, disarm and shut down on its own.

3.11.2 Positioning and movement calculations

To be able to act upon an object we are tracking with a camera we will need to calculate either its global position based on the drones position, or its position relative to the drone. The latter is less work and what we are planning on using to control the drone.

We have from 3.1.7 that the cameras vertical field of view is 43.3° while its horizontal field of view is 70.42°. We use this information, together with its pixel (based on resolution, we use 640*480), and an objects width (in pixels), to calculate an objects distance.
This is shown in formula 1.

$$width_e/(width_a * hfov/width_i) \tag{1}$$

Where $width_e$ is the expected width (in m), $width_a$ is the actualy width (camera, in pixels), $hfov$ is the cameras horizontal field of view and $width_i$ is the cameras width (in pixels).

3.11.3 Search mode calculations

Based on results from computer vision tests we have that with the QR code of size 0.91x0.91m that we will use for testing, we have a max detection range of 13.5m and an angle of around 50°. Based on these results we have concluded that we don't need to adjust the pitch of the

camera when searching for qr codes or other objects, as this will result in the targeting distance being to far away or that the search angle gets to small. We will for the sake of simplicity and to reduce the blind zone under the drone, as well as to account for inaccuracy in the measurements, reduce the angle from the drone to the furthermost point within the camera FoV from 50 deg to 45 deg.
This will give the gimbal a pitch of:

$$90° - \alpha + \frac{\gamma}{2} \tag{2}$$

α represents the pitch of the gimbal
γ represents the vertical field of view for the camera
With actual values: $90° - 45° + \frac{43.3°}{2}$.
These calculations assume that a pitch of 0° will point the gimbal directly forward in the horizontal plane.
With a maximum (absolute) distance of 13.5m, the maximum effective ground distance covered will be $sin45° * 13.5m = 9.55m$ and a drone altitude of $cos45° * 13.5 = 9.55m$
As the camera we currently use has a 43.3° vertical fov there will be a small blind zone below the drone where it cannot see, this will have an angle of $45° - 43.3°$ deg from a vertical line from the drone, a radius of $tan(1.7°) * a$, where a is the altitude of the drone, in this case the blind zone would be $tan(1.7°) * 9.55m = 0.28m$ long, covering an area of $\pi * 0.28m^2 = 0.25m^2$
This means that the actual area covered by the camera, and thus detected by the drone, will be between 0.28m and 9.55m from the drone resulting in a circle with an area of $\pi * 9.55m^2 = 286.5m^2$ subtracting the blind zone of $0.25m^2$ we have a total area scanned of $286.25m^2$.

These values are used in situations where the drone is looking for a QR code on ground level. When searching for a balloon the situation will be slightly different. The code and the camera will still have maximum ranges at which point they can no longer detect the target, but the angle is no longer a limiting factor. The balloon could potentially be on the same plane as the drone(or even higher), as such use of a gimbal would be ideal when looking for balloons, but not really make much of a difference in landing scenarios as either the distance would be too long or the angle too steep to find the target. In our tests the balloon will always be around 1.5-2m above the ground, as such we use the same values and settings for both modes.

Note: Values used in these calculations use the capabilities our software has at the time of writing, improved equipment or better code will increase the maximum detection range, this will allow the drone to fly higher and get a larger area within the cameras FoV. On the other hand improvements like these will likely not improve the codes abilities to detect the QR code at an angle, reducing the potential gain from increasing the cameras vertical FoV or pitching the camera further towards horizontal position.

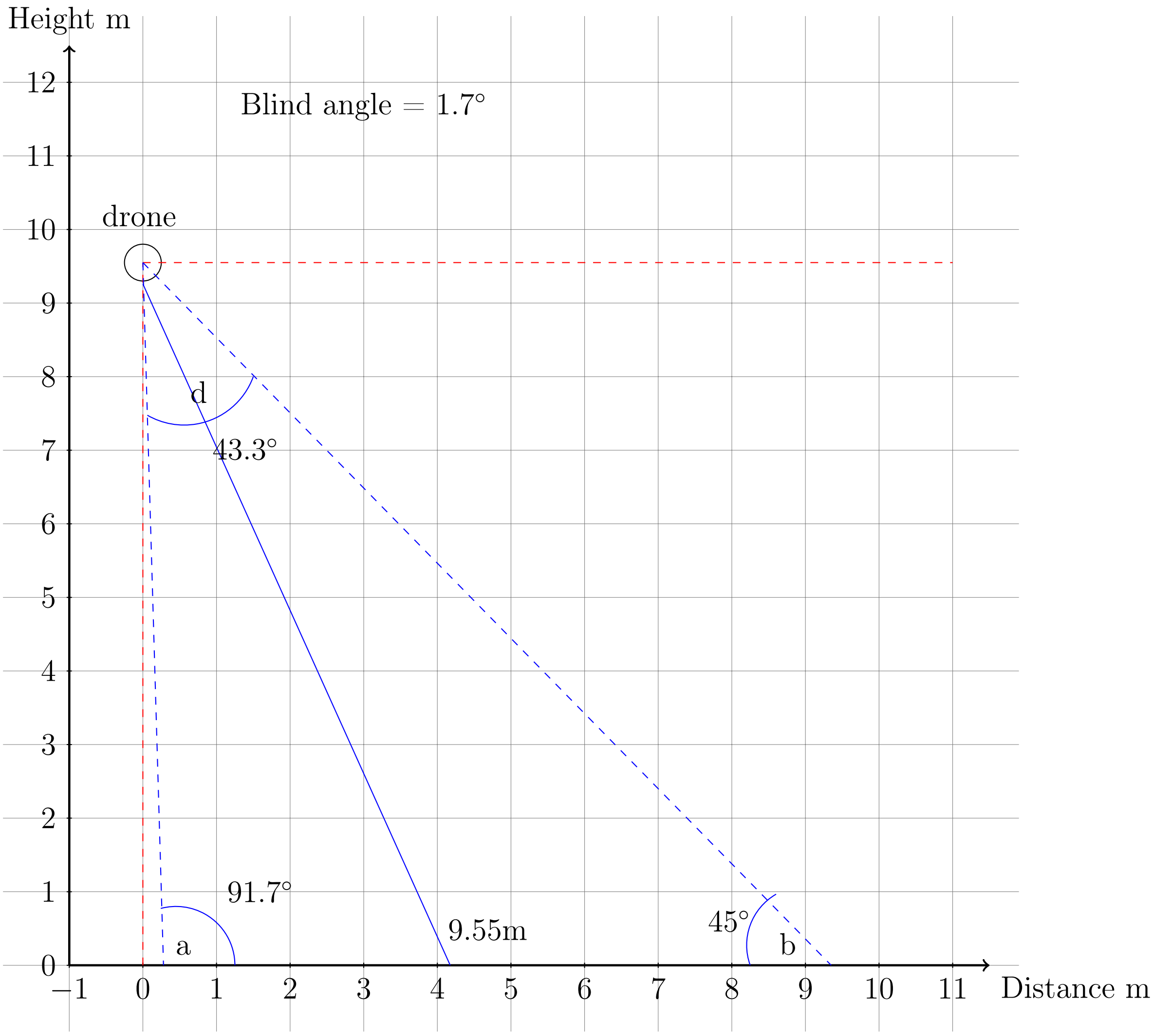

Figure 27: Drones fov from the side

The cameras horizontal field of view is 70.42°, meaning that when standing still, the drone will cover an area of

$$\frac{70.42^\circ}{360^\circ} * r^2 - \frac{70.42^\circ}{360^\circ} * r_2^2 \tag{3}$$

where r = ground distance coverage and r_2 is blind zone distance.

In our case: $\frac{70.42\circ}{360\circ} * \pi * 9.55m^2 - \frac{70.42\circ}{360\circ} * \pi * 0.28m^2 = 56m^2$
To cover the entire area around the drone, it would have to rotate and snap and process pictures at least $360°/70.42° = 5.1$ times evenly during the rotation (6 rounded up). In this case it would cover $pi * (9.55m)^2 - pi * (0.28m)^2 = 286.3m^2$ ground.

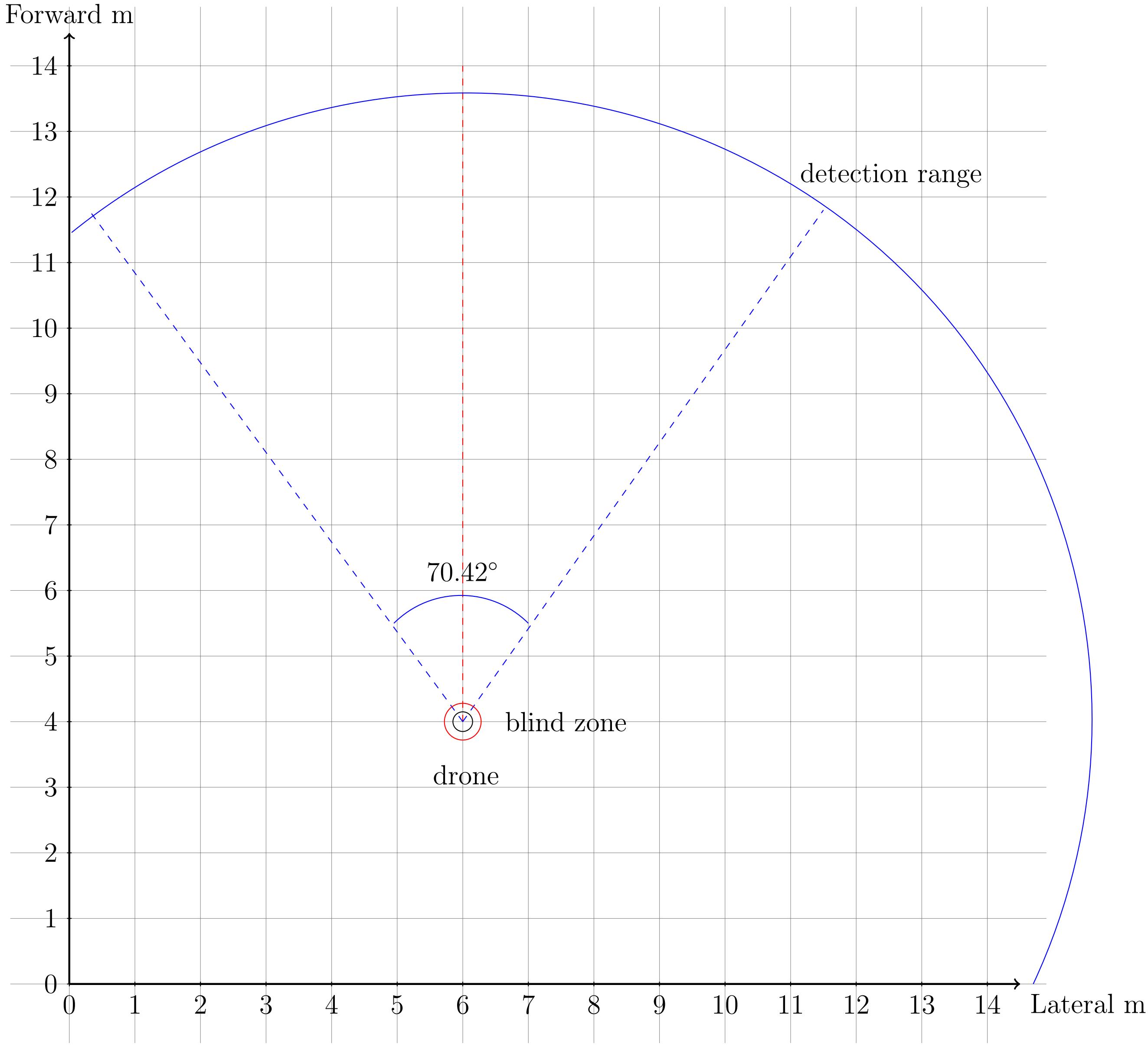

Figure 28: Drones FOV from top

To cover a larger area one would either have to adjust the angle, drone altitude or move the drone around instead of rotating in around one spot. The first two suggestions would require that we are trying to observe an object that can be detected from a distance further away than the qr code (13.5m) or with a steeper angle (angle from ground/qr code to drone, it has to be above 40deg in our case). The last suggestion will (depending on how we move the drone) require more time and therefore battery power, which is already very limited.

3.11.4 Working autopilot logic

Based on the data and calculations done in earlier sections of this chapter we have made plans for an algorithm and created a state diagram for both of our two main autopilot modes.

The qr code landing mode will be based around three states. In the first state the code will initialize the mode and make sure that everything runs as expected. When this is done it will enter the next state. In state 2 the drone will slowly rotate while searching for the qr code, if it finds the qr code it will enter state 3, otherwise it will abort and do an 'emergency landing' on the position it is in right now (this is mostly only for this project, and would not work if landing on a boat). In state 3 it will start moving slowly towards the qr code. This is done by adjusting the drones yaw and its relative x,y speed. We will also have to pitch the gimbal down and try to keep the camera in a position so that the qr code always is in the middle of the picture. When the gimbal is pitched 90 degrees below the drone and the qr code is in the middle of the picture we will initiate the landing procedure (we are now directly above the code). If we somehow lose track of the QR code during this process we will do an emergency landing.
The balloon finder code will be based around 4 states. The first state will, like in the qr landing mode, initialize the mode. When this is done we start state 2. State 2 will be similar to state 2 in the qr code mode, except that we will finish the rotation even if we find a balloon. This is to reduce the probability of error detection in the balloon finding computer vision algorithm, as the algorithm will focus on "best available". When the drone is finished rotating 360°, it will either rotate towards the best balloon it found and start state 3, or cancel the script if it didn't find a balloon at all. State 3 is simply a double check to see if the balloon is still there, if not it will cancel the script (here we could also have tried to re-find the lost balloon). If we find the balloon the code will enter state 4. State 4 is the mode where the drone will follow the balloon. This is done by calculating the distance and relative heading of the balloon, and then rotate towards its heading and adjust the drones relative x and y speed so that we head either towards it (if far away) or away from it (if to close). We do not adjust the drones z speed, that

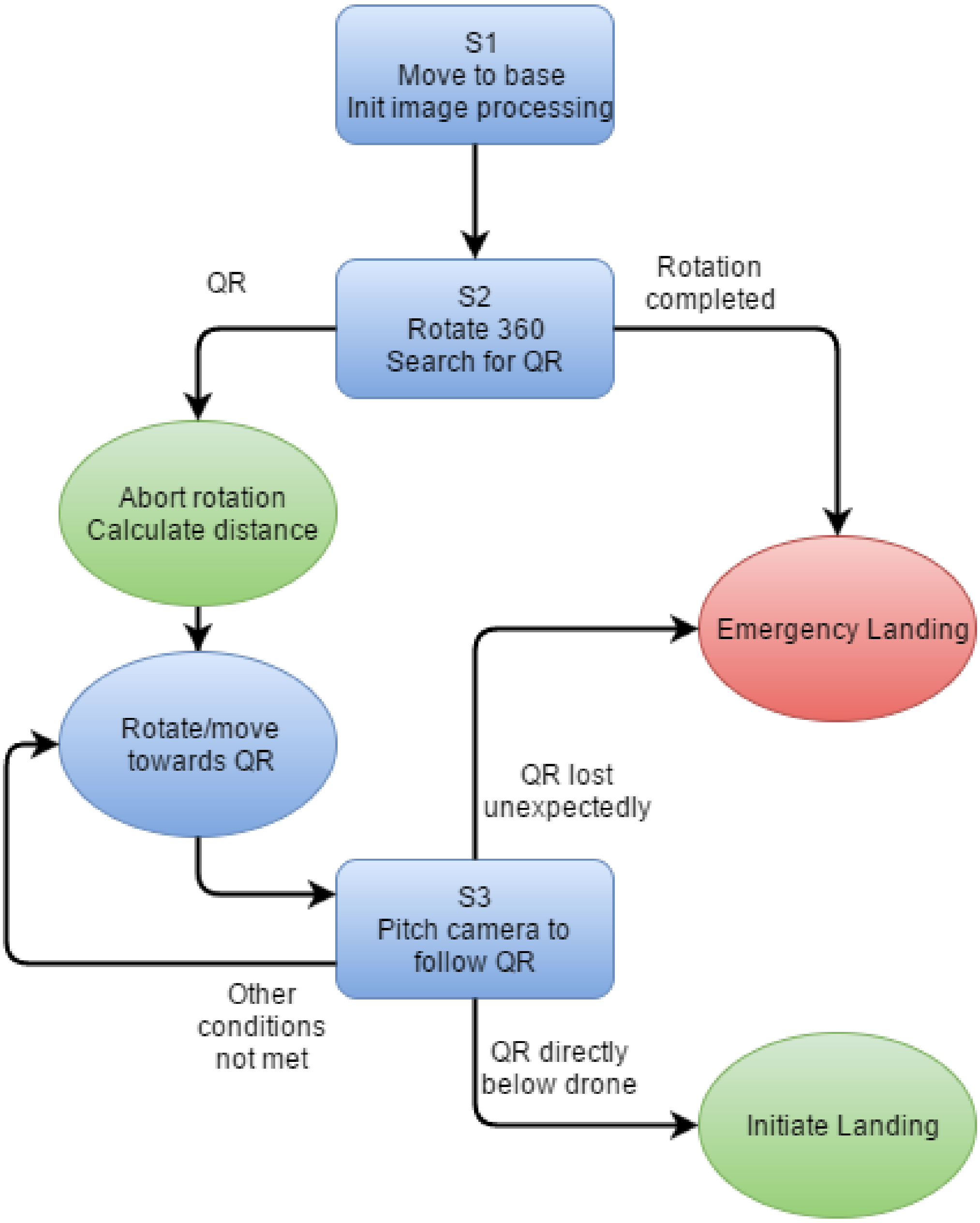

Figure 29: State diagram for landing logic

way we keep it at the same altitude all the time (this can only be done above even terrain). The code will have a timer so that the mode only runs for a given time. If we somehow lose track of the balloon before this we will (if not very close to the cancellation timer) try to re-find the balloon. This is done by returning to state 2, but instead of finishing a complete rotation, we try to find a balloon with similar data as the one we lost and head toward it immediately.

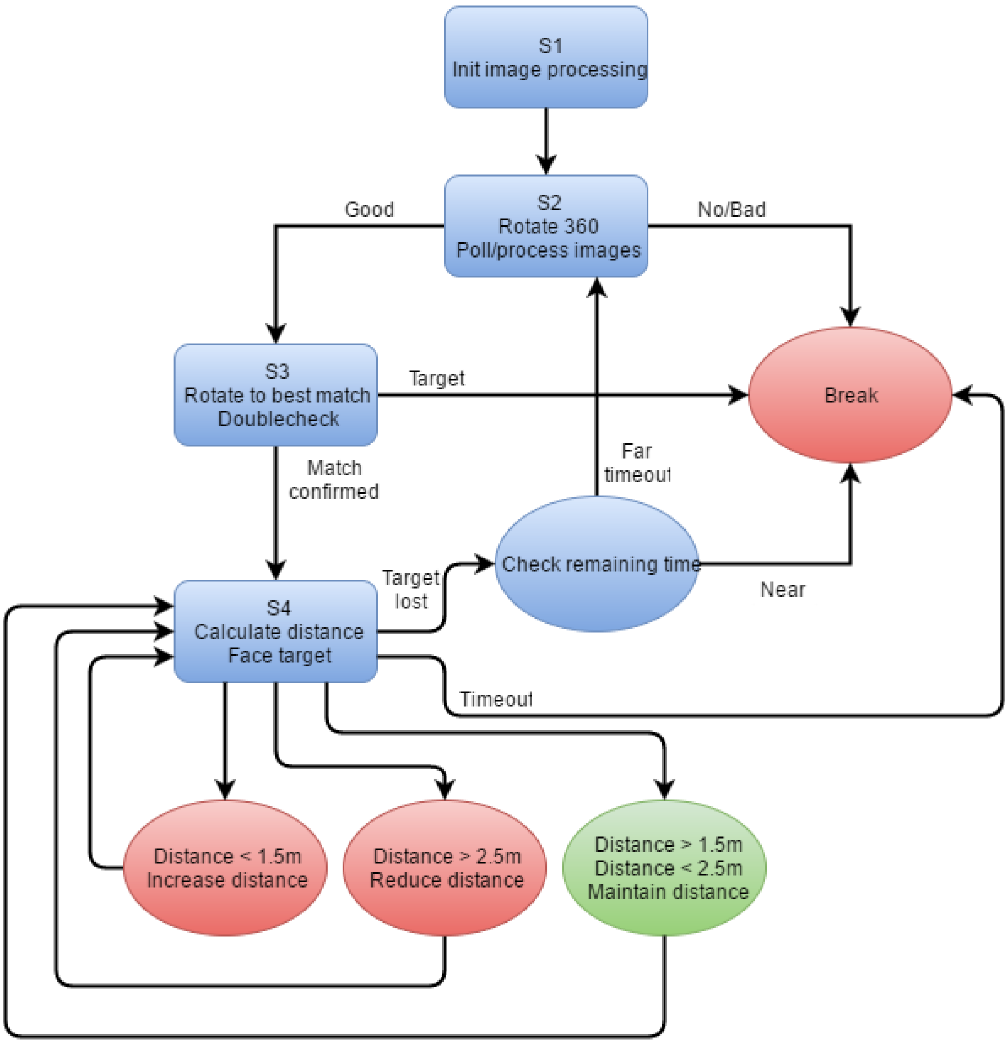

Figure 30: State diagram for balloon finder logic

3.12 Simulation of autopilot

When coding we use SITL to test our changes in the code, as it would be very inconvenient having to run the code on the drone each time we do a small change and want to test it. SITL (Situation in the loop) is as mentioned in 2.5.12, a piece of software that allows us to run a simulation of a drone. This way we can connect to a simulation on our own computer, instead of connecting to the actual drone. SITL uses the same communication protocols, and together with a camera connected to the computer, servers as a safe and stable way to test our code. We will also install SITL on the odroid, so that we can test the code with the Odroids hardware before we test it with the actual drone.

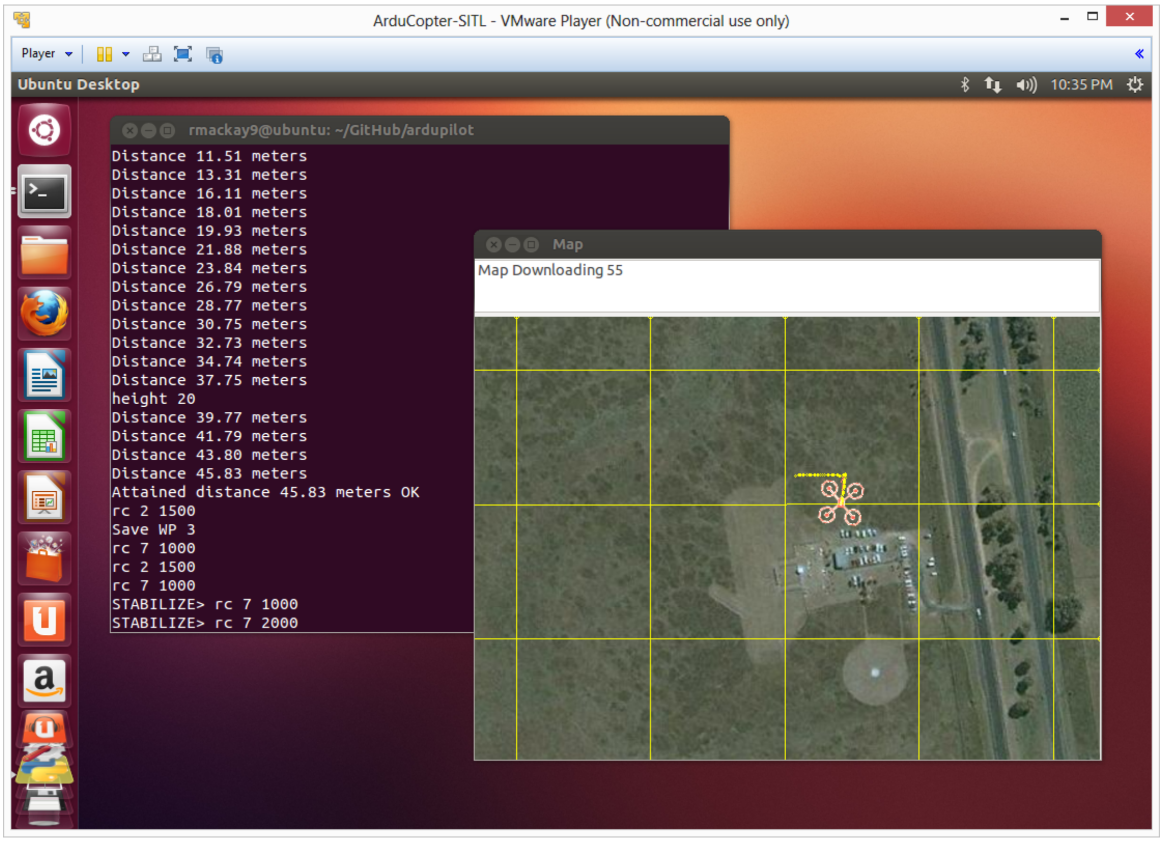

Figure 31: Represantation of someone running SITL

3.13 Testing of drone

Testing of the system will be a complicated process. Due to the potential danger an autonomous drone test can pose to anything within range several tests will be run as simulations before the drone is brought out into the field. At the start of the project, our tests will limited to system tests, ensuring all the components communicate properly with each other. When everything works as planed, simulations of various scenarios will be run in SITL. This allows us to test the autopilot code in every foreseeable situation, while this does not compare to physical tests it will give us ample opportunity to improve the autopilot code.

After this we started conducting physical tests with the drone. Initially tests were focused on ensuring the drone operated as expected and followed commands as expected. For these tests we simply conducted manual test flights of the drone in a large open area, after solving the controller/motor connection problem we mention in Drone Setup. Following the manual tests we will start complete autonomous tests. During these tests we will place the drone 5-8m away from the landing platform while we have a person standing within that same range of the drone with a balloon. This balloon representing a follow target for the drone. The other members of the group will be either, watching outputs and status from the drone on a laptop connected to the drone trough Wi-Fi, or handling the manual controller for the drone, ready to take over if the autopilot starts behaving erratically or encounters some unexpected problems. While the drone is in the air, the target will move relative to the drone, attempting to elicit a reaction. After the searching phase is done the drone will return to its starting position before finding the landing platform and land on it.

During these tests the drone measured its distance to the target and the landing spot when applicable, and time spent completing these various tasks, it then output these values to the control station where we could compare them to physical measurements we made before starting the test or during the test. These comparisons helped us improve the autopilot code and object searching algorithms, as well as simulating various cases with quickly moving targets or targets of varying sizes and distances.

3.13.1 Safety and regulations

During physical tests with this drone safety has been a major concern. The has a weight of around 7kg with the equipment we are using and a potential top speed of up to 70km/h[26]. At that speed the drone can do severe physical damage to anything it collides with. For this reason the drone has always flown autonomously with a speed governor in place limiting the highest reachable velocity of the drone in any absolute direction, except for downwards of course as we can not limit gravity.The speed limit we have implemented is at 2m/s or 7.2km/h. The autopilot is also programed to stabilize and do an emergency landing in case of several unexpected events: Exceptions in the code, loss of connection to the companion computer, base station, or controller, or simply if the drone finds itself in a situation the autopilot is not prepared for.

The drone also has an override for the controller to instantly take direct control. The drone can be placed in two different modes for manual control. Loiter mode, where the drone stabilizes while attempting to hold the same position, the drone can then easily be maneuvered into position for a safe landing. And manual mode, where the controller has full control over the drone with just a degree of auto stabilization to compensate for wind and hard maneuvering. In some situations manual mode is preferred over loiter mode as loiter still falls under a degree of autonomous control and can therefore react unpredictably.

4 RESULTS

4.1 Drone

The DJI s1000+ Drone has been constructed according to plans resulting in a working drone capable of performing as expected. Manual flights with the drone has confirmed this. The drone is controlled by a Pixhawk PX4 Autopilot taking command prompts from an on board Odroid companion computer. The connections from the Pixhawk to the DJI are modified as PX and DJI does not use the same motor setup, details about this can be found in Materials and Methods, Drone setup 3.2.

The drone has a Gimbal connected to a mount front and centre. This gimbal is designed to carry a camera providing good lines of vision for the drone. Unfortunately the Storm 32 gimbal controller encountered some issues late during the project rendering it unusable. This forced us to mount the camera with a hard mount directly on the drone. This forced us to use some improvised solutions for the autopilot as we did not have the ideal lines of sight. Another secondary result of this is the lack of padding between the drone and the camera, this leads to vibrations in the camera distorting the image slightly.

Figure 32: Complete drone

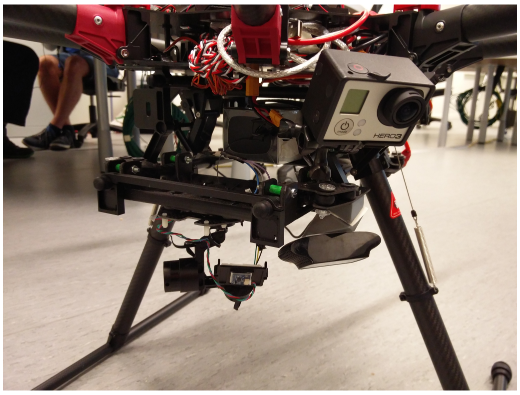

Figure 33: Zoomed in on top of drone

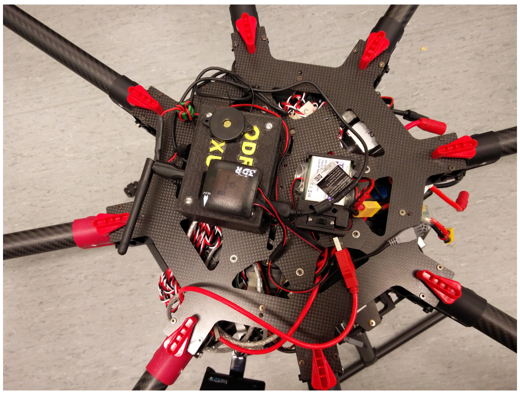

Figure 34: Camera mounting

4.2 Camera field of view

Due to issues with the gimbal, final tests were done with the camera on a stationary mount on the front of the drone. This means the camera is no longer directly under the centre of the drone, but 14cm towards the front. Measurements done for this test have a margin of .5cm.

Vision test 1: Condition: Drone is 25cm above ground.
Blind zone: Drone has a blind zone of 5cm due to angle + 14cm due to camera position.

Field of view: Drone can see items on ground level up to 47cm away from centre, this gives 28cm(47-19) max range.
Vision test 2: Condition: Drone is 94cm above ground.
Blind zone: Drone has a blind zone of 19cm due to angle + 14cm due to camera position.

Field of view: Drone can see items on ground level up to 135cm away from centre, this gives

102cm(135-33) max range.
Vision test 3: Condition: Drone is 8m above ground.
Blind zone: Drone has a blind zone of 162cm due to angle + 14cm due to camera position.

Field of view: Drone can see items on ground level up to 1048cm away from centre, this gives 872cm(1048-176) max range.
Vision test 4: Condition: Drone is 4m above ground.
Blind Zone: Drone has a blind zone of 82cm due to angle + 14cm due to camera position.
Field of View: Drone can see items on ground level up to 532cm away from centre, this gives 436(532-14) max range.

Values in a more easily readable table:

Values	Test 1	Test 2	Test 3	Test 4
Height	25cm	94cm	800cm	400cm
Blind	5+14cm	19+14cm	162+14cm	82+14cm
Line of Sight	28cm	102cm	872cm	436cm

4.3 Autopilot

4.3.1 Autopilot Code

We ended up basing some of our code on/around Randy Mackays balloon finder project [73]. We also had to add a few helper files as well as a main file to manage the 3 modules. This main file creates the necessary processes, initializes logging systems and manages communication between all of the processes. This is also where we have the list of the autopilot modes that we want the drone to run trough (by doing it this way it should be easy for others to use the same framework later).
Figure 35 works as a representation of how the code is built. We would like to mention that most private functions and variables are not represented in this diagram, as they are not needed to understand how the code works. Some connections between the two global objects PositionVector and DroneConfig are also missing, this is due to the fact that most other classes use them. DroneVideo is heavily based upon code by Randy Mackay, while AttitudeHistory, DroneConfig and PositionVector are copied from his code. The rest is mostly written by us, with some influence a few places (commented in code). As the gimbal controller stopped functioning

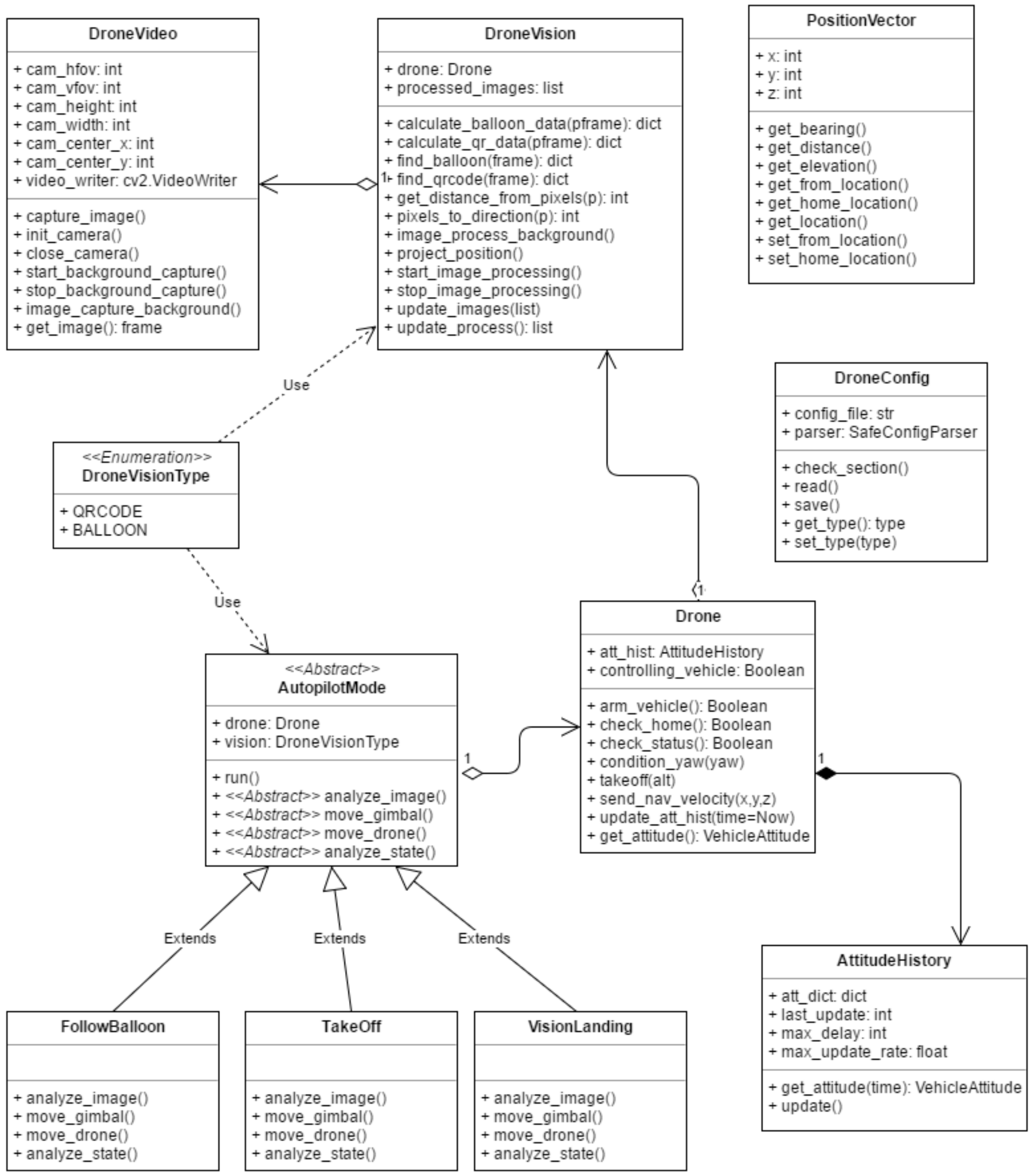

Figure 35: UML Class Diagram

at the end of our project, we modified our code in a way that relied on the drone entering a deadzone when landing, instead of adjusting the pitch of the camera. This is represented by the state diagram shown in figure 36.

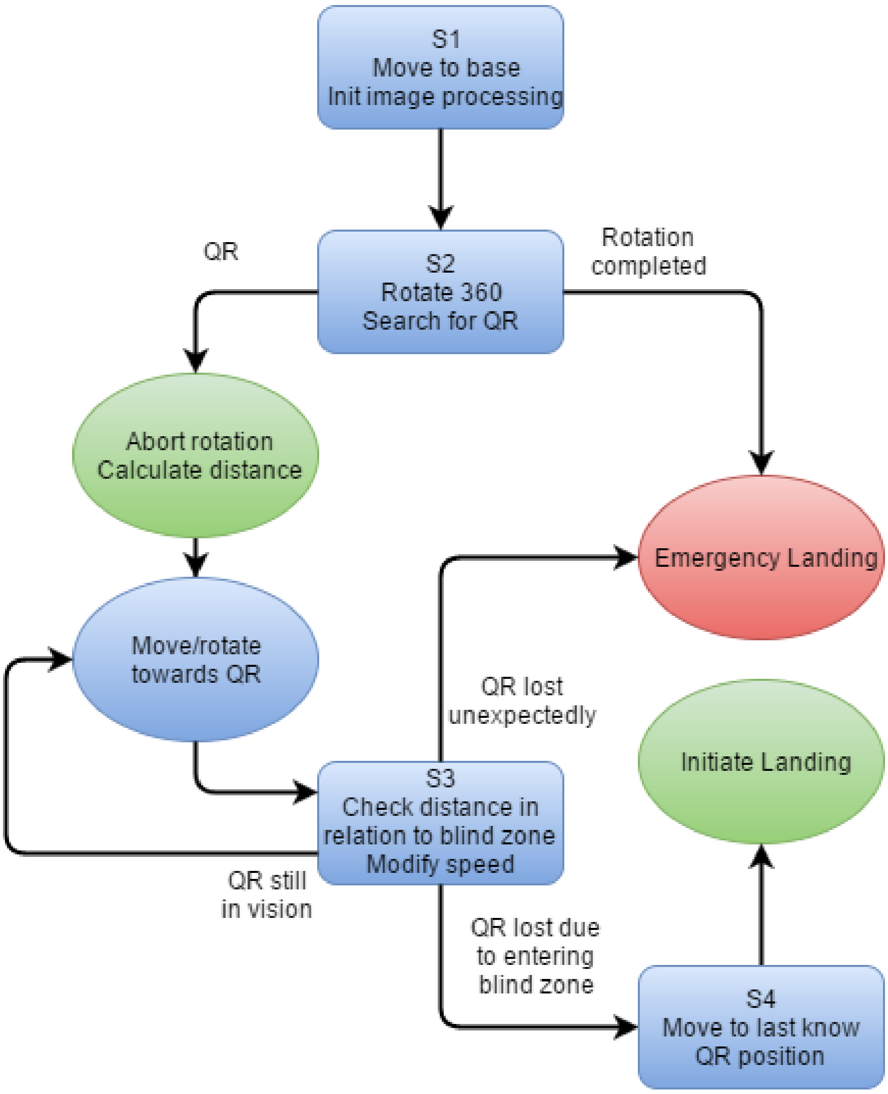

Figure 36: Adjusted Landing Logic

4.4 Test Flights

During the project we did a number of test flights with the drone. In each one we ran different tests, had different problems and different results. We will therefore go trough them one by one and explain what we tested, what problems we had as well as results with a quick explanation of some of the results as they are needed to understand the changes between some of the testflights. At the end we will do a quick summary, with 'end results' from all of the test-flights.

4.4.1 Initial flight tests

We did a few early flight tests without logging them, this was simply to see if the drone worked as intended. We have had issues with the drone not wanting to takeoff evenly at all, meaning that after these tests we still do not have a working drone. The observed problem was that the motors were not spinning evenly, some started before others, and some were not moving at all.

4.4.2 Flight 1: 11.04.2016

What we are testing
We did some problem solving after the previous tests and ended up with changing a few variables before doing another test. There are several variables controlling the power and speed of every motor in relation to the controller, other motors and sensors on the drone. Two of those values were changed, MOT_SPIN_ARMED and THR_MIN. MOT_SPIN_ARMED sets a default spin on the motors as long as they are armed, this is a safety feature to ensure it is clear to the user that the motors are in fact armed. The spin imparted from this variable should not be capable of generating enough thrust to affect the drone. We changed the value from 70 to 250, hoping this would ensure all the motors were active and none tried to stop at any point. THR_MIN controls the minimum throttle sent to the motors to keep them spinning. This ensures the motors does not stop if the controller throttle gets too low, and the motors idle at the same speed. We changed this value from 130 to 300, hoping it would prevent the motors slowing down or stopping independently at low throttles.
Method
Flight controller values were adjusted, and the drone was brought out for careful testing.
Observation
Propellers started spinning immediately, even if the tested had the throttle at zero. The control was acting sluggish and the motors were acting sporadic. Minor throttle engagement caused the drone to flip on its head.
Result Data
No real data, some propellers were broken.

4.4.3 Flight 2: 13.04.2016

What we are testing
After the previously failed test flight we did some more problem solving, and figure out that our flight controllers motor setup is not compatible with that of the drone. We found a workaround

for this in changing the drones two front motors and change motor connections as explained in chapter 1.2 Drone setup. We also adjusted some pid motor values to values tested on our type of drone.

Method

We first reverted the changes from last week, then we did the connection changes needed for the workaround, as well as adjusting some motor pid values on the flight controller. We also did a complete recalibration of the drones accelerometer and gyroscope. Then we went out to test the drone manually.

Observation

Motors no longer engages at 0 throttle. Throttle engagement causes the motors to lift evenly. The drone would probably be able to fly decently like this, but we decided not to try a full takeoff at this date due to weather conditions (heavy wind).

Results

Successfull test, no real data.

4.4.4 Flight 3: 14.04.2016

What we are testing

This test was conducted during good weather conditions to give the drone full lift off, get a feel for its aerial movements and get some preliminary data on battery life time and maneuverability.

Method

Method: Drone was activated and armed as standard. Take off was executed slowly and gradually with a few touchdowns to ensure we had the stability we needed. When the drone was in the air a few easy aerial maneuvers were performed. No extreme maneuvers were completed as we did not have the chance or need to test the drone to its limit.

Observation

The drone moved very smoothly with very good response times and accurate thrust changes. Maneuverability and speed were both within expected margins.

Results

The drone was in the air for a total of slightly more than 4 minutes, this consumed 27% of the battery.

In Figure 37 we have a graph of the output signals provided to the motors from the flight controller during the test flight. As we can see we had a total of 3 liftoffs with very even motor output and for a total of around 4 minutes (x graph is number of lines in the log, not time).

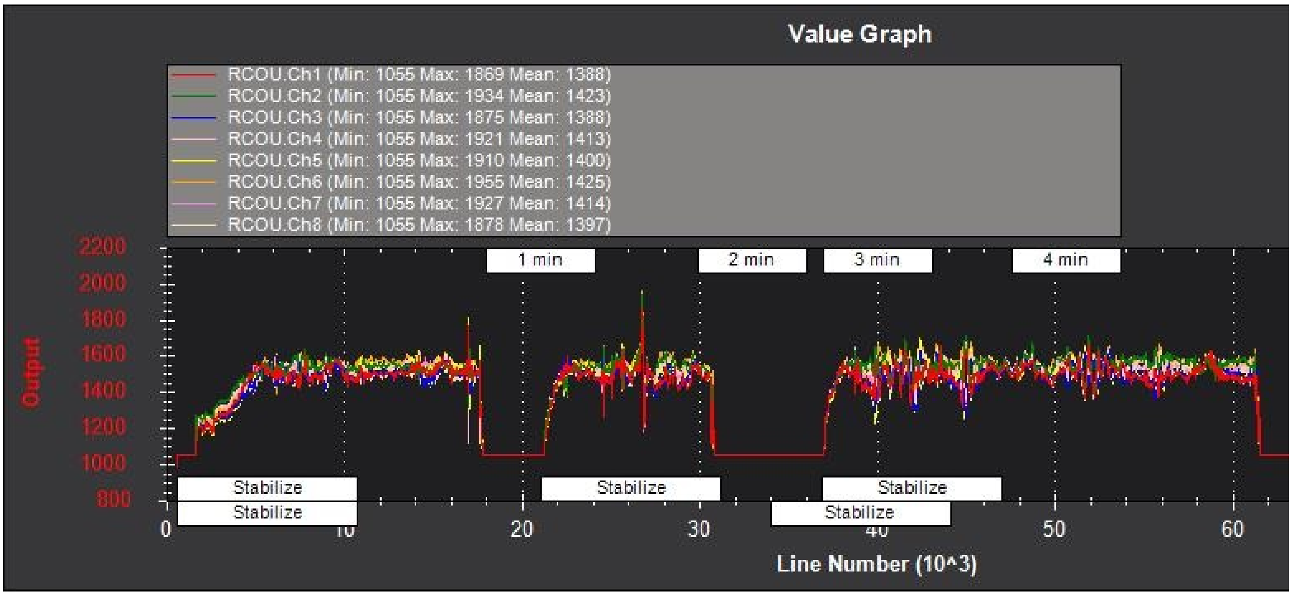

Figure 37: Test flight 3 motor output

4.4.5 Flight 4: 06.05.2016

What we are testing

This was our first attempt to fly the drone with an autopilot. Weather conditions were good. We intended to let the drone lift off autonomously, spin once, then land.

Method

As a security measure we have decided to always test the drone manually before running any type of autopilot. After the manual test we were planning on running a simple script on the odroid.

Observation

When thrust was applied, the drone generated lift unevenly. The flight controller seemed to believe it was standing on uneven ground. Reaction speeds between the controller and the drone was higher than one second, under these conditions it was deemed unsafe to continue the test.

Results

As you can see in Figure 38, the amount of lift provided by the motors were highly varied. These differences seem to all follow the same pattern with the motors providing the same lift relative to thrust at every test. This behaviour would be expected if the drone was standing in a slope.

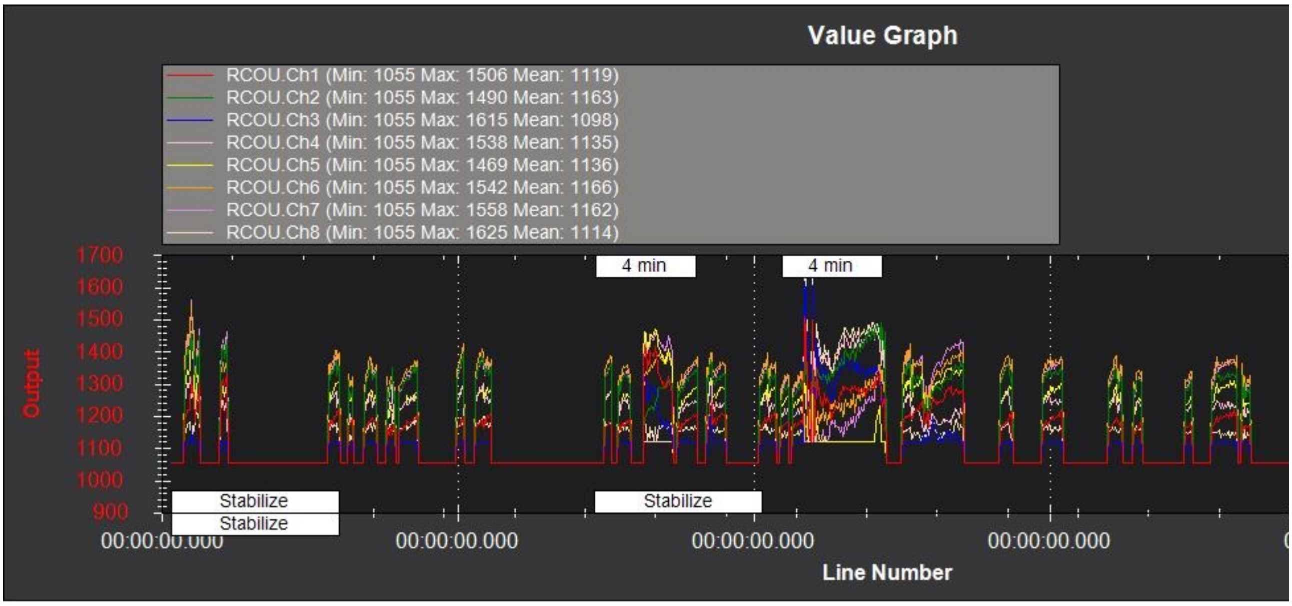

Figure 38: Test flight 4 motor output

4.4.6 Flight 5: 09.05.2016

After the last test we figured out that the drones accelerometer has a tendency to act a bit weird, and requires a recalibration more often than we initially thought.

What we are testing

This is our second autopilot test flight. We want to test communication between the odroid and the flight controller by testing a simple script that takes off, spins in a circle and lands.

Method

We re-calibrated the accelerometer before the test, and as usual we first test the drone manually, these two will be done in every other autopilot test and will not be mentioned in forthcoming logs. We then connect wirelessly to the odroid and run the script.

Observation

Autopilot performed as expected, drone lifted eight meter into the air, before spinning one full rotation then performing a controlled landing.

Results

We did not measure anything specific in this test.

4.4.7 Flight 6: 14.05.2016

We would like to mention that the gimbal controller card stopped working sometime before this test, meaning that we had to make a workaround and put the camera in a fixed position on the drone instead. This means that we had to rework some of the code so that we do not rely on being able to adjust the angle of the camera.

What we are testing

This is our first full test of the autopilot, with all scripts. This means that we want to test the autopilot to make sure that if acts as intended. We also want to log autopilot data, like drone speed, distance to tracked object, if object is found, runtime and other relevant data.

Method

We placed the qr code and the drone 2-3 meters away from each other, and had one person hold a balloon 4-5 meters away from the drone. Then followed the usual procedure and started the autopilot.

Observation

The drone started by taking off to around 8 meters, it rotated around 360° and turned against the one holding the balloon. When rotating we noticed that the drone was not exactly standing in one position in the air, but rather moving around inside an area shaped like a sphere with radius of around 1-2 meters. After rotating, it moved very quickly and positioned itself above the balloon, before it started flying very quickly in circles around the qr code (not a rotation, but actually flying in circles). We took over control of the drone manually and landed it.

Results

It spent 8.3 seconds in the takeoff module (time taken from start to 8m in the air).

Of these 8.3 seconds, 6.4 seconds were spent initializing, 1.9 seconds taking off.

It then spent 1 minute and 3 seconds to rotate 360°, in this time it processed 272 images.

While rotating it kept within a radius of 0.1m (calculated from coordinates, observed seemed to be higher).

The tracked balloon was at one point before it started moving towards the target, calculated to be 10.1 m away from the drone (absolute) and 6.2 in ground distance.

The drones altitude was at the same point measured at 7.97m above its starting point.

The observed distance and altitude at this point seemed to be around the calculated data.

The drone moved towards the balloon with a relative speed of 2m/s (the observed speed seemed to be higher).

The drone lost track of the balloon immediately (observed that the drone was standing above the balloon, it was not supposed to do this).

Afterwards it cancelled its search and started moving towards home coordinates at 2m/s.
No data of qr code, as we did not get to test this code.

Explanation

The autopilots maximum movespeed was set to 2m/s, and this was way to high. There were obviously some miscalculations when calculating the speed as it was not supposed to move more than around 0.5m/s. The circling motion was caused by the drone always trying to move towards the home coordinate in 2m/s and to only stop if it was less than 0.5m away from home.

4.4.8 Flight 7: 15.05.2016

What we are testing

This is a full autopilot test, following up on the results from the previous flight test. The goals of this flight is equal to the last one. We did a few adjustments in the speed of the drone, reducing max speed to 0.5m/s.

Method

We set the drone up the same way we did on flight 6.

Observation

When taking off and rotating to search for the balloon we had the same observation as last test. When following the balloon however, it did not move forward at all. It did however rotate more or less perfectly towards the balloon at all times. After following the balloon for a while, it started rotating again and stopped almost when heading directly towards the qr code. We observed that the rotation time here was much slower than when trying to find the balloon. It then moved a bit forward and landed 0.5m to the right of the qr code.

Results

It spent 7.9 time in the takeoff module (time taken from start to 8m in the air).
Of these 7.9 seconds, 6.1 seconds were spent initializing, 1.8 seconds taking off.
It then spent 59 seconds to rotate 360°, in this time it processed 262 images.
While rotating it kept within a radius 0.05m. (Observed to be between 0.5 and 1m
The tracked balloon was calculated to be 6.3 m away from the drone. (Observed to be around 10-12m)
The drones altitude was calculated to be 7.85 m.
These two values does not make sense, so a ground distance cannot be calculated.
From onboard video we could see that the balloon was staying in the middle of the picture (x direction).
When finding qr code:

Spent 17 seconds rotating 9 degrees.
It processed 16 images in this time.
We calculated the qrcode to be 9.1 m away from the drone, observed to be around 9m.
From onboard video we could see that the qr code was in the middle of the picture when entering deadzone.

Explanation

The balloon distance calculation failed, and in this case the autopilot is programmed to not move. We also found out that the camera was mounted so that it was pointing several (around 10-15) degrees left of the front of the drone, explaining why it landed to the right of the qr code.

4.4.9 Flight 8: 20.05.2016

It should be mentioned that we ran two test on the 2 days preceding this test. These were meant to show our supervisors how the drone worked, and not to gather data. These tests did not go as well as the previous one, as we in the first one was not able to find the balloon, and in the second the odroid decided to crash.

What we are testing

This test is similar to the last test in that we want to gather the same type of data. This is following the corruption of the data on the odroids memory card between these two tests.

Method

Same as on flight 7.

Observation

Similar observations as last test, but this time, when following the balloon, it moved away from it before it started rotating around and search for it again. We did not test qr landing this time due to a few exceptions in the code. It should also be mentioned that we adjusted rotating speeds since last test.

Results

It spent 8.2 time in the takeoff module (time taken from start to 8m in the air).
Of these 8.2 seconds, 6.3 seconds were spent initializing, 1.9 seconds taking off.
It then spent 38 seconds to rotate 360°, in this time it processed 167 images.
While rotating it kept within 0.07m (observed to be between 0.5 and 1m).
The tracked balloon was calculated to be 8.1 m away from the drone. (Observed to be between 10 and 12m)
The drones altitude was calculated to be 7.8 m.

This gives a calculated ground distance of 2.2m
From onboard video we could see that the balloon was staying in the middle of the picture (x direction).
We could see that the balloon disappeared from the video when the drone moved backwards.
Explanation
The exception was due to losing connection with the flight controller. The distance was closer to what was observed this time, and caused the drone to move away from the balloon. The drone adjusts its pitch when moving, causing it to lose track of the balloon.

4.4.10 Flight 9: 24.05.2016

Last test
What we are testing
This is a simple test to ensure everything works and to get some results and video for the presentation.
Method
Same as on flight 8. The balloon we used was destroyed under testing, this means we have do use another balloon of a sub optimal colour.
Observation
The drone acted as expected until the balloon was destroyed. Use of the secondary balloon did not go as well as we hoped as there was a great deal of faulty detection. We believe the autopilot will act as intended given a good target.
Results
Some useful video showing drone flight and drone landing. Unfortunately something happened with the logs files and they ended up being corrupter, we have no other data then the videos and observations.
Explanation
The balloon detection algorithm did not distinguish effectively enough between the balloon and the background with the current active conditions.

4.5 Computer vision

4.5.1 SIFT algorithm

The SIFT algorithm works as intended on detection with a detection fault of approximately one to one-hundred. Detection here works well tested by adding different training images to test how well it behaves on detection of different types objects. The result from this is that the object need to have a bit details to calculate the key-points of it, since our detector detect features and details.

The SIFT algorithm is that is demanding to run and calculations time per detection is long. On windows 10 the algorithm utilizes all the processor cores as it should and have a completion time of 0.18sec on a i5-4200u in the resolution 640 by 480. On the odroid on the other hand that runs linux ubuntu it only utilise one of eight cores of arm processor and takes around one second to complete.

From testing the performance in different resolutions, native 640 by 480 the performance is directly comparable to the resolution of the camera. When resolution is double in number of pixels the processing time doubles. If the number of pixels is divided by 2 the processing goes down to half of its original.

Detection range in different resolutions were also tested. In the native 640 by 480 we got detection range of 13.5m on our large qr code with measures 91cm by 91cm. The detection range is limited by the object getting to small to actually see the qr code in the camera. When changing resolutions the detection range with the same factors as the processing time. Which would say that when we doubled the resolution the detection range is getting doubled as long as the camera gets a clear focus.

4.5.2 Color threshold and detection of blobs

Color threshold and detection of blobs is a method that runs well and have a quite decent performance on the odroid. With a run time of 0.2 seconds per frame. This code utilises all cores on the odroid on linux and all cores on windows.

Color threshold is light sensitive so the performance do vary under different conditions. The color you want to track also has much to say as for example the hue of green and red are near each other because on the hsv scale the color red is both represented at around 0 and around 255 on a scale from 0-255. Detecting other things than actual balloons do happen since this technique use blobs also known as circle like objects. With the color blue the detection we have found the detection works the best with.

Detection range here depends on the size of the balloon. The tracking distance depends on the parameter of smallest balloon allowed. When the distance get to great you cant distinguish a balloon from any other object so there is no point in having the possibility of a too large tracking distance.

5 DISCUSSION

5.1 Computer vision

The computer vision as it stands today it has two main parts one that works well, but is too slow and one that runs well but isn't fail safe and should not be used on a drone.

5.1.1 SIFT algorithm

The sift algorithm works very well with the detections and is a robust as it should be. If the algorithm gets implemented in a bit different way, the performance can be increased and the processing time would go down. Accelerating the image processing so it utilises every core on the odroid and possibly accelerating it with a gpu array, to get a useful framerate. With a useful framerate this can also be used for tracking of moving objects in a environment where the accuracy is important. Since this is a drone that flies by itself with no extra inputs from the observer other than it sensors and the camera. Under testing we also found that the sift algorithm is reliable when tracking other objects than a qr code.

5.1.2 Color threshold and detection of blobs

As known before from reading articles and also the course in image processing by color detection and blob detection would not work well in a non static, outside environment. Since the environment is changing all the time, the lighting condition are changing do to the weather, buildings and other object and the color of the background.
The technique used worked well in some environments and the drone acted accordingly to the inputs from the image processing, while in other conditions or with certain colors, it did not work at all.
The method that is implemented is not robust enough to be used on a drone as it flies in a lot of different conditions, so we would not recommend tracking by this technique on a drone.

5.2 Autopilot

5.2.1 Framework

There are some issues with the framework that we would consider worth working on if someone decides to continue working on the code in the future. First of all there needs to be made

some decisions on whether or not a takeoff and landing module should be required or event a part of the framework at all. There also needs to be made some better general error/exception handling (currently we manually take control of the drone in the case of an error). Some work should also be put in the autopilot modes, as that file is a bit large and complex.
Other then that the framework seems to to what it' supposed to. We can quickly add and remove flight modes if we just want to test one of them (simulations), flight modes are easily made and very little restrictions are added due to the framework, flight modes does not use more cpu than needed, and logging is easily implemented and accessible from almost everywhere.

5.2.2 Balloon Tracking

From results in chapter 4.4, we have that the drone spends a total of over one minute (40 second after speed increase in the middle of the project) rotating 360° when searching for the balloon. In this time frame we processed a lot more images then needed to map the entire area (270 and 160 images, while we have from 3.11.3 that we only need at least 6 processed images to map the area). This means that we should be able to drastically increase the yaw rotation speed, we would have to test this further to check if we are limited by the processing time or the drones capabilities. Currently the drone spends way to much time rotating, especially considering that we are rotating 360° degrees even if we pass what we are searching for, with the possible result of us loosing track of it while finishing the rotation. This could also be fixed by using a more reliable image processing algorithm, so that we could cancel search when we had a hit for an object, instead of having to assume that it is caused by an erroneous detection.

We also had serious issues calculating distance in this autopilot mode, while it worked fine in the QR code detection algorithm even though we used the same type of calculations. We believe that this was caused by us using balloons and a blob detector. When detecting blobs there is no guarantee that we will detect the entire object, and the balloon may or may not be the same size when we use it the day after, causing us to have to recalculate its size (as we use this when calculating distance).

From the tests we have had, we would like to say that it works fine under the correct conditions, but as mentioned in the discussion in chapter 5.1.2, the balloon detection algorithm can be a bit unreliable, so we would have liked to test it out with a more reliable object detection algorithm. As mentioned in the results, there were some issues with the drone moving around a lot when it was supposed to hover in a set position. This, we figured, was due to the inaccuracy of the GPS on the drone. This makes precision work with the drone very hard. A better GPS system (like RTK) would probably solve this problem.

5.2.3 QR Code Landing

The qr code algorithm spends way to much time when rotating and searching, spending up to $17 * (360/9) = 680$ seconds (4.4.8) to rotate 360° is unacceptable, as this is most of a drones battery life and should not be used on the landing procedure. We believe that is is caused by the qr code algorithm taking up to much CPU time, causing increased delay between each drone-movement iteration. More testing should be done on this, as the image processing and the drone movement code should run on different cores (as the image processing one only took up one of the cores in initial tests).

Not being able to track directly below the drone also creates a problem, as we cannot adjust the drones position during the descent. As it is now we do adjustments and then use the drones own landing function, this causes the drone to drift a bit and makes precision landing difficult.

When taking these two as well as the wrongly positioned camera into consideration, we would say that this landing module worked surprisingly well.

5.3 The drone

When testing the drone we found that, the octocopter we used is larger than needed. It is great for later expansion with more heavy equipment than we used. A smaller drone would be more suitable for research, and would increase flight time accordingly, do to its lower power draw. Too have a large drone is great when flying harsh condition, but a smaller drone would be easier for testing.

5.4 Drone tasks

The focus of this thesis is drones in a commercial, maritime setting. The one thing drones can do better than most other platforms, is to provide new lines of vision in complex or disorganised situations. Copter drones are ideal for this as they can easily and quickly get where the camera is needed, and then stay there for a long time. Use of drones to carry cameras also remove the inherent problem a static observer is going to have with LoS blockers or other obstructions.

Autonomously, a drone with a camera can find and track objects, either pre-determined or following certain criteria. This combined with the drones GPS ensure the base station can let the drone follow an object and then know exactly where it is. This is also very useful during search and rescue as the drone can quickly find a lost person, either looking for movement with a normal camera or heat with a thermal camera, the main advantage a drone would be its ability to get a high vantage point as well as the ability to move quickly and cover large areas.

Geographical Surveying:
We believe using drones to survey terrain features and landscape is a good idea. Aerial drones can solve the task in much the same way an airplane or a satellite would traditionally for a fraction of the cost, training and equipment need. Another advantage drones have here is efficiency, while taking off with an airplane to survey ground is a complicated and time taking process, throwing up a drone to do the same thing could be done in a matter of minutes. This also makes it better to use drones in cases where time is limited. Given drones with high endurance this would be a good field of use. We would suggest ships traveling long distances alone and/or looking for specific objects at undetermined locations take use of drones for this purpose. A well equipped drone could take the role of a scout helicopter, helping the ship watch large areas or find a way trough ice drifts and the like.

Search and Rescue:
Search and rescue over large areas are usually done by chopper or small land based vehicles. In difficult terrain, or where such equipment may be unavailable a search is performed manually by foot. In these cases a drone would be much easier to transport to the relevant area and again, a lot cheaper/easier to launch and use. Given the fact that most medium sized commercially available drones are capable of lifting and powering a thermal camera we believe drones could be able to complete the entirety of the search part on its own within areas reaching up to several square kilometres. In some cases drones can even assist with the rescue part of the task as they can transport emergency supplies, first aid kits, life buoys or similar equipment to the person in need of help long before a manned craft or rescue personnel can get to the location. Lastly we think it is worth to mention that these drones can be maintained in locations where people frequently can need help, as on ships far from other help, and then be launched almost instantly in the case of an emergency.

Surveillance:
Drones can bring cameras and other sensors to bear where they are needed, when they are needed. On the other hand, having a drone remain on station to watch a single area for a long time is in most cases unfeasible due to fuel limitations. Any stationary camera can do that just as effectively as a drone in most cases. Using a drone to watch specific things or areas long term would therefore be a relatively expensive undertaking compared to the potential gain. Drones

can be used in situations where the terrain changes drastically on a frequent basis, but then only for as long as the drone can stay on station. In situations where surveillance is needed over several hours or days a single drone is not up to the task, a fleet would have to be used for continuous coverage where a single stationary camera could do the same, this increases the cost of watching the area exponentially.

Supply:
While the main traditional use of drones has been as sensor platforms in an "eye in the sky" capability, with better motors and more effective fuel a drone is capable of lifting surprisingly heavy payloads over long distances. This enables drones to carry equipment, goods, or anything else as long as it fits certain weight and size specifications. All of this means drones can be used to transport things quickly and relatively cheaply. Where earlier a single large transporter would bring deliveries out to several different locations on the same run, a drone can get one relatively small container to exactly where it needs to be a lot quicker than traditional transport. Large drones capable of reaching high speeds while maneuvering with great accuracy could complete this task easily. This would likely come at a greater cost than ordinary deliverance, but in some situations that could be more than worth it or maybe even the only practical solution.

6 CONCLUSION

In this project we have been developing an autopilot for a drone that could take off, track an object and land autonomously. We have also been working on a theoretic part, where we were doing research on the type of drones that exist today and its usages in different industries, with the goal to find potential uses in the maritime industry.

The drone we used was built around a DJI S1000+ frame with an open-source flight controller running the ardupilot software. The drone has a companion computer which is capable for running the scripts for object tracking and sending commands to the flight controller on how the drone should behave. The platform seemed to work well when everything was sorted out between the flight controller and the frame. The frame itself may be a bit to large for the platform used, but gives room for expansions in the future since the drone still have a lot headroom for lifting power, though a smaller one would make it less of a hassle to run test.

Experience from the image processing part of the project, suggests that the SIFT algorithm used for landing has the potential to be used in a larger project, especially if run on faster hardware or if we can find other ways of reducing its needed calculation time.

If drones eventually should be taken into use in a maritime setting it has to be developed with the consideration of the harsh environment that is offshore. The drone we used seems to handle wind condition quite well and the autopilot seems to correct it position accordingly. The big downside with the drone used in this project is that it can't handle humidity and rain. We have found companies that are working on drones that can handle rain and would potentially be usable in such environments.

Another potential issue is the low battery power of most modern drones, but there exists solutions in the form of hydrogen batteries and gas powered drones. Using a fixed wing plane could also be considered in some areas, as these generally use less fuel and can run for a much longer time, with the downside of losing manoeuvrability.

The processing power needed to quickly run image processing algorithms and potential neural networks also pose as a problem, due to the limited space and battery power available on most drones. Processing power on small objects is however a growing field and products such as the NVIDIA Jetson TX1 shows up with the potential to drastically increase the available processing power. Another solution would be to use processing power from an external source, with some sort of communication device.

As for the software, we have built a framework that should be easy to continue building on in the future. We have tested different solutions on how to track objects, and found that reliability is more important than speed.

6.1 Considerations for future work

Future work on this project could follow several different routes. The autopilot could benefit greatly from more work, currently it is capable of finding and following balloons and land on a QR code. While capable, this functionality is ineffective and has no real world uses. There are several ways the efficiency of current functionality can be increased, this would be critical to make the autopilot commercially viable as it is currently way to slow for actual use. On the other hand, the current autopilot is a good base to build further on. Suggested future functionality could be more search scripts with less restrictive prerequisites and thus more potential fields of use. Another important issue in need of resolving would be ensuring the Gimbal is functional and works as expected.

It could also be interesting to test hardware such as the NVIDIA Jetson TX1, sonars for collision detection and RTK GPS for precision work.

7 REFERENCES

References

[1] UAVs: The next big thing?
Accessed at 26.04.16
http://repository.tudelft.nl/islandora/object/uuid%3Abd2a48a7-41f0-4f5a-9936-38a5e9e
Benefits and detriments of commercial and millitary UAVs (30.06.2014)

[2] Microdrones
Accessed at 25.04.16
https://www.microdrones.com/en/applications/areas-of-application/mapping-gis/
Microdrones for mapping and surveying

[3] TheStar.com
Accessed at 25.04.16
http://www.thestar.com/news/insight/2015/01/31/a-drone-boost-for-a-canadian-arctic-i
Drone scouting for ice breakers (31.01.2015)

[4] Project Muse
Accessed at 25.04.16
http://muse.jhu.edu/article/578435
Experimental Flights Using a Small Unmanned Aircraft System for Mapping Emergent Sandbars (Spring 2015)

[5] DroneZon
Accessed at 25.04.16
http://www.dronezon.com/learn-about-drones-quadcopters/introduction-to-uav-photogram
Introduction To UAV Photogrammetry And Lidar Mapping Basics (05.04.2016)

[6] OceanHub
Accessed at 25.04.16
http://innovations.oceanhub.com/maritime-drone-could-become-a-life-saver
Maritime drone could become a life saver! (27.10.2015)

[7] Hellenic Shipping News
Accessed at 25.04.2016

http://www.hellenicshippingnews.com/busan-port-to-utilize-drones-in-crackdown-on-ill
Busan Port to Utilize Drones in Crackdown on Illegally Anchored Vessels (04.08.2015)

[8] Yahoo tech
Accessed at 25.04.2016
https://www.yahoo.com/tech/exclusive-amazon-reveals-details-about-1343951725436982.h
Amazon Reveals Details About Its Crazy Drone Delivery Program (18.01.2016)

[9] Wikipedia
Accessed at 25.04.2016
https://en.wikipedia.org/wiki/Amazon_Prime_Air
Amazon Prime Air (27.02.2016)

[10] The New Yark Times
Accessed at 26.04.2016
http://www.nytimes.com/2012/07/15/sunday-review/the-moral-case-for-drones.html
The Moral Case for Drones (14.07.2012)

[11] e-Tech
Accessed at 25.04.2016
http://iecetech.org/issue/2015-04/Can-drones-replace-satellites
Can drones replace satellites? (April 2015)

[12] SenseFly
Accessed at 25.04.2016
https://www.sensefly.com/applications/agriculture.html
Reap the benefits of precise crop scouting from above

[13] UAV crop dusters
Accessed at 25.04.2016
http://www.uavcropdustersprayers.com
Agriculture UAV Crop Duster Sprayers

[14] The Verge
Accessed at 25.04.2016
http://www.theverge.com/2015/12/15/10220456/intelligent-energy-hydrogen-fuel-cell-dr
Hydrogen fuel cells promise to keep drones flying for hours (15.12.2015)

[15] Gizmag
Accessed at 25.04.2016
http://www.gizmag.com/horizon-energy-systems-hycopter-fuel-cell-drone/37585/
Hydrogen-powered Hycopter quadcopter could fly for 4 hours at a time (19.05.2015)

[16] Digital Trends
Accessed at 25.04.2016
http://www.digitaltrends.com/cool-tech/yeair-gas-powered-drone-kickstarter/
FILL ‘ER UP: This gas-powered drone flies for an hour straight, hits 60mph (26.05.2015)

[17] Alta Devices
Accessed at 28.04.2016
http://www.altadevices.com/applications/unmanned-systems/
Fly All Day: Endurance is Everything.

[18] Newsweek
Accessed at 28.04.2016
http://europe.newsweek.com/google-tests-solar-powered-internet-drones-421561?rm=eu
Google tests solar-powered ’5G’ Internet drones (01.02.2016)

[19] Geek.com
Accessed at 25.04.2016
http://www.geek.com/news/drone-can-fly-for-2-hours-thanks-to-hydrogen-fuel-cell-1650
Drone can fly for 2 hours thanks to hydrogen fuel cell (28.03.2016)

[20] Monash.edu
Accessed at 16.05.2016
http://www.ctie.monash.edu/hargrave/rpav$_{h}ome.htmlBeginnings$
The earliest recorded use of an unmanned aerial vehicle for warfighting (02.02.2003)

[21] Wikipedia
Accessed at 16.05.2016
https://en.wikipedia.org/wiki/History$_{o}f_{u}nmanned_{a}erial_{v}ehiclesWorld_{W}ar_{I}$
The first pilotless aircraft were built during and shortly after World War I

[22] Earlyradiohistory.us Accessed at 16.05.2016
http://earlyradiohistory.us/1963hw40.htm

History of Communications-Electronics in the United States Navy, Captain Linwood S. Howeth, USN (Retired), 1963, pages 479-493:

[23] wierd.com
Accessed at 18.05.2016
http://www.wired.com/2015/04/robot-forces-coming-drone-just-refueled-mid-air/
Our Robot Forces Are Coming: A Drone Just Refueled Mid-Air (29.04.2015)

[24] Quora.com
Accessed at 16.05.2016
https://www.quora.com/What-is-the-maximum-range-of-civilian-drones-UAVs
What is the maximum range of civilian drones/UAV's?

[25] Wikipedia
Accessed at 16.05.2016
https://en.wikipedia.org/wiki/Power-to-weight$_{r}atio$
Power to Weight ratio and potential max output of various types of engine

[26] DJI.com
Accessed at 19.05.2016
http://www.droneexaminer.com/reviews/dji-spreading-wings-s1000-review
Info page, Spread Wings S1000+

[27] Pixhawk.com
Accessed at 19.05.2016
https://pixhawk.org/modules/pixhawk
Pixhawk Autopilot details page

[28] DroneTrest.com
Accessed at 24.05.16
http://www.dronetrest.com/t/beginners-guide-to-drone-autopilots-and-how-they-work/13
Beginners guide to drone autopilots and how they work(02.11.1015)

[29] Drones for Dummies
Accessed at 25.05.2016
http://www.dummies.com/how-to/content/battery-considerations-for-your-drone.html
Battery Considerations for Your Drone

[30] FlyingDrones.expert
Accessed at 25.05.2016
http://flying-drones.expert/drone-flight-time/
Drone Flight Time: What It Is and What It Depends On

[31] https://wiki.paparazziuav.org
Accessed at 25.05.2016
https://wiki.paparazziuav.org/wiki/Main$_{P}age$
Wiki for Paparazzi UAV.

[32] Ardupilot.co.uk
Accessed at 25.05.2016
http://www.ardupilot.co.uk
Ardupilot home page

[33] Northsurveying.com
Accessed at 25.05.2016
http://northsurveying.com/index.php/soporte/gnss-and-geodesy-concepts
GPS, GNSS AND GEODESY CONCEPTS

[34] Wikipedia.com
Accessed at 25.05.2016
https://en.wikipedia.org/wiki/Sonar
Wikipedia page on sonar

[35] Nvidea.com
Accessed at 25.05.2016
http://www.nvidia.com/object/jetson-tx1-module.html Embedded systems

[36] fastcoexist.com
Accessed at 25.05.2016
http://www.fastcoexist.com/3037013/doctors-without-borders-is-experimenting-with-del
Doctors Without Borders Is Experimenting With Delivery Drones To Battle An Epidemic

[37] microdrones.com
Accessed at 25.05.2016
https://www.microdrones.com/en/applications/areas-of-application/monitoring/
SURVEILLANCE MONITORING WITH UAVS

[38] sardrones.com
Accessed at 26.05.2016
http://sardrones.org Home page for SAR drones

[39] researchgate.net
Accessed at 26.05.2016
https://www.researchgate.net/publication/259493156$_{C}omputer_{V}ision_{I}nspired_{R}eal-$
$Time_{A}utonomous_{M}oving_{T}arget_{D}etection_{T}racking_{a}nd_{L}ockingLinktopaperaboutcomputervision$

[40] cs.ubc.ca
Accessed at 26.05.2016
http://www.cs.ubc.ca/ lowe/papers/ijcv04.pdf Link to paper about computer vision

[41] docs.opencv.org
Accessed at 26.05.2016
http://docs.opencv.org/3.1.0/db/df8/tutorial$_{p}y_{m}eanshift.htmlgsc.tab =$
$0Opencvdocumentationmeanshift$

[42] homepage.inf.ed.ac.uk
Accessed at 26.05.2016
http://homepages.inf.ed.ac.uk/rbf/CVonline/LOCAL$_{C}OPIES/AV1011/AV1FeaturefromAcce$

[43] docs.opencv.org
Accessed at 26.05.2016
http://docs.opencv.org/3.0-beta/doc/py$_{t}utorials/py_{f}eature2d/py_{s}ift_{i}ntro/py_{s}ift_{i}ntro.htmlsi$
$introOpencvdocumentationsift$

[44] docs.opencv.org
Accessed at 26.05.2016
http://docs.opencv.org/3.0-beta/doc/py$_{t}utorials/py_{f}eature2d/py_{s}urf_{i}ntro/py_{s}urf_{i}ntro.htmlO$

[45] cs.ubc.ca
Accessed at 26.05.2016
http://www.cs.ubc.ca/research/flann/ Page about flann based matching

[46] visualstudio.com
Accessed at 26.05.2016

https://www.visualstudio.com/products/visual-studio-community-vs Visual studio information

[47] visualstudio.com
Accessed at 26.05.2016
https://cmake.org/ Cmake information

[48] gnu.org
Accessed at 26.05.2016
https://www.gnu.org/software/make/ Make information

[49] opencv.org
Accessed at 26.05.2016
http://opencv.org/ Opencv information

[50] github.com/Itseez
Accessed at 26.05.2016
https://github.com/Itseez/opencv$_contribOpencvcontribinformation$

[51] opencv.org
Accessed at 26.05.2016
http://opencv.org/platforms/cuda.html Cuda information

[52] jetbrains.com
Accessed at 26.05.2016
https://www.jetbrains.com/clion/ Clion information

[53] python.org
Accessed at 26.05.2016
https://www.python.org/doc/essays/blurb/ Python information

[54] python.org
Accessed at 26.05.2016
https://www.techopedia.com/definition/26184/c-programming-language C programming language

[55] whatis.techtarget.com
Accessed at 26.05.2016
http://whatis.techtarget.com/definition/LaTeX Latex information

[56] python.dronekit.io
Accessed at 26.05.2016
http://python.dronekit.io/about/overview.html Dronekit information

[57] Store norske leksikon
Accessed at 26.05.2016
https://snl.no/GPS GPS explanation

[58] Store norske leksikon
Accessed at 26.05.2016
https://snl.no/gyroskop Gyro explanation

[59] Store norske leksikon
Accessed at 26.05.2016
https://snl.no/kompass Gyro explanation

[60] Store norske leksikon
Accessed at 26.05.2016
https://snl.no/akselerometer Accelerometer explanation

[61] Store norske leksikon
Accessed at 26.05.2016
https://snl.no/.search?utf8=%E2%9C%93query=barometer Barometer explanation

[62] Ardupilot.org
Accessed at 26.05.2016
http://ardupilot.org/dev/docs/sitl-simulator-software-in-the-loop.html
About sitl

[63] Futabarc.com
Accessed at 26.05.2016
http://www.futabarc.com/sbus/ about sbus

[64] justanotherlanguage.org
Accessed at 26.05.2016
http://justanotherlanguage.org/book/export/html/162 PWM explanation

[65] freebsd.org

Accessed at 26.05.2016
https://www.freebsd.org/doc/en/articles/serial-uart/ About serial uart

[66] Store norske leksikon
Accessed at 26.05.2016
http://www.rcmodelreviews.com/telemetry.shtml About telemetry

[67] Luftfartstilsynet
Accessed at 26.05.2016
http://luftfartstilsynet.no/selvbetjening/allmennfly/Droner/Om$ $_{D}roner Aboutdronelaws$

[68] Luftfartstilsynet
Accessed at 26.05.2016
http://luftfartstilsynet.no/selvbetjening/allmennfly/Bruk$ $_{a}v$ $_{m}odellflyAboutmodelplane$

[69] Lovdata.no
Accessed at 26.05.2016
https://lovdata.no/dokument/SF/forskrift/2015-11-30-1404 About drone laws

[70] wikipedia.org
Accessed at 03.05.2016
https://en.wikipedia.org/wiki/Multirotor About multirotors

[71] ffden-2.phys.uaf.edu
Accessed at 03.05.2016
http://ffden-2.phys.uaf.edu/webproj/212$_{s}pring_{2}014/Clay_{A}llen/clay_{a}llen/works.htmlAboutmu$

[72] therandomlab.blogspot.no
Accessed at 05.03.2016
http://therandomlab.blogspot.no/2013/03/logitech-c920-and-c910-fields-of-view.html
About logitech c920 fov

[73] https://github.com/rmackay9/
Accessed at 26.05.2016
https://github.com/rmackay9/ardupilot-balloon-finder Link to Randy Mackay's github

APPENDIX

Appendix 1	Forprosjektrapport
Appendix 2	Meeting minutes 1-7
Appendix 3	Fremdriftsrapport
Appendix 4	Regelverk droner
Appendix 5	Source code (in zip)
Appendix 6	Presentation video (drone.mp4)
Appendix 7	Power point presentation

Forprosjekt - Rapport For bacheloroppgave

Tittel:				
Autonom Maritim Drone				

Kandidatnummer(e):			
131121, 120852, 130753			
Dato: May 26, 2016	**Fagkode:** IE303612	**Fagnavn:** Bacheloroppgave	**Dokument tilgang:** Åpen
Studium: Data / Automasjon		**Ant sider/Vedlegg:** 15 / 0	**Bibl. nr:**

Veileder(e):
Ottar L.Osen, Robin T.Bye og Ibrahim Hameed

Sammendrag:

Forprosjektrapporten er gitt som oppgave i faget systemteknikk og systemutvikling som forbereding til bacheloroppgaven. Rapporten gir en innsikt i startfasen til bacheloroppgaven, og inneholder utfordringer fra et tidlig synspunkt i prosjektet. Rapporten inneholder en tidsplan og målsetting med, og skal gi ei oversikt over hvordan arbeidet med bacheloroppgaven vil bli utført.
Bacheloroppgaven "Autonom Maritim Drone" er tildelt av Rolls-Royce. Oppgaven går ut på å finne bruksområder for droner i Maritim Næring. Fokuset for oppgaven i denne omgang er dynamisk posisjonering i forhold til objekt/fartøy i bevegelse og autonom landing på et objekt/fartøy i bevegelse. Vi vil forsøke å finne forskjellige løsninger på dette problemet, både når det gjelder utstyr og programvare, og hvilke eksterne faktorer som kan påvirke dronen.

Postadresse	Besøksadresse	Telefon	Telefax	Bankkonto
Høgskolen i Ålesund	Larsgårdsvegen 2	70 16 12 00	70 16 13 00	7694 05 00636
N-6025 Ålesund	**Internett**	**Epostadresse**		**Foretaksregisteret**
Norway	www.hials.no	postmottak@hials.no		NO 971 572 140

Denne oppgaven er en eksamensbesvarelse utført av studenter ved Høgskolen i Ålesund.

INNHOLD

SAMMENDRAG

Forprosjektrapporten er gitt som oppgave i faget systemteknikk og systemutvikling som forbereding til bacheloroppgaven. Rapporten gir en innsikt i startfasen til bacheloroppgaven, og inneholder utfordringer fra et tidlig synspunkt i prosjektet. Rapporten inneholder en tidsplan og målsetting med, og skal gi ei oversikt over hvordan arbeidet med bacheloroppgaven vil bli utført.

Bacheloroppgaven "Autonom Maritim Drone" er tildelt av Rolls-Royce. Oppgaven går ut på å finne bruksområder for droner i Maritim Næring. Fokuset for oppgaven i denne omgang er dynamisk posisjonering i forhold til objekt/fartøy i bevegelse og autonom landing på et objekt/fartøy i bevegelse. Vi vil forsøke å finne forskjellige løsninger på dette problemet, både når det gjelder utstyr og programvare, og hvilke eksterne faktorer som kan påvirke dronen.

TERMINOLOGI

Begreper

Autonom - Evnen til å ta avgjørelser og operere uten tilbakemelding eller kommander fra en operatør.
Drone - Et ubemannet flygende kjøretøy.
Feature creep - Konstant implementasjon av nye funksjoner utenfor oppgavens rammer.
Version Control System - System som tillater flere brukere å bygge og oppdatere same program samtidig.

Forkortelser

FC - Flight Controller
VCS - Version Control System
GCS - Ground Control Station

1 INNLEDNING

Forprosjektrapporten til bacheloroppgaven er en del av faget ingeniørfaglig systematikk og systemutvikling. Formålet med denne forsprosjektrapporten er å planlegge og definere bacheloroppgaven, både inholdsmessig og formålsmessig. Denne rapporten går ikke inn i detaljer om hvordan vi planlegger å utføre oppgaven, men om forberedelsene vi gjør og omfanget.

Bachelor oppgaven er tildelt av Rolls-Royce. Den går ut på å anskaffe kunnskaper og teste ut bruk av autonome droner i maritim næring.

Droner for kommersiell og militær bruk er en veldig raskt voksende næring. På den sivile siden er det mellom 2015 og 2020 forventet en årlig økonomisk vekst på nesten 20%. Kommersielle droner er forventet å bli brukt i blant annet jordbruk, gruvedrift, byggarbeid og energi-næringer. [1]

Vår oppdragsgiver Rolls-Royce Marine er en verdensledende leverandør av skips design og systemer. Der er over 30.000 kommersielle fartøy (i tillegg til mange militære fartøy) i operasjon rundt om i verden, som bruker utstyr fra Rolls-Royce. [2] Ved NMK i Ålesund ligger Rolls-Royce Training og Technology Centre, som vi har fått oppgaven vår fra. [3]

Som nevnt går oppgaven ut på forskning på mulig bruk av autonome droner i kommersiell maritim drift. Det er en oppgave som andre studentgrupper eventuelt skal kunne fortsette på ved en senere anledning (neste vår). Fokuset for oppgaven i denne omgang er dynamisk posisjonering i forhold til objekt/fartøy i bevegelse og autonom landing på et objekt/fartøy i bevegelse. Vi vil forsøke å finne forskjellige løsninger på dette problemet, både når det gjelder utstyr og programvare, og hvilke eksterne faktorer som kan påvirke dronen.

Det var deltakerenes interesse i droner som førte til at vi søkte (og endte opp med) denne oppgaven. Gruppen ble satt sammen av elever både fra data og automasjon, basert på deltakernes kunnskaper/spesialiseringer inn AI, bildebehandling, mekatronikk og kartsystemer. Oppgaven burde forhåpentligvis utfordre alle deltakerne i gruppen, og gi erfaringer innen et raskt voksende felt.

2 PROSJEKTORGANISASJON

2.1 Prosjektgruppe

Studentnummer(e)
120852
131121
130753

Oppgaver for prosjektgruppen - organisering

Prosjektgruppen er på tre personer, og vil bestå av en prosjektleder, en sekretær og et øvrig medlem. Ansvar for deler av oppgaven blir delt ut avhengig av kunnskap og spesialisering, men alle kommer til å ta del i samtlige oppgaver for å sikre et bredt kunnskapsnivå. Styringsgruppen fra Rolls-Royce avgjør retningen til prosjektet, men avgjørelser om hvordan ting skal gjørest blir tatt felles innad i gruppen.

Oppgaver for prosjektleder

Prosjektleder vil ha det daglige ansvaret for gjennomføringen av prosjektet. Dette innebærer ansvar for at tidsfrister blir holdt, arbeid blir konsentrert i rett retning, holde orden på backlog og sprint, samt sørge for at alle gruppemedlem er kjent med oppgavene sine.

Oppgaver for sekretær

Sekretæren i prosjektet vil utover ansvaret til øvrige medlemmer også ha ansvar over møteorganisering. Dette innebærer å sette opp møter og agenda, skrive referater og dokumentere viktige deler av prosjektet, samt holde orden på all felles dokumentasjon i Confluence.

Oppgaver for øvrige medlem(mer)

Alle gruppemedlemmer vill ha oppgaver direkte knyttet til løsingen av problemet. Naturen til slike oppgaver kommer til å variere utifra hva som er nødvendig til enhver tid. Medlemmer har ansvar for å dokumentere egen kode og fremgangsmetode, dette vil gjøre det lettere for gruppen å samle moduler, samt gjøre det lettere å bygge videre på prosjektet ved en senere anledning.

2.2 Styringsgruppe

Veiledere

Ottar L.Osen, Robin T.Bye og Ibrahim Hameed

Bedriftskontakt Rolls-Royce

Jan Marius Sund og Dag Sverre Grønmyr

3 AVTALER

3.1 Avtale med oppdragsgiver

Avtale gjort med oppdragsgiver angående tidspunkt og innhold for statusmøter. Slike møter vil inneholde statusrapport om oppgaven, nødvendige endringer, og resurser som mangler eller ligger ubrukte.

3.2 Arbeidssted og ressurser

Gruppen vil forsøke å utføre så mye av arbeidet som mulig i datalaben på skolen. Vi vil også potensielt ha tilgang til andre relevante fasiliteter ved NTNU i Ålesund. Det er også potensielt mulig å arbeide hjemmefra i tillegg til avtalte arbeidstider, gruppen burde arbeide sammen så mye som mulig. Skolen har lisenser fra Microsoft som gruppen kan benytte seg av for nødvendig software. Der er også være muligheter for å ha tilgang til utstyr som trengs for fysisk testing av dronen om dette avtalest med skolen. Annet programvare som gruppen trenger vil være tilgjengelig gratis eller ved bruk av en studentlisens. Alle medlemmer i styringsgruppen vil være tilgjengelig for spørsmål om gruppen skulle ha behov for veiledning. Det er også potensiale for hjelp fra annet personell ved NTNU i Ålesund, slik som labmedarbeider.

3.3 Gruppenormer, samarbeidsregler og holdninger

Gruppen arbeider flytende med bidrag fra alle medlemmer. En god standard av arbeidskvalitet er viktig da dette prosjektet kan bli tatt opp av nye grupper etter at vi er ferdig. Vi er enige om gode normer når det gjelder tidsfrister. Alle er forventet å møte presis til avtalte møter. Felles arbeid på skolen er sett på som standard, det er her mesteparten av arbeidet vil gjennomføres. I tillefeller der fravær blir uunngåelig skal slikt meldest fra på forhånd. God arbeidsfordeling er sett på som en sjølfølge. Individuelt arbeid hjemme er ikke negativt, men det er ikke i det miljøet mesteparten av oppgaven skal løsest. Dette oppsettet fører til at hele gruppen har i det minste grunnleggende forståelse for tankegangen og løsningsmetoden som blir brukt til forskjellige deloppgaver. Dette fører også til at kritiske eller tidskrevende avgjørelser kan gjøres felles.

4 PROSJEKTBESKRIVELSE

4.1 Prosjektets mål

Prosjektets hovedmål er som nevnt tidligere å opparbeide kunnskaper om hvordan droner kan brukes i maritim virksomhet. Prosjektet er ment som et mulig 'fortsettelsesprosjekt', altså vil det muligens være andre studenter som ved et senere semester skal fortsette prosjektet i en ny bacheloroppgave.

I vår del a prosjektet skal vi ta for oss muligheten til å bruke en drone til å autonomt følge/overvåke et objekt(skip), samt muligheten til å lande dronen autonomt på objektet. Vi må da se på flere potensielle problemer som kan forekomme, spesielt da bevegelse på objektet som skal observeres/landes på. Bevegelse kan da forekomme i flere former, slik som hiv (fra bølger), rotasjoner og bevegelse fra a til b.

Hensikten er hovedsakelig å anskaffe og dokumentere erfaringer og kunnskap om hvordan vi kan løse de to problemene. Dette vil da gjøres ved å forsøke å bygge en drone og utvikle programvare som kan utføre operasjoner som likner på de to nevnt ovenfor.

Prosessmål

- Arbeid opp mot eksternt spesifiserte mål og restriksjoner.
- Finne nye måter å bruke tilgjengelig teknologi på.

Effektmål

- Autonom drone med følge og lande funksjon.
- Simulator som kan representere det ferdige prosjektet.

Resultatmål

- Rapport som dokumenterer erfaringer gjort i prosjektet.
- Dokumentasjon som tillater enkel gjenopptaking av arbeidet.

4.2 Krav til løsning eller prosjektresultat – spesifikasjon

Prosjektet skal minimum resultere i en avsluttende rapport til arbeidsgiver som går igjennom problemstillingen, fremgangsmåten, erfaringene og resultatene. Prosjektet er hovedsakelig et forskningprosjekt, og det er derfor dokumentert erfaring som er det viktigste. Det vil derfor ikke være et krav at vi ved endt prosjekt har en tilfredsstillende løsning til begge problemstillingene. Dette vil selvfølgelig kreve at vi har gått igjennom og dokumentert årsaken til at prosjektet ikke gikk som planlagt. På denne måten kan erfaringene brukes som grunnlag til et senere prosjekt, uansett resultat.

I og med at prosjektet skal kunne fortsettes på av andre studenter ved en senere anledning, er det også ønskelig at setter opp og dokumenterer på en slik måte at det blir lettere for de som skal ta over.

4.3 Planlagt framgangsmåte(r) for utviklingsarbeidet – metode(r)

SCRUM rammeverk + eXtreme Programming, der vi har adoptert/endret slik at de passer til prosjektet/gruppen.

Vi bruker scrum/XP som utviklingsmetode. Scrum/XP er definert som en iterativ agil utviklingsmetode der mindre deler av oppgaven blir løst for seg og kontakt med kunden blir opprettholdt for kontinuerlig tilbakemelding. Dette gjør at oppgaven blir utført smidig og at ressurser blir brukt der kunden ønsker, når det er nødvendig. Dette gjør at prioriteringer i prosjektet kan endres uten at det har kritisk effekt på fremgangen. Dette åpner også opp for lettere å kunne løse uforutsette problemer eller hindringer som viser seg å være større enn forventet. Fokuset ligger på å produsere fungerende moduler hurtig for så å flette arbeidet sammen til en samlet løsning. Dette kan føre til at prosjektet tar mer tid og ressurser på grunn av "feature creep", noe som kan motvirkest ved å aktivt unngå produksjon av funksjoner som ikke er nødvendige. Extreme programming involverer også sjekk av kode som andre har skrevet, dette gjør koden mer leselig.

En oppgave skal godkjennes av minimum to gruppemedlemmer før den kan bli sett på som ferdig. Testing av oppgaver og delmål skal gjøres hovedsakelig ved bruk av simuleringsprogramvare samt ved bruk av drone der dette er nødvendig eller ved store milpæler. Ved krav om data fra andre moduler, hvor modulene ikke er ferdigstilte eller av andre grunner ikke kan komme med data, skal testdata brukes. Testdataene skal representere data som kommer fra en ferdigstilt modul når dronen er i luften og kjører den relevante koden. Ved testing med bruk av drone skal programmet først kjøre en simulert test av alle relevante oppgaver.

4.4 Informasjonsinnsamling – utført og planlagt

Droner er et veldig populært tema for tiden, og veldig mange forskningsprosjekter og private prosjekter ligger tilgjengelig på internett. Vi ønsker å ta i bruk ArduPilot og Dronekit som grunnlag for koden vår. Dette fordi de er OpenSource og ment spesifikt for prosjekter slik som dette. Der finnes flere samfunn på internett hvor det ligger store mengder informasjon om mange forskjellige prosjekter som vi kan trekke kunnskap fra. Vi ønsker hovedsakelig ikke å arbeide med noe som allerede er gjort, så om vi finner kode som kan gjennbrukes i prosjektet vil vi forsøke å gjøre dette for å spare tid. Bruk av kunnskap fra andres prosjekter vil dokumenteres. Vi vil også søke igjennom forskningsartikler i et forsøk på å finne relevante prosjekter, da det hovedsakelig er privatprosjekter som ligger ute på internett-samfunn for dronebyggere.

4.5 Vurdering – analyse av risiko

Dette er et veldig omfattende prosjekt med tanke på tid og resurser tilgjengelig for gruppen. Lignende oppgaver har blitt gitt på PhD nivå[4]. Hoveddelen av prosjektet skal være fult mulig å realisere innen rammene vi arbeider med, men en fungerende prototype er et veldig optimistisk mål.

Det er mange relevante problemer vi kan støte på, sykdom/skade innad i gruppa vil sannsynlig hvis ha en kritisk effekt på vår evne til å løse oppgaven.

Operasjonelle problemer kan også forekomme dersom vi mister tilgang til lisenser for programvare eller fasiliteter på Høyskolen, men det er liten sannsynlighet for noe av dette da vi bruker OpenSource for de mest kritiske programvarene og vi antar at skolebygget er til full rådighet ut semesteret. Vær og vind kan også sette en stopper for fysisk testing av dronen i realistiske miljøer.

Tekniske problemer vi kan støte på er tap av data eller fysisk skade på utstyr, hovedsaklig dronen. Tap eller korrupsjon av data er lite sannsynlig da vi bruker sky baserte lagringsmetoder med minimal sjanse for feil. Skader på dronen kan bli et stort problem da det er mulig at dette skjer under testing, spesielt med tanke på nyutviklet autopilot. Av den grunn har vi planer om å bruke deler som er i stand til å overleve uheldige møter med underlaget i forskjellige situasjoner.

Finansielle problemer er en bekymring, da vi ikke er garantert tilstrekkelige middeler til å kjøpe en fysisk drone for prototyping. I dette tilfellet kommer vi til å stole på simuleringer for å gi oss de dataene vi trenger, dette vil føre til at resultatet vil være av et betraktelig lavere nivå en det vi håper på.

Tilgjengelig tid kan et problem av flere grunner, lav effektivitet og varierende produktivitet til siden, dette er en omfangsrik oppgave der vi ikke har noen erfaringer om hvor lang tid utvikling kommer til å ta.

4.6 Hovedaktiviteter i videre arbeid

Gruppen har ikke tidligere arbeidet sammen, og vi vet derfor ikke hvor effektivt vi ender opp med å arbeide sammen som en gruppe, samt at vi ikke vet hvor raskt vi arbeider sammenlignet med hverandre. Dette sammen med at vi aldri har arbeidet med et lignende prosjekt gjør det veldig vanskelig for oss å estimere forventet tidsbruk på de forskjellige oppgavene. Tid/Omfang i tabellen nedenfor er derfor ment mer som et mål, og ikke forventet tidsbruk. Det at en av medlemmene på gruppen har ansvaret for en oppgave, betyr ikke at han vil være den eneste som arbeider med den. Den beregnede tiden er i arbeidstimer for en person, slik at vi som en gruppe burde bruke 2d (2 arbeidsdager) på å utføre en oppgave estimert til 6d.

Nr	Hovedaktivitet	Ansvar	Kostnad	Tid/omfang
A1	Bygging av drone	Robin	50.000	4d
A11	Innkjøp	Robin	0	0,5d
A12	Sammensetting	Robin	0	2d
A13	Testing	Inge	0	1,5d
A2	Følging av objekt	Eirik	0	60d
A21	Utvikling av Dronekontroller	Eirik	0	10d
A22	Gimball kontroll modul	Eirik	0	10d
A23	Bildebehandlingsmodul	Inge	0	20d
A24	Drone - kontroll kommunikasjon	Inge	0	5d
A25	Samling av moduler	Eirik	0	5d
A26	Testflyging og bugfiksing	Robin	0	10d
A3	Autonom landing	Robin	0	50d
A31	Landingsmodul	Eirik	0	20d
A32	Videreutvikling av gimballmodul	Eirik	0	10d
A33	Tilpassing av bildebehandlingsmodul	Inge	0	10d
A34	Testing	Robin	0	10d

4.7 Framdriftsplan – styring av prosjektet

Hovedplan

Bygging av drone

Dronen vil bli konstruert så snart gruppen har tilgang til nødvendige deler. Vi vil forsøke å starte på og avslutte denne oppgaven så tidlig som mulig, da det kan ta tid å ordne eventuelle mangler, og andre deler av prosjektet er avhengig av at denne oppgaven er fullført. Senest ønsker vi å få fullført oppgaven før vi starter på oppgave A24, da denne er helt avhengig av at A1 er fullført. Optimalt sett burde A1 være ferdig før avsluttende arbeid på A22 starter, da denne oppgaven aller helst burde testes med en ferdigstil drone. Oppgaven er å anskaffe materialer, samt og bygge og teste dronen som skal brukes under prosjektet.

Følging av objekt

Dette er den største, og forventet mest tidkrevende delen av prosjektet. Vi vil fokusere på å fullføre A2 før vi starter på A3. A2 er allerede startet (A21, A22 og A23) og målet er å være ferdig med A2 ved avslutningen av sprint nummer 3 i begynnelsen av Mars. Oppgaven går ut på å utvikle å teste kode som kreves for å få dronen til å følge etter et objekt.

Autonom Landing

Vi vil i løpet av A2 sette opp mye av det grunnleggende arbeidet som A3 også vil kunne dra nytte av. Det er ikke nødvendig for oss å fullføre A2 før vi starter på A2, men vi ønsker likevel å få gjort dette med mindre noe hindrer videre arbeid med A2 (som at A1 ikke er ferdig når vi bare har A24, A25 og A26 igjen).

Styringshjelpemidler

Gruppen har bestemt seg for å bruke programmene Jira og Confluence til prosjekthåndtering. Begge programmene er utgitt av Atlassian og kjøres som en skytjeneste, og er ment for å kunne brukes sammen.

JIRA er et problem- og prosjekt-oppfølgingsverktøy [5]. Jira gjør det enkelt for gruppen å holde orden på oppgaver som skal løses, oppgaver som er løst og hva som arbeides med av hvem. Verktøyet vil bli satt opp til å fungere med SCRUM, slik at vi kan opprette sprinter og legge til oppgaver i hver sprint basert på oppgaver fra en backlog som vi har opprettet.

Confluence er et verktøy som brukes som en slags wiki for prosjektet. Her kan alle på gruppen opprette, dele og samarbeide på dokumentasjon/informasjon. Det er her vi vil holde all informasjon relatert til prosjektet. Alle filer kan lastes opp, all informasjon relatert til møter kan skrives ned og alt vi krever av dokumentasjon for å fullføre prosjektet kan legges inn.

Utviklingshjelpemidler

Visual Studio
Visual studio er et utviklingsmiljø som er ment til å hjelpe utviklere med å utvikle programmer til (hovedsakelig) windows. Vi vil brukke dette sammen med pluginen 'Python Tools for Visual Studio' til å utvikle python skript som skal kjøres på dronen. Det er ikke nødvendig å bruke Visual Studio, men siden vi får lisens fra NTNU har vi valgt dette.[6]

Git
Git er den VCS'en som vi i gruppen har bestemt oss for å bruke. Sammen med github er git en veldig enkel måte å dele samt holde prosjektfilene oppdaterte. Dette gjør at vi kan arbeide med koden fra flere forskjellige maskiner samtidig.[7]

MavProxy
MavProxy er en uavhengig fungerende GCS for droner. Det kan lett kjøres på de fleste systemer med python og krever veldig lite ressurser fra plattformen. Vi bruker dette til å kjøre simuleringer av flyginger og senere som bakkekontroll for dronen når den selv er i luften.[8]

OpenCV
OpenCV er et bibliotek for computer vision og maskinlæring. Biblioteket har store mengder algoritmer som kan brukes til å trekke informasjon ut av et bilde. Vi bruker OpenCV fordi det er ekstensivt nok for vårt forventet bruk og har en BSD lisens som tillater full bruk og modifikasjon.[9]

SITL (Software in the loop)
SITL er et program som lar oss simulere dronen uten noen form for hardware. Vi bruker SITL fordi det lett kan simulere forskjellige typer droner under varierende forhold. Dette gjør testing av kode og konsept uten bruk av fysisk drone mulig. [10]

ArduPilot
ArduPilot er en serie open-source autopiloter basert på Arduino systemet. ArduPilot har mange varianter som både er kompatible med og spesialisert for forskjellige typer systemer, blant andre Pixhawk som vi ønsker å bruke.[11]

Dronekit - Python DroneKit - Python er et sterkt system for å lage applikasjoner eller skript for dronen. Vi bruker denne versjonen av DK fordi den er spesialisert for Python. DK-P applikasjoner kjører på dronens companion computer og gjennomfører oppgaver som krever mer maskinkraft eller bedre respons en dronens innebygde autopilot klarer. [12]

Intern kontroll – evaluering

En oppgave må testes og godkjennes av minimum to gruppemedlemmer før den kan bli sett på som ferdig. Et delmål vil være ferdig når alle relevante tester er kjørt og godkjent, og all dokumentering er i orden. Gruppen skal møtes hver dag (med mindre annet er avtalt), og hvert medlem skal gå igjennom hva som ble gjort dagen før og hvordan han ligger an på oppgavene sine i forhold til fristen for å få de fullført. Et sprintmøte vil bli hold annenhver uke, her skal alle gruppemedlemmer gå igjennom det de gjorde på sprinten og forklare hva som gikk bra/dårlig og hva som skal gjøres i videre sprinter.

4.8 Beslutninger – beslutningsprosess

Beslutninger i gruppen behandles litt avhengig av påvirkningen av beslutningen. Store beslutninger skal tas opp på statusmøter med veilederne, ved hastesaker skal det sendes mail. Mindre beslutninger, eller beslutninger som bare påvirker gruppemedlemmene vil tatt av gruppen i plenum. Vi vil forsøke å få alle medlemmer i gruppen til å bli enige om en beslutning, om dette ikke er mulig vil vi forhøre oss med veileder eller i verste fall la flertallet bestemme.

5 DOKUMENTASJON

5.1 Rapporter og tekniske dokumenter

Arbeid på dette prosjektet krever flere spesialiserte verktøy og vi har derfor skrevet en guide for installasjon av alle nødvendige program og modifikasjoner vi har brukt. Dette er gjort for å gjøre det lettere for andre grupper å føre arbeidet videre dersom det blir relevant.

Rapporter, dokumenter og notat blir lagret i Confluence for enkel og sikker tilgang. Kode blir oppbevart i et privat github prosjekt som hele gruppen har tillgang til. Den fysiske dronen vil bli oppbevart sikkert på NTNU område eller (midlertidig) hjemme hos en av gruppemedlemmene om dette ikke kan unngås.

Prosjektdokumenter, møtereferat, relevante mailer, arbeidslogger, notater og guider vil alt bli lagret. Dette blir gjort for å gjøre det lettest mulig å videreføre deler av eller hele prosjektet, finne informasjon når vi skal skrive rapport, samtidig som at det gjør alt arbeidet vi har gjort mye mer oversiktlig.

Rutiner vil bli satt opp for å sørge for at all dokumentasjon legges til i Confluence. Dette vil gjøres enten ved å skrive direkte eller å laste opp som vedlegg. Dette fører til at alle arbeidsdokumenter blir organisert øyeblikkelig. Møtereferat vill inneholde deltagelse, tidspunkt, diskusjonspunkt, konklusjoner og avgjørelser som blir tatt på det møtet.

Kode som blir laget skal kommenteres og gjørest lettleselig med tanke på at den kommer til å bli brukt av andre personer etter at vi er ferdig med den. Kode skal også godkjennest av et annet gruppemedlem i forhold til leselighet, effektivitet og implementasjon, igjen for å sikre lett videre utvikling av andre grupper.

Viktige prosjektavgjørelser kommer også til å bli dokumentert med alternativer og begrunnelser. Dette blir gjort for å ha en leselig tankegang der valgene er godt begrunnet for å unngå bortkastet arbeid med store endringer i utførelsen ved senere tidspunkt.

Confluence brukes til distribusjon innad i gruppen. Distribuering av dokument til eksterne aktører blir inntil videre gjennomført manuelt der dette er nødvendig.

Alle dokumenter på Confluence kan vedlikeholdest/modifiserest av alle medlemmer i gruppen om dette blir nødvendig. Dersom store endringer må gjennomførest vill begrunnelse for dette bli lagt inn i relevante dokumenter, nye dokumenter bare for endringer vil gjøre oppstettet uoversiktlig og rotete. Alle avgjørelser som må endrest eller blir irrelevante etter de har blitt tatt vill bli liggende igjen som kommentarer for å opprettholde lesbarheten i arbeids prosessen.

6 PLANLAGTE MØTER OG RAPPORTER

6.1 Møter

Møter med styringsgruppen

Gruppen vil ha statusmøte med styringsgruppen annenhver uke. Vi vil kalle inn alle med sted og tid (ønsker å holde det på samme tidspunkt og samme dag), og med informasjon om hva vi vil ta opp på møtet. Vi vil sende ut møtereferater til alle som sitter i styringsgruppen senest to dager etter møtet, dermed vil også de som eventuelt ikke kan møte også få en oppsummering av hva vi gikk igjennom.

Prosjektmøter

Studentgruppen vil ha to typer møter. Vi vil ha daglige møter hvor vi diskuterer hva hver enkelt person gjorde dagen før og hva som skal gjøres denne dagen. De daglige møtene vil vi forsøke å holde helt i starten av en arbeidsdag og vi ønsker å holde de relativt korte. Vi vil også ha sprint møter, som holdes på slutten/starten av en sprint. Her vil vi diskutere forrige sprint (hva som ble gjort, hvordan det gikk og lignende), og vi vil planlegge neste sprint.

6.2 Periodiske rapporter

Framdriftsrapporter (inkl. milepæl)

Rapportering vil bli gjort i statusmøter, møtereferat vil bli utsendt med status på prosjektet. Når store milepæler er nådd vil det bli sendt ut melding med kort rapport om dette til alle involverte som da kan be om fullstendig rapport på hele den perioden.

7 PLANLAGT AVVIKSBEHANDLING

Om prosjektet ikke går etter planen må studentgruppen gå sammen og analysere problemet, for så å diskutere omfang og eventuelle løsninger. Om det er et stort problem eller vi ikke finner en løsning må de rapporteres til styringsgruppen slik at vi kan finne en løsning sammen. Vi må også forsøke å identifisere om dette er et løselig problem i forhold til rammene eller om vi har støtt på en fundamentell hindring for konseptet og om der er en måte å omarbeide dette på.

Prosjektleder har ansvaret for totaloversikt og at sprinter går som planlagt, hvert enkelt gruppemedlem har ansvar for de oppgavene han har blitt tildelt (og at disse blir utført innen tidsfristen). Prosjektleder kan bruke burndown chart for å passe på at en sprint ligger an til å bli ferdig i tide. Om en person ligger an til ikke å bli ferdig kan han be gruppen om hjelp til dette, da det kan være andre som ligger an til å bli ferdig før planlagt.

8 UTSTYRSBEHOV/FORUTSETNINGER FOR GJENNOMFØRING

Programvare vi anser som kritisk for dette prosjektet er tilgjengelig enten som OpenSource, gratis, eller gratis med bruk av student lisens. Til prosjektadministrasjon bruker vi sky versjonen Jira og Confluence, som er gratis for opp til fem medlemmer. Dette kan skape problemer om styringsgruppen ønsker innsikt, eller ved overføring til andre studentgrupper. Her kan skolen eventuelt tilby en lokal versjon som støtter flere medlemmer, men det er ikke nødvendig for at gruppen skal arbeide optimalt.

Vi vil forsøke å skaffe finansiering til å få kjøpt en drone til å bruke under prosjektet. Å få testet prosjektet med en skikkelig drone vil uten tvil gi de beste resultatene, men om det ikke vil være mulig å anskaffe, vil vi også kunne komme med en mindre optimal løsning ved bruk av simuleringsprogramvare. Vi vil da trenge en drone som er stor nok til å løfte alt vi krever av utstyr, altså en gimbal, kamera, avstandsmåler, 'flight controller' og en 'companion computer'.

9 REFERANSER

[1] Business Insider. THE DRONES REPORT: Market forecasts, regulatory barriers, top vendors, and leading commercial applications [Internett]. [Lastet ned 22 Januar 2016]. Tilgjengelig fra: http://uk.businessinsider.com/uav-or-commercial-drone-market-forecast-2015-2?r=US&IR=T

[2] About Rolls-Royce Marine [Internett]. [Lastet ned 21 Januar 2016]. Tilgjengelig fra: http://www.rolls-royce.com/products-and-services/marine/about-marine.aspx

[3] Leietakere - Norsk Maritimt Kompetansesenter AS [Internett]. [Lastet ned 21 Januar 2016]. Tilgjengelig fra: http://www.nmcc.com/leietakere/

[4] Tickle G.: MIT PhD Student Builds an Autonomous Drone Capable of Avoiding Obstacles While Flying at 30mph [Internett]. Laughing Squid. 2015 [Lastet ned 13 Januar 2016]. Tilgjengelig fra: http://laughingsquid.com/mit-phd-student-builds-an-autonomous-drone-capable-of-avoiding-obstacles-while-flying-at-30mph/

[5] Atlassian.com: JIRA Software - Issue & Project Tracking for Software Teams | Atlassian [Internett]. 2016 [Lastet ned 20 January 2016]. Tilgjengelig fra: https://www.atlassian.com/software/jira

[6] Msdn.microsoft.com. Visual Studio Resources | MSDN [Internett]. 2016 [Lastet ned 15 January 2016]. Tilgjengelig fra: https://msdn.microsoft.com/en-us/vstudio/cc136611.aspx

[7] Git-scm.com. Git [Internett]. 2016 [Lastet ned 15 January 2016]. Tilgjengelig fra: https://git-scm.com

[8] Dronecode.github.io. MAVProxy — MAVProxy 1.4.38 documentation [Internett]. 2016 [Lastet ned 18 January 2016]. Tilgjengelig fra: http://dronecode.github.io/MAVProxy/html/index.html

[9] Opencv.org. OpenCV | OpenCV [Internett]. 2016 [Lastet ned 13 January 2016]. Tilgjengelig fra: http://opencv.org

[10] Dev.ardupilot.com. SITL Simulator (Software in the Loop) | Developer [Internett]. 2016 [Lastet ned 18 January 2016]. Tilgjengelig fra: http://dev.ardupilot.com/wiki/sitl-simulator-software-in-the-loop/

[11] Ardupilot.com. ArduPilot | Open source autopilot [Internett]. 2016 [Lastet ned 13 January 2016]. Tilgjengelig fra: http://ardupilot.com

[12] Python.dronekit.io: Welcome to DroneKit-Python's documentation! [Internett]. [Lastet ned 18 January 2016]. Tilgjengelig fra: http://python.dronekit.io/

Meeting minutes

13.01.16

Attendees:

Ottar L. Osen, Dag Sverre Grønmyr, Jan Marius Grimstad Sund, Robin T. Bye, Ibrahim Hameed, Eirik Fagerhaug, Inge Nilsen, Robin Vattøy.

Agenda:

- Establish the goals of the project
- Decide what physical tasks to focus on.
- Create rough budget, give group an idea of available resources.
- Create a plan for further meetings.

Conclusions:

- The goal of the project has been defined as building a knowledge base.
- Focus on physical part will be to create and program an autonomous drone to search for and land on a target.
- No concrete result on budget, school will provide required resources within limits.
- Plan for meetings every fortnight established.

RR wants a theoretical basis for further development of drones in a commercial maritime setting. The group should also look for potentially useful tech, established or experimental, as well as run theoretical and physical tests when possible.

In addition to this the group could also theorize about scaling(size and quantity), and new problems this could create and potential solutions to those.

This thesis will probably be used as a basis for further degrees at a later date, consider this.

Next meeting:

The next meeting will be held the 21.02.16 at 09:00. Location will be specified when a room can be reserved.

Meeting minutes

10.02.16

Attendees:

Ottar L. Osen, Dag Sverre Grønmyr, Ibrahim Hameed Eirik Fagerhaug, Inge Nilsen, Robin Vattøy

Agenda:

- Drone needs a few parts to be operable.
- Gimbal is needed for all real physical testing.
- Current status of development.

Conclusions:

- Increase focus on concept, less on development.
- No longer assume limiting factors.
- Justify choices made.

Required physical parts will be requisitioned as soon as possible. While waiting for parts find more varied uses for autonomous drones and whether those uses(reformuler) are beneficial or worth the expense.

Focus on direct development of a solution will be dialed down a degree to give more resources to the research part of the project. The group expects a solution to at least one specific problem in the domain, if only to increase our knowledge base with actual physical tests.

Assumptions made early in the process limited our thinking and thus had adverse effects on the results, these assumptions will now be ignored. This change will turn our focus more to a theoretical work space.

Next meeting:

The next meeting will be held the 24.02.16 at 09:00. Location will be specified when a room can be reserved.

Meeting minutes

24.02.16

Attendees:

Jan Marius Grimstad Sund, Robin T. Bye, Eirik Fagerhaug, Inge Nilsen, Robin Vattøy.

Agenda:

- Drone needs specific parts.
- How to control drone in flight.
- Setup of Gantt diagram.

Conclusions:

- NTNU will order required parts, will be delivered within the month.
- Drone control should be modular
- Gantt diagram will be improved.

During this project there will be times when the drone is in need of replacement parts or new parts to solve a different task, these will be provided by NTNU.

Autopilot should be developed with a modular design. This makes further improvements easier and enables another group to continue the work more effectively.

It should also be possible to cancel the drone mission at specific conditions or on command from controller.

Next meeting:

The next meeting will be held the 09.03.16 at 09:00. Location will be specified when a room can be reserved.

Meeting minutes

09.03.16

Attendees:

Ottar L. Osen, Dag Sverre Grønmyr, Eirik Fagerhaug, Inge Nilsen, Robin Vattøy

Agenda:

- Required parts for drone.
- Status and plans for simulations.
- Potential issues with computer vision processing time.

Conclusions:

- Parts for drone has been ordered, are in transit.
- Rudimentary autopilot ready for simple simulations.
- Run tests on computer vision, try to identify bottlenecks.

The frame of the autopilot is ready, simple simulations will be executed with basic commands.

Code for computer vision needs to be made more effective, or another solution must be found, a certain speed is needed here.

Try to create good flight data from simulations to put in thesis.

Next meeting:

The next meeting will be held the 30.03.16 at 09:00. This meeting is delayed one week due to conflicting schedules. Location will be specified when a room can be reserved.

Meeting minutes

30.03.16

Attendees:

Ottar L. Osen, Dag Sverre Grønmyr, Ibrahim Hameed Eirik Fagerhaug, Inge Nilsen, Robin Vattøy

Agenda:

- Status of drone and testing.
- Current computer vision solution.
- What kind of data does employer want.

Conclusions:

- Drone ready for testing by 03.04.16.
- Employer wants hard theoretical data about all operations.

Employer wants concrete values and graphs they can use at a later date.

The group should set up some realistic theoretical scenarios using various known and potential values that could be relevant for our goal. This should give us useful information about drone capabilities and potential areas of use.

Example: graph showing fuel efficiency relative to speed. This would lead to information about range and potential uses in various situations.

Employer wants to know why 0.2 sec is slow for the computer vision part, not just that it is slow.

Run various physical tests, especially one where the drone flies to a specific target among several and lands.

Next meeting:

The next meeting will be held the 21.04.16 at 09:00. This meeting has been pushed back due to easter. Location will be specified when a room can be reserved.

Meeting minutes

21.04.16

Attendees:

Ottar L. Osen, Dag Sverre Grønmyr, Eirik Fagerhaug, Inge Nilsen, Robin Vattøy

Agenda:

- Problem with drone flight characteristics.
- Lightly describe code solution.
- Poster as part of finished product.

Conclusions:

- Comment issue and solution in thesis.
- Poster will be A1, details at a later date.

Problems with flight characteristics identified as compatibility issue between DJI and Pixhawk, we have a potential fix for this and will run tests during the next period.

Code is based on open source work done by outside actors, this is then highly modified to fit our system and goals.

Look for possibilities to access a RTK to increase drone GPS accuracy.

Next meeting:

The next meeting will be held the 04.05.16 at 09:00. This meeting has been pushed back due to easter. Location will be specified when a room can be reserved.

Meeting minutes

04.05.16

Attendees:

Dag Sverre Grønmyr, Ibrahim Hameed Eirik Fagerhaug, Inge Nilsen, Robin Vattøy

Agenda:

- Give advisors access to thesis.
- Computer vision code patent.
- Open source legal requirements.

Conclusions:

- Advisors to be given copies of thesis when requested and at a later date.
- CV code is patented, but open for use in case of students.
- Certain open source libraries require that the entire project code is made available.

Advisors will be given a copy of the first draft when it is done.

There is a patent on parts of the computer vision code we have used, but this is not a problem as students are exempt.

This will force Rolls-Royce to develop their own computer vision solution, but we would recommend they do so anyway to ensure stability and system integrity.

Some of the open source libraries we considered using have stipulations on use requiring us to open the entire project for use by the open source community.

We decided to not use these libraries as there are other options available and the gain is not worth the effort.

Next meeting:

No next meeting has been planned as of yet.

Framdrift 27.01.16 - 10.02.16

Main goal / focus for work done this period.

Gather tools and resources needed to complete the project.

Research about already tested possibilities for use of drones in maritime situations.

Planned activities this period.

Module allowing gimbal to follow objects according to information from the camera.

Module allowing the drone to follow objects based on coordinates sendt from the object (simulation).

Module combining the movement of the drone and the movement of the gimbal in following objects (simulation).

Implement support for range finders (simulation).

Create wiki to contain information and guides.

Completed tasks this period.

All planed modules created and tested successfully in simulation.

Support for range finders implemented.

Wiki created.

Work to gather relevant information started.

Reasons for differences in planned and completed tasks.

No differences in planned and completed tasks.

Changes planned or requested for further project development.

Experiences done during this period.

Work with computer vision will be more demanding than initially expected.

Finding relevant information for the wiki can be problematic as research papers tend to be behind paywalls.

Focus for next period. Complete work with computer vision in relation to following a single specific object. Increase our knowledge base. Identify possibilities for use of different equipment.
Planned activities for next period. Prepare computer vision module for use with the rest of the system. Identify an efficient way to implement drone-gimbal communication and control. If possible modify the drone to use the equipment we have available.
Misc
Requests or needs for further guiding/advice.

Godkjenning/signatur gruppeleder Eirik Fagerhaug	Signatur øvrige gruppedeltakere Robin Vattøy Inge Nilsen

Framdrift 10.02.16 - 24.02.16

Main goal / focus for work done this period.

Define various maritime uses for an autonomous drone.

Develop and compare various solutions for machine vision.

Define general focus for the final solution of the project.

Planned activities this period.

Continue work with drone code.

Document parts and software we will/could use.

Continue work with computer vision (Comparing different solutions)

Completed tasks this period.

Brainstorming about potential drone usage.

Prepared tools and solutions for use towards Linux.

Found various solutions for computer vision and tracking.

Partially redefined goal for the project.

Checked drone, researched and ordered missing parts.

Reasons for differences in planned and completed tasks.

Need to redefine project due to misinterpretation of the goal.

Awaiting delivery of drone parts.

Changes planned or requested for further project development.

We would like to redefine the final goal of the project into one task based on our suggestion and discussions on our meeting this week.

Experiences done during this period.

Setup and use of software is quicker and easier on Linux.

We need to get better at taking notes and documenting meetings as we have tendencies to forget about certain points if we don't. This can cause some misinterpretations.

Focus for next period.

Main focus is to get the drone in the air.

Based on the work done on computer vision the last week we have to end up one one solution.

The usages of drones in maritime areas needs to be transformed from notes into documents.

Planned activities for next period.

Complete test flight of the drone with all equipment.

Make sure software is ready and working on Odroid.

If time we would also like to test the drone with a simple goto command from python code.

Misc

Send a message / research and figure out if landing should be done with ardupilot instead of drone kit as this is a sensitive task.

Requests or needs for further guiding/advice.	
Godkjenning/signatur gruppeleder Eirik Fagerhaug	Signatur øvrige gruppedeltakere Robin Vattøy Inge Nilsen

Framdrift 25.02.16 - 09.03.16

Main goal / focus for work done this period.

Complete construction of drone and perform initial flight test.

Integrated machine vision module into software

Planned activities this period.

Work with framework for drone code and autopilot.

Continue work with building knowledge base.

Continue work with computer vision (Optimize chosen solution)

Assemble drone with parts received.

Completed tasks this period.

Detailed information about potential drone usage collected.

Solution for computer vision operational, but in need of more optimization.

Completed first version of framework for autopilot.

Software tested and ready on Odroid.

Reasons for differences in planned and completed tasks.

Awaiting delivery of drone parts.

Due to lack of physical testing, ability of autopilot framework to complete simple tasks not tested.

Changes planned or requested for further project development.

Experiences done during this period.

Delivery of parts from external sources can lead to unexpected delays.

Focus for next period.

Main focus is to get the drone in the air.

Prepare computer vision solution for use with the autopilot	
Planned activities for next period. Complete test flight of the drone with all equipment. Test autopilot framework code outside of simulator. Create a list of flight modes for drone uses detailed earlier. Develop early versions of these flight modes.	
Misc Send a message / research and figure out if landing should be done with ardupilot instead of drone kit as this is a sensitive task.	
Requests or needs for further guiding/advice.	
Godkjenning/signatur gruppeleder Eirik Fagerhaug	Signatur øvrige gruppedeltakere Robin Vattøy Inge Nilsen

Framdrift 09.03.16 - 30.03.16

Main goal / focus for work done this period.

Main focus is to get the drone in the air.

Prepare computer vision solution for use with the autopilot

Planned activities this period.

Complete test flight of the drone with all equipment.

Test autopilot framework code outside of simulator.

Create a list of flight modes for drone uses detailed earlier.

Develop early versions of these flight modes.

Completed tasks this period.

Secondary equipment

Early versions of flight modes in development.

Early version of computer vision, not fully tested.

Takeoff sequence tested.

Fact finding task continues.

Reasons for differences in planned and completed tasks.

Problems with secondary equipment prevented complete test flight.

Critical work with the physical drone required more time than expected.

Test flight was critical for success of this period.

Changes planned or requested for further project development.

Experiences done during this period.

Ensuring all parts work with the same standard is worth a higher cost.

Focus for next period. Fill all current information into the paper. Get the drone airborne.
Planned activities for next period. Create first draft of paper with current information. Find and implement solution for camera / gimbal. Find a way to read battery status. Complete test flight, manual if not autonomous. Establish in field communication between controller and drone.
Misc
Requests or needs for further guiding/advice.

Godkjenning/signatur gruppeleder Eirik Fagerhaug	Signatur øvrige gruppedeltakere Robin Vattøy Inge Nilsen

Framdrift 31.03.16 - 13.04.16

Main goal / focus for work done this period.

Fill all current information into the thesis.

Get solid information through tests.

Get the drone airborne.

Planned activities this period.

Complete a number of tests for data.

Find and implement solution for camera / gimbal.

Find a way to read battery status.

Complete test flight, manual if not autonomous.

Establish in field communication between controller and drone.

Completed tasks this period.

Communication between controller and drone established.

Found a way to read battery status.

Several test flights completed. Some successful, some not.

Solution for gimbal / camera implemented, new hardware problem detected.

Secondary solution for vision based detection found and prepared.

Reasons for differences in planned and completed tasks.

Motors does not calibrate properly leading to failed take off. This lead us to postpone the meeting as we would potentially have to redo the entire drone. (SOLVED)

Changes planned or requested for further project development.

Define goal for autopilot part of project. (Done in meeting)

Experiences done during this period.

Focus for next period. Complete working code for autopilot.
Planned activities for next period. Implement autopilot. Test autopilot. Prepare the two versions of computervision for use with autopilot.
Misc.
Requests or needs for further guiding/advice. Meeting for this period should have as many members as possible, thus a doodle has been opened. The meeting will be held at first possible opportunity with enough available.

Godkjenning/signatur gruppeleder Eirik Fagerhaug	Signatur øvrige gruppedeltakere Robin Vattøy Inge Nilsen

Framdrift 13.04.16 - 04.05.16

Main goal / focus for work done this period.

Continue work on the thesis.

Complete and test working autopilot code.

Planned activities this period.

Implement autopilot.

Test autopilot.

Prepare the two versions of computer vision for use with autopilot.

Continue work on the thesis

Completed tasks this period.

Autopilot partially implemented.

Autopilot partially tested.(simulation)

Computer vision compatible with autopilot.

Thesis is growing.

Reasons for differences in planned and completed tasks.

Not sufficient time allotted for work on autopilot.

Changes planned or requested for further project development.

Experiences done during this period.

Focus for next period.

Structure and build on the thesis.

Planned activities for next period.

Work further on the thesis.

Complete misc tests with drone.
Misc.
Requests or needs for further guiding/advice.

Godkjenning/signatur gruppeleder Eirik Fagerhaug	Signatur øvrige gruppedeltakere Robin Vattøy Inge Nilsen

Forskrift om luftfartøy som ikke har fører om bord mv.

Dato	FOR-2015-11-30-1404
Departement	Samferdselsdepartementet
Publisert	I 2015 hefte 13
Ikrafttredelse	01.01.2016
Sist endret	FOR-2016-05-20-510
Endrer	
Gjelder for	Norge
Hjemmel	LOV-1993-06-11-101-§15-1, LOV-1993-06-11-101-§4-1, LOV-1993-06-11-101-§5-1, LOV-1993-06-11-101-§5-3, LOV-1993-06-11-101-§7-4§9-1, LOV-1993-06-11-101-§9-5, LOV-1993-06-11-101-§9-7, LOV-1993-06-11-101-§15-1, LOV-1993-06-11-101-§15-4, FOR-1999-12-10-1265, FOR-1999-12-10-1273
Kunngjort	10.12.2015 kl. 15.00
Korttittel	Forskrift om luftfartøy som ikke har fører om bord

Kapitteloversikt:

Hjemmel: Fastsatt av Luftfartstilsynet 30. november 2015 med hjemmel i lov 11. juni 1993 nr. 101 om luftfart (luftfartsloven) § 15-1 første ledd, jf. § 4-1, § 5-1, § 5-3, § 7-4 andre ledd, § 9-1, § 9-5, § 9-7, § 15-4 og § 15-1 andre ledd, jf. delegeringsvedtak 10. desember 1999 nr. 1265 og delegeringsvedtak 10. desember 1999 nr. 1273.

Endringer: Endret ved forskrift 20 mai 2016 nr. 510.

Kapittel 1. Innledende bestemmelser

§ 1. *Formål*

Luftfartsloven fastsetter at alle lovens krav, herunder forskriftskrav, gjelder for luftfartøy som ikke har fører om bord. Formålet med forskriften er å fastsette visse særbestemmelser for luftfartøy som ikke har fører om bord på bakgrunn av den særlige typen luftfart som dette er og enkelte bestemmelser for modellflyging.

§ 2. *Virkeområde*

Forskriften gjelder for all flyging med modellfly eller luftfartøy som ikke har fører om bord i Norge, herunder på Svalbard, samt i luftrommet over norsk kontinentalsokkel og norsk økonomisk sone.

§ 3. *Militær luftfart*

Forskriften gjelder ikke for Forsvarets bruk av luftfartøy uten fører om bord. Forskriften gjelder heller ikke for Forsvarets midlertidige bruk av sivile luftfartøy uten fører om bord til militær luftfart i fare- og restriksjonsområder.

§ 4. *Definisjoner og forkortelser*

I denne forskrift forstås med

a) *luftfartøy som ikke har fører om bord*: innretninger som ikke har fører om bord, som beveger seg i luften og hvor flygingen har et annet formål enn rekreasjon, sport eller konkurranse
b) *modellfly*: innretninger som ikke har fører om bord, som beveger seg i luften og hvor flygingen kun har rekreasjon, sport eller konkurranse som formål
c) *pilot*: den som betjener luftfartøyets styringssystemer og er ansvarlig for føring og sikkerheten under flygingen
d) *fartøysjef:* piloten som er oppnevnt som ansvarlig for å føre luftfartøyet og sikkerheten under flygingen
e) *NOTAM (Notice to Airmen)*: en melding som distribueres ved telekommunikasjon og som inneholder opplysninger om opprettelse, tilstand eller endring av navigasjonshjelpemiddel, tjeneste, prosedyre eller fareforhold som det er viktig i tide å få kjennskap til for personell som har med planlegging og gjennomføring av flyginger å gjøre.

I denne forskrift forstås følgende forkortelser med:

a) *VLOS (Visual Line Of Sight)*: flyging med luftfartøy som ikke har fører om bord som kan gjennomføres slik at luftfartøyet hele tiden kan observeres uten hjelpemidler som kikkert, kamera, eller andre hjelpemidler, unntatt vanlige briller
b) *EVLOS (Extended Visual Line Of Sight)*: flyging med luftfartøy som ikke har fører om bord utenfor pilot eller fartøysjefs synsrekkevidde, der visuell kontroll opprettholdes ved bruk av observatør
c) *BLOS (Beyond visual Line Of Sight)*: flyging med luftfartøy som ikke har fører om bord utenfor synsrekkevidde for pilot, fartøysjef eller observatør
d) *RO*: RPAS-operatør (Remotely Piloted Aircraft Systems)
e) *MTOM*: største tillatte startmasse.

§ 5. ***Tilsyn mv.***

Luftfartstilsynet fører tilsyn med operatører av luftfartøy uten fører om bord.

Luftfartstilsynets inspektører skal gis uhindret adgang til alle aktuelle områder i forbindelse med tilsyn. All relevant dokumentasjon skal gjøres tilgjengelig ved forespørsel.

Kapittel 2. Flyging med modellfly

§ 6. ***Begrensninger for flygingen***

All flyging med modellfly må skje på en hensynsfull måte som ikke utsetter luftfartøy, personer, fugler, dyr eller eiendom for risiko for skade eller for øvrig er til sjenanse for allmennheten.

Modellflyet må til enhver tid være godt synlig for den som fører det slik at full kontroll over modellflyet kan opprettholdes. Flyging med modellfly kan bare skje i dagslysperioden og ikke

a) høyere enn 120 meter over bakken eller vannet
b) nærmere enn 150 meter fra personer, motorkjøretøy eller bygning som ikke er under fartøyførerens kontroll med unntak av under start og landing

Fastsatte begrensninger i andre ledd gjelder ikke for flyging som skjer på en sikker måte i regi av modellflyklubb.

Hvis den som fører modellflyet har bistand fra en person som står ved siden av føreren, og modellflyet til enhver tid er godt synlig for han eller henne, kan flygingen utføres gjennom informasjon fra kamera i modellflyet (first person view).

§ 7. ***Områder hvor det ikke er tillatt å fly modellfly***

Det er ikke tillatt å fly modellfly over eller i nærheten av militære områder, ambassader eller fengsler uten etter tillatelse fra stedlig leder.

Det er ikke tillatt å fly modellfly i restriksjonsområder opprettet med hjemmel i luftfartsloven eller politiloven. Det er ikke tillatt å fly modellfly over eller i nærheten av et sted nødetatene eller Forsvaret har etablert et innsatsområde i forbindelse med ulykke eller annen ekstraordinær hendelse.

Uten tillatelse fra lokal lufttrafikktjenesteenhet er det ikke tillatt å fly modellfly nærmere enn 5 km fra en lufthavn.

§ 8. ***Objektivt ansvar for skade på tredjeperson mv.***

Den som flyr modellfly er uansett skyld ansvarlig for skade eller tap som oppstår utenfor modellflyet som følge av at det benyttes til flyging.

§ 9. ***Alkoholpåvirkning mv.***

Ingen må fly modellfly påvirket av alkohol eller annet berusende eller bedøvende middel. Luftfartsloven § 6-11 og § 6-13 gjelder tilsvarende.

§ 10. ***Våpen mv.***

Det er ikke tillatt å fly modellfly påmontert våpen eller våpensystemer. Det er ikke tillatt å fly modellfly påmontert raketter, fyrverkeri eller annet farlig utstyr.

Kapittel 3. Generelle krav og begrensninger som gjelder for alle RPAS-operatører (RO)

§ 11. ***Krav til ledende personell***

Flyging kan bare skje hvis virksomheten har ansvarlig leder, operativ leder og teknisk leder.

§ 12. ***Transport av gods eller passasjerer***

Transport av passasjerer er ikke tillatt.

Transport av gods er kun tillatt hvis dette er fastsatt særskilt i Luftfartstilsynets tillatelse.

§ 13. ***Krav til høydemåler e.l***.

Piloten og fartøysjefen skal gjennom høydemåler eller andre metoder sikre at luftfartøyet ikke flyr høyere enn 120 meter over bakken eller vannet.

§ 14. ***System for fail-safe***

Alle rotordrevne luftfartøy skal ha et innebygget system som sikrer at luftfartøyet kan lande automatisk ved tap av kontroll fra pilot eller fartøysjef. Alle fly uten fører om bord (fixed wing) skal ha et tilleggssystem som sikrer kontroll over flyet ved tap av kontakt med hovedradio.

§ 15. ***Oppvisningsflyging***

Oppvisningsflyging er kun tillatt hvis det foreligger tillatelse fra Luftfartstilsynet.

§ 16. ***Lufthavner***

Landing og avgang med luftfartøy som ikke har fører om bord kan ikke skje på lufthavner.

I særlige tilfeller kan landing eller avgang fra lufthavn avtales med lokal lufttrafikktjenesteenhet, forutsatt at det er etablert lokale prosedyrer som ivaretar sikkerheten for øvrig lufttrafikk. Hensynet til avvikling av øvrig lufttrafikk skal ha prioritet. Lufttrafikktjenesten er ansvarlig for å etablere tilstrekkelige sikkerhetsavstander.

§ 17. ***Objektivt ansvar for skade på tredjeperson mv.***

Operatøren er uansett skyld ansvarlig for skade eller tap som oppstår utenfor luftfartøyet som følge av at fartøyet brukes til luftfart. Dette gjelder likevel ikke skade på annet luftfartøy, eller på person eller ting i slikt luftfartøy.

§ 18. *Forsikring*

Operatøren er ansvarlig for at det foreligger forsikring som dekker erstatningsplikt ovenfor tredjeperson, jf. luftfartsloven § 11-2.

§ 19. *Alkoholpåvirkning mv.*

Ingen må fly luftfartøy som ikke har fører om bord påvirket av alkohol eller annet berusende eller bedøvende middel. Luftfartsloven § 6-11 og § 6-13 gjelder tilsvarende.

§ 20. *Våpen mv.*

Det er ikke tillatt å fly luftfartøy som ikke har fører om bord påmontert våpen eller våpensystemer. Det er ikke tillatt å fly luftfartøy som ikke har fører om bord påmontert raketter, fyrverkeri eller annet farlig utstyr uten tillatelse.

§ 21. *Meldeplikt om opphør*

Operatøren skal varsle Luftfartstilsynet dersom virksomheten opphører.

Kapittel 4. Krav til virksomheter som opererer innenfor kategori RO 1

§ 22. *RO 1*

Operatør av type RO 1 må sende melding til Luftfartstilsynet om oppstart av ny virksomhet før virksomheten tar til. Meldingen skal inneholde opplysninger om virksomhetens navn, adresse og kontaktinformasjon samt informasjon om type luftfartøy som skal benyttes.

RO 1 er virksomhet hvor luftfartøyet

a) har en MTOM opp til 2,5 kg og
b) har maksimal hastighet 60 knop

som skal operere utelukkende innenfor VLOS i dagslysperioden og innenfor fastsatte sikkerhetsavstander, jf. § 51.

§ 23. *Krav til organisasjon*

Operatøren skal ha ansvarlig leder, operativ leder og teknisk leder. Flere funksjoner kan ivaretas av samme person.

Ansvarlig leder har det overordnede ansvaret for virksomheten. Ansvarlig leder må kunne godtgjøre at organisasjonen er tilpasset virksomhetens størrelse og kompleksitet. Ansvarlig leder må være fylt 16 år.

Operativ leder skal sikre at operasjoner utføres i henhold til virksomhetens operasjonsmanual.

Teknisk leder skal sikre at virksomhetens luftfartøy er luftdyktige.

Hvis virksomheten er særlig kompleks må operatøren etablere og vedlikeholde et kvalitetssystem tilpasset driften. Kvalitetssjef skal sikre at virksomhetens systemer for kvalitetssikring er ivaretatt.

§ 24. *Operasjonsmanual*

Operatøren skal ha en operasjonsmanual tilpasset kompleksiteten i virksomhetens operasjoner. Operasjonsmanualen skal minst inneholde

a) beskrivelse av virksomhetens oppbygging
b) beskrivelse av de operasjonstyper som inngår i virksomheten
c) beskrivelse av vedlikeholdsprosedyrer
d) oversikt over alle luftfartøy som inngår i virksomheten.

§ 25. *Loggføring*

Det skal føres en logg over flygetid. Loggen skal minimum inneholde informasjon om hvilket luftfartøy som ble benyttet, hvem som utførte flygingen, tidspunkt for flygingen og området flygingen fant sted.

§ 26. *Vedlikehold*

Operatøren skal påse at luftfartøy vedlikeholdes i henhold til produsentens anvisninger.

§ 27. *Merking av luftfartøy*

Alle luftfartøy skal være tydelig merket med operatørens navn og telefonnummer.

§ 28. *Krav til pilot*

Pilot må kunne demonstrere tilstrekkelige ferdigheter til at flyging kan skje sikkert og i tråd med regelverket.

Kapittel 5. Krav til virksomheter som opererer innenfor kategori RO 2

§ 29. *RO 2*

Operatør av type RO 2 må ha tillatelse fra Luftfartstilsynet før virksomheten kan ta til. Søknaden må inneholde en risikoanalyse og operasjonsmanual.

RO 2 er virksomhet hvor luftfartøyet

a) har en MTOM opp til 25 kg og
b) har maksimal hastighet 80 knop

som skal operere VLOS eller EVLOS og innenfor fastsatte sikkerhetsavstander, jf. § 51, eller BLOS i samsvar med § 56 – § 59.

§ 30. *Krav til organisasjon*

Operatøren skal ha ansvarlig leder, operativ leder, teknisk leder og kvalitetssjef. Flere funksjoner kan ivaretas av samme person. Endringer i organisasjonen skal meldes til Luftfartstilsynet.

Ansvarlig leder har det overordnede ansvaret for virksomheten. Ansvarlig leder må kunne godtgjøre at organisasjonen er tilpasset virksomhetens størrelse og kompleksitet. Ansvarlig leder må være fylt 18 år.

Operativ leder skal sikre at operasjoner utføres i henhold til virksomhetens operasjonsmanual. Operativ leder må oppfylle kravet som fastsatt i § 36 første ledd.

Teknisk leder skal sikre at virksomhetens luftfartøy er luftdyktige. Teknisk leder må kunne godtgjøre nødvendig kompetanse innenfor områdene flyteknikk, elektronikk og aerodynamikk.

Kvalitetssjef skal sikre at virksomhetens systemer for kvalitetssikring er ivaretatt.

§ 31. *Kvalitetssystem*

Operatøren skal etablere og vedlikeholde et kvalitetssystem tilpasset driften.

§ 32. *Operasjonsmanual*

Operatøren skal ha en operasjonsmanual tilpasset kompleksiteten i virksomhetens operasjoner. Operasjonsmanualen skal minst inneholde

a) beskrivelse av virksomhetens oppbygging
b) beskrivelse av de operasjonstyper som inngår i virksomheten
c) prosedyrer for de operasjoner som skal utføres, inkludert risikoanalyser
d) beskrivelse av krav til kompetanse for vedlikeholdspersonell
e) beskrivelse av krav til kompetanse og vedlikeholdstrening for fartøysjef
f) vedlikeholdsprogrammer
g) oversikt over alle luftfartøy som inngår i virksomheten.

§ 33. *Luftdyktighet*

Operatøren kan bare benytte luftfartøy som det kan dokumenteres er luftdyktig. Det må kunne fremlegges dokumentasjon for fartøyets konstruksjon, styresystem og øvrige system samt for praktisk vedlikehold. Det må også kunne dokumenteres at systemet er testet for de operasjoner som fartøyet er planlagt brukt for.

Det skal etableres et vedlikeholdsprogram for luftfartøyene. Det skal fremgå av vedlikeholdsprogrammet når komponenter skal skiftes ut. Vedlikeholdsprogrammet må inneholde en prosedyre for oppdatering av programmet.

Teknisk flygetid for luftfartøy og kritiske komponenter skal loggføres.

§ 34. *Vedlikehold*

Vedlikehold skal utføres i henhold til luftfartøyets eller systemets vedlikeholdsprogram.

Vedlikehold skal utføres av teknisk personell godkjent av teknisk leder. Vedlikeholdspersonellets kompetanse til å drive vedlikehold på aktuelt luftfartøy eller system skal være dokumentert.

§ 35. ***Merking av luftfartøy***

Alle luftfartøy skal være tydelig merket med identifikasjonsnummer som tildeles av Luftfartstilsynet.

§ 36. ***Krav til pilot og fartøysjef***

For å kunne utføre flyging må pilot eller fartøysjef ha bestått e-eksamen.

Luftfartstilsynet har ansvar for å utarbeide materiell for opplæring og for å gjennomføre e-eksamen.

Pilot og fartøysjef må kunne demonstrere tilstrekkelige ferdigheter til at flyging kan skje sikkert og i tråd med regelverket. Ferdighetene må holdes oppdatert gjennom vedlikeholdstrening.

Kapittel 6. Krav til virksomheter som opererer innenfor kategori RO 3

§ 37. ***RO 3***

Operatør av type RO 3 må ha tillatelse fra Luftfartstilsynet før virksomheten tar til. Søknaden må inneholde en risikoanalyse og operasjonsmanual.

RO 3 er virksomhet hvor luftfartøyet

a) har en MTOM på 25 kg eller mer, eller
b) har maksimal hastighet over 80 knop eller
c) drives av turbinmotor, eller
d) skal operere BLOS høyere enn 120 meter, eller
e) skal operere i kontrollert luftrom høyere enn 120 meter, eller
f) skal operere over eller i nærhet av folkeansamlinger i andre tilfeller enn det som følger av § 51 tredje ledd.

§ 38. ***Krav til organisasjon***

Operatøren skal ha ansvarlig leder, operativ leder, teknisk leder og kvalitetssjef. Flere funksjoner kan ivaretas av samme person. Endringer i organisasjonen skal meldes til Luftfartstilsynet.

Ansvarlig leder har det overordnede ansvaret for virksomheten. Ansvarlig leder må kunne godtgjøre at organisasjonen er tilpasset virksomhetens størrelse og kompleksitet. Ansvarlig leder må være fylt 18 år.

Operativ leder skal sikre at operasjoner utføres i henhold til virksomhetens operasjonsmanual. Operativ leder må oppfylle kravet som fastsatt i § 46 første ledd.

Teknisk leder skal sikre at virksomhetens luftfartøy er luftdyktige. Teknisk leder må kunne dokumentere relevant teknisk kompetanse for de aktuelle system som virksomheten opererer.

Kvalitetssjef skal sikre at virksomhetens systemer for kvalitetssikring er ivaretatt.

§ 39. ***Kvalitetssystem***

Operatøren skal etablere og vedlikeholde et kvalitetssystem tilpasset driften.

§ 40. ***Operasjonsmanual***

Operatøren skal utarbeide en operasjonsmanual tilpasset kompleksiteten i virksomhetens operasjoner. Operasjonsmanualen skal minst inneholde

a) beskrivelse av virksomhetens oppbygging
b) beskrivelse av de operasjonstyper som inngår i virksomheten
c) prosedyrer for de operasjoner som skal utføres, inkludert risikoanalyser
d) beskrivelse av krav til kompetanse for vedlikeholdspersonell
e) beskrivelse av krav til kompetanse og vedlikeholdstrening for pilot og fartøysjef
f) vedlikeholdsprogrammer
g) oversikt over alle luftfartøy som inngår i virksomheten.

§ 41. ***Luftdyktighet***

Operatør kan kun benytte luftfartøy eller system godkjent av Luftfartstilsynet for den aktuelle operasjonstypen. Operatøren skal dokumentere at luftfartøy, system og komponenter er tilstrekkelig sikre for bruk for den aktuelle operasjonstypen. Ved vurderingen skal forskrift 4. mars 2013 nr. 252 om luftdyktighets- og miljøsertifisering for luftfartøyer mv. og sertifisering av design- og produksjonsorganisasjoner (sertifiseringsforskriften) legges til grunn så langt den passer.

I søknaden må det vedlegges dokumentasjon for systemets konstruksjon, styresystem, komponentenes art, tekniske sikkerhetssystemer og gjennomført testprogram som viser at luftfartøyet og systemet kan utføre den aktuelle operasjonstypen.

Luftfartstilsynet kan anerkjenne luftfartøy, system eller komponenter som er godkjent eller sertifisert av andre luftfartsmyndigheter.

Det skal etableres et vedlikeholdsprogram for luftfartøyet eller systemet. Det skal fremgå av vedlikeholdsprogrammet når komponenter skal skiftes ut. Vedlikeholdsprogrammet må inneholde en prosedyre for oppdatering av programmet.

Teknisk flygetid for luftfartøy og kritiske komponenter skal loggføres.

§ 42. ***Ulike operasjonstyper***

Luftfartøy eller system som utelukkende skal benyttes til operasjoner som kan utføres av RO 1- og RO 2-operatører, jf. § 22 og § 29, jf. § 51, trenger ikke godkjenning etter § 41. For disse luftfartøyene gjelder § 26, § 27, § 33 og § 34 tilsvarende.

§ 43. ***Nærmere om testprogram***

Operatør kan ikke igangsette testprogram for et luftfartøy eller system før testprogrammet er godkjent av Luftfartstilsynet. Søknad om godkjenning skal inneholde en beskrivelse av systemtesten, blant annet hvilken operasjonstype som systemet skal testes for, hvor testprogrammet skal gjennomføres, sikkerhetsdokumentasjon for gjennomføring av testprogrammet og sjekkliste for vitale testpunkter.

§ 44. ***Vedlikehold***

Vedlikehold skal utføres i henhold til luftfartøyets eller systemets vedlikeholdsprogram.

Vedlikehold skal utføres av teknisk personell godkjent av teknisk leder. Vedlikeholdspersonellets kompetanse til å drive vedlikehold på aktuelt luftfartøy eller system skal være dokumentert.

§ 45. ***Merking av luftfartøy***

Alle luftfartøy skal være tydelig merket med identifikasjonsnummer som tildeles av Luftfartstilsynet.

§ 46. ***Krav til pilot og fartøysjef***

For å kunne utføre flyging må pilot eller fartøysjef ha bestått e-eksamen. Luftfartstilsynet har ansvar for å utarbeide materiell for opplæring og for å gjennomføre e-eksamen.

Pilot og fartøysjef må kunne demonstrere tilstrekkelige ferdigheter til at flyging kan skje sikkert og i tråd med regelverket.

For å kunne foreta landing eller avgang på lufthavner, må pilot og fartøysjef inneha LAPL, PPL, CPL eller ATPL.

Pilot og fartøysjefen må ha tilstrekkelig kunnskap til å kunne kommunisere med lufttrafikktjenesten ved bruk av gjeldende radiotelefoniprosedyrer. Pilot og fartøysjef som skal benytte radio må inneha flytelefonistsertifikat.

Kapittel 7. Operative bestemmelser som gjelder for alle RO-operatører

§ 47. ***Lufttrafikkregler***

De alminnelige lufttrafikkregler gjelder for luftfartøy som ikke har fører om bord.

§ 48. ***Luftrom***

Pilot og fartøysjef plikter å gjøre seg kjent med gjeldende luftromsorganisering. Pilot og fartøysjef plikter videre å gjøre seg kjent med aktuell luftromsklassifisering og ansvarlig lufttrafikktjenesteenhet for det området hvor en operasjon planlegges utført.

§ 49. ***Vikeplikt for andre luftfartøy***

Luftfartøy som ikke har fører om bord skal vike for andre luftfartøy.

§ 50. ***Forberedelser før flyging***

Pilot og fartøysjef skal før hver flyging gjøre seg kjent med alle tilgjengelige opplysninger av betydning for den planlagte flygingen, herunder om værforholdene.

Pilot og fartøysjef skal forsikre seg om at fartøyet er luftdyktig før flyging finner sted. Enhver flyging skal gjennomføres i henhold til gjeldende bestemmelser, operasjonsmanualen og fartøyets begrensninger.

§ 51. *Sikkerhetsavstander, maksimal flygehøyde*

All flyging må skje på en hensynsfull måte som ikke utsetter luftfartøy, personer, fugler, dyr eller eiendom for risiko for skade eller for øvrig er til sjenanse for allmennheten.

Luftfartøyet må til enhver tid være godt synlig for den som fører det. Ved enhver flyging skal det holdes nødvendige sikkerhetsavstander. Det er ikke tillatt å fly

a) høyere enn 120 meter over bakken eller vannet
b) nærmere enn 150 meter fra folkeansamling på mer enn 100 personer
c) nærmere enn 50 meter fra personer, motorkjøretøy eller bygning som ikke er under pilotens og fartøysjefens kontroll.

Luftfartøy som har en MTOM på 250 gram eller mindre, kan flys VLOS, EVLOS eller BLOS, men ikke høyere enn 50 meter over bakken eller vannet. Sikkerhetsavstandene i andre ledd bokstav b og c gjelder ikke.

Flyging ut over det som følger av sikkerhetsavstandene i andre og tredje ledd, kan bare utføres av RO 3-operatør i tråd med bestemmelsene i kapittel 9 og for øvrig de vilkår som er gitt i tillatelsen.

§ 52. *FPV*

Flyging FPV (first person view) uten BLOS-godkjenning, jf. § 57 og § 64, er kun tillatt så fremt flygingen skjer VLOS og fartøysjefen til enhver tid har visuell kontakt med fartøyet.

§ 53. *EVLOS*

Flyging EVLOS er kun tillatt hvis tillatelsen fra Luftfartstilsynet omfatter denne operasjonstypen.

Ved EVLOS-flyging skal det etableres toveis radiokommunikasjon eller kontinuerlig telefonkommunikasjon mellom pilot og observatør.

§ 54. *Områder hvor det ikke er tillatt å fly*

Det er ikke tillatt å fly luftfartøy som ikke har fører om bord over eller i nærheten av militære områder, ambassader eller fengsler.

Det er ikke tillatt å fly luftfartøy som ikke har fører om bord nærmere enn 5 km fra en lufthavn, hvis ikke flygingen er avklart med lokal flygekontrolltjeneste eller flygeinformasjonstjeneste.

§ 55. *Flyging når det har skjedd ekstraordinære hendelser*

Flyging over eller i nærheten av et sted nødetatene eller Forsvaret har etablert et innsatsområde i forbindelse med ulykke eller annen ekstraordinær hendelse, er kun tillatt med tillatelse fra innsatsleder.

Kapittel 8. Operative tilleggsbestemmelser som gjelder for RO 2-operatører

§ 56. *BLOS*

Flyging BLOS er kun tillatt hvis tillatelsen fra Luftfartstilsynet omfatter denne operasjonstypen.

§ 57. *BLOS-flyging opp til 120 meter i luftrom klasse G*

BLOS-flyging opp til 120 meter i luftrom klasse G eller luftrom klasse G med etablert Radio Mandatory Zone (RMZ), kan kun skje hvis det er utstedt NOTAM for å informere om aktiviteten. NOTAM skal være utstedt minst 12 timer før aktiviteten påbegynnes.

BLOS-flyging i luftrom klasse G med etablert Radio Mandatory Zone (RMZ) kan i særlige tilfeller likevel skje etter tillatelse fra flygeinformasjonstjenesten og på de vilkår som flygeinformasjonstjenesten setter. Flygeinformasjonstjenesten kan kun gi tillatelse til slik flyging hvis det er klart at flygingen kan gjennomføres sikkert og uten å hindre øvrig lufttrafikk.

§ 58. *BLOS-flyging opp til 120 meter i kontrollert luftrom*

BLOS-flyging opp til 120 meter i kontrollert luftrom kan kun skje i aktive fare- eller restriksjonsområder.

BLOS-flyging kan unntaksvis skje utenfor fare- eller restriksjonsområde, etter klarering fra flygekontrolltjenesten og på de vilkår som flygekontrolltjenesten setter. Klarering skal kun gis hvis det kan etableres tilfredsstillende atskillelse mellom luftfartøyet som ikke har fører om bord og ethvert annet luftfartøy.

§ 59. *Påbudt lys*

For all flyging BLOS skal luftfartøyet være utrustet med lavintense lys, hvitt med minst 10 candela, hvor blink fremkalles ved roterende lys (strobelys) og med minimum 20 blink i minuttet.

§ 60. *Flyging i mørke*

Ved flyging i mørke skal luftfartøyet føre lanterner, jf. driftsforskrift 25. april 1974 nr. 4166 for ervervsmessig luftfart med fly.

Kapittel 9. Operative tilleggsbestemmelser som gjelder for RO 3-operatører

§ 61. *VLOS- og EVLOS-flyging i luftrom klasse G*

VLOS- og EVLOS-flyging høyere enn 120 meter er tillatt i luftrom klasse G. Operatøren er ansvarlig for å vurdere om det ut fra flygingens art eller området hvor flygingen skal skje, er nødvendig å utstede NOTAM for å informere om aktiviteten. Eventuell NOTAM skal være utstedt minst 12 timer før aktiviteten påbegynnes.

§ 62. *VLOS- og EVLOS-flyging i kontrollert luftrom og luftrom klasse G med etablert RMZ*

VLOS- og EVLOS-flyging i kontrollert luftrom, høyere enn 120 meter, kan kun skje etter klarering fra flygekontrolltjenesten og på de vilkår som flygekontrolltjenesten setter.

VLOS- og EVLOS-flyging i luftrom klasse G med etablert Radio Mandatory Zone (RMZ), høyere enn 120 meter, kan kun skje etter tillatelse fra flygeinformasjonstjenesten og på de vilkår som flygeinformasjonstjenesten setter. Tillatelse skal ikke gis hvis det er annen lufttrafikk i luftrommet.

I kontrollert luftrom og luftrom klasse G med etablert Radio Mandatory Zone (RMZ) skal det etableres toveis radiokommunikasjon eller telefonkommunikasjon mellom fartøysjefen og lufttrafikktjenesten.

§ 63. *BLOS*

Flyging BLOS er kun tillatt hvis tillatelsen fra Luftfartstilsynet omfatter denne operasjonstypen.

§ 64. *BLOS-flyging høyere enn 120 meter i luftrom klasse G*

BLOS-flyging høyere enn 120 meter i luftrom klasse G eller luftrom klasse G med etablert Radio Mandatory Zone (RMZ), kan kun skje i aktive fare- eller restriksjonsområder.

BLOS-flyging i luftrom klasse G med etablert Radio Mandatory Zone (RMZ) kan i særlige tilfeller likevel skje etter tillatelse fra flygeinformasjonstjenesten og på de vilkår som flygeinformasjonstjenesten setter. Flygeinformasjonstjenesten kan kun gi tillatelse til slik flyging hvis det er klart at flygingen kan gjennomføres sikkert og uten å hindre øvrig lufttrafikk.

§ 65. *BLOS-flyging i kontrollert luftrom*

BLOS-flyging i kontrollert luftrom kan kun skje i aktive fare- eller restriksjonsområder.

BLOS-flyging kan unntaksvis skje utenfor fare- eller restriksjonsområde, etter klarering fra flygekontrolltjenesten og på de vilkår som flygekontrolltjenesten setter. Klarering skal kun gis hvis det kan etableres tilfredsstillende atskillelse mellom luftfartøyet som ikke har fører om bord og ethvert annet luftfartøy.

§ 66. *Kontrolltap*

Ved tap av kontroll over luftfartøyet når det befinner seg høyere enn 120 meter, skal lufttrafikktjenesten varsles umiddelbart.

§ 67. *Påbudt lys*

Luftfartøyet skal være utrustet med lavintense lys, hvitt med minst 10 candela, hvor blink fremkalles ved roterende lys (strobelys) og med minimum 20 blink i minuttet, for all flyging

a) i kontrollert luftrom

b) over 120 meter i ikke-kontrollert luftrom
c) BLOS.

§ 68. ***Flyging i mørke***

Ved flyging i mørke skal luftfartøyet føre lanterner, jf. driftsforskrift 25. april 1974 nr. 4166 for ervervsmessig luftfart med fly.

Kapittel 10. Statsluftfart

§ 69. ***Statsluftfart***

Denne forskriftens bestemmelser, med unntak av § 18, gjelder tilsvarende for sivil statsluftfart med offentligrettslig formål i forbindelse med politivirksomhet, tollvirksomhet, offentlig søk- og redningstjeneste, brannslukking, kyst- og grensevakt eller lignende aktiviteter og tjenester.

§ 70. ***Områder med forbud mot flyging***

Flyging i restriksjonsområder og andre områder hvor forskriften fastsetter forbud mot flyging, er tillatt for sivilt statsluftfartøy som ikke har fører om bord i forbindelse med politivirksomhet, tollvirksomhet, offentlig søk- og redningstjeneste og brannslukking.

Kapittel 11. Sluttbestemmelser

§ 71. ***Overtredelsesgebyr***

Overtredelsesgebyr etter luftfartsloven § 13a-5 kan ilegges for brudd på reglene i kapittel 3 til 9.

§ 72. ***Suspensjon og tilbakekall***

Ved brudd på bestemmelser gitt i lov eller forskrift, eller de vilkår som er satt i tillatelsen, kan Luftfartstilsynet, helt eller delvis, suspendere eller trekke tilbake tillatelsen.

§ 73. ***Dispensasjon***

Luftfartstilsynet kan, hvor formålet er forskning på og utvikling av luftfartøy uten fører om bord, gi tillatelse til operasjoner eller testflyging selv om vilkår som følger av luftfartsloven med tilhørende forskrifter ikke er oppfylt.

§ 74. ***Ikrafttredelse og overgangsregler***

Forskriften trer i kraft 1. januar 2016.

Operatørtillatelse til virksomhet som opererer luftfartøy uten fører om bord som er innvilget av Luftfartstilsynet før ikrafttredelsen, er fortsatt gyldig til utløpet av gyldighetsperioden. Forskriftens bestemmelser gjelder fullt ut for virksomheten fra forskriften trer i kraft hvis ikke annet er fastsatt i denne bestemmelsen.

For tillatelser som utløper første fire måneder etter at forskriften trer i kraft, forlenges tillatelsen med seks måneder.

Hvis operatøren skal fortsette sin virksomhet som RO 1 etter utløp av gyldighetsperioden, må det innen utløpsdato sendes en melding til Luftfartstilsynet, jf. § 22.

Hvis operatøren skal fortsette sin virksomhet, men innenfor kategori RO 2, må søknad være mottatt av Luftfartstilsynet senest 3 måneder før utløpsdato. Hvis operatøren skal fortsette sin virksomhet, men innenfor kategori RO 3, må søknad være mottatt av Luftfartstilsynet senest 4 måneder før utløpsdato.

Kravene til luftdyktighet i § 41 gjelder ikke før 1. januar 2017. Frem til denne dato gjelder kravene i § 33 for RO3.

Fastsatte krav til operativ leder gjelder ikke for personer som allerede er godkjent som operativ leder av Luftfartstilsynet før ikrafttredelsen.

Pilot eller fartøysjef i virksomhet av kategori RO2 og RO3 må innen 1. februar 2017 ha bestått e-eksamen, jf. § 36 og § 46.

0 Endret ved forskrift 20 mai 2016 nr. 510.