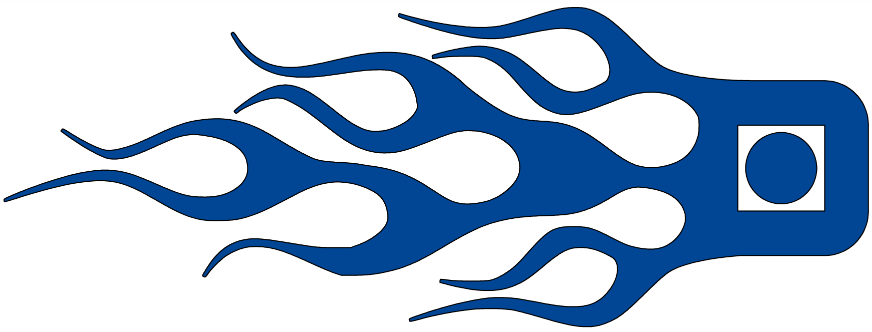
Bachelor's thesis

Magnus Stava
Markus Grorud Gaasholt
Kaung Htet San

Developing waypoint navigation and buoy detection using YOLO for an autonomous surface vessel

May 2021

NTNU
Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Department of ICT and Natural Sciences



NTNU
Norwegian University of
Science and Technology

Bachelor’s thesis

2021

NTNU

Norwegian University of
Science and Technology

Magnus Stava
Markus Grorud Gaasholt
Kaung Htet San

Developing waypoint navigation and buoy detection using YOLO for an autonomous surface vessel

Bachelor's thesis
May 2021

NTNU
Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Department of ICT and Natural Sciences

NTNU
Norwegian University of
Science and Technology

Preface

Autonomous surface vehicle is a topic with rapid development both in the industry and in the scientific environment. Some of Norway's universities have come together to create a national competition for ASV's. This competition aims to gather both students and professionals to showcase their sea drones.

This bachelor thesis is written by three students from Automatiseringsteknikk at NTNU Aalesund. What intrigued us to choose this project was the possibility to dwell deeper into future-oriented topics such as object detection and autonomous navigation at sea. The project also allowed us to design and to build a physical vessel, with all the electrical and software work this involves. The division of labor was split into four parts, construction, object detection, navigation, and report writing.

We would like to thank all our supporters, especially our supervisors for their guidance and the NAVO NTNU group for their assistance. We would also like to offer our gratitude to the lab employees for their support.

Abstract

The Autodrone 2021 is an annual national competition hosted in Horten. Autodrone aims to establish and strengthen the maritime higher education in Norway. Therefore NTNU Aalesund is interested in being a part of the competition, and this thesis is to support upon this interest. Last year (2020) was NTNU Aalesund's first year participating in the competition, but due to covid-19 restrictions, the physical competition got canceled. Nevertheless, last year's group gained important experience and laid the foundation for our vessel design. By moving forward with the work of last year's group, we set the goal to have a functioning vessel that meets all the requirements for the competition. In addition we will implement a vision system to localize and detect objects and a system for the vessel to navigate autonomously. All while keeping focus on modular solutions. This is important as it would make it easy for future groups to implement changes.

Abstrakt (Norwegian)

Autodrone 2021 er en årlig konkurranse arrangert i Horten. Den har som mål å etablere og styrke den maritime utdanningen nasjonalt. Av den grunn ønsker NTNU Ålesund å ha en tilstedeværelse i konkurransen og følgende rapport er for å støtte opp om dette. NTNU Ålesund deltok for første gang i fjor (2020). På grunn av den pågående pandemien ble den fysiske konkurransen avlyst. Erfaringene tilegnet av fjor årets gruppe, la grunnlaget for arbeidet vi har gjort gjennom prosjektet. Vi satte oss et mål om å ha en fungerende drone som møter de kravene som er satt av arrangørene, i tillegg vil det bli implementert et system for objekt deteksjon samt et system for å navigere dronen autonomt. Alle systemene er designet med fokus på det modulære. Dette for å fasilitere implementeringen av oppgraderinger for fremtidige grupper.

Acknowledgement

We would like to thank all the contributors who have given us support through the project, and in particular, we would like to thank:

- Our supervisors and resource persons Anete Vagale, Øystein Bjelland, Ottar L. Osen, and Robin T. Bye for all support, motivational conversations, and guidance throughout the whole project.
- Laboratory engineers Anders Sætersmoen and his assistants for assistance in purchasing and lending equipment.
- NAVO NTNU team for their support throughout the project.
- Last year's group for their assistance in the beginning of the project.
- Family and friends for their emotional support.

Summary and Conclusions

This report is about the development of an autonomous surface vehicle for the national autodrone competition. The report goes through the design process of the vessel and the reasoning for the chosen design. We will also present and explain what hardware and software were used to create the end product, such as YOLO object detection models and the functionality of pixhawk flight controller in-order to achieve autonomous navigation features. The end product is a vessel that utilizes waypoint navigation and YOLO buoy detection. We will present results that will highlight our object detection performance, stereo vision accuracy, in addition various measurements performed by our vessel, such as maximum speed, GPS accuracy, and autonomous navigation. In the final part, we will discuss the results and enhancements for future development.

Terminology

Darknet Framework with implementation of YOLO.

MavLink Communication protocol for drones.

MavROS Bridge between ROS and MavLink.

OpenCV Open source computer vision library.

RCNN Region based convolutional neural networks.

Abbreviations

ABS Acrylonitrile butadiene styrene.

AI Artificial Intelligence.

AP Average precision.

API Application programming interface.

ASV Autonomous Surface Vehicles.

CAD Computer-aided design.

CNN Convolutional neural network.

CM Camera mount.

DOF Degrees of freedom.

EKF Extended Kalman filter.

FDM Fused deposition modeling.

FFF Fused filament fabrication.

GNSS Global navigation satellite system.

GPS Global position system.

IEEE Institute of Electronical and Electronic Engineers.

IMU Inertia measurement unit.

IoU Intersection over Union.

IP Internet protocol.

mAp Mean average precision.

PETG Polyethylene terephthalate glycol.

PLA Polylactic acid.

RC Radio control.

ROS Robot operation system.

RTK Real time kinetic.

RTL Return to launch.

SL Stereolithography.

SLA Stereolithography apparatus.

TCP Transmission control protocol.

TM Thruster mount.

txt Text file.

UDP User datagram protocol.

USV Unmanned surface vessel.

xml Extensible markup language.

YOLO You Only Look Once.

List of Figures

List of Tables

Contents

Chapter 1

Introduction

As part of our bachelor thesis, throughout the spring semester of 2021, we have been developing an autonomous boat for the annual autodrone competition in Horten. We have gained experience in computer vision, computer programming, and electrical construction during studying automation, all of which proved valuable throughout the project. This report will outline our approach towards creating an autonomous vessel within the rules of the autodrone competition.

1.1 Background

Autodrone is a annual competition where teams from different Norwegian universities create an autonomous vessel to complete a speed, obstacle, path, and docking task. The year 2020 was supposed to be the first year of competing for NTNU Aalesund.

1.2 Problem Formulation

Design and develop a vessel that meets the Autodrone competition requirements and implement systems for buoy detection and autonomous navigation.

Problems to be addressed

- Design and construct the hull of the boat.
- Research and implement hardware.
- Develop software for navigation of the vessel.
- Develop object detection.

1.3 Objectives

The objectives for this report are:

1. Construct a physical vessel that meet the competition requirements
2. Implement the software foundation to solve the different competition objectives.
   (a) Speed
   (b) Object avoidance
   (c) Path tracking
   (d) Docking
3. Maintain focus on modular solutions, for simplifying work future groups.

1.4 Approach

To develop an autonomous vessel, we will build a vessel from scratch. Assisting us in this process and choosing material for the hull was NAVO NTNU, a group of naval engineer students. Experience from groups from other ASV competitions helped us pick equipment for the remaining hardware development. To measure the success of the implementation, various parameters such as speed, geolocation accuracy, distance readings, and other relevant measurement were stored locally when the vessel was operating. The gathered data will be presented through graphs and tables to give a clean presentation of the system's performance. As reference we will use some results from last years group.

1.5 Structure of the Report

The remaining of this report will be structure as follows.

Chapter 2 - Theoretical basis: Chapter two gives an introduction to the theoretical background that is responsible for the workings of the vessel.

Chapter 3 - Method: Contains a description of the methodology and materials that were considered throughout the project.

Chapter 4 - Result: Gives an overview of the finished autonomous vessel and its components.

Chapter 5 - Discussion: A summary of implementations and thoughts.

Chapter 6 - Conclusions: This chapter present an overall conclusion of the project.

Chapter 2

Theoretical basis

2.1 ASV seadrone

Autonomous surface vehicles are vessels that operate without assistance from a crew or input from a controller. Today, only a handful of ASV operates in the real world, but more research is being done, and progress is rapid. The transport sector is thought of as a suitable area where ASV can provide their usefulness, and already some cargo ships are built to be fully autonomous.

2.2 Points system

When completing a course, the vessel is given points on how well the completion was. The different courses and their points system is stated in the rules and illustrated in *Figure* 2.1, 2.2, 2.3 and 2.4 .[2]

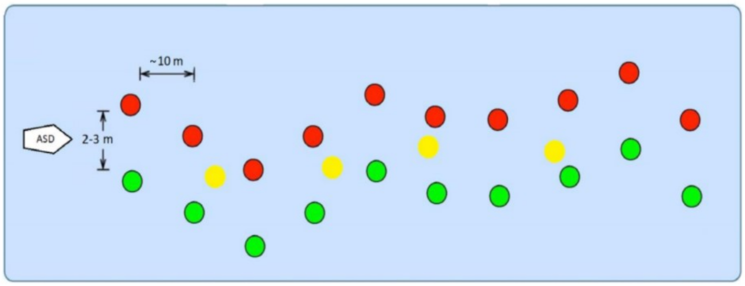


Figure 2.1: Path finding course
The vessel should navigate through gates consisting of red and green buoys while avoiding the yellow buoys. Five points are awarded for every gate that is successfully passed. If the vessel collides with the yellow buoys, a penalty of 10 points is added.

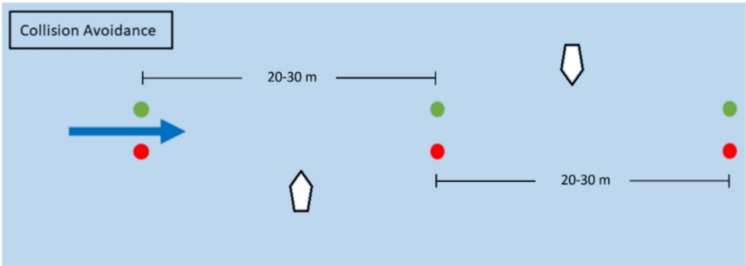


Figure 2.2: Collision avoidance course
The vessel should go from start to finish without colliding with crossing vessels. Ten points are awarded for passing the gates, and a penalty of 30 points is added if the vessel collides with another boat.

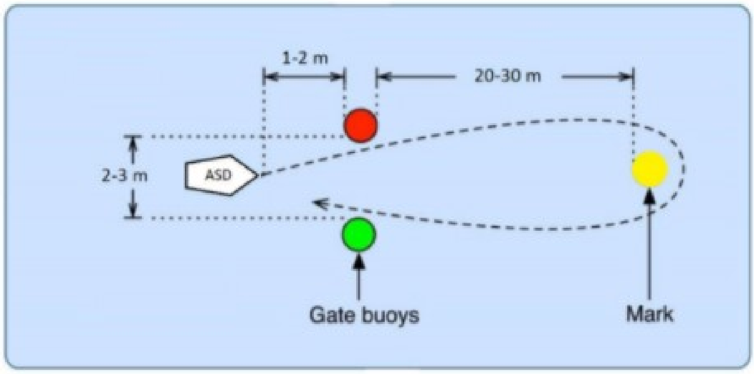


Figure 2.3: Speed course

The vessel should navigate to the yellow buoy, which indicate the turning point. Then it should return to start as fast as possible. In this course, points are given to the three fastest competitors. Eighty points for first place, 50 points for second place, and 20 points for third place.

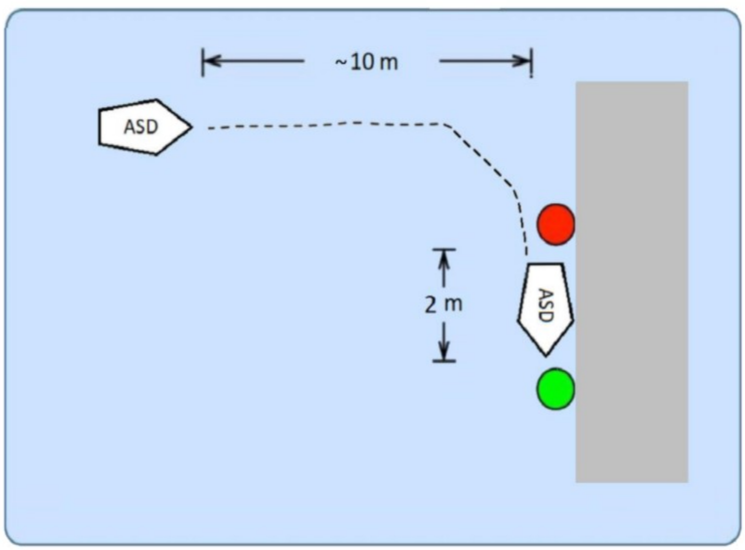


Figure 2.4: Docking course

The vessel should locate and initiate docking on the dock. Ten points are given for reaching the dock. Thirty points for correct docking and 2 points for every second the boat remains at the dock.

Points are also scored based on weight and thrust as seen in *Figure* 2.5

Parameters	Points
ASD weight > 70kg.	Disqualified
Dimensions greater than: - 0.9m width or - 0.9m feet of height - 1.8 m of length	Disqualified
Thrust (t) vs weight (w)	*100*(t / w)*

Figure 2.5: Competition requirement

2.3 Competition requirements

As stated on Autodrone [2] the requirements required to participate is shown in *Figure* 2.6

9.1 Sea Drone Requirements

- **Autonomy**: Drone shall be fully autonomous and shall have all autonomy decisions made onboard the ASD.
- **Communication**: The drone cannot send or receive any **control** information to and from Operators Control Station while in autonomous mode.
- **Deployable**: The ASD should be manually deployable.
- **Energy source**: The drone must be battery powered. All batteries must be sealed to reduce the hazard from acid or caustic electrolytes. The open circuit voltage of any battery (or battery system) may not exceed 60Vdc.
- **Kill Switch**: The drone must have at least one red button located on the drone that, when actuated, must instantaneously disconnect power from all motors and actuators.
- **Wireless Kill Switch**: In an emergency situation the operator control station must be able to actuate the kill switch on board the ASD.
- **Propulsion**: Any propulsion system may be used (thruster, paddle, etc.). However, all moving parts must have protection. For instance, a propeller must be shrouded.
- **Remote-controllable**: The drone must be remote-controllable from an operator control station.
- **Safety**: All sharp, pointy, moving or sensitive parts must be covered and marked.
- **Towable**: The drone must be towable.
- **Visual Feedback**: Teams are required to implement a visual feedback system, indicating status of their ASD. Additional information on this is available in Appendix 15.4 Visual Feedback.
- **Weight**: The entire maritime system (including UAV) shall weigh less than 70 kg.
- **Payload**: The drone must have a place to mount an action camera with an unobstructed view from the front of the drone.

Figure 2.6: Minimum requirements in order to participate in the competition

2.4 Autopilot

The purpose of the autopilot is to assist in controlling the drone. Their internal sensors, such as gyroscopes, accelerometers, and altimeters, allow them to control the vehicle's path and speed. For smaller drones, Pixhawk is a well-used autopilot due to its size and relatively low cost [3].

2.5 GNSS

Global navigation satellite system makes it possible for vessels, aircraft, and any other device to determine its position relative to the earth with the accuracy pending between some meters to a couple of centimeters. The source used for this is the royal observatory of Belgium [11]. After the US department first launched the GPS system, other countries launched their own system. Equipment today often utilize multiple of these systems. This is an advantage in case of a scenario where one of the systems fails; we would still be able to predict position using one of the other systems. The common name for these systems is GNSS

In addition to GPS, the most notable systems are:

1. GLONASS - Russia
2. Galileo - European satellite system
3. BeiDou - China

2.5.1 GPS, GLONASS and BeiDou

By utilizing satellites and with a clear view towards the sky, GNSS gives us the possibility to retrieve geolocation by using a technique known as 3-D trilateration. All modern GNSS equipment performs this calculation automatically.

3-D trilateration state that a point x can be calculated by using multiple known locations in space as shown in *Figure* 2.7. These known locations are our satellites, and GNSS uses a minimum of four satellites to calculate our position.

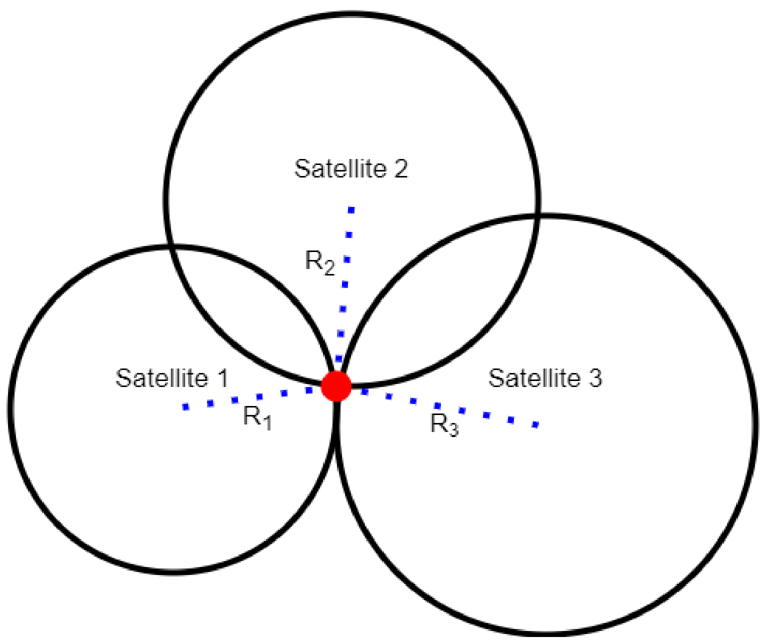


Figure 2.7: GPS location estimation using multiple satellites

To illustrate the concept, we can think of three satellites with known positions. The first problem to solve is to find the distance from the satellite to our GPS receiver. We know the signal speed as radio waves move at a speed of 300000km/s. To calculate the time, satellites continuously emit signals with a sequence of ones and zeros (pseudorandom code) known by all. The GPS receiver generates its identical pseudorandom code. The GPS receiver increasingly delays its internal code until it matches the signal received by the satellite's signal. Then the time of arrival and the time of transmission can be found. The distance is then calculated by using a simple equation as seen in *Equation* 2.1

$$distance = speed * time \tag{2.1}$$

Due to the importance of time, each satellite has an internal atomic clock that makes sure that the time is within three nanoseconds. If the offset becomes significantly larger, this signal is practically useless for calculating the position.

After finding the distance from the different satellites, each of these distances labeled R1, R2, and R3 will have a common area of intersection, which would equal to the location of our point illustrated by a red dot in *Figure* 2.7.

In reality, there are many other factors that causes errors and affect the precision. What systems and methods the satellites uses to solve these problems will not be a topic in this report.

2.5.2 RTK supported GNSS

RTK GPS increases the accuracy by adding a component known as a base station or base receiver. Our standard GPS receiver will be referred to as rover receiver illustrated as a red dot in *Figure* 2.8. The base receiver is located at a known location and since both the receivers are close to each other, the size of errors received from satellites will be identical. The base receiver calculates the direct distance to each of its satellites and then compares it to its measured distance. The difference equals the error size and represents the correction values we send to the rover receiver. The rover receiver then subtracts the correction values to its measured values, improving the position prediction from a few meters to a few centimeters.[9]

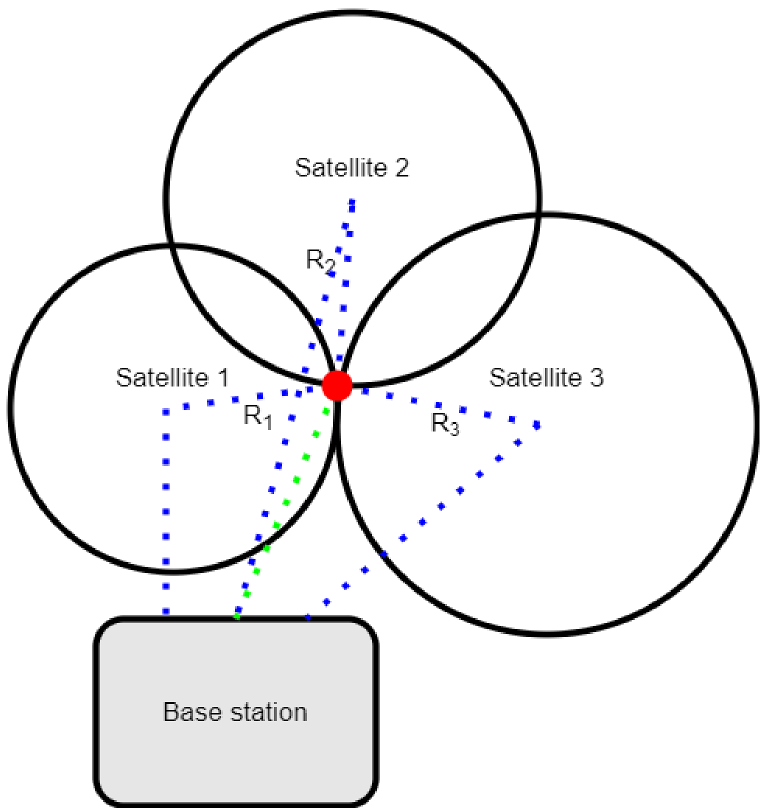


Figure 2.8: GPS system with additional base station for additional accuracy

2.6 Thruster configuration

The thruster configuration decides the movement behavior of the vessel. A favorable thruster configuration among small-sized drones is the X-shaped configuration. This configuration does not depend on any moving parts, making it less likely to break, and simple to build. It utilizes the mounting of the four thrusters, with thruster three and four having reversed blades. This makes the forward motion, as illustrated in *Figure* 2.9.

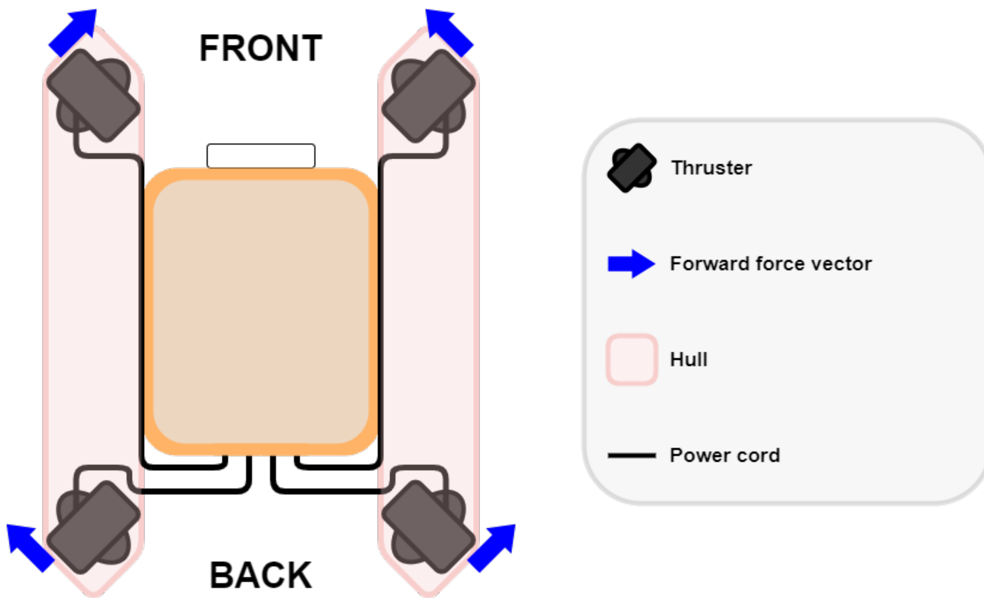


Figure 2.9: X-shaped thruster configuration

The X-shaped configuration allows the vessel to move back, forward, and right and left.
The forward thrust of the boat can be calculated by using each thruster's force vector. The T200 thrusters have a forward force of about 4.81kg and a backward force of about 3.74kg when powered with a 14.8V power source.[7] The theoretical forward thrust of the boat would be as seen in *Equation* 2.2.

$$ForwardThrust = 2 * (4.81kg + 3.74kg) * \sin 45° = 2 * 8.55kg * \frac{\sqrt{2}}{2} = 8.55kg * \sqrt{2} \approx 12.09kg \tag{2.2}$$

2.7 3D printing

3D printing is an additive manufacturing technique used to convert CAD files into physical objects. There are many ways of 3D printing, but a common factor for most of today's techniques is that they are done layer by layer. This method is considered 2.5D printing since it is not printing in all three dimensions simultaneously.

In this project two different 3D printing methods have been utilized. The methods are FFF/FDM and SL/SLA.

2.7.1 FFF/FDM printing

FFF (fused filament fabrication), or FDM (fused deposition modeling), is the method of continuously heating thermoplastic polymers until it melts and extruding it through a nozzle onto itself, so that is fuses together, making one part.[16] This printing method is a reliable and easy way of manufacturing custom parts. The parts are solid along their print lines but often suffer from delamination between their layers if they are not fused well enough together.

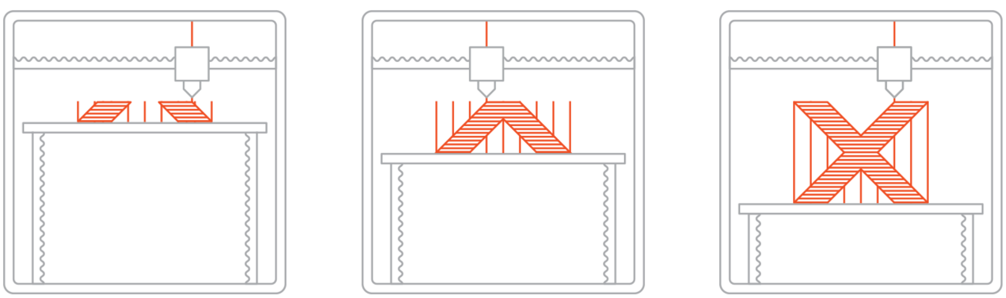


Figure 2.10: A simple visualization of FFF/FDM printing[16]

2.7.2 SL/SLA printing

SL (stereolithography), or SLA (stereolithography apparatus), is the method of precisely curing a photopolymer resin in layers to form a part.[17] The parts take a long time to print and need post-processing consisting of removing support material and final curing under a UV light. This printing method results in highly detailed but also brittle parts.

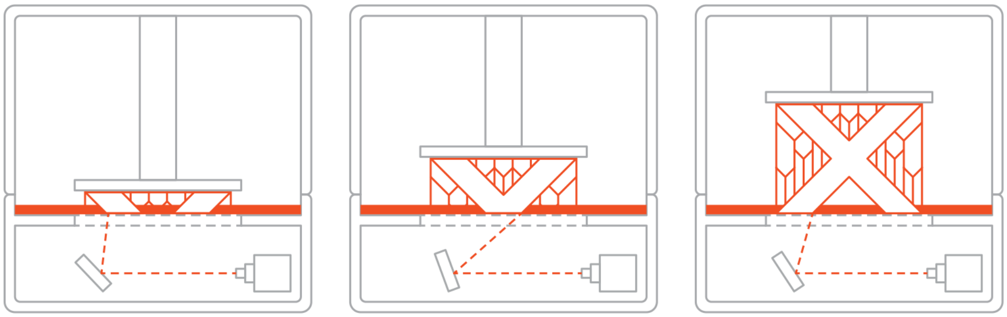


Figure 2.11: A simple visualization of SL/SLA printing[17]

2.8 Object detection

Object detection is the process of localizing an object and classify the object. For this report, we will focus on object detection using neural networks, more specifically the second and third version of the YOLO neural network. [13]

2.8.1 YOLO

YOLO, you only look once, is a neural network made to detect objects in real-time while maintaining accuracy. First published and shown to the world in 2015 by Joseph Redmon [12]. YOLO made it possible to achieve real-time object detection with a high speed without sacrificing performance. YOLO is quickly implemented and requires little preparation to be used, making it a favorable network among hobbyists and students. YOLO solves the problem of detecting an object in real-time with a speed that was unrivaled when it first came. The concept behind YOLO is that the model only sees the image once before making a localization and classification prediction. This is instead of passing the image through the model multiple times which was the standard way of performing object detection before YOLO.

2.8.2 Bounding boxes

To localize, the model generates a set of bounding boxes over areas in the image where it finds a high probability of an object. The model then uses these bounding boxes to predict which class is most likely present inside the box. To determine the performance of our model, we use intersection over union (IoU), which tells us how close our predicted bounding box is to ground truth. Ground truth is a manual box put over the object for training purposes, also known as labeling images. [6]

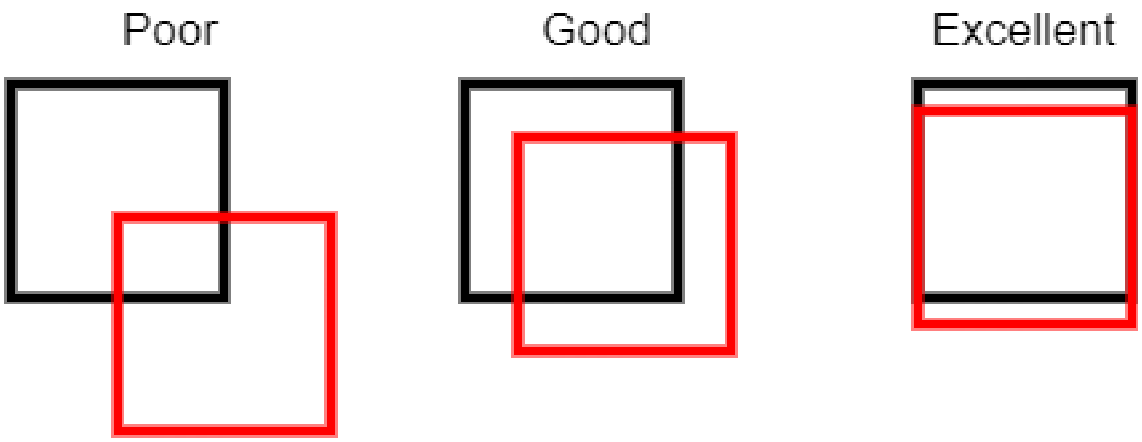


Figure 2.12: Red predicted bounding boxes and black ground truth boxes

IoU is calculated by taking the area where the predicted box overlaps with ground truth divided by the total area of both the predicted and ground truth box.

$$IoU = \frac{AreaOfOverlap}{AreaOfUnion}$$

As illustrated in *Figure* 2.12 we do not expect the model to perfectly fit the box. Generally we would classify a IoU greater then 50 % as a good localization prediction.

2.8.3 mAP, Mean average precision

The mAP is a metric for measuring performance of object detection model. This include both the localization and classification of the model. Mean average precision is defined as shown in *Equation* 2.3:

$$AP = \sum_{Recall} Precision(Recall_i) \tag{2.3}$$

Where

$$Precision = \frac{Tp}{Tp + Fp} \tag{2.4}$$

$$Recall = \frac{Tp}{Tp + Fn} \tag{2.5}$$

Equation 2.4 tell us how many of the model's predictions was correct. *Equation* 2.5 tell us how many of the total positives our model manages to correctly predict as true positive.

1. Tp = *True positive*: Correctly predicted an object that was there.

2. Tn = *True negative*: Predicted an object that was not there.

3. Fp = *False positive*: Correctly predicted no object.

4. Fn = *False negative*: Failed to predict an object that was there.

2.9 Data preparation

For the neural network model to function properly, it depends on good input data. Since neural networks aim to find similarities in the data and make predictions based on these similarities, any abnormal or missing data would affect the model's performance. When working with object detection models, they depend on a label to learn. This is asserted manually on the dataset and is a time-consuming process.

2.9.1 Data augmentation

A computer technique used to increase the dataset without physically taking more images. Using the existing dataset and modifying it, we can make the model believe it is a brand-new image. Data augmentation has the additional benefit of delaying overfitting, meaning the model can train for a longer period of time [14]. Data augmentation can be modification such as:

1. Geometric transformation
   (a) Transformation of images in either rotation or other types of shifting.
2. Color shifting
   (a) It makes the model more adjusted to color variances that occur.
3. Random noise
   (a) Adding random noise to images makes the model stronger against adversarial attacks, which is attacks utilizing noise.
4. Exposure
   (a) It makes the model more resilient to different lighting scenes.

Depending on where the model is implemented, certain augmentations might be more beneficial than others. To determine this, it is necessary to have enough domain knowledge or by trial and failure. In a light-exposed environment, a data augmentation with different exposure levels, such as shown in *Figure* 2.13 might be beneficial.

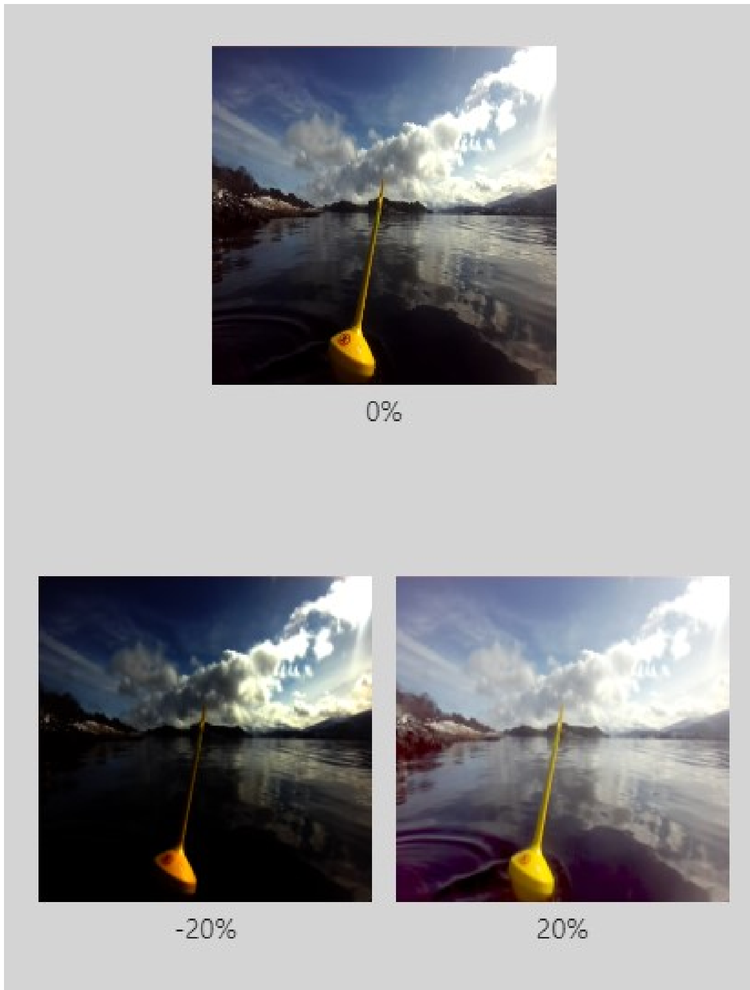


Figure 2.13: Yellow buoy with +/- 20% exposure. Makes model more resilient to lightning

In the same situation, implementing a geometric transformation where the images are flipped does not make sense. It is doubtful that the model will ever encounter a buoy that is upside down.

2.10 Computer stereo vision

To find depth information from an image, it's necessary to extract information in the form of image pixels. Using the relative position of an object's pixel from two different angles, it is possible to calculate the distance from the images by comparing the pixel displacement.

2.10.1 Depth calculation

Following are terminology used when talking about stereo vision [10].

1. **Fixation point** - Point of intersection of the optical axis.
2. **Baseline** - Distance or displacement between the two lenses.
3. **Epipolar plane** - The plane passing through the centers of projection and the point in the scene.
4. **Epipolar line** - The intersection of the epipolar plane with the image plane.
5. **Conjugate pair** - Any point in the scene that is visible in both cameras will be projected to a pair of image points in the two images.
6. **Disparity** - The distance between corresponding points when the two images are superimposed.
7. **Disparity map** - The disparities of all points form the disparity map (can be displayed as an image).

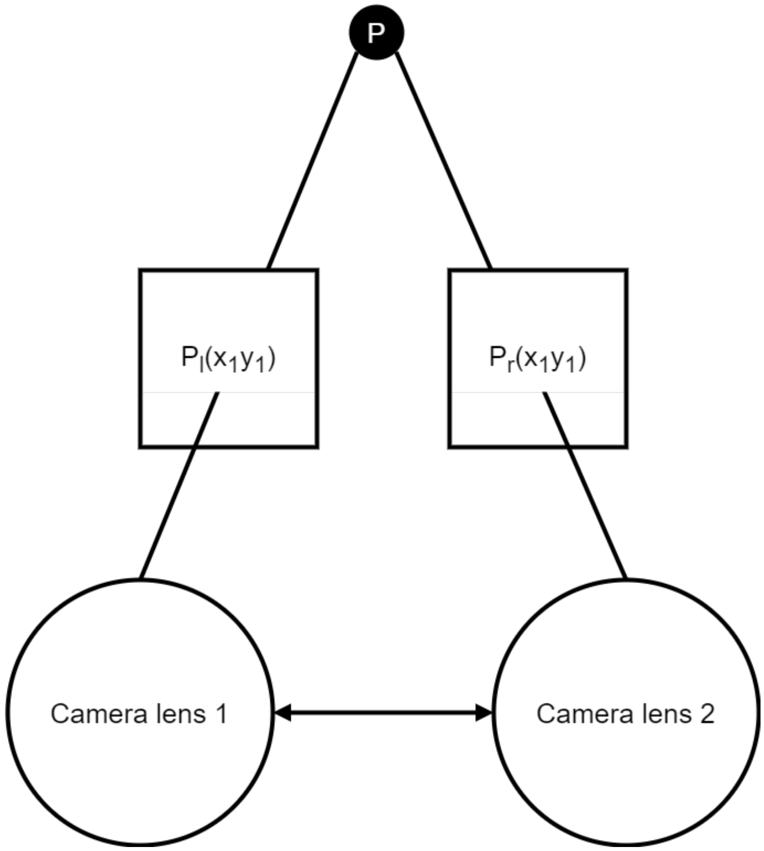


Figure 2.14: Left and right lens perceive same object P with the only difference being their horizontal displacement (baseline). Our interest is finding the depth to point P

By looking at the *Figure* 2.14. We can see that the lenses together with point P creates a plane which Pl and Pr lays on. Pl and Pr represents what each of the lenses can see. The line created between Pl and Pr is parallel with line between $lens1$ and $lens2$.

Therefor we can say that the y values between lens 1 and lens 2 is identical, which mean we can also conclude that the y value between P_l and P_r is identical: $y = y_l = y_r$.

In the following *Figure* 2.15 the two lenses is replaced by O_l and O_r. Object perceived by each lens is replaced by x_l and x_r.

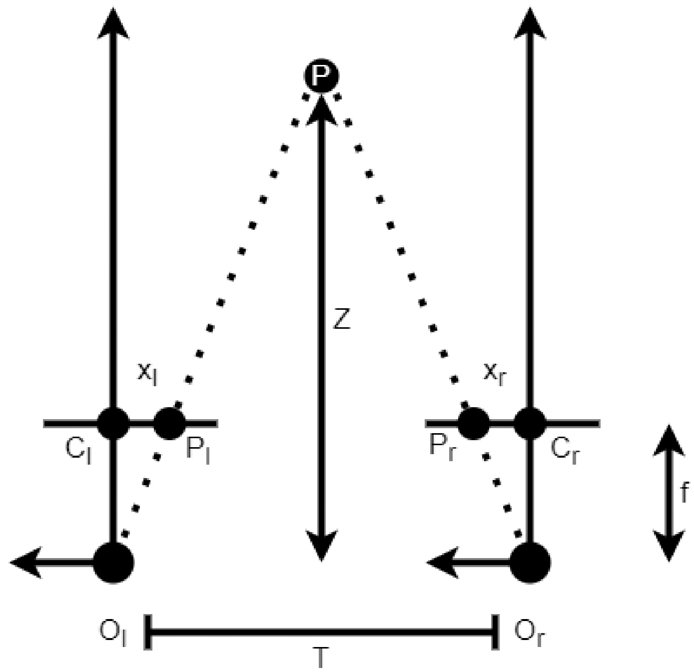


Figure 2.15: T represents baseline.
P represents the object that is being looked at

The calculation of depth Z is done by using a technique known as similar triangles. This states that triangles are equal if they have congruent angles when superimposed and sides are in proportion with each other.

This means that Z can be found by as shown in *Equation* 2.6

$$Z = \frac{f * T}{x_l - x_r} \tag{2.6}$$

Where f is the focal length. T is the baseline and $x_l - x_r$ is the disparity. Meaning we need to find x_l and x_r to calculate the depth.

2.11 Kalman Filter

Kalman filter is a tool to predict the values of a system. The Kalman filter, also called linear quadratic estimation, is a mathematical equation used in control engineering to calculate the means of collected data within a time limit to estimate the state of a process. This is to minimize the mean of the squared error. Kalman filter makes estimations with high accuracy and is used to make predictions and decisions. Kalman filter uses several measurements recorded over time and produces an estimate of unknown variables that are more accurate than the estimate based on a single measurement.[8]

For simplicity, we consider a state with two variables, position p and velocity v. Both these variables themselves tell us something about the system. The GPS sensor might tell us the position with an accuracy of 5 meters, and we might have information about the movement of our system. The point of a Kalman filter is to use both these variables to get a more accurate result. In a system where these variables are correlated, the likelihood of a position depends on the velocity. As illustrated in *Figure* 2.16 [1]

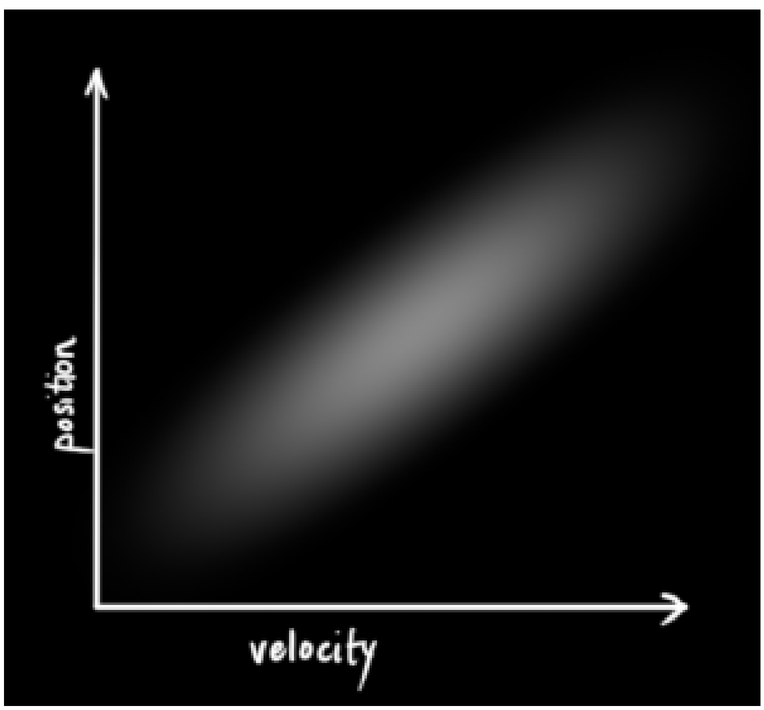


Figure 2.16: Position and velocity are correlated creating a area where observations are more likely to occur

A system where these are correlating could occur if the old variables of the system are used to calculate the new variables. The additional velocity can give us a more accurate position estimate than what the GPS can deliver alone. This is simply because a higher velocity at the previous position means that certain positions are more likely to occur in the future. Higher speed equals greater distance traveled. There might exist multiple relationships such as speed and velocity for a system with more variables. To keep track of the correlation between variables, they are stored in a covariance matrix, as illustrated in *Equation* 2.7. Where each element represents the correlation between the *ith* state variable and *jth* state variable

$$\sum_{.} {}_{ij} \tag{2.7}$$

The kalman filter best estimate can be denoted as $\hat{x}_k$ and its covariance matrix as P_k, which gives us the equation illustrated at 2.8.

$$P_k = \begin{bmatrix} \sum_{.} {}_{pp} & \sum_{.} {}_{pv} \\ \sum_{.} {}_{vp} & \sum_{.} {}_{vv} \end{bmatrix} \tag{2.8}$$

We continue with our same system as before with only velocity and position variables. The best estimate is calculated in the prediction step, which can be represented as F_k. It takes the estimate as input and moves it to a new predicted location, where the system would be if the estimate is correct.

$$p_k = p_{k-1} + \Delta t v_{k-1} \tag{2.9}$$

$$v_k = 0 + v_{k-1} \tag{2.10}$$

Which equals to *Equation* 2.11

$$\hat{x}_k = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix} \hat{x}_{k-1} \tag{2.11}$$

In addition, we also need to update the covariance matrix mentioned earlier to have a complete loop. This is achieved by multiplying every point by a matrix *A*.

$$Cov(Ax) = A\sum A^T \tag{2.12}$$

Then by combining equation 2.12 and equation 2.11 we get a equation for updating the covariance matrix. Illustrated by *Equation* 2.13.

$$P_k = F_k * P_{k-1} * F_K^T \tag{2.13}$$

In a real-world system, you have additional variables not related to the state. These variables can be known, such as when a driver accelerates by pressing the pedal or when a software sends some navigation commands. This extra information can be put into a vector $\vec{u}_k$ with a corresponding control matrix B_k. This means that the best estimate *Equation* 2.8, will be changed to *Equation* 2.14

$$\hat{x}_k = F_k * \hat{X}_{k-1} + B_k * \vec{u}_k \tag{2.14}$$

The precision of the prediction can be improved by including additional external sensors. Sensors are anything that provides a set of values based on a given state. We can represent them through a matrix H_k. This changes the best estimate to *Equation* 2.15

$$\begin{aligned} \vec{\mu}_{\text{expected}} &= H_k * \hat{x}_k \\ \Sigma_{\text{expected}} &= H_k * P_k * H_k^T \end{aligned} \tag{2.15}$$

2.11.1 Extended Kalman Filter

Extended Kalman filter is a non-linear function to predict dynamic systems. Extended Kalman filter predicts values of dynamic systems such as robotic applications. Extended Kalman Filter first linearizes the non-linear functions and then operates on the same concept as kalman Filter mentioned in *Section* 2.11.

2.12 ROS

Robot operating system (ROS) is an open-source robotics middleware. ROS aims to combine the communication and integration between technology used in robotics applications. ROS utilizes python and C++ for development. ROS can be divided into three levels: the filesystem-, computation graph- and community level. Most notable of these for development purposes is the filesystem level.

2.12.1 Filesystem level

The main purpose of the filesystem level is to provide the developer with a centralized build place for the project, and also provide branches to keep the projects dependencies decentralised. The structure of the ROS filesystem is shown in *Table* 2.1

Table 2.1: ROS filesystem structure

Packages	The base level of ROS, at minimum it contain enough structure and information to create a program.
Package manifests	Contains general information about packages.
Metapackages	Used to combine multiple packages.
Metapackage manifests	Contains general information about metapackages.
Message/msg	Information/data that is being sent from one process to another.
Service/Srv	Defines the request and response structure for services provided by each processes.

Chapter 3

Materials and methods

3.1 Materials

The following materials shown in *Table* 3.1 were used in this project.

Table 3.1: List of materials

Equipment	**Amount**	**Price** *(NOK)*
T200 thrusters	4	6000
Basic ESC	4	1000
Hex Here 3 GPS	1	1200
Jetson TX2	1	6000
Pixhawk 1	1	1000
Zed 2	1	3800
Rele Solid State	1	1100
Emergancy stop	1	1200
Radio Reciver	1	-
USB WiFi antenna	1	-
Trolley for transport	1	1000
Paint	1	60
Epoxy	1	220
White pigment for epoxy	1	110

3.2 Data gathering

Data gathered was completed by the previous group. This means the images used for training and the labeling process was not taken or labeled by us. The data used for training was increased by data augmentation.

3.2.1 Data augmentation

To increase the dataset we augmented 15% of the existing images with a +/- exposure of 25%. This was done to increase the robustness of the model against varied light environment.

3.3 Design and preparation

3.3.1 Hull

When choosing the design of the vessel, it was important for us to pick a solution that made future work simple.

When choosing the catamaran as our hull configuration, it was because we wanted a hull that provided enough space to mount our equipment and to still have space for future upgrades. Alternatives such as a single hull would have given the vessel an increased speed, but made it more complicated to get the vessel to move in all directions.

3.3.2 Wiring

Some aspects of the wiring were regulated by the competition requirements, such as the physical kill switch mounted on the vessel and the electronic kill switch operated from the ground control station. See *Section* 2.6 for full overview of the requirements. The emergency switch is wired as the first component, and when pressed, it will cut power to all equipment on board.

The electronic kill switch utilizes a transistor as a switch with the Pixhawk as the input. It is wired after the Jetson TX2 and is part of the circuit that switches on/off the solid-state relay, which in turn swithces on/off the power to the thrusters.

3.3.3 Thruster mounts

The thruster mounts (TM) have been designed in a CAD software to fit the thrusters and the boat's hull perfectly. The thrusters have four threaded holes for fastening them. The TM gives a 45° rotation to the thrusters. To keep all the motors mounted correctly, there are two different types of TMs where the second version is a mirror of the first.

After mounting the thrusters to the TMs, they are mounted to the boat's hull. To compensate for the unevenness of the mounting points on the hull, the TMs are slotted. The TMs are mounted to the hull with an M5 screw and nut.

(a) The topside of the thruster mount (TM) (b) The underside of the thruster mount (TM)

Figure 3.1: Rendered images of the thruster mount

The TMs were originally made in PLA, but it was changed due to PLAs material properties. PLA can absorb water, making the material weaker and more fragile. It was planned to print the parts in ABS or PETG with the Prusa MK3S+ 3D printers available at NTNU in Aalesund. This would give sturdy and waterproof TMs. It was not possible to print in these materials. The TMs ended up being 3D printed with a Formlabs Form 3 SLA printer (see [5] for printer manual). The resin used for the print is standard white resin (FLGPWH02) from Formlabs (see [4] for material datasheet). This gives waterproof and sturdy, but also brittle TMs.

3.3.4 Thruster configuration

Part of the reason for opting to use a catamaran was the possibility of using an X-shaped thruster configuration. An advantage with this configuration is that the boat is able to move in every direction even with thrusters mounted in a fixed position. By having the thruster mounted in this way it limits the amount of moving parts, which makes it less likely for something to break.

An alternative to this would be the azimuth configuration. This would give the vessel increased maneuverability when operating since the thrusters can rotate around their axis. The azimuth mounted thrusters would better handle sharp turns and small adjustments.

Additionally, this type of thruster configuration would likely increase the vessel's maximum speed, since the thrusters force vectors would be parallel with the forward force vector see *Figure* 3.8. Compared to the X-shaped configuration it increase the possibility of equipment breaking due to the increase of moving parts.

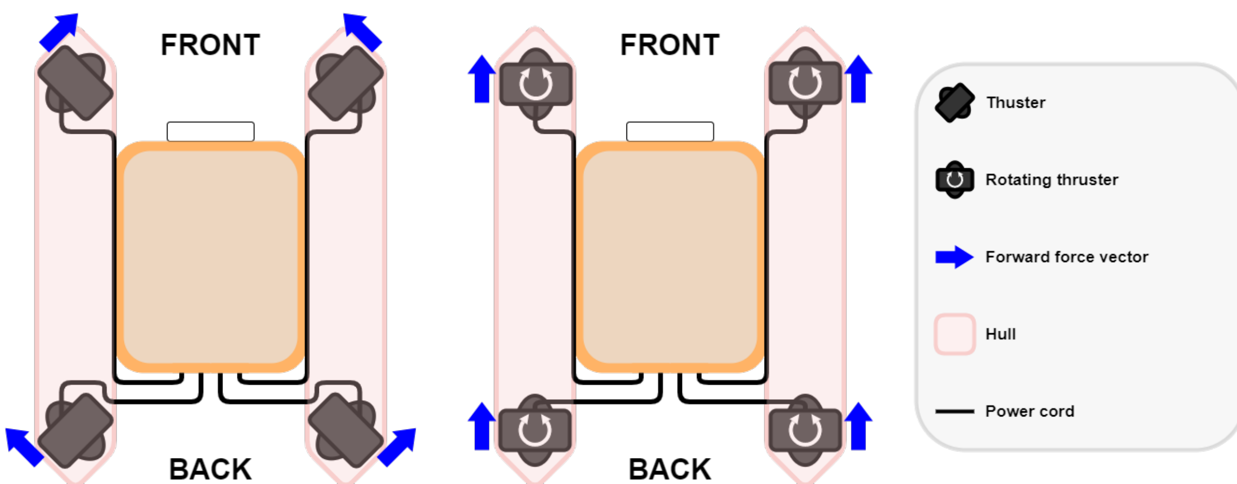


Figure 3.2: X-shaped (left) and azimuth (right) thruster configuration. Since all the thrusters move the vessel forward, the azimuth configuration achieves higher speed

3.3.5 Pixhawk - Kalman filter

To use the predefined extended kalman filter in the PX4 it is necessary to check that the "EK2_ENABLE" is set to one, and that "AHRS_EKF_TYPE" is set to two. The "AHRS_EKF_TYPE" tells the Pixhawk to use the kalman filter for position estimation. All configuration parameters was found in Pixhawks documentation [15].

3.3.6 Jetson TX2

To utilize the tools and equipment that was planned, the Jetson TX2 was set up with the following software shown in *Table* 3.2.

Table 3.2: Software installed on the Jetson TX2

ROS Melodic with packages	Development environment for the project
CUDA	In order to utilize GPU for neural network training
OpenCV	To make training faster
Darknet	Object detection framework with YOLO

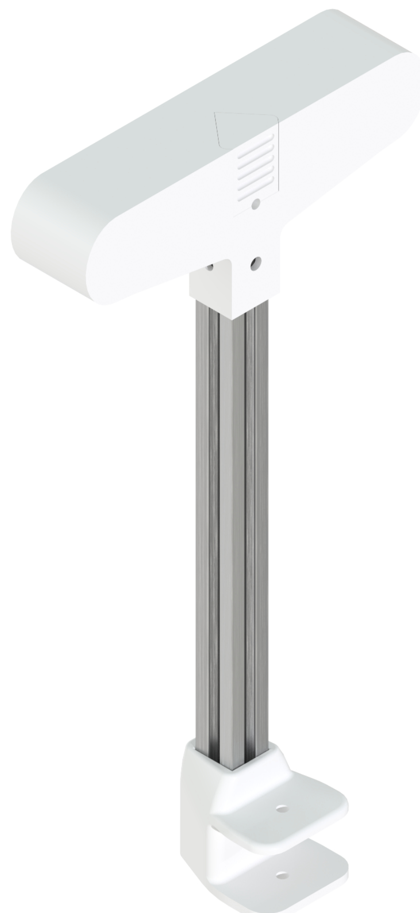
3.4 Stereo vision

Since we need pixels of a object from different angles to calculate the distance as described in *Section* 2.10.1, we opted for a camera that was equipped with two monocular lenses. We choose the Zed 2 Stereo camera due to its good reviews and continuous support for the ROS framework. Zed 2 is a parallel calibrated camera, meaning the only difference between the two lenses is the horizontal displacement. Zed 2 has the additional benefit of doing the depth calculation as an in-built feature meaning that we would only need to extract the information from the zed node in ROS to use it.

3.4.1 Camera mount

The camera is mounted onto the boat using a custom-made camera mount (CM). The camera casing and the fixture are the two main parts of the CM. The casing and fixture are mainly 3D printed in PLA using a Prusa MK3S+ FDM printer.

(a) The front side of the camera mount



(b) The back side of the camera mount

Figure 3.3: Rendered images of the camera mount

The casing fits the camera, making it protected and held steady. The front of the case has a slot for a protective 4mm thick acrylic panel. When the camera is inserted into the casing, the back lid of the casing slides nicely down around the cord, and the acrylic panel snaps into place. The case is fastened to a 20mm x 20mm x 300mm aluminum extrusion using three M4 screws, washers, and oval T-nuts with M4 thread. The case is placed at this height to give the camera a greater field of view.

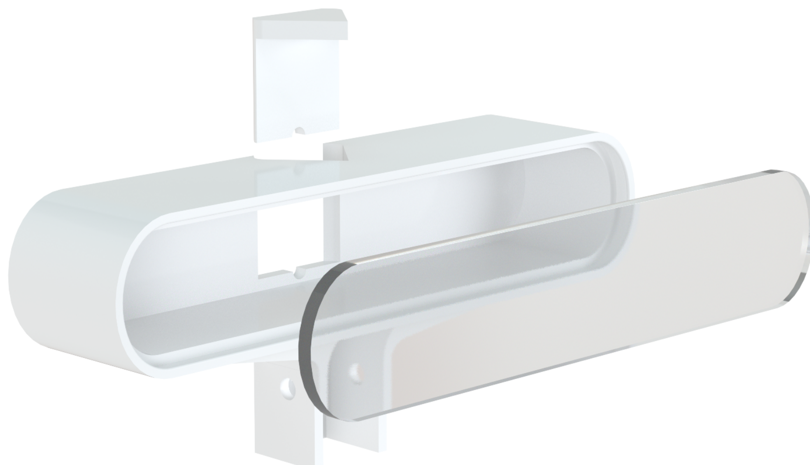


Figure 3.4: A rendered image of the exploded view of the casing

The fixture is clamped to the USV's front using an M4 bolt, two washers, and a nut. On the inside of the fixture there is a slot, as seen in *Figure* 3.5, where a 3mm thick M4 nut is glued on. The fixture is made so that it is easy to insert and remove the aluminum extrusion when transporting the vessel. The aluminum extrusion is held firmly in place by screwing in an M4 screw through the hole and nut at the front of the fixture.

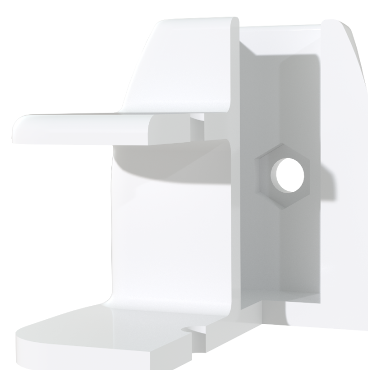


Figure 3.5: A rendered split-view of the fixture. Inside the fixture there is a hexagonal hole for fitting a nut

Due to the simpleness of the fixture, any tool or device that is mounted to a 20mm x 20mm square can easily be mounted onto the ASV.

3.5 Measurements

3.5.1 Stereo depth accuracy

To measure the accuracy of the distance calculated by the zed 2, a test with determined distances will be conducted. This will give us a understanding of how much these readings can be trusted depending on the distance that is being read. The determined distances will be set to:

1. 1m
2. 2m
3. 5m
4. 10m
5. 20m

3.5.2 Accuracy of the GPS

By holding the vessel at a known location we will store the GPS readings locally in the vessel. The measurement will then be extracted in order to analyse how accurate the GPS manage to stay during the test. A displacement more than 2.5 meters would be a unsatisfactory results as this is the promised accuracy according the Here 3+ datasheet.

3.5.3 Speed

Using the IMU inside the Pixhawk we can read the current speed. We will therefore take continuous speed readings during testing and store the highest value locally in the vessel.

3.5.4 Thrust

The thruster configuration decide how efficient the vessel moves. The competition gives points depending on the amount of thrust the vessel can output relatively to its weight. The test was done by reading the output of a handheld digital weight connected to the boat through a rope.

3.5.5 Weight

The weight of the vessel affects both speed and how the boat maneuvers. In the competition it affects points scored in the as shown in *Figure* 2.5. The weight will be measured by placing the vessel on a scale.

3.5.6 YOLO, mAP, precision and Recall

The dataset received from previous year was configured to train on a different operating system than what we used. This meant that the label files had to be converted from .xml files into .txt files. To increase the dataset we prepared a new set of images by adjusting the exposure levels. This we believe can make the model more tolerable to different light environment, see *Section* 2.9.1.

The two first models used the yolov2-tiny configure (cfg) file, which is a smaller model then the standard YOLOv2 cfg. The -tiny version of the models are made for less powerful systems and are perfect for our use-case. As starting weights, we used the accompanied yolov2-tiny.weights.

After training two models we trained an additional model based on the YOLOv3-tiny network architecture, which is the generation after YOLOv2. The difference between YOLOv2 and YOLOv3 is not grand, and it depends on your specific case, which of them will perform better.

To measure the model's performance in total, we will use mean average precision (mAP) as mentioned in *Section* 2.8.3. Additional measurement such as precision and recall will give us basis as to what we can expect from the model.

To measure the localization accuracy of the predicted object, we will use intersect over union (IoU). As mentioned in *Section* 2.8.2

3.6 ROS programming

Programming for the vessel is done in Python client for ROS, rospy. ROS melodic and used packages is installed on the Jetson TX2. The Jetson communicates with the Pixhawk through mavlink protocol, which is accessible in ROS through their package, mavros.

3.6.1 Arm

Main component to run any commands or functions on the vessel. Arm puts the vessel into a operative state and disarm into a non operative state. When arm is first called the home position is set by the Pixhawk.

3.6.2 waypoint_clear_client

Removes previous waypoints in the autopilot. When called it waits for "WaypointsClear" service to become available. When available it sets the bool variable to True, which tells the autopilot to remove previous waypoints.

3.6.3 create_ waypoint

In create_waypoint the program will first clear any previous set waypoints. Before defining the parameters in the waypoint topic. The waypoint topic takes a list as input and we therefor need to put the waypoint information into a list before we push it to the autopilot. The information needed is shown in *Table* 3.3:

Table 3.3: Waypoint variables with definition

Variable name	Meaning
frame	frame of vehicle, for Pixhawk only use global frame (0)
command	Defines what action to perform
is_current	'True' if waypoint is currently active, 'False' if not
autocontinue	If 'True', goes to next waypoint after reaching previous waypoint
param1	Time to hold position when reached waypoint
param2	Radius from waypoint that should be accepted as reached waypoint
param3	Radius to pass around the waypoint
param4	—
x_lat	How far to move in meters, positive is north, negative is south
y_long	How far to move in meters, positive is east, negative is west
z_alt	How far to move in meters, positive is up, negative is down

The information flow of waypoint setting is illustrated in *Figure* 3.6

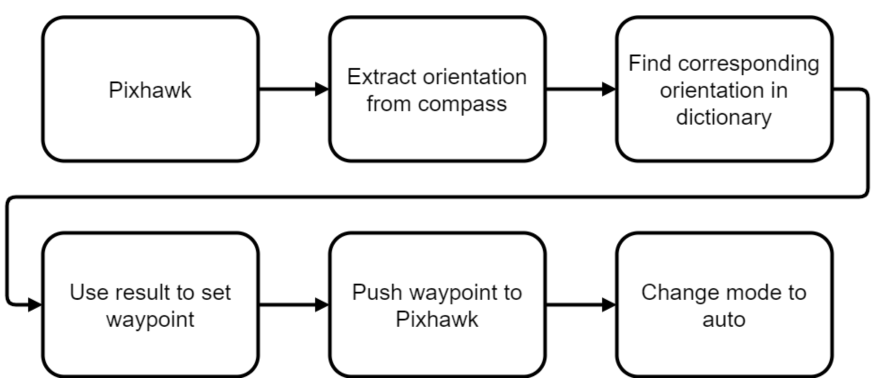


Figure 3.6: The flow of data when setting waypoint

3.6.4 Waypoint values

For our purpose, we consider the coordinate frame of the vessel to be in two dimensions, x, and y since z does not matter. We have a predefined dictionary with x and y values for angles with 10-degree intervals. The x and y represents a set distance of 10 meters in the given direction. This means a orientation of 53 degrees will be regarded as 50 degrees before looking up in the dictionary to find the x and y values.For the future the waypoint can be set to any distance by using the relations between circles. To achieve this it is only necessary to multiply the x and y values with a value a which can be found as shown in *Equation* 3.1

$$a = \frac{radius_{new}}{radius_{old}} \tag{3.1}$$

3.6.5 Return to launch - RTL

The return to launch (RTL) feature is thought to be the vessel's go-to method after completing some specific mission. This could be after going around the yellow buoy in the speed course or after being docked for the maximum amount of time. Therefor another important aspect of the RTL feature was how quickly it returned and with what accuracy. This meant that we both measured the accuracy between the point of return and the point of launch, and the time it used to get back from a distance of approximately 20 meters. The returning position is supposed to be the same position as where the vessel was launched. By disarming and arming the launch position can be reset, this would be necessary to include in the code as part of the start procedure. See *Figure* 3.7

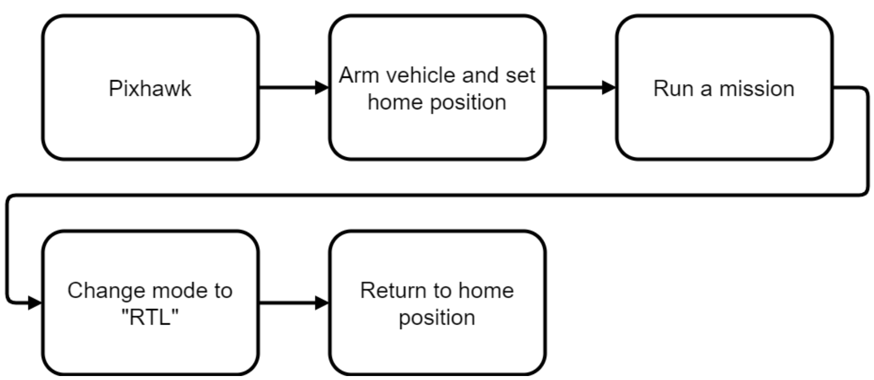


Figure 3.7: The flow of data when vessel using return to launch mode

3.6.6 HOLD mode

The HOLD mode is thought to be the vessels go-to method for remaining in a position. In reference to the competition courses this is useful when dealing with the docking course, which states that the drone should remain docked for at least 30 seconds, see *Section* 2.4 for full course description. The mode is set by sending a request for mode change to the Pixhawk. See *Figure* 3.8 for an illustration.

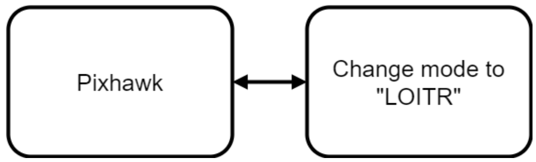


Figure 3.8: The flow of data when the vessel maintains it own position

Chapter 4

Results

The tests were performed by the ocean near NTNU Aalesund.

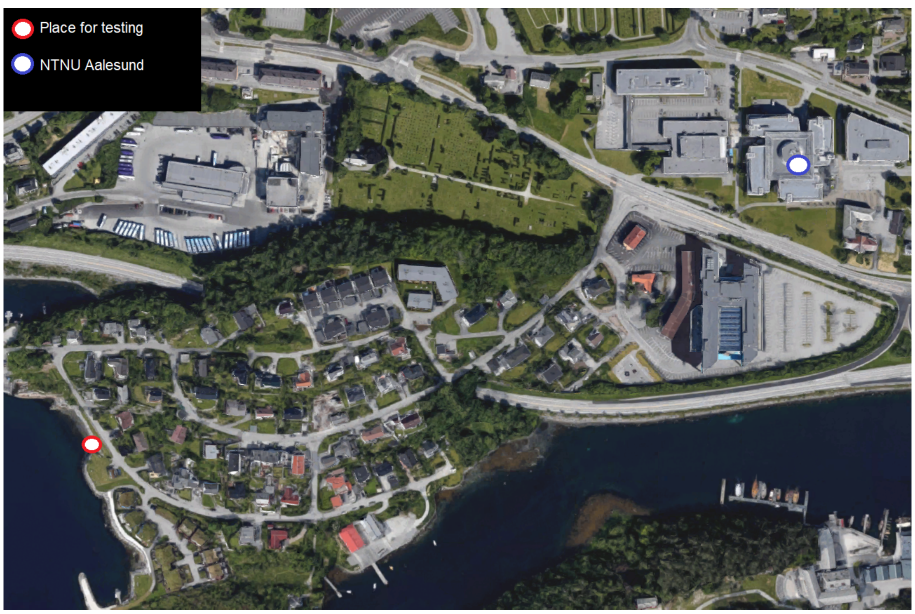
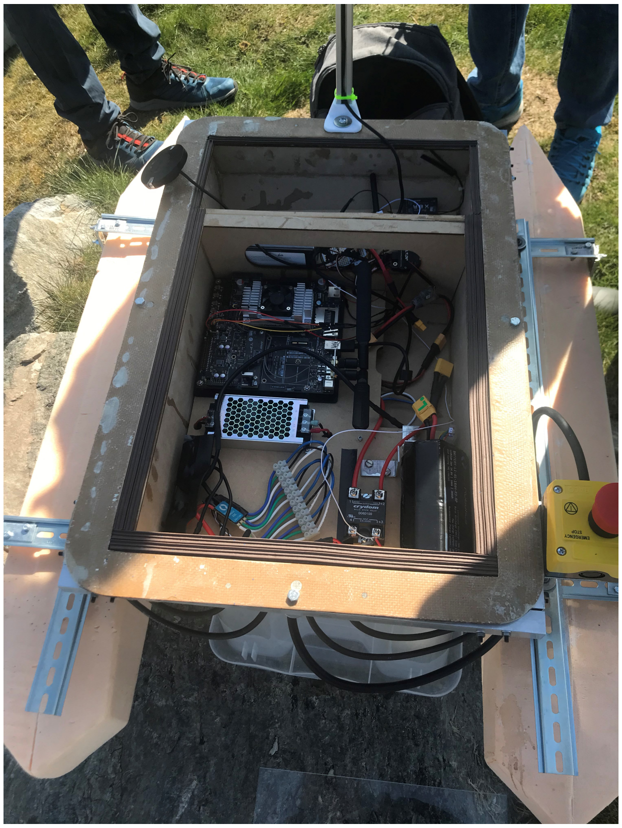


Figure 4.1: Location for testing the vessel

4.1 Vessel

The vessel's appearance.



(a) The inside of the vessel

(b) The outside of the vessel

Figure 4.2: The vessels appearance

4.1.1 Dimensions of the vessel

The dimensions of the vessel is as shown in *Table* 4.1:

Table 4.1: Dimensions of vessel

Dimensions	*Cm*
Width	73
Length	100
Height (with camera mounted)	80

4.1.2 Circuit

The electronic equipment is wired as shown in *Figure* 4.3

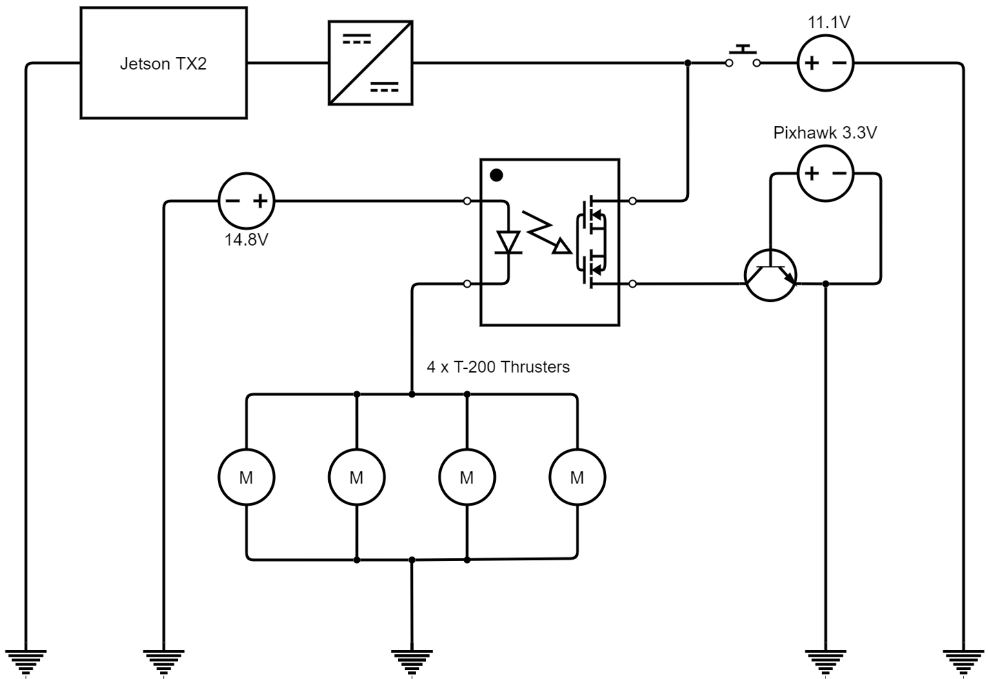


Figure 4.3: Circuit diagram of the electronic components

4.1.3 Operating time

Our testing confirmed that the battery satisfies our requirements. The presented results in *Table* 4.2 show greater than one hour as this was the longest period of time we tested the vessel.

Table 4.2: Operating time

Test	*Time (min)*
1	Greater then 1 Hour
2	Greater then 1 Hour

4.1.4 Weight

The weight of the vessel loaded with all the equipment is shown in *Table* 4.3:

Table 4.3: Weight

Object	*Weight (kg)*
Vessel	15

4.1.5 Speed

The speed of the vessel measured is shown in *Figure* 4.4

Test	Speed (m/s)
1	1,2
2	1,3

Figure 4.4: Max speed measured over two tests

4.1.6 Pull force

The maximum pull force over a two second time illustrated in *Figure* 4.5:

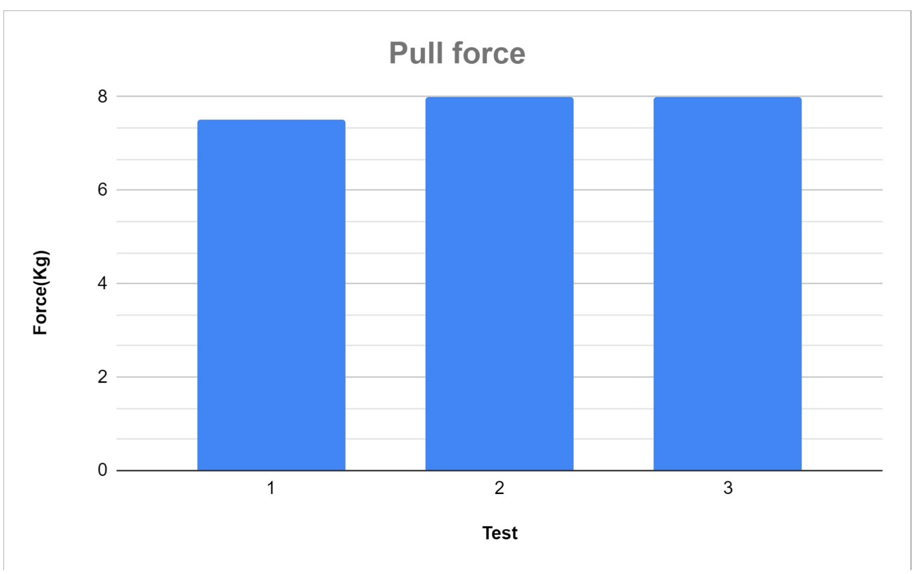


Figure 4.5: Pull force of the vessel on maximum throttle

4.1.7 GPS accuracy

Collected GPS position over a 10 minutes time period. Google maps were used to insert points and measure features to create the circles. Results presented in *Figure* 4.6

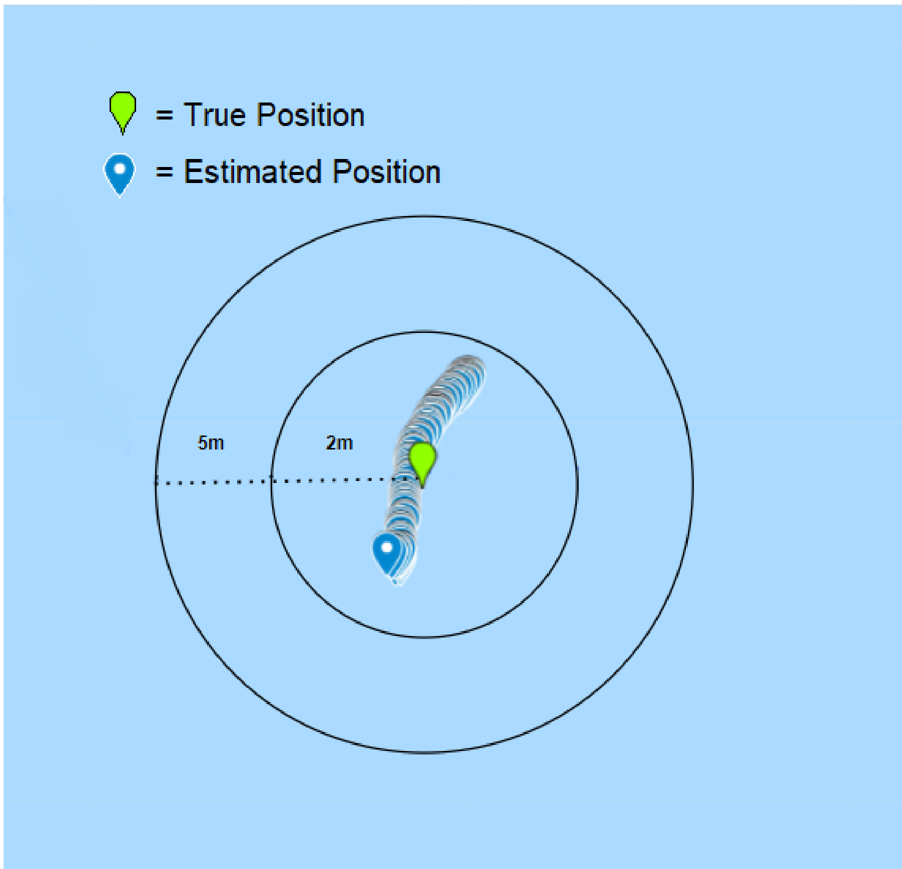


Figure 4.6: GPS accuracy

4.2 YOLO - Object detection

4.2.1 YOLOv2 - mAP score

mAP score of the first (v1) and second (v2) YOLOv2 model illustrated in *Figure* 4.7.

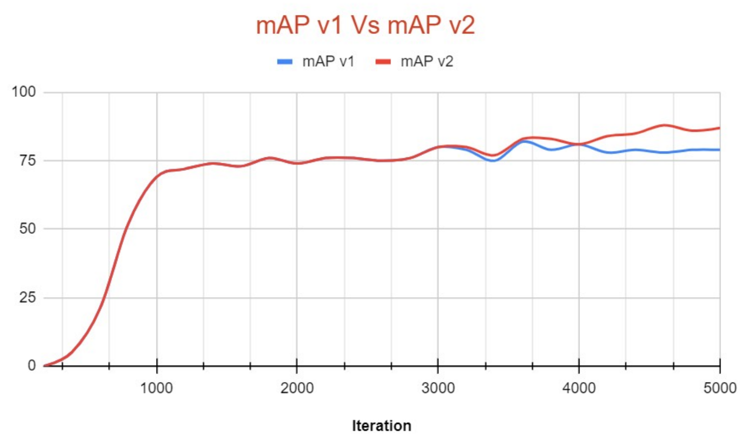


Figure 4.7: YOLOv2 - Mean average precision for first and second model

For the remaining, the results from the second (v2) YOLOv2 model, will be used and referred to as YOLOv2.

4.2.2 YOLO - Localization

Intersect over union score of the YOLOv2 and YOLOv3 model shown in *Figure* 4.8.

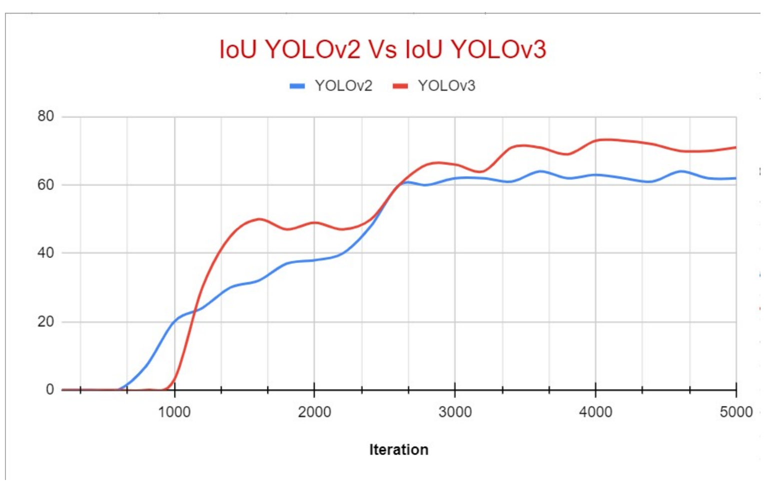


Figure 4.8: YOLOv3 IoU score compared against YOLOv2 IoU score

4.2.3 YOLO - Classification

Recall score of the YOLOv3 model compared to YOLOv2 shown in *Figure* 4.9.

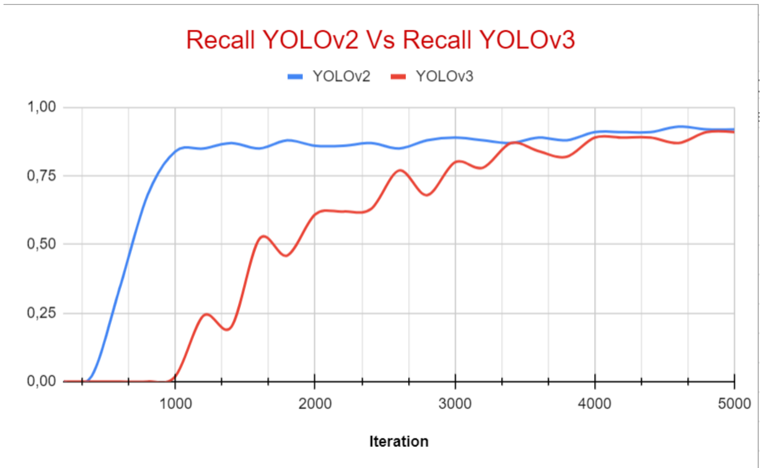


Figure 4.9: YOLOv3 classification score, recall and precision

Precision score of the YOLOv3 model compared to YOLOv2 shown in *Figure* 4.10.

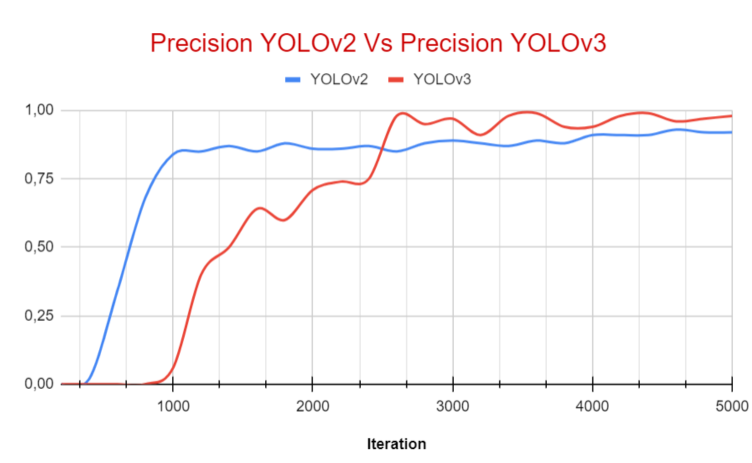


Figure 4.10: YOLOv3 classification score, recall and precision

True positive and false negative distribution of the YOLOv2 model illustrated in *Figure* 4.11.

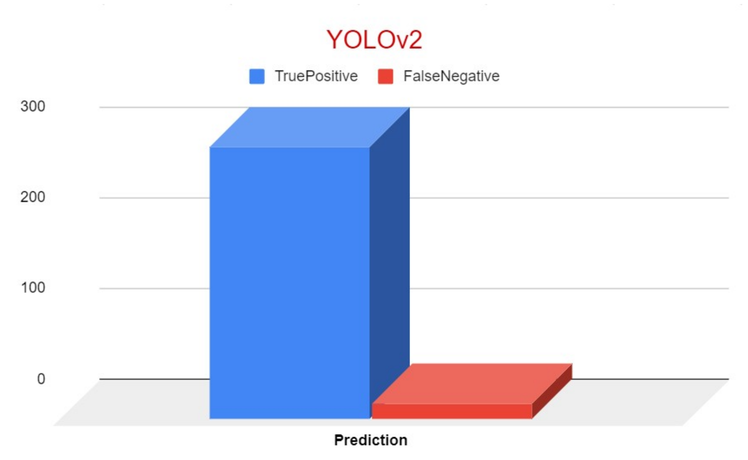


Figure 4.11: True positive and false negative distribution of the YOLOv2 model

True positive and false negative distribution of the YOLOv3 model illustrated in *Figure* 4.12:.

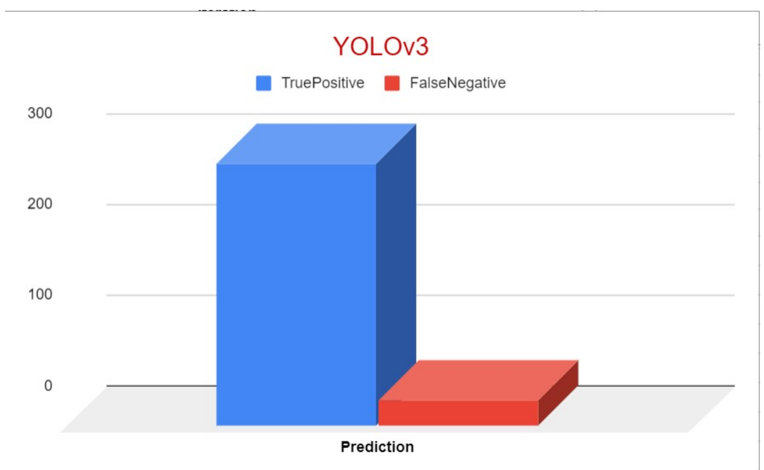


Figure 4.12: True positive and false negative distribution of the YOLOv3 model

4.2.4 YOLO - mAP score

mAP score of the YOLOv2 and YOLOv3 model shown in *Figure* 4.13.

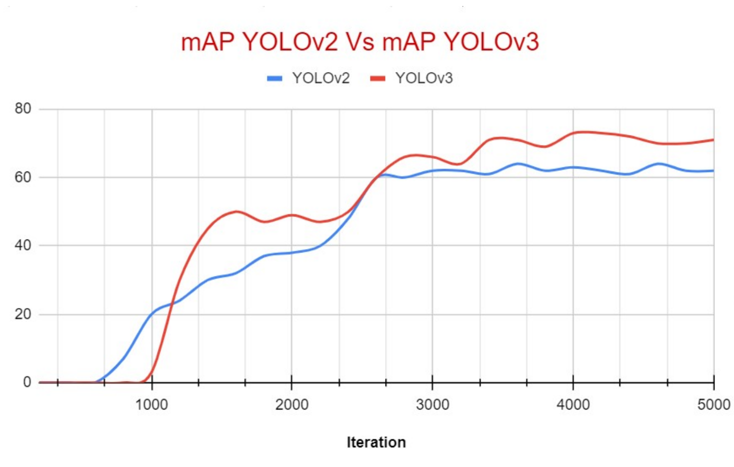


Figure 4.13: Mean average precision for YOLOv2 and YOLOv3 model

Average precision score of the YOLOv2 model as seen in *Figure.* 4.14.

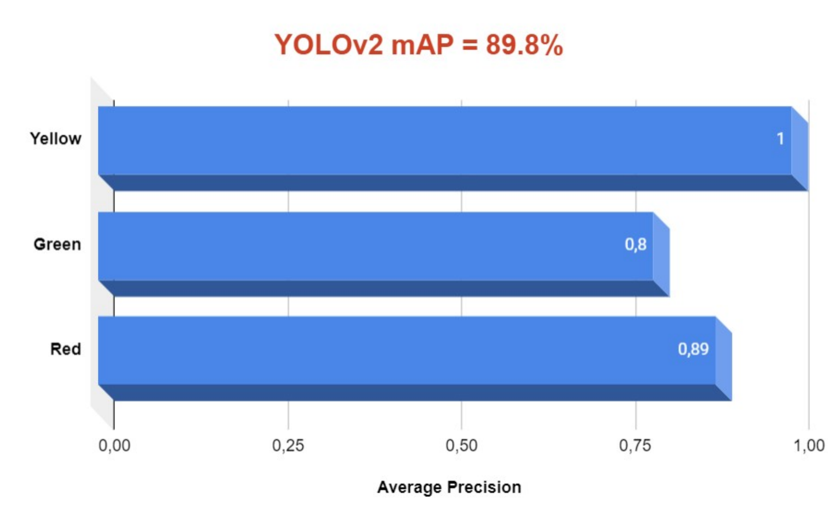


Figure 4.14: YOLOv2 average precision of the three classes

Average precision score of the YOLOv3 model as seen in *Figure* 4.15.

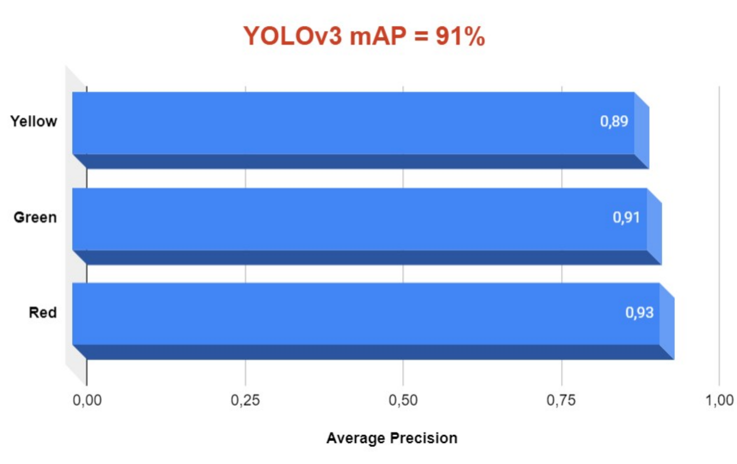


Figure 4.15: YOLOv3 average precision of the three classes

4.3 Navigation

4.3.1 Waypoint navigation

Test was performed by having the vessel set the waypoint 10m ahead of its current direction. Result shown in *Figure* 4.16

Figure 4.16: Accuracy of the waypoint navigation

4.3.2 Hold current position

The vessel was set to hold its position for 300 seconds. Position reading was done in intervals of five seconds. Result shown in *Figure* 4.17

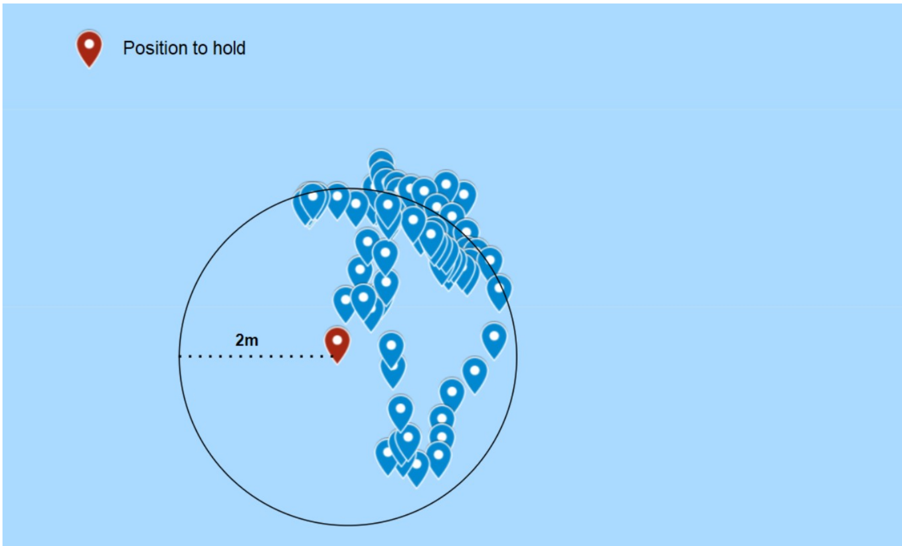


Figure 4.17: Accuracy of the autonomous hold mode

4.3.3 RTL - Return to launch

Path of vessel on its way back to the launch position, illustrated in *Figure.* 4.18

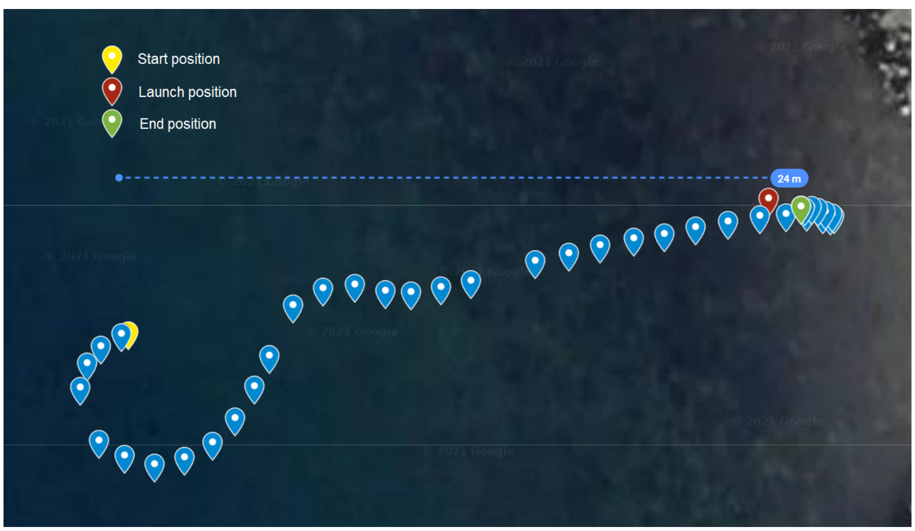


Figure 4.18: Path of the RTL autonomous mode

Time used to return to launch as seen in *Table* 4.4 was:

Table 4.4: Time to complete RTL

Distance	Time (s)
22m	36s

4.4 Stereo Vision

The depth readings is shown in *Figure* 4.19 and the distance error is shown in *Figure* 4.20.

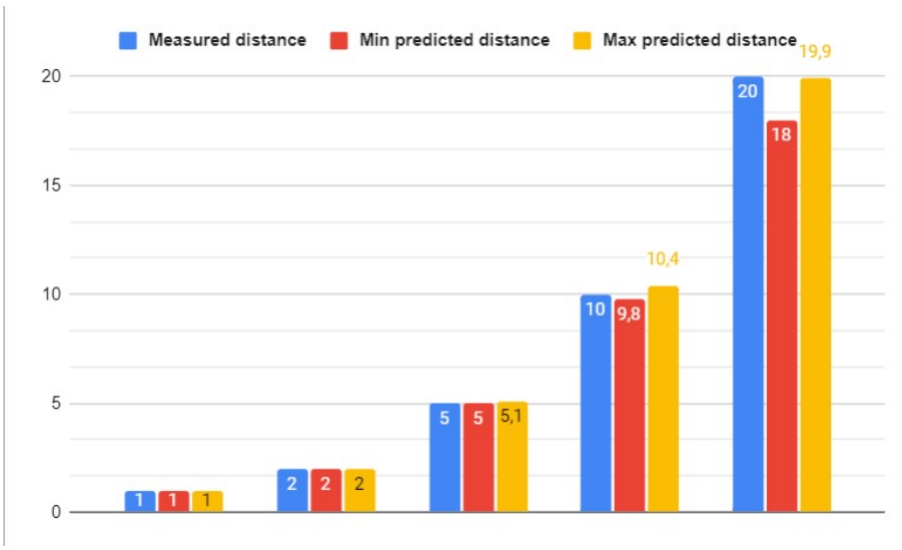


Figure 4.19: Minimum and maximum value calculated from stereo camera

Measured distance	Distance offset
1	0% <-> 0%
2	0% <-> 0%
5	0% <-> 2%
10	-2% <-> -4%
20	-10% <-> -1%

Figure 4.20: Calculated distance offset from measured distance

Chapter 5

Discussion

5.1 Design

The current design of the boat works as an excellent base for future development. It is not optimized for either weight or speed; instead, it is a solid vessel that gives good performance in most areas without outperforming in any of them. Future redesigning of the hull might be considered once the software is at a level where all the courses are completed; as it is in software, we believe improvements are easiest achieved.

5.1.1 Operating time and dimensions

The operating time might differ slightly from our measurements depending on the system's load. The presented results were taken under normal operating conditions. During operation, the vessel never managed to achieve a low battery level. For that reason, we can only say with certainty that the operating time is greater than one hour since we never performed tests longer than this. According to the information provided by the battery charger, we only used 20% of the battery's capacities. We, therefore, feel confident to say that the operative time would remain greater than one hour also during heavy load. For the environment that the vessel was designed for, we do not find it likely that the boat will encounter more demanding conditions than what it encountered already by the shore of Aalesund.

5.2 Maximum speed

The maximum speed was measured to $1.3 m/s$. This gives the boat a theoretical completion time of 46 seconds for the speed course, based on the maximum course length, which according to the competition description, is 60m. It also assumes a constant speed of 1.3 m/s so in reality a slight slower completion time would be expected, considering the vessel has to set a waypoint and find the point to turn. Nevertheless, 1.3 m/s or roughly 2.5 knots is a very satisfactory result. It is below the speed limit for the obstacle course which is set to 4 knots, or 2 m/s. This means that there is a possibility for a speed increase.

We concluded that the maximum speed can not be improved without significant changes, such as redoing the hull or the thruster configuration. In the current design, we believe that there are other aspects that would improve the vessel's overall performance more than an increase in the maximum speed. In the future, it might be reconsidered when the software aspect have been improved.

5.3 Weight and pull force

Based on the measurements from the weight and the pull force, the vessel would have gained 53 points in competition. See *Figure* 2.5. Considering the goal is to maximize points achieved, the aim should be to either maximize the thrust or minimize the weight. With the current setup, the simplest of those two is, in our opinion, to minimize weight since this would not demand any big reconstruction of the vessel. Weight reduction can be achieved with changes such as:

1. Replace acrylic top plate with a lighter material.

2. Replace the metal frame around the hull.

3. Replace metal with carbon.

Together these changes could cut the weight with 1 - 2 kg. This would mean an increase from 53 to 57 – 61 points.

The weight decrease would also affect the vessel's buoyancy, which could affect the boat's speed capabilities slightly.

5.4 GPS accuracy

Throughout the period of measurement, the displacement of location was minimized to a radius of approximately two meters. This is an improvement based on the accuracy stated in the GPS manual, which states an expected error of 2.5 meters. The reason for our 0.5 meter improvement in precision is due to the Kalman filter that we enabled in the Pixhawk. The practical consequences of this extra accuracy is negligible. The GPS accuracy is a critical element when navigating with waypoints as the systems performance depends on the GPS accuracy. The distance between the obstacles the vessel is navigating through is between two and three meters. Therefor the current error of two meters does not gives us reliable estimation to navigate the vessel through obstacles.

When we chose to buy and use the Here 3+ GPS, it was done for two reasons. The first reason was that the old GPS had a far worse accuracy which was not useable. The second reason was the possibility to upgrade the GPS with an RTK component. The reason for not opting to make this upgrade directly was that we wanted to test the GPS module before purchasing a relatively costly module. The RTK module for the here 3+ GPS would provide an accuracy that the current system can not compete against, which according to their manual, is an error of 0.025 meters. This increase would mean that navigation in smaller areas could be based on waypoints if the information about the surrounding domain is sufficient. Without the RTK module, we believe there is minimal optimization that can be done that will affect the current accuracy in any way. This is because the Kalman filter used by the Pixhawk has a reputation for being of high quality.

5.5 YOLO - Detection evaluation

The two YOLOv2 models and the YOLOv3 model all show satisfactory results. Comparing the mAP score of the three models, there seems to be a noticeable increase in performance with the YOLOv3 model. The exposure augmentation that was applied on part of the training set make us confident that the model will function in various light conditions. When comparing against the result from last year seen in *Figure* 5.1, we notice an increase in mAP score for the YOLOv2 and YOLOv3, with 0.5% and 1.7% respectively. All the models were trained on the same dataset as last year's group, only difference being the additional augmented images. In the end we feel confident in claiming that model is more robust and is a improvement from the work last year's group did.

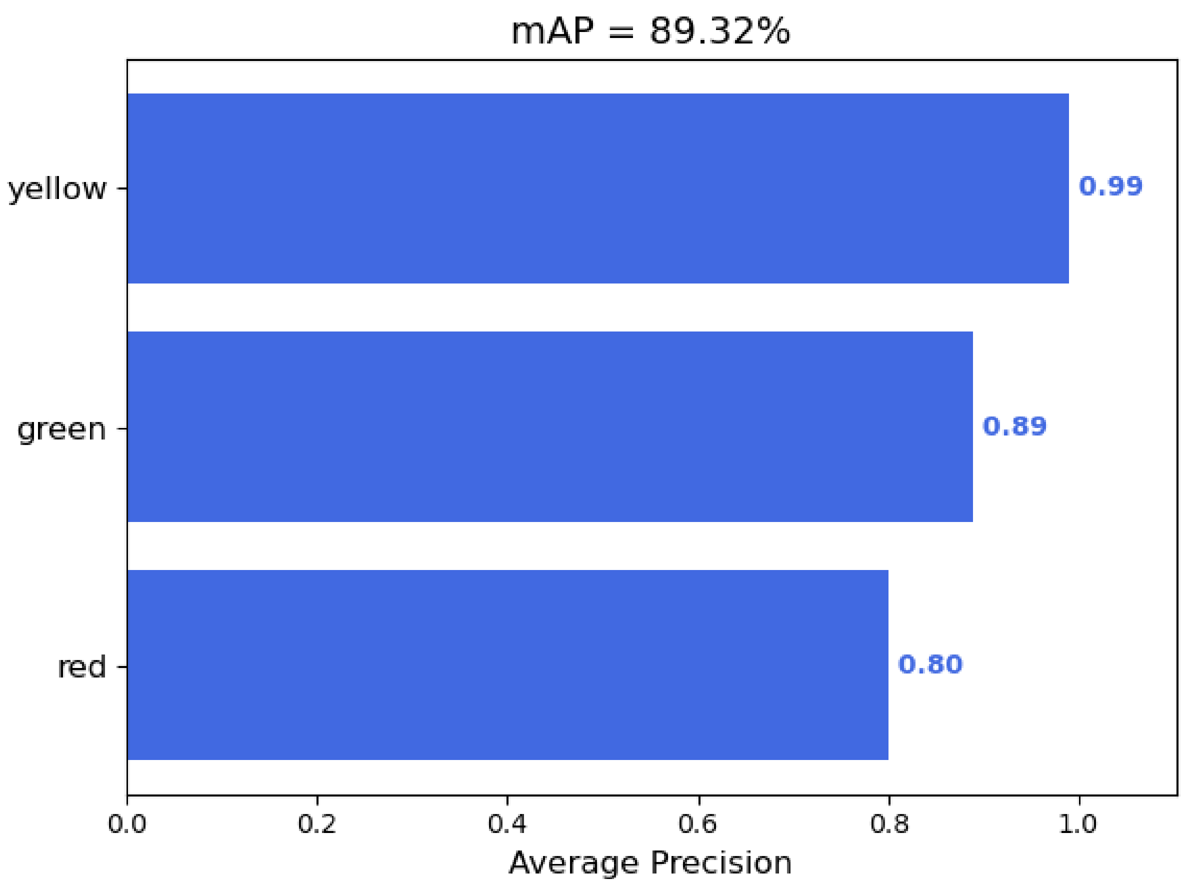


Figure 5.1: mAP score from previous group

The mAP score for our and last year's group is both very high. It should be noted that the model trained for this task is very specialized to tackle the specific problem of the Autodrone competition. By simply comparing our mAP score to other more standardized models, an mAP score of close to 90% will seem very high. The difference is that these other models cover a wider specter of objects comparable to our specialized buoy model. With the models trained in this project, a reoccurring problem is the similarities in the training and testing dataset. This means that the mAP score of our and last year's model is artificially high due to the model being tested on images that are almost identical to the training images.

To get a better model, we would recommend increasing the current dataset with additional images with more unique backgrounds and distances. We believe it is possible to achieve a better model by increasing the dataset, although it might not manage to achieve a better mAP score of 91%. How big or small these additional images would improve the model would need to be tested to figure out. We also want to mention that there is a limit in how different the images should be since we do not want to lose the model's specialty to detect buoys in the ocean.

5.6 YOLO - Localization evaluation

The intersect over union (IoU) score, was satisfactory for both the YOLOv2 and YOLOv3 model. With YOLOv2 reaching a score of 60% and the YOLOv3 a score of 75%. Normally a result greater then 50% is very good. These results builds trust in the performance of our models. The IoU score from both the models can be a result of the similarities of background in the images. We would recommend to increase the variations in the images that the model is trained on. This would give the benefit of a more robust model even if it might affect the score on the different metrics. It is a worry that the current model's metrics is artificially high due to the similarities in the training and testing dataset.

5.7 Stereo, accuracy over multiple distances

The result shows the calculated depth becomes increasingly inaccurate over longer distances, most notable when the distance is greater than $10m$. When the distance was less than $10m$ the error of measurements was between 0 - 4%; when less than $5m$, the error of measurement fell to between 0 - 2%. Meaning all measurements below $10m$ can be trusted to be accurate to a high degree. The inaccuracy for distances above $10m$ have minimum effect on the system. When an distance to the object is larger than $10m$, it is too far away to affect our immediate path. We believe that avoidance action only has to be considered when a object is closer than 3 meters.

The measurements are more accurate for shorter distances and therefore suitable to use in the docking course if the waypoint accuracy is sufficient. A disadvantage with the current setup is that the camera is stationary, meaning distance readings from the sides are not possible. This is valuable information that we are currently not gathering. We would recommend mounting two distance sensors on the side to solve this. We would have opted for a 45-degree tilt to detect a more significant number of objects that are in our forward path.

5.8 Waypoint navigation

An essential part of any autonomous vessel is its ability to navigate. The results show that our design utilizing the autonomous features of the Pixhawk has the potential to solve the competition courses.

5.8.1 Navigation - custom waypoint

The tests we performed show that the vessel can set a waypoint and navigate towards it. The path the vessel chose towards the set waypoint was relatively inefficient initially. We believe this is due to magnetic interference from the batteries affecting the Pixhawk's compass. To solve this, it could be mounted a shield around the Pixhawk, blocking most of the magnetic interference. Another solution to improve the results is by upgrading the GPS with an RTK module.

We do not utilize any external input to determine the distance that the vessel sets the waypoint in the current solution. However, we have taken into account that this should be doable in the future, and a custom distance can easily be set by adjusting one variable as explained in *Section* 3.6.4. If the upgrade to an RTK component is done, it would be possible to navigate to the dock in the docking course using the waypoint system and the stereo camera depth readings. We feel that the results presented show that the waypoint navigation can be the backbone to solve many of the competition courses in the future.

5.8.2 Waypoint - Hold mode

The results in *Section* 4.17 show that it would not be possible to use this feature alone to maintain the position next to the dock. The vessel's movement goes from 0 - 2m from the starting position. However, the precision of this method depends on the accuracy of the GPS and compared to the result from the GPS accuracy test, illustrated in *Figure* 4.6 the error is similar. This means that a possible solution to fix the inaccuracy would be to upgrade the current GPS module with its accompanied RTK module.

During testing, it was slightly windy, with winds up to 8 m/s, according to weather services. The wind affects the vessel's ability to hold its position to a limited degree. This is because of how the HOLD function works. When the vessel is told to hold its position, it sets a waypoint where it is currently located and then holds this position with a two meters accuracy, which is what our GPS provides. This means it only starts to regulate its position once the GPS tells the computer that it is outside the ring of two meters. In total, we can confidently say that the result showcases that the boat can hold its position within the limitations that the current hardware includes. With the RTK component, the HOLD accuracy would increase its accuracy drastically.

5.8.3 Waypoint - Return to launch

The results from the RTL mode show that when activated, the vessel accelerates and moves toward the launch point. An unexpected element was the path at the beginning of the RTL feature. During the initial moment after the RTL was activated, the vessel moves rather inefficiently towards the goal. The inefficiency can be seen in *Figure* 4.17 by comparing the first half with the second half of the figure. It takes a turn with a relatively large radius and a slightly smaller turn to adjust for the overshoot before moving toward the launch point in a straight line. It slightly overshoots the target position but adjusts to being within the launch position. We believe this is due to the magnetic interference from the batteries, as in the waypoint test mentioned in *Section* 5.8.1

The vessel used to travel the 22 meters was 36 seconds; this equals to an average speed of 0,6m/s. This is significantly slower than the theoretical completion time of 17 seconds that a speed of 1,3m/s would result in.

Chapter 6

Conclusion

When considering the problem formulation mentioned in *Section* 1.2, we are satisfied with our vessel's performance. We wish that we would have gotten further with certain aspects, such as software development and combining the developed systems. At the moment, we feel that progress is achieved at an increased rate. Throughout the project, the group has worked to have an operative vessel that meets the competition standards mentioned at the beginning of the report. The results that were presented affirm that this has been achieved. The part of our project that involved developing buoy detection and autonomous navigation are today two independent functioning systems. The detection system detects the correct buoy, and the navigation system has the basis for autonomous navigation.

The object detection system scores high on all metrics. With mAP score of 90% it might be artificially high, which can be explained by the relatively low difference between images in the dataset, even with data augmentation. To expand the dataset with new images should be a priority for future groups.

The navigation system has the fundamentals set, such as navigation through waypoints, holding the drone's position for a specified time and return to the launch. The navigation system works with expected precision due to our GPS module. The GPS was our most important upgrade throughout the project, and represents our thought when selecting parts. We wanted to create a vessel with modular solutions throughout the project that would make simplify further improvements. The selected GPS has a RTK module that would increase the overall accuracy of the waypoint navigation, as it is heavily dependent on the GPS. Additional results such as max speed of 1.2 m/s is the reason for why we believe that the vessel can be competitive in the Autodrone competition not only this year but also in the future years.

Finally, we would like to repeat that we feel the boat works as an excellent base for future development. It is not optimized for either weight or speed; instead, it is a solid vessel that gives good performance in most areas without outperforming in any of them. The project turned out to be a challenge, with a grand scope of new material for the group to learn. In addition, the covid-19 situation was demanding both mentally and physically, but also made working in a group harder than usual. We believe the work that has been done provides a sound basis for further development by next year's group, and we hope that the vessel can be part of the Autodrone competition for many years to come.

6.1 Further work

The hardware requirements have been implemented, meaning that the priority for future work should be to increase the drone's autonomous solutions and improve flaws that we detected. The group considers the following task as the objectives to grant the most significant improvements:

- Combine navigation and object detection system.
- Upgrade GPS with the RTK module.
- Extend the vessel with additional proximity sensors.
- Shield the Pixhawk from magnetic interference.
- Continue software development.
- Increase the image dataset.

Bibliography

[1] How a kalman filter works, in pictures. URL https://www.bzarg.com/p/how-a-kalman-filter-works-in-pictures/#mathybits.

[2] Autodrone. Rules and task descriptions. URL https://drive.google.com/file/d/1znsjKdCcPNmfPysDiKEFOYOPmX390RNk/view.

[3] T. Editors of Encyclopaedia Britannica. Automatic pilot. *Encyclopedia Britannica*, 2008.

[4] Formlabs. *Material data sheet, standard.* Formlabs, 4 2017. URL https://formlabs-media.formlabs.com/datasheets/1801089-TDS-ENUS-0P.pdf. Formlabs standard resin data sheet.

[5] Formlabs. *Form 3, Low Force Stereolithography (LFS) 3D Printer.* Formlabs, 9 2019. URL https://media.formlabs.com/m/5a3f3cadc639b857/original/-ENUS-Form-3-Manual.pdf. Form 3 manual.

[6] Prakhar Ganesh. Object detection : Simplified. URL https://towardsdatascience.com/object-detection-simplified-e07aa3830954.

[7] Blue Robotics Inc. T200 thruster. URL https://bluerobotics.com/store/thrusters/t100-t200-thrusters/t200-thruster-r2-rp/.

[8] Rudolph Emil Kalman. A new approach to linear filtering and prediction problems. *Transactions of the ASME–Journal of Basic Engineering*, 82(Series D):35–45, 1960.

[9] North Group LTD. Gps, gnss and geodesy concepts. URL https://northsurveying.com/index.php/soporte/gnss-and-geodesy-concepts#chapter-1-from-stones-to-satellites.

[10] R. Paz-Vicente A. Linares-Barranco M. Domínguez-Morales, A. Jiménez-Fernández and G. Jiménez-Moreno. Stereo matching: From the basis to neuromorphic engineering, current advancements in stereo vision. *Spanish Science and Education Ministry Research Projects*, (1):10, 11 July 2012. doi: https://www.intechopen.com/books/current-advancements-in-stereo-vision/stereo-matching-from-the-basis-to-neuromorphic-engineering.

[11] Royal Observatory og Belgium. How gnss works. URL http://gnss.be/how_tutorial.php.

[12] Joseph Redmon. You only look once: Unified, real-time object detection. URL https://arxiv.org/pdf/1506.02640.pdf.

[13] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. *arXiv*, 2018.

[14] Khoshgoftaar T.M. Shorten, C. A survey on image data augmentation for deep learning. (1):60, 2019. doi: https://doi.org/10.1186/s40537-019-0197-0.

[15] PX4 Dev Team. Using the ecl ekf. URL https://docs.px4.io/v1.9.0/en/advanced_config/tuning_the_ecl_ekf.html.

[16] Alkaios Bournias Varotsis. Introduction to fdm 3d printing. 2019. URL https://www.3dhubs.com/knowledge-base/introduction-fdm-3d-printing/.

[17] Alkaios Bournias Varotsis. Introduction to sla 3d printing. 2019. URL https://www.3dhubs.com/knowledge-base/introduction-sla-3d-printing/.

Appendices

Appendix A

Preproject report

FORPROSJEKT - RAPPORT

FOR BACHELOROPPGAVE

TITTEL: USV Sjødrone				
KANDIDATNUMMER(E):				
DATO: **28.01.2021**	EMNEKODE: **IE303612**	EMNE: **Bacheloroppgave**		DOKUMENT TILGANG: - Åpen
STUDIUM: **AUTOMASJON**			ANT SIDER/VEDLEGG: /	BIBL. NR: - Ikke i bruk -
OPPDRAGSGIVER(E)/VEILEDER(E): Anete Vagale, Øystein Bjelland, Robin T. Bye, Ottar L. Olsen				

OPPGAVE/SAMMENDRAG:

For å få fullført en bachlorgrad ved NTNU Aalesund kreves det en utførelse av en 20 poengs bachloroppgave i sjette semester. Vi har valgt oppgaven om å videreutvikle ett ubemannet overflate skip (USV) også kalt sjødrone som skal delta i en nasjonal konkurranse mellom universiteter og høyskoler. Denne oppgaven ble valgt da den virket spennende og utfordrende. Oppgaven la til rette for at vi kan benytte oss av læringen vi har hatt gjennom de foregående semestrene.

Formålet med oppgaven er å lage en autonom båt som kan løse fire hinderløyper. Dette er tenkt å løses ved å benytte moderne metoder for bildegjenkjenning og beslutningstaking.

Postadresse
Høgskolen i Ålesund
N-6025 Ålesund
Norway

Besøksadresse
Larsgårdsvegen 2

Internett
www.hials.no

Telefon
70 16 12 00

Epostadresse
postmottak@hials.no

Telefax
70 16 13 00

Bankkonto
7694 05 00636

Foretaksregisteret
NO 971 572 140

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

INNHOLD

TABELLER

FIGURER

1 INNLEDNING

Vi er tre automasjon studenter som skal benytte det vi har lært under studium til å lage en autonom båt. Automasjon er et fremtids rettet felt som i dag brukes i flere industrier for å effektivisere produksjoner. Du finner eksempler på automasjon i alt fra jordbruk til romforskning.
Bachelor oppgave går ut på å bygge en automatisert båt for Autodrone. Autodrone er en nasjonal konkurranse for autonome sjødroner mellom norske Universiteter og Høgskoler. Studenter fra hele Norge designer og lager automatiserte båter som skal navigere seg gjennom fire ulike løyper. Båtene blir dermed tildelt poeng basert på gjennomførelsen. For å løse oppgaven er det planlagt å benytte oss av teknikker som lært i fagene intelligente systemer, sanntids datateknikk, datakommunikasjon og industrielle styresystemer. Med den kunnskap vi har fra disse og flere emner og veilederne våre har vi grunnlaget for å konstruere en båt som skal kunne oppnå kravene for konkurransen.

NTNU Aalesund har et annet student lag (Norwegian Automatic Vehicle Operations – NAVO), som ble opprettet våren 2020. Dette laget skulle konkurrere i den internasjonale konkurransen Roboboat, som kan sies å være inspirasjonen til den nasjonale Autodrone konkurransen. NAVO NTNU vil gi oss ekstra veiledning og er en stor ressurs for oss.

2 BEGREPER

- YOLO - You Only Look Once
- Darknet – Rammeverk for neural network lagd i C
- NMEA 0183 – En standard kommunikasjon mellom marin elektronikk
- ASD – Autonomus sea drone
- LiDAR – Light detection and Range
- IP – Internet protocol
- OCS – Operator control station
- DHCP – Dynamic host configuration protocol
- GPS – Global positioning system

3 PROSJEKTORGANISASJON

3.1 Prosjektgruppe

Studentnummer
509364 – Kaung Htet San
485469 – Markus Grorud Gaasholt
488568 – Magnus Stava

Tabell 1: Studentnummer for alle i gruppen som leverer oppgaven for bedømmelse i faget IE303612

3.1.1 Oppgaver for prosjektgruppen – organisering

Følgende er en grov oversikt over oppgave organiseringen som prosjektet kan deles inn i.

- Konstruksjon av skrog.
- Implementering av elektronikk
- Manuell kontroll av båt.
- Objekt deteksjon.
- Identifisere objekter som rød, grønn og gul bøye eller båt.
- Lokasjon av objekter.
- Distanse til objekter

- GPS lokasjon
- Prediksjon av bevegelse bestemt av kamera + sensor + gps.
- Levere GPS informasjon gjennom NMEA 0183 protokoll.

Fremstilt i risikomatrisen er tre punkter/tema som er relevante med tanke på bachelor oppgaven.

Se 5.6 for fullstendig oversikt.

3.1.2 Oppgaver for prosjektleder

-Ansvarsområde

Planlegge prosjekt.
Opprettholde god gruppedynamikk.
Sørge for at prosjektet blir gjennomført.
Evaluere prosjektet.

-Arbeidsoppgaver

Holde seg oppdatert på gruppens fremgang i oppgaven.
Kontakte veiledere ved assistanse/spørsmål som de kan bistå med.
Sende ut varsel om møter til gruppemedlemmer.

3.1.3 Oppgaver for sekretær

Ansvarsområde

Sette opp møter.
Bistå prosjektleder med møter, skrive referat.

Arbeidsoppgaver

Skrive møtereferat.
Loggføring.

3.1.4 Oppgaver for øvrige medlem(mer)

-Ansvarsområder

- Følge timeplanen satt opp av gruppa. hei
- Gi full innsats for å fullføre oppgaver innenfor tiden som er satt av.
- Følge de normene og reglene fastsatt av gruppen.

Arbeidsoppgaver-

- Se gantt diagram på 5.6.

3.2 Styringsgruppe

Styringsgruppen består av Anete Vagale, Øystein Bjelland, Robin T. Bye og Ottar L. Olsen.

Det foreligger også tilbud for bistand fra en gruppe med master studenter som har et lignende prosjekt. Denne gruppen vil bli referert til som NAVO NTNU og vil bli benyttet som en referansegruppe.

4 AVTALER

4.1 Arbeidssted og ressurser

Prosjektet vil bli utført på NTNU Ålesunds lokaler, hovedsakelig lab bygget. Her har vi tilgang til verktøy for å utføre hardware delen samt store pc-skjermer som hjelper til i software utvikling.

Av personal har vi veilederne og

ressurspersonene som er oppført. Utenom disse har vi tilgang på bistand fra lab personalet samt fra NAVO NTNU gruppen som har tilbudt sin assistanse hvis det er behov.

Ved stengning av laboratorier pga. FHIs covid-19 anbefalinger vil vi benytte oss av hjemmekontor.

4.2 Gruppenormer – samarbeidsregler – holdninger

- Minimum et muntlig møte ukentlig for å kunne diskutere problemstillinger som har oppstått og for å skaffe overblikk over fremgangen.
- En gruppesamtale skal benyttes for å unngå å kaste bort tid ved å prøve å løse problemer alene.
- Ved uenigheter skal det ikke benyttes usaklige argumenter. Gruppemedlemmene skal også være mottakelig for konstruktiv kritikk.
- Det skal opprettholdes kommunikasjon mellom gruppemedlemmer.
- Ved fravær skal dette meldes ifra minimum 5 dager før. Ved akutt sykdom skal det gis beskjed så tidlig som mulig.

Som en ferdig utdannet automasjonsingeniør, åpner det seg utallige dører for hvor veien videre skal gå. Avhengig av hvor, er det noen generelle grunnpilarer som står fast, dette er holdninger som har grunnlag i personlighet fremfor yrke. Noe har også grunnlag i den situasjonen verden står ovenfor, med et økende fokus på klimapolitikk. Som ingeniør kan man velge i hvilken grad man ønsker å forholde seg til dette.

Det er definitivt at vi skal ha problemer og utfordringer gjennom hele prosjekt periode. Vi skal løse problemer og utfordringer møtes vi gjennom prosjekt med stor entusiasme. Problem løsning er hoved del av ingeniør profesjon. Med utviklingen av nye teknologier er automasjon ingeniør arbeid er alltid i utvikling. Vi skal møte utfordringer vi har aldri møt før i arbeids liv. Vi har lært å løse problemer ved bruk av kunnskap vi har lært gjennom studie og er i konstant læring.

5 PROSJEKTBESKRIVELSE

5.1 Problemstilling - målsetting - hensikt

Problemstillingen er å lage en autonom båt som klarer å løse de fire hinderløypene som er oppgitt i spesifikasjonene til konkurransen. Båten skal detektere hindringer, basert på avstand fra båt til hindring og bestemmer den beste retningen for båten å ta. Målet er at båten kunne ta den samme handlingen fra begynnelse til slutt.

Ved et vellykket prosjekt burde vi ha klart å løse de fleste av hindrene. Vi har som målsetting å kunne utføre samtlige av hindrene. Dette er noe vi mener er innenfor rekkevidde.

Verdimål

- Prosjektet skal lært oss om samarbeid, problemløsning og innsikt innen hvordan automasjon kunnskap løser hverdagsutfordringer og får innsikt i hverdagen arbeid til en ingeniør
- .

Effektmål

- Kunnskap
- Ferdigheter
- Generell kompetanse

Resultatmål

- Målet med prosjektet er å lage en fungerende autonom båt som er innenfor de spesifikasjoner som er gitt av konkurransen. Båten skal også kunne gjennomføre de fire løypene lagd av konkurranse arrangøren.

5.2 Krav til løsning eller prosjektresultat – spesifikasjon

Oppgaven inkluderer flere spesifikasjoner som krav for å kunne delta i konkurransen.

- Båten kan ikke overskride en vekt på 70kg.
- Båten kan ikke overskride dimensjonene 0.9m x 0.9m x 1.8m.
- Båten må være fult autonomisert, med mulighet for manuell styring.
- Batteri drevet samt ha en kill switch montert på båten og en på kontrollen.
- Båten må ha muligheten for å bli tauet.

- For å bli tildelt poeng må OCS(operator control station) ha muligheten for å koble til et kablet Ethernet nett og spør om en IP adresse fra DHCP server. Etter å ha tilkoblet skal vi levere GPS data gjennom NMEA 0183 protokol.
- Båten må ha et visuelt feedback system.

Basert på NAVO NTNU's estimater vil total verdi av prosjektet havne på mellom 35 000 – 45 000. Dette estimatet har vi kommet frem til etter samtaler med NAVO NTNU. Finansieringen av komponenter som vi ikke har tilgang på utføres av NTNU Aalesund. Viktig å presisere her at noe av komponenter kan vi benytte oss av

Båten skal for å kunne kalle prosjektet fullført klare å fullføre halvparten av hinderløypene fra konkurransen. Dette inkluderer også at konkurransens spesifikasjoner har blitt nådd.

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

Konkurranses kan deles opp i 4 deler. Oppdage hindringer, unngå hindringer, planlegge fremtidig vei og, legge til kai.
For å kunne løse problemet ved bilde deteksjon, identifikasjon og lokasjon er det planlagt å benytte YOLO, you only look once. YOLO er et deteksjons algoritme som har sin store fordel for å være ekstremt raskt.

Ettersom vi i denne oppgaven hovedsakelig er interessert i fire ulike objekter, rød, grønn og gul bøye samt møtende båter lager vi vårt eget dataset ved hjelp av www.roboflow.com. Ved å lage vårt eget dataset får vi også muligheten til å trene modellen på bilder som er tatt av samme kamera som vi skal benytte os av. Om dette vil ha noen form for påvirkning på resultatet er uvisst.

Vi skal bruke erfaring fra kunstig intelligens faget samt tilgjengelig informasjon fra nettet om YOLO for å implementere den autonome delen i dronen. For å kunne ta avgjørelser om fremtidige bevegelser benytter vi oss av random forest, som er en ensamble teknikk som vil si at den benytter seg av flere svake modeller for å generere en "sterk" modell. Disse modellene er en rekke true/false påstander som, når satt sammen blir til en avansert/sterk modell. Planen er at via outputen vi får fra YOLO skal vi kunne få båten til å ta en avgjørelse basert på disse påstandene.

Det vurderes også om vi får tilgang å benytte oss av en LiDAR for ekstra input om omgivelsene.

5.4 Informasjonsinnsamling – utført og planlagt

NAVO NTNU gruppen ga oss datablad om utstyret de benytter, dette ble brukt for inspirasjon. Vi har blitt oppmerksomme på anvendelsen av LiDAR i forbindelse med ASD. Dette er noe vi vurderer å benytte men, da LiDAR utstyr tilgjengelig på universitetet skanner rommet i 2D fremfor 3D må vi undersøke videre hvilke konsekvenser dette vil ha.

Videre skal vi gå igjennom rapporten til forrige års bachelor gruppe som hadde en lignende oppgave for å se på erfaringene de tilegnet seg. Denne rapporten vil gi oss et bedre overblikk på oppgaven vi skal utføre.
Fra Roboboat konkurransen som arrangeres i USA og som Autodrone konkurransen er inspirert av eksisterer det flere filmer og rapporter om andres grupper erfaring. Likhet med rapporten fra forrige års bachelor gruppe er disse rapportene også god informasjon om hva som fungerte og ikke fungerte.
Informasjonsmøte med NAVO team er noe vi har gjennomført og noe som vi planlegger å fortsette med, da dette er en god bonus ressurs for oss. Videre har vi god assistanse fra delegerte veilederne.

5.5 Vurdering – analyse av risiko

Vår vurdering for å realisere prosjektet på et godt nivå innenfor den tidsrammen som er gitt er god. Vi har en arbeidsplan hvor vi har definert prioriteringsnivå på de ulike oppgavene. Vi ser foreløpig ingen grunn til å avgrense prosjektet videre utenom konkurransereglene.

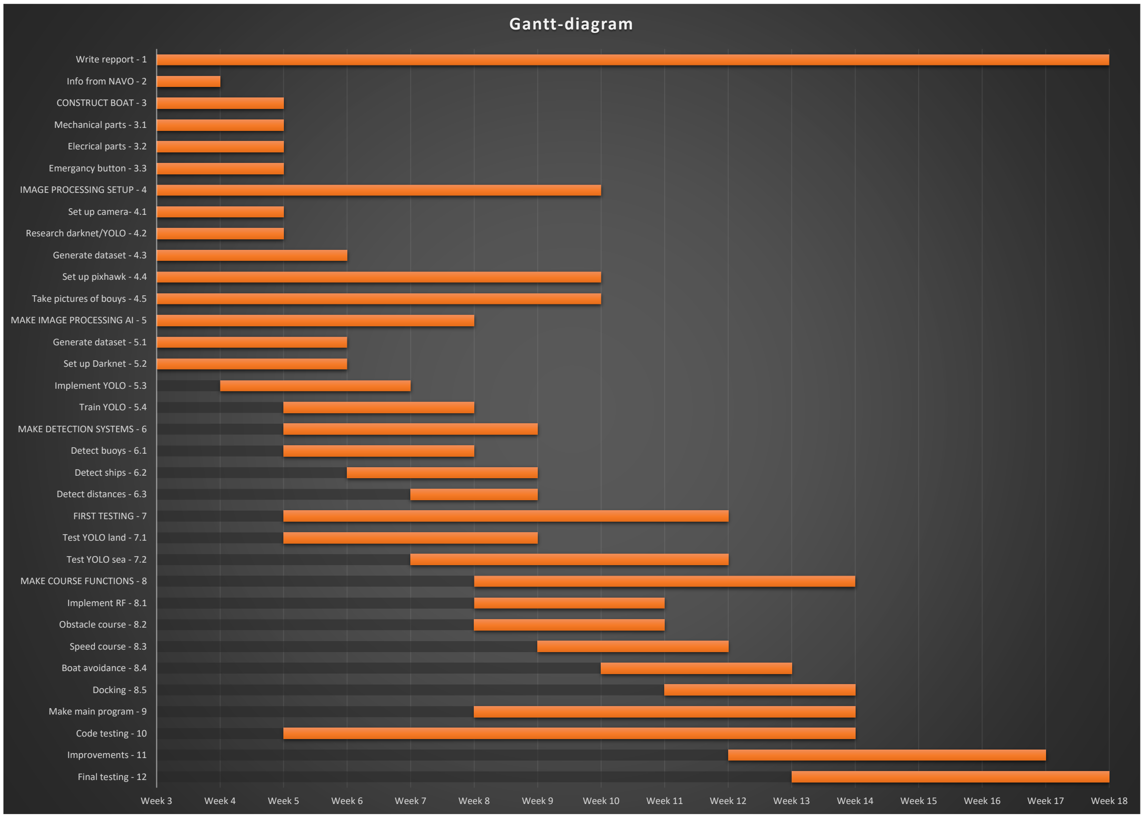
Særlig viktig komponent for å lykkes er å få en god start samt at gruppen leser seg opp på temaer de er mindre komfortable med. Ved en situasjon hvor medlemmer ikke har lest seg opp tilstrekkelig om temaet kan dette føre til en overbelastning av arbeidsmengde for enkelte medlemmer som vil forsinke hele fremgangen. Dette vil også føre til misnøye innad i gruppen som oftest ikke fører til økt fremgang.

Annen risiko er muligheten for stengte lokaler i forbindelse med coronaviruset. Akkurat nå er det uvisse tider vi lever i. Nye beskjeder kan dukke opp på noen få timers varsel som endrer hele situasjonen. Ved en full nedstengning av arbeidslokaler vil dette føre til økt ansvar blant gruppemedlemmene for å opprettholde arbeidsmengden. Dette kan være utfordrende med tanke på den psykiske påvirkningen coronaviruset påfører individer. Dette og flere konsekvenser av en sosial nedstengning på grunn av coronaviruset fører til at dette er en av de største risikofaktorene for dette prosjektet.

Hendelse / Situasjon	Aktuelt	Sannsynlig	Konsekvens	Risiko	Kommentar / Tiltak
Covid-19 restriksjoner	Ja	Middels sannsynlig	Betydelig	Høy	Covid-19 har preget forrige år og vi anser det som relativt sannsynlig at det vil fortsette og prege samfunnet i tiden fremover. Tiltak blir hjemmekontor
Karantene/Isolasjon	Ja	Lite Sannsynlig	Betydlig	Middels	Vi har frem til nå hatt lite smitte innad i NTNU Ålesund og med begrenset sosial omgangskrets grunnet nåværende restriksjoner, ser vi det som lite sannsynlig. Tiltak blir hjemmekontor på hele gruppen.
Ikke fullført Bachlor	Ja	Veldig usannsynlig	Stor	Lav	Vi ser det som veldig usannsylig å ikke fullføre bachlor oppgaven sett utifra de kravene vi har satt i gruppen. Disse gir et godt utgangspunkt.

5.6 Hovedaktiviteter i videre arbeid

Refererer til gantt diagram for oversikt over planlagt arbeid.



Figur 1: Ganttdiagram som viser tidslinje av arbeidsoppgaver

5.7 Framdriftsplan – styring av prosjektet

5.7.1 Hovedplan

En uoffisiell prosjektstart ble satt til 11.januar hvor gruppen satte seg ned for å diskutere forventninger til oppgaven samt legge en plan for ukene fremover. Dette inkluderer datoer for gruppemøter samt arbeidsfordeling. I den første uken setter alle seg grovt inn i delene som prosjektet består av. Videre vil hvert gruppemedlem som har hovedansvaret for en del presentere fremgangen på gruppemøtene. Dette gjør det enkelt for gruppen å identifisere hvordan utviklingen i hver del går fremover.

Etter at båten er konstruert er planen å ha få den på vannet raskest mulig for å kunne utføre ukentlig testing. Disse testene er planlagt til å foregå på Søndager.

Fullstendig liste av aktiviteter og hovedaktiviteter kan ses på 5.6.

For arbeidsfordeling se bilde under.

Magnus Stava = MS, Markus Gaasholt = MG, Kaung San = KS								
Tasks	**Taskname**	**Kickoff**	**Duration**	**Responsibitily**	**Assistance**	**Completed**	**Cost**	**Completed by**
Write report	Write repport - 1	3	15	All		0 %	kr -	
Meeting with NAVO NTNU group	Info from NAVO - 2	3	1	All		60 %	kr -	
Construction of boat	CONSTRUCT BOAT - 3	3	2			50 %	kr 40 000,00	
Boat mechanical construction	Mechanical parts - 3.1	3	2	KS	MS	90 %	kr -	
Boat electrical construction	Elecrical parts - 3.2	3	2	KS	MG	10 %	kr -	
Add emergency button	Emergancy button - 3.3	3	2	KS	MG	0 %	kr -	
Image processing	IMAGE PROCESSING SETUP - 4	3	7			0 %	kr -	
Set up Pixhawk	Set up camera- 4.1	3	2	MG	MG/MS	0 %	$ 195,07	
Research darknet/YOLO	Research darknet/YOLO - 4.2	3	2	MG	KS	20 %	kr -	
Research random Forest	Generate dataset - 4.3	3	3	MG	KS	0 %	kr -	
Set up camera	Set up pixhawk - 4.4	3	7	KS	KS	0 %	kr -	
Take pictures for dataset	Take pictures of bouys - 4.5	3	7	MG	MS	0 %	kr 209,70	
making the AI	MAKE IMAGE PROCESSING AI - 5	3	5			0 %	kr -	
Generate dataset in roboflow	Generate dataset - 5.1	3	3	MG	KS	0 %	kr -	
Set up Darknet framwork	Set up Darknet - 5.2	3	3	MS	MG	0 %	kr -	
Implement YOLO	Implement YOLO - 5.3	4	3	MG	MS	0 %	kr -	
Train YOLO on custom dataset	Train YOLO - 5.4	5	3	MG	MS	0 %	kr -	
Detection of obstacles	MAKE DETECTION SYSTEMS - 6	5	4			0 %	kr -	
Detect boyes (green,red,yellow)	Detect buoys - 6.1	5	3	MS	MG	0 %	kr -	
Detect ship	Detect ships - 6.2	6	3	MG	MS	0 %	kr -	
Detect distance to obstacle	Detect distances - 6.3	7	2	MS	MG	0 %	kr -	
First testing of complete boat	FIRST TESTING - 7	5	7			0 %	kr -	
Test YOLO on land	Test YOLO land - 7.1	5	4	All		0 %	kr -	
Test YOLO on sea	Test YOLO sea - 7.2	7	5	All		0 %	kr -	
Make functons	MAKE COURSE FUNCTIONS - 8	8	6			0 %	kr -	
Implement random forest/decision tree	Implement RF - 8.1	8	3	KS	MS	0 %	kr -	
Implement statements for obstacle course	Obstacle course - 8.2	8	3	MS	KS	0 %	kr -	
Implement statements for speed course	Speed course - 8.3	9	3	KS	MS	0 %	kr -	
Implement statements for boat avoidance	Boat avoidance - 8.4	10	3	MS	KS	0 %	kr -	
Implement statements for docking	Docking - 8.5	11	3	KS	MS	0 %	kr -	
Make a main program containing all functions	Make main program - 9	8	6	MS	KS/MG	0 %	kr -	
General testing	Code testing - 10	5	9	All		0 %	kr -	
Improvements	Improvements - 11	12	5	All		0 %	kr -	
Testing/Troubleshooting	Final testing - 12	13	5	All		0 %	kr -	

Figur 2: Oversikt over arbeidsoppgavene og deres detaljer

5.7.2 Styringshjelpemidler

Benytter oss av Microsoft Teams som er en programvare for grupper å jobbe samtidig på de samme dokumentene.
Bachelor rapporten vil bli utformet via Overleaf, en online kompilator for LaTex.
Benytter oss av Gantt-diagram lagd i teams gruppen for å holde oversikt over fremgangen på prosjektet

5.7.3 Utviklingshjelpemidler

- Microsoft Teams
- Overleaf
- Darknet rammeverk
- Gantt diagram
- Python

5.7.4 Intern kontroll – evaluering

Benytter oss av Gantt diagrammet for å følge med på fremgangen ved hvert enkelte mål. Nåværende status tas opp ved gruppas ukentlige interne møte.

5.8 Beslutninger – beslutningsprosess

Oppgaven fremsto som veldig klar da konkurransen ga oss presise og definerte mål som hver deltaker må kunne løse for å kunne prestere godt. Metoder til å oppnå ønsket mål blir diskutert på møte og de som er ansvarlige for å oppnå spesifikt mål bestemmer hvilken metoder de skal bruke.
For hovedprosjekt skal vi evaluere framgang og diskutere videre om prosess og arbeidsmengde til enkelt individual.

6 DOKUMENTASJON

6.1 Rapporter og tekniske dokumenter

- VantTec Technical Design Report 2019 for roboboat.
- Autodrone konkurranseregler
  https://drive.google.com/open?id=1znsjKdCcPNmfPysDiKEFOY0PmX390RNk
- Darknet: Open Source Neural Networks in C
  https://pjreddie.com/darknet/

7 PLANLAGTE MØTER OG RAPPORTER

7.1 Møter

7.1.1 Møter med styringsgruppen

Mer info om dette kommer senere.

7.1.2 Prosjektmøter

Gruppen er innstilt på å ha minimum et prosjektmøte i uka. Dette vil resultere i totalt 16 prosjekt møter. Dette er satt som et minimum og vi anser det som veldig sannsynlig at dette antallet vil være høyere.

Møtedatoer
torsdag 28. januar 2021
torsdag 4. februar 2021
torsdag 11. februar 2021
torsdag 18. februar 2021
torsdag 25. februar 2021
torsdag 4. mars 2021
torsdag 11. mars 2021
torsdag 18. mars 2021
torsdag 25. mars 2021
torsdag 1. april 2021
torsdag 8. april 2021
torsdag 15. april 2021
torsdag 22. april 2021
torsdag 29. april 2021
torsdag 6. mai 2021
torsdag 13. mai 2021

Figur 3: Møtedatoer

7.2 Periodiske rapporter

7.2.1 Framdriftsrapporter (inkl. milepæl)

Framdriftsrapporter vil bli skrevet hver fjerde uke som vil gi en total på fire framdriftsrapporter. Dette inkludert med møtene med styringsgruppene føler vi burde være tilstrekkelig.

8 PLANLAGT AVVIKSBEHANDLING

Hvis det blir oppdaget at fremgangen er bak den initiale planen vil vi undersøke årsaken til dette. Det kan være spesielle hendelser som gjør at fremgangen i perioder ikke er som ønskelig. Hvis dette er noe som vi ikke klarer å løse innad i gruppen vil vi ta kontakt med veiledere for bistand/anbefalinger for fremtidige tiltak.

9 UTSTYRSBEHOV/FORUTSETNINGER FOR GJENNOMFØRING

Gruppen har behov for å få utlevert noe kritisk utstyr for å kunne fullføre prosjektet. Etter samtale med NAVO NTNU ble det konkludert med at de har en ekstra pixhawk. Utenom dette har vi behov for:

Utstyr	Antall	Pris per enhet	Tilgang på skolen
T200 Blue robotics	x4		Yes
RealSense Depth Camera D435	x1		
Båt skrog	x1		Yes
Blue Robotics Battery	x1	2470 kr	No
Blue ESC	x4		---
GPS Pixhawk module	x1	360 kr	---
Pixhawk	x1		Yes

Tabell 2: Utstyrsbehov

10 REFERANSER

VEDLEGG

Vedlegg 1 VantTec Technical Design Report 2019 for roboboat

Vedlegg 2 Autodrone konkurranse regler og spesifikasjoner

Appendix B

Gantt diagram

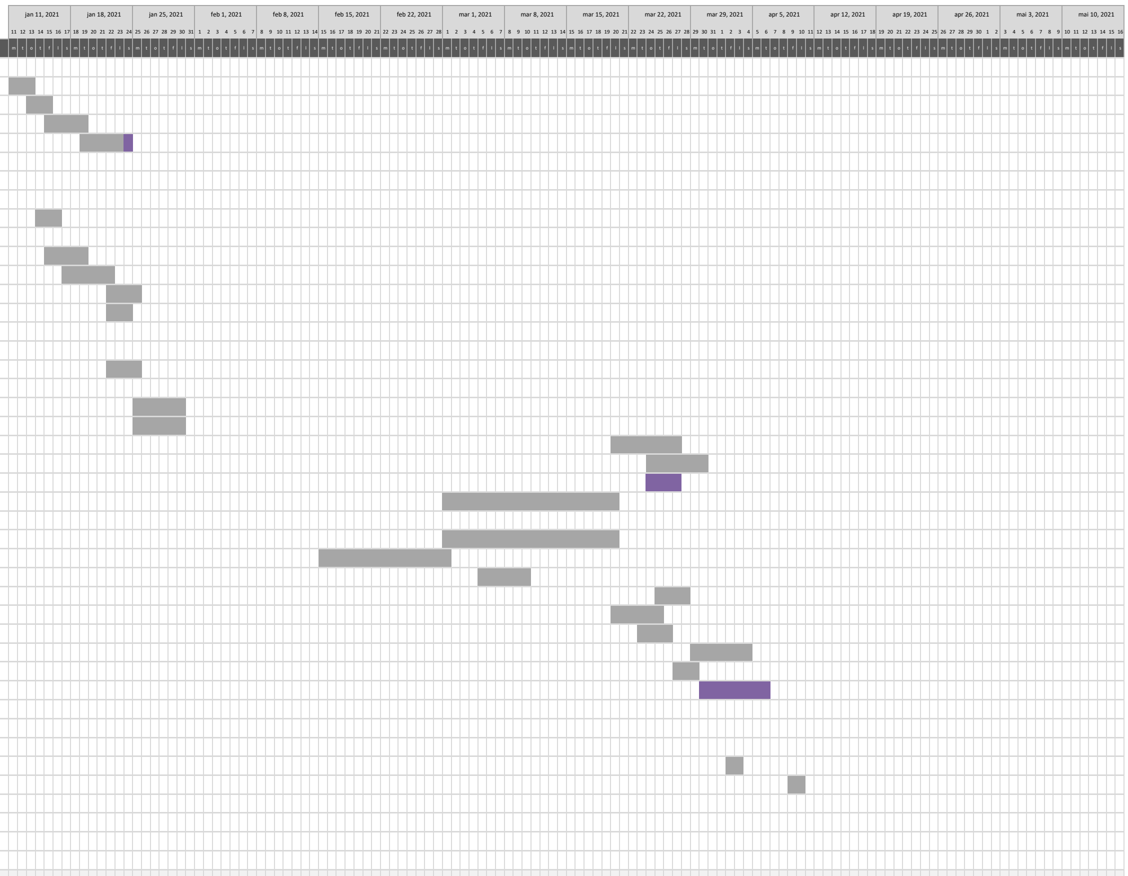
Group 14 USV Sjø Drone

NTNU Aalesund
Magnus Stava = MS
Markus Grorud Gaasholt = MGG
Kaung Htet San = KHS

Project Start: søn, 1.10.2021
Display Week: 1

TASK	ASSIGNED TO	PROGRESS	START	END
Mechanical				
Boat waterproofing	MS/NAVO NTNU	100 %	1.10.21	1.13.21
Boat hull construction	MS/NAVO NTNU	100 %	1.13.21	1.15.21
Boat frame construction	MS/NAVO NTNU	100 %	1.15.21	1.19.21
Boat mechanical construction	MS/NAVO NTNU	95 %	1.19.21	1.24.21
Camera frame	MGG	100 %	10.3.21	3.20.21
3D printing	MGG	100 %	---	---
Water coating floating devices	MS	100 %		
Create the lid	MGG	100 %	1.14.21	1.16.21
Electrical				
Mount Thrusters	MS	100 %	1.15.21	1.19.21
Mount Radio reciver	MS	100 %	1.17.21	1.22.21
Mount Jetson TX2	MS	100 %	1.22.21	1.25.21
Mount Camera	MGG	100 %	1.22.21	1.24.21
Mount and waterproof wiring	MS	100 %	date	date
Connect components to electrical circuit	MS	100 %	date	date
Mount Emergency button	MS	100 %	1.22.21	1.25.21
Software				
Software research		100 %	25.01.2021	30.01.2021
Ubuntu setup	MS/KHS	100 %	25.01.2021	30.01.2021
Code for waypoint setting	MS/KHS	100 %	20.03.2021	27.03.2021
Code for depth reading	MGG	100 %	24.03.2021	30.03.2021
Code for obstical detection	MS	0 %	24.03.2021	27.03.2021
Set up Pixhawk	KHS	100 %	01.03.2021	20.03.2021
Picture for database	MGG	100 %		
Set up Zed 2 (Camera)	MS	100 %	01.03.2021	20.03.2021
Set up Jetson TX2	MS	100 %	15.02.2021	01.03.2021
Connect/Communication between Zed 2 and Jetson	MS	100 %	05.03.2021	10.03.2021
Connect/Communication between pixhawk and Jetson	MS	100 %	25.03.2021	28.03.2021
Implement YOLO on jetson with Zed 2	MS	100 %	20.03.2021	25.03.2021
Train YOLO custom dataset	MS	100 %	23.03.2021	26.03.2021
Set waypoint with pixhawk and Here 3 GPS	MS/KHS	100 %	29.03.2021	04.04.2021
Store compass value	KHS/MS	100 %	27.03.2021	29.03.2021
Obstacle avoidance with depth detection	MGG	0 %	30.03.2021	06.04.2021
Testing				
Manuell Test		100 %	date	date
Test of camera/YOLO object detection		100 %	date	date
Test of waypoint setting		100 %	02.04.2021	03.04.2021
Boat waypoint test		100 %	09.04.2021	10.04.2021
Boat YOLO test on water		0 %	date	date
Speed course test		0 %	date	date
Report				
Report	Everyone	100 %	10.1.21	date

Insert new rows ABOVE this one



Appendix C

Time Sheet

	Markus	Kommentarer	San	Kommentarer	Stava	Kommentarer
mandag 18. januar 2021	3		4		4	
tirsdag 19. januar 2021			4		3	
onsdag 20. januar 2021			4		1	
torsdag 21. januar 2021					-	
fredag 22. januar 2021					-	
lørdag 23. januar 2021	2		0		-	
søndag 24. januar 2021	3		0		-	-
Uke 4	**Markus**	**Kommentarer**	**San**	**Kommentarer**	**Stava**	**Kommentarer**
mandag 25. januar 2021			4	Leser dokumentere om forrige NAVO tekniske rapport	-	-
tirsdag 26. januar 2021	1		2	Leser dokumentere om forrige NAVO tekniske rapport	5	Båt konstruksjon med NAVO NTNU
onsdag 27. januar 2021			3	Leser rapport fra forrige sjødrone bachelor rapport	-	undersøker pixhawk funksjoner på nett
torsdag 28. januar 2021	3		4	Leser rapport fra forrige sjødrone bachelor rapport	2	Innlevering forprosjekt
fredag 29. januar 2021					-	
lørdag 30. januar 2021					-	
søndag 31. januar 2021					-	
Uke 5	**Markus**	**Kommentarer**	**San**	**Kommentarer**	**Stava**	**Kommentarer**
mandag 1. februar 2021			1	Undersøker og leser dokumentasjoner om pixhawk	-	INDU 4.0
tirsdag 2. februar 2021			3	Undersøker og leser dokumentasjoner om pixhawk	-	INDU 4.0
onsdag 3. februar 2021			2	Undersøker og leser dokumentasjoner om pixhawk	-	INDU 4.0
torsdag 4. februar 2021			3	Undersøker og leser dokumentasjoner om pixhawk	5	Lese gjennom forrige bachlor.
fredag 5. februar 2021			6	Installere missionplanner	7	Overleaf, TX2 setup, Pixhawk setup, Random Forest setup
lørdag 6. februar 2021					7	Darknet (YOLO) setup, begynt å se på ROS og mulige ROS-pakker for path finding delen
søndag 7. februar 2021					-	
Uke 6	**Markus**	**Kommentarer**	**San**	**Kommentarer**	**Stava**	**Kommentarer**
mandag 8. februar 2021			0		4	Påbegynt RF regressor samt openCV. Båtkonstruksjon ble flyttet til Tirsdag.
tirsdag 9. februar 2021			0		6	Noe problemer med RF, revurderer strategi ser på reinforcement fremfor Fuzzy logic RF. 90% FERDIG MED BÅT KONSTRUKSJON
onsdag 10. februar 2021			3	Research Ardupilot	9	Begynt å se på Deep RL, samt fullført mekaniske delen med båten. Mekanisk ble ferdig :)
torsdag 11. februar 2021	3	3D moddelering av thrustermounts	4	fikk GPS og tester	2	Gruppemøte, Open AI ROS package, Anders produkt bestilling
fredag 12. februar 2021	5	3D moddelering og printing av thrustermounts (Bug i fila ødela mange prints)	0		-	
lørdag 13. februar 2021					-	
søndag 14. februar 2021					-	
Uke 7	**Markus**	**Kommentarer**	**San**	**Kommentarer**	**Stava**	**Kommentarer**
mandag 15. februar 2021	3	Fiksing av TM. Fått bestillt inn skruer+skiver til MT. Rahul ROS crash course			6	Gazebo simulation, Open AI, Rahul ROS crash intro, artikkel bruk av Stereo kamera + GPS for path tracking/following
tirsdag 16. februar 2021	6	Sett på filer tilsendt fra bachelor 2020. + div småarbeid	5	Programmere ROS og Ardupilot	7	ROS tutorials, fått bekreftet bestilt rele samt kamera. Lest gjennom artikkel ang stereko kamera osv
onsdag 17. februar 2021			5	Tester Ardupilot	9	Forsterket båt leddene, ROS tutorials, Bekreftet bestilling av Kamera av Anders
torsdag 18. februar 2021			5	Kalibere PX4	-	
fredag 19. februar 2021			6	tester PX4	-	
lørdag 20. februar 2021			6	Tester PX4	2	Fullført ROS basic tutorials. Elektronikk montering utsatt pga glemt å ta ut ESC og mangel av skruer.
søndag 21. februar 2021						
Uke 8	**Markus**	**Kommentarer**	**San**	**Kommentarer**	**Stava**	**Kommentarer**
mandag 22. februar 2021			3	Undersøker og leser dokumentasjoner om ROS	5	Skruer som var bestilt ankommet. Begynt montering av Thrusters samt dra ledninger. Møte med rahul angående elektronikk oppkobling/system overview
tirsdag 23. februar 2021			2	Installere Ubuntu	9	Fullført elektronikk oppkobling for manuell test. ESC + Batteri. Sent utstyrsliste av det siste som mangler til Anders.
onsdag 24. februar 2021			3	Installere ROS pakker	8	Veiledermøte, gruppemøte, liten kobling lagd for strøm til pixhawk. Rapport skriving. Forberedelser til manuell test
torsdag 25. februar 2021	2	kamera	5	Tester ROS pakker og biblioteker	3	Siste forberedelser til manuell test, manuell test, ROS
fredag 26. februar 2021			4	Programmere for	6	Rapport arbeid, Kalman Filter, SLAM via Stereo Camera
lørdag 27. februar 2021					--	
søndag 28. februar 2021					--	
Uke 9	**Markus**	**Kommentarer**	**San**	**Kommentarer**	**Stava**	**Kommentarer**
mandag 1. mars 2021	0,5	mail Paul Steffen	2	Programmere Ros	2	INDU 4.0, nødstopp oppkobling, begynt rewiring av ESC
tirsdag 2. mars 2021	0,5	mail Paul Steffen	3	Gjeninstallere Ubuntu	1	Rewired ferdig ESC, gjort klar nødstopp m/ledning,
onsdag 3. mars 2021					1	Koblet opp nødstopp, bestilt ny GPS og gjenværende komponenter med Anders
torsdag 4. mars 2021					--	
fredag 5. mars 2021	2	Lastet ned VM	3	Programmere ROS	6	Koblet opp mobil strømkilde for jetson TX2, lagd plate for montering av komponenter. Dratt og montert nødstopp ledning inn til båten.
lørdag 6. mars 2021	2	JÆVLA FØKKING UBUNTU HÆLVETTE! GRJERGJEWGOKWEGKPOWGKW. God S	2	Programmere ROS	6	ZED 2 SDK lastet ned på Jetson, Tilhørende ROS packages lastet ned, Montert feste nødstopp, feste pixhawk, ESC kabel managment
søndag 7. mars 2021	3	windows Zed greier			--	
Uke 10	**Markus**	**Kommentarer**	**San**	**Kommentarer**	**Stava**	**Kommentarer**
mandag 8. mars 2021			4	Studere om ROS Python	2	Koblet opp Zed 2 og Jetson, Error ved oppstart av Rviz, etter feilsøking løste problem med reinstallere ROS
tirsdag 9. mars 2021			3	Studere om ROS Python	2	Problem gjennoppsto, fikk løst og testet videre ZED 2 funksjoner
onsdag 10. mars 2021			1	Studere om ROS Python	--	INDU 4.0
torsdag 11. mars 2021	0,2	Utarbeidet plan for kameraholder	2	Skriver Rapport	--	INDU 4.0
fredag 12. mars 2021	0,2	-||-	1	Gjeninstallere Ubuntu og ROS	--	INDU 4.0
lørdag 13. mars 2021	0,2	-||-			--	
søndag 14. mars 2021	0,2	-||-			--	
Uke 11	**Markus**	**Kommentarer**	**San**	**Kommentarer**	**Stava**	**Kommentarer**
mandag 15. mars 2021	0,2	-||-	8	ROS programmering	--	INDU 4.0
tirsdag 16. mars 2021	0,2	-||-	5	Ros programmering	4	Mavros setup på Jetson
onsdag 17. mars 2021	0,2	-||-	3	Skriver Rapport	9	MavLink read up, GANTT forberdring, Jetson/Zed 2 bug oppstår ved source/devel/setup.bash. Fått ut dybdemålniger fra Zed 2. Obj deteksjon modul corrupted SDK ....
torsdag 18. mars 2021	0,2	-||-	6	ROS programmering	9	Løst Obj deteksjon og det funker, neste steg er å laste opp custom weights. custom param er lastet opp og filer er rekonfigurert. Begynt med logikk koding for dybdeavlesning fra Zed publication.
fredag 19. mars 2021	0,2	-||-	6	ROS programmering i Ubuntu	3	Convert bilde info fra XLM til txt. Lagt inn og gjort klar til å teste training av modellen på egendefinert dataset
lørdag 20. mars 2021	0,2	-||-			-	
søndag 21. mars 2021					-	
Uke 12	**Markus**	**Kommentarer**	**San**	**Kommentarer**	**Stava**	**Kommentarer**
mandag 22. mars 2021	5	3d-Kameraholder. Trenger desp. Printere...	3	Gjeninstallere Ubuntu og ROS	3	yolov2-tiny som start punkt for weights og begynt treningen. Noe oppstarts problemer. Satte på to modeller for trening på Jetson
tirsdag 23. mars 2021	9	Skrevet mail til IIR og IHB. Jobbet videre med kameraholder. Endret fra bruk av TPU til PLA og akryl	6	Programmere arm, disarm, mode i python	6	Oppdaget feil med modellen som ble trent. Feilen lå i config filen. Endret dette og restartet treningen. Ser mer korrekt ut nå. Fortsatt på¨GANTT dia og levert dritten
onsdag 24. mars 2021	2	Må testes for COVID før jeg får brukt labbene på skolen.	6	Programmere arm, disarm, mode i python	1	Rapport
torsdag 25. mars 2021			5	Programmere arm, disarm, mode i python	--	
fredag 26. mars 2021	4,5	prøver å lære meg slangespråk	4	Debugging	1	Rapport
lørdag 27. mars 2021			5	Debugging	--	
søndag 28. mars 2021					--	
Uke 13	**Markus**	**Kommentarer**	**San**	**Kommentarer**	**Stava**	**Kommentarer**
mandag 29. mars 2021	5	Jobber med slangespråk opplæring	2	Studere MAV LINK i ROS	--	
tirsdag 30. mars 2021	6	Jobber med slangespråk opplæring	2	Programmere for waypoint objekt	2	Rapport og YOLO testing. Tror alt er good må bare trenes lenger ?:)
onsdag 31. mars 2021	6	lærer fortsatt python	5	Programmere Way Point i MAVLINK	2	Rapport og depth subscriber code shell
torsdag 1. april 2021	5	startet på ros	5	Programmere Way Point i MAVLINK	--	
fredag 2. april 2021	5	prøver meg litt frem i python for å lære	0		--	
lørdag 3. april 2021			0		--	
søndag 4. april 2021			0		--	
Uke 14	**Markus**	**Kommentarer**	**San**	**Kommentarer**	**Stava**	**Kommentarer**
mandag 5. april 2021			5	Programmere endringer i Koden	--	
tirsdag 6. april 2021	3	Prøvd å printe kamerafeste (feilet)	4	Programmere endringer i Koden	3	Begynt med YOLO nedlasting til PC, tar for lang tid på jetson. Litt rapport samt gruppemøte.
onsdag 7. april 2021				Møte	1	Rapport
torsdag 8. april 2021	3	Prøvd å printe kamerafeste (feilet)	3	Tester ROS i Gazebo og Turtle simulasjon	--	
fredag 9. april 2021			3	Tester ROS i Gazebo og Turtle simulasjon	--	
lørdag 10. april 2021			2	Tester ROS i Gazebo og Turtle simulasjon	3	YOLO config fil + testing. Noe rapport inni mellom slagene
søndag 11. april 2021			-		6	YOLO config fil + testing. Noe rapport inni mellom slagene
Uke 15	**Markus**	**Kommentarer**	**San**	**Kommentarer**	**Stava**	**Kommentarer**
mandag 12. april 2021			5	Studere MAVROS pakke i ROS	2	SMå modifikasjoner på CFG filen(AI nettverket), hentet garnblåsene for å teste
tirsdag 13. april 2021	2	Sliping av TM	5	Studere MAVROS pakke i ROS	2	Rapport
onsdag 14. april 2021	3	Printing av kamerafeste og satt på to av de nye TM	3	Skriver Rapport	-	
torsdag 15. april 2021	4	kamerafestet ble feil, endret litt på modellen. Prøvde å printe på nytt tre ganger, men alle feilet.	4	Skriver rapport	6	Back to school <3, rapport all day + ta bilder fra kaien for å teste YOLO modellen
fredag 16. april 2021			4	Gjeninstaller Ubuntu og ROS	3	Rapport, testet YOLO modellen med bilder fra div lengder 6/10
lørdag 17. april 2021			6	Gjeninstaller Ubuntu og ROS	5	Pixhawk nå kommunikasjon med TX2, kommunikasjon via Mavlink /MavROS
søndag 18. april 2021					4	Telemetry kommunikasjon good, Noe problemer fortsatt med MavROS
Uke 16	**Markus**	**Kommentarer**	**San**	**Kommentarer**	**Stava**	**Kommentarer**
mandag 19. april 2021	6	printa, laserkutta og slipt kamerafeste. mangler bare mellom aluframe og båt	4	Programmere Navigation	4	Rapport
tirsdag 20. april 2021	4	Festet kamera på båt	5	Programmere Navigation	9	Rapport fått på plass en del av figurene for teori delen
onsdag 21. april 2021	2	litt div. + NSO	8	Programmere Navigation	5	Rapport, planlagt testing til neste uke. Jobbet litt med den trådløse switchen
torsdag 22. april 2021		NSO	5	Programmere Navigation	3	Kort dag. Rapport. Lang dag imorgen :)
fredag 23. april 2021		NSO	5	Programmere Navigation	6	Rapport
lørdag 24. april 2021		NSO			9	:/
søndag 25. april 2021		NSO			6	Trent ny modell for å bruke til sammenligning. Prøvd å få darknet over på pcen siste steg mislykket. Samt noe rapport. Starter dagen imorgen med noen timer rapport
Uke 17	**Markus**	**Kommentarer**	**San**	**Kommentarer**	**Stava**	**Kommentarer**
mandag 26. april 2021					11	Rapport
tirsdag 27. april 2021	4	Rapport	3	Rapport Skriving	10	Rapport, samt begynt med ferdigstille YOLO resultater
onsdag 28. april 2021	10	Rapport	2	Rapport Skriving	11	Rapport
torsdag 29. april 2021	4	😭 <-- this is the 'loudly cying face' emoji	5	Rapport Skriving	13	Rapport, Kommet godt igang med test forberedelser. Satser på noen tester på Laurdag. Rapporten har kommet veldig godt igang. Objekt deteksjonen har 3 individuelle modeller for å sammenligne resultater.
fredag 30. april 2021	10	Rapport	6	Rapport Skriving	9	Rapport, GPS test klart, Waypoint 50% klar(?)
lørdag 1. mai 2021	8	Rapport	8	Rapport Skriving	11	Rapport, Forberedelser til test på Søndag
søndag 2. mai 2021	9	Teste båt og rapport	9	Tester båt og rapport	13	Testing, fullføring av rapport til første utkast for veilederne å rette
Uke 18	**Markus**	**Kommentarer**	**San**	**Kommentarer**	**Stava**	**Kommentarer**
mandag 3. mai 2021	5	Rapport			6	
tirsdag 4. mai 2021	6	Rapport	6	Rapport Skriving	10	Waypoint klart til testing, samt holding av posisjon klar til test.
onsdag 5. mai 2021	5	Rapport	7	Fixer koden på grunn av endringer i plan	8	Rapport, satt in 10m radius delen til San i wp setting. TODO for imorgen, ta inn signal fra RC og ta inn yaw signal fra IMUen til PX4
torsdag 6. mai 2021	7	Rapport	5	Fixer koden på grunn av endringer i plan	9	Fjernkontroll nødstopp klart, Forberedt til testing på søndag iform av software. Samt en liten tur innom rapport
fredag 7. mai 2021	6	Rapport	5	Fixer koden på grunn av endringer i plan	13	Ish klar til testing
lørdag 8. mai 2021	5	Rapport	9	Tester båt og rapport	13	Testing gikk så som så, fikk info som gjorde feilsøking enklere. Ny testing imorgen. Rapport på kvelden
søndag 9. mai 2021	6	Rapport	9	tester båt og rapport	10	Testing noe mer vellykket. RTL good, Hold not good then good. WP not good. Rapport på kvelden
Uke 19	**Markus**	**Kommentarer**	**San**	**Kommentarer**	**Stava**	**Kommentarer**
mandag 10. mai 2021	8	Rapport	3	Skriver Rapport	12	Rapport, 3 av 3 modus klar til testing
tirsdag 11. mai 2021	4	Testing av båt, letet etter telefonen til Stava i sjøen	5	Skriver Rapport	4	Testing, kort dag. Telefon i sjøen, øl på kvelden
onsdag 12. mai 2021	6	Rapport	5	Skriver Rapport	10	Relativt sen start på dagen. Ellers fokus på rapport, sett på data fra testene of fremstilt dem. Generel retting av språk og en del på discussion.
torsdag 13. mai 2021	7	Rapport	4	Skriver Rapport og fikser feil i Rapport	9	Rapport 200%
fredag 14. mai 2021	4	Rapport	6	Skriver Rapport og fikser feil i Rapport	3	Kort dag, rapport og noe fremgang på WP. Nå hjem og sove :/
lørdag 15. mai 2021	5	Rapport	5	Skriver Rapport og fikser feil i Rapport	8	Rapport
søndag 16. mai 2021	8	Rapport	6	Små arbeider med båt	10	Rapport, noe arbeid på båt med epoxy og lignende småting.
Uke 20	**Markus**	**Kommentarer**	**San**	**Kommentarer**	**Stava**	**Kommentarer**
mandag 17. mai 2021			5	Skriver Rapport og fikser feil i Rapport	5	Montert sammen båten på morgenen, fikset opp i manglende resultater fra objekt deteksjon på kvelden. Gjennombrudd mtp waypoint også. Meget god 17 mai totalt
tirsdag 18. mai 2021	6	Rapport og video og testing	6	Rapport Skriving og testing av båt	11	Rapport, siste testing WP fungerte ish.
onsdag 19. mai 2021	16	Rapport og video	14	Skriver Rapport og fikser feil i Rapport	16	Rapport
torsdag 20. mai 2021	6	Innlevering av bachelor	6	Innlevering av bachelor	6	Innlevering av bachelor
Total	271		393		488	

Appendix D

YOLOv2 cfg

```
[net]
# Testing
batch=1
subdivisions=1
# Training
#batch=64
#subdivisions=8
width=608
height=608
channels=3
momentum=0.9
decay=0.0005
angle=0
saturation = 1.5
exposure = 1.5
hue=.1

learning_rate=0.00001
burn_in=1000
max_batches=6000
policy=steps
steps=4800,5400
scales=.1,.1

[convolutional]
batch_normalize=1
filters=16
size=3
stride=1
pad=1
activation=leaky

[maxpool]
size=2
stride=2

[convolutional]
batch_normalize=1
filters=32
size=3
stride=1
pad=1
activation=leaky

[maxpool]
size=2
stride=2

[convolutional]
batch_normalize=1
filters=64
size=3
stride=1
pad=1
```

```
activation=leaky

[maxpool]
size=2
stride=2

[convolutional]
batch_normalize=1
filters=128
size=3
stride=1
pad=1
activation=leaky

[maxpool]
size=2
stride=2

[convolutional]
batch_normalize=1
filters=256
size=3
stride=1
pad=1
activation=leaky

[maxpool]
size=2
stride=2

[convolutional]
batch_normalize=1
filters=512
size=3
stride=1
pad=1
activation=leaky

[maxpool]
size=2
stride=1

[convolutional]
batch_normalize=1
filters=1024
size=3
stride=1
pad=1
activation=leaky

###########

[convolutional]
batch_normalize=1
```

```
size=3
stride=1
pad=1
filters=512
activation=leaky

[convolutional]
size=1
stride=1
pad=1
filters=40
activation=linear

[region]
anchors =  0.57273, 0.677385, 1.87446, 2.06253, 3.33843, 5.47434, 7.88282, 3.52778, 9.77052, 9.16828
bias_match=1
classes=3
coords=4
num=5
softmax=1
jitter=.2
rescore=0

object_scale=5
noobject_scale=1
class_scale=1
coord_scale=1

absolute=1
thresh = .6
random=1
```

Appendix E

YOLOv3 cfg

```
[net]
# Testing
batch=1
subdivisions=1
# Training
#batch=64
#subdivisions=8
width=608
height=608
channels=3
momentum=0.9
decay=0.0005
angle=0
saturation = 1.5
exposure = 1.5
hue=.1

learning_rate=0.0001
burn_in=1000
max_batches = 7000
policy=steps
steps=5600,6300
scales=.1,.1

[convolutional]
batch_normalize=1
filters=16
size=3
stride=1
pad=1
activation=leaky

[maxpool]
size=2
stride=2

[convolutional]
batch_normalize=1
filters=32
size=3
stride=1
pad=1
activation=leaky

[maxpool]
size=2
stride=2

[convolutional]
batch_normalize=1
filters=64
size=3
stride=1
pad=1
```

```
activation=leaky

[maxpool]
size=2
stride=2

[convolutional]
batch_normalize=1
filters=128
size=3
stride=1
pad=1
activation=leaky

[maxpool]
size=2
stride=2

[convolutional]
batch_normalize=1
filters=256
size=3
stride=1
pad=1
activation=leaky

[maxpool]
size=2
stride=2

[convolutional]
batch_normalize=1
filters=512
size=3
stride=1
pad=1
activation=leaky

[maxpool]
size=2
stride=1

[convolutional]
batch_normalize=1
filters=1024
size=3
stride=1
pad=1
activation=leaky

###########

[convolutional]
batch_normalize=1
```

```
filters=256
size=1
stride=1
pad=1
activation=leaky

[convolutional]
batch_normalize=1
filters=512
size=3
stride=1
pad=1
activation=leaky

[convolutional]
size=1
stride=1
pad=1
filters=24
activation=linear



[yolo]
mask = 3,4,5
anchors = 10,14,  23,27,  37,58,  81,82,  135,169,  344,319
classes=3
num=6
jitter=.3
ignore_thresh = .7
truth_thresh = 1
random=1

[route]
layers = -4

[convolutional]
batch_normalize=1
filters=128
size=1
stride=1
pad=1
activation=leaky

[upsample]
stride=2

[route]
layers = -1, 8

[convolutional]
batch_normalize=1
filters=256
size=3
```

```
stride=1
pad=1
activation=leaky

[convolutional]
size=1
stride=1
pad=1
filters=24
activation=linear

[yolo]
mask = 0,1,2
anchors = 10,14,  23,27,  37,58,  81,82,  135,169,  344,319
classes=3
num=6
jitter=.3
ignore_thresh = .7
truth_thresh = 1
random=1
```

Appendix F

Custom waypoint

```
import rospy
import time
import math
from math import pi,cos,sin,tan,atan
from std_msgs.msg import String
from sensor_msgs.msg import NavSatFix
from mavros_msgs.msg import *
from mavros_msgs.srv import *
from std_msgs.msg import Float64
from mavros_msgs.msg import Waypoint    # define waypoints
from mavros_msgs.srv import WaypointPush # push Waypoint

#Wp dictonaries x and y values for 10m setting
#Should be redone for more precise waypoint setting
arming = None
keys = []
#X values with 10m radius from vessel
y_values = [0,-1.74,-3.42,-5,-6.43,-7.66,-8.66,-9.4,-9.85,-10,
-9.85,-9.4,-8.66,-7.66,-6.43,-5,-3.42,-1.74,0,1.74,3.42,5,6.43,
7.66,8.66,9.4,9.85,10,9.85,9.4,8.66,7.66,6.43,5,3.42,1.74,0]
#Y values with 10m radius from vessel
x_values = [10,9.88,9.4,8.66,7.66,6.43,5,3.42,1.74,0,-1.74,
-3.42,-5,-6.43,-7.66,-8.66,-9.4,-9.85,-10,-9.85,-9.4,-8.66,
-7.66,-6.43,-5,-3.42,-1.74,0,1.74,3.42,5,6.43,7.66,8.66,9.4,
9.85,10]
#Key values representing orientation of the vessel
#When orientation matches keys, set the x and y values from the corresponding key as WP
degrees_values = [360,350,340,330,320,310,300,290,280,270,260,
250,240,230,220,210,200,190,180,170,160,150,140,130,120,110,100,
```

```
90,80,70,60,50,40,30,20,10,0]
#store values in dict

a = 0
dict_x = {}
dict_y = {}
for i in degrees_values:
  key = degrees_values[a]
  x = x_values[a]
  y = y_values[a]
  dict_x[key]=x
  dict_y[key]=y
  a += 1

WP_lon = open(r"WP_lon_.txt","w")
WP_lat = open(r"WP_lat_.txt","w")
WP_x = open(r"WP_xValue.txt","w")
WP_y = open(r"WP_yValue.txt","w")
WP_launch_lon = open(r"WP_launch_lon","w")
WP_launch_lat = open(r"WP_launch_lat","w")
#Global variables
check = None
glob = ''
rc_input = None
current_mode = None
wp_set = False
lat = None
lon = None
#Pre determined modes
MODE_MANUAL = "MANUAL"
```

```
MODE_GUIDED = "GUIDED"
MODE_AUTO = "AUTO"

#Get values from matching key
def dict_to_float(key, dicts):
  #print dicts.keys()
  if key in dicts.keys():
    res = dicts[key]
    return res

### SET MODE TO HOLD ###
def change_mode(mode):
  global current_mode
  rospy.wait_for_service('/mavros/set_mode')
  try:
    if current_mode != mode:
      change_mode = rospy.ServiceProxy('/mavros/set_mode', SetMode)
      response = change_mode(custom_mode=mode)
      rospy.loginfo(response)
      #mode_change = rospy.ServiceProxy('/mavros/set_mode',SetMode)
      #mode_change(0,mode)
      #current_mode = mode
  except rospy.ServiceException as e:
    print("Mode change failed: %s" %e)

#Arms and Disarms the vehicle.
def arm(status):
  global arming
  rospy.wait_for_service('/mavros/cmd/arming')
  try:
```

```
    if arming != status:
      arming_cl = rospy.ServiceProxy('/mavros/cmd/arming', CommandBool)
      response = arming_cl(value = status)
      rospy.loginfo(response)
      arming = status
    else:
      print("Aleady set!")
  except rospy.ServiceException as e:
    print("Arming failed: %s" %e)

#Defines waypoint
#x_lat and y_long defined in meters
def create_waypoint(x,y):
  waypoint_clear_client()
  time.sleep(1)
  wl = []
  wp = Waypoint()   #Creates new instance of WayPoint
  wp.frame = 0 # PX4 Only supports Global frame
  wp.command = 22  #Takeoff command
  wp.is_current = False
  wp.autocontinue = True
  wp.param1 = 0  # HOLD time at WP
  wp.param2 = 0  # Acceptance radius, if inside WP count as reached
  wp.param3 = 0  # Pass Radius. If 0 go through WP
  wp.param4 = 0#float('nan')  # Yaw, 0 for our USV situation
  wp.x_lat = 0  #movement North/South
  wp.y_long = 0 #movement East/West
  wp.z_alt = 0
  wl.append(wp)

```

```
wp2 = Waypoint()   #Creates new instance of WayPoint
wp2.frame = 0 # PX4 Only supports Global frame
wp2.command = 16  #Navigate to waypoint.
wp2.is_current = False
wp2.autocontinue = True
wp2.param1 = 0  # HOLD time at WP
wp2.param2 = 0  # Acceptance radius, if inside WP count as reached
wp2.param3 = 0  # Pass Radius. If 0 go through WP
wp2.param4 = 0 #float('nan')  # Yaw, 0 for our USV situation
wp2.x_lat = x #movement North/South
wp2.y_long = y #movement East/West
wp2.z_alt = 0
wl.append(wp2)

wp2 = Waypoint()   #Creates new instance of WayPoint
wp2.frame = 0 # PX4 Only supports Global frame
wp2.command = 19  #HOLD position
wp2.is_current = False
wp2.autocontinue = True
wp2.param1 = 30  # HOLD time at WP
wp2.param2 = 0  # Acceptance radius, if inside WP count as reached
wp2.param3 = 0  # Pass Radius. If 0 go through WP
wp2.param4 = 0 #float('nan')  # Yaw, 0 for our USV situation
wp2.x_lat = x #movement North/South
wp2.y_long = y #movement East/West
wp2.z_alt = 0
wl.append(wp2)

wp2 = Waypoint()   #Creates new instance of WayPoint
wp2.frame = 0 # PX4 Only supports Global frame
```

```
    wp2.command = 20  #RTL
    wp2.is_current = False
    wp2.autocontinue = True
    wp2.param1 = 0  # HOLD time at WP
    wp2.param2 = 0  # Acceptance radius, if inside WP count as reached
    wp2.param3 = 0  # Pass Radius. If 0 go through WP
    wp2.param4 = 0 #float('nan')  # Yaw, 0 for our USV situation
    wp2.x_lat = 0 #movement North/South
    wp2.y_long = 0 #movement East/West
    wp2.z_alt = 0
    wl.append(wp2)

    print("This is X value: %s"%x)
    print("This is Y value: %s"%y)

    rospy.wait_for_service('mavros/mission/push')
    try:
      global wp_set
      serviceReq = rospy.ServiceProxy('mavros/mission/push', WaypointPush)
      serviceRes = serviceReq(start_index=0, waypoints=wl)
      flag = serviceRes.success
      if flag == True:
        wp_set = True
        print('SUCCESS: PUSHING WP \n')
      elif flag == False:
        print('FAILURE: PUSHING WP \n')
    except rospy.ServiceException as e:
      rospy.loginfo("setWayPoint failed: %s\n" %e)

  # Activate the start of the mission
```

```
def takeoff():
  rospy.wait_for_service('/mavros/cmd/takeoff')
  try:
    takeoff_cl = rospy.ServiceProxy('/mavros/cmd/takeoff', CommandTOL)
        response = takeoff_cl(altitude=0, latitude=0, longitude=0,
         ↪  min_pitch=0, yaw=0)
        rospy.loginfo(response)
    print ('Takeoff')
    print response
  except rospy.ServiceException, e:
        print("Takeoff failed: %s" %e)

#Clears currently loaded waypoints
#Does not remove the currently active waypoint
def waypoint_clear_client():
  rospy.wait_for_service('mavros/mission/clear')
  global check
  try:
    response = rospy.ServiceProxy('mavros/mission/clear', WaypointClear)
    check = False
    #print("Waypoint mission clear: %s"%response.call().success)
    return response.call().success
  except rospy.ServiceException, e:
    print "Service call failed: %s" % e
    return False

#
def get_orientation(msg):
  heading = round(int(msg)/10)*10
  global glob
```

```
  glob = int(heading)

#get message from
def compass_callback(msg):
  msg.data
  get_orientation(msg.data)

#
###setting up way point
def set_wp(orientation):
  global check
  #rospy.wait_for_service('')
  if not check:
    key = glob
    x = dict_to_float(key,dict_x)
    y = dict_to_float(key,dict_y)
    x_lat = latitude(x)
    y_lon = longitude(y)
    create_waypoint(x_lat,y_lon)
    WP_x.write(str(x_lat))
    WP_x.write("\n")
    WP_y.write(str(y_lon))
    WP_y.write("\n")
    check = True

###Translate desired distance into latitude
def latitude(meters):
  earth = 6371000
  m = (1 / ((2 * pi / 360) * earth))
  new_latitude = lat + (meters * m)
```

```
  return new_latitude

###Translate desired distance into longitude
def longitude(meters):
  earth = 6371000
  m = (1 / ((2 * pi / 360) * earth)) / 1000 # 1 meter in degree
  new_longitude = lon + (meters * m) / cos(lat * (pi / 180))
  return new_longitude

####get gps position from ROS message library
def gps_callback(gps):
  global lat
  global lon
  lat = gps.latitude
  lon = gps.longitude

####get remote control channels
def rc_callback(msg):
    radio = get_radio(msg.channels)

####get message for remote control with the value above 2000 as input getting from radio comfirmed
# the value below 1000 as not getting input from radio
def get_radio(msg):
  global rc_input
  value = msg[1]

  if value > 2000:
    rc_input = True
  if value < 1000:
```

```
    rc_input = False
  else:
    rc_input = rc_input

### SAVES POSITION ###
def save_position():
  WP_lon.write(str(lon))
  WP_lon.write("\n")
##### WRITE LAT AND LONG TO FILES IN 5 SEC INTERVALS #####
  WP_lat.write(str(lat))
  WP_lat.write("\n")

def current_postion():
  WP_launch_lon.write(str(lon))
##### WRITE LAT AND LONG TO FILES IN 5 SEC INTERVALS #####
  WP_launch_lat.write(str(lat))



rospy.init_node('waypoint_node', anonymous=True)  #Initilizing ROS node with
↪ name
rate = rospy.Rate(10)
pub = rospy.Publisher('global',String,queue_size=10)  ##
sub_compass = rospy.Subscriber('mavros/global_position/compass_hdg',
↪ Float64, compass_callback) # subscribe to compass heading
sub_gps = rospy.Subscriber('mavros/global_position/global', NavSatFix,
↪ gps_callback) # subscribe to compass heading
sub_rc = rospy.Subscriber('mavros/rc/in', RCIn, rc_callback)

while not rospy.is_shutdown():
  sub_gps
```

```
sub_compass
if rc_input == True:
  rospy.sleep(1)
  set_wp(glob)
  if wp_set == True:
    change_mode(MODE_AUTO)
    current_postion()
    wp_set = False

if rc_input == True:
  save_position()
  print('POSITION SAVED')
if rc_input == False:
  waypoint_clear_client()
  print("WP CLEARED")
  change_mode(MODE_MANUAL)
rospy.sleep(1)
```

Appendix G

RTL code

```
import rospy
import time
from std_msgs.msg import String
from sensor_msgs.msg import NavSatFix
from mavros_msgs.msg import *
from mavros_msgs.srv import *
from std_msgs.msg import Float64

### SIMPLE CODE USED FOR MEASURING ACCURACY OF HOLD MODE ###

RTL_lon = open(r"RTL_lon.txt","w")
RTL_lat = open(r"RTL_lat.txt","w")

LAUNCH_lon = open(r"RTL_LAUNCH_lon.txt","w")
LAUNCH_lat = open(r"RTL_LAUNCH_lat.txt","w")

### POSITION CORDINATES ###
lat = None
lon = None

### RADIO CONTROLER ###
rc_input = False

current_mode = None

MODE_RTL = "RTL" #
MODE_MANUAL = "MANUAL" #
def rc_callback(msg):     #get remote control channels
    radio = get_radio(msg.channels)

```

```
def gps_callback(gps):    #get gps position from ROS message library
  global lat
  global lon
  lat = gps.latitude    #variable of lat assign to gps latitudes
  lon = gps.longitude    #variable of lon assign to gps longitude

def get_radio(msg):      #get message for remote control with the value above 2000 as input getting from radio comfirmed
  global rc_input      # the value below 1000 as not getting input from radio
  value = msg[1]
  if value > 2000:
    rc_input = True
  if value < 1000:
    rc_input = False
  else:
    rc_input = rc_input

### SET MODE TO HOLD ###
def change_mode(mode):
  global current_mode
  rospy.wait_for_service('/mavros/set_mode')
  try:
    if current_mode != mode:
      mode_change = rospy.ServiceProxy('/mavros/set_mode',SetMode)
      mode_change(0,mode)
      #print("Mode changed to: %s" %mode)
      current_mode = mode
      #response = mode_change.mode_sent
      #print(response)
```

```
  except rospy.ServiceException as e:
    print("Mode change failed: %s" %e)

### SAVES POSITION ###
def save_position():
##### WRITE LAT AND LONG TO FILES IN 5SEC INTERVALS #####
  RTL_lon.write(str(lon))
  RTL_lon.write("\n")
  RTL_lat.write(str(lat))
  RTL_lat.write("\n")

### SAVES LAUNCH POSITION ###
def launch_position():
  LAUNCH_lon.write(str(lon))
  LAUNCH_lon.write("\n")
  LAUNCH_lat.write(str(lat))
  LAUNCH_lat.write("\n")
  print lon
###
rospy.init_node('RTL_node', anonymous=True)  #Initilizing ROS node with name
pub = rospy.Publisher('global',String,queue_size=10)  ##
sub_gps = rospy.Subscriber('mavros/global_position/global', NavSatFix,
    gps_callback) # subscribe to compass heading
sub_rc = rospy.Subscriber('mavros/rc/in', RCIn, rc_callback)
sub_gps
rospy.sleep(1)
launch_position()
while not rospy.is_shutdown():
  sub_rc
  sub_gps
```

```
  if rc_input == True:
    change_mode(MODE_RTL)
    save_position()
    print('Going home')
  if rc_input == False:
    change_mode(MODE_MANUAL)
    print lat
  rospy.sleep(1)
```

Appendix H

HOLD code

```
import rospy
import time
from std_msgs.msg import String
from sensor_msgs.msg import NavSatFix
from mavros_msgs.msg import *
from mavros_msgs.srv import *
from std_msgs.msg import Float64

### SIMPLE CODE USED FOR MEASURING ACCURACY OF HOLD MODE ###

HOLD_lon = open(r"HOLD_lon_test.txt","w")
HOLD_lat = open(r"HOLD_lat_test.txt","w")

### POSITION CORDINATES ###
lat = None
lon = None
rc_input = None
current_mode = None
MODE_HOLD = "LOITER" #SAME AS MODE 'HOLD', DIFFERENCE BEING THAT THE VESSEL
↪ CAN BE OVERRIDED BY RC CMDS. ALSO HOLD DEMANDS MORE TESTING TO FUNCTION
MODE_MANUAL = "MANUAL"  # Variable to set vehicle mode to manual
def gps_callback(gps):    #get gps position from ROS message library
  global lat
  global lon
  lat = gps.latitude    #variable of lat assign to gps latitudes
  lon = gps.longitude    #variable of lon assign to gps longitude
###
def rc_callback(msg):    #get remote control channels
    radio = get_radio(msg.channels)
###
```

```
def get_radio(msg):    #get message for remote control with the value above
  ↪ 2000 as input getting from radio comfirmed
  global rc_input    # the value below 1000 as not getting input from radio
    ↪
  value = msg[1]
  if value > 2000:
    rc_input = True
  if value < 1000:
    rc_input = False
  else:
    rc_input = rc_input

### SET MODE TO HOLD ###
def change_mode(mode):
  global current_mode
  rospy.wait_for_service('/mavros/set_mode')
  try:
    if current_mode != mode:
      mode_change = rospy.ServiceProxy('/mavros/set_mode',SetMode)
      mode_change(0,mode)
      #print("Mode changed to: %s" %mode)
      current_mode = mode
      #response = mode_change.mode_sent
      #print(response)
  except rospy.ServiceException as e:
    print("Mode change failed: %s" %e)

### SAVES POSITION ###
def save_position():
  HOLD_lon.write(str(lon))
```

```
  HOLD_lon.write("\n")
##### WRITE LAT AND LONG TO FILES IN 5 SEC INTERVALS #####
  HOLD_lat.write(str(lat))
  HOLD_lat.write("\n")

###

rospy.init_node('hold_node', anonymous=True)  #Initilizing ROS node with
↪  name
sub_gps = rospy.Subscriber('mavros/global_position/global', NavSatFix,
↪  gps_callback) # subscribe to compass heading
sub_rc = rospy.Subscriber('mavros/rc/in', RCIn, rc_callback)
change_mode(MODE_MANUAL)
a = 0
while not rospy.is_shutdown():
  sub_rc
  sub_gps
  if rc_input == True:
    change_mode(MODE_HOLD)
    a += 5
    save_position()
    print('Hold started %d' % a)
  if rc_input == False:
    change_mode(MODE_MANUAL)
    a = 0
    print "MANUAL MODE"
  rospy.sleep(5)
```

Appendix I

Progress reports

16.02.2021

Fremgangsrapport for gruppe 14 USV Sjødrone

Gruppemedlemmer:

- **San**
- **Markus**
- **Magnus**

San

Forrige ukes status

- Fortsatt/blitt kjent med pixhawk software.
  - Mission planner.
  - ArduPilot.
- Begynt med oppkobling med GPS modulen.

Status

- Koblet opp GPS via pixhawk og mission planner.
  - Fått ut signal og sett på driftingen.
- Begynt å se på ROS/MAVROS package iforbindelse med GPS

Mål for de neste sju dagene

- Koble opp GPS i ROS.
  - Ta inn GPS signalet og lese ut via ROS.
- Se på/teste filter metoder.

Markus

Forrige ukes status

- Kutte ut topplate for båten.
- Møte med forrige års bachelor for oppdatering.
- Få båten på vannet for første manuell test.
- Utlevering av ZED 2 kamera.
- Få tilgang på bilder med label/begynne og sette label på bilder uten label.

Status

- *Har fått utlevert kamera.*
- *Har fått laget ferdig motorfester*

- *Møtt Rahul for å bli kjent med ROS*

Mål for de neste sju dagene

- Koble opp Zed 2 kamera
- Oppdatere firmware på kamera
- Forbedre dataset.
- Manuell testkjøring

Magnus

Forrige ukes status

- Assisterte NAVO NTNU med mekanisk konstruksjon
- Fortsatte og bli kjent med ROS rammeverket.
    - Utforske RL pakker som er i ROS(Open AI ROS)
- Møtt Rahul (NAVO NTNU leder) for ROS «kræsjkurs».

Status

- Montert thrusterne samt tilhørende ESCer. Lagd kobling opp mot batteriet.
- Møtt Rahul (NAVO NTNU leder) for videre samtaler om elektronisk oppsett.
- Gått igjennom ROS tutorialsene for å ha en grei forståelse før jeg begynner.

Mål for de neste sju dagene

- Få bestilt opp det siste av utstyr vi trenger. Pågående prosess med Anders
    - Telemetry radio.
    - USB WiFI forsterker.
    - Ny GPS HEX Here 3.
- Manuell test
- Begynne med implementering av ROS packages.

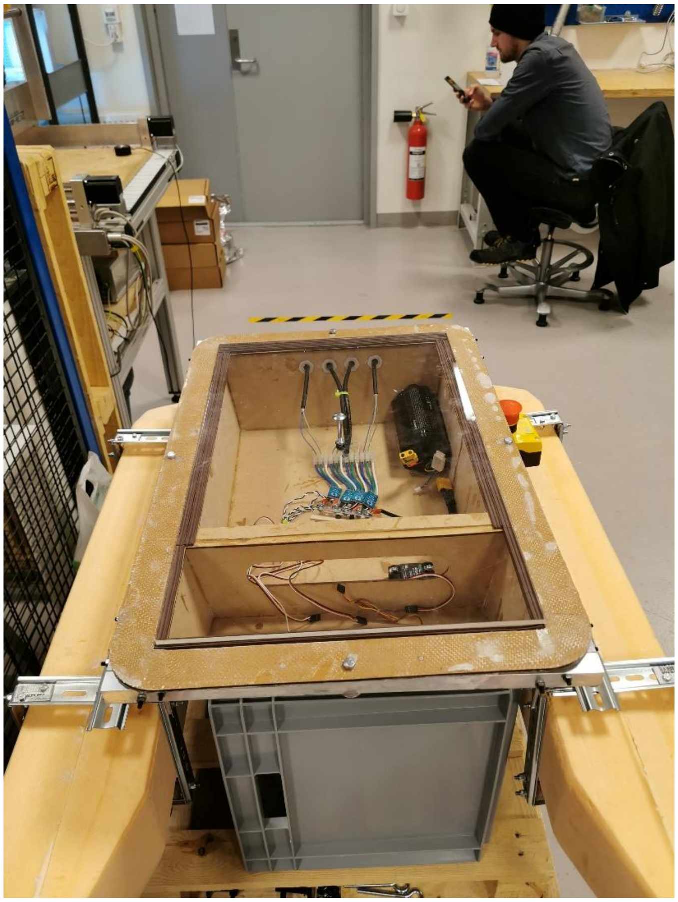
Generelt for gruppa

Den manuelle testen har blitt utsatt til Torsdag. Dette ble gjort da det ikke var tilgang på noen skruer som passet på laben.

Bestilte skruene på Onsdag 17 feb og fikk de utlevert på Mandag 22 feb og begynte med montering av elektronikk deretter.

Fikk også utlevert ZED 2 kameraet på Tirsdagen 23 feb.

For den manuelle testen trengs det bare å koble opp thrusterne og radio reciveren til pixhawken. Dette vil foregå på Torsdag 25 feb.



Denne ukens mål er å få kjørt båten på vann for første gang, begynt med oppkobling av kameraet samt begynne med ROS implementering rettet opp mot GPS. Hvis manuell test er vellykket, gir dette oss mulighet for å fokusere mer mot software.

Ukens offisielle gruppemøte ble denne gangen flyttet til Onsdag og dermed noe manglende utfylling av ukeplanen.

Planlagt arbeid de neste dagene:

Onsdag:

- Møte med veilederne.
- Gruppemøte.
- Forhøre med Anders om bestillinger
- Kamera oppsett

Torsdag:

- Manuell test.
- Planlegge montering av kamera på båt.

Fredag:

03.03.2021

Fremgangsrapport for gruppe 14 USV Sjødrone

Gruppemedlemmer:

- **San**
- **Markus**
- **Magnus**

San

Forrige ukes status:

- Koblet opp GPS via pixhawk og mission planner.
    - Fått ut signal og sett på driftingen.
- Begynt å se på ROS/MAVROS package iforbindelse med GPS

Mål fra forrige uke:

- Koble opp GPS i ROS.
    - Ta inn GPS signalet og lese ut via ROS.

- Se på/teste filter metoder

Status:

- *ROS programmering av kalman filter for GPS*
- *Koble opp ROS program via PIXHAWK 4*
-

Mål for de neste sju dagene:

- Videre arbeid med ROS programmering, koble opp pixhawk fra QGroundControl og lese av data.
- Hvis fort ferdig med punkt over, fortsette med signal filter for forbedring.
- Integrere program med PIXHAWK

Markus

Forrige ukes status:

- *Har fått utlevert kamera.*
- *Har fått laget ferdig motorfester*
- *Møtt Rahul for å bli kjent med ROS*

Mål fra forrige uke:

- Koble opp Zed 2 kamera
- Oppdatere firmware på kamera
- Forbedre datasett
- Manuell testkjøring

Status:

- *Vært med på test av manuell styring*
- *Funnet ut at det er mulig å bruke telefonen til å ta bilder med kameraet.*
- *Kontaktet en labansvarlig for å prøve å få printet motorfestene i ABS eller PETG, men dette ville han ikke la oss gjøre.*
- *Funnet ut at det er mulig å printe motorfestene i ASA istedenfor ABS dersom dette materialet er tilgjengelig for oss.*

Mål for de neste sju dagene:

- Koble opp Zed 2 kamera
- Oppdatere firmware på kamera
- Ta bilder av bøyer
- Slutte å prokrastinere
- Printe ut nye motorfester i et vannresistent materiale. Eventuelt et ekstra sett i PLA som vi byttter til rett før konkurransen

Magnus

Forrige ukes status:

- Montert thrusterne samt tilhørende ESCer. Lagd kobling opp mot batteriet.
- Møtt Rahul (NAVO NTNU leder) for videre samtaler om elektronisk oppsett.
- Gått igjennom ROS tutorialsene for å ha en grei forståelse før jeg begynner

Mål fra forrige uke:

- Få bestilt opp det siste av utstyr vi trenger. Pågående prosess med Anders
  - Telemetry radio.
  - USB WiFI forsterker.
  - Ny GPS HEX Here 3.
- Manuell test
- Begynne med implementering av ROS packages.

Status:

- Gjort siste forberedelser og gjennomført manuell test
- Begynt oppsett av ROS knyttet opp mot Jetson med ROS packages vi tenker å benytte oss av
  - MavROS
  - ZED 2 ROS Wrapper
  - ROS QGoundControl --> link mellom Jetson og pixhawk for å sende og motta data.
- Fortsette samtaler med Anders om deler.

Mål for de neste sju dagene:

- Fortsette med oppsett av ROS knyttet opp mot Jetson TX2 og pixhawk
- Koble opp hardware krav for konkurransen, typ nødstopp og evt annet.

Generelt for gruppa

Manuell test har blitt utført. Båten fløyt godt i vannet og brøyt vannoverflaten godt både frem og bakover.

Mandag til Onsdag har gått til Industri 4.0 og de tilhørende innleveringer i faget.

Video fra manuell test.

https://drive.google.com/file/d/1KtadfEdReLEcgjmJTykQhxo9dz0kpbQz/view

https://drive.google.com/file/d/1Kr9l5b-30j47nrcLkCJpPgi16nsMR-DY/view?usp=sharing

Ukens mål

Mandag til Onsdag har blitt brukt mye til Industri 4.0 forelesninger og lab øvninger i 3DExperience. Gjenværende uke kommer til å brukes til å komme bedre i gang med software siden både mtp kamera og med innhenting av data fra pixhawken.

Noe gjenværende hardware krav som vil utføres senere på dagen når hodet ikke er helt klar for software utvikling. Kravene som mangler er blant annet en manuell nødstopp montert på båten, og å lage en krets for å kunne ta over styringen fra fjernkontrollen.

Mangler også feste for kamera.

Neste Mandag til Onsdag vil igjen mye av tiden forsvinne til Industri 4.0 forelesninger. Siste forelesning for semesteret.

Planlagt arbeid de neste dagene:

Onsdag:

- Industri 4.0

Torsdag:

- ROS – Jetson software fortsettelse
- ROS - QGroundControl
- ZED 2 Firmware uppdate
- Jobbe med hardware krav

Fredag:

- ROS - QGroundControl
- Jobbe med hardware krav
- Planlegge kamera feste

Lørdag:

- Kamera feste konstruksjon
- Software hvis tid til overs.
- Jobbe med hardware krav

Søndag:

- Software utvikling
- Tunglab stengt.

Mandag:

- Avlesning av GPS data via ROS fra QGroundControl
- Industri 4.0

Tirsdag:

- Industri 4.0

Onsdag:

- Industri 4.0

09.03.2021

Fremgangsrapport for gruppe 14 USV Sjødrone

Gruppemedlemmer:

- **San**
- **Markus**
- **Magnus**

San

Forrige ukes status:

- *ROS programmering av kalman filter for GPS*
- *Koble opp ROS program via PIXHAWK 4*

Mål fra forrige uke:

- Videre arbeid med ROS programmering, koble opp pixhawk fra QGroundControl og lese av data.
- Hvis fort ferdig med punkt over, fortsette med signal filter for forbedring.
- Integrere program med PIXHAWK

Status:

- Jobber med Mavros og Grasshopper
- Installert QGC. Q ground control
- PX4 linker med QGC
- ROS programmering for kalman filter

Mål for de neste sju dagene:

- Kalman filter programmering
- Implementere Kalman filter på GPS
- ROS implementering på PX4

Markus

Forrige ukes status:

- *Vært med på test av manuell styring*
- *Funnet ut at det er mulig å bruke telefonen til å ta bilder med kameraet.*
- *Kontaktet en labansvarlig for å prøve å få printet motorfestene i ABS eller PETG, men dette ville han ikke la oss gjøre.*
- *Funnet ut at det er mulig å printe motorfestene i ASA istedenfor ABS dersom dette materialet er tilgjengelig for oss.*

Mål fra forrige uke:

- Koble opp Zed 2 kamera
- Oppdatere firmware på kamera
- Ta bilder av bøyer
- Slutte å prokrastinere
- Printe ut nye motorfester i et vannresistent materiale. Eventuelt et ekstra sett i PLA som vi byttter til rett før konkurransen

Status:

- *Klart å koble opp zed2 kamera*
- *Oppdatert firmware på kamera*
- *Ikke rukket å ta bilder av bøyer pga. industri 4.0*
- *Skolen har ikke ASA tilgjengelig for 3D-printing.*
- *Fått satt opp et virtuelt ubuntu miljø på datamaskinen slik at det kan gjøres testing av kamera og kode.*

Mål for de neste sju dagene:

- Kontakte medstudenter og sjekke om de har privat 3D-printer og om de har ABS eller ASA som vi kan få bruke.
- Designe og lage stativ til kameraet. Dette stativet skal monteres på båten og skal være sammenleggbart slik at det blir kompakt under transport.
- Det er uvisst om det blir tid til å ta bilder av bøyer grunnet undervisning i industri 4.0, men jeg skal prøve å finne tid.
- Sette meg litt inn i det virtuelle ubuntu miljøet og lære meg standard kommandoer.

Magnus

Forrige ukes status:

- Gjort siste forberedelser og gjennomført manuell test
- Begynt oppsett av ROS knyttet opp mot Jetson med ROS packages vi tenker å benytte oss av
  - MavROS
  - ZED 2 ROS Wrapper
  - ROS QGoundControl --> link mellom Jetson og pixhawk for å sende og motta data.
- Fortsette samtaler med Anders om deler.

Mål fra forrige uke:

- Fortsette med oppsett av ROS knyttet opp mot Jetson TX2 og pixhawk
- Koble opp hardware krav for konkurransen, typ nødstopp og evt annet.

Status:

- Fått kontakt mellom Jetson og ZED 2 via ROS
- Bestilt og utlevert ny GPS samt radio reciver for manuell kontroll (en del vi lånte av NAVO NTNU gruppen frem til nå)
- Kommet lengre med hardware krav, lite som mangler.
- Tid forsvunnet i Industri 4.0

Mål for de neste sju dagene:

- Here3 GPS ROS kommunikasjon
- Jetson Zed 2 kommunikasjon

Generelt for gruppa

I uken som har vært har mye tid gått til forelesning i faget Industri 4.0 samt eksamensinnleveringer i faget.

Av hardware har nødstopp knappen blitt montert samt layout av materiale er blitt fastsatt.

Fått utlevert GPS av typen Here3+.

Bildet under er noe utdatert og mangler blant annet monteringen av nødstoppen.

Av software har vi fått koblet opp og fått kommunikasjon mellom Jetson TX2 og Zed 2 kamera via Zed ROS_Wrapper.

Ukens mål

Ble nevnt i forrige fremgangsrapport at denne uken vil Mandag til Onsdag forsvinne til Industri 4.0, dette viste seg å bli utvidet til hele uken da forelesningene også ble utvidet til hele uken.

Legge til rette for å fullføre det av hardware krav som gjenstår men hovedfokus blir software fremover.

Software blir fokus å fortsette med koblingen mellom jetson og Zed 2. Her med et fokus på å få satt på plass YOLO via Darknet ROS package.

Med pixhawken blir fokus å koblet på den nye GPSen og få signal utlest via QGroundControl. For så i fremtiden kunne benytte oss av waypoints til å navigere.

Planlagt arbeid de neste dagene:

Onsdag:

- Industri 4.0

Torsdag:

- Industri 4.0

Fredag:

- Industri 4.0

Lørdag:

- Planlegging av kamerafeste.
- Pixhawk - Here 3 oppkobling/testing.

Søndag:

- Software utvikling.
- Tunglab stengt.

Mandag:

- Avlesning av GPS data via ROS fra QGroundControl.
- YOLO/Darknet implementering på Jetson mha Zed 2.

Tirsdag:

- YOLO/Darknet implementering på Jetson mha Zed 2.
- Få på plass/legge til rette for kamera feste.

Onsdag:

- Fremgangsrapport

16.03.2021

Fremgangsrapport for gruppe 14 USV Sjødrone

Gruppemedlemmer:

- **San**
- **Markus**
- **Magnus**

San

Forrige ukes status:

- Jobber med Mavros og Grasshopper
- Installert QGC. Q ground control
- PX4 linker med QGC
- ROS programmering for kalman filter
-

Mål fra forrige uke:

- Kalman filter programmering
- Implementere Kalman filter på GPS
- ROS implementering på PX4
-

Status:

- . Begynner med Klaman filter programmering
- Installert Ardupilot sitl for ROS
- PX4 blir implementere med Ardupilot sitl gjennom ROS

Mål for de neste sju dagene:

- .Forsatt med Kalman filter programmering
- Test Kalman filter
- Begynne med logikken til start prosedure med tanke på speed course

Markus

Forrige ukes status:

- *Klart å koble opp zed2 kamera*
- *Oppdatert firmware på kamera*
- *Ikke rukket å ta bilder av bøyer pga. industri 4.0*
- *Skolen har ikke ASA tilgjengelig for 3D-printing.*
- *Fått satt opp et virtuelt ubuntu miljø på datamaskinen slik at det kan gjøres testing av kamera og kode.*

Mål fra forrige uke:

- Kontakte medstudenter og sjekke om de har privat 3D-printer og om de har ABS eller ASA som vi kan få bruke.
- Designe og lage stativ til kameraet. Dette stativet skal monteres på båten og skal være sammenleggbart slik at det blir kompakt under transport.
- Det er uvisst om det blir tid til å ta bilder av bøyer grunnet undervisning i industri 4.0, men jeg skal prøve å finne tid.
- Sette meg litt inn i det virtuelle ubuntu miljøet og lære meg standard kommandoer.

Status:

- Tid forsvunnet i Industri 4.0
- Fått printet ut thruster festene i SLA takket være Øystein Bjelland
- Begynt med å sette meg inn i Ubuntu miljøet
- Begynt design av kamera feste

Mål for de neste sju dagene:

- Fortsette med design av kamera feste
- Fortsette med ubuntu/jetson

Magnus

Forrige ukes status:

- Fått kontakt mellom Jetson og ZED 2 via ROS
- Bestilt og utlevert ny GPS samt radio reciver for manuell kontroll (en del vi lånte av NAVO NTNU gruppen frem til nå)
- Kommet lengre med hardware krav, lite som mangler.
- Tid forsvunnet i Industri 4.0

Mål fra forrige uke:

- Here3 GPS ROS kommunikasjon
- Jetson Zed 2 kommunikasjon

Status:

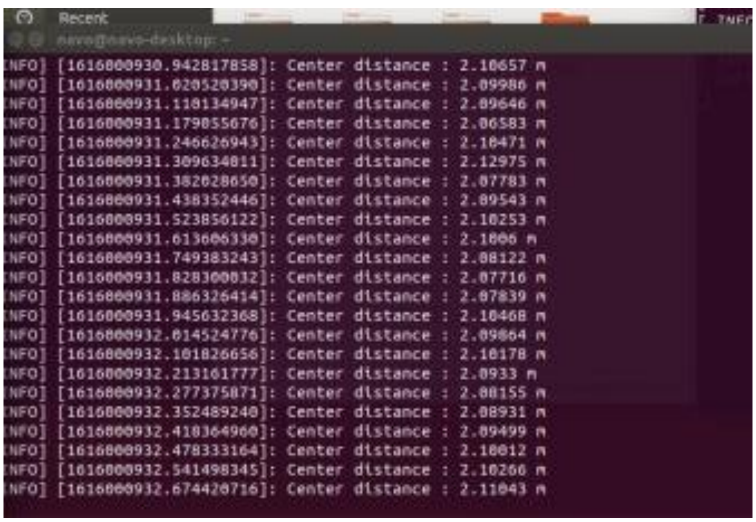
- Jetson Zed 2 kommunikasjon fungerer
    - Objekt deteksjon testet og fungerer
    - Dypde avlesning fungerer
- Noe debugging av Here3 GPS ROS kommunikasjon gjenstår før dette fungerer

Mål for de neste sju dagene:

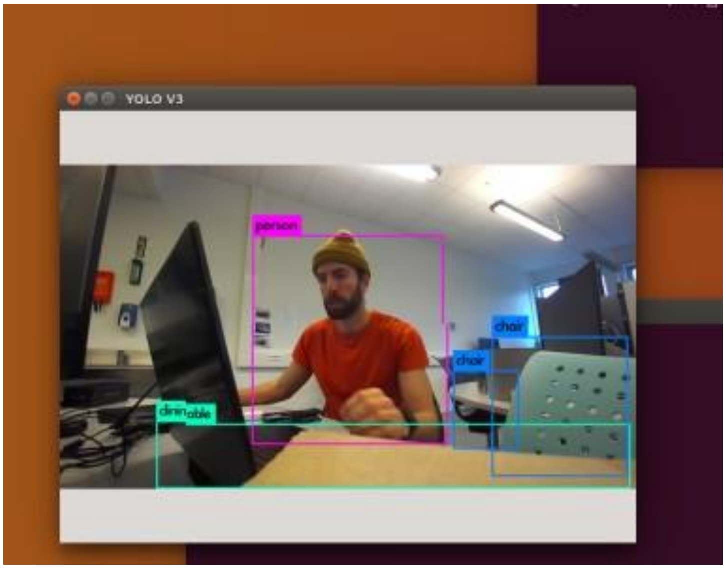
- Implementere egen parametre og weight filer for bildedeteksjon
- Here3 GPS ROS kommunikasjon

Generelt for gruppa

Som sist har mye av tiden i uken som har vært gått til å fullføre ulike Industri innleveringer. Likevel har vi fått jobbet noe og hatt noe fremgang på software siden. På hardware har vi ikke jobbet noe i uken som har vært. Vi har fått koblet opp software komponentene med hverandre og kjørt de programmene med standard innstillinger. Dette vil si at vi får lest ut dybde data fra zed 2 i meter og fåt testet det med darknet og YOLO deteksjon algoritme.



Dybde utlesning fra ZED 2 kamera via Jetson TX2



Objekt deteksjon via med Zed 2 og Jetson via ROS med standard YOLO configuration og weights.

Ukens mål

Målet for de neste dagene blir og få byttet ut standard configuration og standard weights i YOLO med filene fra forrige års gruppe. Samt vil vi begynne med styre logikken. Dette gjøres i python i ROS.

Parallelt som dette pågår vil vi begynne med logikken med start prosedure. Som nevnt er den første banen vi planlegger og løse speed course. Hvor viktigste delen for å løse dette blir å sette korrekt waypoints. Her planlegger vi å bruke kamerat til å lokalisere snu punktet også bruke pixhawks innebygde kommandoer for å sette waypoints den retningen som ble lokalisert.

Av hardware vil vi hvis vi får utlevert de to nye floating devices begynne planleggingen av å smøre på et lag epoxy for å gjøre den hakket mer robust og vanntett.

Samt prøve å få på plass kamera feste slik at vi kan begynne og planlegge for å få testet komplette båten på vannet.

Planlagt arbeid de neste dagene:

Onsdag:

- Industri 4.0
- Zed 2 – Jetson Oppkobling

- GPS – MAVROS kommunikasjons kommandoer

Torsdag:

- Rapport
- Implementasjon av egendefinerte weights og configuration files
- Kamera design

Fredag:

- Software utvikling

Lørdag:

- Skrei fiske hvis fint vær :)

Søndag:

- Software utvikling.
- Tunglab stengt.

Mandag:

- Legge på coating på floating devices
- Software - utvikling

Tirsdag:

- Legge på coating på floating devices
- Software - utvikling
- Begynne montering av kamera feste.

Onsdag:

- Veiledermøte

23.03.2021

Fremgangsrapport for gruppe 14 USV Sjødrone

Gruppemedlemmer:

- **San**
- **Markus**
- **Magnus**

San

Forrige ukes status:

- . Begynner med Klaman filter programmering
- Installert Ardupilot sitl for ROS
- PX4 blir implementere med Ardupilot sitl gjennom ROS
-

Mål fra forrige uke:

- .Forsatt med Kalman filter programmering
- Test Kalman filter
- Begynne med logikken til start prosedure med tanke på speed course
-

Status:

- *Fortsatt programmere med Kalman filter*
- Begynner med set point programmering
- *Programmere for Pixhawk*

Mål for de neste sju dagene:

- Ferdig programmere med Kalman og set point programmering
- Begynner med å tester programmer

Markus

Forrige ukes status:

- Tid forsvunnet i Industri 4.0
- Fått printet ut thruster festene i SLA takket være Øystein Bjelland

- Begynt med å sette meg inn i Ubuntu miljøet
- Begynt design av kamera feste

Mål fra forrige uke:

- Fortsette med design av kamera feste
- Fortsette med ubuntu/jetson
-

Status:

- Forrige uka har gått bort til verv i Studenttinget NTNU hvor det bar oppmøte i Trondheim fra tirsdag til lørdag.
- Har funnet et grunnkonsept som skal fungere for kamerafeste, men trenger da tilgang på tunglabb for laserkutter, div. Verktøy og matereale, og Manulab for 3D-printer med PLA og TPU.
- Kommet greit i gang med 3D-modellen av kamerafestet, men vil ikke bruke mer tid på det om jeg ikke får tilgang på de ressurser jeg trenger.
- Har sendt forespørsel om bruk av manulab via mail.
- Har sendt forespørsel om bruk av tunglab via mail.

Mål for de neste sju dagene:

- Få produsert kamerafestet
- Snakke med noen venner om jeg eventuelt kan få bruke deres 3D-printere istedenfor skolens printere. Jeg vil helst unngå dette da det går utover deres personlige resurser.
- Siden jeg nylig har vært i Trondheim og har tatt fly/buss, har jeg satt meg selv i hjemmekarantene. I løpet av uka skal jeg testes for COVID-19 og dersom resultatet er negativt vil jeg forsøke å få tak i kameraet som per dags dato ligger hos Magnus.
- Gå gjennom mye ROS-tutoreals.
- Lage grunnkoden for å hente parametere fra ZED 2 kamera.

Magnus

Forrige ukes status:

- Jetson Zed 2 kommunikasjon fungerer
    - Objekt deteksjon testet og fungerer
    - Dypde avlesning fungerer
- Noe debugging av Here3 GPS ROS kommunikasjon gjenstår før dette fungerer

Mål fra forrige uke:

- Implementere egen parametre og weight filer for bildedeteksjon
- Here3 GPS ROS kommunikasjon

Status:

- Begynt med trening av parametrene. Dette måtte gjøres på nytt da weight filen fra forrige gruppe ikke var blant filene dems og ikke noe de lenger hadde liggende på sine pcer.
- Fortsatt noe mangler på kommunikasjonen mellom Here3 GPS og ROS/TX2. Brukt mer tid enn forventet med kamera.

Mål for de neste sju dagene:

- Fortsette med treningen av kamera. Dette gjøres i bakgrunnen og parallelt med annet arbeid.
- Videre fremgang på rapport
- Bruke tid på GPS kommunikasjonen nå som kamera delen er fungerende.

Generelt for gruppa

Uken ble litt avbrutt av nye korona tiltak. Vi kommer fremover å jobbe mer individuelt og for hver av oss. Med unntak av arbeid som må gjøres på båten som da vil bli gjort på tunglaben. Som et tiltak på de nye restriksjonene kommer vi til å kjøre muntlig teams-møte hver andre dag for å høre litt om status på hverandres punkter, i tillegg til chatgruppen vi har.

Vi har gitt en grov fordeling til hvert av gruppe medlemmene når det kommer til kodingen. Her da med fokus på innhenting av info fra ROS publications og subscriptions.

Ellers var det liten set back at parameter filen fra gruppen tidligere ikke inneholdt noe weights. Forhørt med dem og dette var en fil de ikke lenger hadde lagret noe sted.

Ukens mål

Ved slutten av uken har vi som mål at grunnkoden for innhenting av data fra sensorene skal være ferdig. Dette vil sette oss opp i en god posisjon til å kunne gjennomføre ny test med blåser og hele pakken, da med fokus på speed course som vi har som første utfordring å løse. Til denne banen kommer vi til å benytte oss av PixHawks egen funksjon for set av waypoints.

Planlagt arbeid de neste dagene:

30.03.2021

Fremgangsrapport for gruppe 14 USV Sjødrone

Gruppemedlemmer:

- **San**
- **Markus**
- **Magnus**

San

Forrige ukes status:

- *Fortsatt programmere med Kalman filter*
- Begynner med set point programmering
- *Programmere for Pixhawk*
-

Mål fra forrige uke:

- Ferdig programmere med Kalman og set point programmering
- Begynner med å tester programmer

Status:

- *Fk*

Mål for de neste sju dagene:

- Fr

Markus

Forrige ukes status:

- Forrige uka har gått bort til verv i Studenttinget NTNU hvor det bar oppmøte i Trondheim fra tirsdag til lørdag.
- Har funnet et grunnkonsept som skal fungere for kamerafeste, men trenger da tilgang på tunglabb for laserkutter, div. Verktøy og matereale, og Manulab for 3D-printer med PLA og TPU.
- Kommet greit i gang med 3D-modellen av kamerafestet, men vil ikke bruke mer tid på det om jeg ikke får tilgang på de ressurser jeg trenger.
- Har sendt forespørsel om bruk av manulab via mail.
- Har sendt forespørsel om bruk av tunglab via mail.

Mål fra forrige uke:

- Få produsert kamerafestet
- Snakke med noen venner om jeg eventuelt kan få bruke deres 3D-printere istedenfor skolens printere. Jeg vil helst unngå dette da det går utover deres personlige resurser.
- Siden jeg nylig har vært i Trondheim og har tatt fly/buss, har jeg satt meg selv i hjemmekarantene. I løpet av uka skal jeg testes for COVID-19 og dersom resultatet er negativt vil jeg forsøke å få tak i kameraet som per dags dato ligger hos Magnus.
- Gå gjennom mye ROS-tutoreals.
- Lage grunnkoden for å hente parametere fra ZED 2 kamera.
-

Status:

- .

Mål for de neste sju dagene:

- .

Magnus

Forrige ukes status:

- Begynt med trening av parametrene. Dette måtte gjøres på nytt da weight filen fra forrige gruppe ikke var blant filene dems og ikke noe de lenger hadde liggende på sine pcer.
- Fortsatt noe mangler på kommunikasjonen mellom Here3 GPS og ROS/TX2. Brukt mer tid enn forventet med kamera.

Mål fra forrige uke:

- Fortsette med treningen av kamera. Dette gjøres i bakgrunnen og parallelt med annet arbeid.
- Videre fremgang på rapport
- Bruke tid på GPS kommunikasjonen nå som kamera delen er fungerende.

Status:

- .

Mål for de neste sju dagene:

- .

Generelt for gruppa

Uken ble litt avbrutt av nye korona tiltak. Vi kommer fremover å jobbe mer individuelt og for hver av oss. Med unntak av arbeid som må gjøres på båten som da vil bli gjort på tunglaben. Som et tiltak på de nye restriksjonene kommer vi til å kjøre muntlig teams-møte hver andre dag for å høre litt om status på hverandres punkter, i tillegg til chatgruppen vi har.

Vi har gitt en grov fordeling til hvert av gruppe medlemmene når det kommer til kodingen. Her da med fokus på innhenting av info fra ROS publications og subscriptions.

Ellers var det liten set back at parameter filen fra gruppen tidligere ikke inneholdt noe weights. Forhørt med dem og dette var en fil de ikke lenger hadde lagret noe sted.

Ukens mål

Ved slutten av uken har vi som mål at grunnkoden for innhenting av data fra sensorene skal være ferdig. Dette vil sette oss opp i en god posisjon til å kunne gjennomføre ny test med blåser og hele pakken, da med fokus på speed course som vi har som første utfordring å løse. Til denne banen kommer vi til å benytte oss av PixHawks egen funksjon for set av waypoints.

Planlagt arbeid de neste dagene:

06.04.2021

Fremgangsrapport for gruppe 14 USV Sjødrone

Gruppemedlemmer:

- **San**
- **Markus**
- **Magnus**

San

Forrige ukes status:

- *Fortsatt programmere med Kalman filter*
- Begynner med set point programmering
- *Programmere for Pixhawk*

Mål fra forrige uke:

- Ferdig programmere med Kalman og set point programmering
- Begynner med å tester programmer

Status:

- *Fortsatt programmere med Kalman.*
- *Ferdig med Set point programmering og navigering*
- *Skriver rapport*

Mål for de neste sju dagene:

- Integrere ferdig programmert program med båt
- Skriver rapport
-

Markus

Forrige ukes status:

- Forrige uka har gått bort til verv i Studenttinget NTNU hvor det bar oppmøte i Trondheim fra tirsdag til lørdag.
- Har funnet et grunnkonsept som skal fungere for kamerafeste, men trenger da tilgang på tunglabb for laserkutter, div. Verktøy og matereale, og Manulab for 3D-printer med PLA og TPU.
- Kommet greit i gang med 3D-modellen av kamerafestet, men vil ikke bruke mer tid på det om jeg ikke får tilgang på de ressurser jeg trenger.
- Har sendt forespørsel om bruk av manulab via mail.
- Har sendt forespørsel om bruk av tunglab via mail.

Mål fra forrige uke:

- Få produsert kamerafestet
- Snakke med noen venner om jeg eventuelt kan få bruke deres 3D-printere istedenfor skolens printere. Jeg vil helst unngå dette da det går utover deres personlige resurser.
- Siden jeg nylig har vært i Trondheim og har tatt fly/buss, har jeg satt meg selv i hjemmekarantene. I løpet av uka skal jeg testes for COVID-19 og dersom resultatet er negativt vil jeg forsøke å få tak i kameraet som per dags dato ligger hos Magnus.
- Gå gjennom mye ROS-tutoreals.

- Lage grunnkoden for å hente parametere fra ZED 2 kamera.

Status:

- Fikk testet meg for korona mandag forrige uke. Resultat: NEGATIV.
- Har gjennomført div. Python kurs for å bli kjent med programmeringsspråket.
- Har så vidt begynt å lære meg ROS
- Jeg har ikke fått printet ut kamerafeste enda da skolen har vært stengt.

Mål for de neste sju dagene:

- Fortsette med ROS tutoreals.
- Lage grunnkoden for dybdeinfo fra ZED 2 kamera.
- Printe ut / laserkutte kamerafeste dersom dette lar seg gjøre.
- Skrive på rapporten.

Magnus

Forrige ukes status:

- Begynt med trening av parametrene. Dette måtte gjøres på nytt da weight filen fra forrige gruppe ikke var blant filene dems og ikke noe de lenger hadde liggende på sine pcer.
- Fortsatt noe mangler på kommunikasjonen mellom Here3 GPS og ROS/TX2. Brukt mer tid enn forventet med kamera.

Mål fra forrige uke:

- Fortsette med treningen av kamera. Dette gjøres i bakgrunnen og parallelt med annet arbeid.
- Videre fremgang på rapport
- Bruke tid på GPS kommunikasjonen nå som kamera delen er fungerende.

Status:

- Mindre effektiv/mistet fokus etter stenging av skolen.
  - Dro kort tur hjem i påsken for å nullstille hodet.
- YOLO trening pågår fortsatt (12 dagen)
- Kode skjelettet for depth ferdig og klar til å utvikles videre.
  - Neste steg her blir å sammkjøre denne delen med San sin del.

Mål for de neste sju dagene:

- Fortsette med rapport

- Få på plass pixhawk/jetson/ROS som var originalt planlagt å få gjort forrige uke.

Generelt for gruppa

Ukens mål

Begynne å se på mulighetene på å koble individuelle delene sammen.

Planlagt arbeid de neste dagene:

- Rapport
- YOLO training pågår i bakgrunnen.
- Få på plass kamerafeste
- Pixhawk/ROS/Jetson kommunikasjon
- Samkjøre kode skjelettene som er på klar nå.

13.04.2021

Fremgangsrapport for gruppe 14 USV Sjødrone

Gruppemedlemmer:

- **San**
- **Markus**
- **Magnus**

San

Forrige ukes status:

- *Fortsatt programmere med Kalman.*
- *Ferdig med Set point programmering og navigering*
- *Skriver rapport*
-

Mål fra forrige uke:

- Integrere ferdig programmert program med båt
- Skriver rapport

Status:

- *Skriver rapport*
- *Fixer program (bug)*

Mål for de neste sju dagene:

- Integrere ferdig programmert program med båt
- Skriver rapport

Markus

Forrige ukes status:

- Fikk testet meg for korona mandag forrige uke. Resultat: NEGATIV.
- Har gjennomført div. Python kurs for å bli kjent med programmeringsspråket.
- Har så vidt begynt å lære meg ROS
- Jeg har ikke fått printet ut kamerafeste enda da skolen har vært stengt.
-

Mål fra forrige uke:

- Fortsette med ROS tutoreals.
- Lage grunnkoden for dybdeinfo fra ZED 2 kamera.
- Printe ut / laserkutte kamerafeste dersom dette lar seg gjøre.
- Skrive på rapporten.
-

Status:

- Endelig fått printet ut kamerafestet uten at printeren feilet
- Sitter i NSO, så det har gått noen dager til det
- Jobbet bittelitt med ROS/Python
- Har slipt ned de nye motorfestene og har installert to av dem på båten
- Har kommet med en ny og enklere løsning på hvordan kamerafestet skal være under frakt av båten

Mål for de neste sju dagene:

- Jobbe med rapport
- Python, ROS
- Jobbe videre med kamerafeste

Magnus

Forrige ukes status:

- Mindre effektiv/mistet fokus etter stenging av skolen.
  - Dro kort tur hjem i påsken for å nullstille hodet.
- YOLO trening pågår fortsatt (12 dagen)
- Kode skjelettet for depth ferdig og klar til å utvikles videre.
  - Neste steg her blir å sammkjøre denne delen med San sin del.
-

Mål fra forrige uke:

- Fortsette med rapport
- Få på plass pixhawk/jetson/ROS som var originalt planlagt å få gjort forrige uke.

Status:

- YOLO ferdig trent og testet til et godkjent nivå
- Skolen offisielt åpen så gruppen sitter nå samlet igjen
- Del tid brukt til rapport

Mål for de neste sju dagene:

- Fortsette med rapport
- Forhåpentligvis få båten på vannet for å teste rettet mot speed course

Generelt for gruppa

Ukens mål

Mer og mer av tiden fremover vil bli brukt mot rapporten da tiden for levering begynner å nærme seg. Flott at skolen er åpen igjen da det gjør det enklere å jobbe individuelt samt som gruppe.

Som nevnt ovenfor er kamerafeste klar som gjør at vi har forhåpninger å få båten ut på vannet for å gjøre noen tester knyttet opp mot speed course.

20.04.2021

Fremgangsrapport for gruppe 14 USV Sjødrone

Gruppemedlemmer:

- **San**
- **Markus**
- **Magnus**

San

Forrige ukes status:

- *Skriver rapport*
- *Fixer program (bug)*

Mål fra forrige uke:

- Integrere ferdig programmert program med båt
- Skriver rapport
-

Status:

- *Skriver rapport*
- *Integrere program med båt*

Mål for de neste sju dagene:

- Skriver rapport
- Tester program

Markus

Forrige ukes status:

- Endelig fått printet ut kamerafestet uten at printeren feilet
- Sitter i NSO, så det har gått noen dager til det
- Jobbet bittelitt med ROS/Python
- Har slipt ned de nye motorfestene og har installert to av dem på båten
- Har kommet med en ny og enklere løsning på hvordan kamerafestet skal være under frakt av båten

Mål fra forrige uke:

- Jobbe med rapport
- Python, ROS
- Jobbe videre med kamerafeste

Status:

- Har laget ferdig kamerafestet og fått festet det til båten. Måtte lage nytt kamerafeste da det gamle var bittelitt for lite for kameraet.
- Fått skiftet ut to av de gamle moterfestene med de nye
- En del tid har gått bort til lesing av sakspapirer til NSOs landsmøte som er nå snart

Mål for de neste sju dagene:

- Mesteparten av de neste dagene kommer til å gå bort til NSOs landsmøte da dette vare fra torsdag morgen til søndag kveld krever.
- Resterende tid er prioritert rapportskriving

Magnus

Forrige ukes status:

- YOLO ferdig trent og testet til et godkjent nivå
- Skolen offisielt åpen så gruppen sitter nå samlet igjen
- Del tid brukt til rapport

Mål fra forrige uke:

- Fortsette med rapport
- Forhåpentligvis få båten på vannet for å teste rettet mot speed course

Status:

- Mye rapport
- Testet YOLO ved sjøen
- Klar for å ta inn kordinater for set point

Mål for de neste sju dagene:

- Fortsette med mye rapport
- Iløpet av helgen få testet waypoint setting
- Mer rapport

Generelt for gruppa

Halvparten av arbeidsdagen blir nå brukt til skriving i rapport da mai kommer nærmere og nærmere. De siste to ukene har Markus fått på plass de siste detaljene for kamerafestet. YOLO modellen med kamerat har blitt testet separert fra båten til et godkjent nivå.

Angående rapporten har den kommet godt i gang. Her er planen at rapporten skal være i en tilstand hvor hovedelementene er fylt ut til den 30 april, slik at det er god tid til å korrekturlese samt gjøre endringer.

Ukens mål

Målet for uken er relativt enkel. Fortsette med hovedfokus på rapport og få sendt posisjons kommandoer over Mavlink hvor posisjoner defineres via latitude og longitude.

03.05.2021

Fremgangsrapport for gruppe 14 USV Sjødrone

Gruppemedlemmer:

- **San**
- **Markus**
- **Magnus**

San

Status:

- Mye rapport
- Gjennomført tester for å måle systemet

Mål for de neste sju dagene:

- Skrive rapport
- Tester

Markus

Status:

- Mye rapport
- Gjennomført tester for å måle systemet

Mål for de neste sju dagene:

- Rapport
- Tester

Magnus

Status:

- Mye rapport
- Gjennomført tester for å måle systemet
- Rapport

Mål for de neste sju dagene:

- Rapport
- Tester

Generelt for gruppa

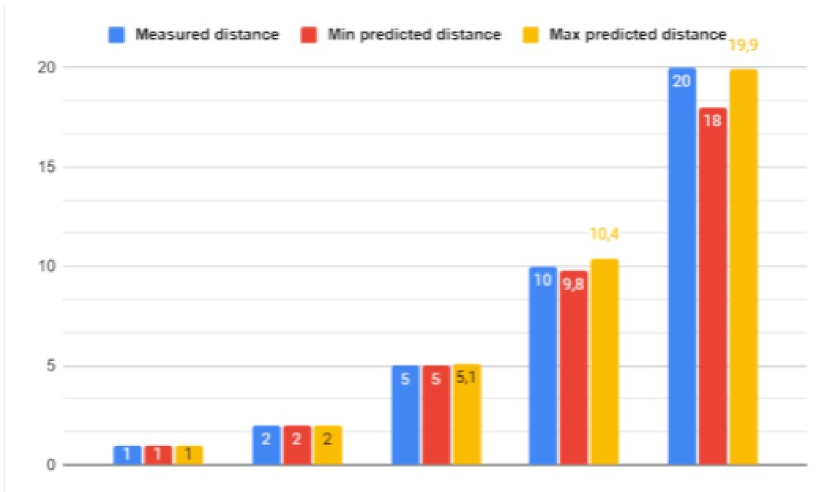
Gjennomført test med følgende resultater:

GPS Nøyaktighet over en 10minutters periode:

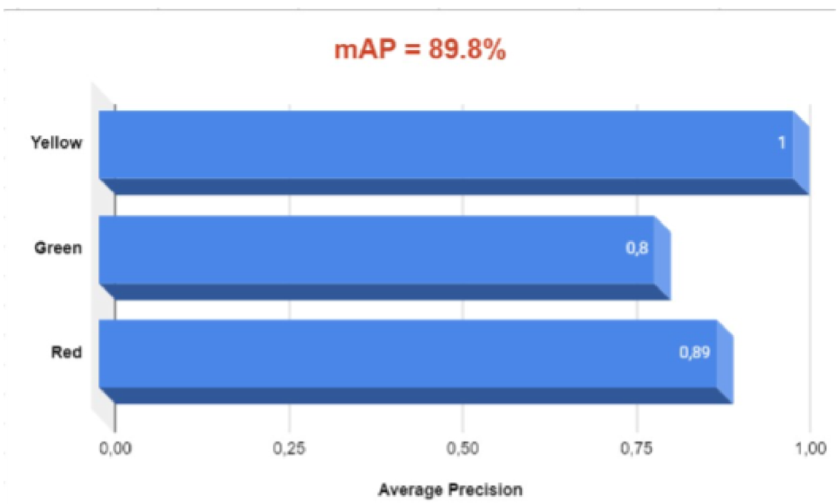
Hastighet på båten:

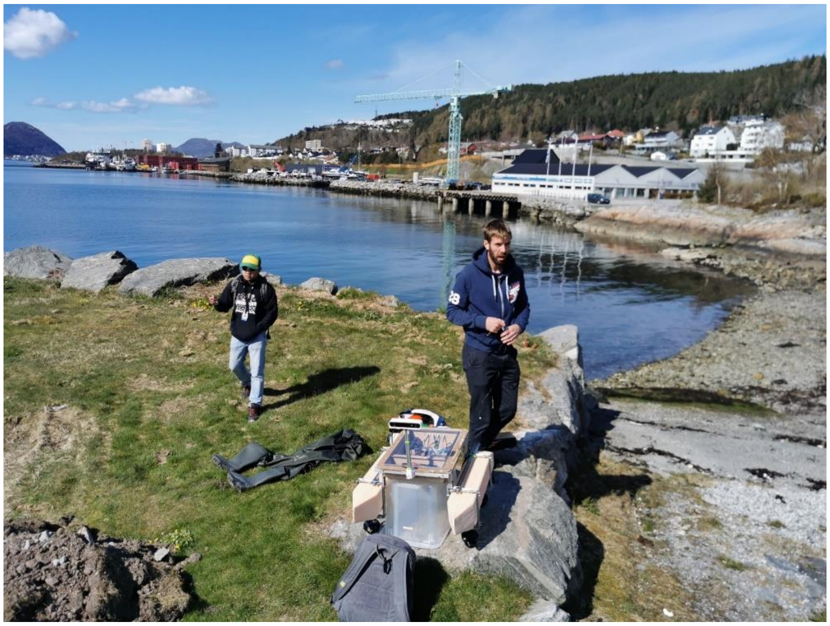
Test	Speed (m/s)
1	1,2
2	1,3

Nøyaktighet på kamera dybde målinger:

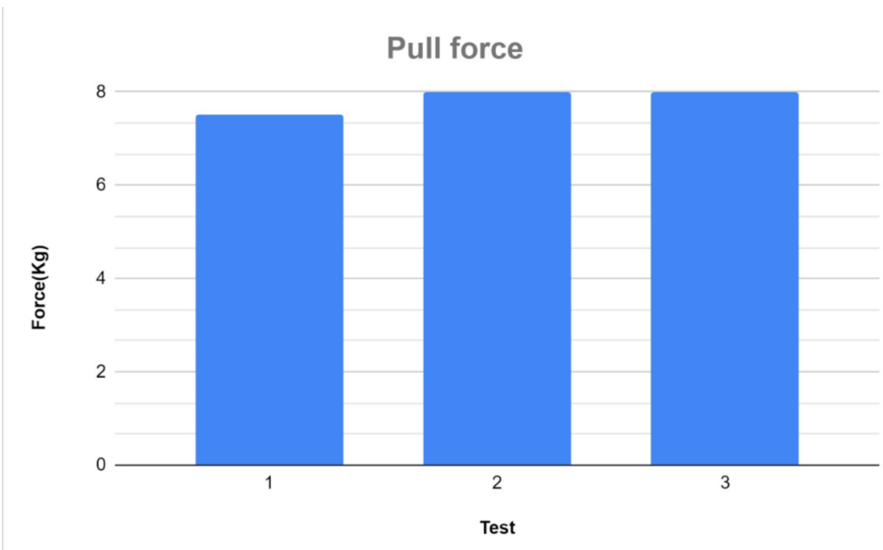


MAP score på object deteksjon modellen:





Båtens trekk kraft



Ukens mål

Videre arbeid på rapport og forberedning av nye to tester på søndag/lørdag er planen for uka. Disse to testene er tenkt å være en test av waypoint og en test for båtens egenskap til å holde seg i ro på et punkt.

Appendix J

Autodrone 2021 rules and task description

Rules and Task Descriptions

Version 0.3, Created Feb 18, 2020

AUTODRONE

The rules herein are based on the RoboBoat Rules Task Description with permission from the RoboBoat International Competition

AutoDrone Rules and Task Descriptions

1 Objective

AutoDrone is a national championship in operation of Autonomous Sea Drones (ASD) established to strengthen the maritime higher education in Norway and as a response to the increasing interest for autonomous operation of surface ships. New technology for smarter ships and new ways of solving sea-based transport more efficiently and cleaner represents challenges and opportunities for the maritime sector. The AutoDrone championship will act as an incubator for developing, testing and benchmarking innovations in the maritime domain.

This vision is achieved by providing a venue and mechanism, whereby practitioners of robotics and maritime autonomy come together at AutoDrone to share knowledge, innovate, and collaboratively advance the technology of ASD systems.

2 AutoDrone Information and Updates

All questions, comments, and suggestions should be posted on the AutoDrone Questions facebook group. Teams are encouraged to actively participate in the online community and monitor it for latest news and updates regarding the AutoDrone Championship.

3 Venue Overview

The championship in 2020 will be held at Strandpromenaden 50, Horten, in conjunction with the Kongsberg Maritime Subsea facilities there. Horten Autonomy Test arena will be used, and the area around the pier will be divided into a single competition course with 4 different missions.

Figure 1 Strandpromenaden 50 Aerial View

4 Team Village

During the competition, each team is provided with a covered workspace. Drone and other equipment will be locked up overnight. Electricity (one outlet) and internet connection (wireless) are available in the workspace.

5 Schedule

The competition is held, rain or shine. The general competition's schedule is available on the AutoDrone website. The schedule is subject to changes due to inclement weather stoppages (lightning, etc.), and safety considerations. It is the Team's responsibility to check the website for the most up to date version of the schedule.

6 Participation and Eligibility Requirements

There is no requirement for teams to be associated with a school or university. Please consider that at least 3 members are needed for safe AutoDrone operations. Faculty, industrial and governmental partners may be used to support the on-site team.

One drone per team may be entered in the competition. Each team must designate a team member as their *team leader*. The team leader is the only person allowed to speak for the team, to request drone deployment, run start, run end, or drone retrieval.

The team leader must be conversationally fluent in Norwegian or English to communicate with AutoDrone staff.

7 Registration and Fees

To participate in the competition, all teams must register via the registration responsible listed at www.autodrone.no and submit the registration fee.

8 Team Deliverables

In addition to the mission tasks, each team must document some of their efforts leading up to the competition by adding media content to the AutoDrone facebook page.

Each team leader is responsible for adhering to the instructions and deadlines listed on the AutoDrone website.

8.1 Facebook contribution

Teams must contribute to the AutoDrone facebook page with the following information:

- Team information (name and team contact information).
- Team member information (name, picture, contact information).
- Media (pictures, video, etc.) taken during development and testing.
- Link to team websites if available.

9 Competition General Requirements

The following is a list of minimal requirements for a sea drone to be permitted access to a course. **Teams that arrive at the competition failing to meet these requirements will not be permitted on the course, until they modify their drone to meet all the requirements.**

9.1 Sea Drone Requirements

- **Autonomy**: Drone shall be fully autonomous and shall have all autonomy decisions made onboard the ASD.
- **Communication**: The drone cannot send or receive any **control** information to and from Operators Control Station while in autonomous mode.
- **Deployable**: The ASD should be manually deployable.
- **Energy source**: The drone must be battery powered. All batteries must be sealed to reduce the hazard from acid or caustic electrolytes. The open circuit voltage of any battery (or battery system) may not exceed 60Vdc.
- **Kill Switch**: The drone must have at least one red button located on the drone that, when actuated, must instantaneously disconnect power from all motors and actuators.
- **Wireless Kill Switch**: In an emergency situation the operator control station must be able to actuate the kill switch on board the ASD.
- **Propulsion**: Any propulsion system may be used (thruster, paddle, etc.). However, all moving parts must have protection. For instance, a propeller must be shrouded.
- **Remote-controllable**: The drone must be remote-controllable from an operator control station.
- **Safety**: All sharp, pointy, moving or sensitive parts must be covered and marked.
- **Towable**: The drone must be towable.
- **Visual Feedback**: Teams are required to implement a visual feedback system, indicating status of their ASD. Additional information on this is available in Appendix 15.4 Visual Feedback.
- **Weight**: The entire maritime system (including UAV) shall weigh less than 70 kg.
- **Payload**: The drone must have a place to mount an action camera with an unobstructed view from the front of the drone.

9.2 Interference

Interference with course elements may result in a run termination. Any sea drone entangled in, dragging, pushing or damaging competition elements or the landscape may be deemed interfering.

All drones must stay within the bounds of their assigned course. Any drone leaving its assigned course may be deemed interfering.

9.3 Judges' Decisions

All decisions of the Judges are final.

9.4 Sea Drone Recovery

No team member is allowed in the water any time. Competition officials will be responsible for recovering lost drones. Officials will make all reasonable efforts to recover a lost drone but cannot guarantee that they will be able to do so. All teams recognize that by entering the competition, they risk damage to, or the loss of, their drone. The judges, officials, host and sponsors can take no responsibility for such damage or loss.

10 Weight and Thrust Measurements

Drones are weighed before every deployment.

Thrust is measured after the drone is deployed in the water. The thrust value used is the highest scale reading that is stable for at least two seconds. Teams may opt to repeat their thrust measurement at each deployment.

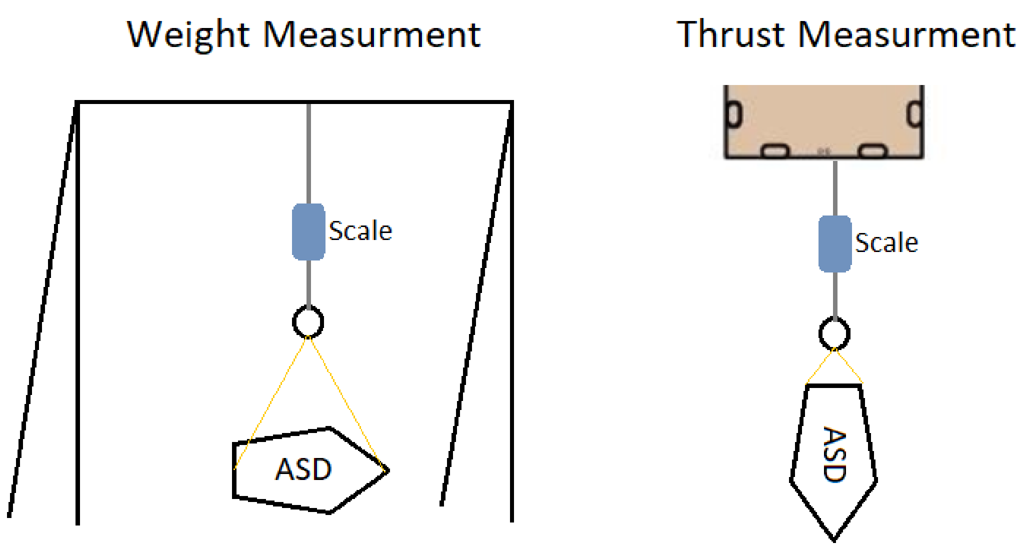


Figure 2 Weight and Thrust Measurement setup

10.1 Task scoresheet

Table 1: Weight and thrust scoresheet

Parameters	Points
ASD weight > 70kg.	**Disqualified**
Dimensions greater than: - 0.9m width or - 0.9m feet of height - 1.8 m of length	**Disqualified**
Thrust (t) vs weight (w)	*100*(t / w)*

11 Mission Tasks

Teams demonstrate advanced autonomous drone behavior through performance of the mission tasks. Mission tasks will be performed according to the championship schedule. One task at a time with one team at a time in a given order. A single GPS position, representing the center of the task's 'entrance' will be provided at the competition (in decimal degree format).

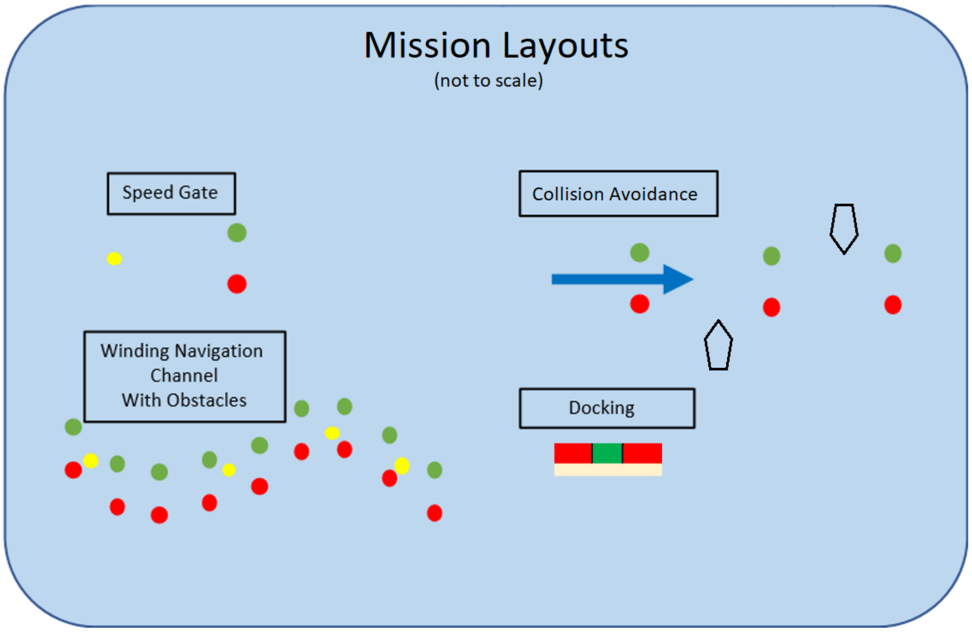
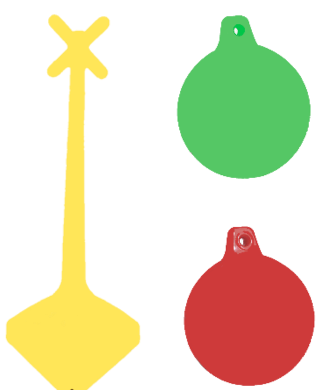


Figure 3 Mission layouts

11.1 Buoys

The championship will use only three types of buoys, two round of equal size and shape but of different color (Ø155 mm), and one yellow beach marker buoy:



Red buoy: Art.nr. 25-017 from Biltema
Green buoy: Art.nr. 25-017 from Biltema spraypainted green (Art.nr. 36-577 from Biltema)
Yellow buoy: from Sommerbutikken

Figure 4 Championship buoys

11.2 Tries

Each team will be able allowed two attempts for each mission task, the time of the attempts will be scheduled.

11.1 Obstacle Channel

Successful completion of the Obstacle Channel task demonstrates the ability to sense and maneuver through a complex path, staying within the defined pathway, and avoiding contact with obstacles along the way.

The drone passes between multiple sets of gates designated by pairs of red and green buoys. The entire drone should pass through all sets of the gates without touching the buoys. To score points for a gate it must be passed correctly one time, no points are scored on subsequent passing of same gate. The drone must also avoid intermittent yellow buoys placed within the pathway described by the location of the red and green pairs of buoys (gates) hitting an obstacle buoy will give penalty points.

This is a timed mission and for every minute spent on the mission points are subtracted from an initial starting number of points.

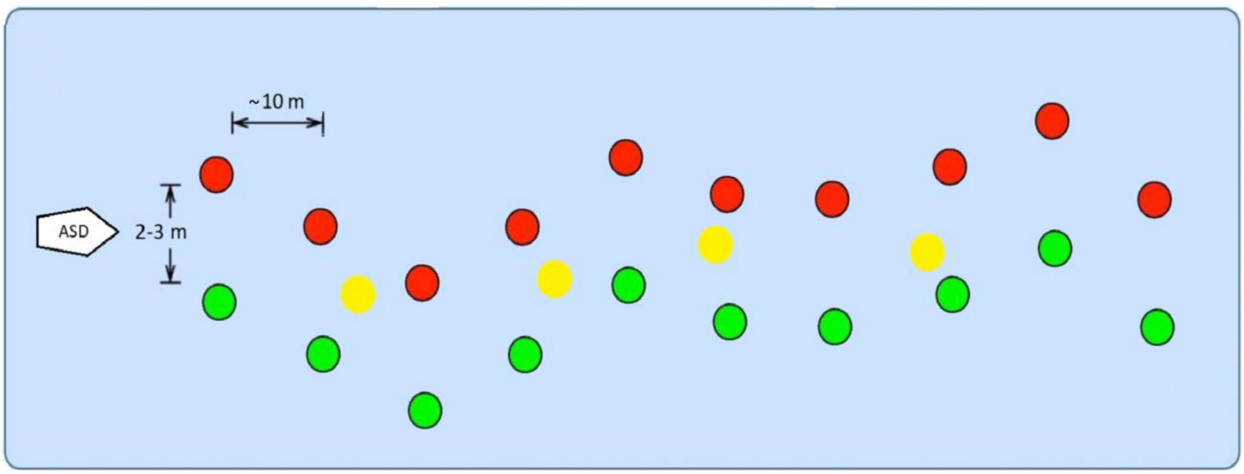


Figure 5 Obstacle Channel overview

Points		
Gate	**Obstacle**	**Time**
+5	-10	100 - 10/min

11.2 Collision avoidance

The purpose of the Collision Avoidance task is to demonstrate complex path planning. The drones must move through an area with crossing traffic from both port and starboard. The ASD should behave according to Convention on the International Regulations for Preventing Collisions at Sea (COLREGs). The direction of the arriving crossing traffic will be randomized.

The ASD must hold a speed below 4 knots and complete the mission in less than 10 minutes.

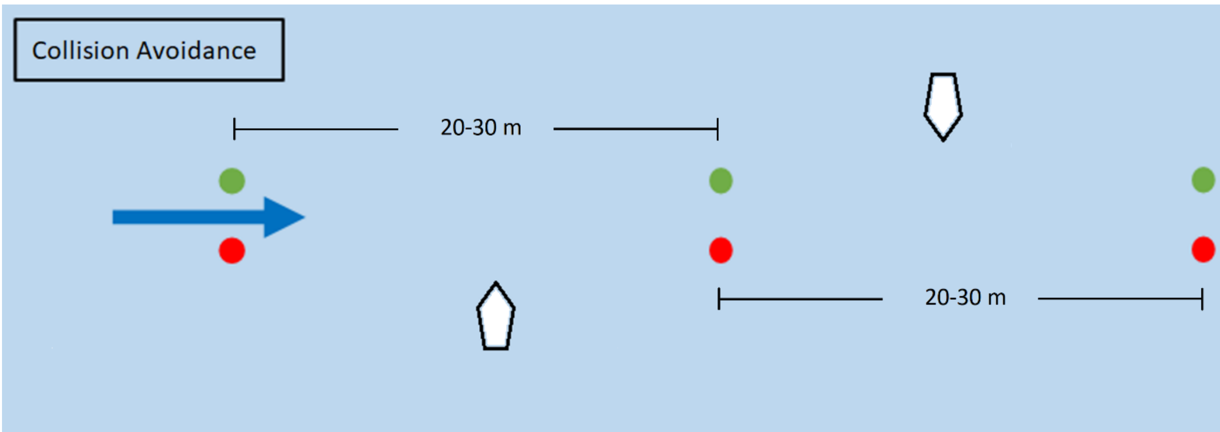


Figure 6 Collision Avoidance (crossing traffic for illustration only)

Points		
Gate	**Collision**	**COLREGs**
+10	-30	+30/rule followed

11.3 Visual Docking

Successful completion of the Visual Docking task will demonstrate the ability to localize the docking area, and maneuver into docking position in a defined area.

The docking area will be an area of a pier marked with a green and red buoy two meters apart. The ASD should navigate to the area between the buoys and dock with its port or starboard side to the pier. It should be docked at least for 30 sec (can use thrusters to keep the ASD at dock). Then after this time the ASD must leave the dock.

Points are given for reaching the docking area, docking correctly and then for every second the ASD is kept docked.

The mission must be completed in less than 10 minutes.

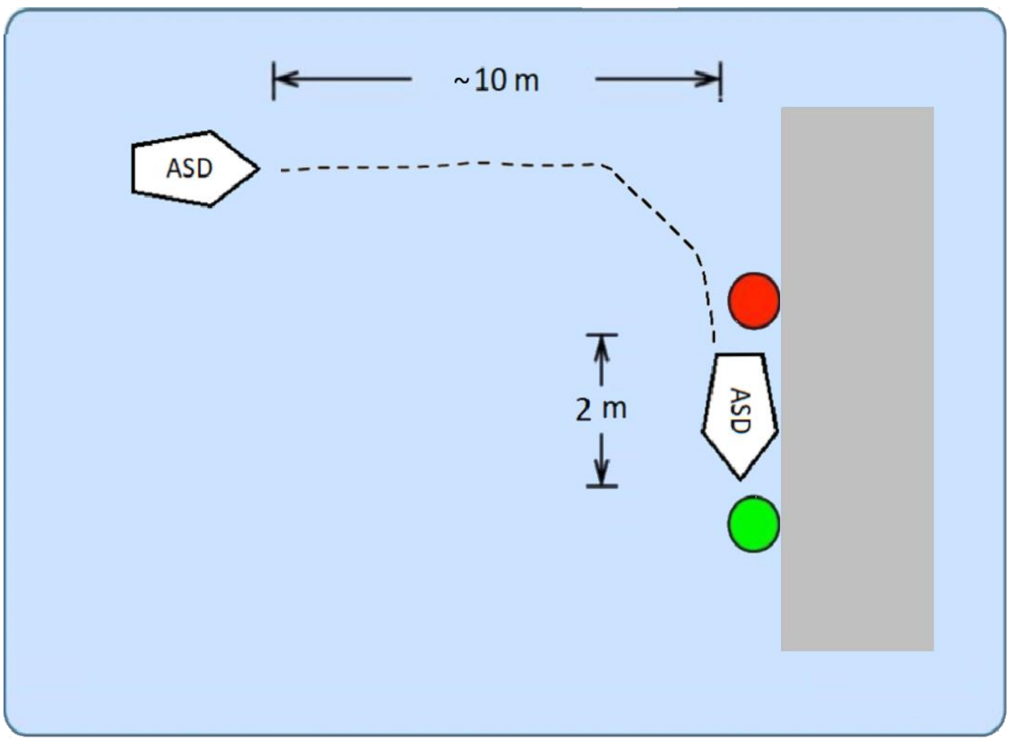


Figure 7 Docking (illustration only)

Points		
Dock Reached	**Docking correct**	**Docking time**
+10	+30	+ 2 /sec docked

11.4 Speed Gate

Successful completion of the Speed Gate task demonstrates the ASD's hull form efficiency coupled with its propulsion system, and the resulting maneuverability. Furthermore, it demonstrates object recognition and decision making with respect to sensing the task elements.

The drone must enter through the gate buoys, go around the Mark buoy (counterclockwise or clockwise), and exit through the same gate buoys, as quickly as possible. The gate buoys are moored 2-3m apart, and the Mark buoy is placed 20-30m from the gate buoys.

This is a timed challenge. Time starts when the bow (front) of the drone crosses the Gate buoys (entry) and stops when the stern (back) of the drone crosses the Gate buoys (exit).

Completing the course in less than 10 minutes gives 20 points, additional points are given to the three fastest ASDs.

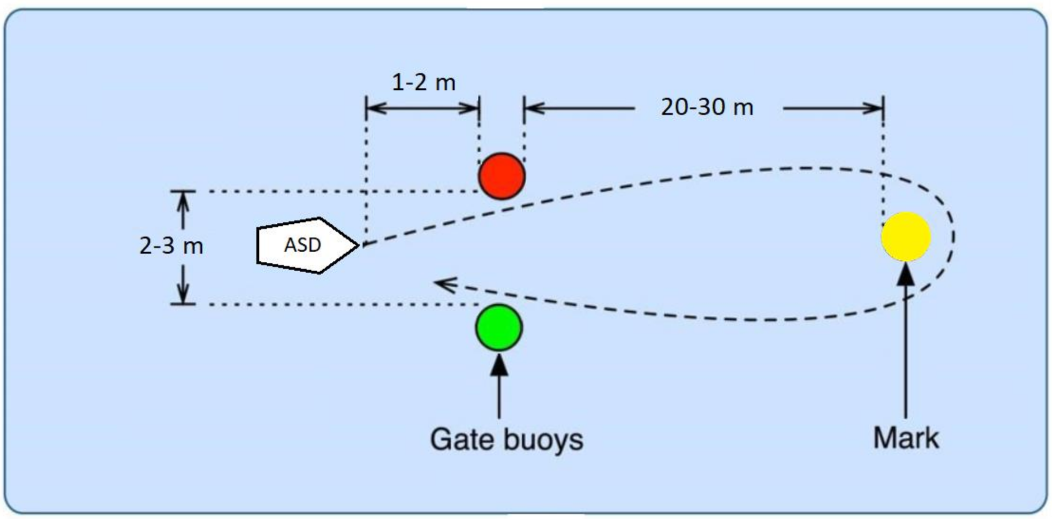


Figure 8 Speed Gate (Illustration only)

Scoring	
Time	**Points**
Fastest	80
2nd	50
3rd	20

12 Communication with Championship Mission Task Server

To earn points for their mission task completions teams must be able to communicate with the championship mission task server. All competitor Operator Control Stations (OCS) will be required to have the ability to connect to a wired Ethernet network (provided) by competition organizers) and request an IP address from a DHCP server. Once connected, they should establish a TCP connection to a server, with a given address and port number and provide GPS data through an NMEA 0183 sentence (to be used with OpenCPN chart plotter).

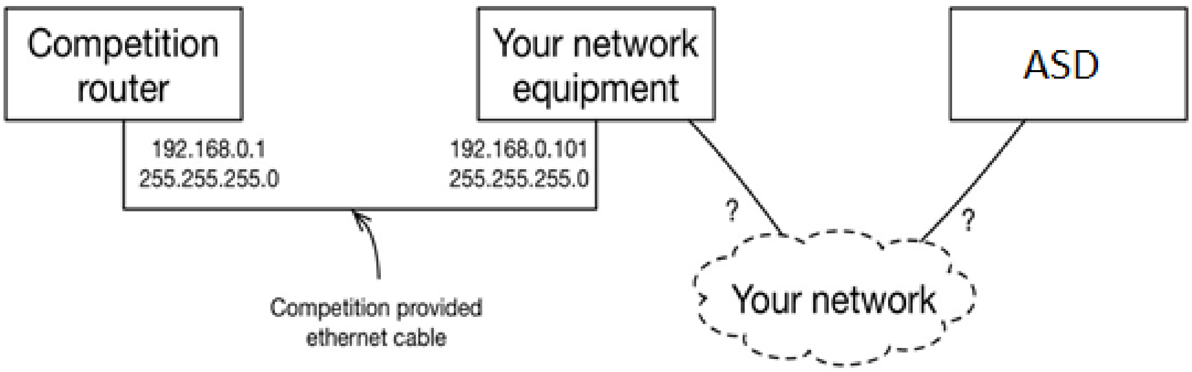


Figure 9 Virtual network connectivity overview

13 Appendices

13.1 Visual Feedback

This section describes a lighting system that will serve as a visual status indicator for the team's drone.

With unmanned systems being integrated into everyday use, it is safety critical for these systems to indicate their operational status. Resources and general guidelines will be outlined in this document to permit teams to acquire, integrate, and test a system that meets the safety requirement set forth for AutoDrone operations.

The lighting system shall consist of, at minimum, three lights: Red, Amber/Yellow, and Green/Blue. Lights shall be arranged in a vertical configuration and mounted such that they provide a **360 degree daylight visibility**, when viewed from shore or nearby vessel.

Lighting system colors shall correspond with the applicable mode of the team's autonomous system, as indicated in table below. The lights may be flashing or steady on/off according to the state of the system.

Table 2: Light color and correlating modes

Color	Mode
Red	E-Stop active (propulsion disabled)
Amber/Yellow	Tele-Operation / Manual Operation
Green/Blue	Autonomous operation

Visual indication system can be purchased commercially or can be a custom array of RGB LEDs. Keeping the below specifications in mind, design and selection of the final system is the team's decision.

Figure 10 Visual Feedback Indicators (Left – Commercial; Right – Custom)

To provide visibility in sunlight, teams should use lighting systems that have clear enclosures for the light to shine through; rather than colored enclosures with standard light bulbs.

13.2 Onboard Emergency Stop

All drones must have an onboard emergency stop capable of being actuated by personnel from a support craft. For personnel safety, the switch may be triggered from a distance by a wooden or plastic pole/paddle. Keeping this in mind, teams should select rugged and reliable components for their safety system.

A large, red button should be installed in such a way that safety personnel, from the support craft can easily actuate the button. The engage/disengage button should be red in color and have a 'press to activate and turn to reset' feature. This button, momentary contact switch or not, on actuation, should cut power to the thrusters immediately on actuation. The thrusters must remain in a powered-down state until the judge gives permission for the team to reinitialize the system. An example of a suitable button is shown in Figure 11.

Figure 11 Example of a Kill Switch