KTH Mechatronics Advanced Course

MF2059, HT 2021

FINAL REPORT

Autonomous Docking Prototype

Team Candela AutoDocking

Arvid Bergvall, Malinda Blomqvist, Charles Blane Brinkley IV, Bix Eriksson, Anton Grahn, Sebastian Granlund, Anton Haga Lööf, Pontus Lundman, Simon Nilsson, Tobias Wiklund Oinonen

Candela Technology AB
Industrial Supervisor: Bruno Bosco
KTH Supervisor: José Manuel Gaspar Sánchez

[2021-12-19]

Abstract

Candela is an innovative boat company based out of Stockholm that produces all-electric, hydrofoil speed boats. They have recently started exploring solutions for public transportation and autonomous navigation and their first vessel to utilize such technology is the ”P-30”; a ship that will replace the current fleet of diesel SL commuter ferries in Stockholm. The goal of this project is to explore autonomous solutions specifically for the docking sequence of the P-30. As such, the scope involves designing, fabricating, and testing a prototype P-30 to analyse how well the conceived methodology achieves autonomous docking in the same circumstances as the current SL ferries.

This report displays the development of an autonomous docking boat prototype. From state of the art solutions and design concept to implementation and testing the actual boat. The final concept includes the embedded use of GNSS, stereo vision camera, and ultrasonic sensors to localize and control two trolling motors that maneuver a small Jon boat. Different functionalities were developed in subsystems and these were then integrated together toward the end of the project. This resulted in a robust testing platform with verified functionality of every subsystem and some initial success at a fully autonomous docking prototype. To continue the development of this prototype, there is an outline of suggested future plans.

Acknowledgements

We would like to thank Candela and Bruno Bosco for providing us with such a fascinating project and offering their continuous feedback and support. We also would like to thank our project coach, José Manuel Gaspar Sánchez, for his guidance and insightful advice toward the project approach and course deliverables. Similarly, thank you to the rest of the supervising staff who have given us further advice and critique that has improved our project considerably.

We would also like to thank Skeppsholmsgårdens Folkhögskola for letting us test at their facilities and Johan Nilsson for his experienced guidance and material contributions. We would like to thank Ocado Technology AB for lending us key hardware components that made this project possible as well.

AutoDocking Team,
Stockholm, December 2021

Contents

List of Figures

List of Tables

Acronyms

AAWA Advanced Autonomous Waterborne Applications. 15

AGM Absorbent Glass Mat. 20, 68, 72, 76

AHRS Attitude Heading Reference System. 12

ANS Autonomous Navigation System. 16

ASV Autonomous Surface Vehicle. 13, 14, 39

AUV Autonomous Underwater Vehicle. 13, 14

BLDC Brushless DC Motor. 27

BMS Battery Management System. 24

CAD Computer Aided Design. 82, 99

CAN Controller Area Network. 54, 78, 82, 83, 92, 94, 95, 97, 110, 114, 121, 130, 143, 147–149

CNN Convolutional Neural Network. 49, 51

DGNSS Differential Global Navigation Satellite System. 36

DMA Direct Memory Access. 110

DOF Degrees of Freedom. 30–33, 98, 99, 101

EKF Extended Kalman Filter. 13, 41

EPS Electrical Power Steering. 26

ESC Electronic Speed Controller. 84, 86, 94, 96, 125

FMCW Frequency-Modulated Continuous Wave. 47

FOV Field of View. 44, 46, 73, 146

GLONASS Global’naya Navigatsionnaya Sputnikovaya Sistema. 36, 102

GNSS Global Navigation Satellite System. 11, 35, 36, 41, 68, 70, 102, 148

GPIO General Purpose Input/Output. 92

GPS Global Positioning System. 12, 14, 36, 102

I^{2}C Inter-Integrated Circuit. 56, 57, 90, 102, 103, 110, 114, 115, 120, 147

IMU Inertial Measurement Unit. 13, 14, 37, 68, 147, 155

KF Kalman Filter. 41, 42, 68

KV Constant Velocity, No-load speed at 1 volt. 18

LiDAR Light Detection and Ranging. 11, 12, 14, 16, 44, 45, 48, 72, 73, 154

LiFePO4 Lithium Iron Phosphate. 20, 22–25

LQR Linear-Quadratic Regulator. 15, 39, 103, 117, 122, 126, 155

M2M Machine-to-Machine. 70

MIMO Multiple Input Multiple Output. 99, 102

MPC Model Predictive Control. 15, 40, 155

NED North-East-Down. 30, 31

NTRIP Networked Transport of RTCM via Internet Protocol. 36, 71

PID Proportional–Integral–Derivative. 15, 39

PLM Product Lifecycle Management. 4

PWM Pulse-Width Modulation. 86, 120, 153

RBPF Rao-Blackwellized particle filters. 49

ROS Robot Operating System. 105–107

RPM Revolutions Per Minute. 117

RTCM Radio Technical Commission for Maritime Services. 36

RTK Real-Time Kinematics. 11, 36, 70, 102

SDK Software Development Kit. 106

SLAM Simultaneous localization and mapping. 48, 49

SMC Sliding Mode Control. 15, 39

SoTA State of the Art. 3, 9, 20

TOF Time-of-Flight. 44, 52

UART Universal Asynchronous receiver-transmitter. 55, 96, 147

VRLA Valve Regulated Lead-Acid. 20

Chapter 1

Introduction

This chapter describes the background, scope, requirements and provides a reader's guide to the report.

1.1 Background

Climate change is the most existential issues of our time. The automotive industry has been heavily criticized and increasingly regulated for their part in this issue, but marine transportation is often overlooked despite the fact that traditional 7.5 m petrol boats consume roughly 15x more fuel than a family car. Candela, an innovative boating company based in Stockholm, is making strides to correct this issue by providing a zero-emission alternative. Not only are their boats greener, they are also more efficient, save owners money, and provide increased performance and comfort compared to their petrol alternatives.

Candela designs and manufactures 100% electric, hydrofoiling boats [92]. Their first and only current boat model is the C7, see Figure 1.1; a speed boat that utilizes an actively controlled hydrofoiling system to lift its carbon fiber hull above the waves when speeds reach 17 knots. This reduction in drag means the electric motors of the boat can operate significantly more efficiently, resulting in a range comparable to conventional petrol boats. To keep the vessel stable during "flight", an array of sensors such as IMU, GNSS, and ultrasonics gather data that is processed and utilized to control hydraulic actuators at a rate of 100 Hz.

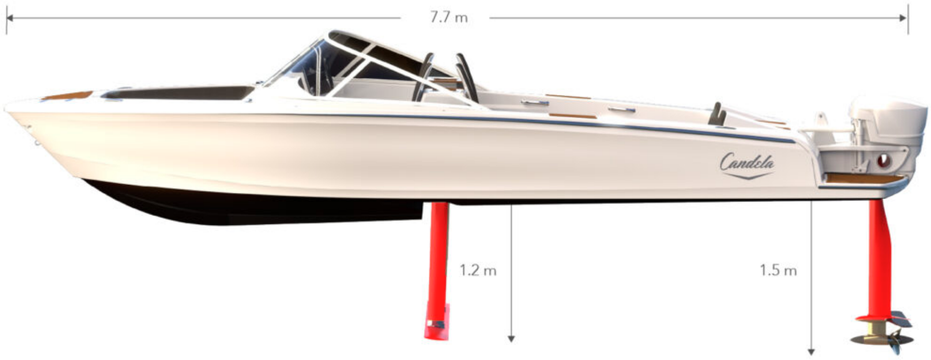


Figure 1.1. Candela C7 speed boat [92].

The success of the C7 has allowed Candela to grow and turn their sights to more ambitious endeavours... such as revolutionizing the public transportation sector. Their first venture into this new market is the P-30, see Figure 1.2. This new vessel will utilize the same electric and hydrofoiling technology of the C7, but scale it up with dual independently controlled motors that can transport 30 commuters. This ship is intended to replace the current fleet of diesel SL ferries in the Stockholm area and it is already in early development. With a speed of 30 knots and zero wake, it will be the world's fastest electric passenger ship and reduce commute times by 50%. Not only will it be fast, it's cutting-edge technology will also cut operational costs in half and provide passengers a smoother ride.

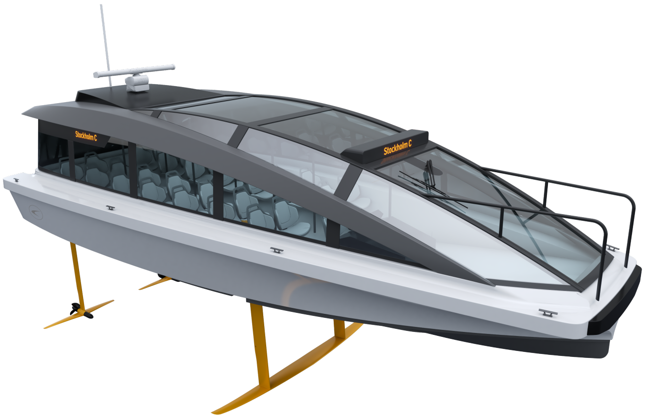


Figure 1.2. Candela P-30 concept rendering [92].

In parallel to developing the P-30, Candela is also exploring various means of autonomous navigation that can be integrated into their new vessels. This added functionality will increase safety in the long term, lower operational costs even further, and improve the consistency of the route scheduling when applied to commuter ships such as the P-30. This project is part of this endeavour by exploring viable solutions for autonomously docking the P-30 specifically. A scaled down prototype of the P-30 will be designed, fabricated, and then used for testing a methodology toward autonomous docking.

The main purpose of this report is to convey the research done toward various State of the Art (SoTA) solutions for autonomous docking, navigation and other associated vehicles that pertain to the project task. It also describes the necessary research done for designing the prototype vessel. In addition to said research, this report also discusses project requirements, the generated concepts, evaluation of said concepts and the final design that was ultimately chosen. The last portion of the report describes the future plans for the project moving forward.

1.2 Scope

To better illustrate how the project scope developed during the initial problem identification phases of the project, this section is broken down into two subsections; *Initial Scope* and *Refined Scope*. The *Initial Scope* refers to the desired scope initially pitched by stakeholders and the *Refined Scope* is the more simplified and focused version that ultimately guided project tasks.

1.2.1 Initial Scope

The initial scope presented by stakeholders was broad in that the overall desire was to automate everything the current SL ferry does while docking and in a similar environment. In the initial pitch presented by stakeholders, the desire for the following AutoDocking prototype “key features” were shown...

- ”Sensing depth and determining if docking is possible.”
- ”Automatically align with and approach the dock.”
- ”Avoiding collision with dock or other obstacles.”
- ”When in position, automatically deploy a gangway to allow passengers to leave or enter the ferry.”
- ”Holding the docking position until gangway is retracted, and operator gives the signal to leave the dock.”

This list encapsulated everything the real P-30 would be required to do, but it was also stated that ”The project can be taken on in part or as a whole depending on the outcome of the pre-study and consequent time plan”... The *Refined Scope* delves into how how these desired features were refined and made more feasible.

1.2.2 Refined Scope and Delimitations

After several meetings with the stakeholders and internal team meetings, the scope was better understood and areas of greater/less importance were identified. A more feasible and focused project scope was generated and is summarized by the points below.

- No gangway will be included in the project.
- The prototype's hull will be purchased to increase feasibility and does not need to mirror the P-30 (no hydrofoils and specific shape doesn't matter).
- There will be dual, independently controlled outboard motors placed at the stern of the boat for portability to the P-30.
- The dock will mirror those used by current SL ferries and require minimal modifications.
- When viable, a communication protocol similar to the P-30 will be used.
- Some basic obstacle detection and avoidance will be implemented, but advanced avoidance protocols and detection (such as those needing the training of large data sets) will not be required.
- The prototype will operate well in calm environmental conditions and attempts will be made to make it more robust if more time is available later on.
- The prototype will be designed to operate well during normal visibility and daylight conditions and efforts will be made to improve on this only if time allows.

In general, the stakeholders did not want to impose many hard requirements and instead wanted autonomous docking methodologies freely explored. Many of the items listed were therefore conceived by the team to increase feasibility and establish requirements.

1.3 Requirements

To better solve the overall task, the scope was broken down into a list of requirements that the project had to satisfy in order to be considered successful. Input was gathered from the stakeholders (*Stakeholder Requirements*) regarding desired system function and then further detailed and quantified (*Technical Requirements*) so test cases could be built upon them. Polarion , a Product Lifecycle Management (PLM) software develop by Seimens, was used to create requirements and link specific stakeholder/technical requirements to one another [48].

1.3.1 Stakeholder Requirements

The following system requirements were those identified by the stakeholder. Candela was hesitant to place many hard requirements on the project (as not to stifle innovation), so many were also identified by the team to help guide the project. Those that are labeled *extra scope* are requirements that were initially requested by Candela, but not required to consider the project a success. These requirements will be fulfilled if the group has time and resources remaining when the fixed requirements are fulfilled.

The following list describes the stakeholder requirements. They are categorized into five types... *Project Admininstration, System Portability, Autonomous Protocol, Obstacle Detection and Avoidance,* and *System Robustness.*

- Project Administration
  - **Purchasing:** Purchase requests shall be reviewed by the stakeholder prior to placement.
- System Portability
  - **Dual Outboard Motors:** The vessel shall have dual, outboard electric motors that are controlled independently.
  - **Dock Design:** The layout of the dock shall mimic those used by the SL ferry system in Stockholm.
  - **Communication Protocol:** The prototype's embedded system architecture shall utilize the same communication protocol as the P-30 concept.
- Autonomous Protocol
  - **Autonomous Navigation:** The prototype vessel shall reliably align and approach the dock fully autonomously without the need of operator intervention.
  - **Docking Time:** The time it takes to successfully dock shall be comparable to that of the current SL ferry system.
  - **Dock Departure Protocol:** The vessel shall hold its position at the dock until signaled to leave by the operator.
  - **Abort Signal:** At any point in time, a driver shall be able to stop all operations by means of some signal.
- Obstacle Detection and Avoidance
  - **Environmental Obstacles:** The vessel should be able to detect various obstacles and avoid them accordingly.

  - **Depth Sensing (Extra Scope):** The vessel should be able to detect the water depth and quickly abort the navigational path if depth is too shallow.

- System Robustness
  - **Lighting Conditions:** All vessel sensors shall be robust against various lighting conditions similar to those experienced during normal SL ferry operation.
  - **Drivetrain Environmental Robustness (Extra Scope):** Motor control system should be robust against localization errors induced by various environmental disturbances (i.e. wind, wakes, current, etc.).
  - **Sensor Environmental Robustness (Extra Scope)** The on-board sensors should be robust against various environmental conditions that the SL ferries regularly encounter.

1.3.2 Technical requirements

To better quantify and detail specifics for the respective stakeholder requirements, a list of technical requirements were created. These are reflected in the list below and categorized in the same methodology as the stakeholder requirements...

- Project Administration
  - **Budgetary Approval:** Project budget shall be 50,000 kr (not including tax), but variable at the discretion of stakeholders depending on how purchases are deemed as value added.
- System Portability
  - **Dock Form Factor:** The docks used for testing shall be a rigid flat wall, without anything guiding the vessel, and detectable at varying heights.
  - **Dock Material:** The docks contact surface shall be covered in a layer or rubber or other high friction material similar to those used in the various SL ferry dock locations.
  - **Dock Modification:** No excessive modifications to pre-existing SL ferry docks should be required to successfully detect it or hold the position of the vessel.
  - **Motor Rotation Range:** The dual motors shall each have a minimum range of movement of ± 30 °.
  - **CAN Bus Protocol:** A CAN communication protocol shall be used in the case of multiple processors and as long its viable for the particular sensor/purpose in question.
- Autonomous Protocol

  - **Navigational Control in Nominal Conditions:** The navigational control system shall operate with minimal error during nominal environmental conditions (no waves/wakes, no current, and low wind speed).
  - **Dock Impact Speed:** The prototype vessel shall not impact the dock at more than 0.03 m/s velocity (or < 0.6 m/s^2 deceleration).
  - **System Range:** The vessel shall be able to identify the dock and make necessary path adjustments from a radial distance of 50 m.
  - **Dock Accuracy:** The boat shall make contact to the dock within a ± 0.5 m accuracy of true position.
  - **Docked Angular Movement:** The boat shall not wander further than ± 5° from the predefined angle.
  - **Dock Slipping:** There shall be no slipping from the contact point between the dock and vessel.
  - **Docking Time Allowed:** The vessel shall not take longer than 1 minute to successfully dock from its initial starting point.
  - **Departure Heading Correction:** While leaving the dock, the boat shall reverse from the dock at the same predefined angle as docking for 10 m before correcting its heading to depart further.
  - **Manual Abort Timing:** When an abort signal is sent, the vessel shall begin abort sequence no longer than .1 s from when signal is sent.
  - **Navigational Control in Poor Environmental Conditions (Extra Scope):** The control system of the vessel should be robust against errors induced by extreme environmental conditions (large waves/wakes, current, high wind speed, etc.).

- Obstacle Detection and Avoidance

  - **General Abort Detection:** If a large object is detected in the way of the docking path, then the boat shall stop and wait until re-initialized by the operator.
  - **Depth Induced Navigational Abort (Extra Scope):** The boat should abort its autonomous docking sequence if detected water depth is less than 0.7 m.
  - **Avoid Other Boats (Extra Scope):** The vessel should be able to detect other boats in the water and avoid them accordingly.
  - **Detect Swimmers (Extra Scope):** The vessel should be able to detect people swimming in the water and avoid accordingly.

- System Robustness

  - **Lighting Condition Robustness:** The vessel's sensors shall be operational during various daytime lighting conditions.

- **Lighting Recognition:** The system shall be able to detect if the lighting condition is insufficient for proper operation and notify the operator.
- **Additional Lighting Conditions (Extra Scope)** On-board sensors should be robust against all lighting conditions, including nighttime and artificial lighting.
- **Sensor Environmental Robustness (Extra Scope)** The on-board sensors should be robust against weather conditions regularly seen by Stockholm's ferry system, such as heavy rain, fog, snow, etc.

1.4 Readers guide and report disposition

As seen in the table of contents, this report consists of 9 chapters (including this introduction chapter). Starting with Chapter 2, the reader will learn about similar solutions to autonomous maritime navigation as well as be introduced to various models, hardware, and communications protocols that could be of interest for this application of an autonomous docking prototype.

Continuing with Chapter 3, the reader is shown the methodology used in the project. This includes a representation of the team and project management as well as the engineering approach and budget for the project. Reading further, the concept generation and evaluation for all subsystems are presented in Chapter 4, resulting in a final concept for the docking prototype.

The system architecture and an overview of the final prototype follows in Chapter 5; software tools for integration are described here as well. This chapter also presents the implementation of each subsystem in fine detail.

Chapter 6 describes how each of the subsystems in the final concept were verified and validated through unit and integration testing, followed by system level testing for the docking prototype as a whole. The results for these tests are presented in Chapter 7 and are followed up by an analysis discussion on said results in Chapter 8. This chapter also brings up matters of ethics and sustainability. The chapter ends with a conclusion and are followed up by the last chapter (Chapter 9), that suggests further work to the finalized prototype. For the curious reader, a list of references and appendices are found at the very end of the report.

Chapter 2

State of the art

This chapter describes the SoTA, featuring earlier research and implementation of fully or partially autonomous marine vessels and other vehicles. Furthermore, it presents theory and hardware needed in order to model, control, position and drive a marine craft for the scope of this project. Lastly, the chapter presents existing hardware and strategies in order to detect objects, obstacles and hazards in different ranges.

2.1 Earlier similar work - Marine Vessels

2.1.1 Wärtsilä SmartDock

In 2018 the Finnish company Wärtsilä succefully tested a fully automated ferry docking system. The test was done on the Norweigan Ferry Folgefonn which successfully docked in the harbour of Stord. Today they are delivering this as a commercial solution [127] [126]. The SmartDock system as they call it, is a part of Wärtsiläs SmartMove Suite. It provides sensors for accurately determining the Wind Speed, Pitch, Roll, Heading, Position and Ship Model. With this information from the sensors the system can safely and autonomously dock the ferry from a range of 2000 meters [128]. The sensor used to create an accurate position of the vessel when docking is the CyScan. CyScan is a Dynamic Position laser sensor and is able to detect and locate prisms stationed at the dock, see Figure 2.1. With the information about the relative position to the prisms the vessel can accurately determine its location and from there on proceed to dock [74].

Figure 2.1. The prism located at the dock side [75].

2.1.2 Hammerhead Vision System

The United States Navy has along with the Jet Propulsion Laboratory developed an autonomous suite for unmanned surface vessels. Two key components of the system are the hammerhead system and the R4SA. The hammerhead system is a vision-based system for detection and tracking of static and dynamic objects from a high-speed vehicle. The R4SA is a framework for implementing navigation behaviour such as obstacle detection and target following. The hammerhead system consists of four cameras that are arranged in a left-facing pair and a right-facing pair to gain a very wide field of view [47]. The images captured by the stereo cameras are used to create a local hazard map and a list of contacts which is then transmitted to the R4SA system. The R4SA system then uses the given data for hazard avoidance and for the tracking behaviour [47]. An example of how the hazard map can look like can be seen in Figure 2.2.

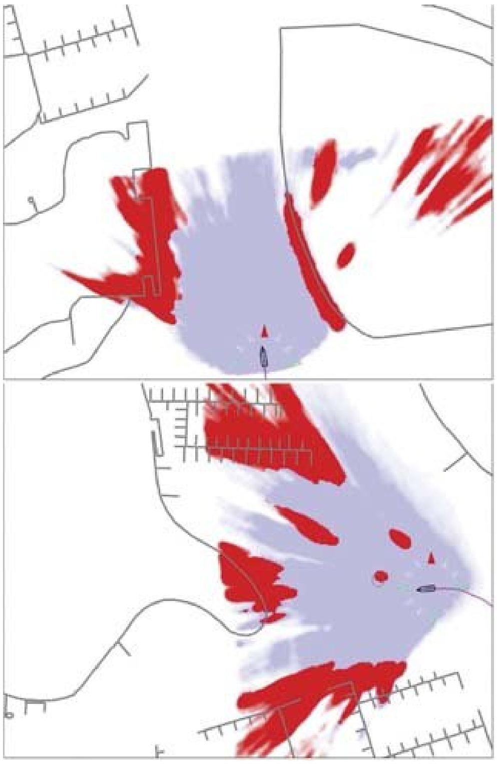


Figure 2.2. Red shows areas with high hazard probabilities and blue indicates where there is coverage by the hammerhead system [47].

2.1.3 Yanmar

Yanmar is multinational company which originates from Japan. They provide engines for a great variety of applications such as agriculture, construction and marine [135]. The company also went in to the autonomous boat business in 2017. Yanmar has since then had great success in their concept testing. Their proof of concept vessel uses a twin-propeller, twin-rudder ship combined with a thruster for increased maneuverability (Figure 2.3). When it comes to the autonomous part of the concept, Yanmars first iteration utilized Real-Time Kinematics (RTK). This gave the boat great positioning accuracy, but the vessel could only dock at ports fitted with RTK stations. To further develop this, the company moved over to a system where they combined LiDAR technology with Global Navigation Satellite System (GNSS). This new combination allowed the boat to autonomously dock at ports even if they where not fitted with an RTK station [136].

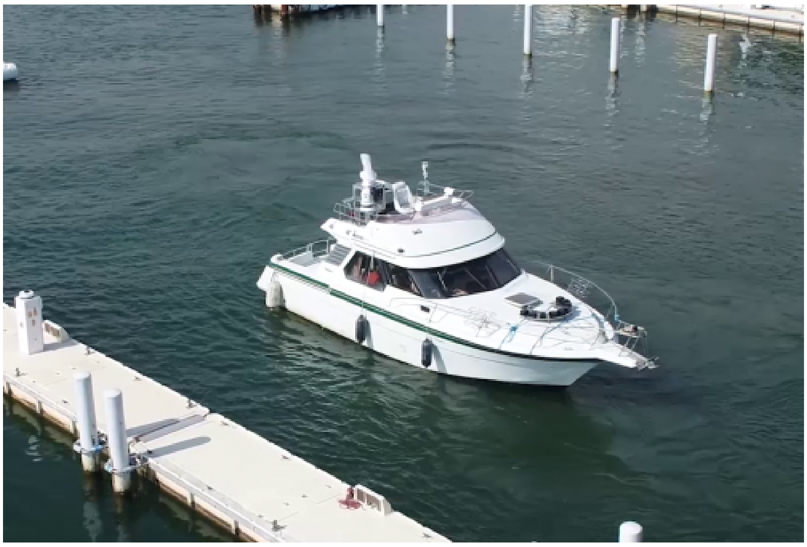


Figure 2.3. Yanmars concept vessel with a LiDAR unit in the front [136].

2.1.4 Raymarine Assisted Docking

Raymarine is a company that specializes in high performance marine electronics for the recreational boating market and light commercial marine industry. Their aim is to deliver the best intelligent navigation systems and sensors for safer marine operation. The Raymarine DockSense™ Control is an object recognition and motion sensing assisted docking solution. The solution utilizes intelligent FLIR machine vision camera technology (3D Stereovision) to analyze real-world imagery to create a 3D-map and effectively generate a virtual bumper around the boat, as seen in Figure 2.4. The vision system is integrated with the vessel's propulsion and steering system to help the driver with docking maneuvering, making it partially autonomous. It also uses Global Positioning System (GPS) and an Attitude Heading Reference System (AHRS) to compensate for wind and currents. [95].

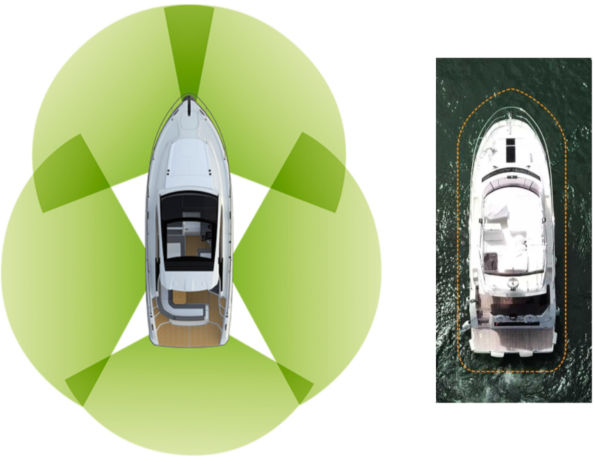


Figure 2.4. Raymarine vision system that creates a virtual bumper around the boat [94].

2.1.5 Autonomous Docking Research

In 2007, Martins et al. [77] presented a paper where they performed the docking of an Autonomous Surface Vehicle (ASV) with an Autonomous Underwater Vehicle (AUV) based in visual information. A hybrid systems approach is used in the docking manoeuvre synthesis, as displayed in Figure 2.5, and the control law is represented by a hybrid automaton with both discrete and continuous states. To determine the relative position and attitude between the two vehicles, the visual information was fused with Inertial Measurement Unit (IMU) measurements in an Extended Kalman Filter (EKF).

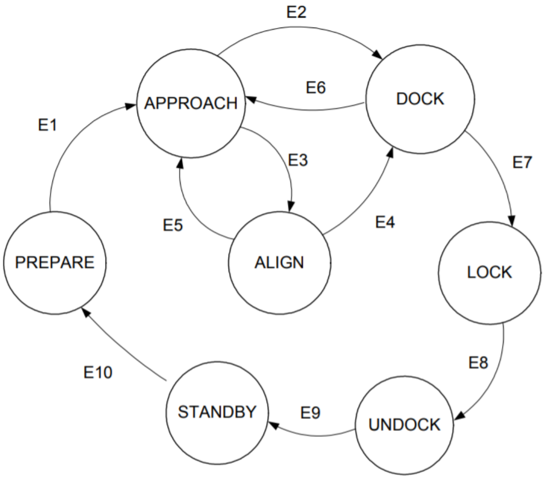


Figure 2.5. Discrete states of the hybrid manoeuvre used by Martins et al. [77].

A hybrid systems docking approach was also used by Braga in 2010 where he developed a control strategy for an AUV to dock into a mounted underwater docking station [12] and by Rosa in 2017 where he developed autonomous docking between an AUV and ASV based on GPS and IMU data [103].

In the work from 2019, Leite developed a self-guided docking architecture for ASVs [63]. The architecture incorporates data readings from GPS, LiDAR, camera and IMU and is presented in Figure 2.6. The docking operation first begin when correct position and orientation requirements are met. To fulfill these conditions, the Catch-Zone Approach and Reorientation Module are used. When the requirements are met, the Docking Approach may be initiated, which is the module conducting the approach towards the dock. The Supervisory Control System in the architecture is used to supervise and monitor the docking manoeuvre.

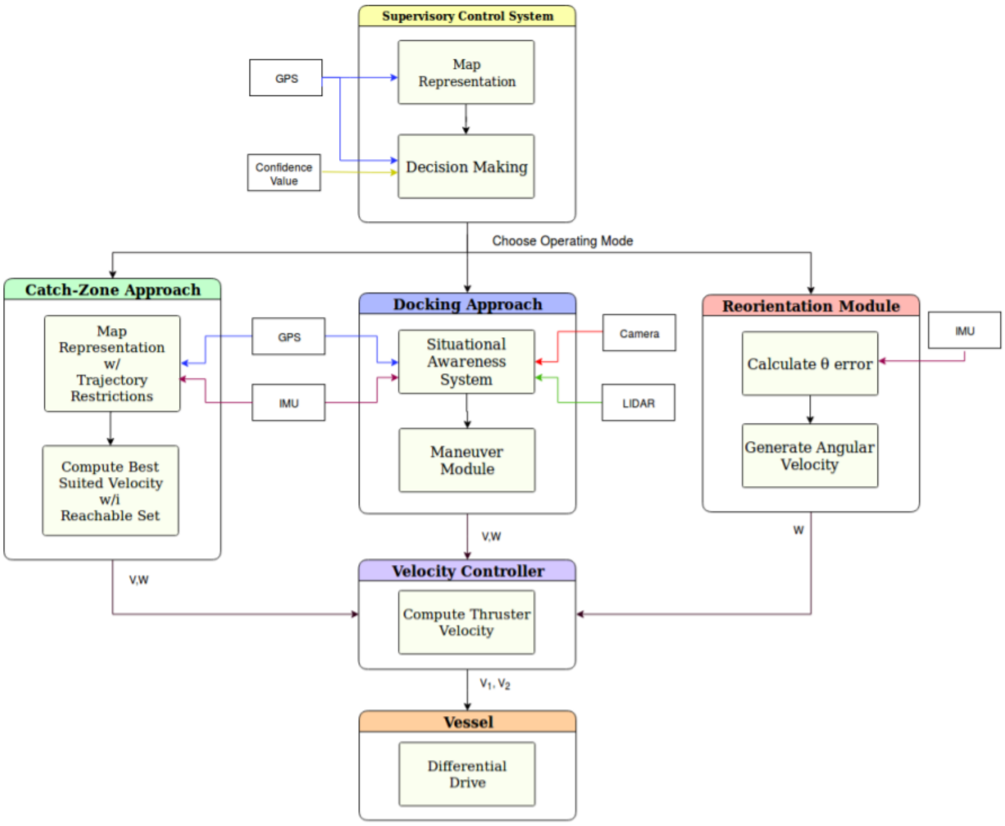


Figure 2.6. Overview of the self-guided docking architecture developed by Leite [63]

On the topic of control and guiding the vessel towards the dock, plenty of papers exist. In 2019, Wang et al. presented a comprehensive article on the matter [125]. As stated, common algorithms for motion control are Proportional–Integral–Derivative (PID) control, optimal control with e.g. LQR, Sliding Mode Control (SMC) and Model Predictive Control (MPC). The relationship between all algorithms discussed in the article is displayed in Figure 2.7. Also stated is that the research on the autonomous docking of ships is not very extensively presented in the article, though the main control algorithms found to be used for this purpose are mainly PID and/or neural network based.

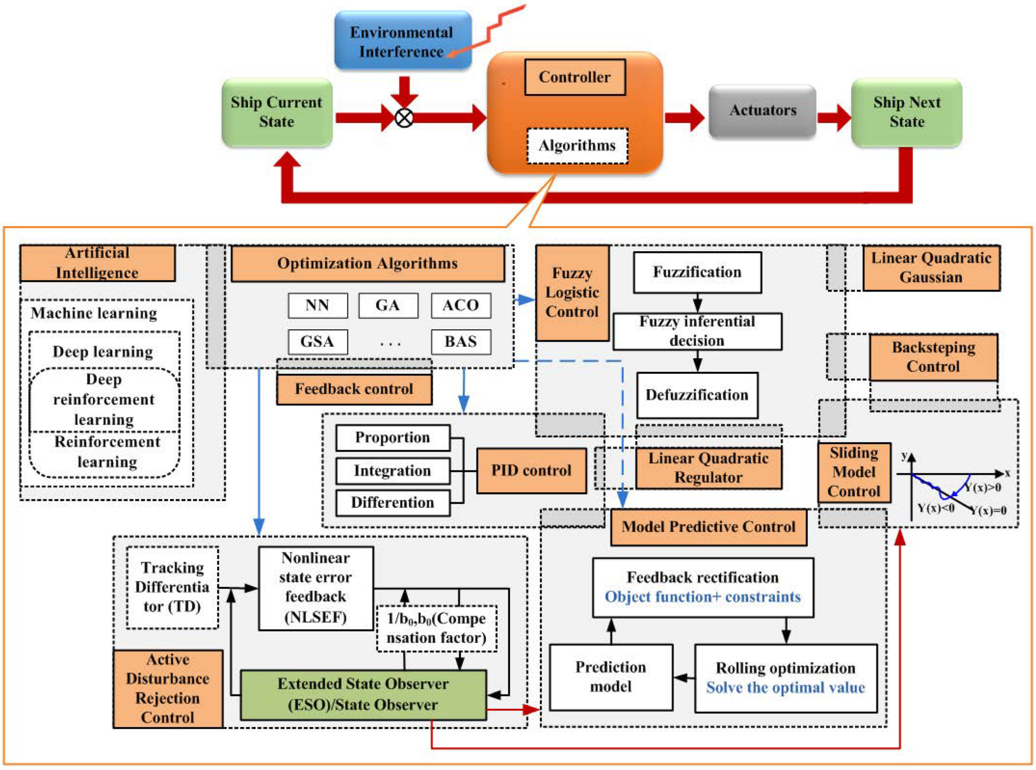


Figure 2.7. Relationship of motion control algorithms as stated by Wang et al. [125].

2.1.6 AAWA Initiative

The Advanced Autonomous Waterborne Applications (AAWA) Initiative is a large and well-funded project co-operation between institutes and maritime companies. The aim is to produce specifications and designs for the next generation of advanced ship solutions [100]. They emphasize the importance of sensor fusion to provide reliable and sufficient performance for situational awareness in all conditions for autonomous vehicles. The fused sensor data should seamlessly be integrated with path planning and reactive collision avoidance systems and their proposed architecture for this module integration can be seen in Figure 2.8. For situational awareness, their research suggests that short range radar or LiDAR can provide accurate range, velocity and angular measurements of objects, while cameras provide better spatial resolution to classify objects. Near-IR cameras or thermal cameras could also be used for night-time imaging but may be prone to failure under difficult weather conditions, where radar would be a better performer [100].

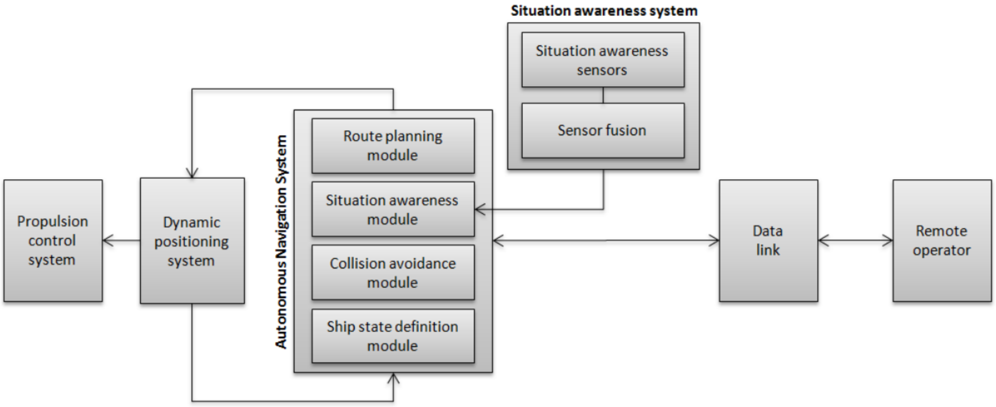

Situation awareness system
Situation awareness sensors
Sensor fusion
Autonomous Navigation System
Route planning module
Situation awareness module
Collision avoidance module
Ship state definition module
Propulsion control system
Dynamic positioning system
Data link
Remote operator

Figure 2.8. ANS architecture [100].

2.2 Earlier similar work - Other Autonomous Vehicles

Autonomous driving is not only a hot topic among boats. The car industry has been aiming for self driving cars for a long time, and for example, self parking is a common feature in new cars. Even though cars and boats are very different, there are lots of similarities in detecting the surrounding required for autonomous driving.

2.2.1 Tesla

Tesla is a new thinking and innovative car company. It might also be the car company with most progress into the field of autonomous driving. They all ready have a number of driving assisting and safety features such as autopark, autosteer and emergency breaking. All Tesla's cars are sold with the hardware for full autonomous driving. This makes it possible to offer full autonomous driving in the future to customers buying a car today, just by updating software [7]. The hardware implemented by Tesla to make autonomous driving possible consists of a number of cameras, distance sensors and radar as seen in Figure 2.9. There are eight cameras providing 360 degree vision around the car as well as twelve ultrasonic sensors around the car to detect cars or other objects while driving and parking. At last there is a robust radar detecting forward objects. By this setup, Tesla expects their cars to be able to fully autonomously and safely drive the fastest route between two different places.

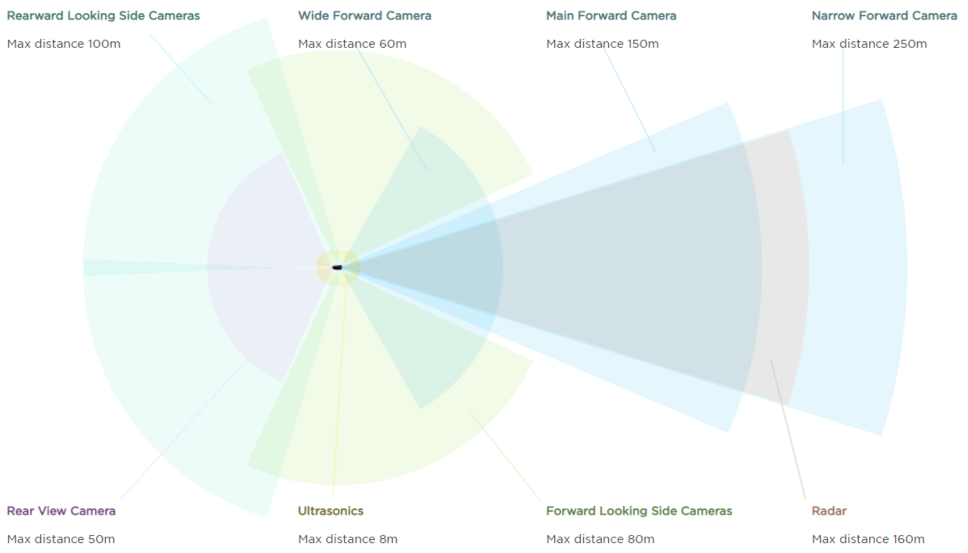


Figure 2.9. Tesla's setup of sensors in order to assist the diver, and possibly provide self driving in the future [7].

2.3 Drivetrain

The drivetrain subsystem of the project includes the various hardware components needed to make a scaled model of the P30 and propel it. Boat hull, outboard motors, actuation and power supply all fall into this category.

2.3.1 Outboard Motors

There are several ways to drive a boat forward; outboard/inboard motors, bow mounted, thrusters etc. In order to make the project as portable to Candela's P30 as possible, only stern mounted, electric outboard motors have been researched.

Further limitations by the portability to the P30 include:

- Dual motors.
- Electric to control thrust via voltage signal.
- Individually controlled to allow vector steering (±30°).

In an electric motor the throttle is controlled only with a voltage signal from an on board computer. Since the P30 is also electric, having as similar motors as possible ensures better portability to the real system. There are different options for electric motors depending on the scale of the system relative to the P30. Options are RC motors for model boats, small electric trolling motors or full size outboard motors. Important to note is that as the motors and scale of the project gets bigger, the price of components increase exponentially. However the project scale is ultimately a decision concerning a lot of different factors, not only the motors.

RC motors

There are plenty of powerful RC motors on the market. An RC boat can be driven at very fast speeds compared to its size, since the drag of the water becomes close to zero and because the motor doesn't need a lot of torque, only a very high RPM. An example is the Volantex V792 RC speed boat, which has a potential top speed of 33 knots using an 1800kv (Constant Velocity, No-load speed at 1 volt (KV)) brushless DC motor. However there are still problems with with a small RC boat design that makes it hard to follow the limitations of the project. Generally, a DC motor is usually fit in the middle of the boat, with a rigid axis going through the whole boat to the propeller. There are examples of RC outboard motors that fit the scope of the project, however these are hard to find, very expensive and not nearly as powerful as their rigid shaft counterparts. Both the regular onboard motors and outboard motors are shown in Figure 2.10. It is difficult to fit two of these motors along with a drive-by-wire solution that allows both motors to be individually rotated on such a small scale.

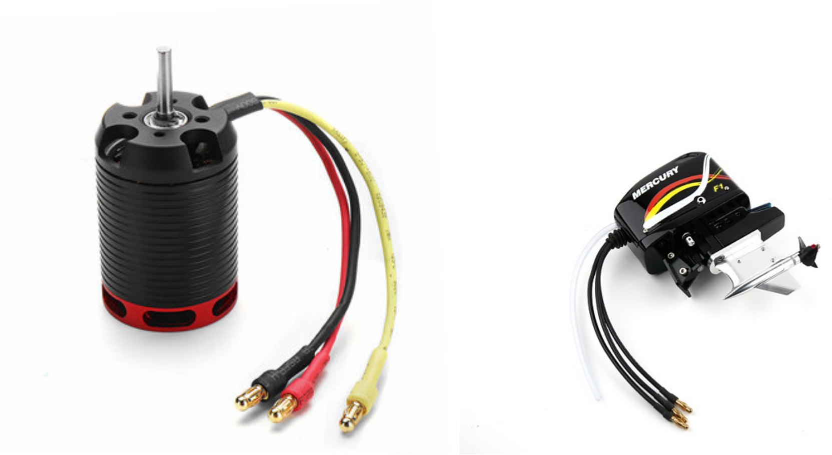


Figure 2.10. Onboard and Outboard RC motors [89].

Electric Trolling Motors

There are also many viable motors on the market for scale models larger than RC boats. One example is small electric trolling motors, shown in Figure 2.11. These are mainly used for fishing, attached to smaller boats such as jon boats, kayaks and row boats. They come in a variety of different models and amounts of thrust, ranging from 133N up to 490N. These motors are fit to the back of a smaller vessel, and are controlled by an operator rotating them in any direction. Because they are big enough to fit the driving DC motor in the water with the propeller, the operator can rotate the motor 360 degrees without the need of a rudder. The thrust is controlled by rotating the handle connected to a potentiometer adjusting the voltage to the motor. The maximum speed of these motors is not very high, a single trolling motor with a forward thrust of 245N is capable of propelling a three meter long jon boat at roughly five kph [54]. 245N is the max thrust available for one 12V motor, higher thrust systems require 24V, which in turn doubles the amount of batteries needed.

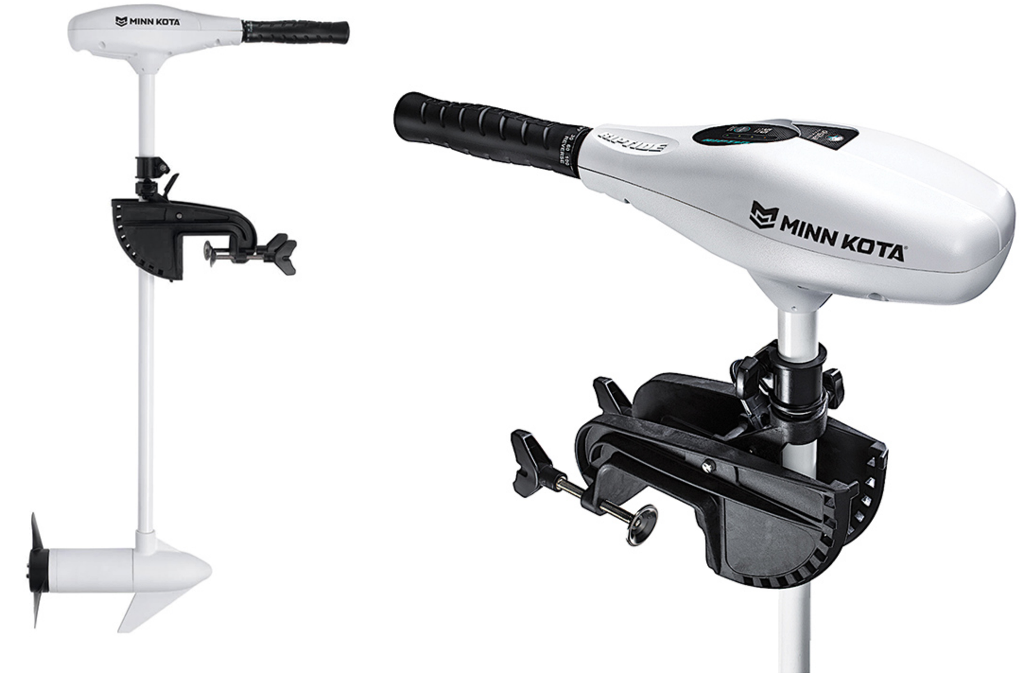


Figure 2.11. Example of a small electric trolling motor [83].

Custom built motors

Due to the simplicity of an electric boat motor, one could realistically be built to be tailored to the scope of the project. An electric boat motor is essentially a brushed DC motor with a propeller attached. In order to control the speed, a potentiometer is used to modify the voltage going to the motor, and to control direction, the whole motor is rotated around its y-axis.

The advantages of designing and building a custom boat motor are mainly cost and customization. By buying parts it is possible to meet all requirements set by the project scope, from thrust to choice of control system to size and form factor of the motor 2.12. Cost is reduced due to reduced price of individual parts, cost of labour is removed, as well as a lot of parts that come with a commercial motor is not necessary. However there are also problems with building a custom motor. Even though the circuitry is not a problem to design, since it is under water the enclosure needs to be very high tolerance and precisely made to ensure no leakage. To make a motor at such high standards is time consuming since ensuring good performance needs extensive testing.

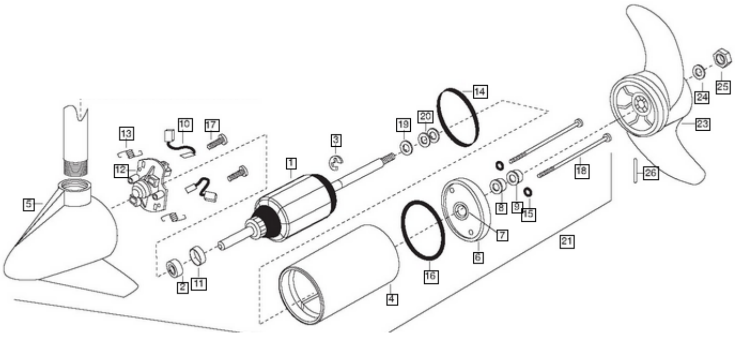


Figure 2.12. Parts diagram of an electric trolling motor [82].

2.3.2 Accumulator

The ”accumulator” refers to the main power source of the vessel, or more specifically, the system of rechargeable batteries with the absence of a generator. Since the boat will be far from shore while testing, it must have its own power supply. There is a plethora of battery types and the technology is constantly changing. For that reason, this particular SoTA discussion will focus specifically on batteries that are commonly used in pertinent applications, such as marine and electronic hobbyist applications. Such battery types include Absorbent Glass Mat (AGM), LiFePO4, and custom lithium cell battery packs.

AGM Deep Cycle Batteries

AGM batteries have a similar construction to the average lead-acid battery commonly found in everyday cars. With these batteries, a series of lead-alloy plates act as electrodes and are suspended in an electrolyte (sulfuric acid most commonly). While discharging, electrical current is produced because negatively charged ions in the sulfate flow from the positive (cathode) terminal of the battery to the negative (anode) terminal [3].

AGM batteries work in this same manner, but rather than being flooded with liquid acid, a fine mesh of glass fiber absorbs the fluid and the excess is removed (see Figure 2.13). They are also classified as a Valve Regulated Lead-Acid (VRLA) batteries because they are fully sealed [124]. This makes them much safer since acid can't leak out and are less prone to vibration induced damage. AGM batteries are commonly found in aerospace, racing, and boating applications for that reason.

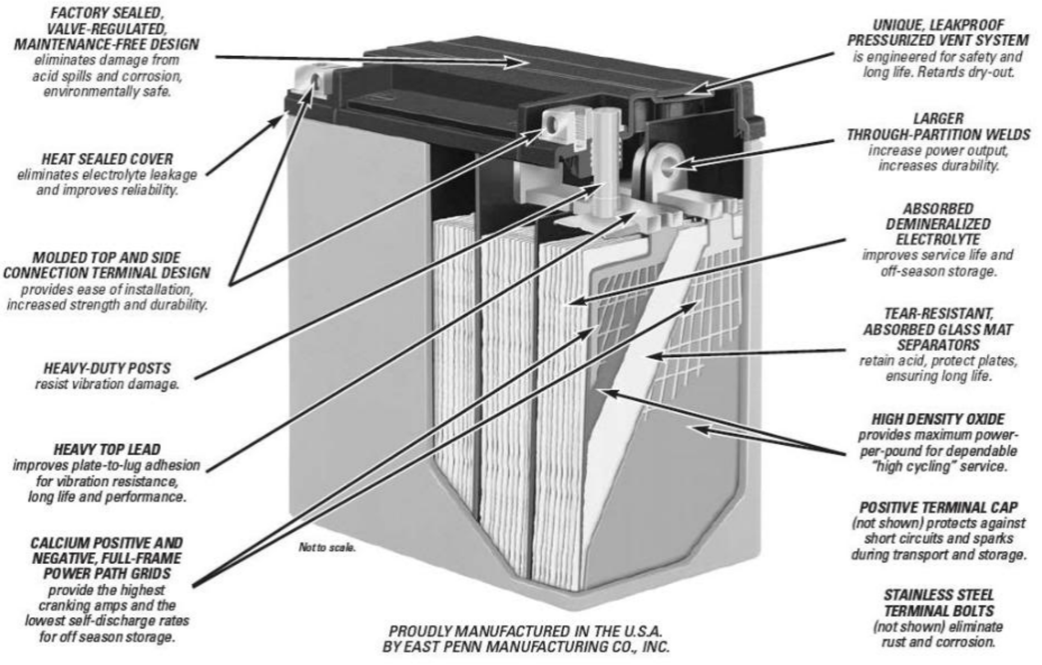

FACTORY SEALED, VALVE-REGULATED, MAINTENANCE-FREE DESIGN eliminates damage from acid spills and corrosion, environmentally safe.
HEAT SEALED COVER eliminates electrolyte leakage and improves reliability.
MOLDED TOP AND SIDE CONNECTION TERMINAL DESIGN provides ease of installation, increased strength and durability.
HEAVY-DUTY POSTS resist vibration damage.
HEAVY TOP LEAD improves plate-to-lug adhesion for vibration resistance, long life and performance.
CALCIUM POSITIVE AND NEGATIVE, FULL-FRAME POWER PATH GRIDS provide the highest cranking amps and the lowest self-discharge rates for off season storage.
Not to scale.
PROUDLY MANUFACTURED IN THE U.S.A. BY EAST PENN MANUFACTURING CO., INC.
UNIQUE, LEAKPROOF PRESSURIZED VENT SYSTEM is engineered for safety and long life. Retards dry-out.
LARGER THROUGH-PARTITION WELDS increase power output, increases durability.
ABSORBED DEMINERALIZED ELECTROLYTE improves service life and off-season storage.
TEAR-RESISTANT, ABSORBED GLASS MAT SEPARATORS retain acid, protect plates, ensuring long life.
HIGH DENSITY OXIDE provides maximum power-per-pound for dependable "high cycling" service.
POSITIVE TERMINAL CAP (not shown) protects against short circuits and sparks during transport and storage.
STAINLESS STEEL TERMINAL BOLTS (not shown) eliminate rust and corrosion.

Figure 2.13. AGM battery construction [72].

One major negative of sharing the same construction as normal lead-acid batteries is their weight. Unlike modern lithium batteries, AGM batteries still consist of heavy lead plates and fluid (even if they are contained in fiber mats) which makes them very dense. This means that AGM battries have a very low energy density (Wh/kg, capacity per kg) and most weigh between 30-45 kg. This same construction also makes them more sensitive to cold weather in comparison to more modern battery types.

Like other lead-acid batteries, their low internal resistance does mean that they have a high power density though; making them perfect for applications that require large amounts of current to be drawn instantaneously. This is another reason why they are commonly used in cars...the electric starter motor quickly draws large amounts of current and these are up for that task. This also makes them a perfect candidate for powering larger DC motors, such as trolling motors.

More importantly, they are also considered a *deep cycle* battery. This means that unlike conventional lead-acid batteries, AGM batteries have a relatively long life cycle thanks to their glass matting and sealed structure. An AGM battery can withstand roughly 500-1300 discharge cycles depending on the depth of discharge (how much of the capacity is depleted). The two parameters are closely correlated and is represented in Figure 2.14. With that being said, these batteries do not have a constant discharge rate. Once an AGM battery is discharged to roughly 50% capacity, voltage drops drastically. A good charger is also important in the case of cycled use since they are sensitive to overcharging. This can damage the batteries and lower capacity permanently just as much as excessive discharging.

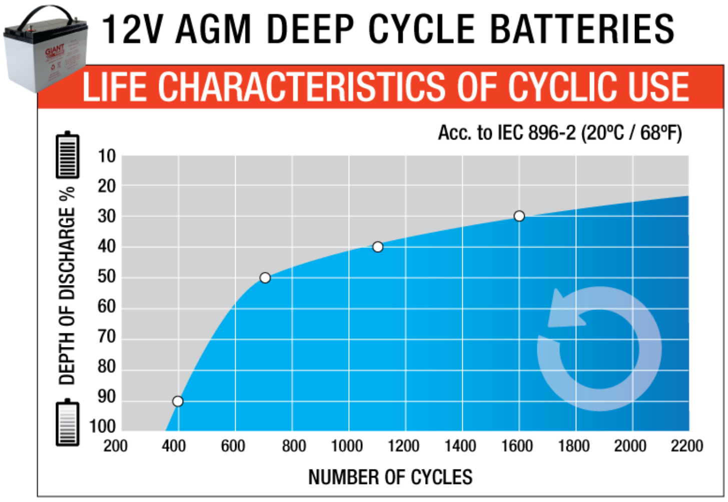


Figure 2.14. AGM battery lifecycle as a function of discharge depth [3].

Since AGM batteries are commonly used in the automotive industry, they are easily found in 12V form factors and are readily available up to 100 Ah capacities. In addition, this technology is well established and inexpensive to manufacture, so they are much cheaper compared to more modern lithium battery architectures (1200 - 2400 kr).

LiFePo4 Battery

In recent years, lithium battery types have been increasingly used in a multitude of different applications. Specifically, LiFePO4 batteries are commonly sold in 12V configurations and are used as an upgrade in applications where AGM batteries are common (aerospace, automotive, marine, etc.). For that reason, there physical form factor usually mirrors conventional lead-acid batteries.

Lithium-ion batteries work in the same way as conventional batteries in that current is produced by the flow of ions between cathode and anode, but as the name implies, ions are sourced from lithium rather than a lead alloy [43]. In the specific case of LiFePO4 batteries, iron phosphate is also combined with lithium on the cathode and the anode consists of graphite with some metallic backing (Figure 2.15)[66]. Lithium is a very light metal and has a high electrochemical potential, all of which means that these batteries have a substantial energy density and weigh roughly 50% less than AGM batteries of similar capacity [68]. This composition also makes it much less sensitive to temperature extremes; particularly cold weather where AGM batteries do not operate well.

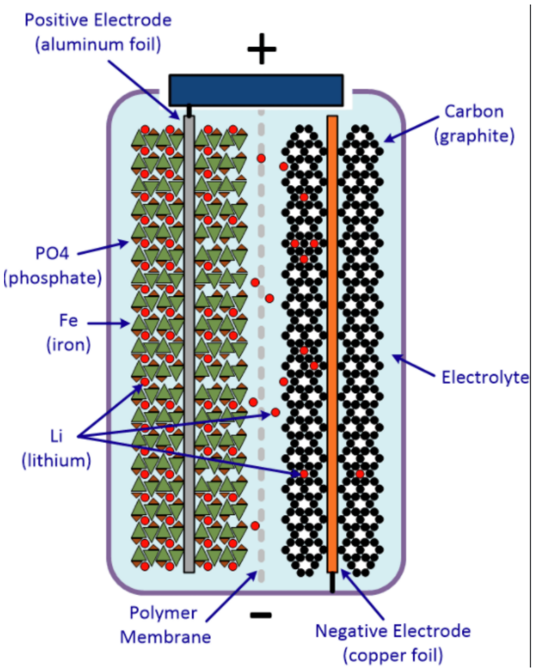


Figure 2.15. Illustration of the chemical reaction in LiFePO4 batteries [67].

The real reason why LiFePO4 batteries are able to take the place of AGM/lead-acid batteries though is because of its power density. Like its competition, low internal resistance equates to the ability to support large current demands, so these work well in the same applications [68]. The real benefit of LiFePO4 over AGM batteries though is its lifespan. These batteries are also considered batteries, but these are specifically used for constant discharge applications and have a life span of 2,000 - 10,000 cycles (depending on depth of discharge again) [66]. Not only that, but the voltage discharge curve is flat and can supply constant voltage until its total Ah capacity is depleted (see Figure 2.16 for comparison). The *c-rate* (speed a battery can be charged relative to discharged) of a LiFePO4 battery is significantly better than an AGM battery as well (10x).

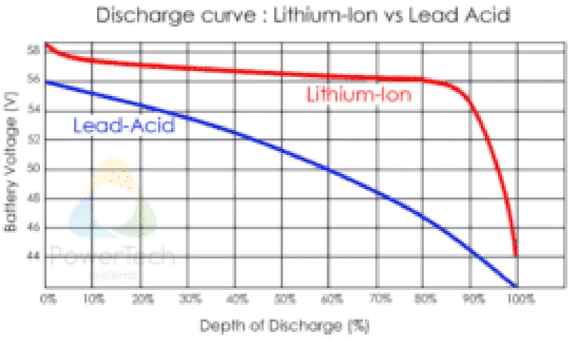


Figure 2.16. Comparison of discharge curves between LiFePO4 and AGM batteries [69].

The most major drawback of LiFePO4 batteries in comparison to AGM is the cost. Lithium batteries are composed of expensive materials and are a relatively new technology, so prices are roughly 4x AGM batteries. With that being said, LiFePO4 batteries are generally cheaper than some of the better performing lithium battery types. One of the other major justifications for this cost is the Battery Management System (BMS). Lithium batteries can be dangerous and catch fire when overcharged, so most lithium batteries come equipped with some form of BMS; a small PCB board that regulates charging of each cell.

Custom Lithium Cell Battery Pack

With the growing availability of lithium-ion batteries, hobbyists have harnessed the opportunity to create their own custom battery packs for numerous applications ranging from phone chargers to entire electric cars. Although many utilize whatever is readily available, the most commonly sourced battery type is the 18650 cell. This number refers to the cells dimension (18mm diameter, 65mm long, and 0=cylinder) [1]. The main appeal for these is their price and availability. These cells have been utilized in everything from rechargeable flashlights to Teslas for years and entire companies have been built around recycling and selling them at a discount.

Nominal voltage for these type of cells are 3.7V, but they are available in numerous parameters/compositions depending on the manufacturer. The accumulator options are endless when combining these cells in series/parallel. Figure 2.17 depicts a common example of an 18650 battery pack (3s5p...3 cells in series and 5 in parallel) with the inclusion of a BMS.

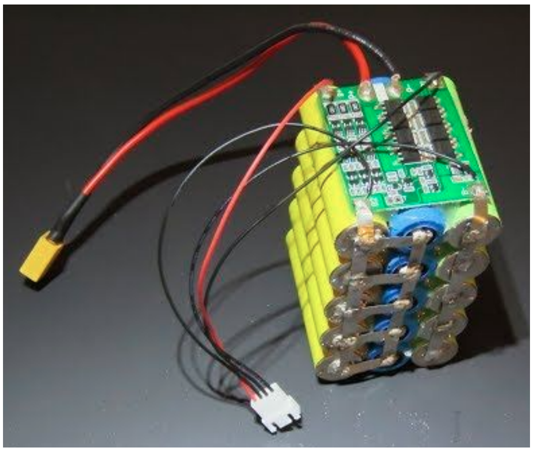


Figure 2.17. Example of a custom lithium-ion battery pack made of 18650 cells [46].

The biggest advantages of 18650 cells for an accumulator is the versatility and cost. By combining smaller cells that are hand-picked, the power source can be built to meet any voltage, capacity, specific power, or any other specific requirement. Also, by purchasing recycled cells or buying in bulk, accumulators can be built cheaper than using purpose built LiFePO4 batteries and outperform.

From the same source of the advantages stems some disadvantages. This modular approach also demands some specialized equipment to build and a substantial amount of time to create. A spot welder is needed to safely attach the conductive strips that connect the individual cells and an appropriate BMS also needs to be sourced. The work of attaching these cells together can be tedious in larger projects and the safety/reliability of the accumulator is ultimately decided by the diligence of the technician. In general, these types of cells are also more volatile and prone to combustion than LiFePO4 batteries [56].

2.3.3 Drive-by-wire Systems

Due to the autonomous nature of this project, a drive-by-wire system will be a must. Such a system replaces traditional mechanical and/or human-machine interfaces with a system of electromechanical actuators for steering. The following section explores some current marine steering systems and other pertinent methodologies of electromechanical actuation.

Boat Steering Hydraulics

Hydraulic drive-by-wire systems utilizes hydraulic actuators to control the steering. Modern versions of this include electro-hydraulic power steering(EHPS) where the hydraulic pump is controlled by an electric motor. This have been used for a long time in the automotive industry to improve the stability and ease of control over the steering. For the same reason as in the automotive industry, hydraulic cylinders have been used to control the steering of the boat engine and alleviates much of the physical effort that is associated with traditional cable driven boat steering. Typically this is a common solution for steering boats equipped with high torque engines or in the case where multiple engines are mounted in the stern. This is also the type of steering mechanism that is currently used by Candela.

Although hydraulics offer many benefits over cable driven systems much of the preferred way of implementing a steer-by-wire system is to remove the hydraulics and instead control the steering directly with a purely electrical system. The benefit of that is that an electrical system is using fewer moving parts and removes the hydraulic pumps and fluids which makes a simpler system to control[37].

Electrical Power Steering

Electrical Power Steering (EPS) are a common alternative to hydraulically controlled steering and utilizes a system of motors, gears and angle sensors to control the steering. This is a common concept applied to modern cars as it allows a central computer to control the steering which makes it possible to include systems like the auto-pilot and traffic accident avoidance systems.

One interesting showcase of the EPS system are the ABB azipod thruster. The azipod is a version of a broader variatey of motors called azimuth thrusters that are able to turn 360 degrees around its vertical axis. A common setup is to equip vessels with a pair of azimuth thrusters in which the 360 degree control gives the the ship great manuverability and are a common soloution for creating dynamic positioning systems. The azipod is a gearless electric propulsion system that can be fully integrated into a submerged pod [91]. The thruster is divided into a propulsion and a steering module. The steering module is using designated steering motors that are responsible for controlling the angle of the propulsion motor. This is done through connecting the steering motor to the propulsion module through a set of pinions, a gear rim, a slew bearing and a slipping unit, which allow the 360 degree rotation.

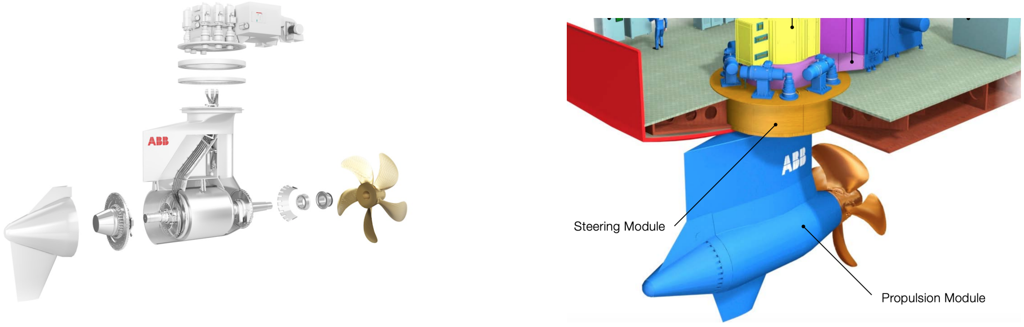


Figure 2.18. Assembly of the ABB azipod (L) and the steering and propulsion module (R) [91].

The azimuth thrusters are usually designed in the power ranges of 1-22MW systems which is targeted to mainly equip large surface vessels. The steering module used in the azipod can however be scaled down to match a smaller propulsion system with the use of a servo or stepper motor. This is common in RC boat application. While stepper and servo motor are both similar in that they can be used in similar applications, there are however some key differences dividing the two.

A stepper motor is a Brushless DC Motor (BLDC) that rotates in a set amount of steps. A permanent magnet rotor is surrounded by a set of windings (stators), that when current is applied in a set order defined by the motor driver, makes the rotor rotate in incremental steps. This means that the stepper motor can run in ”open-loop”, since it is possible to keep track on the amount of steps moved without feedback. However, when stepper motors carry a load, it is possible that they slip and miss a step every once in a while. Usually this isn’t a problem due to one step being very small, but for applications where the position is crucial, it can be a good idea to run the stepper motor in ”closed-loop” instead by using an external encoder that provides angular feedback. Another key characteristic of the stepper motor is it’s high holding torque. Current runs through the stators, even at rest, which allows a stepper motor to keep its position very well. However, this torque falls off as speed increases [41].

A servo motor on the other hand cannot be run in open loop since it relies on a feedback device to read parameters, such as torque, motor velocity, voltage and current level. Due to this, they are very accurate at high speeds and will produce a fairly constant torque output. Due to the need of feedback control and the need of position feedback, servo systems are usually more expensive [44].

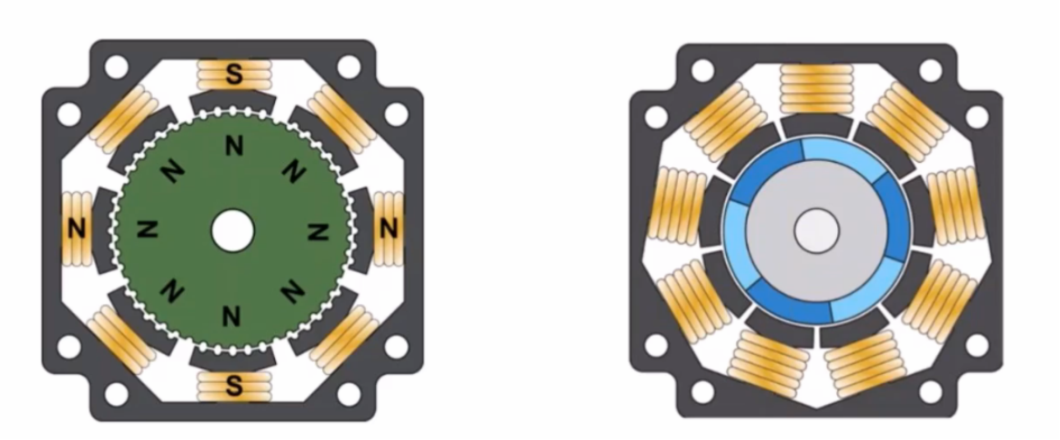


Figure 2.19. Stepper motor(L) and servo motor(R), steppers have up towards 100 poles where servos have a maximum of 12. [106]

Electronically Controlled Trolling Motors

Some higher-end trolling motors are equipped with a drive-by-wire system of their own rather than requiring the physical input of an operator. This is often achieved with a small DC motor that is mated to the shaft via a gear and operates at the same voltage as the larger motor. Minn Kota takes this same approach with their Terrova series of motors as seen in Figure 2.20 [117]. These types of motors are usually reserved for bow mounting useage (in the front of the boat) due to their size.

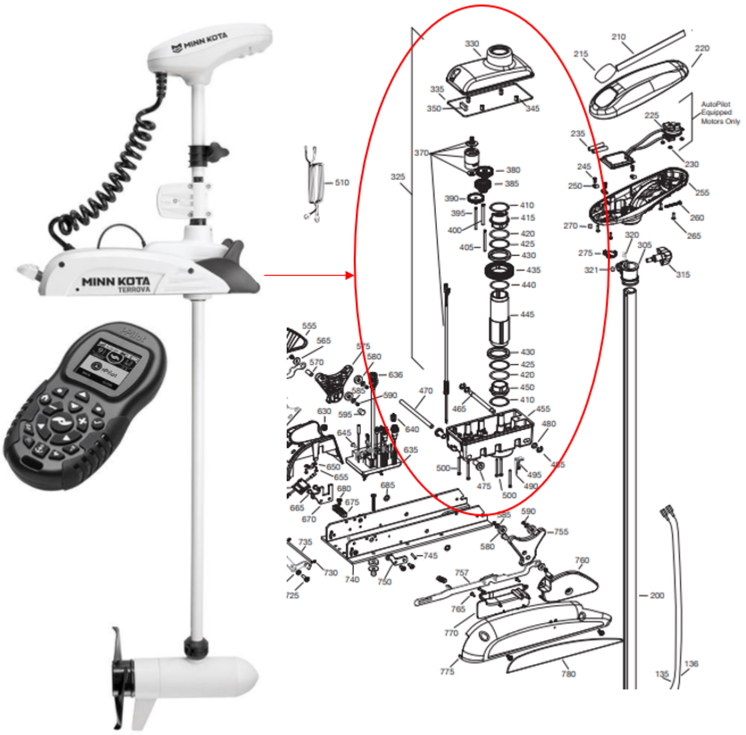


Figure 2.20. Minn Kota Riptide Terrova exploded view [117].

Often times, these smaller DC motors are then controlled with a Bluetooth handheld or foot control that allow operators to adjust the desired angle with little effort. In the case of the Terrova, it is even equipped with Minn Kota's "i-Pilot" system which includes GPS and allows for inputting of way points and heading cruise control. Other competitors offer similar functions as well.

Such products provide a large range of functionality, but come at a premium price point. Most of these enhanced trolling motors start at 14,000 kr. These drive-by-wire solutions are also comparatively slow to those used in other industries and/or applications. Trolling motors are used as a secondary motor, usually for the sole purpose of moving a fishing line through the water. For that reason, the application only requires slow movement and maneuverability is not a concern.

2.4 Modelling of Marine Crafts

The equations that describe the dynamics of ship manoeuvring are complex. Hence, when modelling marine crafts, simplifications are done and only a limited range of operational conditions are considered [11].

2.4.1 Coordinate System and Vector Notation

Many different types of reference frames exist but in this project two ones are mainly of interest: the North-East-Down (NED) coordinate system and a body-fixed frame of reference.

NED frame

The NED frame has the origin defined at a fixed point relative to the earth surface and has axis pointing in the North, East and Down directions [119]. For marine craft operating in a local area, with approximately constant longitude and latitude, an Earth-fixed tangent plane on the surface is used for navigation. Usually, this is referred to as flat Earth navigation and is for simplicity denoted by subscript n [30].

Body-fixed frame

A body-fixed frame has the origin defined at a point fixed to the moving vessel and has axis pointing in surge, sway and heave directions [119], as displayed in Figure 2.21. The position and orientation of the craft are described relative to the inertial reference frame while the linear and angular velocities of the craft should be expressed in the body-fixed coordinate system. The body frame is denoted by subscript b [30].

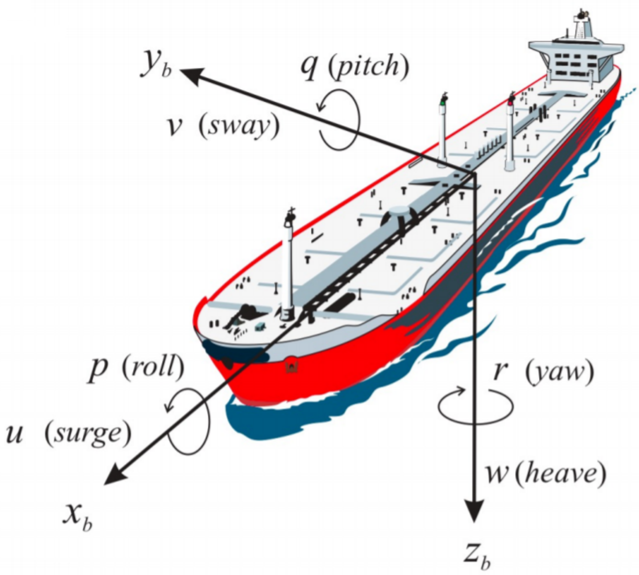


Figure 2.21. The 6 DOF velocities of a ship [30].

2.4.2 Kinematics

The kinematic equations of motion for a general 6-DOF marine vessel is derived and written in vector notation as [30]

$$\dot{\eta} = J_k(\eta)\nu \tag{2.1}$$

$$M\dot{\nu} + C(\nu)\nu + D(\nu)\nu + g(\eta) + g_0 = \tau + \tau_{env} \tag{2.2}$$

in which

- η is a vector containing the position and orientation in the NED frame,
- ν is a vector containing velocities in the body-fixed frame,
- $J_k(\eta)$ is a transformation matrix depending on Euler angles and maps the velocities in the body-fixed frame to the NED frame,
- M is the system inertia matrix, including added mass,
- $C(\nu)$ is the Coriolis–centripetal matrix, including added mass,
- $D(\nu)$ is the damping matrix,
- $g(\eta) + g_0$ are gravitational/buoyancy and hydrostatic forces and moments,
- τ is a vector of control inputs such as thruster forces,
- τ_{env} is a vector containing enviromental forces such as forces from the wind, from waves and from water currents.

Low speed model simplifications

By assuming that the pitch and roll are small, which is a good approximation for most conventional ships, underwater vehicles and rigs [30], and by neglecting the heave, since for a surface vessel the heave motion is not controllable [88], the general 6-DOF model may be reduced to a 3-DOF model only describing the horizontal motion (surge, sway and yaw) of the vessel. Under the assumption of low speeds, the model may be simplified further by neglecting the impact Coriolis and centripetal forces [9] as well as assuming that linear viscous damping dominates [119]. This yields the resulting simplified system as

$$\dot{\eta} = R(\psi)\nu \tag{2.3}$$

$$M\dot{\nu} + D\nu = \tau + \tau_{env} \tag{2.4}$$

where D is the linear damping matrix and $R(\psi)$ is the rotation matrix, which contains the surge, sway and yaw terms of $J_k(\eta)$, and is given by

$$R(\psi) = \begin{bmatrix} \cos\psi & -\sin\psi & 0 \\ \sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2.5}$$

This simplified model is hence described as the one displayed in Figure 2.22.

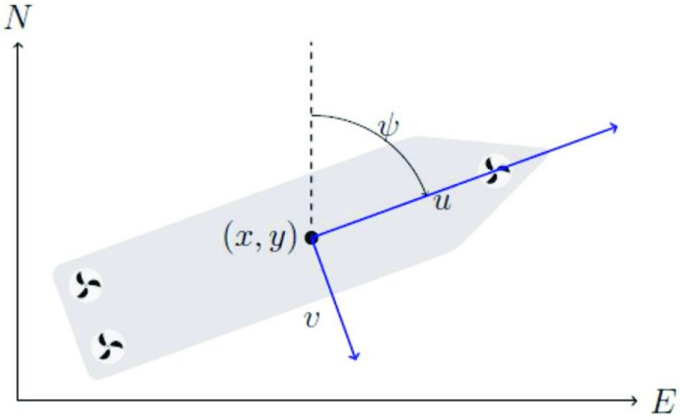


Figure 2.22. Schematic figure of the 3-DOF system model [78].

2.4.3 Hydrodynamic Forces and Moments

When modeling any marine vessel, the hydrodynamic forces and moments that are acting on it are needed to be taken into account. Commonly, it is assumed that they can be linearly superposed by considering two sub-problems [27]. Firstly, the radiation-induced forces and moments can be identified as three new components.

- *Added mass* due to the inertia of the surrounding fluid,
- *Radiation-induced potential damping* due to the energy carried away by generated surface waves
- *Restoring forces* due to Archimedes (weight and buoyancy).

Typically during low speed operation for surface vessels, the restoring forces can be neglected. Secondly, there are forces on the body when the body is restrained from oscillating and there are incident regular waves, *Froude-Kriloff* and *Diffraction forces* [29].

Added mass

The force from added mass is typically baked into the the system inertia matrix as $M = M_{RB} + M_A$, where M_{RB} is the rigid body inertia matrix and M_A is the added mass matrix. For low speed applications in 3-DOF, these can be calculated as [29]

$$M_{RB} = \begin{bmatrix} m & 0 & 0 \\ 0 & m & 0 \\ 0 & 0 & I_z \end{bmatrix}, \; M_A = \begin{bmatrix} -X_{\dot{u}} & 0 & 0 \\ 0 & -Y_{\dot{v}} & 0 \\ 0 & 0 & -N_{\dot{r}} \end{bmatrix} \tag{2.6}$$

where m and I_z are the mass and inertia of the vessel and $X_{\dot{u}}$, $Y_{\dot{v}}$ and $N_{\dot{r}}$ are hydrodynamic coefficients in accordance to SNAME (1950) [109]. These can be calculated by applying Kirchhoff's equations which relates the fluid energy to the forces and moments acting on the vehicle. But by approximating the submerged part of a ship as a half cylinder, the hydrodynamic force coefficients can be calculated as

$$-X_{\dot{u}} \approx 0.05m \tag{2.7}$$

$$-Y_{\dot{v}} = \frac{1}{2}\rho\pi D^2 L \tag{2.8}$$

$$-N_{\dot{r}} = \frac{1}{24}(0.1mB^2 + \rho\pi D^2 L^3) \tag{2.9}$$

where m is the mass of the vessel, ρ is the density of water, D is the hull draft, L is the length of the vessel and B is the vessel breadth [29].

Potential damping

The potential damping is highly nonlinear. But, for drive at low speeds, the linear viscous damping dominates. This means that $D = D_P$, where D_P is calculated in 3-DOF as [30]

$$D_P = \begin{bmatrix} -X_u & 0 & 0 \\ 0 & -Y_v & 0 \\ 0 & 0 & -N_r \end{bmatrix}, \tag{2.10}$$

where X_u, Y_v and N_r are hydrodynamic coefficients in accordance to SNAME (1950) [109]. According to seakeeping theory, the viscous damping can be estimated as

$$-X_u = \frac{m - X_{\dot{u}}}{T_{surge}} \tag{2.11}$$

$$-Y_v = \frac{m - Y_{\dot{v}}}{T_{sway}} \tag{2.12}$$

$$-N_r = \frac{I_z - N_{\dot{r}}}{T_{yaw}} \tag{2.13}$$

where T_{surge}, T_{sway} and T_{yaw} are time constants with values typically ranging between $100 - 250$ [30].

2.5 Control Allocation of Azimuth Thrusters

2.5.1 Actuator forces

Thrusters which individually can be rotated around the z-axis and hence can produce two force components (F_x, F_y) in the horizontal plane are usually referred to as azimuth thrusters. Using vector and matrix notation, these components relate to the vector of control inputs of the vessel as

$$\tau = T(\alpha)f \tag{2.14}$$

where α is a vector of azimuth angles, $T(\alpha)$ is the thrust configuration matrix and f is a vector of thruster force magnitudes [30]. The thrust configuration matrix is generally calculated as

$$T(\alpha) = \begin{bmatrix} \sum_{i=1}^{n} \cos\alpha_i \\ \sum_{i=1}^{n} \sin\alpha_i \\ \sum_{i=1}^{n}(l_{x,i}\sin\alpha_i + l_{y,i}\cos\alpha_i) \end{bmatrix} \tag{2.15}$$

where $l_{x,i}$ and $l_{y,i}$ are the coordinates of the thruster i in the vessels local frame of reference [52].

2.5.2 Input Signal Linearization

In order to steer the ship using its actuators we need to treat the actuator signals as input signals. Using the thruster configuration described by Equation 2.14 yields a nonlinear input signal relation, which can be hard to solve [30]. One way to work around this would be to instead linearize the input configuration and then instead use this linearized system as input. This is done by calculating and using the Jacobian matrix, which is defined as the matrix consisting of all partial derivatives of the nonlinear system [129]. One thing to note though is that a linearized system only is valid in close proximity to the point it is linearized around [76].

Also, two different linearizations are needed because of the two ways to steer the ship. One way would be to use simultaneous steering, i.e., the same input signals for both thrusters and another one would be to use differential steering, where the actuators are controlled independently [52].

Simultaneous Steering

For simultaneous steering, each thruster is considered to have independent, but the same, thurst and rotation angle, as $u_s = [f_i \quad \alpha_i]^T$. By then linearizing the thrust allocation matrix, Equation 2.15, around a working point $a_s = [f^* \quad \alpha^*]^T$, yields the linearized thrust matrix as

$$T_s = \sum_{i=1}^{n} \begin{bmatrix} \cos\alpha^* & -f^*\sin\alpha^* \\ \sin\alpha^* & f^*\cos\alpha^* \\ l_{x,i}\sin\alpha^* - l_{y,i}\cos\alpha^* & f^*(l_{x,i}\cos\alpha^* + l_{y,i}\sin\alpha^*) \end{bmatrix} \tag{2.16}$$

The vector of vessel control inputs, Equation 2.14, can hence now be written as [52]

$$\tau_s = T_s u_s = \sum_{i=1}^{n} \begin{bmatrix} \cos\alpha^* & -f^*\sin\alpha^* \\ \sin\alpha^* & f^*\cos\alpha^* \\ l_{x,i}\sin\alpha^* - l_{y,i}\cos\alpha^* & f^*(l_{x,i}\cos\alpha^* + l_{y,i}\sin\alpha^*) \end{bmatrix} \begin{bmatrix} f_i \\ \alpha_i \end{bmatrix} \tag{2.17}$$

Differential Steering

As for differential steering, each thruster is also considered to have independent thrust and rotation angle, only here they are allowed to differ from each other, as described by

$$u_d = \begin{bmatrix} f_1 & \cdots & f_n & \alpha_1 & \cdots & \alpha_n \end{bmatrix}^T \tag{2.18}$$

By then linearizing the thrust allocation matrix, Equation 2.15, around another working point $a_d = [f_1^* \quad \cdots \quad f_n^* \quad \alpha_1^* \quad \cdots \quad \alpha_n^*]^T$ would result in the linearized thrust matrix

$$T_d = \begin{bmatrix} \cos\alpha_1^* & \sin\alpha_1^* & l_{x,1}\sin\alpha_1^* - l_{y,1}\cos\alpha_1^* \\ \vdots & \vdots & \vdots \\ \cos\alpha_n^* & \sin\alpha_n^* & l_{x,n}\sin\alpha_n^* - l_{y,n}\cos\alpha_n^* \\ -f_1^*\sin\alpha_1^* & f_1^*\cos\alpha_1^* & f_1^*(l_{x,1}\cos\alpha_1^* + l_{y,1}\sin\alpha_1^*) \\ \vdots & \vdots & \vdots \\ -f_n^*\sin\alpha_n^* & f_n^*\cos\alpha_n^* & f_n^*(l_{x,n}\cos\alpha_n^* + l_{y,n}\sin\alpha_n^*) \end{bmatrix}^T \tag{2.19}$$

With this, the vector of vessel control inputs, Equation 2.14, can now be written as [52]

$$\tau_d = T_s u_d = \begin{bmatrix} \cos\alpha_1^* & \sin\alpha_1^* & l_{x,1}\sin\alpha_1^* - l_{y,1}\cos\alpha_1^* \\ \vdots & \vdots & \vdots \\ \cos\alpha_n^* & \sin\alpha_n^* & l_{x,n}\sin\alpha_n^* - l_{y,n}\cos\alpha_n^* \\ -f_1^*\sin\alpha_1^* & f_1^*\cos\alpha_1^* & f_1^*(l_{x,1}\cos\alpha_1^* + l_{y,1}\sin\alpha_1^*) \\ \vdots & \vdots & \vdots \\ -f_n^*\sin\alpha_n^* & f_n^*\cos\alpha_n^* & f_n^*(l_{x,n}\cos\alpha_n^* + l_{y,n}\sin\alpha_n^*) \end{bmatrix}^T \begin{bmatrix} f_1 \\ \vdots \\ f_n \\ \alpha_1 \\ \vdots \\ \alpha_n \end{bmatrix} \tag{2.20}$$

2.6 Positioning

An accurate position and orientation of the vessel will be needed during the docking. In this section, how this can be measured is presented.

2.6.1 Global Navigation Satellite System

GNSS is a constellation of satellites that transmits signals with positioning and timing data to GNSS receivers, in order to determine the location of said receiver. GNSS provides global coverage with multiple satellites. Europe has Galileo, The

United States of America has GPS, Russia has Global'naya Navigatsionnaya Sputnikovaya Sistema (GLONASS) and China has BeiDou Navigation Satellite System [110]. GNSS requires a clear path and can easily be blocked by buildings and when travelling through tunnels. To combat this when developing Autonomous Road Vehicles, a technique called Dead Reckoning is used. It uses the last known location together with input from sensors, such as accelerometers, to calculate the change in location by estimating speed and heading. This is however not an issue when used together with naval vehicles since the signal is harder to block in open waters. Independent use of GNSS offers an accurate position within a couple of meters.

2.6.2 GNSS Correction Services

In order to increase the accuracy of GNSS, there are different correction services. Two of them are Differential Global Navigation Satellite System (DGNSS) and RTK. Correction data is sent with the Radio Technical Commission for Maritime Services (RTCM) protocol, which is sent with Networked Transport of RTCM via Internet Protocol (NTRIP). The GNSS receiver must therefore be able to operate with NTRIP in order to utilize correction services.

Differential GNSS

DGNSS is a technique were several GNSS receivers are used together to get a higher accuracy than independent use of only one receiver. The accuracy for DGNSS is around a meter, depending on equipment [62]. Therefore, it is not suitable for use with Autonomous Vehicles, which requires a much more accurate localization.

Network RTK

Network RTK uses a number of reference stations situated around the world, and provides the GNSS receiver with its true position by calculating its own error in position. This can be seen in Figure 2.23. This provides an accuracy of up to 30 mm if the receiver is within 70 km from a base station, and as low as 25 mm if within 35 km. SWEPOS is Lantmäteriet's support system for satellite positioning in Sweden and they provide a service for Network RTK. Network RTK uses a two-way transmission. When using a single temporary base station the communication can be done with Telemetry Radio, but SWEPOS requires an internet connection.

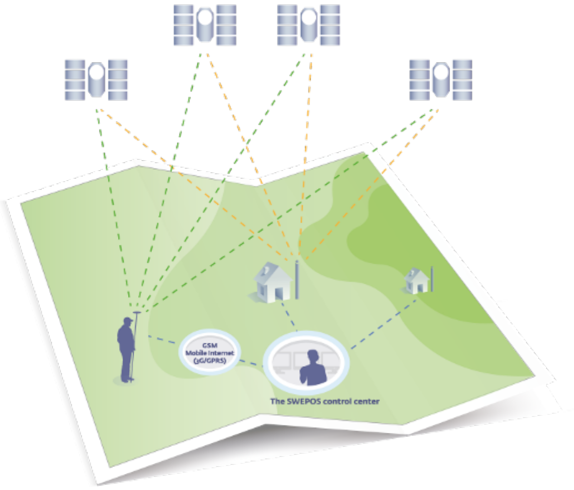


Figure 2.23. When measuring with network RTK, users send in their position to the service's operations management center and get back ”tailored” correction data for their position [61].

2.6.3 Heading

To determine the correct heading of autonomous marine vessels, some kind of IMU is often used. Standard IMU combine 3-axis accelerometers, gyroscopes and sometimes magnetometers as well. Sometimes a fully integrated electronic compass module is used, they combine 3-axis magneto-resistive sensors with accelerometers and are usually tilt-compensating the angular heading. This can provide an accurate heading within 1 degree.

2.7 Control Theory

In this section, control theory which is used in previous similar work, and most likely will come into play in this project, is presented briefly.

2.7.1 Hybrid Systems

Hybrid systems are systems which incorporate both continuous and discrete event models into it and are extensively used in air traffic management systems and for intelligent vehicle control [12]. As an example, the gear shift of a car could be modeled as a hybrid system, as displayed in Figure 2.24. Here, the position of the car on the road is denoted as x_1 and the velocity of it as x_2 and are considered outputs of the system. For inputs, the model has two control signals. The gear, denoted as **gear** $\in \{1, 2, 3, 4\}$, and the throttle position, denoted $u \in [u_{min}, u_{max}]$. The function α_i denotes the efficiency of gear i. The shifting of gear is necessary since little power can be generated by the engine at low or high engine speed [53].

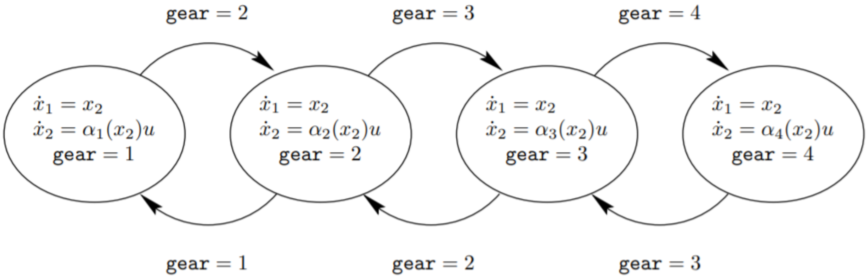


Figure 2.24. A hybrid system modeling a car with four gears [53].

Hybrid Automata

A formal way to model a hybrid system is to do it as a hybrid automaton. This is done as

$$H = (Q, X, Init, f, D, E, G, R) \tag{2.21}$$

where H is a collection in which

- Q is a set of discrete states,
- X is a set of continuous states,
- $Init$ is a set of initial states,
- f is a vector field,
- D is a domain,
- E is a set of edges,
- G is a guard condition,
- R is a reset map.

Also, $(q, x) \in (Q, X)$ is referred to as a state of H [53].

In an attempt to make this a bit clearer, an example is displayed in Figure 2.25 in which a transition, $(q, x) \to (q', x)$, is described. Starting from an initial state, $(q_0, x_0) = (q, x) \in Init$, the continuous state, x, evolves according to the dynamics described by the vector field, $f(q, x)$, while the discrete state, q, remains constant. This evolution can go on as long as x remains in the domain $D(q)$. Suppose then that, at some point, x reaches the dynamics of a guard condition, $G(q, q')$, than, the discrete state of the system may change according to the edge, E. In this case this would imply that the discrete state of the system changes from q to q'. At the same time the continuous state gets reset to some value which is described by the reset map, $R(q, q', x)$.

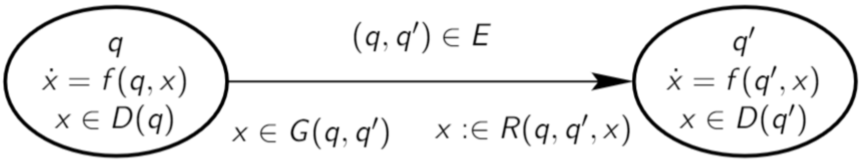


Figure 2.25. Example of a transition described by a hybrid automaton.

2.7.2 Controllers

The controllers commonly used for motion control of ASVs are mentioned in section 2.1.5. A combination of these control systems are used in hybrid systems, where they regulate different states of the vessels. A shorter overview of these types of controllers are given below.

PID Control

PID control is commonly used in feedback and feed-forward control. Here, the error between the output and input to the system is regulated to achieve desired output. Increasing the proportional part of the controller increases the speed of the system but is also more prone to introduce instability. The integrating part is used to eliminate steady state error in the output, but is also prone to introduce instability. The derivative part may be used to increase stability, though it is not unusual to omit this part since differentiating noisy measurements may be precarious [34].

Linear Quadratic Control

LQR is a type of optimal control. Simply put, the goal of LQR is to minimize the square of the error between the output and the reference as well as the square of the control signal. The regulator can also be extended to handle Gaussian noise [35].

Sliding Mode Control

SMC is a nonlinear control method which use high gain control to approach the sliding surface in finite time and then slide on the (stable) surface to approach the desired point. The method is robust to disturbances on the input and can be extended to be robust against model uncertainties. Typically for SMC is that the trajectory does not stay identically on the sliding surface and instead oscillates around it [57]. A typical SMC phase portrait is displayed in Figure 2.26.

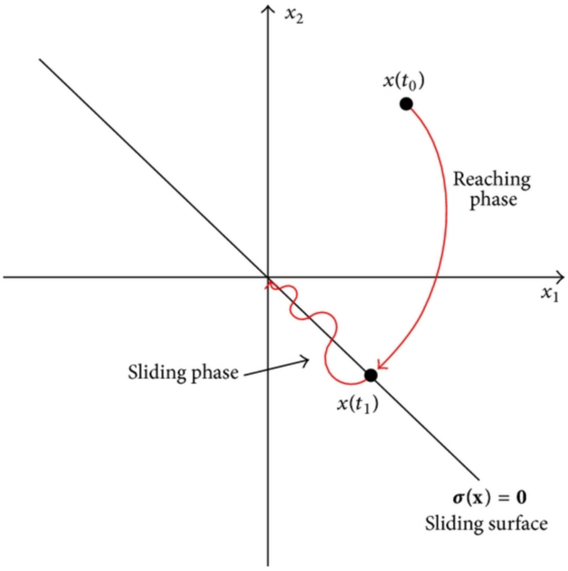


Figure 2.26. Typical Sliding Mode Control phase portrait [134].

Model Predictive Control

MPC uses the predictive power in a model of the controlled object to calculate the control signal. The control signal can be chosen so that a given criterion is minimized, subject to physically motivated bounds on the signals [35]. Simply put, MPC uses at each sampling instant the plant's current input and output measurements, the plant's current state, and the plant's model to calculate a future control sequence that optimizes a given performance and satisfies constraints on the control action as well as uses the first control in the sequence as the plant's input [5].

2.7.3 Gain Scheduling

Gain scheduling is a common approach to deal with control of nonlinear systems. The idea is that the control parameters will depend on the operating conditions. Considering a nonlinear system, the method is then to linearize it around a number of equilibrium points, which would result in a collection of linear systems. For each these system, linear control is then used to control each of these systems [76]. Between the equilibrium points, a gain schedule is then used to determine the appropriate gain to be used by the controller. The overall method is briefly summarized in the block diagram in Figure 2.27.

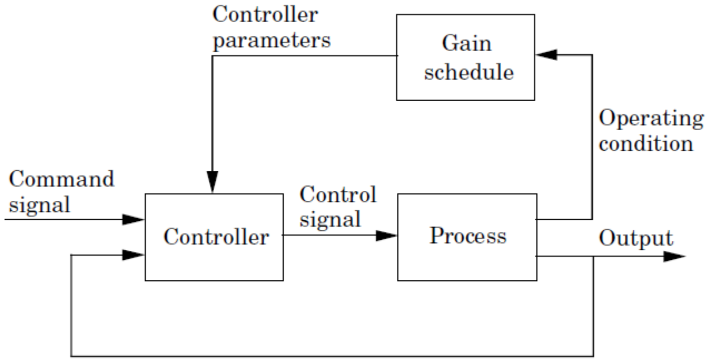


Figure 2.27. Gain scheduling block diagram

2.7.4 Kalman Filtering

Kalman Filtering, also know as Linear Quadratic Estimation, is used for sequentially estimating the states of a linear Gaussian dynamical system. The algorithm used has a low computational complexity and can therefore be implemented for real-time applications [59]. KF is used to solve a large variety of problems, one example is loss of signal with GNSS applications where dead reckoning is used, as mention in Section 2.6.1. The way KF work is recursively. As displayed in Figure 2.28, where a simplified overview of the algorithm is shown, the algorithm consists mainly of two parts, the prediction step and the correction step. Firstly, a prediction of the system is made based on what inputs are available. Then, based on measurements, correction is made to the prediction. This is then iterated until the prediction is sufficiently accurate [13]. Though KF only can be used on linear systems of equations, it can be extended to work with nonlinear systems. The KF is then instead called an EKF and this is done by utilizing Taylor series expansion on the nonlinear system to be filtered [57].

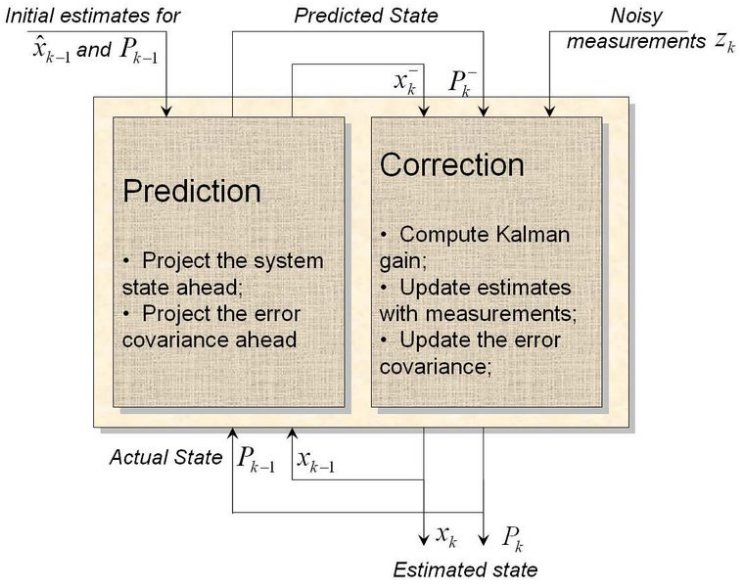


Figure 2.28. Prediction/Correction algorithm cycle of the KF [101].

2.7.5 Path Planning

Path planning can be defined as the problem of finding a route between two positions in a mobile space whilst taking into account that the route should be collision-free, physically feasible within spatial constraints as well satisfy certain optimisation criteria. Commonly, these optimisation criteria for path and trajectory planning include minimisation of path length, time, and energy consumption, as well as measures of safety or risk. Path planning is typically defined solely within geometric space whereas trajectory planning has a wider scope and includes dynamics such as angle limits, velocity and acceleration constraints. As for path planning algorithms, they could be classified into taking one of three approaches [122].

An algorithm taking a classical approach involves firstly to model the environment and the secondly to perform a search for the optimal path in this environment. This approach is suitable where there is no need for path re-planning or local collision avoidance. An advanced approach is taken when dealing with dynamic obstacles, path re-planning or local collision avoidance in real time and often do not require environmental modelling beforehand. These approaches include machine learning algorithms, potential field methods and velocity space methods. As for hybrid approaches, these include algorithms which include several different path planning algorithms [122]. Displayed in Figure 2.29 is an example of a potential field method, where the goal is given attractive potential and the object in the way is given repulsive potential. This makes the vessel moving trough the plane move to the goal while at the same time being repelled by the object [16].

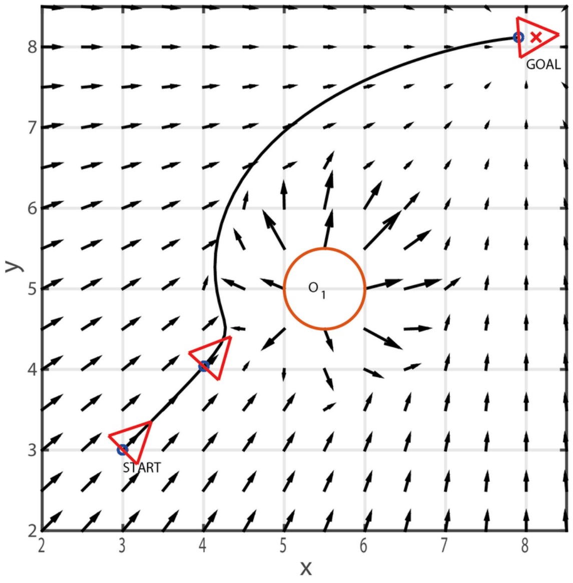


Figure 2.29. Example of artificial potentials path planning [28].

2.7.6 Path Following Using Pure Pursuit

In order to follow a path, different types of algorithms could be used. One commonly used one is pure pursuit. This is a tracking algorithm that works by, from the vehicles initial position, calculate what curvature the vehicle has to follow in order to reach some goal position. The goal of the algorithm is to always find some goal position some distance ahead on the path the vehicle is going on [123]. The algorithm does this by calculating a theoretical circle, with some radius, around the vehicle, calculating where this circle intersect with the path to be followed and steers towards that point. This point is always located at one radius from the vehicle and is often called the look-ahead point. The distance to this point is the look-ahead distance [17]. This is graphically visualized in Figure 2.30 below.

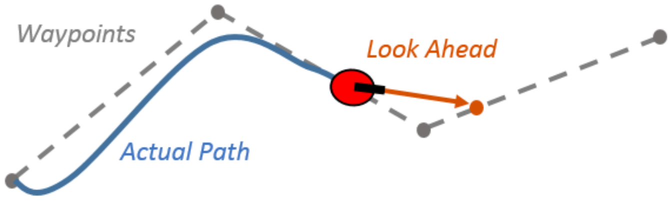


Figure 2.30. The Pure Pursuit algorithm [93]

2.8 Dock and Obstacle Detection

The Dock and Obstacle detection subsystem has two main purposes. The first is to be able to detect and localize the dock. The other is to detect and localize obstacles on the way there. In this section hardware and software that would enable fulfilling those two purposes was investigated.

2.8.1 Hardware

To be able to detect objects, sensors that can interpret the surroundings is needed. In similar projects as seen in Section 2.1 the way this has been done is using sensors like LiDAR, Stereo Cameras and/or Radar. In this section these sensors will be presented and explained.

LiDAR

LiDAR utilizes laser light pulses to measure the distance to surrounding objects (see an illustration of this in Figure 2.31). To get these distances the Time-of-Flight (TOF) is measured, in other words the time it takes for the beam to hit the surrounding and return. The distance can then be calculated by multiplying the time of flight with the speed of light and dividing it by two to get the one way distance[107].

The main advantages of LiDAR is the accuracy. The short wavelength of LiDAR enables it to detect small objects at a far distance. A LiDAR with a wavelength of 1500 nm is possible to accurately detect an object the size of a tire at a distance of 150 to 200 meters [86]. Another big advantage for the LiDAR is the possibility of a 360 Field of View (FOV). Enabling the sensor to detect obstacles all around the vessel [50]. Another trait that speaks for the LiDAR is that its not dependent on good lightning conditions. Since it sends out it's own light it operates well during the darker hours too [133].

The LiDAR also has it's down sides. The biggest disadvantages with LiDARs is the hefty price tag. Even though the price has come down in recent years a 3D LiDAR with 360 degree FOV can cost anywhere from 35 000 kr to 150000 kr [81]. Another drawback with LiDAR is its performance in bad weather [133]. The rain makes the surfaces that the LiDAR beam bounces of wet and because of this the surfaces becomes less reflective. As a result the range at which the LiDAR can operate in reduces. At a precipitation of 25 mm/h the effective range can be reduced by up to 20 percent. Another way that the LiDARs effective range can be reduced is in the case of foggy weather. The small droplets caused by the fog makes the beams scatter. The scattering of the beams reduces the operational range of the LiDAR by up to 50 percent. [8] .

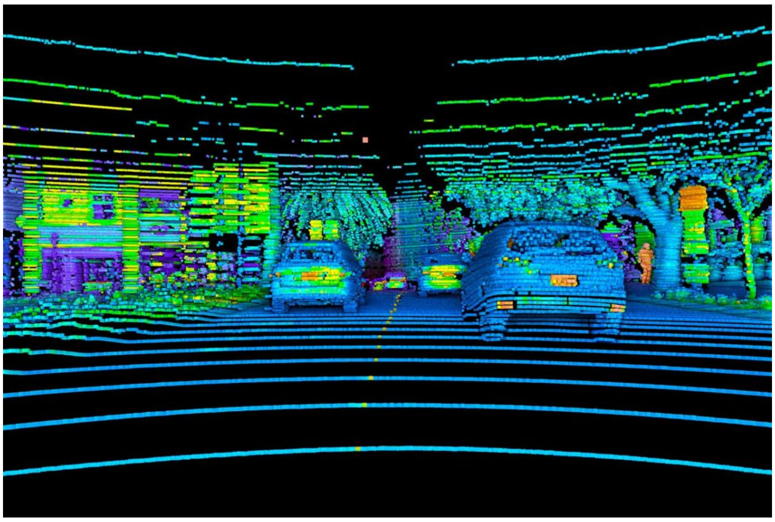


Figure 2.31. Visualization of LiDAR mapping [65].

Stereo Vision

Stereo vision is a technology utilizing two passive visible light cameras. The cameras are positioned a known distance from each other and have a parallel viewing direction. The distance to points in the image is calculated by comparing the two images and measuring the disparity between the two. As you can see in Figure 2.32 both cameras capture C and D. Then the parallax between what each camera sees makes it possible to calculate the distance to C and D.

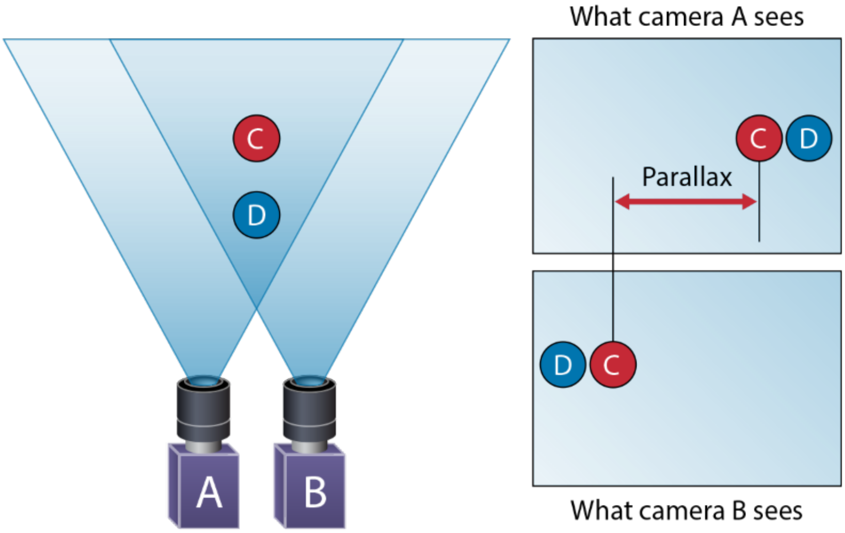


Figure 2.32. How the distance is derived with two cameras [45].

One of the advantages with using a stereo camera is the possibility to use algorithms to enhance the visuals. These algorithms can be used to improve the detection in for example bad weather situations. Improving the detection will as a result also improve the operational range. As shown in this paper an effective object detection can still be achieved in rainy condition if a visual post processing algorithm is applied [32]. Another paper found that the same concept can be applied in foggy weather. If the image is post processed a good object detection can be accomplished in foggy conditions as well.[55]

The stereo cameras have qualities that are advantageous, but it also has its downsides. One of the main limiting factors of a stereo camera is localization range. The distance of which the cameras can detect objects depends a lot of what FOV, but as an example the ZED 2 by Stereolabs has a FOV of 110 degrees and a range of 20 m [137]. Something that must be considered when using a stereo camera is the need for good external lightning. Since the cameras are only passive and do not produce any light on there own the surrounding needs to be well lit.

Radar

Radar is a well known technology with a lot of variation. Common of all radars is that they send out electromagnetic waves and then receive the reflected response. Radar frequencies can range from 5 MHz to 130 GHz [132]. So called millimeter wave radar is commonly used in ranges of up to 100 m, which is applicable in this project. Millimeter wave radar commonly has a frequency of about 75 GHz. The specific radar technology used in these types of implementations is called Frequency-Modulated Continuous Wave (FMCW). It is a strategy for obtaining accurate radar distance measurements based on modulating the frequency of the signal in a repeating wave pattern [131]. The shape of the signal wave can take on many different shapes, a saw-tooth configuration for example gives the best range capability while a square wave has shorter range but higher accuracy. An example of a saw-tooth wave can be seen in Figure 2.33, the horizontal axis is time and the vertical is frequency .

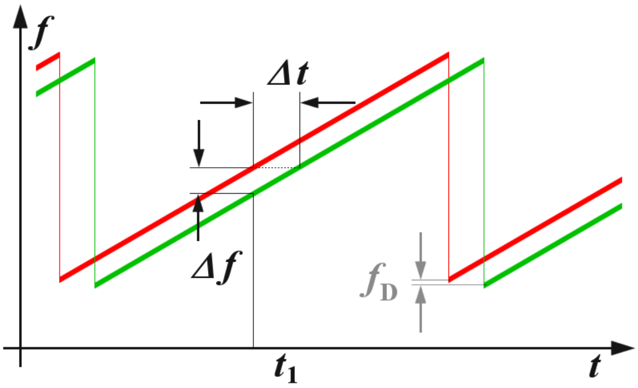


Figure 2.33. Example of a FMCW waveform [131].

The distance to an object is calculated based on the time-shift Δt between the outgoing and the received signal. See Equation 2.22, where R is the distance in meters and c_0 is the speed of light in m/s.

$$R = \frac{c_0 \Delta t}{2} \tag{2.22}$$

2.8.2 Real-Time Detection

Markers

Usage of some kind of marker attached to the dock was investigated. The advantage of some kind of marker is that the vision detection problem is greatly simplified. One can even go a step further and use a localization tag like the AprilTags, see Figure 2.34, developed by the University of Michigan [121]. A predefined tag has the benefit of being of a known shape and size. This gives the vision system the ability to relatively easily determine the distance and angle to the tag based on the perceived size in the cameras field of view. A tag can also be scaled up almost indefinitely to increase the range of the location system.

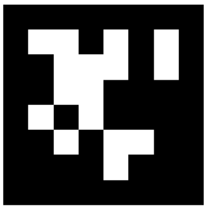


Figure 2.34. An example of an AprilTag [121].

Template Matching

Template matching is a technique within computer vision that uses templates to find objects in images. These templates can be features of the object you want to detect or the whole object. The idea is that with a template scan through the image and with different techniques determine the likelihood that the object is in the image. As an example of template matching you can see in Figure 2.35 were the cup is the template and the image in the middle is the image we want to scan for it [113]. One of the big advantages with using template matching is that it doesn't demand any training. The matching is done at runtime and even if it sees the same image multiple times it will still give the same output. This is also one of its biggest weaknesses since the algorithm doesn't learn from its mistakes and improve over time [51].

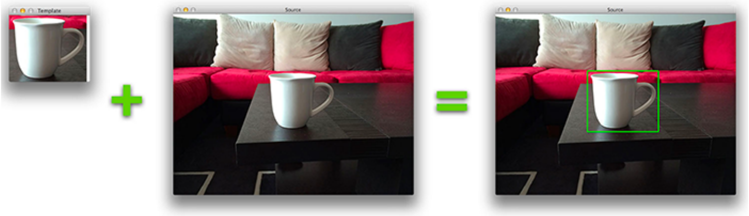


Figure 2.35. Example of template and image to match it with [104].

Grid Mapping

Occupancy grid mapping is a way to represent the environment and is popular in autonomous vehicles for localisation, collision avoidance and path planning [64]. The issue of mapping in robots is usually referred to as Simultaneous localization and mapping (SLAM). Occupancy grids uses cells that indicate if an area is either occupied, free or undetected. LiDAR and radar are commonly used sensor for creating the occupancy grid maps, but the depth images generated from a stereo vision camera can also be used to create the occupancy grid map [64].

SLAM is considered a difficult problem since the robot needs a consistent map to determine its localization and it needs a good estimation of its location in order to get the map. Rao-Blackwellized particle filters (RBPF) was introduced as an effective way to solve the SLAM issue[40]. RBPF can be complex and require a lot of samples, but samples can reduced using recent sensor information, the odometry and a scan-matching process, which will improve performance[40].

Image recognition with Neural Network

Image recognition in deep learning is used for identifying images and categorizing them in predefined and pretrained classes.

Image recognition can be implemented using a Convolutional Neural Network (CNN). The CNN will give a probability score for every predefined class and then determine which label most likely applies for the given image input. Depending on the implementation, the CNN could give a label for the image as a whole or it could divide the given image input into various sections for detecting multiple classes in one image. When dividing the image into sections it also becomes possible to get the position of the class in the image. An example of this can be seen in Figure 2.36.

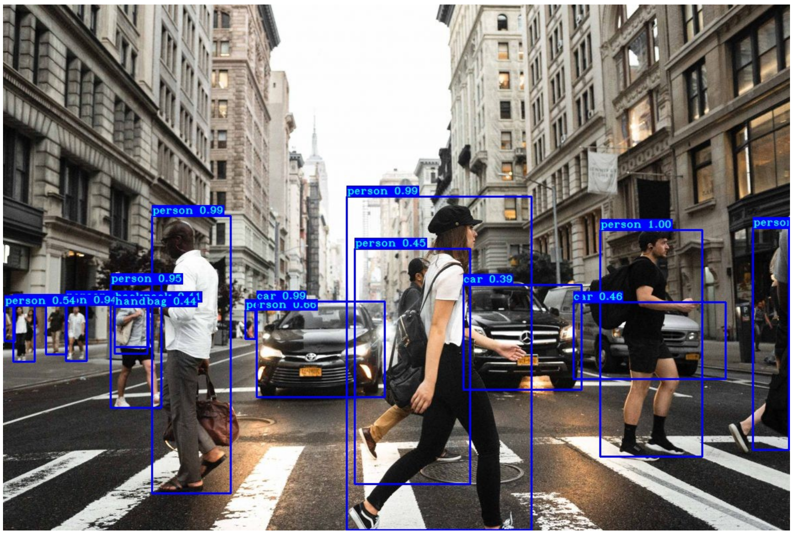


Figure 2.36. Example of image detection using the YOLOv3 algorithm [20].

Neural networks have achieved promising results in various fields of pattern recognition, not only image recognition but also fields such as voice recognition, determining/predicting weather and advertising [2].

A neural network consist of nodes (also known as neurons) that are placed in layers. The layers are placed in a sequence and the every node in each layer have a connection to every node in the next layer in the sequence as seen in Figure 2.37.

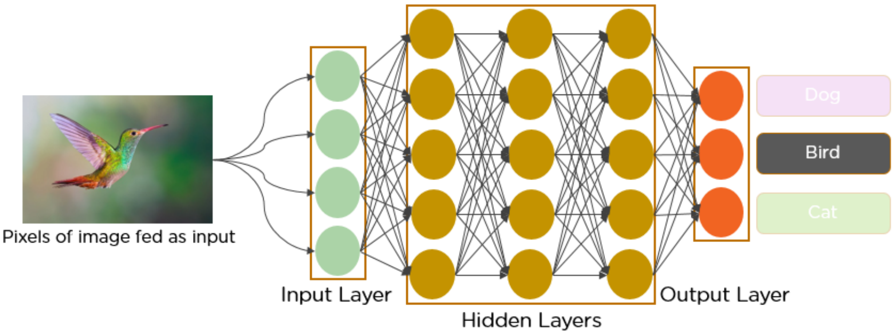


Figure 2.37. Example of a Convolutional Neural Network [73].

In a way, a neural network attempts to imitate the way a brain works. Each of the connections between the nodes have a parameter called a weight which is one of the main parameters to be tweaked during training and to infer a prediction from the given input data. Since every node is connected between the layers, there will be a lot of parameters to take into account both when making a prediction and when training. For example using a small 32x32 color image, there would be 3072 weight connections for every new node that the input layer is connected to. So if the second layer has as many nodes as the input layer, that would mean 3 145 728 connections between just two layers. A more efficient method for connecting layers than fully connecting them is to introduce convolutional layers which looks at local regions of an image and not the image as a whole. It could for example connect to 5x5 neurons and keep the local connection weights fixed, dropping the number of weights to only 5x5x3 = 75 [4]. These convolutional layers are great at detecting patterns, making it a good choice for image recognition [4]. Although the number of weights can be dropped, inference and training can still take a long time, a good GPU can be used for making the process faster [79] and making it more viable for realtime detection.

YOLOv3 is a state of the art algorithm for detecting objects in real time using 106 fully convolutional layers [96]. The YOLO algorithm generates bounding boxes around a prediction and also provides probabilites for each bounding box. The algorithm also uses non-max suppresion to make sure that each object is only detected once. Using an Nvidia Titan X graphics card and an input resolution of 256x256, YOLOv3 was able to run at 78 frames per second [96]. This is actually slower than the previous version, YOLOv2, but v3 has an increased accuracy.

Data augmentation for training datasets

Large datasets are usually required when training a neural network. The more complex the network is, the more parameters there are to train. This means that more data is likely to be required for training the network. Training a CNN on a small dataset can make it prone to overfitting leading to it being very inaccurate on unseen data and only making it accuracte on the training data [115]. This could cause issues when the available data is limited or if there is not enough resources or time to collect data. A way to solve limited datasets is to use data augmentation to increase and diversify the dataset without actually collecting more data. An example of a basic augmentation can be seen in Figure 2.38.

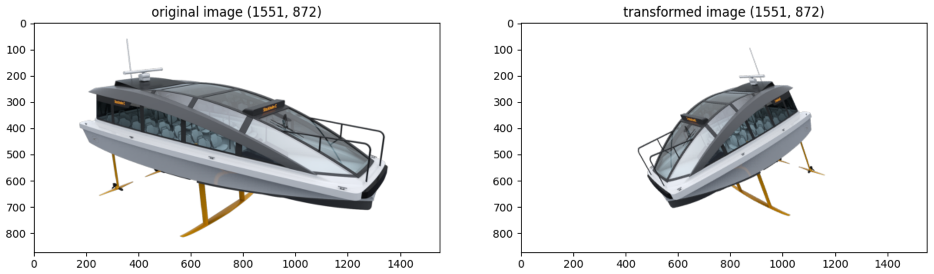


Figure 2.38. Image transformed by flipping it, giving it a random rotation and warped perspective.

Experiments have shown that by purely using cropping augmentation on a dataset, the accuracy can be improved from 64.5% to 79.1% [115].

Computing

As mentioned in Section 2.8.2 a CNN requires a lot of computing power for both training and inference. For training the CNN an option could be to use Google Cloud. This service provides access to a remote computer with great computing power. The GPUs that Google Cloud offer are Nvidia K80, P100, P4, T4, V100 and A100 [38]. The GPUs available at Google Cloud are specifically designed for performing simulations and large-scale calculations.

While running inference however, Google Cloud would not be an optimal choice if real time detection is desired. The reason for this is that Google Cloud runs on a remote computer which means that there would be delays and a stable internet connection would always be required for transferring. Nvidia offers a series of embedded computing boards called Jetson. Jetson are specifically designed for accelerating machine learning applications and is a low-power system. For example the Nvidia Jetson Xavier NX runs on 10W and the onboard GPU has 384 CUDA cores and 48 Tensor cores [87].

2.9 Near-Field

The main purpose of this subsystem is to highly accurate assist the boat during the final meters of the docking procedure and also to ensure contact to the dock during embarkation/disembarkation. Additionally, it is also supposed to be able to detect any hazards within close range around the boat. In order to fulfil this, a number of sensors and solutions were investigated.

2.9.1 Proximity sensing

As there are many options for proximity sensing, the main inspiration for the possible sensor solutions were taken from the automotive industry, which utilizes ultrasonic sensors and mmWave radar to a high extent.

mmWave Radar

As mmWave radar operates in high frequencies as presented in Section 2.8.1, it's able to detect movements that are a fraction of a millimeter within a range from a few centimeters up to over 100 meters, depending on object [97]. It can provide range, velocity and angle to the detected object. This is done by measuring the time for the optical signal to return, how the signal has changed from the reflection and how the phase shift between in- and outgoing signal differs. The mmWave radar is also impervious to environmental conditions such as sunlight, rain, fog, dust and snow and can penetrate most materials except metal, making it possible to be mounted behind surfaces without needing a clear line of sight [97].

Ultrasonic sensor

An ultrasonic sensor detects the presence of an object and measures the distance to it, without making physical contact. The effective measuring range in air is from a few centimeters up to several meters, depending on exact sensor and object properties. The sensor generates and emits ultrasonic pulse waves that are reflected back by an object and captured by the sensor, as seen in Figure 2.39. As the speed of sound is a known variable, the sensor can capture the time difference between it's emitted and received echo signal, which is called a TOF measurment. The TOF can be used to calculate the distance between the sensor and the object accordingly to Equation 2.23, where t_r is the round-trip time and $v_s = 343m/s$ is the speed of sound in dry air at 20°C.

$$d = \frac{t_r v_s}{2} \tag{2.23}$$

Ultrasonic sensors can detect any object regardless of shape, transparency and color with the requirement that the material of the object is solid or liquid, i.e. non-sound absorbent [85]. However, the heading angle and object shape decides how well the object will reflect the ultrasonic waves. Round or tilted object may scatter a majority of the waves transmitted to the object, see Figure 2.40. This means that the object of interest preferably should be large, dense, flat and smooth, facing the sensor at a 90° angle to yield the best sensor response [85]. Ambient conditions as temperature, humidity and debris is also a factor needed to take into consideration. Dust, rain, snow could alter the sensor field-of-view and disturb the performance. If the sensor would be partially submerged in water or covered by snow, mud or ice, ranging performance could be reduced.

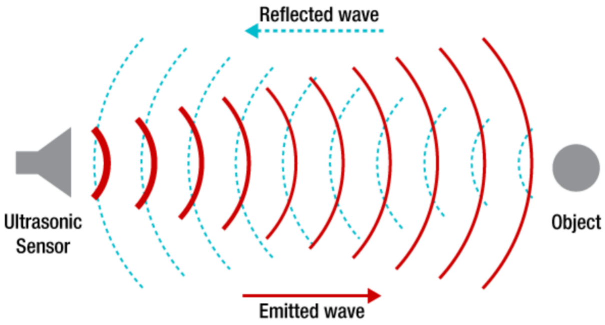


Figure 2.39. Ultrasonic sensor waves [85].

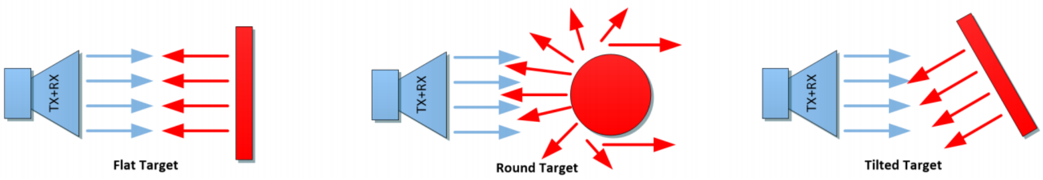


Figure 2.40. Round and tilted targets may scatter the waves. [85].

2.9.2 Dock contact sensing

To determine if the boat is in contact and stays in contact with the dock, the option of using load cells were investigated.

Load cell

The purpose of a load cell is to measure force and transform it into some readable information, usually an electrical signal. There are many different types of load cells and the range of measurable load varies a lot [130]. Some sensors types for larger loads are hydraulic and pneumatic. They use liquid and gas within the sensor and a pressure gauge measuring the pressure due to the load. There is also capacitive load cells for smaller loads. These contains two flat plates functioning like a capacitor. When the plates are pressed together the capacitance change it can be re calculated the the load upon the sensor. One of the most commonly used type of load cell uses strain gauges. The gauges change resistance when in strain and by measuring the change, the load can accurately be determined.

Also within the stain gauge load cells there is a great variety of cells. Still they all use the same technique, by sticking the strain gauge to a strong but elastic material such as steel or aluminum, the cell can bear a lot of weight and only temporally deform slightly. [42] That causes the resistance in the strain gauge to change, see Figure 2.41. The resistance change is very small, but by putting the strain gauge in a Wheatstone bridge, the resistance change can be measured very accurately. The slight resistor change gives sensor output in millivolts. To be able to read the analogue voltage output signal, the sensor may be connected to an A/D converter. A/D converters does not always have high enough resolution to register every mV shift, and differentiate it from other noise. In those cases a load cell amplifier may be used. The amplifier receive the mV, and amplify it to 0-10 V or 4-20 mA.

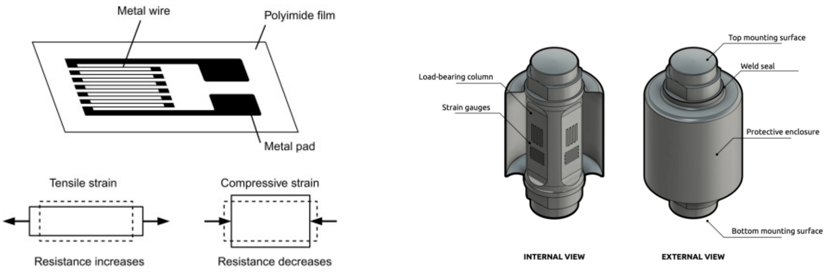


Figure 2.41. When the load cell is compressed, the resistance in the strain gauge will change [42] [118]

.

2.10 Communication protocols

There are a number of communication protocols available to create an integrated system. Both for communication between micro controllers and communication with sensors. This section will cover the most common relevant protocols.

2.10.1 CAN-Bus

CAN is an electronic communication bus with its roots from the automotive industry where it is crucial that plenty of systems are able to communicate with each other. CAN consists of a backbone to which you connect several nodes (systems). The protocol allows each node to both transmit and receive messages from every other node. Meaning when a message is sent by one node, all the other nodes hears that message. For a node to only act on messages of interest, there's a possibility to set address filters to the node. The most common type of CAN message can carry 8 bytes of information. The structure of a data frame can be seen in Figure 2.42.

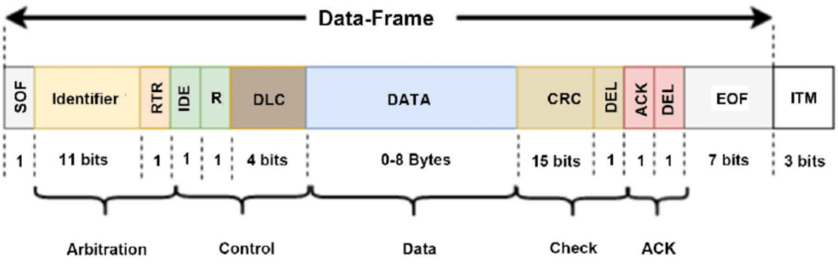
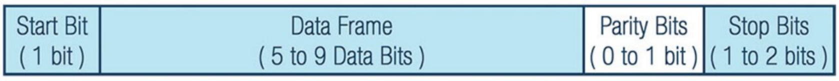


Figure 2.42. Common structure of a CAN data frame [15]

2.10.2 UART

Universal Asynchronous receiver-transmitter (UART) a communication protocol used for asynchronous serial communication. Mainly in device to device communication between computers, micro controllers and Embedded Systems. UART transmits data through a packet that is visualized in Figure 2.43. The communication is serial, meaning that the bits are sent one by one in a series. As seen in Figure 2.43 the communication starts with a start bit then the data followed by a parity bit that checks for data corruption and finally a stop bit.UART is transmitted at certain baud rates ranging from 9600 to 1500000. [120]



Start Bit (1 bit)	Data Frame (5 to 9 Data Bits)	Parity Bits (0 to 1 bit)	Stop Bits (1 to 2 bits)

Figure 2.43. Packet for UART [120]

The UART protocol needs three cable for transmitting, one for receiving and one for common ground, as depicted in Figure 2.44. This simple connection method makes UART a good alternative in many intra-controller communication situations.

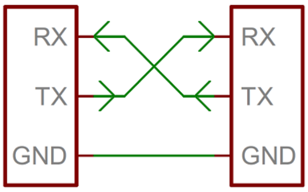


Figure 2.44. Wiring UART [105]

2.10.3 SPI

SPI stands for *serial peripheral interface* and is a high rate serial communication interface. It uses a master-slave architecture, meaning a master device (for instance a micro controller) communicates with one or several slaves (for instance sensors).

The SPI bus specifies 4 logic signals: SCLK, MOSI, MISO and CS. The *SCLK* is the clock line which is controlled by the master device and are used to synchronize the data transfer between the master and the slaves. MOSI stands for *Master Output Slave Input* which are used when the master needs to send data to the slaves. MISO stands for *Master Input Slave Output* and are used to send data from the slave to the master. CS stands for *Chip Select* and are controlled by the master to chose which slave on the bus that it wants to communicate with. These 4 signals constitute the basic layout of the SPI interface [19]. The interface can easily be modified to connect extra slaves to the bus by configuring additional Chip-Select peripherals on the master device. An example of how SPI communicates can be seen in Figure 2.45.

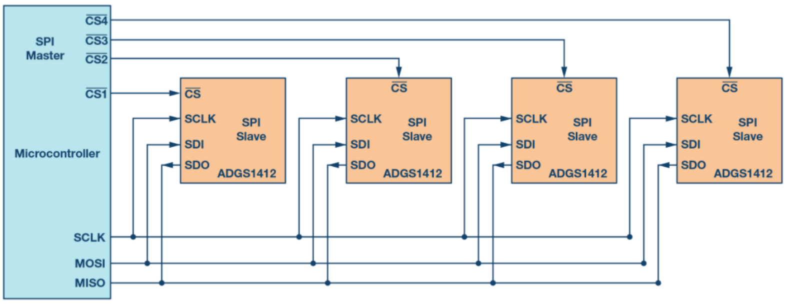


Figure 2.45. The structure of the SPI interface

The data transfer starts when the master device pulls the chip-select line low(active low) for the slave that it wishes to communicate with. This will start the data transfer over the MOSI and MISO line with the selected slave. The data is transferred in a serial manner which is synchronized by the tick on the SCLK line. Once the communication is done, the chip-select line is once again pulled high to stop the data transfer.

2.10.4 I^{2}C

I^{2}C stands for ”inter-integrated circuit”. I^{2}C uses two wires for data transmission, Serial Data (SDA) and Serial Clock (SCL). SDA is where the data are sent and received meanwhile SCL carries the clock signal and are controlled by the master. The message sent over the SDA line follows a specific structure, see Figure 2.46. [10]

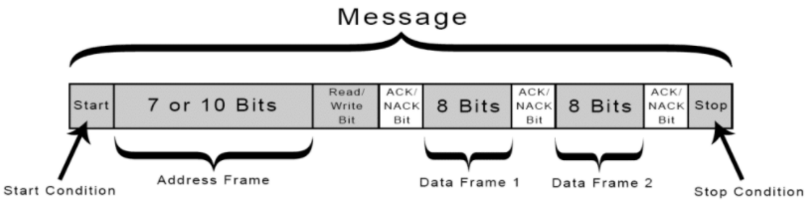


Figure 2.46. The structure of a message sent through I^{2}C.[10]

Chapter 3

Methodology

3.1 Team structure

To more efficiently research and develop specific aspects of the project, the team was divided into four different subsystems...**Drivetrain, Positioning and Control, Obstacle Detection and Path Planning, and Near-Field**. These were identified during an initial brainstorming session where pre-conceived knowledge and approaches were discussed (Figure 3.1).

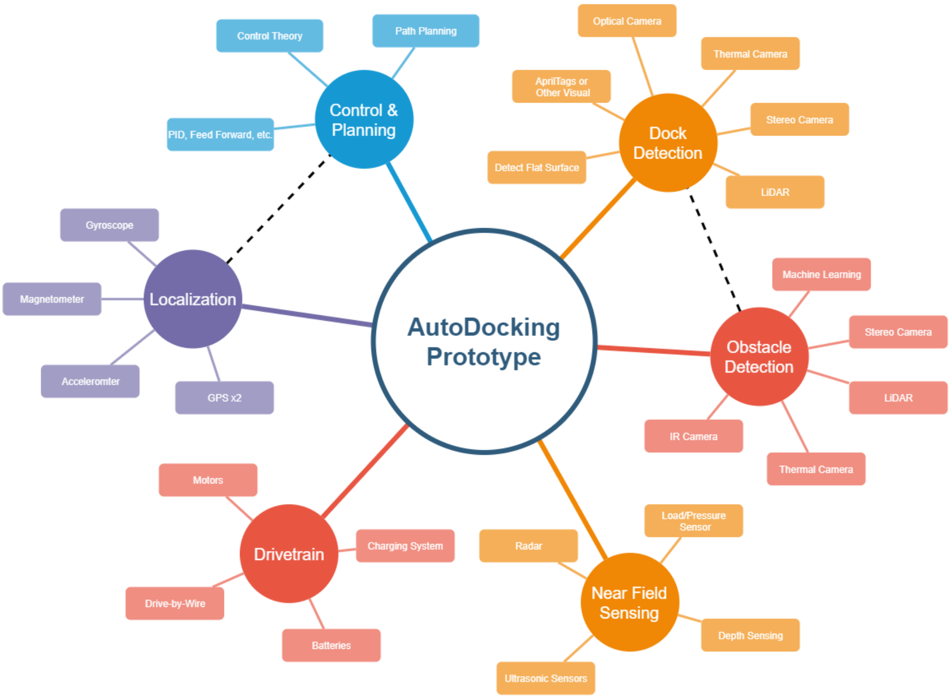


Figure 3.1. Mind map used to develop team structure.

These were divided based on system dependencies, functionality, and associated hardware. By doing this, subsystems could be researched, designed, fabricated, and tested by the same members prior to integrating into the entire system. Figure 3.1 shows how ”Control & Planning” were combined with ”Localization” to form the *Positioning and Control* subsystem and how ”Dock Detection” and ”Obstacle Detection” were combined to form the *Obstacle Detection and Path Planning* subsystem. Taking this approach was vital for the success of a project of this scale with several different functions working harmoniously together to achieve their shared system level requirements.

3.2 Project management

The initial planning and requirement setting was done in the *Polarion* [48]. Polarion is a browser-based platform for requirements, quality, and application lifecycle Management.

The main communication channel for the project was *Discord*, a communication platform designed for group communication with different channels for different topics [22]. This gave the group the ability to both talk to the whole team as well as in their specific subsystem.

The main means of storing and sharing general files was done on the file-sharing software *Google Drive* [39]. It provided a simple and free way of managing files like images, spreadsheets and presentations etc. For storing and managing code the group utilized the version handling software *GitHub.* It provided a solid platform for all the different software parts needed for the project [33].

For the higher level scheduling the project, *Monday.com* was used to create a Gantt chart [84]. This allowed the team to allocate time resources depending on the demands of each major task in the project. Monday’s interface provided an easy way to quickly adjust and monitor the overall project schedule in comparison to other means. The final Gantt charts used for project planning can be found in the Appendix.

Once the project progressed from earlier planning phases to actual fabrication and testing, a Kanban board was used to more quickly monitor and organize the progress of day-to-day activities for each subsystem. Again, Monday.com provided the web interface for this. Within each Kanban board (1 for main project, and others for each subsystem), individual tasks were first written on the ”backlog” and then transferred through ”working on it”, ”on hold”, ”stuck”, and ”done” labels depending on the status of the task.

3.3 Workflow

The workflow of the development process correlated to the structure of a V-model development cycle, see Figure 3.2. This model is a graphical representation of a developmental life-cycle that stresses verification/validation at key developmental milestones to mitigate risk, guarantee quality, and improve communication with stakeholders. There are two main streams to this model; the one on the left (downward sequence) is the specification/design stream and the one on the right is the testing stream (upward sequence). Connecting these two streams is the implementation phase where all the hardware and software is physically created.

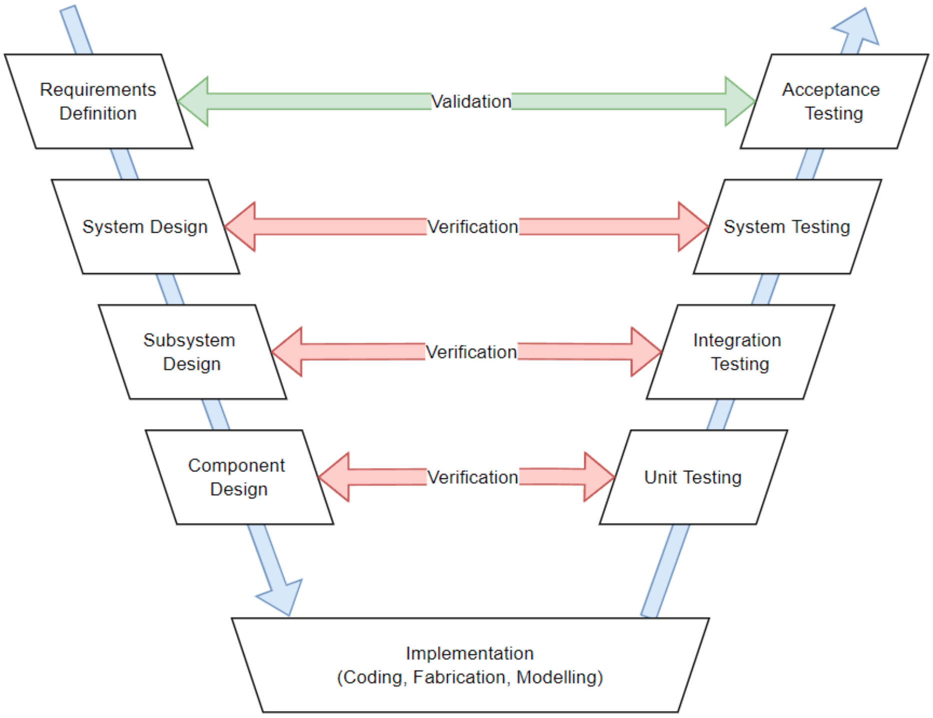


Figure 3.2. V-model describing the workflow

As seen in earlier chapters, the project began by identifying the requirements of the stakeholders and quantitative technical requirements the project should achieve. From here, the overall system concept was designed and the system architecture was established. The design then took a ”top-down” approach (correlating to the downward sequence of the left stream) where the subsystems were designed and the required components could be chosen/designed thereafter. Doing this allowed for the four different subsystems to be efficiently developed in parallel to one another while still being informed by specifics of the larger system architecture. Each subsystem was responsible for delegating mechanical, electrical, and software developmental tasks within their subsystems as per the needs for their functionality.

Once each subsystem had finished the design of their respective components, the hardware was acquired/fabricated, code was developed, and some initial assembly was done (implementation phase). From here, the development process worked up the right stream and each aspect of the project was systematically verified through testing. Each component was tested until working as intended (unit testing), then integrated into the subsystems for further testing with correlating hardware (or reworked and tested until standards were met), and finally the entire system could be tested with all subsystems together. The results from this testing were then compared to the requirements initially established to validate whether the prototype completed these sufficiently. By verifying/validating at each incremental stage of integration, issues could be better identified/amended quickly and everything could continue being developed in parallel until system testing was reached.

3.4 Budget and purchasing

The budget that stakeholders had provided for us was 50,000.00 kr (also noted in the requirements listed prior). With that being said, it was also mentioned that further funding could be negotiated if rightfully justified. All expenses, order status, and the project's BOM were captured in our *PO Tracker* Google Drive spreadsheet (see Appendix). This link was shared with stakeholders to provide them with real-time budgetary status.

Per stakeholder request, the costing for each line item was broken down further to capture shipping and VAT costs on this spreadsheet. Notes, order status, quantities, supplier info, and ordering person responsible was also added to provide further value information on purchases. Additional columns for shipping and invoicing specifics were also included, but hidden in the images found in the Appendix to save space.

Invoicing was completed through KTH's ITM Service Center when feasible. Restrictions on approved suppliers made purchasing certain items a challenge; this was especially true when ordering some of the more uncommon such as those of nautical nature. On the few occasions where ordering through the Service Center was not an option, student purchases were made and reimbursed by the stakeholder. The service center was also not communicative when it come to the costing breakdown on some of the order. With all this being said, some of the shipping and VAT costs were not fully captured by the spreadsheet, but mentioned in the correlating notes and communicated to stakeholder.

Chapter 4

Concept design

This chapter describes the concept design and evaluation. It also presents the final concept and its specifications.

4.1 Concept Platform

In order to more easily realize the concepts, a boat hull will be purchased. The boat, a Kimple 330 Angler Wide, is a small aluminium boat and can be seen in Figure 4.1. The boat is 3,3 m long, 1,37 m wide and has a weight of 56 kg [58]. This is small enough to be easily handled, but large enough to fit dual motors and the sensors required for autonomous driving. Its aluminum construction also makes it easy to modify and secure items to it.

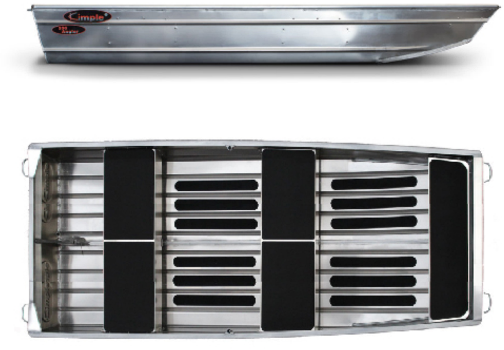


Figure 4.1. The proposed vessel to use as a platform [58]

4.2 Drive-by-Wire Design

Driving the vessel autonomously requires that the motors of the boat be controlled by a centralized computer. This is achieved by implementing some form of drive-by-wire control which utilizes an actuating system to control the motors. This entails a system that is able to control both the steering angle and the thrust of the motors. For portability to the P-30, this system is restricted to having both motors mounted at the stern of the boat with a maximum turning angle of ±30 degrees. From this, 3 concepts have been conceived.

Two out of said 3 concepts utilize some form of DC motor. It was decided that all concepts as such would utilize stepper motors specifically rather than servo motors. The reasoning for this is due to several factors. The motors will need to produce 3.5 Nm of torque at minimum (based on some rough drag calculations) and will preferably not exceed 24V. This is too keep the overall cost of the accumulator down; the higher the voltage, the more batteries or boost regulators will be needed.

The key differences between stepper and servo motors are discussed in Section 2.3.3. The main pros of a servo motor is that it can operate at high speeds, maintain torque at high speeds, and it has great positional accuracy at all speeds. This comes at a cost of higher voltage and amperage draw, as well as a much higher price though. For this application, the motors don't have to rotate any faster than $\sim$ 120 rpm to achieve a sufficient response time from -30 to 30 degrees (around 0.1 seconds). This means that a servo motor that can hold its torque for up to 4000 rpm is not needed since a stepper motor wont have significant torque drop until $\sim$ 300 rpm. To add, a $\geq$ 4 Nm servo motor at 24V is much harder to source than a comparable stepper motor. With all things considered, stepper motors are the better choice in this particular application.

In addition, it was established early on that all concepts conceived would have to incorporate some form of angle feedback via an encoder, potentiometer, or hall effect sensor. There is a need of accurate angle control in this application and there is a multitude of forces acting on the system due to it's maritime nature. Even though stepper motors are specifically used in open-loop cases where "steps" and holding torque are sufficient for maintaining position, these unpredictable forces may cause missed steps and inaccurate angle positions. Having some angle reading allows a control system to be built around that feedback and ensure accurate positioning at all times. An *absolute encoder* is also specifically preferred since it will retain positional readings even when powered off.

4.2.1 Direct Drive

The first concept, *direct drive* utilizes a hollow bore stepper motor that is attached directly to the shaft of the motor (Figure 4.2). This concept is beneficial for its ease of implementation and low cost. The hollow bore would also allow a handle to be attached to the end of the shaft which would make it possible to control the steering manually if needed. Components of the outboard motor could also be purchased separately (since the stock handled portion is no longer needed) and assembled, saving even more money.

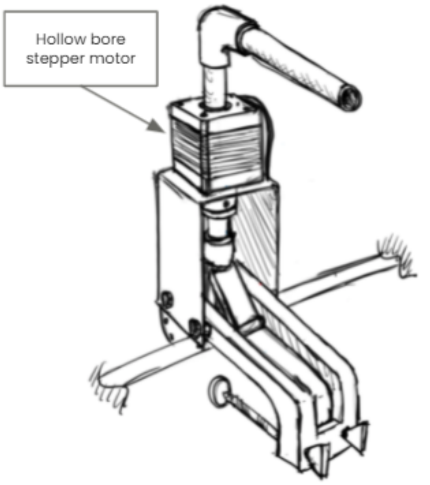


Figure 4.2. Direct drive concept sketch.

This approach may be limiting on the other hand. A direct configuration does not support the use of gears well (unless substantial time and money was spent designing a planetary system) and due diligence would then have to be done in selecting the motor. This is difficult though without knowing the rest of the prototype's parameters. In addition, hollow bore motors are far less common and have a limited parameter choices compared to normal stepper motors. This is especially true when considering that a ”closed-loop” stepper motor is desired (incorporated encoder with motor); these are even less available. A secondary sensor would have to be augmented onto the system most likely.

4.2.2 Geared Pulley

In the *geared pulley* approach, a standard closed-loop stepper motor is connected to an outboard motor shaft via a pulley system. The main advantage of using a system like this is in the ability to tailor the angular speed/acceleration of the outboard motor in relation to the the stepper. This is done by simply changing out the driven pulley diameter. Torque on the motor shaft can also be increased by simply placing a larger sized pulley on the motor shaft as well. Metal driver pulleys are readily available for standard stepper motor shafts, but the driven pulley on the shaft will most likely need to be fabricated by additive manufacturing created from a modified commercial item.

Tensioning of the belt will be done via the stepper motor mounting bracket that is secured on the boat hull clamp. The bracket can be moved back and secured with bolts, thus putting a load on the belt (see Figure 4.3). To prevent belt-slip, a toothed pulley will be used specifically. If testing shows this is still insufficient, an additional idler pulley can be explored as an option. This is unlikely given the expected speeds.

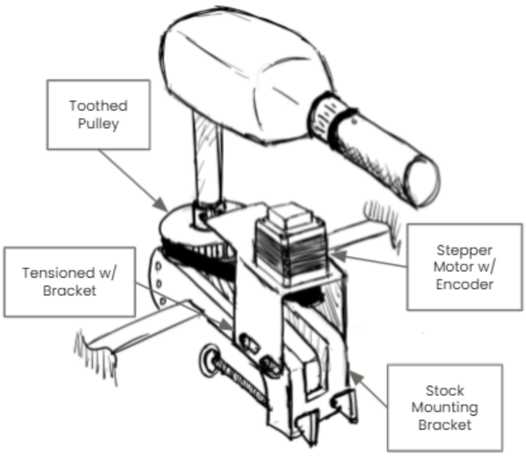


Figure 4.3. Geared pulley concept sketch.

The most prominent downside of this approach is the complexity. This concept has the most components of the conceived options and therefore will take somewhat longer to design and source hardware. This may also equate to cost as well, but it may also have little effect given that the stepper motor in this concept is generally cheaper.

4.2.3 Linear Actuator

This particular concept revolves around the use of a electric linear actuator. A linear actuator consists of some form of DC motor that is then connected to a lead screw directly or by the means of some gearing that then extends/retracts the actuator. This offers a large linear force that can then rotate the outboard motor when combined with a lever arm placed on the shaft (Figure 4.4. Beyond the large force, another advantage of this design is that these actuators are commonly found available with absolute encoders and that the system overall is easy to implement.

In this particular application, the idea is to design a bracket that can interface with outboard motor fixture with the intention of using additive manufacturing methodologies to create. See Figure 4.4 for a better idea on how this may look.

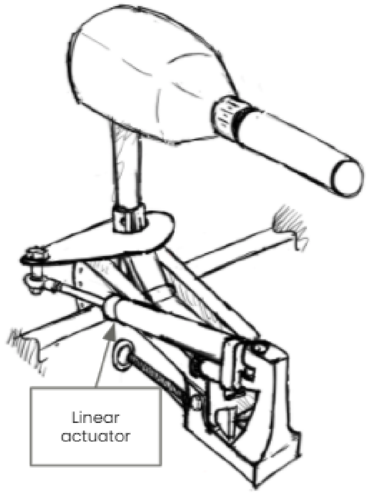


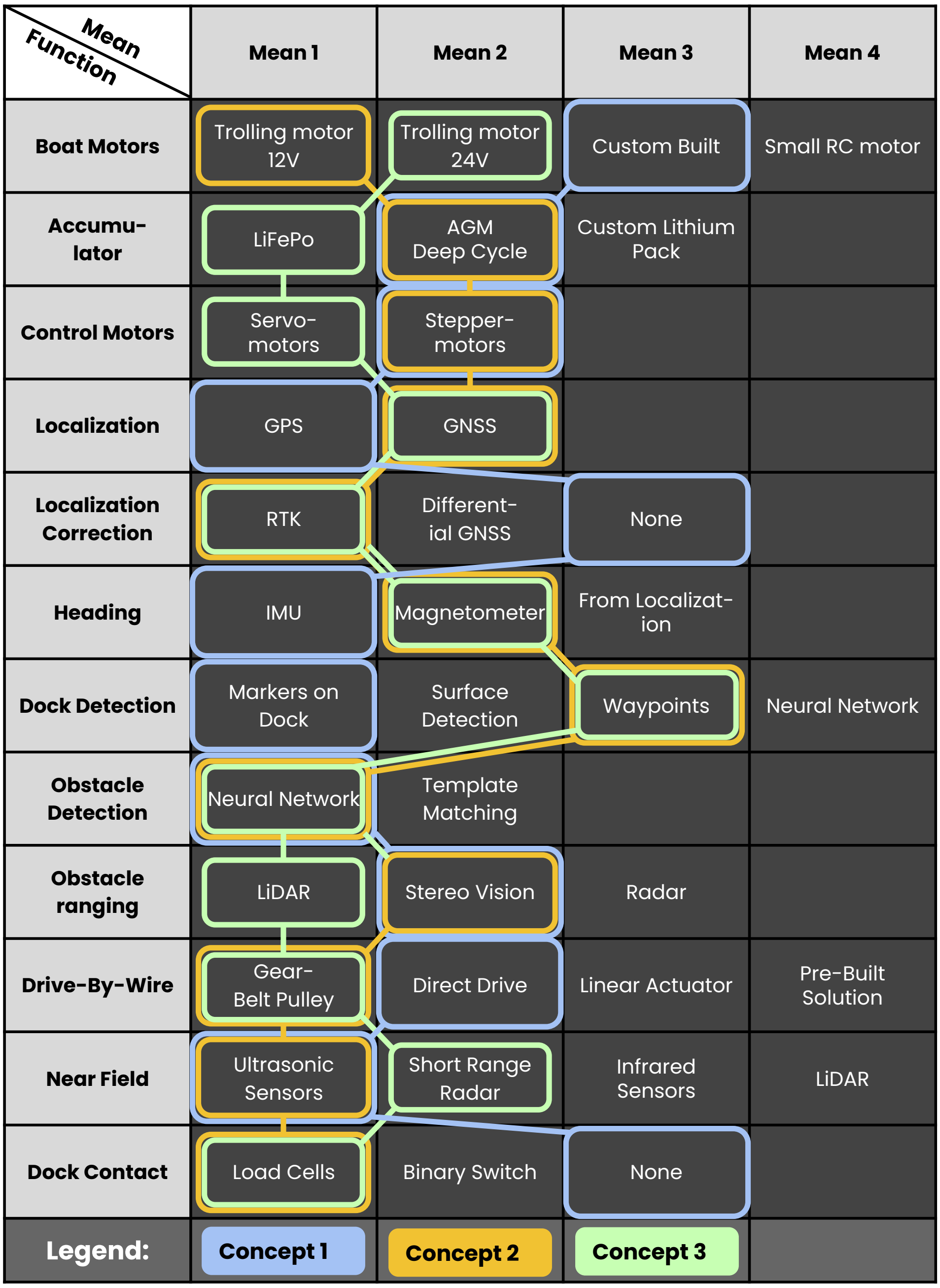
Figure 4.4. Linear actuator concept sketch.

Speed is the biggest draw back in this approach. Since the rotary motion of the DC motor is being translated into linear motion of the lead screw, speed is geared down. This may make such a concept too slow in responding to system demands. The footprint of this design is also much larger than the other concepts due to how it must be positioned. Another disadvantage is that torque applied to the shaft will not be consistent depending on the actuator angle in relation to the instantaneous angle of the outboard motor.

4.3 Morphological Matrix

A morphological matrix was made to illustrate the different options that were available to base the concepts on, see Table 4.1. It is built of a column of functions to the left, these are problems that has to be solved. The rows of the matrix represent the means, the possible solutions of these problems.

Table 4.1. Morphological matrix.



Mean Function	Mean 1	Mean 2	Mean 3	Mean 4
Boat Motors	Trolling motor 12V	Trolling motor 24V	Custom Built	Small RC motor
Accumulator	LiFePo	AGM Deep Cycle	Custom Lithium Pack	
Control Motors	Servo-motors	Stepper-motors		
Localization	GPS	GNSS		
Localization Correction	RTK	Differential GNSS	None	
Heading	IMU	Magnetometer	From Localization	
Dock Detection	Markers on Dock	Surface Detection	Waypoints	Neural Network
Obstacle Detection	Neural Network	Template Matching		
Obstacle ranging	LiDAR	Stereo Vision	Radar	
Drive-By-Wire	Gear-Belt Pulley	Direct Drive	Linear Actuator	Pre-Built Solution
Near Field	Ultrasonic Sensors	Short Range Radar	Infrared Sensors	LiDAR
Dock Contact	Load Cells	Binary Switch	None	
Legend:	**Concept 1**	**Concept 2**	**Concept 3**	

Three different concepts where deduced from the matrix each with their own distinct colour. The three concepts, *Concept 1, 2 and 3* will be discussed in detail in the following text.

4.4 Concept 1: Markers on Dock

This concept represents what the team calls a ”lean sensor approach” due to its relatively low cost. It is defined by having a less accurate GNSS system which introduces the need for another way of accurately determining the position of the dock. In this case, it is solved with AprilTag markers on the dock.

4.4.1 Drivetrain

The thrust motors in this concept would be custom designed for this particular application and incorporate a BLDC motor submerged in a sealed propeller enclosure (similar to the azimuth and trolling motor examples in Section 2.3 . This may be a good solution for saving money while still achieving a system with the desired performance. It would however take a considerable amount of time to design and assemble. Also, there are limitations in what is feasible to fabricate and there is a risk that the motors may get damaged by moisture if not designed and built properly. The batteries to drive these motors are deep cycle AGM batteries. The AGM batteries provide a cheap but heavy solution. The weight and cycling capabilities of these may effect overall system robustness and expandability poorly, but would most likely be negligible compared to other aspects.

This concept includes the ”direct” drive-by-wire solution that was discussed in more depth in Section 4.2.1. The direct drive layout provides a cheaper and easier to implement solution since there is less components required in comparison to the other layouts. With that being said, there is less versatility in gearing the system for more torque/speed as required. Because of this there is a risk that the system won’t be able to respond as fast as necessary for a safe and effective steering system. It is also harder to find hollow bore stepper motors (especially those that are waterproof and include encoders), so sourcing options are quite limited.

4.4.2 Positioning

The general idea of this concept in terms of positioning is to utilize the bare minimum when it comes to GNSS without any accuracy increasing correction techniques. Instead, more reliance is put into the readings from the IMU. Correction and accuracy is held up through data morphing and estimation using KF. This would reduce costs though would increase the workload in terms of implementation, validation and verification.

4.4.3 Obstacle Detection

The lower accuracy GNSS utilized in this concept makes it unfeasible to use GNSS to guide the boat all the way in to the dock. A new function needs to be added in order to position the boat in to the right spot, this is done with markers on the dock, more specifically AprilTags, see Figure 4.5. The AprilTags are recessed in to the docks rubber front to protect them from damage. The AprilTags with the right software give quite an accurate distance and angle to the dock guiding the boat in from ca 25 meters until the Near-Field sensors can take over at around 2-4 meters.

The obstacle detection is handled by a neural net running on an embedded computer. When an object is found, the positioning of that object relative to the boat is handled by a stereo vision camera. It has a range of approximately 20 meters which should be sufficient to be able to avoid a collision because of the vessels proposed speed and stopping ability. An IR stereo vision camera was also considered. The IR could add benefits for operating the vessel in dark conditions. According to the simplified scope, the vessel will be operating in well lit environments and have the possibility for artificial lighting, an IR stereo camera system was therefore disregarded.

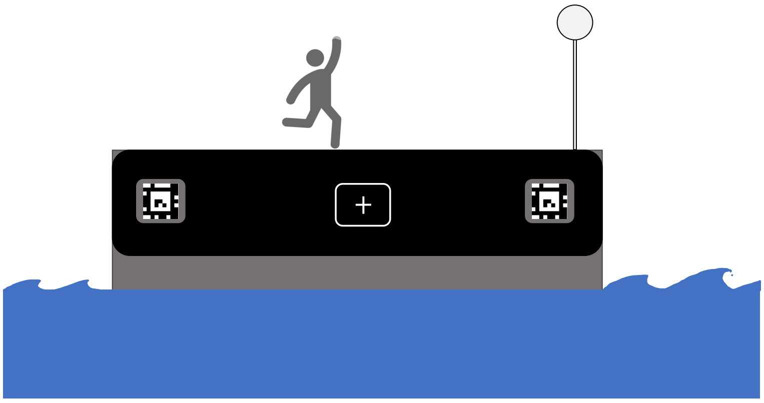


Figure 4.5. Dock with AprilTag modification.

4.4.4 Near-Field

For this concept, two ultrasonic sensors at the front of the boat is used for the near-dock approach as they provide good accuracy and robustness at a low cost. This configuration can be seen in Figure 4.6. The change in distance during the dock approach could be used to control the motors to ensure a low impact speed. The distance difference between the two sensors could together with trigonometrical calculations be used to decide the heading angle to the dock, but scattering of ultrasonic waves may be a problem. As for ensuring contact to the dock, the thought is to utilize the same ultrasonic sensors by assuming contact when the distance to the dock reaches zero. To keep it positioned to the dock, the motors could be controlled if the distance starts to deviate from zero. Additional ultrasonic sensors could be placed around the boat hull for basic hazard detection. To work in different ambient conditions, IP67-certified sensors with additional cover for debris could be used.

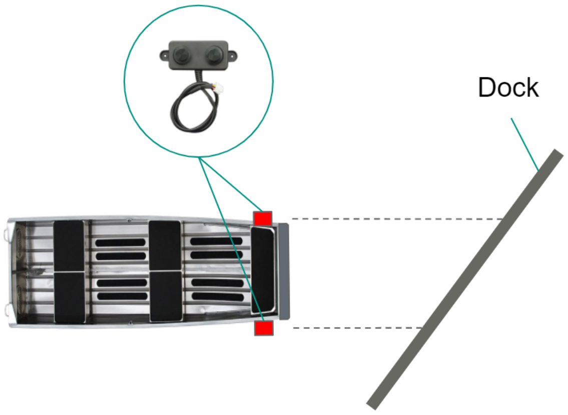


Figure 4.6. Ultrasonic sensor configuration for the near-dock approach.

4.5 Concept 2: Waypoint with Stereo Camera

This concept is defined by its use of accurate GNSS-positioning eliminating the need for markers on the dock. It uses a stereo vision system for object detection and avoidance.

4.5.1 Drivetrain

Two commercial 12V trolling motors that each produce approximately 220 Newton of thrust are used in this concept. Although this may not be the best performing option, it strikes a good balance between cost, ease of implementation, and hazard avoidance robustness. It is hard to quantify exactly how well the vessel will maneuver this early in the design phase, but early conservative estimates show that these motors should provide sufficient power to meet the technical requirements and respond fast enough to potential hazards. The batteries in this concept are the same as *Concept 1* (AGM deep cycle). Again, not the best performing option, but they provide a balanced solution in regards to cost and performance.

The geared belt pulley is used in this concept's drive-by-wire system. This solution is discussed in detail in Section 4.2.2. It may be more complex than the direct drive system in *Concept 1*, but the performance, versatility, and expandibility are greatly improved. Having the ability to gear up/down the stepper motor allows us to tune the system performance based on testing data. Also, using standard stepper motors means more options are available for selection.

4.5.2 Positioning

This concept uses high accuracy GNSS together with the correction technology Network RTK from SWEPOS. SWEPOS provides a Machine-to-Machine (M2M) Sim

Card for internet connection, to enable usage of NTRIP, and have base stations located around Stockholm, as seen in Figure 4.7. This results in positioning with high accuracy, i.e. an error as little as 30 mm since we are close enough to their base stations, and would be increasing the reliability of the system. Due to this increase in positional accuracy, heading can be determined together with an magnetometer. Though this somewhat would increase the cost of the system, it increases the reliability and eases the implementation.

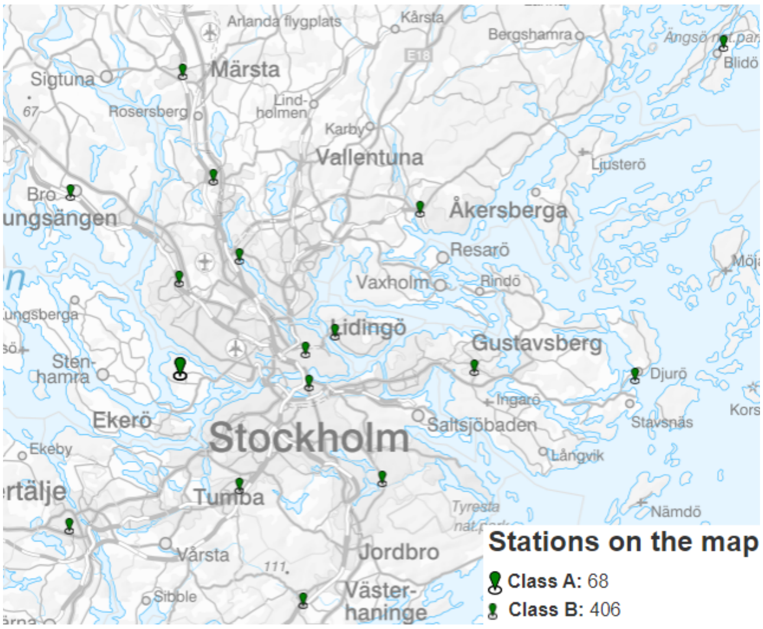


Figure 4.7. A map over SWEPOS Base Stations in Stockholm [60].

4.5.3 Obstacle Detection

Because of the increase in GNSS accuracy the need for an additional function to guide the boat in to the dock is not necessary. The boat will be able to be guided in to a predefined waypoint very accurately. The object detection will, just as in *Concept 1*, be handled with a neural network running on an embedded computer and the object localization will be achieved with a stereo camera system.

4.5.4 Near-Field

This concept uses the same configuration of the ultrasonic sensors as presented in Section 4.4.4 for the near-dock approach and control of boat impact speed. The difference lies in the dock contact where this concept would use two compression load cells beneath the front bumper to ensure that the boat is in contact with the dock, see configuration in Figure 4.8. When in contact, the force measurements would be used to control the motors to provide enough thrust to keep the boat in contact with the dock during the entire boarding/disembarkation process. Just as in *Concept 1*, additional ultrasonic sensors would be placed around the boat hull for basic hazard detection.

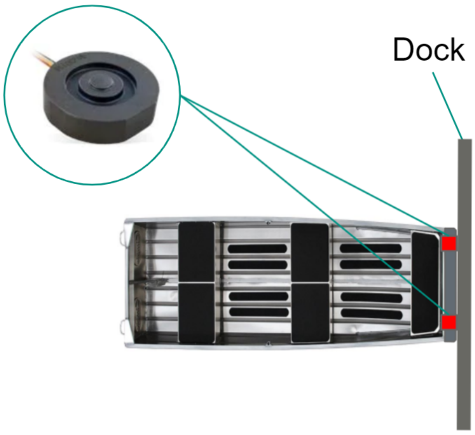


Figure 4.8. Load cell configuration for keeping dock contact.

4.6 Concept 3: Waypoint with LiDAR

This concept relies, just like *Concept 2*, on high accuracy GNSS to guide it in to the right position on the dock. Instead of a stereo vision system, it utilizes a LiDAR sensor to determine a detected objects position and the near-field ranging is handled by a radar system. The drivetrain consists of a high performance 24V configuration.

4.6.1 Drivetrain

The drivetrain in this concept has the same drive-by-wire system as *Concept 2*, but incorporates powerful 24V motors with approximately 350 Newtons of thrust each. The 24V motors provide a a better preforming system that can respond better to its environment, but is considerably more expensive. The motors themselves cost twice as much as their 12V counterparts and the system then requires twice as many batteries to meet the demands of these motors too. In addition, this concept utilizes LiFeP04 batteries which are better performing than AGM, but also cost nearly 4x as much. Although this is the more ideal system, it is most likely too expensive for our given budget.

4.6.2 Positioning

Concept 3 utilizes the same positioning system as *Concept 2*, see Section 4.5.2.

4.6.3 Obstacle Detection

Concept 3 is similar to *Concept 2* in that that it uses high precision GNSS that eliminates the need for other localization techniques like markers on the dock. It uses a single camera that is used for object detection with a neural network. For object localization the concept uses a LiDAR system.

There are several LiDAR options on the market. One high-end 3D LiDAR is the Robosense RS-LIDAR-16. Although not being the most expensive LiDAR, the RS-LIDAR-16 still costs about 35 000 kr [81] [98]. This is significantly out of budget for this project. On the cheaper end we have the Livox Mid-40. With the sacrifice of the FOV the Mid-40 becomes a lot cheaper at around 5000 kr [70]. While the accuracy and range of the LiDAR system is very appealing, it does require more work in implementation and testing; especially in the fusion of the LiDAR system and the vision system used for object detection.

Figure 4.9. Robosense RS-16 to the left and Livox Mid 40 to the right [98] [71].

4.6.4 Near-Field

The near-dock approach for this concept is solved with mmWave radar and this configuration can be seen in Figure 4.10. This option provides better environmental robustness as opposed to the ultrasonic sensors. It also provides three outputs of interest in forms of range, velocity and angle in one single reading. As it performs in a greater range than the ultrasonic sensors, additional radar units could be placed around the boat hull to deliver more advanced hazard detection. The radar option comes at a higher cost than the ultrasonic option and could be more challenging to implement correctly. This concept would utilize the same load cell configuration as presented in Section 4.5.4 to ensure dock contact during the boarding/disembarkation process.

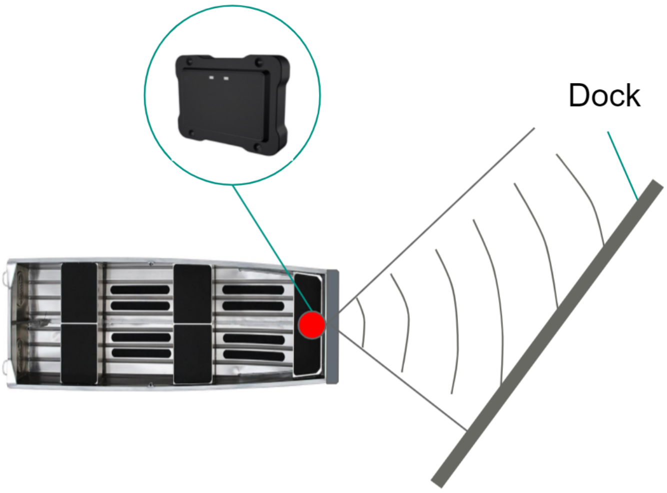
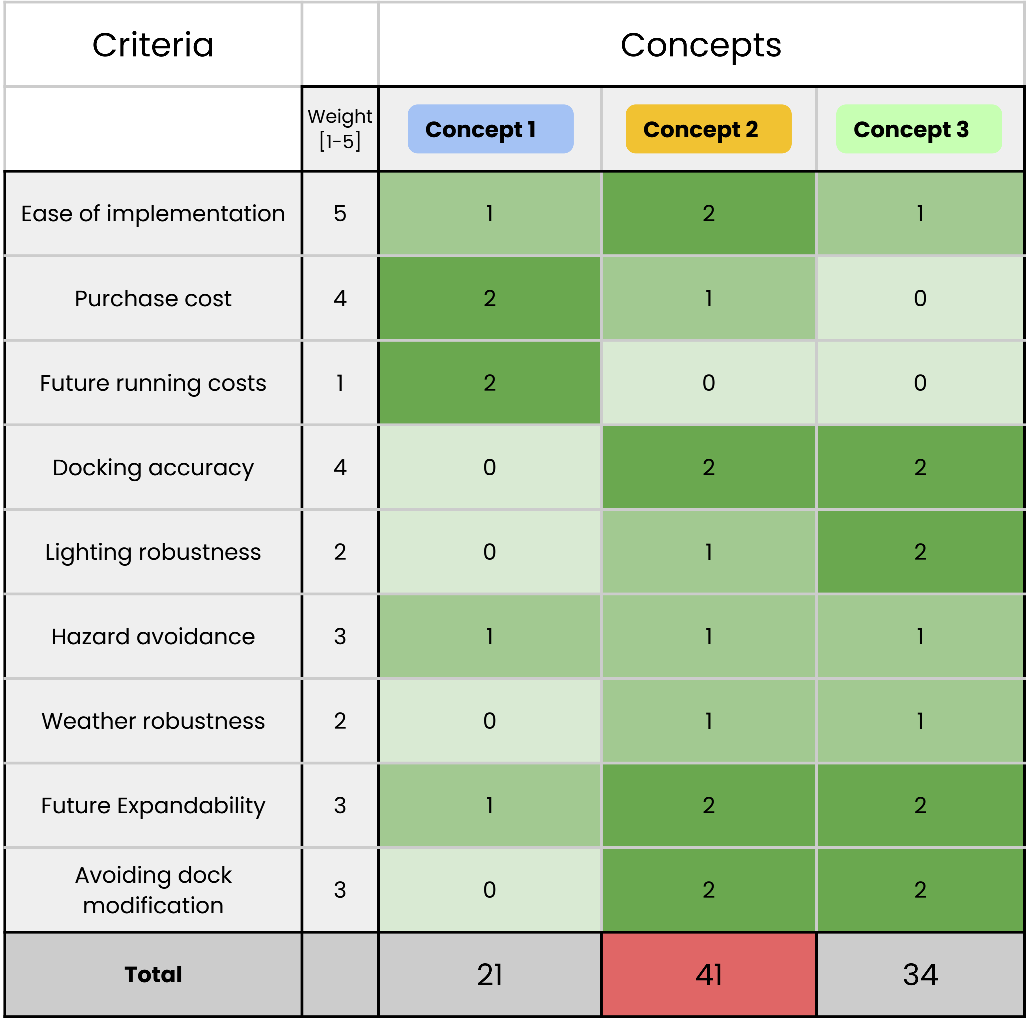


Figure 4.10. Radar configuration for the near-dock approach.

4.7 Evaluation

With all the pros and cons in mind for each concept generated, the group made a final evaluation with help of a weighted decision matrix, see Table 4.2. This gave the opportunity to go through all the important criteria of the system in a structured way. Points were given to the different concepts in a range of 0-2, with 0 being the worst an 2 the best. The evaluation criteria were weighted in terms of importance, with 5 being the highest and 1 the lowest. The points for each criteria were multiplied with the weight for a combined score in each category. The scores were then summed to a total score for each concept.

Table 4.2. Weighted decision matrix.



Criteria		Concepts		
	Weight [1-5]	Concept 1	Concept 2	Concept 3
Ease of implementation	5	1	2	1
Purchase cost	4	2	1	0
Future running costs	1	2	0	0
Docking accuracy	4	0	2	2
Lighting robustness	2	0	1	2
Hazard avoidance	3	1	1	1
Weather robustness	2	0	1	1
Future Expandability	3	1	2	2
Avoiding dock modification	3	0	2	2
Total		21	41	34

The winning concept turned out to be the one named ***Concept 2***. The concept represents a balanced approach, with good scores in the important categories of ease of implementation, cost and docking accuracy. The group agreed that *Concept 2* is the most promising concept and that it would be the concept to continue work on in this project.

4.8 Final Concept

The final concept is based on *Concept 2*, see the morphological matrix in Table 4.1 and the detailed description in Section 4.5 for full details. A sketch was developed to illustrate how the prototype will look with all components and strategies outlined in the concept description, see Figure 4.11.

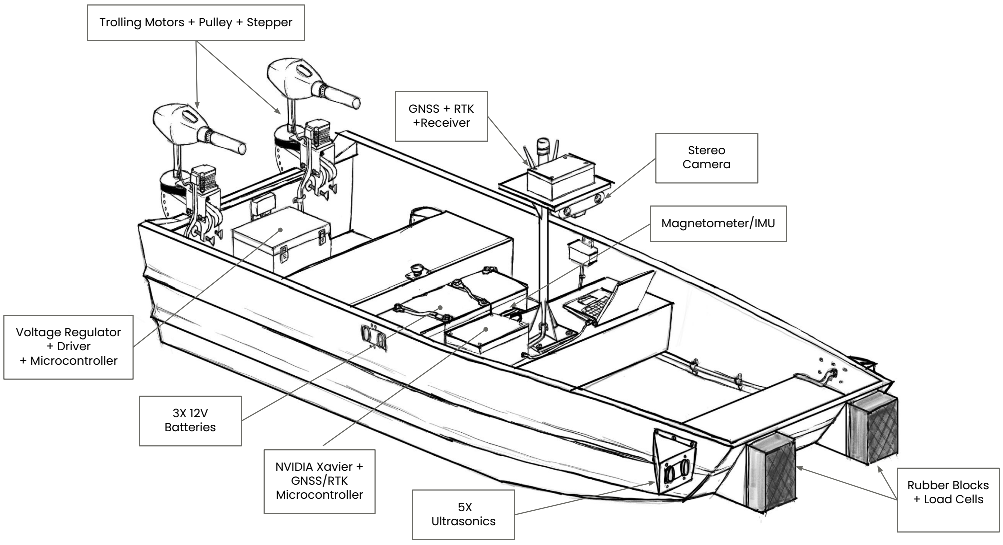

Trolling Motors + Pulley + Stepper
GNSS + RTK +Receiver
Stereo Camera
Magnetometer/IMU
Voltage Regulator + Driver + Microcontroller
3X 12V Batteries
NVIDIA Xavier + GNSS/RTK Microcontroller
5X Ultrasonics
Rubber Blocks + Load Cells

Figure 4.11. Sketch of the final concept.

The weight of the whole system is estimated at approximately 315 kg, with everything considered, and should reach a top speed of around 2.2 m/s. The main contributor to the weight is three AGM batteries. Three 100 Ah batteries are necessary to sufficiently power the two outboard motors for approximately 4 hours testing. Each subsystem does its own computations on separate micro controllers to preserve the ability of independent testing/development and improve diagnostics. Communications between the integrated systems are done via CAN-bus protocol to ensure portability with Candela's existing vessels.

The docking approach for the concept can be split in to four stages:

1. **Approach and avoidance**: At 50 meters from the docking position, an operator activates the AutoDocking protocol. The precision GNSS system coupled with path planning software guides the boat towards the dock. The obstacle detection system is active to avoid hazards or completely abort the approach if need be. It remains active throughout the docking sequence.

2. **Dock impact**: At approximately 3-4 meters, the ultrasonic sensors detects the dock and makes sure that the impact to the dock is at a safe and comfortable speed.

3. **Stationary at dock**: The force sensors makes sure that the boat is stationary as it it pressing up against the dock.

4. **Leaving the dock**: At a given signal, the boat backs out perpendicularly to the dock to a predefined distance. It then turns around and starts heading to its next destination.

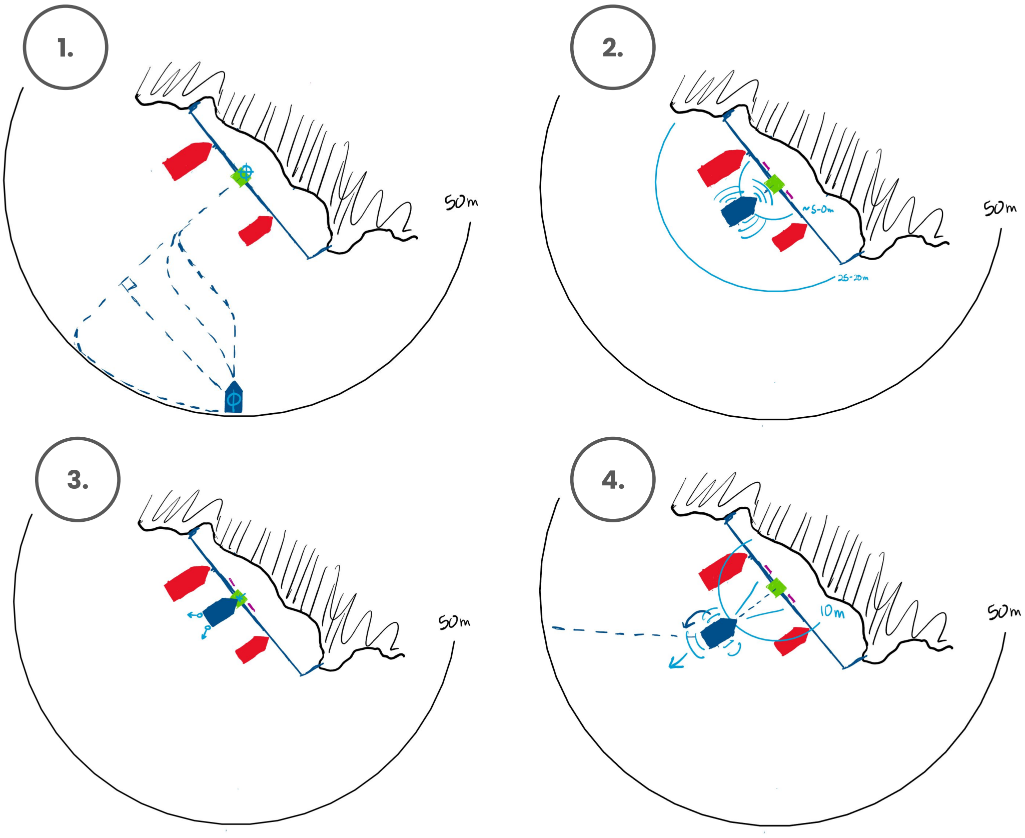


Figure 4.12. The four stages of the docking procedure.

Chapter 5

Implementation

This chapter gives a detailed description of how the team has implemented our hardware and integrated them to meet project requirements. As described in Chapter 3, the system has been divided into four major subsystems *Drivetrain*, *Positioning and Control*,*Obstacle detection and Path planning* and *Near-Field*. Each subsystem is tasked with a specific functionality and are interfaced with the rest of the system through a CAN-bus. On a holistic systems level, the prototype works in five steps...

1. The boat localizes it self with the help of an accurate GNSS-system with RTK-correction, the heading is read from a magnetometer.
2. A path from the boats current position and heading is calculated to the predefined dock position. The path is only valid if it does not pass over objects detected by the stereo camera.
3. The path is discretized into separate waypoints and a control system us used to follow said waypoints.
4. When the boats gets close to the dock ($< 3.5m$), the near-field control system takes over and regulates the approach to the dock, controlling both the speed and heading.
5. The boat then applies a constant force pushing against the dock to keep the boat stationary.

Throughout these five steps, the Drivetrain subsystem actuates on the reference control inputs generated from the other systems.

5.1 Final system prototype

In this section, an overview of the entire system prototype is introduced to provide a higher-level idea of how the final concept was realized. Details about specific areas of the system architecture are discussed in more detail later in this chapter.

5.1.1 System architecture

To better represent how the entire boat is structured, a system architecture diagram was made. This provided the team with a clearer vision of where certain data came from, the communication protocol being used,and each subsystem's respective role in the overall function of the prototype. The full detailed diagram can be found in the Appendix, but a more simplified version can be found below in Figure 5.1. As the project progressed, this diagram was updated to reflect any changes that had occurred. The detailed diagram also has further information on hardware specifics and further power distribution info.

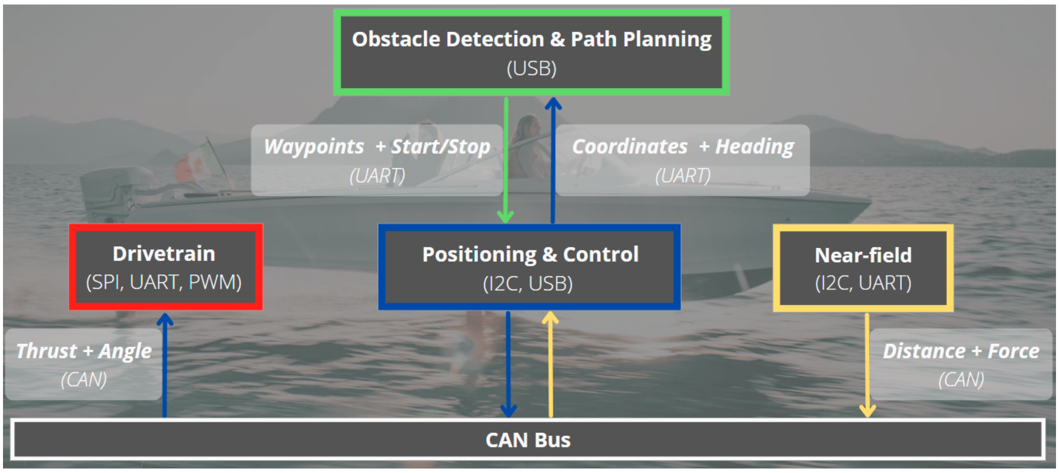


Figure 5.1. Simplified system architecture diagram.

The colored boxes denote the four major subsystems of the project. Within each box the utilized communication protocols are listed. The arrows show the flow of information being transferred to and from each subsystem.

5.1.2 Physical model

The finalized prototype can be seen below in Figure 5.2 with the CAD rendering on the left and the physical prototype boat on the right.

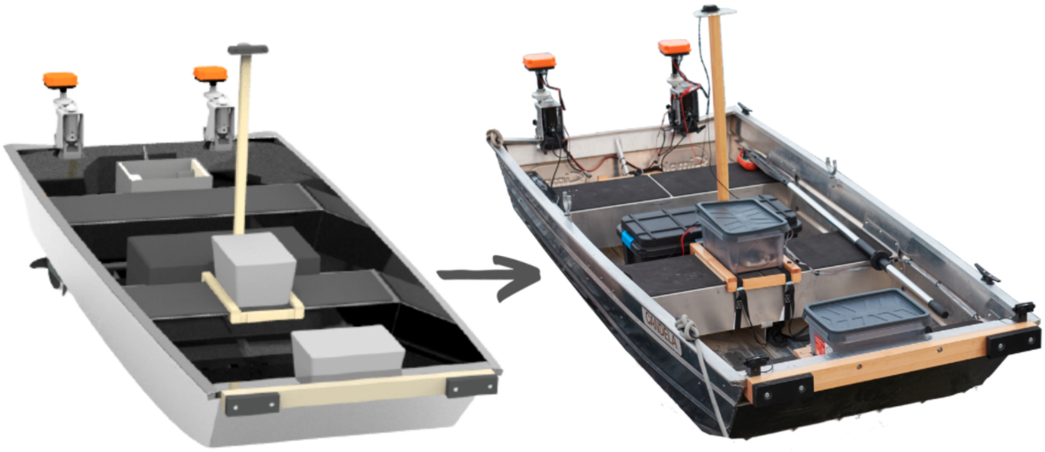


Figure 5.2. The final boat CAD rendering (left) and finished physical prototype (right).

The prototype ended up very similar to the final concept outlined in Figure 4.11. Some notable changes were however made during the development process. One of these is moving the ultrasonic sensors from the sides of the bow to the top of the bow. All of the subsystem electronics were packaged in water-resistant boxes and special care was taken to use connectors and cables that can withstand the outdoor climate (such as cable glands and rubber grommets).

After each day of testing, the boat had to by physically pulled out of the water by team members and stored. Therefore, the batteries and electronic enclosures had to be designed in such a way that they could be modular and easily removed to reduce weight of the boat before lifting out of the water. To facilitate this, all cables leading into the enclosures or attached to the motor assemblies were augmented with waterproof connectors for simplified disassembly.

To test the prototype in the water a dock surface had to be created. The goal of the surface was to provide similar conditions to what the SL-ferries encounter today but at a smaller scale. The finished dock surface can be seen in Figure 5.3.

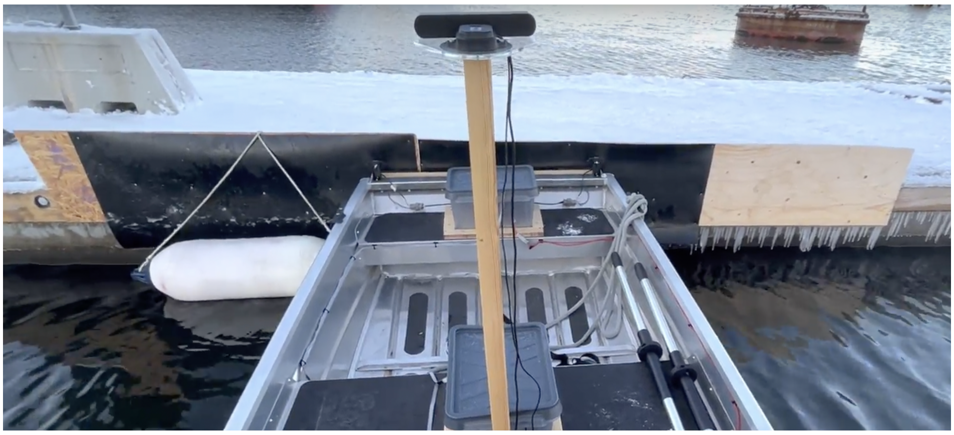


Figure 5.3. The constructed dock covered in a rubber sheet.

5.2 Software development tools

This section discusses the various software tools that were used to aid in the development of the project.

5.2.1 STM32CubeMX

STM32CubeMX is a graphical tool developed by STMicroelectronics to aid in the setup of their STM32 microcontrollers [112]. The tool provides a more intuitive way of setting up a new controller, configuring clock speed, I/O ports, threads and more. It also includes some more advanced features such as a power-consumption calculator.

5.2.2 STM32CubeIDE

STM32CubeIDE is a development platform for STMicroelectronic's STM32 family of 32-bit ARM Cortex-M core-based microcontrollers [111]. This platform supports several programming languages such as C, C++, Pascal and JAVA. It also supports a number of different debugger tools to be able to monitor the microcontrollers during run-time.

5.2.3 FreeRTOS

FreeRTOS is a real time operating system. Real time operating systems are mostly used for embedded purposes where consistent and reliable multi-tasking of functions are required. FreeRTOS is small but provides key multi-tasking functionalities such as threading, semaphores, timers and other advanced kernel scheduling features [31].

5.2.4 MATLAB

MATLAB is a software and programming language developed by The MathWorks, Inc. [80]. The program is specifically made for testing and development purposes and is used mainly in engineering and economics applications. The language is interpreting i.e not compiled to an executable file.

5.2.5 Simulink

Simulink is a program for modeling, simulating and analysing dynamic systems. It is a part of the MATLAB ecosystem and is also developed by MathWorks [108]. Simulink has a wide selection of libraries to aid in the modeling of complex systems. One example applicable to this project is the Marine Systems Simulator toolbox extension that is designed for simulation of marine vessels.

5.2.6 Onshape

To design the mechanical components of the boat, an online Computer Aided Design (CAD) program called *Onshape* was used [14]. By being online, CAD files could easily be shared and collaborated on by all members and edited on any computer without additional downloads. This software lacks some of the rendering and finite element analysis features of more expensive softwares, but it still suits the needs of this particular project's modelling requirements.

5.3 CAN-Bus Implementation

The physical implementation of the CAN bus was done with the MCP2562 High-Speed CAN Transceiver by Microchip Technology Inc. [49]. The transceiver can handle a bit rate up to 1 Mb/s. To be able to interface with the chip a small PCB was designed following recommendations in the data-sheet, see Figure 5.4.

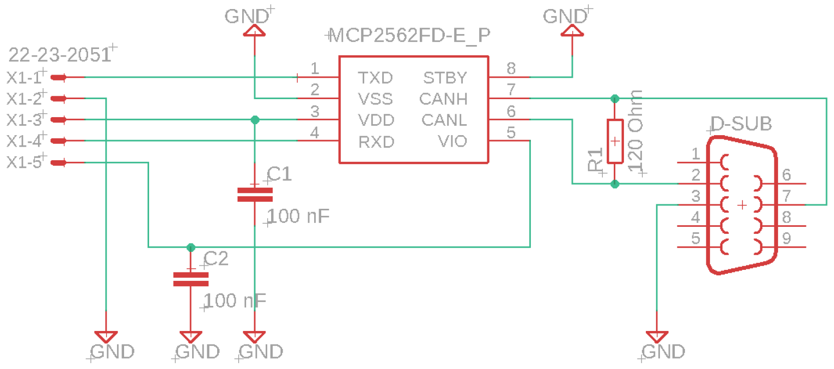


Figure 5.4. Schematic of the CAN-transceiver and its interfaces, created in Autodesk EAGLE.

A D-Sub 9-pin contact was used to connect a ribbon cable that carries the CAN-messages the long distances in the boat. The resistor R1 is a terminating resistor, it is only mounted on the CAN-boards at the end points of the ribbon cable. The purpose of the terminating resistors are to ensure that the signals are not reflected back from ends of the cables, interfering with new signals. From this schematic PCB:s where milled and components mounted see Figure 5.5.

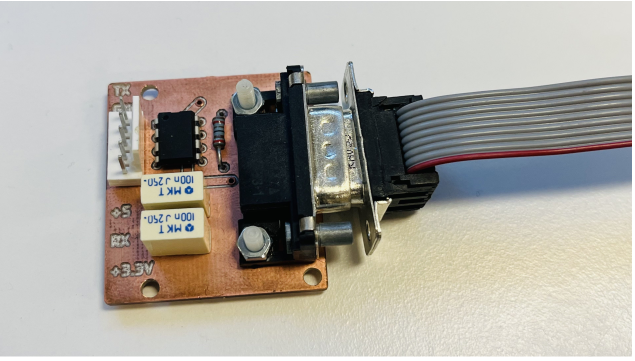


Figure 5.5. The finished CAN-transceiver board with ribbon cable attached.

The CAN-protocol was implemented on the micro controllers and connected to the transceiver PCB through the white Molex connector. The bus was configured with a 500 Kb/s bit rate.

5.4 Drivetrain

The Drivetrain subsystem is responsible for actuating the trolling motors (based on reference angle and thrust values received from the other subsystems),the accumulator and power distribution to the other subsystems. The following is a description of the hardware and methodologies used to complete these functions.

5.4.1 General approach

A full view of the Drivetrain subsystem can be seen below in Figure 5.6. The boat hull platform used for the prototype is the Kimple Angler 330 wide. It was chosen due to having a flat front (which is needed for frontal docking similar to how current SL ferries dock), being lightweight and easy to modify with its aluminum construction. This was the cheapest option available with the required weight capacity of 350 kg.

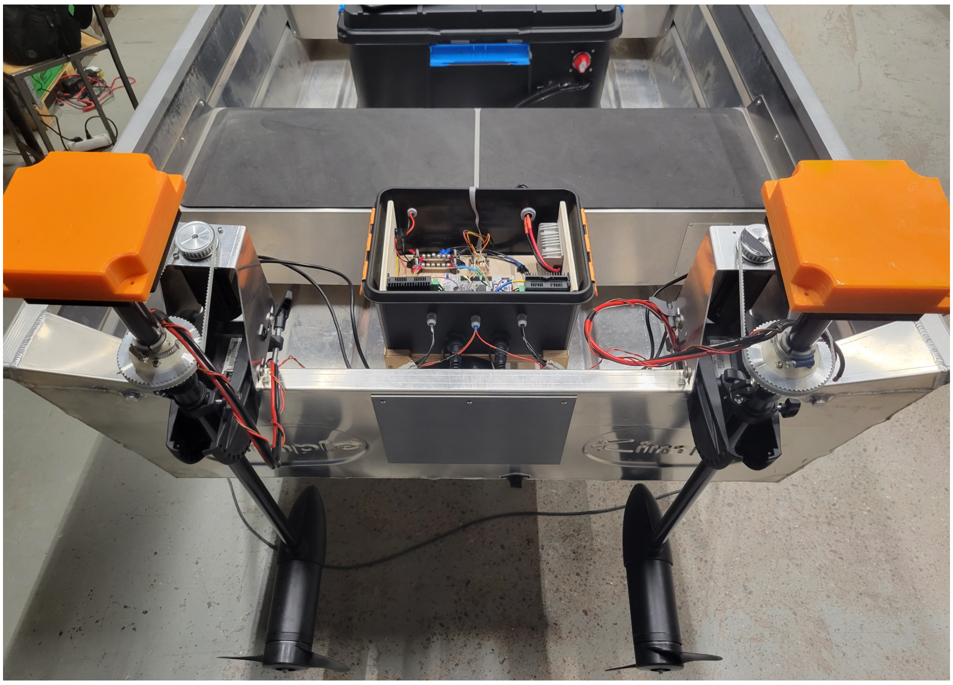


Figure 5.6. Full Drivetrain subsystem (utilizing initial revision of stepper motor bracket).

All peripherals of this subsystem are controlled and fed power through a central electronics box. As with the other enclosures, additional measures for water resistance were made via the enclosure itself and the cabling feedthroughs/connectors. The enclosure is positioned in the back middle of the boat to minimize cable length as well as keeping the the overall system layout symmetrical in attempts toward keeping the center of mass inline with boat.

The two trolling motors are mounted on the back of the boat using the stock motor mounts and placed as far apart as possible to promote maneuverability. As per the final concept, steppers are used to rotate the external motors; these are attached to the stock motor mount using an aluminum sheet metal bracket and some spacers (see Figure 5.7). The stepper rotor is mechanically connected to the trolling motor shaft using a toothed pulley system with a 2:1 gear ratio that increases torque to rotate the shaft. The bracket is slotted where interfacing with the motor mount so as to tension the built when assembling. On the stepper bracket itself are additional belt tensioners to further ensure no belt slip. To hold the pulleys position on the shaft are custom 3D printed clamps which tighten down with hose clamps. The orange 3D printed enclosure on the top houses the Electronic Speed Controller (ESC).

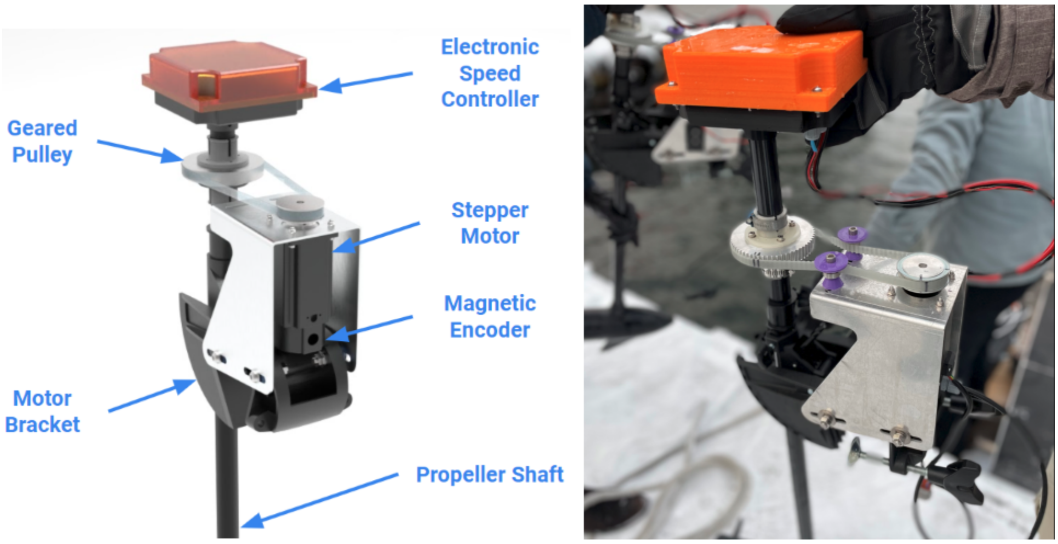


Figure 5.7. Full motor assembly photo including belt tensioner.

5.4.2 Trolling motor

One of the first major components to be determined for this subsystem was the trolling motors. Since many of the other hardware decisions were dependent on the parameters of this item, it was important that this decision was made first.

As mentioned in Section 4.5, we were restricted to using a 12V power source for use with the trolling motor. It was desirable to have as much thrust available despite this limit to ensure controller stability with autonomous . It was ultimately decided then to choose a pair of *Minn Kota Endura Max 55* motors for use in this application [26]. This motor utilizes a brushed DC motor that is advertised at producing 55 lbf of thrust (244.65 N), at 50 A, and at a price point that was acceptable to our budget. This particular model is built to withstand fresh water conditions. It was originally considered to use salt water capable motors, but given that the waters in the Stockholm archipelago are quite brackish (salinity of .65%), freshwater versions were deemed acceptable. Future testing proved this assumption to hold true.

Electronic speed controller

The original plan for controlling the propeller speed of these motors was to utilize the stock controller housed in the head of the motor. Although details about this were not known before purchase, it was assumed that the motor used an analog voltage that is regulated by a potentiometer connected to the handle (see Figure 5.8). If this were true, then the potentiometer could be bypassed by an analog signal of our own choosing.

Upon initial inspection of the part, this was proven to be true, but it also consisted of several relays (presumably for direction changes) and covered in a thick layer of potting compound (see the left image in Figure 5.8). This created multiple issues because it meant that the potentiometer could not be accessed and the relays meant the motor was susceptible to contact ”bounce” when switching directions (which would happen often when controlling velocity near the dock). For that reason, it was decided that a aftermarket ESC would be needed.

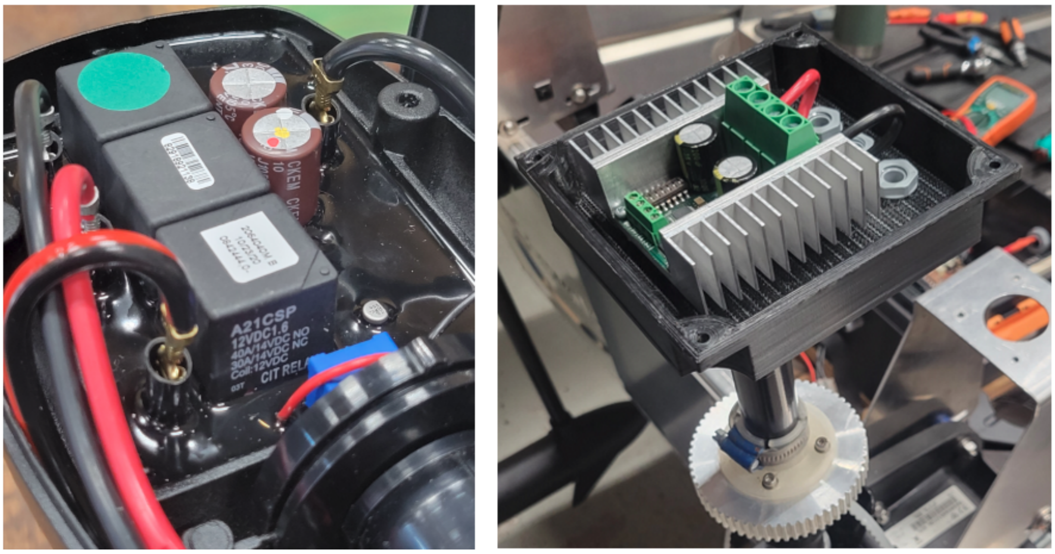


Figure 5.8. Original trolling motor ESC (left) vs. new aftermarket ESC (right).

Due to the high current draw of the trolling motor and the brushed DC motor design, options were limited. The *SyRen 50* was eventually chosen [114]. Unlike the stock relays, this utilized a high frequency h-bridge design to supply the motor with the required current. The SyRen 50 ESC is rated to 50A continuous, 6-30V output, thermal/over-current protection, and could utilize several different input signals such as analog, Pulse-Width Modulation (PWM), and serial. Several other advanced features, but not utilized in this project.

5.4.3 Stepper motor

To choose the exact stepper motor for use, an estimated dynamic equation for rotating the motor in the water was derived. This was in turn used to derive a trajectory planner that was implemented in MATLAB. Various motor parameters found on the market were then applied to this equation and the cheapest motor that offered the required performance was chosen. The derived dynamic equation can be seen in Equation 5.1 and 5.2 below.

$$J_{tot} = J_{stp} + \frac{J_{motor}}{n^2} \tag{5.1}$$

$$\dot{\omega} = \frac{K_m V}{R J_{tot}} - (\frac{K_m^2}{R J_{tot}} + \frac{d}{J_{tot} n^2})\omega \tag{5.2}$$

Table 5.1 describes the various parameters being used in said equations above. The values that read ”TBD” denote the parameters that need to be plugged in for the stepper motor in question to test for performance.

Parameter	Value	Description
J_{stp}	TBD [kgm^2]	Stepper motor inertia
J_{motor}	.188 kgm^2	Estimated motor inertia
J_{tot}	TBD [kgm^2]	Total combined inertia
n	2	Gear ratio
w	TBD [rad/sec]	Angular velocity
V	24 [V]	Voltage
R	TBD [Ω]	Terminal resistance Ω
K_m	TBD [Nm/A]	Torque constant
d	0.304 Nm/(rad/s)	Dampening coefficient from drag (linear)

Table 5.1. Parameters for simplified dynamic equation

To get an initial idea of what stepper motors to look for, an estimate of the required holding torque was needed. It was assumed that the max torque would be induced from the drag of the water at max angular velocity. The geometry of the propeller was conservatively simplified to a flat plane and horizontal cylinder (see Figure 5.9). It was then suggested that the fastest the propeller should ever need to turn is 2 rev/sec. This rotational velocity was translated to linear velocity and the total drag forces were then calculated by integrating across the surface of the propeller (assuming it was acting linearly along the surface). Torque was then calculated by taking the total drag force acting on both sides and applying it 2/3 from the shaft centerline on either side of the shaft. Results showed a max torque of **3.818 Nm**. This torque value was also then used to determine the damping coefficient in the dynamic equation.

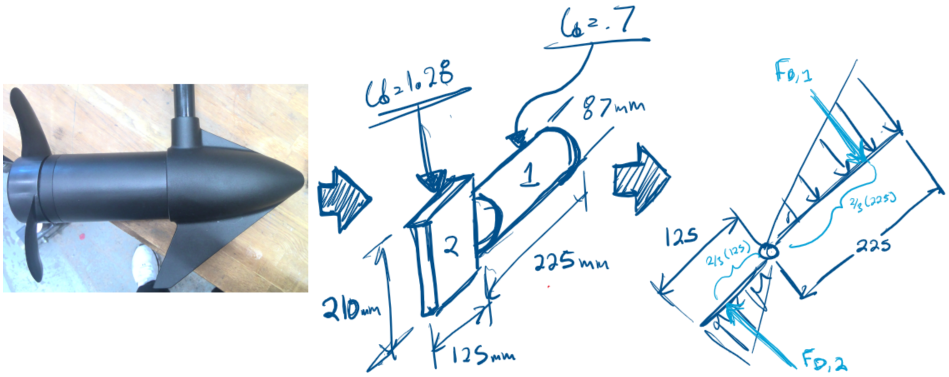


Figure 5.9. Simplification of trolling motor geometry for drag calculations.

With the required torque loosely known, various stepper motors could be browsed for and tested in our MATLAB script. In the end, the stepper motor chosen was of a **Bipolar, Nema 24 (dual shaft)** form factor with **4 Nm** of torque and a max current of **4.2 A**. The stepper is rated for a voltage between 24-48V, and Figure 5.10 shows the torque curve with the stepper operating at 36 and 48V. However, since the maximum speed needed for a full rotation is never above 200 rpm, the torque degradation due to max voltage is negligible. It was thefore deemed acceptable to run the motors at 24V. This decision was further motivated by difficulties in finding a boost converter from the systems 12 volt main battery that could supply more than 20A. Any higher voltage would require a higher current draw than was found available.

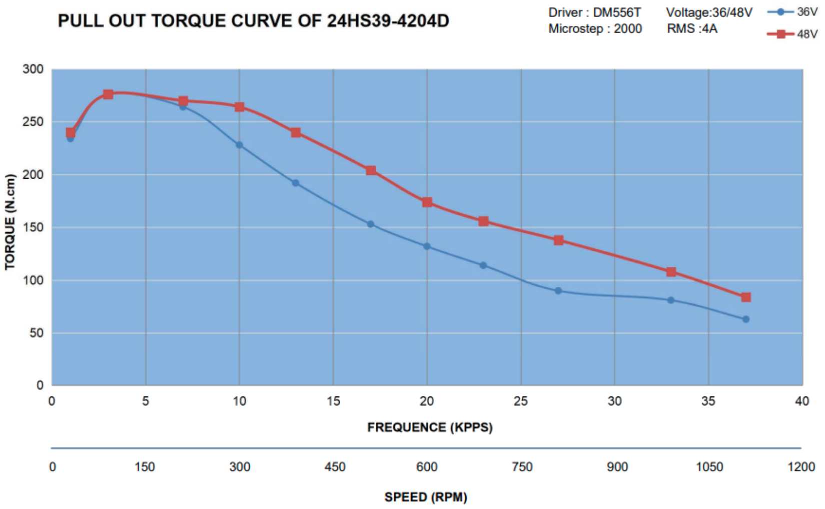


Figure 5.10. Stepper motor torque curve.

The reference trajectory for this particularly chosen motor can be seen in Figure 5.11. This final trajectory was generated after several attempts with differing max angular velocity inputs (acceleration is dependent on the max velocity per the dynamic equation). Ultimately, $\omega = 16\,\text{rad/sec}$ gave us the fastest overall actuating time for the full range of movement... **0.18 seconds**. The vertical red line in the plot correlates to the original goal actuating time with a constant angular velocity of 2 rev/sec (as used in torque calculations). Clearly this trajectory is slower than the original goal, but 0.18 seconds was decidedly fast enough to ensure stability of the vessel.

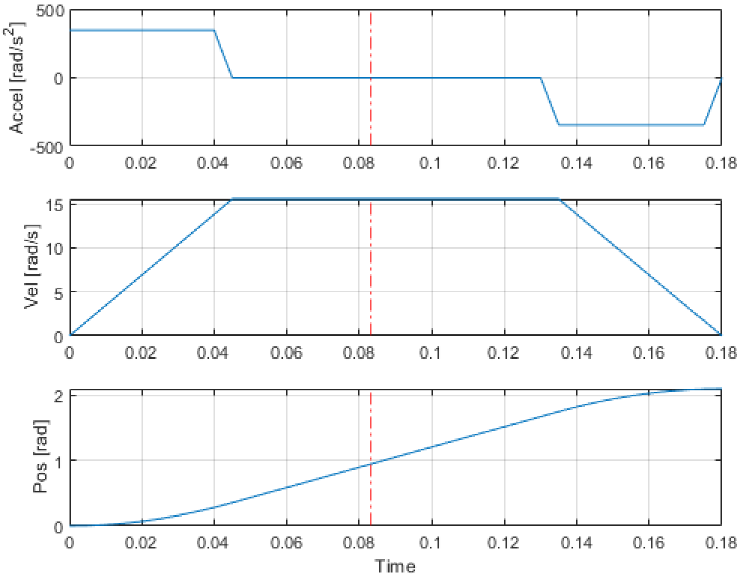


Figure 5.11. Reference trajectory of chosen stepper motor.

Given that these calculations were made before any hardware was physically acquired, many assumptions and numerical estimations were used. Regardless, this analysis was just being used to choose the appropriate stepper motor, so they were considered sufficient.

Stepper driver

Choosing a suitable driver for the stepper was a much simpler task than the stepper itself. The two main factors needing consideration for the driver was the current rating and the capable pulse/rev input frequency supplied. Given that it was originally desired to have a angular velocity of 2 rev/sec, this conservatively equates to a required 1480 Hz input frequency being generated by the stepper driver with 800 steps/rev (1.8°step angle and a micro-stepping of 4). Most drivers can supply well beyond this input frequency, so it was a non-issue.

The *DM556T* stepper driver sold by Stepper Online was eventually decided on [21]. This driver was well within budget, could supply a pulse input frequency up to 200 kHz, and operate at 24V/5.6A (max). In addition, this is the same driver cited as being used in the torque curve of Figure 5.10 as well, so compatibility was further verified.

Absolute Encoder

As mentioned earlier in the concept generation, an absolute encoder would be required so as to retain the known position after turning off the power. Absolute encoders tend to be expensive, so a generic purpose rotary magnetic encoder (*AMS5048A adapter board*) was used instead [6]. This particular version uses SPI communication and has a 14-bit resolution which allowed for faster and more accurate reading of the encoder than the I^2C counterpart.

Since the stepper motor chosen was of a dual shaft form factor, the provided rotary magnet was easily epoxied on the exposed rear shaft. A 3D printed bracket was then fabricated that mounted the board centered to the magnet and $\approx$ 1mm away as per the manual specifications. A housing was also then fabricated and mounted around this small assembly to keep it to protect it from the wet testing environment.

SPI is intended for short distance communication, so there were some concerns about EMI being an issue; especially given that the stepper motor phase wires were nearby. To mitigate any potential noise issues, FTP shielded Cat6a ethernet cables were used between the encoder and drivetrain electronics enclosure. Waterproof glands and Ethernet connectors were used to ensure further robustness against the weather while still allowing quick disassembly of all components.

5.4.4 Accumulator and Power Distribution

Since the highest current item (the trolling motors at 50A/ea.) were rated at 12V, the AGM battery setup was chosen accordingly. When deciding how many batteries were needed, a rough calculation was made based on how much current was going to be used. With trolling motors, steppers, and misc. electronics hardware consuming 55.5 Ah at a 37% duty cycle, that meant needing at least 2 batteries then to get $\approx$ 4 hours of testing time in. There were also some concerns on the steep discharge curve of AGM batteries and the effects the predicted cold weather would have on the system. Therefore, it was ultimately decided to purchase 3 ea. AGM batteries rated at 108 Ah (324 Ah total). These were then wired in parallel and contained in a waterproof enclosure, see Figure 5.12.

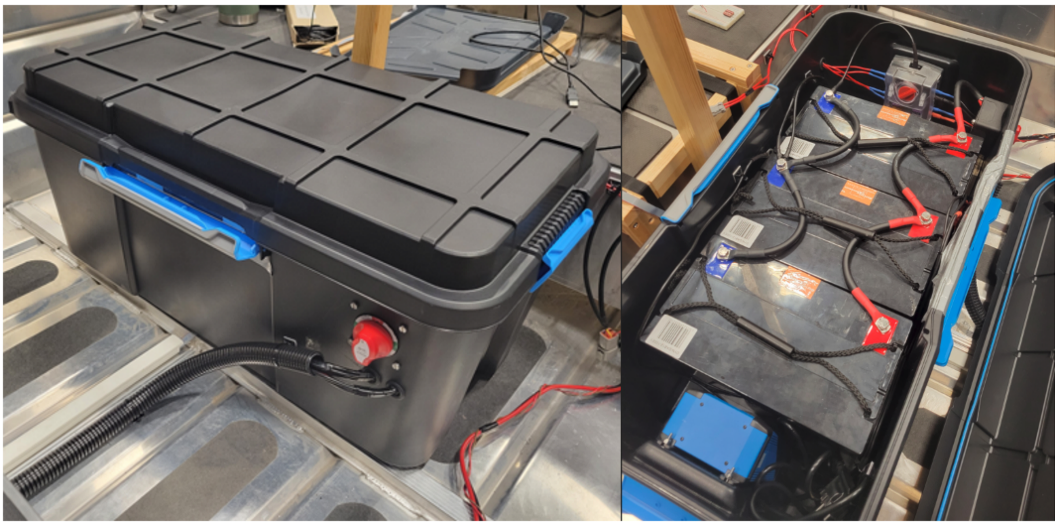


Figure 5.12. Accumulator enclosure including AGM batteries, power switch, fuse box, and charger.

The enclosure was positioned strategically behind the second row of seats in the boat to locate the center of mass toward the physical center of the boat as much as possible. Initial simulations showed this increased stability of the autonomous control. From this enclosure, 12V was supplied to all other subsystem enclosures and a fuse box was also placed inline to protect the other hardware from any possible current spikes. At each subsystem enclosure, additional buck/boost converters were placed inside as per the needs of that subsystem. A 3 battery balance charger was also placed in the enclosure to re-charge the batteries after each testing session while ensuring there longevity.

A 3-position high-current rated switch was also added to the exterior of the box. This switch allowed for Drivetrain subsystem to powered off without resetting the rest of the system if needed. This conserved energy and allowed for quick shut-downs in the case of emergencies as well.

5.4.5 Electronics

The following is a quick overview of the other key electric components used in the Drivetrain subsystem. Most of the electronics are housed in a enclosure placed near the high current components at the rear of the boat. See Figure 5.13 below to see the various electrical components contained in here.

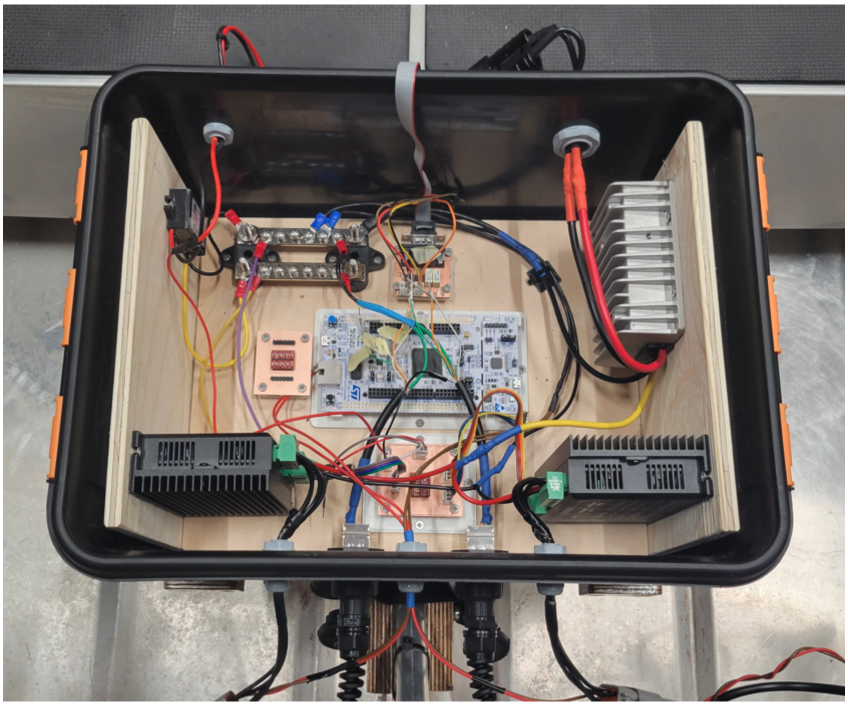


Figure 5.13. Drivetrain enclosure housing development board, buck/boost converters, stepper drivers, logic converters, and CAN transceiver.

Development Board

In the beginning of the project, it was recommended by stakeholders to utilize a Teensy 3.6 development board because the processor used on this particular board (MK66FX1M0VMD18, cortex M4F) was the same one Candela uses for there current C7 boats. Using the same processor would therefore have increased portability, but unfortunately initial testing proved that this was not a viable option. For one, another major stakeholder requirement was to use FreeRTOS, and unfortunately the only ported version of FreeRTOS available for the Teensy platform was incompatible with this particular board. It was revealed that the Teensy 3.6's handling of dynamic memory while printing floats caused a hard fault to occur. Second of all, the Teensy used the Arduino IDE for compiling and flashing code, so no proper debugging capabilities really existed beyond this serial printing.

As an alternative, a *NUCLEO-F429ZI* development board was used instead. This board was specifically chosen because it is similar in performance to the Teensy 3.6 in areas such as processor (ARM 32-bit cortex-M4), clock speed (180 MHz), ram (256K), and the plethora of timers, General Purpose Input/Output (GPIO) and supported communication protocols. Most members of the team were also familiar with the STM ecosystem (STM32CubeMX and STM32CubeIDE), so the learning curve was a non-issue. The programming on this is also done in C, so portability of the code itself is also improved compared to with the Teensy 3.6 that uses C++.

Manual Override Ccontroller

After some initial testing, it was found that using paddles to position the boat for test setups and avoiding collisions was cumbersome. It was therefore decided that a manual override controller should be added. A controller was developed using two analog joysticks housed in a 3D printed handheld enclosure, see Figure 5.14. This was then connected to the analog pins of the STM development board via a re-purposed ethernet cable.

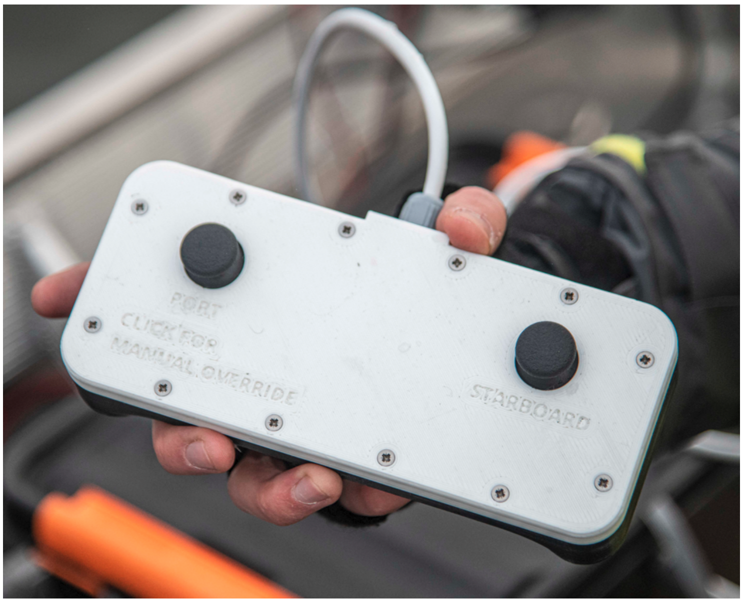


Figure 5.14. Joystick controller used to manually override autonomous system.

The left joystick on the controller controls both motor thrusts while the right controls the angle of both motors. Clicking both joysticks simultaneously enters the manual override mode and clicking both joysticks again returns back to autonomous mode.

5.4.6 Software Implementation

The Drivetrain subsystem is responsible for actuating the stepper and trolling motor in real-time; ensuring quick and reliable responses. To facilitate this, a multi-threaded control structure was implemented with FreeRTOS.

Control Overview

A high-level architectural flowchart of the program can be seen in Figure 5.15. The software has been implemented with consideration of boat operation in both autonomous mode and manual modes. This was facilitated by implementing the remote controller which can override the autonomous system if activated. The threads are set up to run sequentially between the each-other by a signal-flag system. The final implementation of the program contains 4 threads as seen in the Figure. These threads are Read Encoder, Control Stepper, Control Thrust, Remote Controller. The control flow of the program starts with reading the encoder to retrieve the current angle position. This signals the Control Stepper thread to iterate as long as the error is above a certain threshold. Once the error is below the threshold the Control Thrust thread is signaled to start. In this thread the force reference is calculate to a voltage value which is sent to the ESC to control the thrust of the motor. If the vessel is running in autonomous mode this is the overall control flow and after the thrust has been applied to the motors, the Read Encoder thread is once again signaled to start. If however the boat is operated in manual mode, an additional thread is signaled to start which reads the analog values of the remote controller joysticks and updates the thrust and angle references.

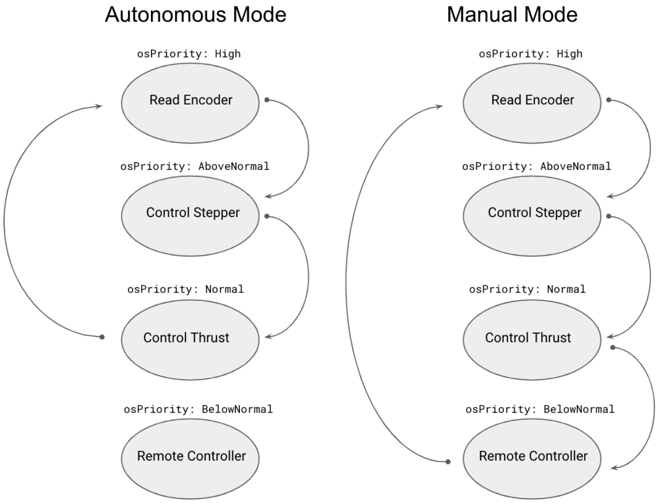


Figure 5.15. Architectual software overview of the drivetrain subsystem.

CAN Interrupt

There are 2 hardware triggered interrupts implemented in the subsystem. The first one is the CAN interrupt which is set whenever a message is received on the CAN bus. This message contains the the angle and thrust references that are past on from other subsystems. The message can either contain a reference for separate thrust commands and separate angle command for each motor, or if the thrust and angle should be equal between the two motor, a single reference can be sent for thrust and angle.

In order to distinguish between the references, a software filter has been implemented that checks the StdId of the message received. The CAN bus sends 3 StdIds of interest; 0x01, 0x02 and 0x03. For differential thrust and angle control, 2 CAN messages are sent on StdId 0x01 and 0x02, where each of the message contain the separate thrust and angle command as depicted in Figure 5.16. If however the same force and angle should be applied to the two motors, a single message is only required which then is sent with StdId 0x03.

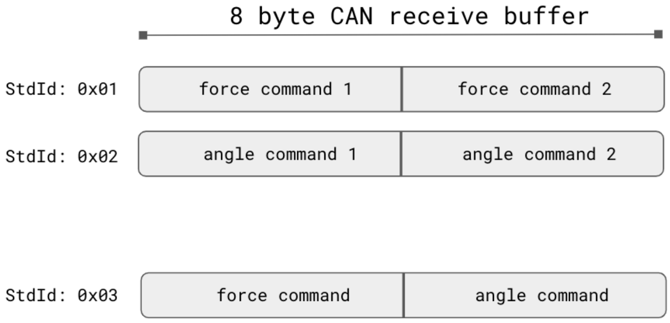


Figure 5.16. Overview of the CAN receive structure.

Controller Interrupt

A remote controller was implemented that allowed the boat to be manually overridden. The controller was activated by pressing both of the controller's joysticks simultaneously. This triggers the second hardware interrupt that causes the autonomous operation to be overridden. Once in manual mode the remote controller could be used to steer and actuate the thrust. When the boat was in manual mode, the Thrust and Angle commands sent over CAN would no longer update the references. They were instead updated by the joystick movement of the controller.

To reactivate the autonomous drive, the joystick buttons were to be pressed again. The interrupt would once again be triggered which engages the autonomous mode operation of the boat.

Thrust Control

The thrust controller has been implemented to give the trolling motor the correct thrust actuation. Due to lack of sensors and accessible motor parameters of the trolling motor, an open loop control approach had to be implemented. To facilitate this, a test was conducted which mapped the motor thrust values to motor voltage. This is discussed in more detail in the Verification and Validation chapter of this report.

To control the applied motor voltage, an Electronic Speed Controller ESC was implemented. The speed controller was controlled by a 8-bit serial command. To allow for both forward and backward thrust the 8-bit value was divided into two parts. Values between 127-0 mapped to negative 0-12V (reverse thrust) and values between 129-255 mapped to 0-12V (forward thrust).

Once a thrust reference was received, the force of the reference was recalculated to its corresponding motor voltage according to relationship that was established during the tests. The motor voltage was then mapped linearly to the correct value between 0-255 and sent to the ESC as an 8-bit UART command running at a 9600 baud rate.

Stepper Control

The angle of each motor was controlled by a stepper motor. The stepper motor operates by reading the angle value of the encoder that was mounted to the shaft of the stepper. The value of the encoder was fed back to the controller which applied a PWM signal to the stepper driver as long as the error was above 1.2 degrees. Once the error was below 1.2, the duty cycle of the PWM was set to 100 percent, which will engage the holding torque of the stepper and stop further rotation.

5.5 Positioning and Control

This section will describe the implementation of the Positioning and Control subsystem.

5.5.1 General Approach

The purpose for this subsystem is to guide the vessel towards the dock from a distance of 50 meters until it is close enough for the Near-field subsystem to take over and finish up the docking sequence. This is to be done in the following manner...

1. The global position is measured in geodetic coordinates and the heading of the vessel is measured in degrees. These are sent over to the Obstacle Detection and Path Planning subsystem over UART.

2. A path consisting of waypoints is returned. This is to be received repetitively at a constant time interval in order to account for deviations from the path.

3. Current position and heading is measured and the distance and bearing towards the next waypoint, which is of at least a set distance away, is calculated.

4. This error is sent into the control system which calculates how the vessel needs to be actuated in order to minimize this error.
5. The actuator commands are sent over to the Drivetrain subsystem using CAN.

The actuation in steps 3 to 5 are to be repeated at a fixed frequency. The same goes for the receiving of new waypoints in step 2, only this is done at a slower frequency. Initially, it was agreed that the actuation would be done at a frequency of 5 Hz whereas the receiving of a new path would be done at a rate of 0.25 Hz.

What was initially thought to be the solution did not always turn out to be that way. Hence, time has been put into designing models of other structures than what is to be showed, using other types of methods for control and path following. In the end it is the result that counts and moving forward in this chapter, only what was actually implemented will be discussed.

5.5.2 The Closed Loop System

The overall closed loop system was initially modelled as a typical feedback system where the vessel acted as the plant and with a non-typical control allocation block, as displayed in Figure 5.17. The purpose of this block is to convert the control output, calculated by the controller, into actuator commands, which could be interpreted by the trolling motors of the boat.

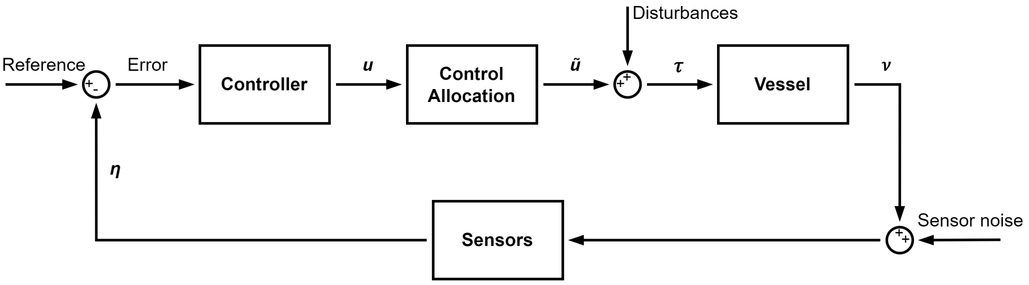


Figure 5.17. Overall block diagram structure.

This block diagram model was iterated and renewed. For ease of implementation, environmental disturbances, modelling errors, and sensor noise were not taken into account. Also, the control allocation was linearized, as described in Chapter 2.5, in order to ease with the controller design.

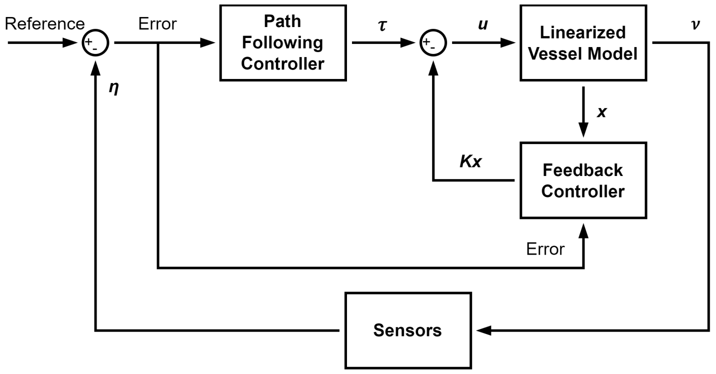


Figure 5.18. Final block diagram structure.

5.5.3 Control Allocation

The thurster configuration used was determined by the stakeholders as well as limited due to the dimensions of physical boat. This motor placement is displayed in Figure 5.19, where the boat is illustrated as a rectangle with a body-fixed coordinate system in the center, with the associated values of the placement in Table 5.2.

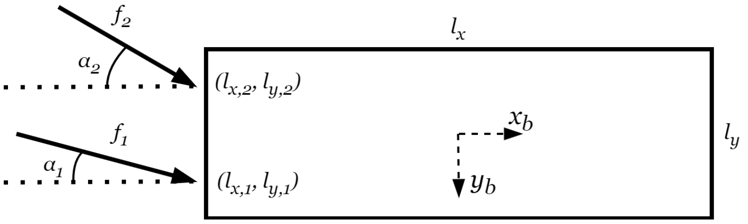


Figure 5.19. Vessel thruster configuration.

Parameter	Value	Unit
$l_{x,1}$	-1.65	[m]
$l_{y,1}$	0.378	[m]
$l_{x,2}$	-1.65	[m]
$l_{y,2}$	-0.378	[m]

Table 5.2. Parameter values of the thruster placement.

This thruster configuration results in having four independent control variables, namely the magnitude of thrust generated by each of the motors as well as the direction of the thrust. This results in having four control variables whilst working in only 3-DOF, hence resulting in an over-actuated control problem, or in the case of simultaneous steering, an under-actuated control problem with only two control variables. Initially, the over-actuated control allocation problem was tried to be solved using optimization, but since this would result in solving a non-convex optimization problem this method was discarded. Thereafter, solving the problem though the use of numerical iterative methods was considered. But due to not finding any earlier work on similar problems as ours, this was also discarded. This left us with the option of linearizing the thruster configuration, which is what was done in the end.

Linearized Control allocation

Following the approach presented in Section 2.5.2, the thruster configuration of the vessel was linearized. From this, both a model for differential and simultaneous steering was developed. Following Equation 2.17 resulted in a control input for simultaneous steering as

$$\tau_s = T_s u_s = 2 \begin{bmatrix} \cos\alpha^* & -f^* \sin\alpha^* \\ \sin\alpha^* & f^* \cos\alpha^* \\ l_{x,1} \sin\alpha^* & f^*(l_{x,1}\cos\alpha^*) \end{bmatrix} \begin{bmatrix} f \\ \alpha \end{bmatrix} \tag{5.3}$$

where f^* and α^* are the working point of which the system is linearized around. Due to the symmetry in thruster placement, $l_{x,1} = l_{x_2}$ and $l_{y,1} = -l_{y_2}$, Equation 2.17 was simplified to that of Equation 5.3.

As for the case of differential steering, this was done in a similar fashion by using Equation 2.20, resulting in

$$\tau_d = T_d u_d = \begin{bmatrix} \cos\alpha_1^* & \sin\alpha_1^* & l_{x,1}\sin\alpha_1^* - l_{y,1}\cos\alpha_1^* \\ \cos\alpha_2^* & \sin\alpha_2^* & l_{x,2}\sin\alpha_2^* - l_{y,2}\cos\alpha_2^* \\ -f_1^*\sin\alpha_1^* & f_1^*\cos\alpha_1^* & f_1^*(l_{x,1}\cos\alpha_1^* + l_{y,1}\sin\alpha_1^*) \\ -f_2^*\sin\alpha_2^* & f_2^*\cos\alpha_2^* & f_2^*(l_{x,2}\cos\alpha_2^* + l_{y,2}\sin\alpha_2^*) \end{bmatrix}^T \begin{bmatrix} f_1 \\ f_2 \\ \alpha_1 \\ \alpha_2 \end{bmatrix} \tag{5.4}$$

where f_1^*, f_2^*, α_1^* and α_2^* are the working points of which the system is linearized around. Moving forward, only the model for simultaneous steering was used for modelling, design and simulations. This was mainly due to a lack of time.

5.5.4 Vessel Model 1

For the first model of the vessel, the methodology described in Chapter 2.4 was used the derive a Multiple Input Multiple Output (MIMO) transfer function model of the system, as described by Equations 2.3 and 2.4. The vessel was modelled in 3-DOF using the low speed assumptions, described in Section 2.4.2, and under the assumptions of no environmental disturbances. Numerical values for the parameters used are displayed in Table 5.3. The mass of the vessel was estimated to be that of maximum load. The inertia was estimated from the CAD-model. Recalling from Section 2.4.3, the hydrodynamic coefficients of the added mass matrix and the potential damping matrix was calculated using Equations 2.7 through 2.9 as well as 2.11 through 2.13, and are displayed in Table 5.4.

	Parameter	**Value**	**Unit**
Acceleration of gravity	g	9.81	[kg/s^2]
Mass of vessel	m	315	[kg]
Inertia of vessel	I_z	269.3	[kg/m^2]
Density of water	ρ	997	[kg/m^3]
Hull draft	d	0.2	[m]
Length of vessel	L	3.3	[m]
Breadth of vessel	B	1.37	[m]
Time constant in surge	T_{surge}	175	[s]
Time constant in sway	T_{sway}	175	[s]
Time constant in yaw	T_{yaw}	175	[s]

Table 5.3. Parameter values used for modelling

Parameter	**Value**	**Unit**
$X_{\dot{u}}$	-15	[-]
$Y_{\dot{v}}$	-207	[-]
$N_{\dot{r}}$	-190	[-]
X_u	1.80	[-]
Y_v	2.89	[-]
N_r	2.62	[-]

Table 5.4. Calculated hydrodynamic parameters

This model was then turned into a state space model on the form

$$\begin{aligned} \dot{x} &= Ax + B\tau \\ y &= Cx \end{aligned} \tag{5.5}$$

which combined with the resulting linearized thruster configuration matrix, as of Equation 5.3, turns the system into

$$\begin{aligned} \dot{x} &= Ax + B_s u_s \\ y &= Cx \end{aligned} \tag{5.6}$$

where $B_s = BT_s$. This makes the system explicitly look like

$$\begin{aligned}
\underbrace{\begin{bmatrix}\dot{u}\\ \dot{v}\\ \dot{r}\end{bmatrix}}_{\dot{x}} &= \underbrace{\begin{bmatrix}\dfrac{-X_u}{m+X_{\dot{u}}} & 0 & 0\\ 0 & \dfrac{-Y_v}{m+Y_{\dot{v}}} & 0\\ 0 & 0 & \dfrac{-N_r}{I_z+N_{\dot{r}}}\end{bmatrix}}_{A}\underbrace{\begin{bmatrix}u\\ v\\ r\end{bmatrix}}_{x}\\
&\quad \underbrace{+2\begin{bmatrix}\dfrac{1}{m+X_{\dot{u}}} & 0 & 0\\ 0 & \dfrac{1}{m+Y_{\dot{v}}} & 0\\ 0 & 0 & \dfrac{1}{I_z+N_{\dot{r}}}\end{bmatrix}\begin{bmatrix}\cos\alpha^* & -f^*\sin\alpha^*\\ \sin\alpha^* & f^*\cos\alpha^*\\ l_{x,1}\sin\alpha^* & f^*(l_{x,1}\cos\alpha^*)\end{bmatrix}}_{B_s}\underbrace{\begin{bmatrix}f\\ \alpha\end{bmatrix}}_{u_s} \qquad (5.7)\\
\underbrace{\begin{bmatrix}y_u\\ y_v\\ y_r\end{bmatrix}}_{y} &= \underbrace{\begin{bmatrix}1 & 0 & 0\\ 0 & 1 & 0\\ 0 & 0 & 1\end{bmatrix}}_{C}\underbrace{\begin{bmatrix}u\\ v\\ r\end{bmatrix}}_{x}
\end{aligned}$$

Due to C being equal to the identity matrix, essentially $y = x$.

5.5.5 Vessel Model 2

From the system testing, another model was created using system identification. This was mainly due to that the previous model did not match the behavior of the actual vessel well enough, as shown later in Section 7.3.3. The test cases of time of acceleration to 1.8 m/s from standstill and turning radius at 1.8 m/s, here looking at the change in speed of the vessel, was used to approximate a linear model of the vessel in the surge and yaw direction. The first test is essentially the same as applying a step input of maximum thrust of the vessel, whereas the second one isn't really as straight forward of a test, but since there was no way to measure angular velocity, this was the best test avaliable. This resulted in a 2-DOF model.

System Identification from Step Response

A general linear first order transfer function can be written as

$$G(s) = \frac{k}{\tau s + 1} \qquad (5.8)$$

where k is the gain and τ is the time constant of the system. Looking at a step response for this transfer function both of these parameters can easily be determined. The k is equal to the dc-gain of the transfer function and τ is the time it takes for the system to reach approximately 63% of its final value. Also by knowing the maximum thrust of each motor from the testing in Section 7.1.1, as well as what the maximum torque this would apply to the vessel if the motors are rotated to its maximum position, the gain and time constant could be determined for each linear model. These parameters are displayed in Table 5.5 with the identified transfer functions given as

$$G_{surge}(s) = \frac{k_{surge}}{\tau_{surge} s + 1} \tag{5.9}$$

$$G_{yaw}(s) = \frac{k_{yaw}}{\tau_{yaw} s + 1} \tag{5.10}$$

Parameter	**Value**
k_{surge}	$5.316 \cdot 10^{-3}$
τ_{surge}	1.90
k_{yaw}	$1.098 \cdot 10^{-3}$
τ_{yaw}	5.80

Table 5.5. System parameters identified using system testing

The transfer functions was then converted into a MIMO state space model on the same form as in Equation 5.5. This was first done numerically in Matlab followed by applying the same linearized control allocation as in Equation 5.6, only that here the degree of freedom corresponding to velocity in sway was omitted. This resulted in the model being described as

$$\begin{aligned}
\underbrace{\begin{bmatrix} \dot{u} \\ \dot{r} \end{bmatrix}}_{\dot{x}} &= \underbrace{\begin{bmatrix} -0.5263 & 0 \\ 0 & -0.1725 \end{bmatrix}}_{A} \underbrace{\begin{bmatrix} u \\ r \end{bmatrix}}_{x} \\
&\quad + \underbrace{2 \begin{bmatrix} 0.1250 & 0 \\ 0 & 0.0313 \end{bmatrix} \begin{bmatrix} \cos\alpha^* & -f^* \sin\alpha^* \\ \sin\alpha^* & f^* \cos\alpha^* \\ l_{x,1} \sin\alpha^* & f^*(l_{x,1} \cos\alpha^*) \end{bmatrix}}_{B_s} \underbrace{\begin{bmatrix} f \\ \alpha \end{bmatrix}}_{u_s} \\
\underbrace{\begin{bmatrix} y_u \\ y_r \end{bmatrix}}_{y} &= \underbrace{\begin{bmatrix} 0.0448 & 0 \\ 0 & 0.0121 \end{bmatrix}}_{C} \underbrace{\begin{bmatrix} u \\ r \end{bmatrix}}_{x}
\end{aligned} \tag{5.11}$$

5.5.6 Sensors

Positioning

For positioning, the u-blox C099-F9P application board was used. It offers an easy evaluation of the ZED-F9P high precision GNSS module together with multi-band RTK. It allows concurrent reception of BeiDou, Galileo, GLONASS and GPS, delivering centimeter-level accuracy. I^{2}C was used for communication.

Heading

In order to receive an accurate measurement of the current heading, the SparkFun HMC6343 Breakout was used. It is a fully integrated high end electronic compass module that computes a heading direction accurate within a few degrees. It is also tilt compensated and calibrated to handle magnetic distortions. It combines a 3-axis magneto-resistive sensors with a 3-axis accelerometer. It has an on-board microprocessor for heading computations. The communication with HMC6343 was also used through I^{2}C

5.5.7 Feedback Controller

The feedback controller was designed mainly using LQR. Due to our system input being linearized, the optimal would be to solve the LQR problem each sampling in real time. This was not done here for ease of implementation. Instead, a small set of working points was chosen for which the LQR problem was solved using Matlab. Then, based on the control error, different control parameters was chosen according to a predefined gain schedule. This principle is illustrated in Figure 5.20 and the working points and gain schedule are to be found in Table 5.6. The error talked about is that to be described in detail in Section 5.5.8.

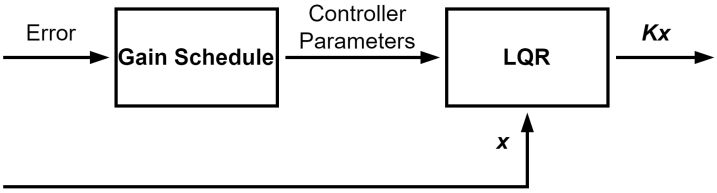


Figure 5.20. Feedback controller block diagram structure.

Working Point	Range of operation
$(f^*, \alpha^*) = (20\text{N}, 25°)$	Error $\in [20°, 30°]$
$(f^*, \alpha^*) = (30\text{N}, 12.5°)$	Error $\in [5°, 20°)$
$(f^*, \alpha^*) = (50\text{N}, 0°)$	Error $\in (-5°, 5°)$
$(f^*, \alpha^*) = (30\text{N}, -12.5°)$	Error $\in (-20°, -5°]$
$(f^*, \alpha^*) = (20\text{N}, -25°)$	Error $\in [-30°, -20°]$

Table 5.6. Gain schedule used in the feedback controller

5.5.8 Path Following Controller

In order to drive from point A to point B, a heading following controller was implemented. The reference in motor angle was derived from the error in bearing between the current position and goal position and the vessels current heading, seen in Figure 5.21. To compute the bearing between the vessels current position and goal position, the Haversine formula was used [99]. The bearing between two sets of geodetic coordinates can be seen in Equation 5.12 where φ denotes latitude, λ longitude, subscript 0 the current position and subscript 1 the goal position, a the square of half the chord length between the two coordinates and finally b denoting the bearing:

$$\begin{aligned} a &= \sin^2(\frac{\varphi_1 - \varphi_0}{2}) + \cos(\varphi_0) \cdot \cos(\varphi_1) \cdot \sin^2(\frac{\lambda_1 - \lambda_0}{2}) \\ b &= 2 \cdot \text{atan2}(\sqrt{a}, \sqrt{1-a}) \end{aligned} \tag{5.12}$$

The thrust reference was derived through an iterative testing process and based on the vessels stability. In order to keep the vessel stable and robust against large errors in heading, the thrust reference was switched between different linearized working points.

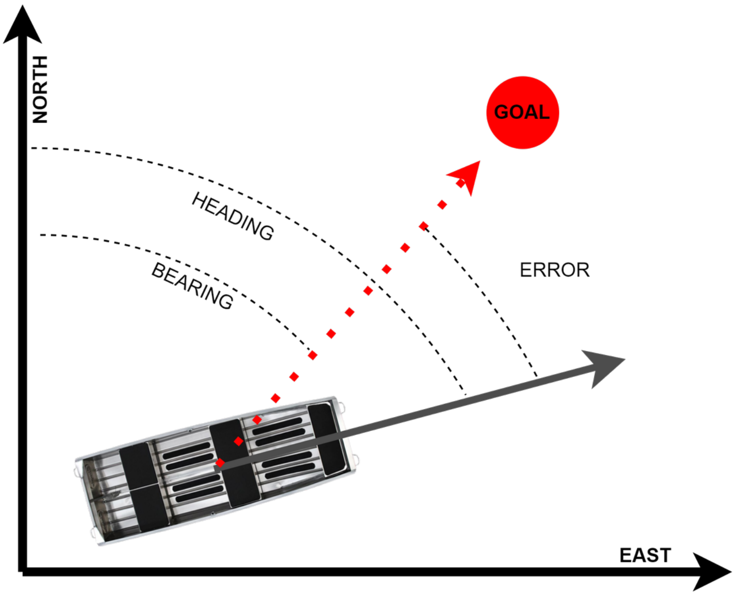


Figure 5.21. Illustration of heading tracking using bearing.

5.6 Obstacle Detection and Path Planning

This section will describe the implementation of the Path Planning and Obstacle Avoidance subsystem.

5.6.1 General Approach

With a predetermined goal position a path needs to be calculated to that goal while avoiding obstacles along the way. The process can be described in the following steps.

1. Retrieve two global geodetic coordinates, one for each corner of the goal dock.
2. Calculate center position between the coordinates and the angle of approach.
3. Generate a Dubins-Path from the boats current position and angle to the goal position and angle.
4. Check the path to make sure that it only passes over free water. If there is an obstacle in the path the path is recomputed with a different radius and checked again.
5. Discretise the path and send it over UART to the control system computer.

When all the steps are completed steps 3 to 5 are repeated at a set frequency. This makes the path planning agile enough to avoid new obstacles that are detected by the stereo camera. If no valid path is found, the path planner sends a stop signal to the control computer.

5.6.2 Computational Platform

The chosen computer platform for the subsystem is the Jetson Xavier NX from NVIDIA, a single board computer designed for embedded applications with high computational needs. It runs a custom version of the Linux distribution Ubuntu called Jetpack. One of the main benefits of using a single board computer in this setting is that the software can be programmed and tested directly on the goal computer.

5.6.3 Software Platform

To implement the Stereo Vision, Path Planning and communication for the NVIDIA Jetson Xavier NX The Robot Operating System (ROS) was used. ROS is a worldwide recognized open-source platform built for robotics development. ROS is mainly used in research but can be found in many real world applications as well. The platform support code written in C++, Python and Lisp. There are many pros to using ROS, one of the most prominent is how easy it is to implement processes along side each other. ROS uses nodes to represent each process and passes the information between these separate processes through topics. Programs can be built separately and then ran together under a ROS master making the nodes able to communicate with each other passing messages as seen in Figure 5.22 [102]. During the project new ROS nodes was added when needed. This was made easy with ROS adding new functionality seamlessly with the current functionality. Another pro with ROS is its open-source nature. In many cases during the course of this project functionality that was necessary was already developed by the community making it a plug and play solution.

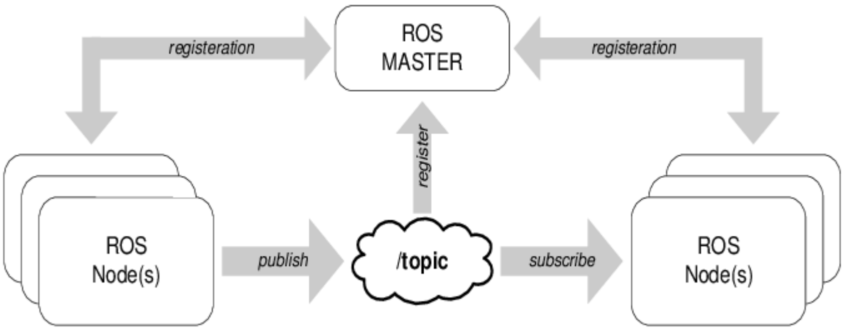


Figure 5.22. Illustration of ROS nodes, topics and Master [116].

.

5.6.4 Stereo Camera

The Zed 2 camera from Stereolabs was chosen for the implementation of the stereo camera. The Zed 2 has a resolution of up to 4416x1242 and a depth range between 0.2 – 20 meters. The camera does not have an official IP rating but is designed to resist rain and water spray and should be able to operate between temperatures of -10 to +45 degrees Celsius according to Stereolabs [137]. The camera also comes with a powerful Software Development Kit (SDK) and a wrapper for ROS, making the integration simple and fast.

To mount the stereo camera, a mast was created to elevate the camera to a height of 1m above the boat hull see Figure 5.23. This gives the camera a wide field of view without obstructions. It also doubles as a mount for the GNSS-antenna which benefits from an as unobstructed placement as possible.

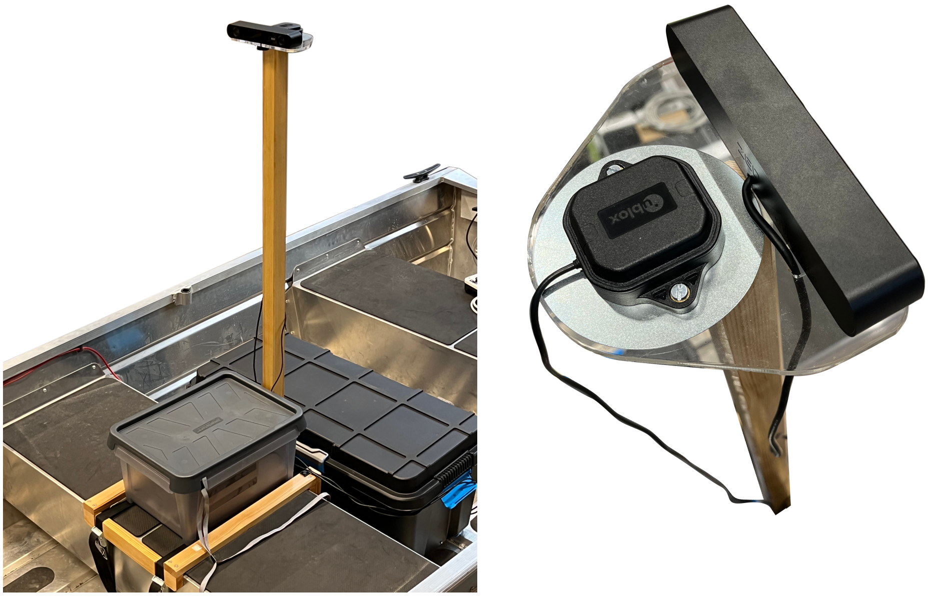


Figure 5.23. Left: Complete mast assembly. Right: Mounting plate on mast with GNSS-antenna and ZED 2 stereo camera.

5.6.5 Mapping

The implementation of the grid-based SLAM is done using a package called gmapping which provides a laser-based SLAM [36]. The Zed 2 calculates a depth image using its stereo cameras, this depth image contains a depth value for each pixel in the images which is encoded on 32 bits. Since the gmapping package uses laserscans for the SLAM process, the depth images are converted to laserscans using a ROS package called "depthimage_to_laserscan" [18]. With the depth image to laserscan package it is possible to select only a specified pixel height of the depth image which will be used to generate the laser scan. Selecting a pixel height makes it possible to specify which areas will be used for the mapping. The generated laser scan will then be sent to the gmapping node which generates an occupancy grid, the occupancy grid consists of costs between 0-100 and an unknown point has the cost of -1. In this implementation, a 0 value is free space and any value above 0 is considered to be an obstacle. The occupancy grid that is generated is then used to determine where it is possible for the boat to drive and where it is not. Since the boat is larger than one square in the grid, the costs are inflated in order to create a sort of padding to prevent the boat from colliding with the wall when a path is planned. A visualisation of this can be seen in Figure 5.24, where the white area is free space, the black area is a detected obstacle, the grey area is the padding, and the rest is unknown area.

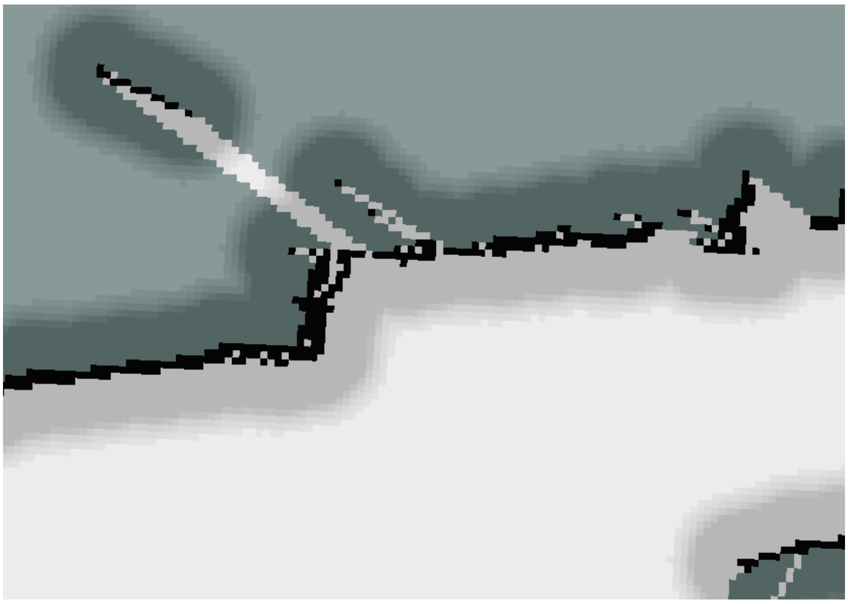


Figure 5.24. Example of grid mapping with inflation.

5.6.6 Path Planning

The path planning implementation and choice of path planning algorithm was heavily dictated by the strict requirements set on the path being created. The path had to be smooth, without sharp corners and a with generous turning radius. Furthermore it is not only a planning of the boats position but also the it's heading. It has to start from the boats current position and heading and end in the docks position and a heading perpendicular to the dock. It also has to be relatively efficient, so re-planning can be done multiple times during a docking sequence.

A number of different algorithms where considered, one simple algorithm stood out as a good choice, Dubins path. Dubins path was originally defined by L. E. Dubins in 1957 [23]. It is an optimization of of a path between two points on a 2D-plane. The start and end points have defined tangents to the path, basically headings. The path is constrained by a predefined turning radius. A visual representation of a Dubins path can be seen in Figure 5.25.

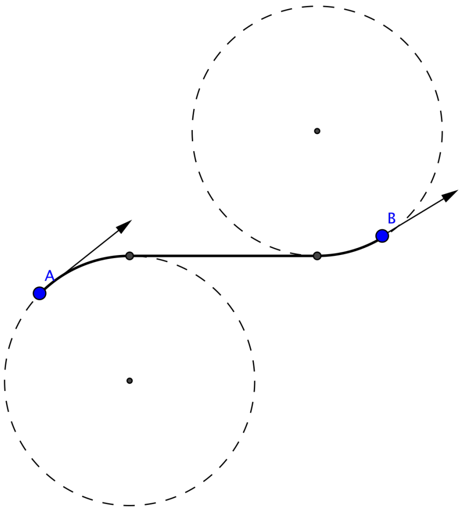


Figure 5.25. Example of a Dubins path [25].

Where A is the starting point and B is the end point. The path is simply created with two circle sectors and a straight line. Where the circle sectors have the predefined radius. The algorithm was implemented through a Python package called *Dubins-Curves* [24]. It provides a very efficient Dubins path implementation written in C/C++, but accessible in Python.

5.7 Near-field

This section will describe the implementation of the Near-field subsystem.

5.7.1 General approach

To summarize the chosen concept for the near-dock approach first described in Section 4.5.4: two ultrasonic sensors were used to ensure low impact speed at a perpendicular angle to the dock by minimising the approaching angle derived from the distance measurements. Two load cells were used to ensure that the front of the boat keeps stationed at the dock during passenger embarking and disembarking. It does so by correctly actuating the motors in case of any force deviations from any of the front edges on the boat, e.g. if the boat would start sliding.

5.7.2 Sensors

The following sensors where used for implementing the Near-field sensing and control.

Ultrasonics

The ultrasonic sensors used was A02YYUW Waterproof Ultrasonic Sensor by DFRobot. The detecting range for a flat object is between 3-450 cm and the measuring generates a digital output following the UART protocol. The sensor has waterproof grade IP67 and has a specified temperature operating range from -15°C to 60°C.

Load cells

The load cells used was the compact compression load cell FX29K0-100A-0100-L from TE Connectivity thanks to it's form factor, robustness and flexibility in configuration options. This specified model is configured for digital output following the I^{2}C protocol and has a fixed I^{2}C address. It's dimensioned to measure up to 500 N but has an 2.5x over-range capability in case of any unexpected overloads. The load cell has an a compensated temperature range from 0-40°C which means that means that accuracy is not tested or compensated outside that range.

5.7.3 Development Board

Set up of development board and software for near-field system.

Hardware

In the final setup all sensors were connected to a Nucleo STM32L476RG. This was sufficient thanks to low power consumption, high performance processor, multiple I^{2}C-buses, CAN-bus and a familiar development platform.

Software

All code for this software was written in C in the development environment STM-CubeIDE. The near-field board receives input from the sensors and then sends it on to the control algorithm located on the control board. The control board does the calculations and sends the thrust to the motors. The information from the ultrasonic sensors are received by interrupts and Direct Memory Access (DMA). The load-cells are being polled with a specific time interval. In order to increase performance the real time operating system FreeRTOS was used. That way ultrasonic sensors and load-cells could be placed in different threads and run individually. The load-cells are zero during the approach of the dock and will therefore not send information, but the same moment the same moment the load-cells sense an impact, a load-cell message will be sent to signal the control loop to go into docking mode. Once the load-cells start sending the ultrasonic will stop sending. However, if the load-cells stop receiving inputs and have lost contact with the dock, the system returns to relying on ultrasonic and continue the step of approaching the dock.

5.7.4 Control loop

Later on through the project Near-field over took some responsibility for close range control. The controller started at 3.5 m, when ultrasonic had a steady sensing of the dock. It then slowly guided the boat straight in towards the dock. To simplify the setup the motors were decided to keep still at 0° and maneuvering was achieved by differential thrust. The ultrasonic sensors gave two different positions from each end of the front of the boat. By assuming the dock being a flat surface the angle between the boat and the dock could be calculated, see Figure 5.26 The position used in the control algorithm was the shortest perpendicular distance from the dock to the boat. In order to calculate the the velocity the position was saved and used to calculate the distance of which the boat had traveled until the next reading. Through dividing the distance by the interval between the readings the velocity was calculated.

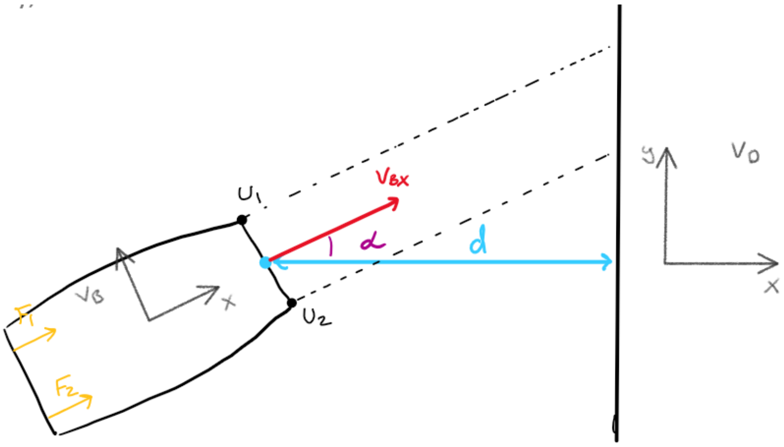


Figure 5.26. This figure illustrates different parameters of the Near-field docking stage. Yellow shows thrust of motors, red the velocity, blue the perpendicular distance from dock to boat and purple the angle of the boat.

The first controller was a PI cascade controller, see Figure 5.27. Starting with a P controller of the position and a PI controller of the velocity. The velocity controller calculate a required thrust for the boat to achieve the correct velocity which is then split evenly on the two motors of the boat. To adjust the angle a P controller calculate a small correction force which was added to the thrust on one motor and subtracted on the other. This differential thrust cause a momentum and the boat to turn without changing the resulting force on the boat. The controller was tested with a simplified mass-damper model of the boat in Simulink.

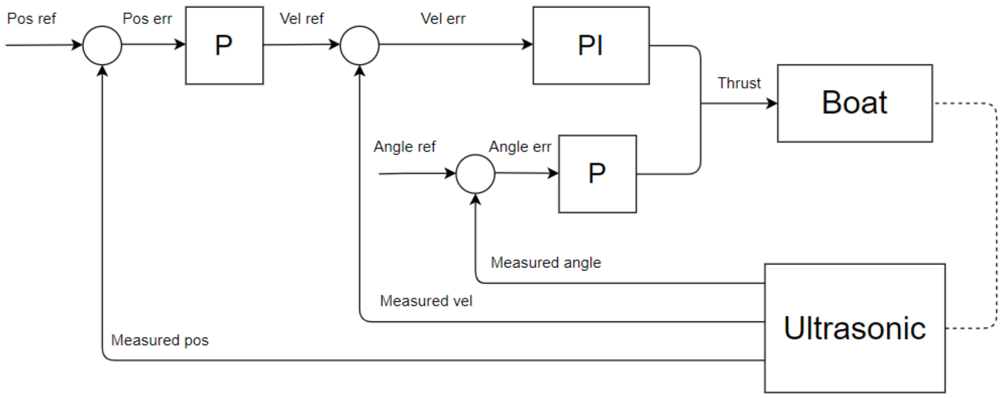


Figure 5.27. Block diagram of PI cascade controller calculating required motor thrust

5.7.5 Front bumper

The bumper in the front was implemented to to make impacts to the dock slightly smother as well as increase the friction to keep the boat from slipping sideways while stationary at the dock. The bumper consist of a piece of wood and rubber blocks. The bumper needed to be designed in a way to allow mounting and measuring the load of the load cells. The front of the boat has about 5 cm flat surface. To get a larger surface to place the bumper a triangular piece of wood was screwed on the boat. The larger piece of wood was screwed on with four M8 bolts along the whole front of the boat to make it steady enough for the forces of the impact. The load cells were placed in the wood at each end of the front of the boat. The rubber blocks had a small screw to push on the load cells during an impact. To be able to measure load individually there are two blocks one corresponding to each load cell.

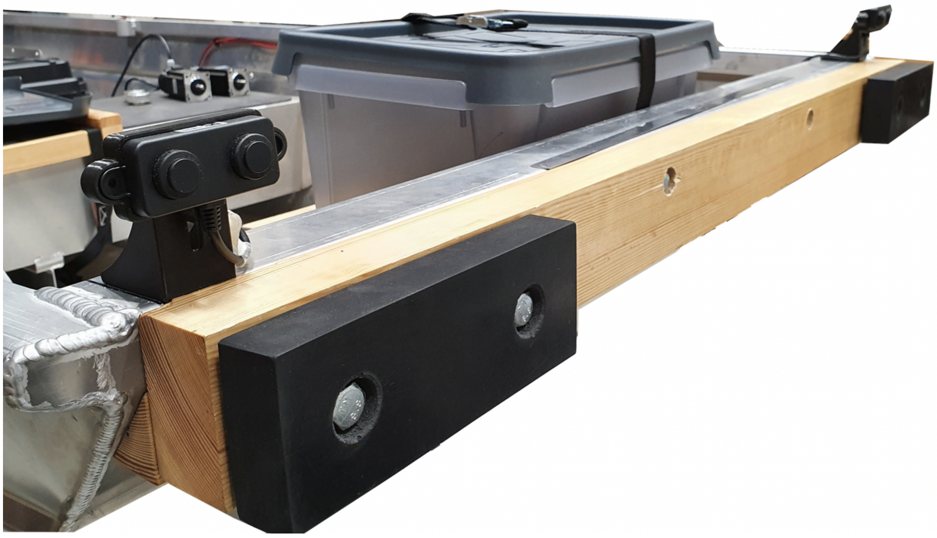


Figure 5.28. Image of the completed bumper assembly with ultrasonics mounted above.

The rubber blocks needed to be mounted robust enough to withstand the sideways forces from the boat during the docking stage but still allow the force forward do be measured in the load cells. This was done by placing two M10 bolts through each block. Bearings were 3D-printed and placed in the hole of the rubber to provide a low friction linear movement. To not get accidental inputs from the blocks, small springs were inserted between the wood and the rubber, see Figure 5.29.

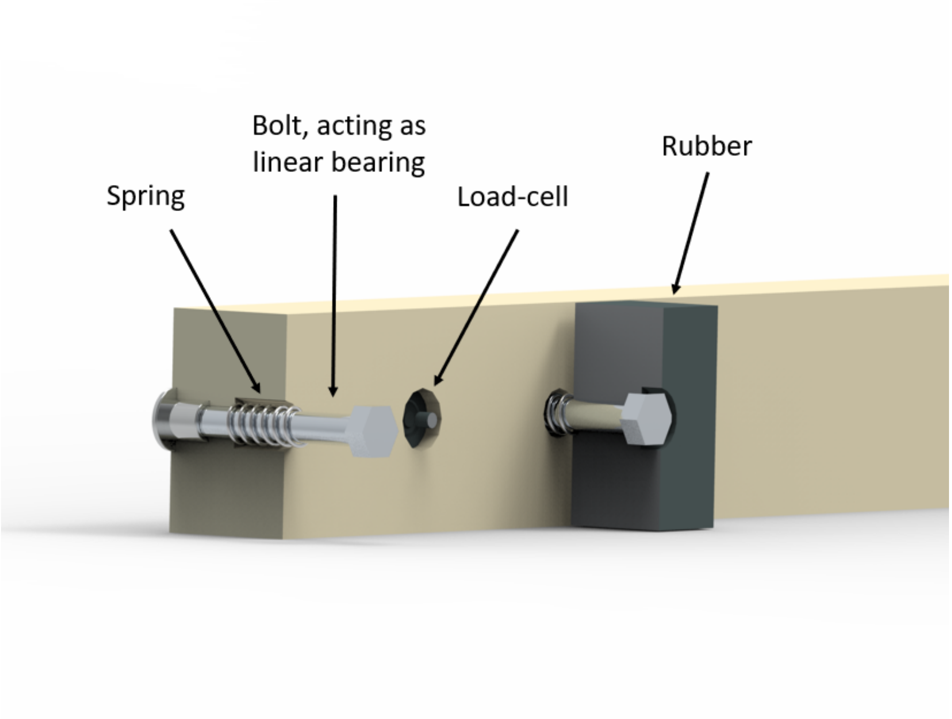


Figure 5.29. The bumper was designed to allow the normal force to the dock to be measured by the load-cells but still withstand the shear forces during the docked stage.

5.7.6 Electrical circuit

A small circuit board was manufactured to provide power to all sensors and the development board. A buck converter transformed the voltage from the batteries that provided 12V down to 5V, which then was fed into the circuit board. The development board was powered by 5V but was also feeding 3,3V back to the power board since the load cells and CAN bus required 3,3V. All 5V sensors were powered by the buck converter to avoid running too much current through the GPIOs of the development board. The pull up resistors required for I^2C communication were also integrated on the board, see Figure 5.30. The resistors are surface mounted on the bottom side of the board to not be in the way and the ground plane are on top of the board. The development board, buck converter, power board and the CAN-board were all mounted in a water-proof enclosure with waterproof connectors on the outside of the enclosure, allowing plugging the sensors in- and out and portability of the electronics enclosure. This configuration is illustrated in Figure 5.31.

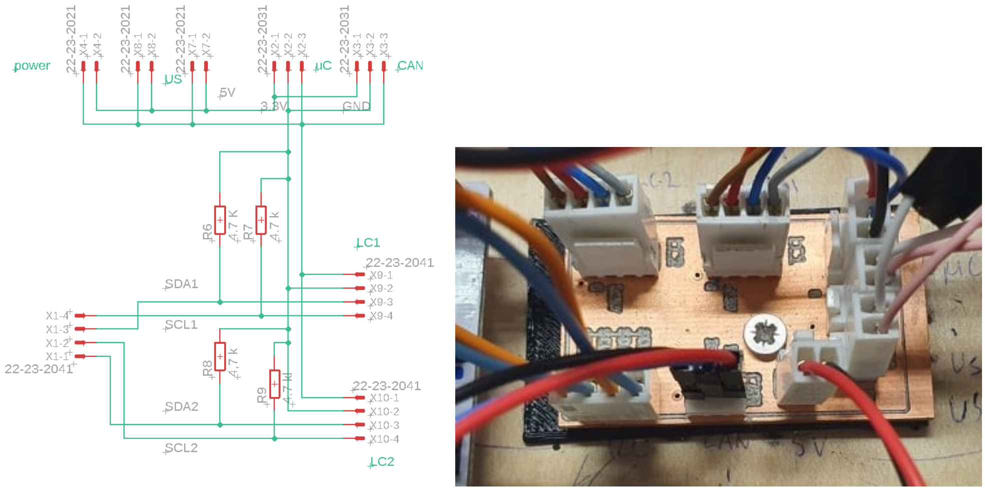


Figure 5.30. The power board to distribute 5V and 3,3 V to the micro controller and all sensors. Also integrated is the pull up resistors for I^{2}C communication.

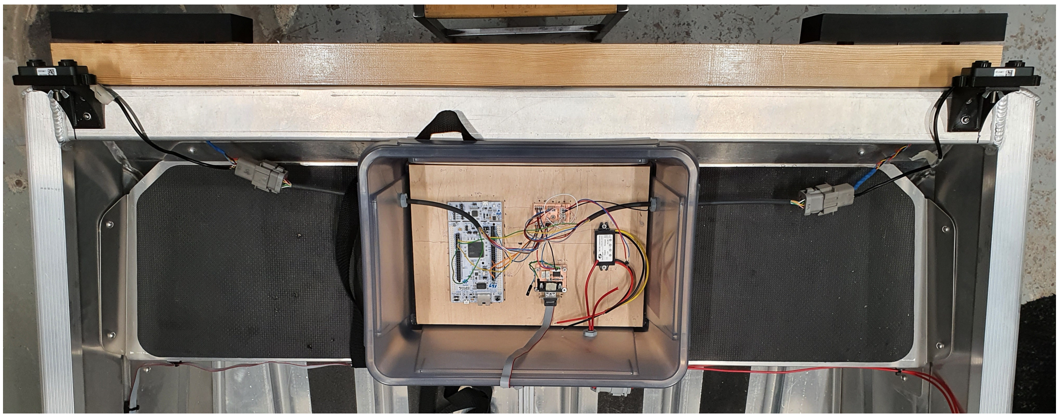


Figure 5.31. Waterproof enclosure and connectors for the sensors.

Chapter 6

Verification and Validation

This chapter explains our approach to testing and describes the motivations for why the eventual results from these tests validate the requirement of the project. Due to the team being structured into various nodes of functionality, it was easy to test individual components and then slowly build complexity before testing further.

In general, each subsystem first tested their individual components, integrated them with other subsystem hardware, tested further, and then began integrating with other subsystems. Individual aspects of the subsystem integration were then tested, once verified, the system was tested as a whole. This correlates to the workflow described in Chapter 3.

6.1 Unit tests

This section will outlines the unit tests that were made to validate and test individual components of each system, independently.

6.1.1 Drivetrain

Trolling motor thrust

In order to be able to control the thrust output from the trolling motors, the actual voltage being applied to the motor needed to be mapped to the resulting thrust. Independent testing was done by mounting one trolling motor to a small boat the group was able to borrow. A small luggage scale was then fastened to the boat and by running the motor at varying voltages, a corresponding output thrust could be identified. These values could then have a best-fit regression line applied to them to get an equation that relates the two factors of voltage vs, thrust.

Stepper motor

Before integration in the drivetrain, the stepper motors were tested in isolation to verify angular velocity and microstepping resolution and to ensure that these would be sufficient in turning the trolling motors. Test code implemented to see how the stepper reacted to different microstepping resolutions, which can be changed directly on the stepper motor drivers, as well as how different pulse frequencies affected it. This was done through changing the delay between pulses to create a square wave of differing frequencies. This frequency was incread as much as possible to extract the maximum possible Revolutions Per Minute (RPM) of the stepper before it started visually missing steps. A PI controller were also investigated such that the transition from current angle to goal angle would be as smooth as possible. The thought behind this was to prevent unnecessary jerk on the trolling motor, possibly causing belt slippage.

6.1.2 Positioning and Control

Position sensor

The ZED-F9P GNSS unit was tested and verified together with the provided u-blox software U-Center, which can communicate with the unit through serial communication. The software has a map-overlay to verify your position and also gives accuracy estimates.

Heading sensor

The magnetometer was tested through comparison with a handheld compass. The magnetometer was positioned close to a buck converter which had a considerable disturbance on the magnetic field. This showed through multiple readings on both the magnetometer and the compass. However, after moving the magnetometer further away, the disturbance was minimal. The magnetometer had some deviations compared with the compass, but was then calibrated accordingly.

Path Following LQR Controller

The path following LQR controller was sampled at 5 Hz and tested and tuned through multiple iterations in Matlab and Simulink, by using a fixed path of points as references. When the error between the vessels position and the goal position reached a threshold, a new reference position was used, until it had reached the final position.

6.1.3 Near-Field

Initially all sensors within the Near-Field were tested separately to confirm their functionality and the setup on the development board.

Ultrasonic sensors

The ultrasonic sensors were tested to verify the distance reading by letting them measure a known distance. To confirm that there would be no trouble with the sensors interfering with each other, these were also tested running simultaneously. Another potential issue was with the sensors measuring water instead of the intended dock surface. Hence, a test was structured to evaluate how far below the sensor it could detect obstacles. This was made by placing the sensors at the end of a 3 m long table with a vertical board at the end and controlling when the received distance switched.

Load cells

The load-cells were readings were verified by placing them on a scale and a comparison between the value of the scale to the values received from the load-cells were made.

Bumper

Before finalizing the bumper concept, a simple prototype was built to verify the functionality, seen in Figure 6.1. The concerns were that when hitting a corner of the rubber block it would lock the movement of the block and prevent it from putting pressure on the load-cell.

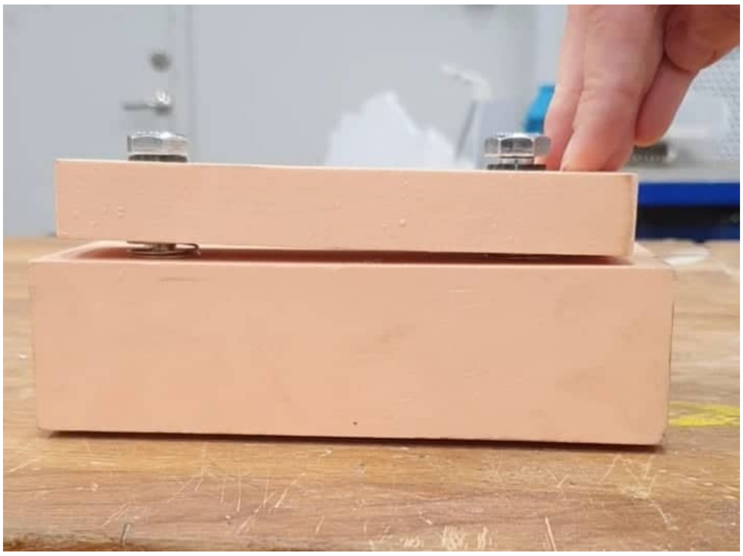


Figure 6.1. Prototype to verify functionality of the bumper.

Near-field Control loop

The control algorithm was built and tested in MATLAB and Simulink. This was done by using a simplified simulated model of the boat. The tuning of the controller was done in several steps. Starting with the open loop and then adding on parts and tuning parameters until the result behaved as desired. In order to verify how robust the system was against noise, multiple square wave signals were added to the feedback signal in the control loops. One big difference between the model and the actual boat is how it measures distance. The ultrasonic sensors measures the distance between the boat and the dock meanwhile the model calculates the velocity and integrate it to receive the position. That means that the model only knows how far it has traveled, which means that the reference will be 3.5 m for the model and 0 for the actual boat. To further evaluate the the control some calculations and a pole-zero map as well as a bode plot was made.

6.1.4 Obstacle Detection and Path Planning

Mapping accuracy

To test the accuracy of the map and the depth images, the camera was placed towards a flat wall at distances between 2-15 meters. The distance from the depth images was extracted from the middle point of the image and the distance to the wall in the map was determined by the distance between the current position of the camera and the first object on the map in front of the camera.

Path planning

A simulation was ran in order to test the time to calculate the path between the boat and a hard coded point on the map. The boat was moved around using fake GNSS signals to test the planning from different distances. The distances ranged from 18-36 m, 10 individual tests where made and an average was calculated.

6.2 Integration tests

This section will cover the testing that was made in the effort to integrate different systems and sensors in to one homogeneous system.

6.2.1 Stepper motor and encoder integration

The encoder allows the error between the current position and desired reference angle position to be calculated and the stepper motor to be actuated accordingly, but its not as simple as it sounds and many factors play a role in the actual performance. Although it was initially calculated that t a 1480 Hz input frequency to the driver should get the desired performance, this was based on estimated dynamics and neglected factors such as controller stability, stepper resonance, physical robustness, and the available torque while half-stepping.

To determine the optimal stepper motor control parameters in terms of PWM frequency, half-stepping, and error tolerance, iterative testing was conducted while in the water. The stepper is signaled to actuate the full range of motion (120°) and PWM frequency is first increased to be as fast as possible while retaining stability and without inducing too much vibration. Following this, half-stepping is increased to provide smoother actuation. This process is repeated until performance is satisfactory. The last item adjusted is the error tolerance to ensure stability and robustness against disturbances.

In order to quantify the actuating time, a phone camera was used to capture the event in slow-motion and the time stamps of the frames were analysed to determine the actuating time. This value was then considered in the formulation of the control algorithms to determine feasible rates in which reference angles could be sent for actuation.

6.2.2 Manual override controller integration

The controller was verified for robustness to make sure that the manual override interrupt was working as intended during all phases of the docking sequence. Tests were conducted where a docking sequence was ran, such as near-field docking, path planning or path following, and a controller interrupt was sent to initiate manual override. After noting that the vessel was in manual mode, another interrupt was sent to make sure that it returned to automatic mode as intended. In doing so, we could ensure reliable performance regardless of the docking sequence and the particular messages being received.

6.2.3 Positioning and Control

The heading and positioning sensor systems were integrated and tests were conducted to verify the functionality of I^2C communication, with both shared and individual ports. The controller was also tested with the sensors and hard-coded bearing references, to see if reasonable outputs were achieved using the magnetometer.

6.2.4 Near-field integration

Indoor tests were conducted to verify that all Near-Field sensors were able to run together simultaneously and get accurate readings.

When all the sensors were mounted onto the bumper, pressure was applied to the rubber blocks and the readings were plotted to the terminal on a computer to verify the force measuring mechanism of the bumper. Distance measuring verification for the ultrasonic senors were made by moving a large flat object (a whiteboard) towards the front of the boat to simulate a dock approach. The readings were plotted to the terminal in real-time to estimate if the readings were true to reality.

It was also verified that the sensor readings could be transmitted over CAN by connecting the near field node to the control node and then plot to the terminal from the control node.

6.3 System tests

This section will cover the testing of the whole system working together. These tests were conducted in the water.

6.3.1 Boat parameters

A number of test were carried out to better understand the boats characteristics and verify that the whole drivetrain works as expected. The following test where carried out:

- Acceleration to 1 m/s and 1.8 m/s.
- Braking time from 0.4 m/s, 0.8 m/s and 1.8 m/s.
- Braking distance from 0.4 m/s, 0.8 m/s and 1.8 m/s.
- Stopping time from 1 m/s and 1.8 m/s.
- Stopping distance from 1 m/s and 1.8 m/s.
- Turning radius at 1,4 m/s and 1.8 m/s.

Acceleration time is measured from recorded GNSS speed data. Braking distance is defined as actively stopping the boat by setting a reference velocity to 0 m/s and utilizing a simple PI velocity controller to stop the boat. In the braking test, the boat was considered stopped when the speed went below 0.1 m/s. This is an important metric, the braking distance has to be well below the range of the obstacle detection system, which in this case has a range of approximately 15-20 m.

Stopping distance is defined as shutting of the motors completely at the given speed and then letting it come to a stop by itself. In the stopping test the boat was considered stopped when the speed went below 0.2 m/s. In the turning radius test, a constant force was applied to the motors. After a predetermined time, the boat turned the motors to the max angle of 30 degrees and made a full circle. The radius was then measured via recorded GNSS positional data.

6.3.2 Near-field docking

Once the boat was in the water, tests were conducted to verify the functionality of the sensors and the near-field control loop. The Simulink setup was written in C code and implemented on a micro-controller. The parameters tuned in the simulation was transferred as a starting point for real life testings. First the control was tested with a low constant velocity reference to verify the functionality of the velocity control. Then, position control and angle control was added to the system and parameters were tuned to fit the actual boat.

6.3.3 Model Evaluation

In order to verify the models used for designing the path following LQR controller, simulations were made to compere how the models compared to the test results from the boat parameter testing. Essentially, a step response of each model (from applying maximum thrust compared to what velocity was measured) was carried out and compared to the results. The acceleration test to 1.8 m/s in surge and looking at the change of speed in the surge direction during the turning radius test for yaw was of particular interest. The results from this test was then used to inform any simplifications that could be made too.

6.3.4 Positioning and Control

The path following LQR controller was tested together with the heading and positioning sensors, using both a single fixed point on the dock and a hard-coded path made up of multiple waypoints. The geodetic coordinates were logged for evaluation together with the references. At this point, one of the models had been scrapped based on the tests from the model evaluation testing, so only the model based on system identification (as described in 5.5.5) was used.

6.3.5 Environment mapping

The mapping was tested on land before starting the testing on the water in order to verify that a clear map can be generated. The test was conducted by walking trough a hallway at Brinellvägen 83 while holding the Zed stereo camera. The mapping was also tested on the boat while in the water. During the water test, the boat was driven around the harbor of Skeppsholmen with the Zed camera mounted on the mast. The water test was recorded to a rosbag so it could be evaluated afterwards.

Chapter 7

Results

In this chapter, the results from the various unit, integration, and system level test cases originally identified in Chapter 6 are presented. These results are then used to confirm which requirements have been fulfilled.

7.1 Unit tests

This section covers the validated testing results of each independent component mentioned in Chapter 6.

7.1.1 Drivetrain

Trolling motor thrust

After the data points had been obtained from statically testing the trolling motor thrust, a best-fit regression line was made. Initial assumptions were of a linear relationship, but the plotted data ended up resembling a polynomial fit line better. These data points were plotted and a polynomial regression line was fitted using MATLAB and an additional ”polyfitZero” function that ensured the fit line passed through the origin, see Figures 7.1 and 7.2 below [90].

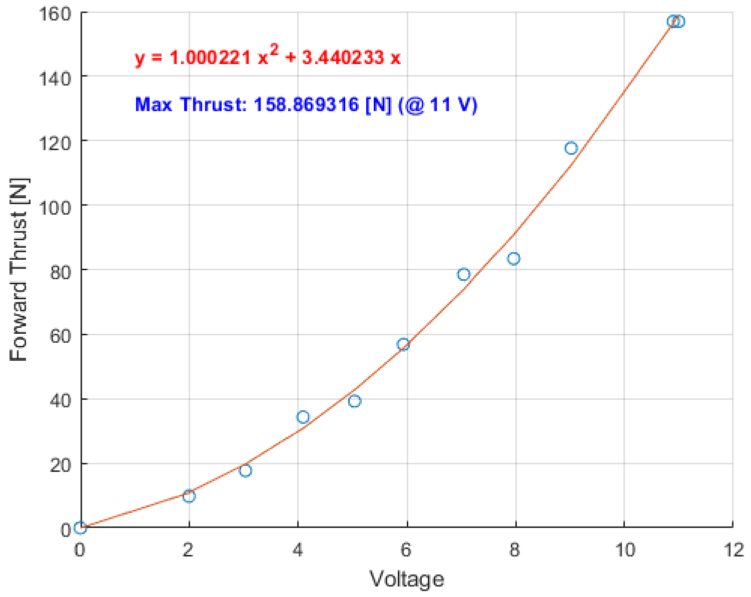


Figure 7.1. Forward thrust tests of trolling motor (singular motor)

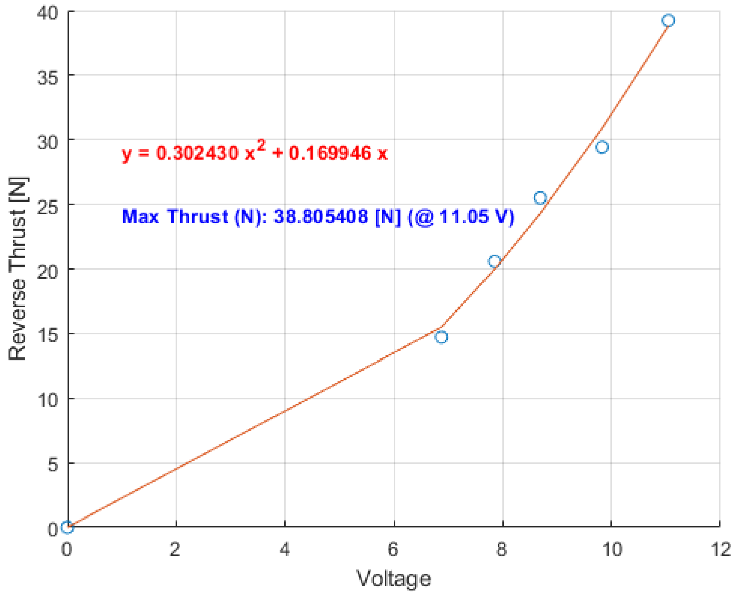


Figure 7.2. Reverse thrust tests of trolling motor (singular motor)

As shown in the plots, the max forward thrust was determined to be **158.869 N** and the max reverse thrust was found to be **38.805 N**. These were used as saturation limits when simulating the boat dynamics and when sending reference thrust inputs to the Drivetrain controller. The correlating polynomial equations that were used to generate ESC input signals in the Drivetrain controller can be seen in equations 7.1 (forward thrust) and 7.2 (reverse thrust) below.

$$F_{forward}(V) = 1.00V^2 + 3.44V \tag{7.1}$$

$$F_{reverse}(V) = 0.302V^2 + .170V \tag{7.2}$$

Stepper motor

The stepper motors were tested with no load in order to gauge how fast they can reliably be ran without loss of steps. The result of this was that the motors could reliably be ran with a pulse frequency of **0.1 ms**. This paired with a microstepping factor of 4 gives a theoretical sweep time of **0.027 s** (without load). The implementation of a PI controller to control the sweep proved to be unnecessary since the same result could be achieved with a simple statement in the code increasing delay between pulses as the error from the goal angle reduces.

7.1.2 Positioning and Control

Position sensor

The GNSS unit was verified to have a position accuracy of ±1 cm in u-center.

Heading sensor

After calibrating the magnetometer, the accuracy of the outputted heading was estimated to $\pm 1^\circ$, when comparing to a handheld compass.

Path following controller

After iterative testing, final weights of the controller objective function were determined to be...

$$Q = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad R = \begin{bmatrix} 0.2 & 0 \\ 0 & 5000 \end{bmatrix} \tag{7.3}$$

Using these at each working point resulted in the control parameters displayed in Table 7.1. Using this controller in simulations for a specific path resulted in the behavior as seen in Figure 7.3.

Working Point	Control Parameters
$(f^*, \alpha^*) = (20\text{N}, 25°)$	$K_{25°} = \begin{bmatrix} -2.00 \cdot 10^{-1} & -9.47 \cdot 10^{-3} \\ -1.50 \cdot 10^{-4} & 3.34 \cdot 10^{-5} \end{bmatrix}$
$(f^*, \alpha^*) = (30\text{N}, 12.5°)$	$K_{12.5°} = \begin{bmatrix} -2.14 \cdot 10^{-1} & -4.84 \cdot 10^{-3} \\ -1.14 \cdot 10^{-4} & 5.37 \cdot 10^{-5} \end{bmatrix}$
$(f^*, \alpha^*) = (50\text{N}, 0°)$	$K_{0°} = \begin{bmatrix} -2.19 \cdot 10^{-1} & 0 \\ 0 & 9.16 \cdot 10^{-5} \end{bmatrix}$
$(f^*, \alpha^*) = (30\text{N}, -12.5°)$	$K_{-12.5°} = \begin{bmatrix} -2.14 \cdot 10^{-1} & 4.84 \cdot 10^{-3} \\ 1.14 \cdot 10^{-4} & 5.37 \cdot 10^{-5} \end{bmatrix}$
$(f^*, \alpha^*) = (20\text{N}, -25°)$	$K_{-25°} = \begin{bmatrix} -2.00 \cdot 10^{-1} & 9.47 \cdot 10^{-3} \\ 1.50 \cdot 10^{-4} & 3.34 \cdot 10^{-5} \end{bmatrix}$

Table 7.1. LQR controller gains used at each working point

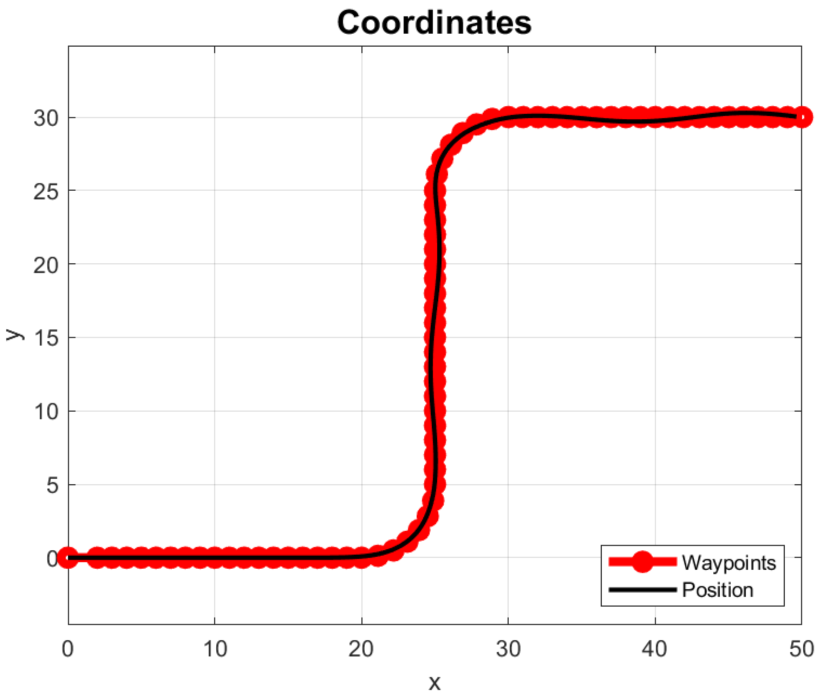


Figure 7.3. Simulation of the path following LQR controller

Plots on how the velocity of the vessel, coupled thrust, and how coupled motor angle changed over time are shown in Figure 7.4.

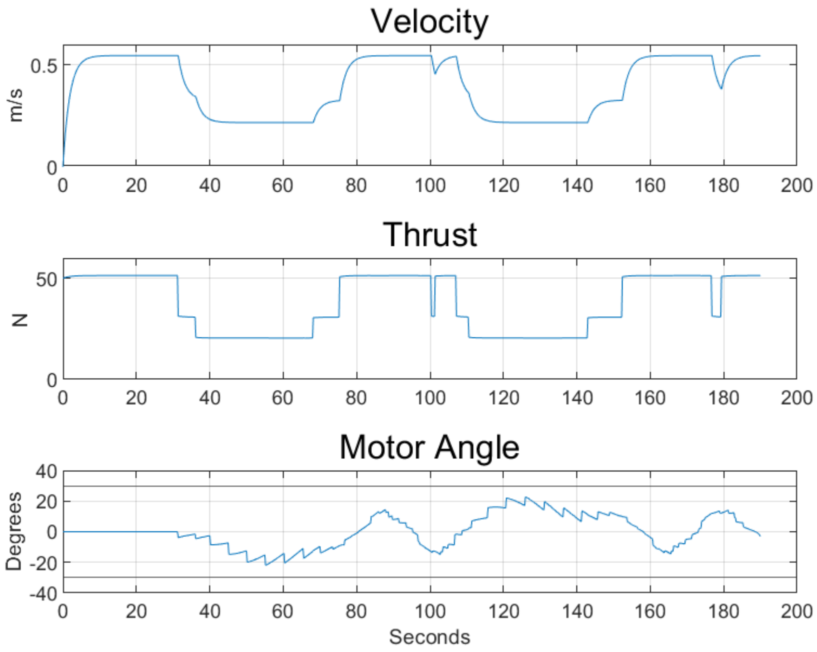


Figure 7.4. Simulation values

7.1.3 Near-Field

Ultrasonic sensors

The ultrasonic sensor readings were rather consistent and accurate, but could occasionally receive some noise (approximately ± **10 mm**) within the distance of 1800 mm. It was also found that the ultrasonic sensor needed to be about 30 cm above the a flat surface (such as the water) to not get false inputs.

Load cells

Load cells were accurate compared to the scale used for verification to around approximately ± **0.01 kg**.

Bumper

The bumper prototype worked well. It didn't get locked and therefore confirmed the design. Since the linear movement of the rubber blocks is very small, about 2 mm, the force from the springs in the most compressed state is 0.4 N each, 0.8 N on each block. This affects the load measured on the load cells, but since it is small compared to other forces during the docking it was overlooked.

Control loop

The simulated boat model also performed well after some tuning. The control parameters can be found in Table 7.2. These was used as a starting point for the tuning of the system on the actual boat. All graphs and plots indicated a stable loop even with some simulated disturbances.The simulated graphs of the docking can be seen in Figure 7.5, both with and without disturbances.

K_pPos	K_pVel	K_iVel	K_pAlpha
0.012	90	8	80

Table 7.2. The control parameters in the simulated control loop

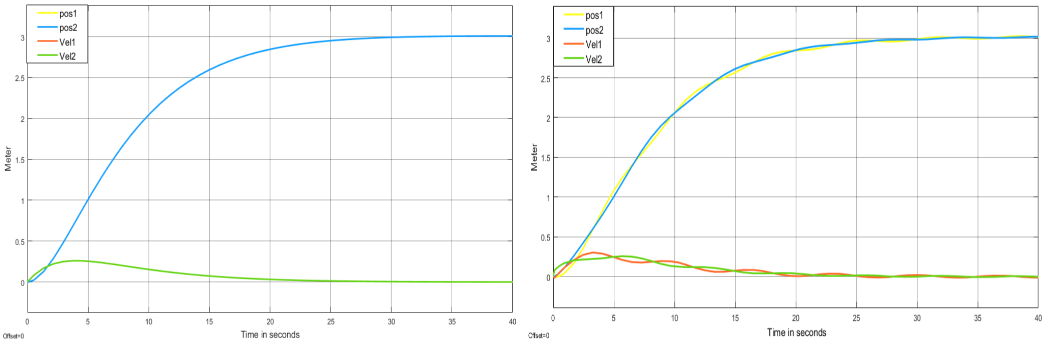


Figure 7.5. The graph to the left is the simulated position and velocity during the docking. The graph to the right is is the same but with a 10 degree start angle and a square wave disturbance on the feedback. As can be seen on the right graph the position follows the square wave but is not unstable.

7.1.4 Obstacle Detection & Path Planning

Mapping accuracy

From the Mapping Accuracy test the collected data can be seen in Table 7.3. Where tape measurements are in the left column, the measured distances of the Gmapping package in the middle and the registered distances from the depth images to the right. The mapping performance started to decrease at around **9 m**, resulting in there not being a result for the mapping at **15 m**.

Measured Distance [m]	Gmap Distance [m]	Depth Image Dist
2	2.05	2.05
3	3.1	3.15
4	4.1	4.05
5	5.15	5.03
7	6.85	6.95
9	8.74	8.9
11	9.3	10.9
13	9.2	12.88
15	-	14.8

Table 7.3. Collected distance data from tape measurements , Depth Images and the Gmapping package

Path planning

The Dubins path generation was timed as outlined in Section 6.1.4, the test resulted in an average time of 0.14 sec with a variance of ± **0.01 sec**.

7.2 Integration tests

This section cover the results of the integration tests of systems and sensors.

7.2.1 Stepper motor and encoder integration

After several iterations of testing, it was found that the following stepper motor control parameters gave the best balance between response, stability, and smooth operation.

- Stepper driver frequency = 625 Hz
- microstepping = 4 (800 pulse/rev)
- Error tolerance = 1.2 °
- Actuation time (full range) = 0.411 +/- 0.013 seconds

Although this is slower than initially calculated, anything faster was too aggressive and produced too much vibration that risked damaging the assembly and/or lead to instability with the controller. Regardless, further simulations and system level testing showed that this speed was more than responsive enough for our purposes.

7.2.2 Manual override controller integration

The implementation of the controller for manual override worked well for its designed purpose; to provide basic maneuverability to and from testing starting positions. The software interrupt induced when pressing down both joysticks worked as intended and instantly disables all CAN angle and thrust readings and instead reads input from the joysticks. This happens reliably regardless of the test being conducted and the current messages being received. The vessel also returns to autonomous mode as intended if the interrupt is raised again.

There was a hardware issue with one of the joysticks where the internal potentiometer for reading the x-axis was broken; instead of transmitting an analog value between 0-1023, it varies between 0-790. This was handled by disabling the one joysticks x-value reading and only reading thrust from the y-value, while reading angle reference from the second joystick.

7.2.3 Path following

The sensors worked great together with the controller implemented in MATLAB. It gave reasonable outputs together with hard-coded references for both thrust and bearing. However, the actuation of the two motors were poor, due to some unknown delays.

7.2.4 Near-field integration

The bumper worked well and the force was transmitted to the load-cells correctly. Also, the ultrasonic sensors worked well and everything was running in parallel on the development board. When trying to send information over CAN, the code would occasionally stop running. After some troubleshooting it was found to be caused by a CAN notification interrupt which was disabled. The communication then performed as intended.

7.3 System tests

This section cover the results of the tests of the whole system working together out on the water.

7.3.1 Boat Parameters

The results from the test cases outlined in Section 6.3.1 can be seen in Table 7.4. These were later used to inform the dynamical model and simulation being used for the path following controller.

Test Case	**Average Result**	**Note**
Acceleration time to 1 m/s	1.4 s	
Acceleration time to 1.8 m/s	6.4 s	
Braking time from 0,4 m/s	2.2 s	Cut-off at 0.1 m/s
Braking time from 0,8 m/s	3.9 s	Cut-off at 0.1 m/s
Braking time from 1,8 m/s	8.0 s	Cut-off at 0.1 m/s
Braking dist. from 0,4 m/s	0.58 m	Cut-off at 0.1 m/s
Braking dist. from 0,8 m/s	2.35 m	Cut-off at 0.1 m/s
Braking dist. from 1,8 m/s	4.63 m	Cut-off at 0.1 m/s
Stopping time from 1,0 m/s	27.7 s	Cut-off at 0.2 m/s
Stopping time from 1,8 m/s	34.2 s	Cut-off at 0.2 m/s
Stopping dist. from 1,0 m/s	11.41 m	Cut-off at 0.2 m/s
Stopping dist. from 1,8 m/s	15.03 m	Cut-off at 0.2 m/s
Turning radius at 1,4 m/s	3.8 m	
Turning radius at 1,8 m/s	4.1 m	

Table 7.4. Results from initial testing in the water.

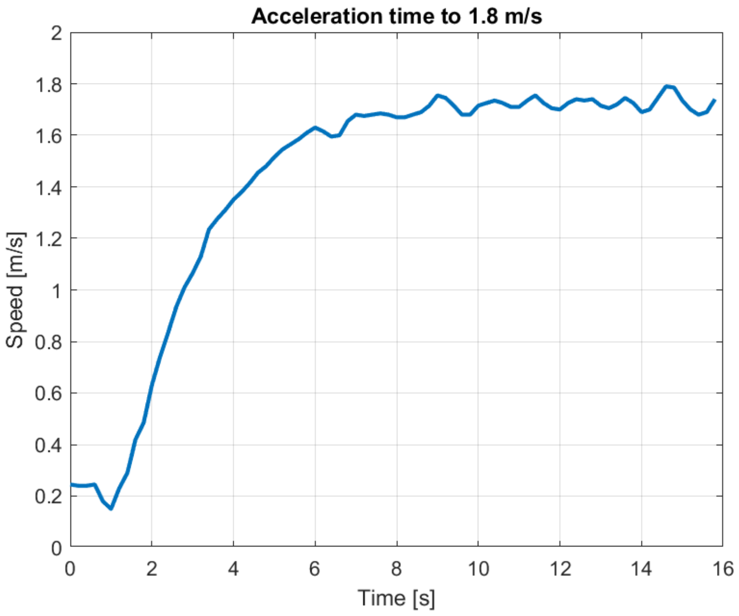


Figure 7.6. Average acceleration time to 1.8 m/s

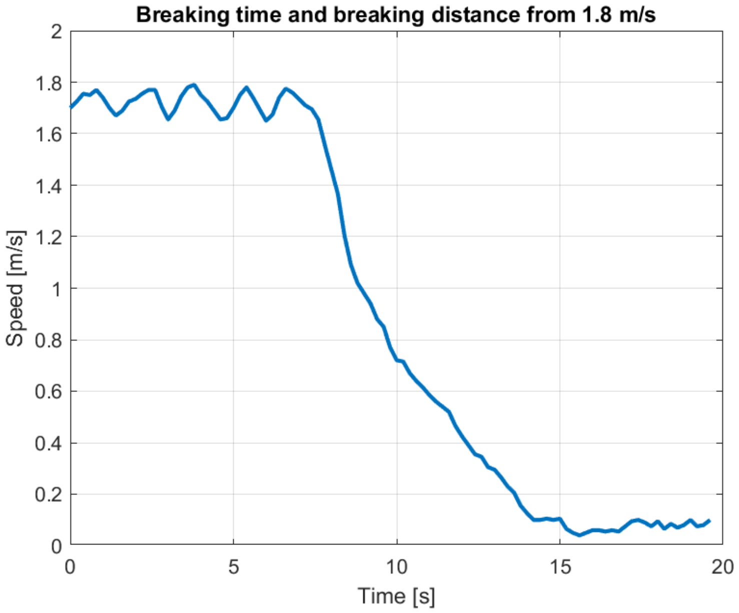


Figure 7.7. Average breaking distance and breaking time from 1.8 m/s

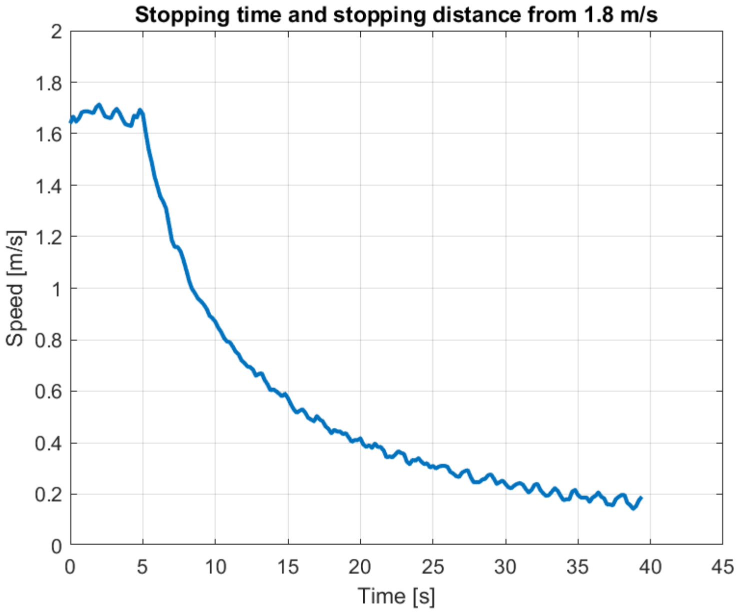


Figure 7.8. Average stopping distance and stopping time from 1.8 m/s

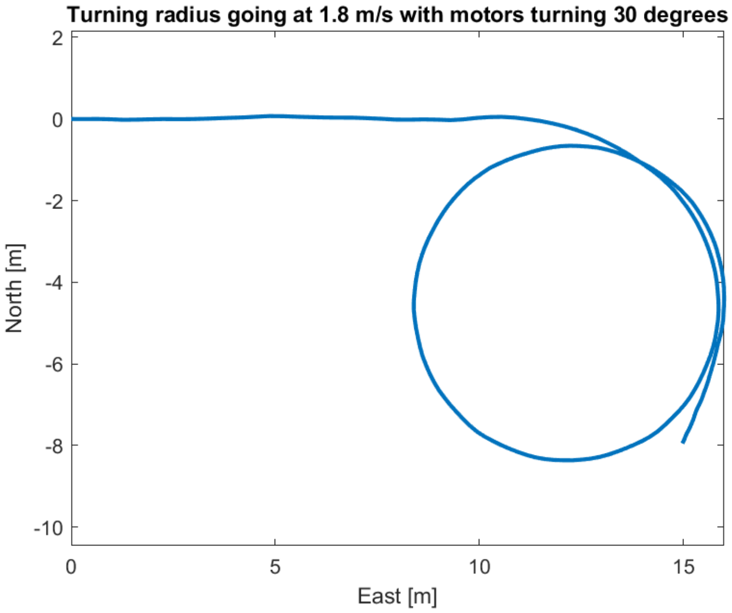


Figure 7.9. Turning radius going at 1.8 m/s with motors turning 30 degrees

7.3.2 Near-field docking

The different parts of the control were verified in steps. The velocity control performed well, also the angle control performed very well. The position control did not work as expected. One thing noticed was that the velocity calculated from the ultrasonic sensors were noisy and hard for the controller to follow. To solve this issue quickly, the velocity from the GNSS was used. This new controller still did not work properly and the final result was to skip the position control and hard-code the velocity reference. The velocity reference was divided into three stages correlating to the distance of the dock, as seen in Table 7.5.

Distance to dock [m]	Velocity reference [m/s]
3.5 - 1.2	0.3
1.2 - 0.5	0.2
0.5 - 0.1	0.05

Table 7.5. The velocity reference for different distances to the dock.

The final tuning of the velocity control and angle control gave the control variables as follows in Table 7.6.

K_pvel	K_ivel	K_palpha
70	5	30

Table 7.6. Final determined control gain parameters

The final control parameters are a bit lower than the simulated ones, see Figure 7.2. That was expected since there are large delays in the actual system and also there are much more disturbances in the real world.

The resulting speed of the boat during the docking tests can be seen in Figure 7.10. The boat contacted the dock about 20 seconds in to the test and was stationary after that. The average dock impact speed was measured to **0.05 m/s** with some runs as low as 0.03 m/s.

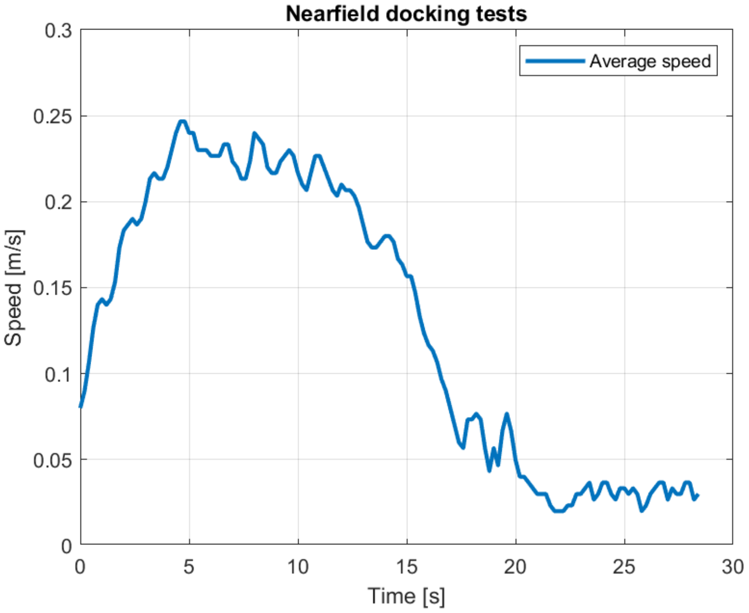


Figure 7.10. Average speed between 3 near-field test rounds.

7.3.3 Model Evaluation

The evaluation was done by comparing the step responses of the vessel models with the real data. This was done in both the surge and yaw direction. Conducting the step responses, using maximum thrust, of the two vessel models was done in MATLAB. In Figure 7.11, the response of the first one is shown. Looking at it, it can be seen that the dc-gain of the system is $\mathbf{k} \approx \mathbf{0.55}$ m/s with the time constant being $\tau \approx \mathbf{180}$ s. Looking at Figure 7.12, the response of the second model is shown together with the data logged from the ”Acceleration time to 1.8 m/s” test. For this model, it can be seen that the dc-gain of the system in $\mathbf{k} \approx \mathbf{1.7}$ m/s whereas the time constant being $\tau \approx \mathbf{1.9}$ s. This was then done in a similar manner in the yaw direction, but only for the second vessel model since the results for the the first model was intuitively unrealistic. This resulted in the dc-gain of $\mathbf{k} \approx \mathbf{2.9}$ m/s and time constant $\tau \approx \mathbf{5.9}$ s.

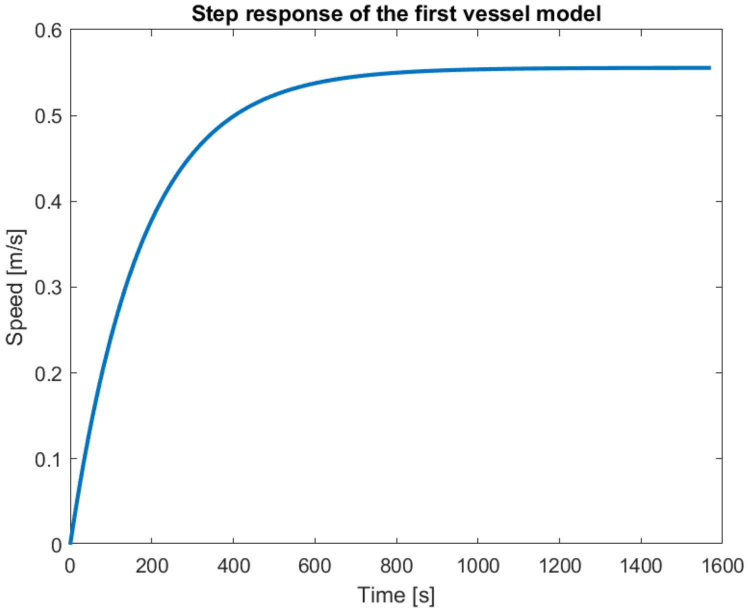


Figure 7.11. Step response using maximum thrust of the first vessel model

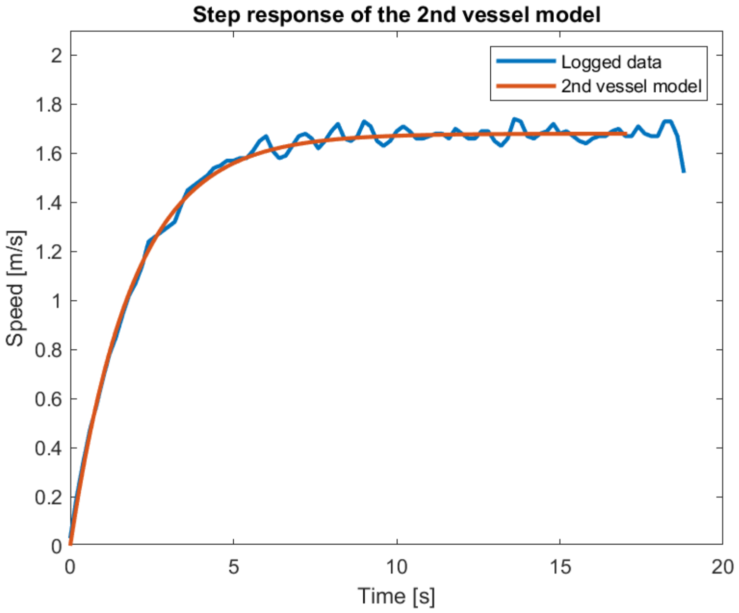


Figure 7.12. Step response using maximum thrust of the second vessel model

7.3.4 Positioning and Control

The testing of the path following control was done in two ways. First, it was tested for following a path consisting of a few waypoints. Then, it was tested against a path consisting of more waypoints, approximately ten times more.

Path Following of few waypoints

Shown below in Figure 7.13 - 7.15 are paths generated by the vessel following a small set of waypoints. As seen in the plots there is quite a bit of variance, but it is consistently following the predetermined path.

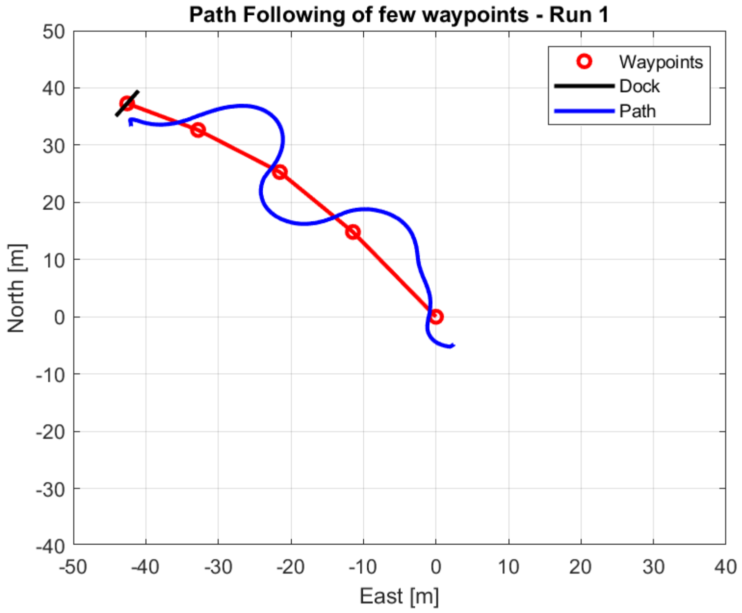


Figure 7.13. Path following of a few waypoints - Run 1

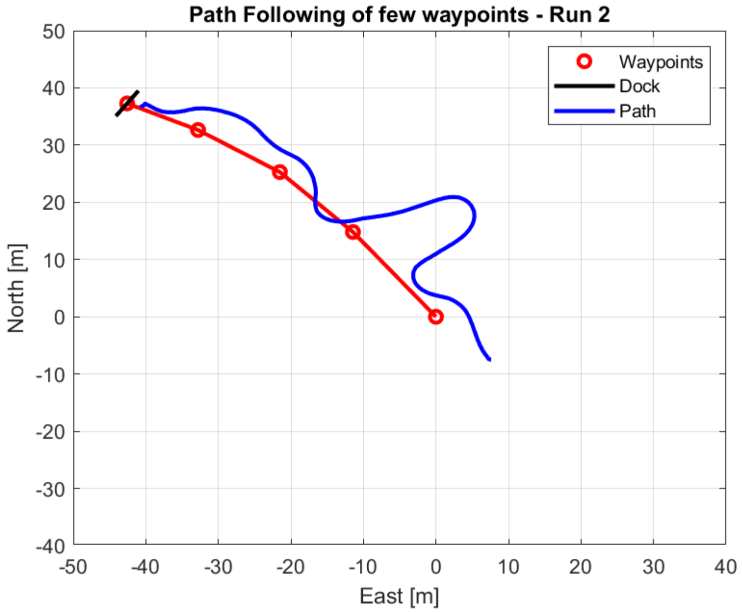


Figure 7.14. Path following of a few waypoints - Run 2

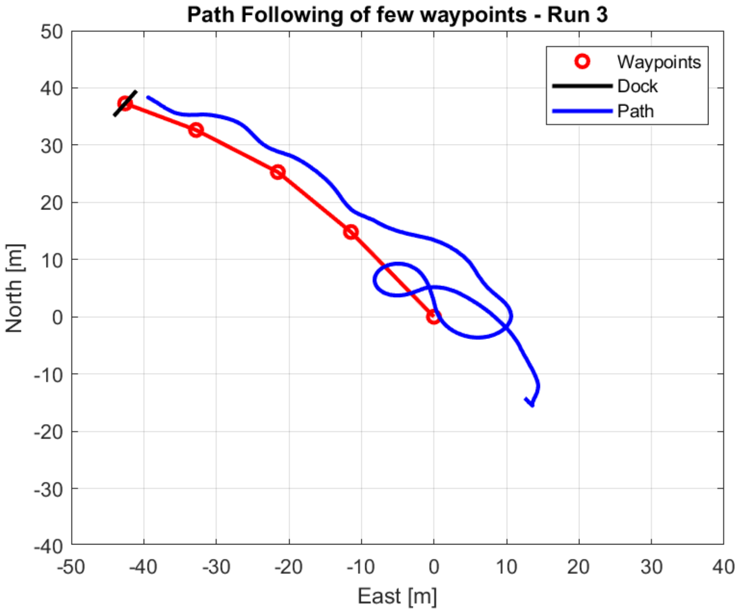


Figure 7.15. Path following of a few waypoints - Run 3

Path Following of Path Generated with Dubin's Path

Similar to the above case with a small set of waypoints, calculating a path using Dubin's Path with roughly 50 waypoints give similar inconsistencies. However the algorithm is causing a lot less overcorrecting and the path taken is much more smooth. This is shown below in Figure 7.16 - 7.18.

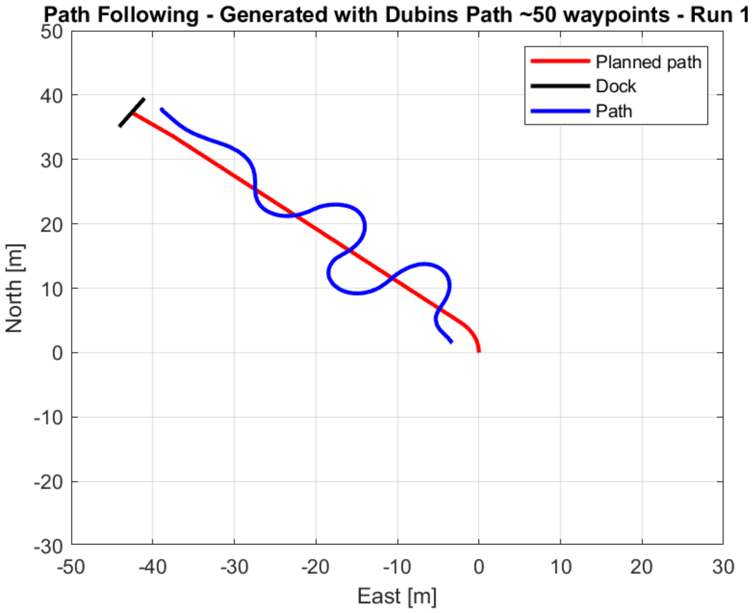


Figure 7.16. Path following of path consisting of approximately 50 waypoints, generated with Dubins Path - Run 1

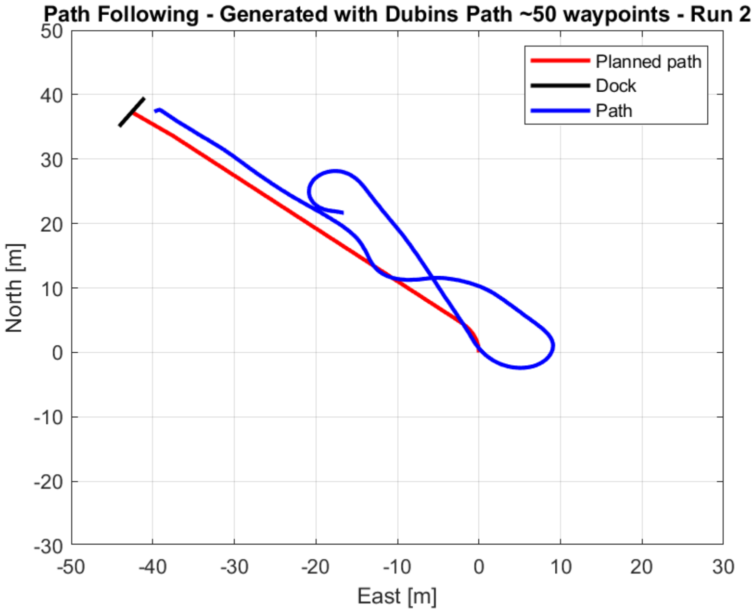


Figure 7.17. Path following of path consisting of approximately 50 waypoints, generated with Dubins Path - Run 2

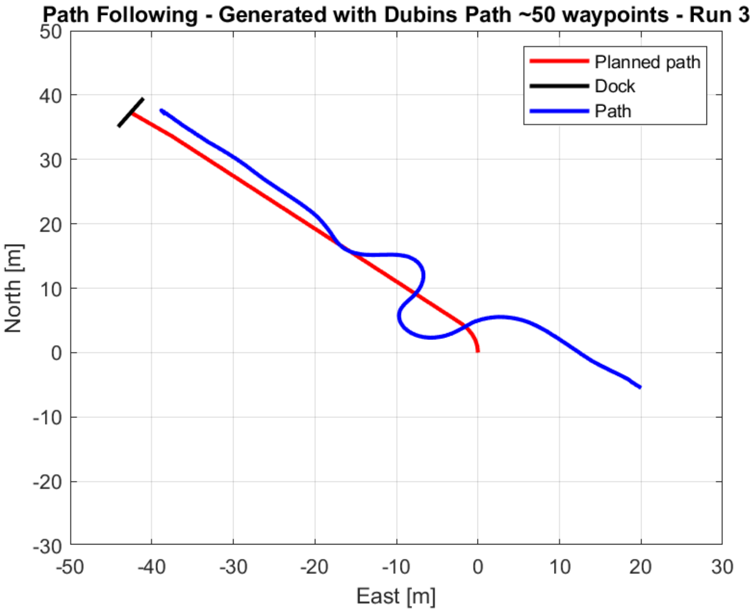


Figure 7.18. Path following of path consisting of approximately 50 waypoints, generated with Dubins Path - Run 3

7.3.5 Environment mapping

In this section the results from the mapping verification and validation will be disclosed. The mapping of a hallway in Brinellvägen 83 can be seen in 7.19. As a ground truth to mapping the floor planning of Brinellvägen 83 can be seen in 7.20.

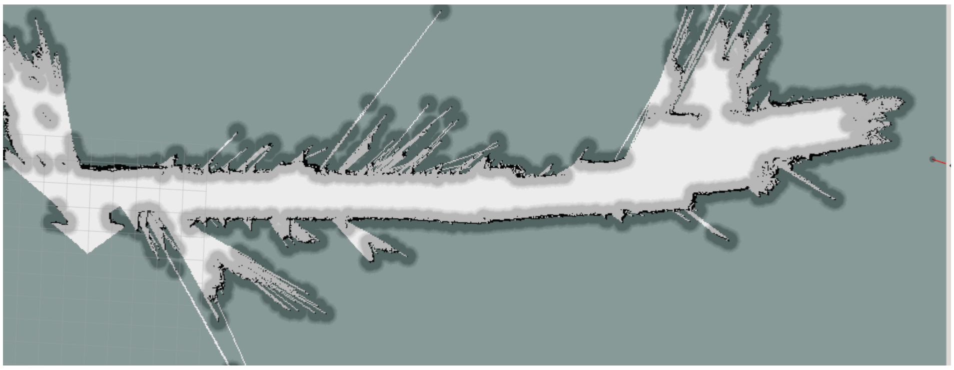


Figure 7.19. Mapping of a hallway at Brinellvägen 83.

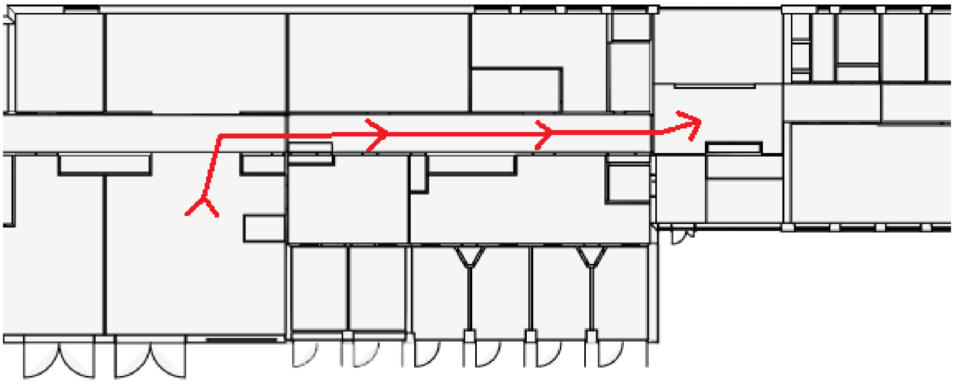


Figure 7.20. The floor planning of Brinellvägen 83. With the path walked in Figure 7.19 marked out.

After conducting the water mapping test, the result was compared to the satellite map of the area as seen in Figure 7.21.

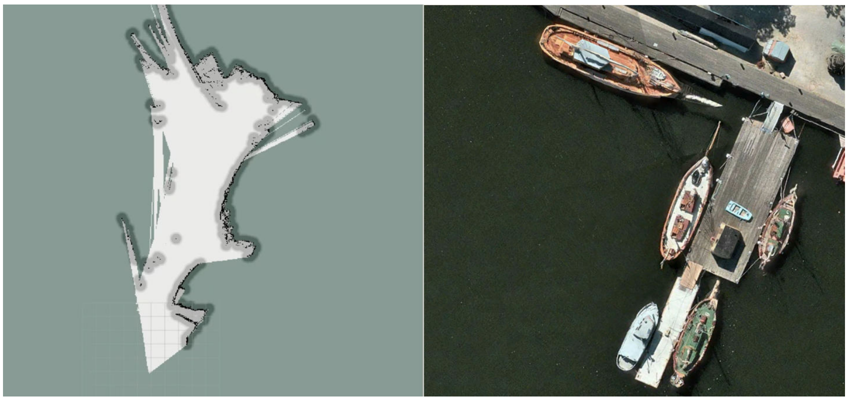


Figure 7.21. Mapping of the test location at Skeppsholmen. With the Gridmap to the left and a satellite image (Apple Maps) on the right.

7.4 Fulfillment of requirements

This section will cover the stakeholder requirements and technical requirements and classify if they were achieved or not.

7.4.1 Stakeholder requirement analysis

The stakeholder requirements outlined in Section 1.3.2 were evaluated based on what the group has achieved. The results can be seen in Table 7.7.

Requirement	**Fulfilled?**	**Note**
Purchasing	Yes	Extra expense approved
Dual Outboard Motors	Yes	
Dock Design	Yes	
Communication Protocol	Yes	
Autonomous Navigation	Partly	Path following without full system integration
Docking Time	Partly	No complete run, but sum of all stages
Dock Departure Protocol	No	Not attempted
Abort Signal	Yes	With manual controller
Environmental Obstacles	Yes	Objects are detected but no classification is done
Depth Sensing	No	Extra scope
Lighting Conditions	No	Camera not robust against dark conditions
Drivetrain Environmental Robustness	No	Extra scope
Sensor Environmental Robustness	No	Extra scope

Table 7.7. Stakeholder requirements and if they were fulfilled or not.

7.4.2 Technical requirement analysis

The technical requirements outlined in Section 1.3.2 were evaluated based on what the group had achieved. The results can be seen in Table 7.8.

Requirement	**Goal**	**Fulfilled?**	**Note**
Project Administration			
Budgetary Approval	50 000 SEK excl. tax	Yes	51 000 SEK, extra expense approved
System Portability			
Dock Form Factor	Flat wall, no guides	Yes	
Dock Material	Rubberized surface	Yes	
No Dock Modification	No excessive modification	Yes	
Motor Rotation Range	± 30 degrees	Yes	
CAN Bus Protocol	Main communication protocol	Yes	
Autonomous Protocol			
Navigational Control in Nominal Conditions	Operate with minimal error	Yes	
Dock Impact Speed	No more than 0.03 m/s	Partly	Achieved in some cases, avg. of 0.05 m/s
System Range	Active from 50 m	Yes	
Dock Accuracy	±0.5 m from goal position	No	Not able to test the full system
Docked Angular Movement	±5 deg. from predefined angle	Yes	
Dock Slipping	No slipping when docked	Yes	
Docking Time Allowed	Less than 1 min.	Partly	No complete run, but sum of all stages
Departure Heading Correction	Leave at predefined angle	No	Not implemented
Manual Abort Timing	Manual abort in less than 0.1 s	Yes	With hand controller
Navigation in Poor Environmental Cond.	Robust against extreme environmental cond.	Partly	Handled wakes from ferries okay

Table 7.8. Technical requirements and if they were fulfilled or not.

Requirement	**Goal**	**Fulfilled?**	**Note**
Obstacle Detection and Avoidance			
General Abort Detection	Stop for large object in path	No	
Depth Induced Navigational Abort	Stop if depth is < 0.7m	No	Extra scope
Avoid Other Boats	Detect and avoid other boats	No	Extra scope
Detect Swimmers	Detect and avoid swimmers	No	Extra scope
System Robustness			
Lighting Condition Robustness	Operational in various daytime conditions	Partly	
Lighting Recognition	Identification of lighting condition	No	Extra scope
Additional Lighting Conditions	Set up to handle all lighting conditions	No	Extra scope
Sensor Environmental Robustness	Robust against all weather conditions	No	Extra scope

Table 7.9. Continuation of Table 7.8, technical requirements and if they were fulfilled or not.

Chapter 8

Discussion and Conclusion

This chapter will cover a discussion of the obtained results and what could have been done differently. It will also discuss the ethical and environmental impacts of the project.

8.1 Discussion of test case results

The vessel is designed using two stern mounted motors as per the stakeholder requirements. This resulted in that changing the heading of the boat is a very difficult task; causing a lot of over-correction when trying to maintain a set path. It would have been very helpful to implement a bow thruster to aid with this. By being able to apply a lateral force to the front of the boat, turning would be a lot more responsive. This would reduce the turning radius of the boat significantly, allowing the control algorithm to much more easily and quickly remove a positional error from the path it is trying to follow. Having a bow thruster would also simplify correcting for wind and current forces acting on the boat by a large margin. Looking back at the State of the art in Chapter 2, companies working towards autonomous docking, such as Yanmar and Volvo Penta, often make use of bow thrusters in order to be able to not only correct heading in an efficient way, but also to allow sideways docking.

8.1.1 Drivetrain

When testing the trolling motor thrust, it was surprising how the max values were greatly lower than what was advertised by the manufacturer. With that being said, this static testing of the motor is knowingly not the most accurate way to achieve the appropriate thrust values; the velocity and direction of the fluid flow entering the propeller also plays a factor. To properly simulate the thrust being generated by the motors, specific motor parameters would need to be known as well. This was requested from the manufacturer, but not given. Not to mention the that the luggage scale was not the most accurate tool in determining thrust. Never the less, this rough approximation was deemed sufficient for our particular application due to the low speeds and the overall control algorithm that should compensate for discrepancies in the error between the actual and theoretical thrust values.

The manual override controller proved to be a very useful implementation to ease the testing phase. It was not in the initial scope but was added in later when we realized it wasn't optimal to use the oars for the boat to row around when going to and from start/end points. Because it was implemented on such a short notice, its functionality is very basic, only providing control to thrust and steering. Perhaps the controller should have been in the initial scope of the project to ensure a more proper implementation of it. This would have been useful as it would have optimized the limited time for in water testing, allowing us to further improve verification and validation of the system as a whole.

8.1.2 Obstacle Detection and Path Planning

Concerning mapping accuracy, the test proved good performance in the data directly from the depth image, with deviations of 1-5%. When the data was transferred in to the SLAM map the error increased, especially at longer distances with deviations up to 30%. The reason is not entirely clear why this happened and is something that should be explored further.

Moving on to the path planning test the results showed a fast and very reliable generation of paths to the goal. This allows subsystem to re-plan quickly and send new paths, or stop messages to the control system.

For the indoor mapping, the resulting accuracy was considered quite good. The system properly detected walls and open spaces without many false-positives. In the mapping tests out on the water, the performance was less stable. If the stereo camera had plenty of clear obstacles in the FOV, the performance was similar to the indoor test. If the system was activated in open water however, the mapping resulted in many false-positives and shaky performance. This likely stems from the nature of SLAM mapping where the mapping algorithm simultaneously tries to map it's surroundings and localize itself in that map. If there is nothing to detect, the system struggles.

8.1.3 Positioning and Control

The positioning and control system worked reasonably, though it was far from optimal. When evaluating the model, it was clearly seen that the initial model was way off. Though a lot of time was put into creating it, something must definitely have gone wrong. It is unknown if this and issue from the overall approach or a simple mistyping somewhere. The step response created, using system identification, seems to be following the logged data better. When using it for control design though it is assumed to behave linearly, which is most likely not the real-world case.

One major issue with the control design, which needs to be discussed, is the control allocation. To increase performance, the control allocation problem definitely needs to be more optimally computed, and not approximated using a few linearizations. Solving it using optimization was not done here mainly due to it being a very hard problem to solve in this scenario due to the over-actuated thruster configuration (to many variables, not enough unknowns). Other better ways of solving the control allocation would have been to linearize it in real-time or to linearize it around more working points for the use of gain scheduling. Also, interpolation could be used for the gain scheduling to possibly increase performance.

The varying results in positioning could also be due to varying performance with delay in actuation of the motors of up to 1000 ms. This could be due to some issues with the CAN communication not identified. The path following controller was developed to work with a continuously updated waypoint list. However, when sending 50 waypoints over UART without any type of acknowledgement, which means that the devices does not actually know whether the transmission was received or not, half of the times, the first 15-30 waypoints were sometimes lost. The long list of waypoints also required a lot of computational power and could be the reason for the sampling of the controller being higher than anticipated. This might have been resolved by using either a single, more powerful micro controller for implementation of both the drive train and path following controllers or by using a different communication protocol for sending the path planned coordinate list, for example I^2C, in order to use a master-slave communication which requires acknowledgment after each byte sent. This would ensure no missed transmissions. We could also have limited the waypoints transmitted and only sent the next 10-15 points of the list, which would be the 10-15 meters ahead of the boat. The tests with waypoints generated with Dubins Path was unfortunately never tested in real time on the controller, only with a hard-coded waypoint path, due to the performance issues presumably caused by this communication. This would need to be addressed in order to verify and validate the controller properly.

Another issue was the unreliability of some of the magnetometer readings. There was some detected electromagnetic interference from the buck-converter that was placed close to the magnetometer and possibly from other electronic sources. The metal hull may have radiated noise as well that could could have interfered with the magnetometer readings. One way this could have been tackled would have been to place the magnetometer further away from the buck-converter which was particularly in close proximity. Another solution to this issue could have been to ”clean” the sensor readings through the use of either a Kalman filter or some other type of sensor fusion in order to reduce the noise in the readings, perhaps by utilizing an IMU.

As seen in the results, the motion of the vessel is sometimes oscillatory. This is most visible in the first run, shown in Figure 7.13. The vessel starts by overshooting. Then, the controller tries to compensate for this throughout its journey to the dock. It looks like the system is sometimes not able to steer fast enough for it to align the vessel with the path. If this is mainly caused by any of the individual issues or a combination of them, is not known. As seen early in the run in Figure 7.14, there is some issue where the vessel starts heading north-east. What is causing this is not known, but since it looks like the vessel is not even trying to correct itself, this could be due to either miss-readings from the magnetometer, thinking we are on the correct path, or something blocking the code responsible for actuating the motor (either in the motor control or the transmission of actuator commands). One issue with the testing, which is clearly seen in Figure 7.15, is the difficulties of finding the correct starting position. This also seems to be an issue with the testings when following a hard-coded path generated using the Dubins path algorithm. As previously mentioned, the path was supposed to be updated repetitively. This was not able to be tested, hence we are not able to know if the oscillatory behavior, as seen in Figures 7.16 and 7.18, would have been an issue then as well. What can be noted though is that during the path following testing of the path created using Dubins path, which consists of about ten times more waypoints then the first path, the motion seems to converge faster towards the path and contain less oscillations. This is especially clear in Figure 7.18. Though in these tests, the controller seem to have a steady state error.

8.1.4 Near-Field

During testing, it was clear that some things did not work as expected. The position controller did not work and to get around that, the reference velocity was hard coded in specific intervals. The highest updating frequency was limited to 2 Hz, which is very slow. The control loop of near-field was running on the microcontroller responsible for control (in the middle of the boat). In order to run the loop, the near-field system sent the latest sensor input over CAN-bus to the control board. However if the near-field system sent data faster than 2 Hz the control processor never had time to execute the instructions to the motors. In order to increase it's update frequency and get the control loop to perform better, the control loop was implemented on the Near-Field microcontroller, but there was never time to try the new set up.

Also there were some issue with the velocities calculated from the ultrasonic sensors. The distance received was a bit noisy. Since the boat moves between 15 cm and 2 cm between each reading during the docking process (with a frequency of 2 Hz), even a small noise of about 10 mm has a large impact on the velocity. This caused the velocity calculated from the ultrasonic sensors to vary a lot. To get a steadier input velocity, the velocity readings from the GNSS was used instead. This worked surprisingly well. The delay is larger in the GNSS and it could have cause the system to become unstable, but did not.

In the end, the over all performance of the near-field system was sufficient. The ultrasonic sensors were reliable at detecting the dock and not picking up waves or other objects. The bumper transmitted the force to the load-cells and were robust enough to handle the forces at the dock. With the current bumper design, there is a risk of overloading the load cells. The plan was to start testing without load cells and than rely on the robustness of the load cell and risk damaging them. In the testing phase the time was short and all tests were done with the load cells instead. This ended up being not a problem and the load cells handled the impacts well.

The biggest issues arose in relation to updating frequency. For example, when the boat hit the dock the load cells got an input which was sent via CAN to the control board. However, for the signal to be recognised, multiple signals needed to be sent. Due to the tendency of bouncing slightly towards the dock it was important for the boat do go into docking mode immediately once in contact, but since the new input was not acknowledged by the control loop, it resulted in the boat bouncing a few times before going into docking mode.

Another issue was the robustness of the ultrasonic sensors. When the boat had been outside in lower than -10 degrees Celsius for a couple of hours, the sensors started to send lengths of about 4 cm while a couple of meters from the dock. The sensors were rated to work down to -15°C, but no other explanation could be found.

8.2 Reflections upon results

This project has been all about investigating a way of completing autonomous docking and to discover our approaches strengths and weaknesses. In practice, all subsystems where tested and validated independently, they were not however fully combined in to a fully integrated system that could achieve docking from start to finish.

In the end, not all technical requirements were met in this project. This can be attributed to a large initial scope and short time-span of the project. Another large limiting factor in getting everything together was very harsh testing conditions. The group tested the boat during two weeks in December in Stockholm, with temperatures consistently below -10°C.

That being said, the group is very satisfied with the general approach taken in the concept phase of the project. The assumption that the coordinates of the dock are known limits many of the uncertainties in dock localization, and is a reasonable assumption considering the ferries are going to travel along a predefined route.

The decision to use RTK-corrected GNSS for localization also makes a lot of sense given the amount of base stations in the and because it reduces the need for other localization methods.

8.3 Ethics

At this early prototyping stage, not a whole lot of ethical issues arise. The issues that could be elaborated on are the possible effects an autonomous boat implementation would have in the future.

One possible effect an implementation like this would have is that current jobs could be completely replaced by this autonomous system. A lot of discussion has been done on the paradox of automation and whether it is a good or a bad thing that industry jobs are being replaced by autonomous systems. It may be true that this system, when refined, could replace the jobs (such as a ferry captain), it would also create more jobs associated with it. Jobs such as automation engineers and maintenance for the system. It is also worth reiterating that this project scope is only aimed at the docking sequence; full automation is still far away.

Another ethical issue that needs to be addressed is the possible danger an autonomous boat would have. Extensive testing must be performed to ensure that this boat operates as planned. If this is not ensured, the boat could become a huge danger for both the passengers and the people around it. A matter of debate is how do you decide if the boat is safe enough? Is the boat safe enough when the fatality rate is lower than with a human operator? Or does it have to be significantly safer than a human?

8.4 Sustainability

This prototype would have a relatively low environmental impact in comparison to the fossil fuel counterparts currently being used for the SL ferry system. Mainly since this boat is fully electric, but also since its autonomous. The electrification reduces the emission during operation compared to a petrol driven boat. In this project, the material choices for the prototype were not chosen with regard to environmental impact in mind. In further development of this product, the material choices are definitely something that should be more thoroughly looked at. It is also important that the product not only have a low impact on the environment when operating, but also during the production and end of life.

8.5 Conclusion

Even though not all goals of the project were achieved, both the project group and the stakeholder has been very happy with the work that was successfully completed. It has worked as a pathfinder towards an approach for autonomous docking, grounded in the state-of-the-art and ending in a concept developed in a structured way. It has also provided the stakeholder with a hardware platform for continued testing and autonomous boating development.

Chapter 9

Future Work

This chapter will discuss areas of future development that could make the system more robust and technologically advanced, both as a full prototype and at the subsystem level.

9.1 Overall system improvements

Some efforts have already been made in making the system water resistant, but these were in no way up to the standards or IP65 or similar. With a larger budget, properly sealed and ventilated enclosures, better connectors, and more insulated wiring could be used instead that would increase robustness to the climate. Although this wasn't necessarily an issue during testing, longer time spend out on the water would most likely cause some issues.

Another area that the entire system could improve upon is a deep analysis of the threading and delays within the various code for the entire system. Although some care was taken to improve system response at each level of integration, the debugger used did not supply advanced visualization tools for this context switching, so it was difficult to identify areas of improvement. The use of additional software and further analysis/tuning could improve the overall system response considerably.

A important aspect of this project that needs to be addressed is the state machine that switches between the path following, near-field control loop, and docking mode (when in contact with dock). The aim of the this project was to realize a system that successfully switches between these in real-time to complete the overall mission of autonomous docking, but thus far testing was only conducted on these 3 states independently. It would therefore be highly encouraged to get this system working together as a cohesive unit. Original plans were to include a state for departing the dock as well, so this hopefully could be implemented as well.

9.2 Near-Field

To improve the near-field system there are multiple things that can be done. First, additional testing of the system with the control loop and sensor input on the same microcontroller to increase update frequency of the control loop. Then, also investigate the implementation of a low-pass filter to improve the velocity calculated from the ultrasonic sensors. That may solve the noisy behaviour of the position control. Otherwise, also try adjusting the controller to include a functioning position control. It would also be good to implement some sort of improved safety against overload of the load-cells.

To complete all the requirements that was set from the start, there are a number of things that need to be done.

- Update control loop to make motors independent of each other and allow for an angled thrust.
- Update the code for reversing the boat when docking is completed.
- Add additional ultrasonic sensors to side and rear of the boat to verify a safer docking and backing out.

9.3 Drivetrain

One area for future improvement is that the software currently has no way of assessing the prioritization level of the thrust and angle references that are sent over the CAN network. In the case that several references are sent over CAN at the same time, this would cause conflicts where both of the messages would update the thrust and angle references, effectively causing them to overwrite each other. Some type of system needs to be implemented that decides which subsystem's CAN message should be received and acted on.

For the motor assemblies, stiffening of the various components would allow for improved actuation speed by reducing vibrations induced by aggressive movements. This could be done in a number of ways. For one, stiffening the stern of the boat (the sheet aluminum currently flexes considerably). The shaft of the trolling motors are also quite thin and flex; custom made trolling motors with thicker shafts would be better. Also the sheet metal bracket mounting the stepper was seen flexing during slow-motion video, so redesigning this with thicker aluminum and/or another material could improve rigidity.

To further improve on the response of the stepper motors, varying the PWM based on a reference trajectory would be helpful. Currently, a constant PWM is being applied based on detected angle error, but modulating this based on better known dynamics could further increase angular acceleration while dampening as it neared the reference angle. This would also improve the vibrations mentioned in the last paragraph.

Better testing and simulation of the trolling motor thrust would also be helpful in generating proper input signals based on reference thrust. The current static testing does not fully capture the real-world dynamics, so a more realistic model could improve performance and also be used in overall system modelling. This would take a considerable amount of time and further research on propeller design as well as additional technical specifications on the trolling motors being used (these unfortunately were not made available to this project).

This next suggestion is quite minor, but the handheld controller that has been implemented to drive the boat manually might need some de-tuning since the current configuration is actuating the motors quite aggressively. Also, replacing the one broken joystick means more advanced maneuvers could be achieved by controlling the angle and thrust of one motor per joystick. More advanced maneuvers can also be realised with the addition of buttons on the controller.

9.4 Obstacle Detection and Path Planning

The stereo camera system could be more thoroughly tested and tuned to remove false-positives and optimize mapping performance. It could also be interesting to look into other non-SLAM mapping algorithms.

More advanced object detection based on machine learning could be implemented to make the system more intelligent than just looking for open space. This would make the system more robust against small objects like a person swimming in the water or a kayak, for example. It would also potentially increase the range of the system beyond the 20 meters of the depth mapping, by classifying objects further away.

In the beginning of this project, LiDAR was considered but in the end the stereo vision method was chosen over LiDAR. This was due to the short time frame and tight budget of the project. In further development it would be interesting to look into using a LiDAR for mapping instead. The LiDAR could provide a more accurate 2D map, but this is something that needs to be tested.

9.5 Positioning and Control

The waypoints generated using the Dubins Path Algorithm were unfortunately never tested in real-time with the path following controller. This was an important aspect of our approach and implementing this may have improved the system stability considerably. Therefore, one of the first suggested improvements to this prototype would be to determine the issues causing the waypoint transmission to be received unreliably and then implement the LQR path following controller using the generated waypoints in real-time.

Another thing to consider for future work is that the system needs to be more thoroughly tested to identify key performance issues. With a more powerful microcontroller, other more computationally heavy control algorithms could be tested. For example MPC could be used to compensate for the high overshoot in the system. With the current thruster configuration it was also hard to actuate the vessel in the sway direction (y-direction in the vessel's body-fixed frame of reference). For more maneuverability, implementation of a bow thruster should be considered. Furthermore, some kind of sensor for measuring forward and angular velocity should be implemented for more accuracy (such as an IMU perhaps). Sensors measuring disturbances such as wind and currents would help make the controller more robust against said disturbances if used in conjunction with some feed-forward control.

As for modelling, the initial theoretical model should be investigated to see what went wrong. Considering the theory used in this work was derived from generalized vessel design theory, it should be correct and be able to be used for our particular case as well. As for the controller, the main issue to improve would be how the control allocation problem was solved (i.e. optimization, iteration or real-time linearization).

Bibliography

[1] *18650 Batteries - Rechargeable 3.7V Li-Ion Cells - 18650 Battery Store.* 18650BatteryStore.com. URL: https://www.18650batterystore.com/collections/18650-batteries (visited on 05/24/2021).

[2] Oludare Isaac Abiodun et al. *State-of-the-art in artificial neural network applications: A survey.* 2018. DOI: https://doi.org/10.1016/j.heliyon.2018.e00938. URL: https://www.sciencedirect.com/science/article/pii/S2405844018332067.

[3] *Absorbent Glass Mat (AGM) Battery Information - Battery University.* URL: https://batteryuniversity.com/learn/article/absorbent_glass_mat_agm (visited on 05/23/2021).

[4] Saad Albawi, Tareq Abed Mohammed, and Saad Al-Zawi. *Understanding of a convolutional nerual network.* 2017. DOI: 10.1109/ICEngTechnol.2017.8308186. URL: https://ieeexplore.ieee.org/document/8308186.

[5] *An Introduction to Model-based Predictive Control (MPC).*

[6] *AS5048A - 14-bit rotary position sensor with digital angle (interface) and PWM output - ams.* ams. URL: https://ams.com/en/as5048a (visited on 12/18/2021).

[7] *Autopilot — Tesla.* May 22, 2021. URL: https://www.tesla.com/autopilot (visited on 05/22/2021).

[8] Andrew Baker-Campbell. *Fair Weather Friend: How Do LiDAR Systems Cope In Rain &amp; Fog?* May 2021. URL: https://www.autovision-news.com/sensing/sensor-technology/lidar-systems-rain-fog/ (visited on 05/19/2021).

[9] Alexander Bärlund et al. “Nonlinear MPC for Combined Motion Control and Thrust Allocation of Ships”. In: *IFAC-PapersOnLine* 53.2 (2020). 21th IFAC World Congress, pp. 14698–14703. ISSN: 2405-8963. DOI: https://doi.org/10.1016/j.ifacol.2020.12.1834. URL: https://www.sciencedirect.com/science/article/pii/S2405896320324472.

[10] *Basics of the I2C Communication Protocol.* Dec. 15, 2021. URL: https://www.circuitbasics.com/basics-of-the-i2c-communication-protocol/ (visited on 12/15/2021).

[11] Mogens Blanke. “Ship propulsion losses related to automatic steering and prime mover control”. PhD thesis. Lyngby, Denmark: Technical University of Denmark, 1981.

[12] José Miguel Almeida Braga. “Control of underwater vehicles on autonomous docking maneuvers”. MA thesis. Porto, Portugal: Facualdade De Engenharia Da Universidade do Porto, 2010.

[13] Robert Grover Brown. *Introduction to random signals and applied Kalman filtering : with MATLAB exercises.* eng. 4th ed.. 2012.

[14] Onshape Business a PTC. *Onshape — Product Development Platform.* URL: `http://www.onshape.com/en/` (visited on 12/18/2021).

[15] *CAN data frame structure caption.* URL: `https://www.researchgate.net/figure/CAN-data-frame-structure-caption_fig3_328607559` (visited on 12/18/2021).

[16] Howie Choset et al. “Potential Functions”. In: *Principles of Robot Motion: Theory, Algorithms, and Implementations.* 2005, pp. 77–106.

[17] R C Coulter. *Implementation of the Pure Pursuit Path Tracking Algorithm.* eng. 1992.

[18] *depthimage$_t$o$_l$aserscan.* Dec. 16, 2021. URL: `http://wiki.ros.org/depthimage_to_laserscan` (visited on 12/16/2021).

[19] Piyu Dhaker. *Introduction to SPI.* URL: `https://www.analog.com/en/analog-dialogue/articles/introduction-to-spi-interface.html` (visited on 12/19/2021).

[20] Anushka Dhiman. *YOLOv3 Object Detection in TensorFlow 2.x.* URL: `https://medium.com/analytics-vidhya/yolov3-object-detection-in-tensorflow-2-x-8a1a104c46a8` (visited on 05/23/2021).

[21] *Digital Stepper Driver 1.8˜5.6A 20-50VDC for Nema 23, 24, 34 Stepper Motor.* URL: `https://www.omc-stepperonline.com/digital-stepper-driver-18~56a-20-50vdc-for-nema-23-24-34-stepper-motor-dm556t.html` (visited on 12/18/2021).

[22] *Discord.* (Visited on 12/18/2021).

[23] L. E. Dubins. *On Curves of Minimal Length with a Constraint on Average Curvature, and with Prescribed Initial and Terminal Positions and Tangents.* The Johns Hopkins University Press, 1957. URL: `https://www.jstor.org/stable/pdf/2372560`.

[24] *Dubins-Curves.* Dec. 15, 2021. (Visited on 12/15/2021).

[25] *Dubins1 by Salix Alba.* Dec. 15, 2021. (Visited on 12/15/2021).

[26] *Endura Max 55 lb. / Hand - 36” — Minn Kota Motors.* URL: `https://minnkotamotors.johnsonoutdoors.com/freshwater-trolling-motors/endura-max?id=13726` (visited on 12/18/2021).

[27] O. M. Faltinsen. *Sea Loads on Ships and Offshore Structures.* 1990.

[28] Giuseppe Fedele et al. “Obstacles Avoidance Based on Switching Potential Functions”. In: *Journal of Intelligent & Robotic Systems* 90 (June 2018). DOI: `10.1007/s10846-017-0687-2`.

[29] Thor I. Fossen. *Guidance and Control of Ocean Vehicles.* John Wiley & Sons, Ltd, 1994.

[30] Thor I. Fossen. *HANDBOOK OF MARINE CRAFT HYDRODYNAMICS AND MOTION CONTROL.* Chichester, United Kingdom: John Wiley & Sons, Ltd, 2011.

[31] *FreeRTOS.* (Visited on 12/16/2021).

[32] Kshitiz Garg and Shree K. Nayar. “Vision and Rain”. In: *International Journal of Computer Vision* 75.1 (Feb. 2007), pp. 3–27. DOI: `10.1007/s11263-006-0028-6`. URL: `https://doi.org/10.1007/s11263-006-0028-6`.

[33] *Github.* (Visited on 12/18/2021).

[34] T. Glad and L. Ljung. *Reglerteknik : grundläggande teori.* 4., [omarb.] uppl.. Lund: Studentlitteratur, 2006. ISBN: 91-44-02275-1.

[35] T. Glad and L. Ljung. *Reglerteori: flervariabla och olinjära metoder.* Studentlitteratur, 2003. ISBN: 9789144030036. URL: `https://books.google.se/books?id=3sTsMAAACAAJ`.

[36] *gmapping.* Dec. 16, 2021. URL: `http://wiki.ros.org/gmapping` (visited on 12/16/2021).

[37] Carlos Gonzalez. *What’s the Difference Between Pneumatic, Hydraulic, and Electrical Actuators?* URL: `https://www.machinedesign.com/mechanical-motion-systems/linear-motion/article/21832047/whats-the-difference-between-pneumatic-hydraulic-and-electrical-actuators` (visited on 05/24/2021).

[38] Google. *Cloud GPUs.* URL: `https://cloud.google.com/gpu` (visited on 05/19/2021).

[39] *Google Drive.* (Visited on 12/18/2021).

[40] Giorgio Grisetti, Cyrill Stachniss, and Wolfram Burgard. “Improved Techniques for Grid Mapping With Rao-Blackwellized Particle Filters”. In: *IEEE Transactions on Robotics* 23.1 (2007), pp. 34–46. DOI: `10.1109/TRO.2006.889486`.

[41] *How a Stepper Motor Work.* URL: `https://howtomechatronics.com/how-it-works/electrical-engineering/stepper-motor/` (visited on 05/24/2021).

[42] *How a Strain Gauge Load Cell Works? What is a Load Cell? — FUTEK.* May 24, 2021. URL: `https://www.futek.com/how-a-load-cell-works` (visited on 05/24/2021).

[43] *How does a lithium-Ion battery work?* Let's Talk Science. URL: https://letstalkscience.ca/educational-resources/stem-in-context/how-does-a-lithium-ion-battery-work (visited on 05/23/2021).

[44] *How Servo Motors Work.* Sept. 29, 2020. URL: https://www.kollmorgen.com/en-us/blogs/_blog-in-motion/articles/how-servo-motors-work/ (visited on 05/24/2021).

[45] *How Stereo Vision works.* URL: https://ams.com/stereovision (visited on 05/23/2021).

[46] *How To Build A DIY Electric Bicycle Lithium Battery From 18650 Cells.* URL: https://www.ebikeschool.com/how-to-build-a-diy-electric-bicycle-lithium-battery-from-18650-cells/ (visited on 05/24/2021).

[47] Terry Huntsberger et al. *Stereo Vision-Based Navigation for Autonomous Surface Vessels.* Jan. 2011. URL: https://www.researchgate.net/publication/220648049_Stereo_Vision-Based_Navigation_for_Autonomous_Surface_Vessels.

[48] Siemens Product Lifecycle Management Software Inc. *Application Lifecycle Management (ALM), Requirements Management, QA Management — Polarion - Software.* URL: https://polarion.plm.automation.siemens.com (visited on 05/23/2021).

[49] Microchip Technology Inc. *MCP2561/2 Data Sheet.* URL: http://ww1.microchip.com/downloads/en/devicedoc/20005167c.pdf (visited on 12/14/2021).

[50] Princy A. J. *Automotive LiDAR Sensor: A 360-Degree Vision Eye for Autonomous Cars.* May 2021. URL: https://www.researchdive.com/blog/Automotive-LiDAR-Sensor-A-360-Degree-Vision-Eye-for-Autonomous-Cars (visited on 05/19/2021).

[51] Rovani Jean. *End-to-end Object Detection with Template Matching using Python.* URL: https://www.sicara.ai/blog/object-detection-template-matching (visited on 05/22/2021).

[52] Esaias Jerrelind. "Linear Quadratic Control of a Marine Vehicle with Azimuth Propulsion". MA thesis. Linköping University, Automatic Control, 2021.

[53] Karl Henrik Johansson, John Lygeros, and Shankar Sastry. *Modeling of Hybrid Systems.* UNESCO Encyclopedia of Life Support Systems (EOLSS). 2004.

[54] *Jonboatyoutube.* URL: https://www.youtube.com/watch?v=Rq44smZ6axg&t=139s (visited on 05/24/2021).

[55] Sarthak Katyal et al. "Object Detection in Foggy Conditions by Fusion of Saliency Map and YOLO". In: *2018 12th International Conference on Sensing Technology (ICST)*. IEEE, Dec. 2018. DOI: 10.1109/icsenst.2018.8603632. URL: https://doi.org/10.1109/icsenst.2018.8603632.

[56] *Key Differences Between Lithium Ion And Lithium Iron Batteries — EnergyLink*. URL: https://goenergylink.com/blog/key-differences-between-lithium-ion-and-lithium-iron-batteries/ (visited on 05/23/2021).

[57] Hassan K Khalil. *Nonlinear systems; 3rd ed.* The book can be consulted by contacting: PH-AID: Wallet, Lionel. Upper Saddle River, NJ: Prentice-Hall, 2002. URL: https://cds.cern.ch/record/1173048.

[58] *Kimple 330 Angler Wide — Båtar & Motorer — Jordnära Alingsås*. May 24, 2021. URL: https://jordnara.se/produkt/kimple-330-angler-wide/ (visited on 05/24/2021).

[59] Narayan V. S. K. Kovvali. *An introduction to Kalman filtering with MATLAB examples*. eng. Synthesis digital library of engineering and computer science. 2014. ISBN: 1-62705-140-6.

[60] Lantmäteriet. *Map support*. URL: https://swepos.lantmateriet.se/services/mapservice.aspx (visited on 05/23/2021).

[61] Lantmäteriet. *Network RTK*. URL: https://www.lantmateriet.se/en/maps-and-geographic-information/gps-geodesi-och-swepos/GPS-och-satellitpositionering/Metoder-for-GNSS-matning/Natverks-RTK/ (visited on 05/23/2021).

[62] Lantmäteriet. *Open data*. URL: https://www.lantmateriet.se/en/maps-and-geographic-information/gps-geodesi-och-swepos/swepos/swepos-tjanster/oppna-data/ (visited on 05/23/2021).

[63] P. Leite. "A Self-Guided Docking Architecture for Autonomous Surface Vehicles". In: 2019.

[64] You Li and Yassine Ruichek. "Occupancy Grid Mapping in Urban Environments from a Moving On-Board Stereo-Vision System". In: *Sensors* 14.6 (2014), pp. 10454–10478. ISSN: 1424-8220. DOI: 10.3390/s140610454. URL: https://www.mdpi.com/1424-8220/14/6/10454.

[65] *LiDARs for self-driving vehicles: a technological arms race*. URL: https://www.automotiveworld.com/articles/lidars-for-self-driving-vehicles-a-technological-arms-race/ (visited on 05/23/2021).

[66] *Lithium iron phosphate battery*. In: *Wikipedia*. Page Version ID: 1022457975. May 10, 2021. URL: https://en.wikipedia.org/w/index.php?title=Lithium_iron_phosphate_battery&oldid=1022457975 (visited on 05/23/2021).

[67] *lithium iron phosphate construction - Google Search.* URL: https://www.google.com/search?q=lithium+iron+phosphate+construction&tbm=isch&ved=2ahUKEwjr2_r8kuPwAhXEuyoKHbSfDCUQ2-cCegQIABAA&oq=lithium+iron+phosphate+construction&gs_lcp=CgNpbWcQAzoECCMQJzoCCAA6BggAEAgQHjoECAAQGFDHsw
sclient=img&ei=UAusYOvzBMT3qgG0v7KoAg&bih=937&biw=1903&hl=en#imgrc=SAA_5HPYRQlANM (visited on 05/24/2021).

[68] *Lithium-based Batteries Information.* URL: https://batteryuniversity.com/learn/article/lithium_based_batteries (visited on 05/23/2021).

[69] *Lithium-Ion Battery.* DiverBattery. URL: https://diverbattery.com/lithium-ion-battery/ (visited on 05/24/2021).

[70] Livox. *Mid-40 lidar sensor.* URL: https://www.livoxtech.com/mid-40-and-mid-100 (visited on 05/19/2021).

[71] *Livox Showcase - UAV 3D Mapping.* URL: https://www.livoxtech.com/showcase/1/ (visited on 05/24/2021).

[72] *Look at an AGM battery for your boatto keep the power on.* URL: http://www.fishingandboats.com/agm-battery.html (visited on 05/24/2021).

[73] Manav M. *Convolutional Neural Networks (CNN).* URL: https://www.analyticsvidhya.com/blog/2021/05/convolutional-neural-networks-cnn/ (visited on 05/24/2021).

[74] Guidance Marine. *CyScan®.* URL: https://www.guidance.eu.com/cyscan (visited on 05/19/2021).

[75] Guidance Marine. *Laser Targets.* URL: https://www.guidance.eu.com/laser-targets (visited on 05/19/2021).

[76] J. Mårtensson. "EL2620 Lecture Notes 2020". In: 2020.

[77] Alfredo Martins et al. "Autonomous Surface Vehicle Docking Manoeuvre with Visual Information". In: May 2007, pp. 4994–4999. DOI: 10.1109/ROBOT.2007.364249.

[78] Andreas Bell Martinsen et al. "Reinforcement Learning-Based Tracking Control of USVs in Varying Operational Conditions". In: *Frontiers in Robotics and AI* 7 (Mar. 2020). DOI: 10.3389/frobt.2020.00032.

[79] MathworksDL. *What is deep learning?* URL: https://se.mathworks.com/discovery/deep-learning.html (visited on 05/19/2021).

[80] *MATLAB by The MathWorks Inc.* Dec. 16, 2021. (Visited on 12/16/2021).

[81] Vanessa Mazzari. *How to select the right LiDAR?* Aug. 2019. URL: https://blog.generationrobots.com/en/how-to-select-the-right-lidar/ (visited on 05/19/2021).

[82] *Minn Kota assembly diagram.* URL: https://motors.johnsonoutdoors.com/PartStream/Parts.aspx (visited on 05/24/2021).

[83] *Minn Kota Riptide Transom.* URL: https://minnkotamotors.johnsonoutdoors.com/saltwater-trolling-motors/riptide-transom (visited on 05/24/2021).

[84] *monday.com pricing and plans.* monday.com. URL: https://monday.com/pricing (visited on 05/23/2021).

[85] Akeem Whitehead Mubina Toa. *Ultrasonic Sensing Basics.* URL: https://www.ti.com/lit/an/slaa907c/slaa907c.pdf?ts=1618827457631&ref_url=https%253A%252F%252Fwww.google.com%252F (visited on 05/21/2021).

[86] Charles Murray. *Autonomous Cars Look to Sensor Advancements in 2019.* Jan. 2019. URL: https://www.designnews.com/electronics-test/autonomous-cars-look-sensor-advancements-2019 (visited on 05/19/2021).

[87] Nvidia. *Jetson Xavier NX Module and Developer Kit.* URL: https://www.nvidia.com/en-us/autonomous-machines/embedded-systems/jetson-xavier-nx/ (visited on 05/23/2021).

[88] Ludwig Nyberg. “Thrust Allocation for Jet Driven Surface Vessels”. MA thesis. Stockholm, Sweden: KTH Royal Institute of Technology, 2019.

[89] *Outboard RC boat motor.* URL: https://hobbyking.com/en_us/cnc-aluminum-outboard-style-boat-drive-w-2630kv-motor-suit-hobbyking-h2o-style-650ep.html (visited on 05/24/2021).

[90] *polyfitZero.* URL: https://se.mathworks.com/matlabcentral/fileexchange/35401-polyfitzero (visited on 12/18/2021).

[91] *Product Introduction Azipod XO2100 and XO230.* URL: https://library.e.abb.com/public/6c1b0250efd18e73c1257a530040dcf2/XO2100_XO2300_Product_Intro_lowres.pdf (visited on 05/24/2021).

[92] *Public transport.* Candela. URL: https://candelaspeedboat.com/public-transport/ (visited on 05/22/2021).

[93] *Pure Pursuit Controller - MATLAB by The MathWorks Inc.* Dec. 18, 2021. (Visited on 12/18/2021).

[94] *Raymarine DockSense Alert.* URL: https://www.raymarine.com/assisted-docking/docksense-alert.html (visited on 05/20/2021).

[95] *Raymarine DockSense Control.* URL: https://www.raymarine.com/assisted-docking/docksense-control.html (visited on 05/20/2021).

[96] Joseph Redmon and Ali Farhadi. “YOLOv3: An Incremental Improvement”. In: *arXiv* (2018).

[97] Alan Rankin Robert Ferguson Matthieu Chevrier. *mmWave radar: Enabling greater intelligent autonomy at the edge.* URL: https://www.ti.com/lit/wp/sszy035/sszy035.pdf?ts=1618760691280 (visited on 05/21/2021).

[98] *RoboSense RS-LiDAR-16 3D Laser Range Finder.* URL: https://www.generationrobots.com/en/403308-robosense-rs-lidar-16-laser-range-finder.html (visited on 05/19/2021).

[99] C. C. Robusto. "The Cosine-Haversine Formula". In: *The American Mathematical Monthly* 64.1 (1957), pp. 38–40. ISSN: 00029890, 19300972. URL: http://www.jstor.org/stable/2309088.

[100] AAWA Initiative Rolls Royce. *Remote and Autonomous Ships The next steps.* URL: https://www.rolls-royce.com/~/media/Files/R/Rolls-Royce/documents/customers/marine/ship-intel/aawa-whitepaper-210616.pdf (visited on 05/22/2021).

[101] Michailas Romanovas et al. "Application of fractional sensor fusion algorithms for inertial MEMS sensing". In: *Mathematical Modelling and Analysis - MATH MODEL ANAL* 14 (June 2009), pp. 199–209. DOI: 10.3846/1392-6292.2009.14.199-209.

[102] *ROS - Robot Operating System.* Dec. 15, 2021. (Visited on 12/15/2021).

[103] M. Rosa. "Autonomous Surface Vehicle based docking for an Autonomous Underwater Vehicle". In: 2017.

[104] Adrian Rosebrock. *OpenCV Template Matching (cv2.matchTemplate).* URL: https://www.pyimagesearch.com/2021/03/22/opencv-template-matching-cv2-matchtemplate/ (visited on 05/23/2021).

[105] *Serial Communication.* Dec. 14, 2021. (Visited on 12/14/2021).

[106] *servo control of stepper motors.* 2017. URL: https://calischs.pages.cba.mit.edu/step/servo-stepper/ (visited on 05/24/2021).

[107] Bhupendra Sharma. *What is LiDAR technology and how does it work?* Feb. 2021. URL: https://www.geospatialworld.net/blogs/what-is-lidar-technology-and-how-does-it-work/ (visited on 05/19/2021).

[108] *Simulink by The MathWorks Inc.* Dec. 16, 2021. (Visited on 12/16/2021).

[109] SNAME. *Nomenclature for treating the motion of a submerged body through a fluid (1st ed.)* Technical, research bulletin. Society of Naval Architects, and Marine Engineers, 1950.

[110] European Union Agency for the Space Programme. *What is GNSS?* May 17, 2021. URL: https://www.euspa.europa.eu/european-space/eu-space-programme/what-gnss (visited on 05/22/2021).

[111] *STM32 IDEs.* Dec. 15, 2021. (Visited on 12/15/2021).

[112] *STM32Cube initialization code generator.* Dec. 15, 2021. (Visited on 12/15/2021).

[113] Paridhi Swaroop and Neelam Sharma. "An Overview of Various Template Matching Methodologies in Image Processing". In: *International Journal of Computer Applications* 153.10 (Nov. 2016), pp. 8–14. DOI: 10.5120/ijca2016912165. URL: https://doi.org/10.5120/ijca2016912165.

[114] *SyRen 50A 6V-30V Regenerative Motor Driver.* URL: https://www.robotshop.com/eu/en/syren-motor-driver.html (visited on 12/18/2021).

[115] Luke Taylor and Geoff Nitschke. “Improving deep learning using generic data augmentation”. In: *arXiv preprint arXiv:1708.06020* (2017).

[116] *Tele-Operated Mobile Robot for 3D Visual Inspection Utilizing Distributed Operating System Platform.* Dec. 15, 2021. (Visited on 12/15/2021).

[117] *Terrova 80 lb. / FP / US2 / i-Pilot - 60” — Minn Kota Motors.* URL: https://minnkotamotors.johnsonoutdoors.com/freshwater-trolling-motors/terrova (visited on 05/24/2021).

[118] *The working principles of a compression load cell manufactured by...* May 24, 2021. URL: https://www.flintec.com/sv/viktgivare/lastceller/how-does-compression-load-cell?__geom=%E2%9C%AA (visited on 05/24/2021).

[119] Tobias Valentin Rye Torben. “Hybrid Control of Autonomous Ferries”. MA thesis. Norwegian University of Science and Technology, 2018.

[120] *UART: A Hardware Communication Protocol Understanding Universal Asynchronous Receiver/Transmitter.* Dec. 14, 2021. (Visited on 12/14/2021).

[121] The APRIL Robotics Laboratory at the University of Michigan. *AprilTag.* URL: https://april.eecs.umich.edu/software/apriltag (visited on 05/18/2021).

[122] Anete Vagale et al. “Path planning and collision avoidance for autonomous surface vehicles I: a review”. In: *Journal of Marine Science and Technology* (Jan. 2021). ISSN: 1437-8213. DOI: 10.1007/s00773-020-00787-6. URL: https://doi.org/10.1007/s00773-020-00787-6.

[123] Vines. *Introduction to Pure Pursuit Tracking Algorithm.* URL: https://vinesmsuic.github.io/2020/09/29/robotics-purepersuit/ (visited on 12/18/2021).

[124] *VRLA battery.* In: *Wikipedia.* Page Version ID: 1022692059. May 11, 2021. URL: https://en.wikipedia.org/w/index.php?title=VRLA_battery&oldid=1022692059 (visited on 05/23/2021).

[125] Le Wang et al. “State-of-the-Art Research on Motion Control of Maritime Autonomous Surface Ships”. In: *Journal of Marine Science and Engineering* 7.12 (2019). ISSN: 2077-1312. DOI: 10.3390/jmse7120438. URL: https://www.mdpi.com/2077-1312/7/12/438.

[126] Wärtsilä. *Look, Ma, No Hands! Auto-docking ferry successfully tested in Norway.* URL: https://www.wartsila.com/insights/article/look-ma-no-hands-auto-docking-ferry-successfully-tested-in-norway (visited on 05/19/2021).

[127] Wärtsilä. *Wärtsilä offers the world’s first commercially available auto-docking system.* URL: https://www.wartsila.com/media/news/03-06-2019-wartsila-offers-the-world-s-first-commercially-available-auto-docking-system-2460994 (visited on 05/19/2021).

[128] Wärtsilä. *Wärtsilä SmartMove*. URL: `https://www.wartsila.com/smartmove#SmartDock` (visited on 05/19/2021).

[129] Eric W Weisstein. *Jacobian*. URL: `https://mathworld.wolfram.com/Jacobian.html` (visited on 12/18/2021).

[130] *What is a Load Cell? — Types of Load Cells — RealPars*. May 24, 2021. URL: `https://realpars.com/load-cell/` (visited on 05/24/2021).

[131] Christian Wolff. *Frequency-Modulated Continuous-Wave Radar (FMCW Radar)*. URL: `https://www.radartutorial.eu/02.basics/Frequency%20Modulated%20Continuous%20Wave%20Radar.en.html` (visited on 05/20/2021).

[132] Christian Wolff. *Waves and Frequency Ranges*. URL: `https://www.radartutorial.eu/07.waves/Waves%20and%20Frequency%20Ranges.en.html` (visited on 05/19/2021).

[133] RF Wirless World. *Advantages of LiDAR — disadvantages of LiDAR*. URL: `https://www.rfwireless-world.com/Terminology/Advantages-and-Disadvantages-of-LiDAR.html` (visited on 05/19/2021).

[134] Inseok Yang, Dongik Lee, and Dong Han. "Designing a Robust Nonlinear Dynamic Inversion Controller for Spacecraft Formation Flying". In: *Mathematical Problems in Engineering* 2014 (July 2014), pp. 1–12. DOI: `10.1155/2014/471352`.

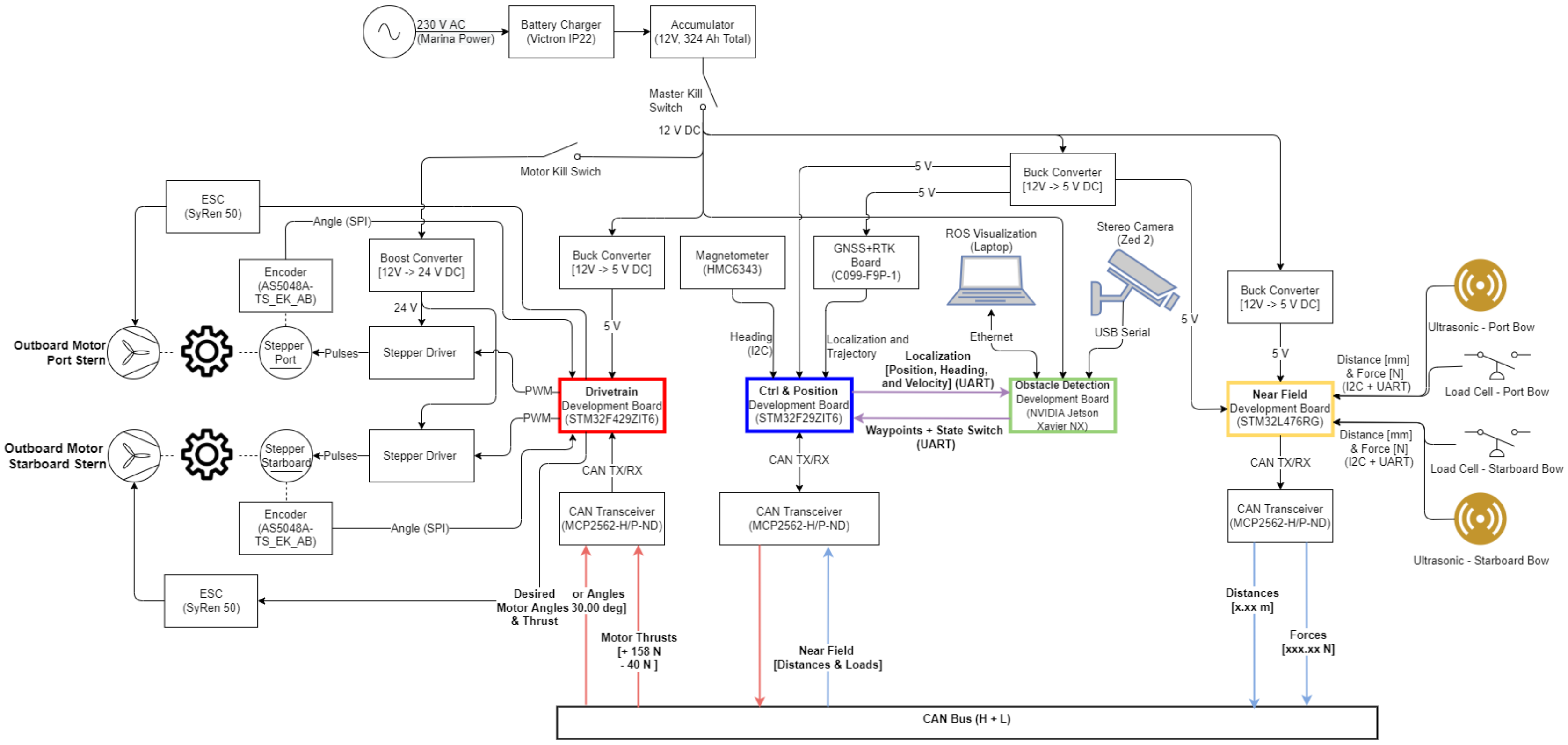
[135] *Yanmar*. URL: `https://www.yanmar.com/eu/` (visited on 05/19/2021).

[136] Yanmar. *Yanmar Develops Autonomous Technologies for Maritime Work*. URL: `https://www.yanmar.com/global/about/ymedia/article/yanmar-develops-autonomous-technologies-for-maritime-work.html` (visited on 05/19/2021).

[137] *ZED 2 Camera and SDK Overview*. Dec. 15, 2021. (Visited on 12/15/2021).

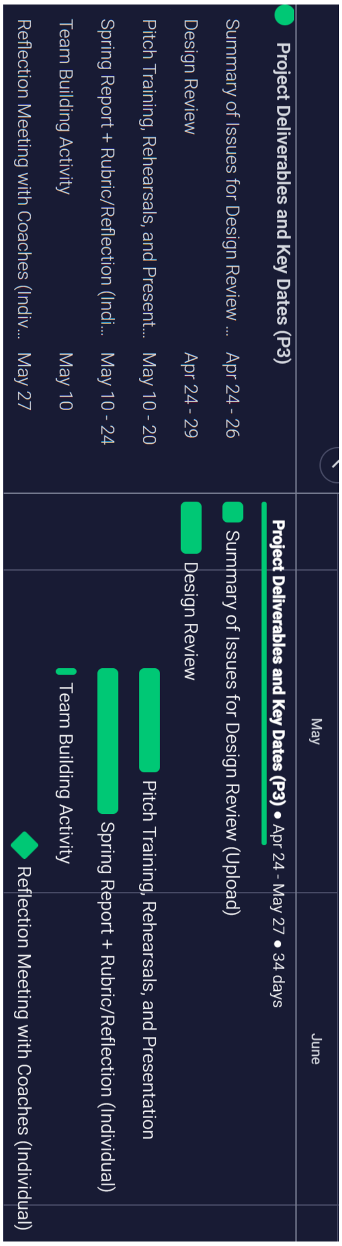
Appendix A

System Architecture


230 V AC
(Marina Power)
Battery Charger
(Victron IP22)
Accumulator
(12V, 324 Ah Total)
Master Kill
Switch
12 V DC
Motor Kill Swich
ESC
(SyRen 50)
Angle (SPI)
Boost Converter
[12V -> 24 V DC]
24 V
Encoder
(AS5048A-
TS_EK_AB)
Outboard Motor
Port Stern
Stepper
Port
Pulses
Stepper Driver
PWM
Buck Converter
[12V -> 5 V DC]
5 V
Drivetrain
Development Board
(STM32F429ZIT6)
Outboard Motor
Starboard Stern
Stepper
Starboard
Pulses
Stepper Driver
PWM
Encoder
(AS5048A-
TS_EK_AB)
Angle (SPI)
CAN TX/RX
CAN Transceiver
(MCP2562-H/P-ND)
ESC
(SyRen 50)
Desired
Motor Angles
& Thrust
or Angles
30.00 deg]
Motor Thrusts
[+ 158 N
- 40 N]
Magnetometer
(HMC6343)
Heading
(I2C)
5 V
5 V
Buck Converter
[12V -> 5 V DC]
GNSS+RTK
Board
(C099-F9P-1)
Localization and
Trajectory
Ctrl & Position
Development Board
(STM32F29ZIT6)
Localization
[Position, Heading,
and Velocity] (UART)
Waypoints + State Switch
(UART)
CAN TX/RX
CAN Transceiver
(MCP2562-H/P-ND)
Near Field
[Distances & Loads]
ROS Visualization
(Laptop)
Ethernet
Stereo Camera
(Zed 2)
USB Serial
Obstacle Detection
Development Board
(NVIDIA Jetson
Xavier NX)
5 V
Buck Converter
[12V -> 5 V DC]
5 V
Near Field
Development Board
(STM32L476RG)
CAN TX/RX
CAN Transceiver
(MCP2562-H/P-ND)
Distances
[x.xx m]
Forces
[xxx.xx N]
Ultrasonic - Port Bow
Distance [mm]
& Force [N]
(I2C + UART)
Load Cell - Port Bow
Distance [mm]
& Force [N]
(I2C + UART)
Load Cell - Starboard Bow
Ultrasonic - Starboard Bow
CAN Bus (H + L)

Appendix B

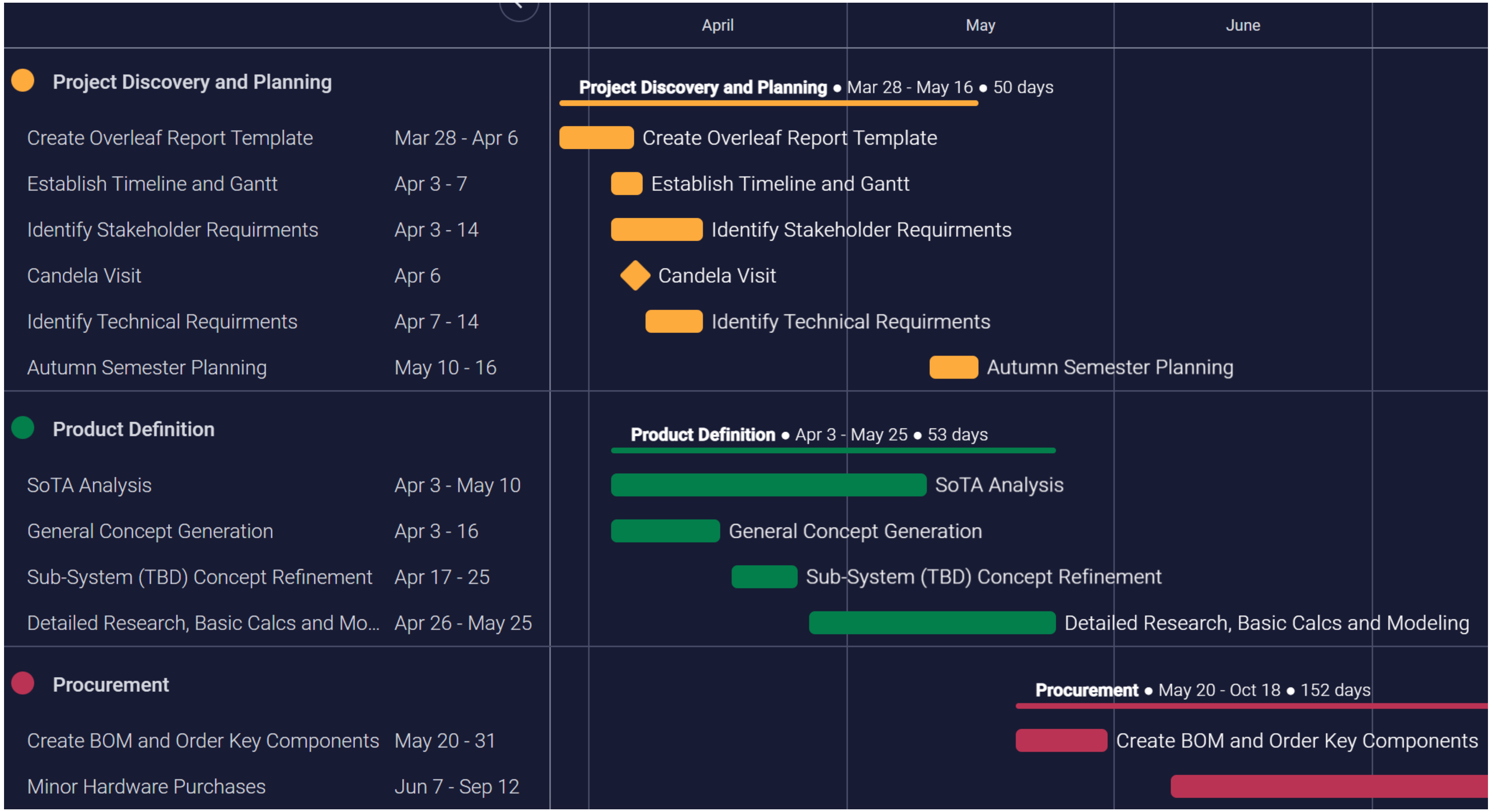
Gantt Chart



Project Deliverables and Key Dates (P3)

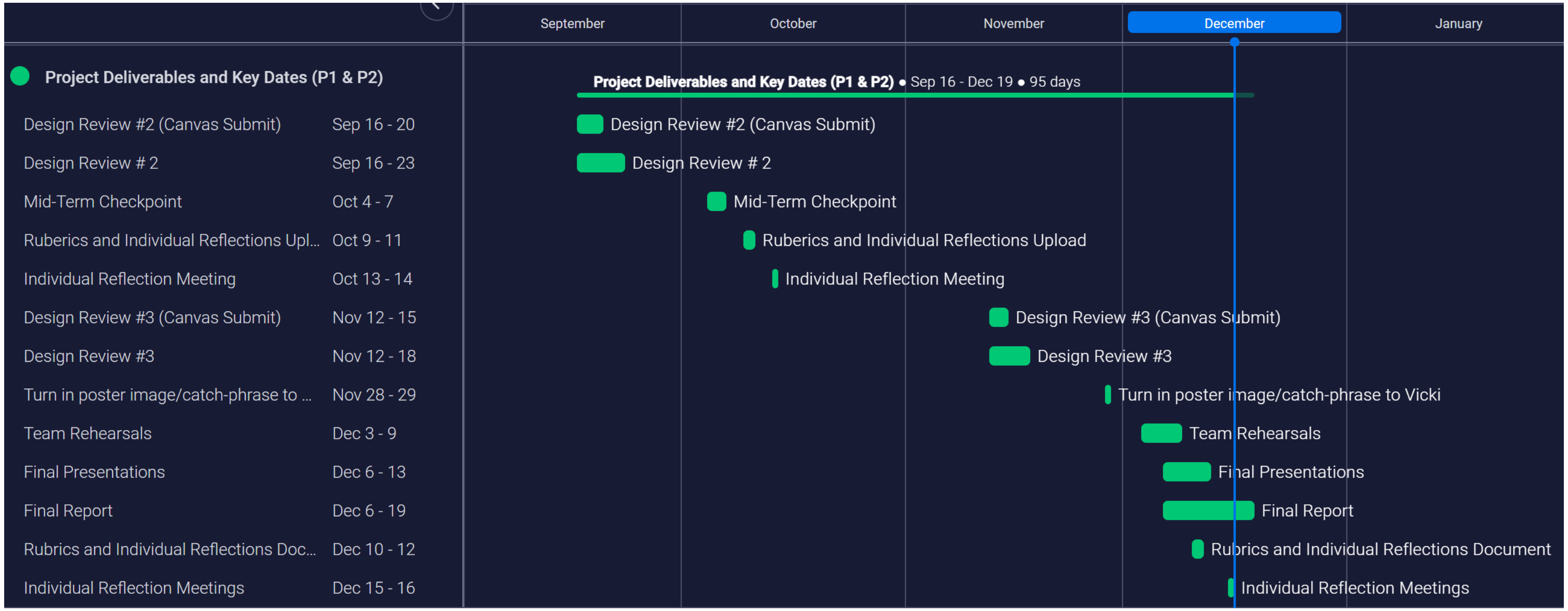
Summary of Issues for Design Review ...	Apr 24 - 26
Design Review	Apr 24 - 29
Pitch Training, Rehearsals, and Present...	May 10 - 20
Spring Report + Rubric/Reflection (Indi...	May 10 - 24
Team Building Activity	May 10
Reflection Meeting with Coaches (Indiv...	May 27

May
June
Project Deliverables and Key Dates (P3) • Apr 24 - May 27 • 34 days
Summary of Issues for Design Review (Upload)
Design Review
Pitch Training, Rehearsals, and Presentation
Spring Report + Rubric/Reflection (Individual)
Team Building Activity
Reflection Meeting with Coaches (Individual)



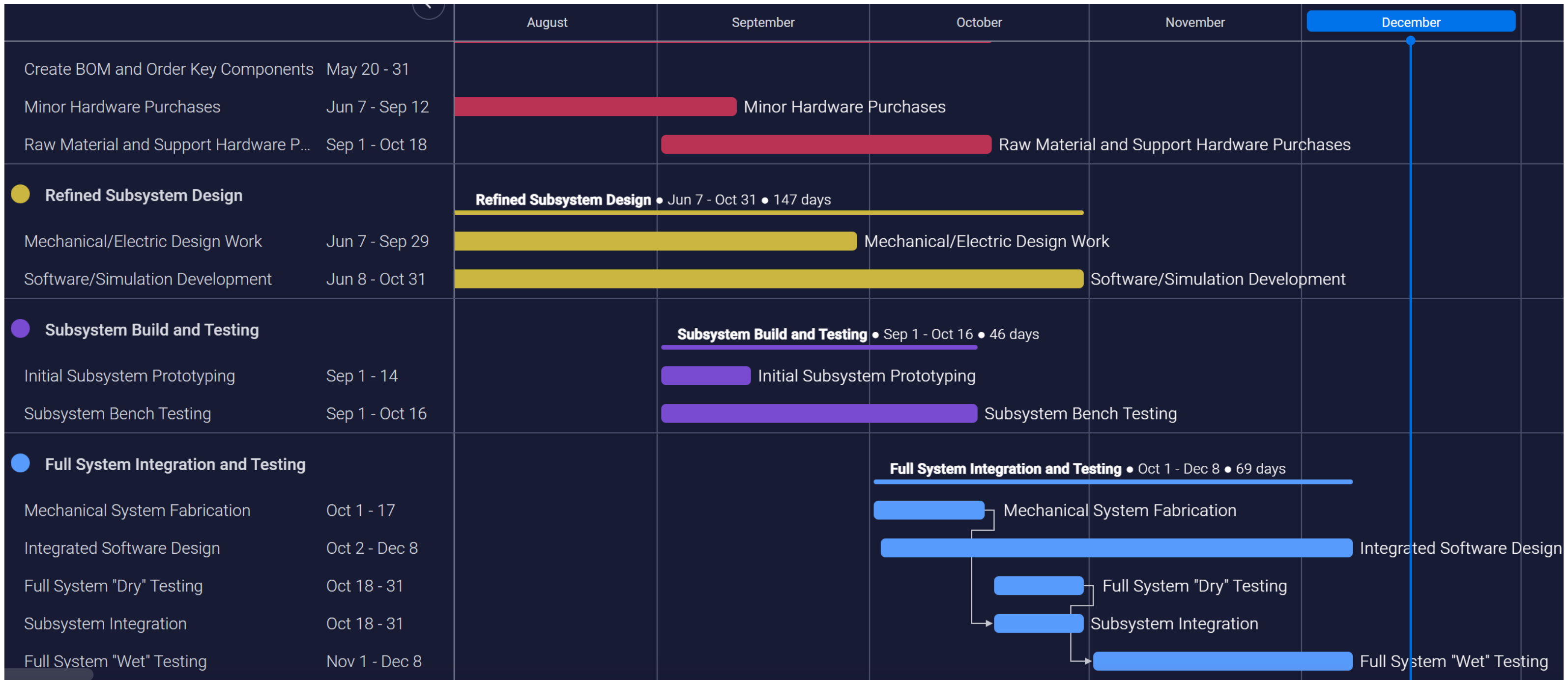
● **Project Discovery and Planning**	
Create Overleaf Report Template	Mar 28 - Apr 6
Establish Timeline and Gantt	Apr 3 - 7
Identify Stakeholder Requirments	Apr 3 - 14
Candela Visit	Apr 6
Identify Technical Requirments	Apr 7 - 14
Autumn Semester Planning	May 10 - 16
● **Product Definition**	
SoTA Analysis	Apr 3 - May 10
General Concept Generation	Apr 3 - 16
Sub-System (TBD) Concept Refinement	Apr 17 - 25
Detailed Research, Basic Calcs and Mo...	Apr 26 - May 25
● **Procurement**	
Create BOM and Order Key Components	May 20 - 31
Minor Hardware Purchases	Jun 7 - Sep 12

April
May
June
Project Discovery and Planning • Mar 28 - May 16 • 50 days
Create Overleaf Report Template
Establish Timeline and Gantt
Identify Stakeholder Requirments
Candela Visit
Identify Technical Requirments
Autumn Semester Planning
Product Definition • Apr 3 - May 25 • 53 days
SoTA Analysis
General Concept Generation
Sub-System (TBD) Concept Refinement
Detailed Research, Basic Calcs and Modeling
Procurement • May 20 - Oct 18 • 152 days
Create BOM and Order Key Components



● Project Deliverables and Key Dates (P1 & P2)	
Design Review #2 (Canvas Submit)	Sep 16 - 20
Design Review # 2	Sep 16 - 23
Mid-Term Checkpoint	Oct 4 - 7
Ruberics and Individual Reflections Upl...	Oct 9 - 11
Individual Reflection Meeting	Oct 13 - 14
Design Review #3 (Canvas Submit)	Nov 12 - 15
Design Review #3	Nov 12 - 18
Turn in poster image/catch-phrase to ...	Nov 28 - 29
Team Rehearsals	Dec 3 - 9
Final Presentations	Dec 6 - 13
Final Report	Dec 6 - 19
Rubrics and Individual Reflections Doc...	Dec 10 - 12
Individual Reflection Meetings	Dec 15 - 16

September
October
November
December
January
Project Deliverables and Key Dates (P1 & P2) ● Sep 16 - Dec 19 ● 95 days
Design Review #2 (Canvas Submit)
Design Review # 2
Mid-Term Checkpoint
Ruberics and Individual Reflections Upload
Individual Reflection Meeting
Design Review #3 (Canvas Submit)
Design Review #3
Turn in poster image/catch-phrase to Vicki
Team Rehearsals
Final Presentations
Final Report
Rubrics and Individual Reflections Document
Individual Reflection Meetings



Create BOM and Order Key Components	May 20 - 31
Minor Hardware Purchases	Jun 7 - Sep 12
Raw Material and Support Hardware P...	Sep 1 - Oct 18
● **Refined Subsystem Design**	
Mechanical/Electric Design Work	Jun 7 - Sep 29
Software/Simulation Development	Jun 8 - Oct 31
● **Subsystem Build and Testing**	
Initial Subsystem Prototyping	Sep 1 - 14
Subsystem Bench Testing	Sep 1 - Oct 16
● **Full System Integration and Testing**	
Mechanical System Fabrication	Oct 1 - 17
Integrated Software Design	Oct 2 - Dec 8
Full System "Dry" Testing	Oct 18 - 31
Subsystem Integration	Oct 18 - 31
Full System "Wet" Testing	Nov 1 - Dec 8

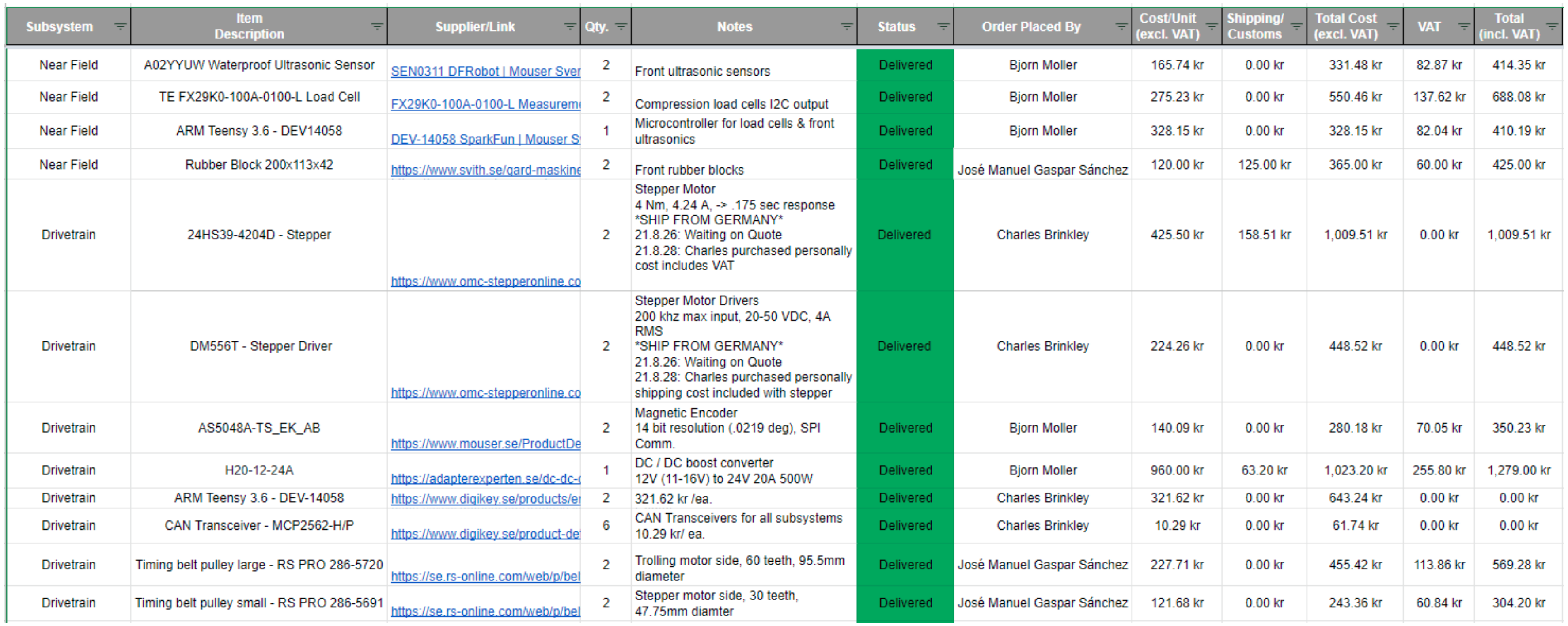
August
September
October
November
December
Minor Hardware Purchases
Raw Material and Support Hardware Purchases
Refined Subsystem Design • Jun 7 - Oct 31 • 147 days
Mechanical/Electric Design Work
Software/Simulation Development
Subsystem Build and Testing • Sep 1 - Oct 16 • 46 days
Initial Subsystem Prototyping
Subsystem Bench Testing
Full System Integration and Testing • Oct 1 - Dec 8 • 69 days
Mechanical System Fabrication
Integrated Software Design
Full System "Dry" Testing
Subsystem Integration
Full System "Wet" Testing

Appendix C

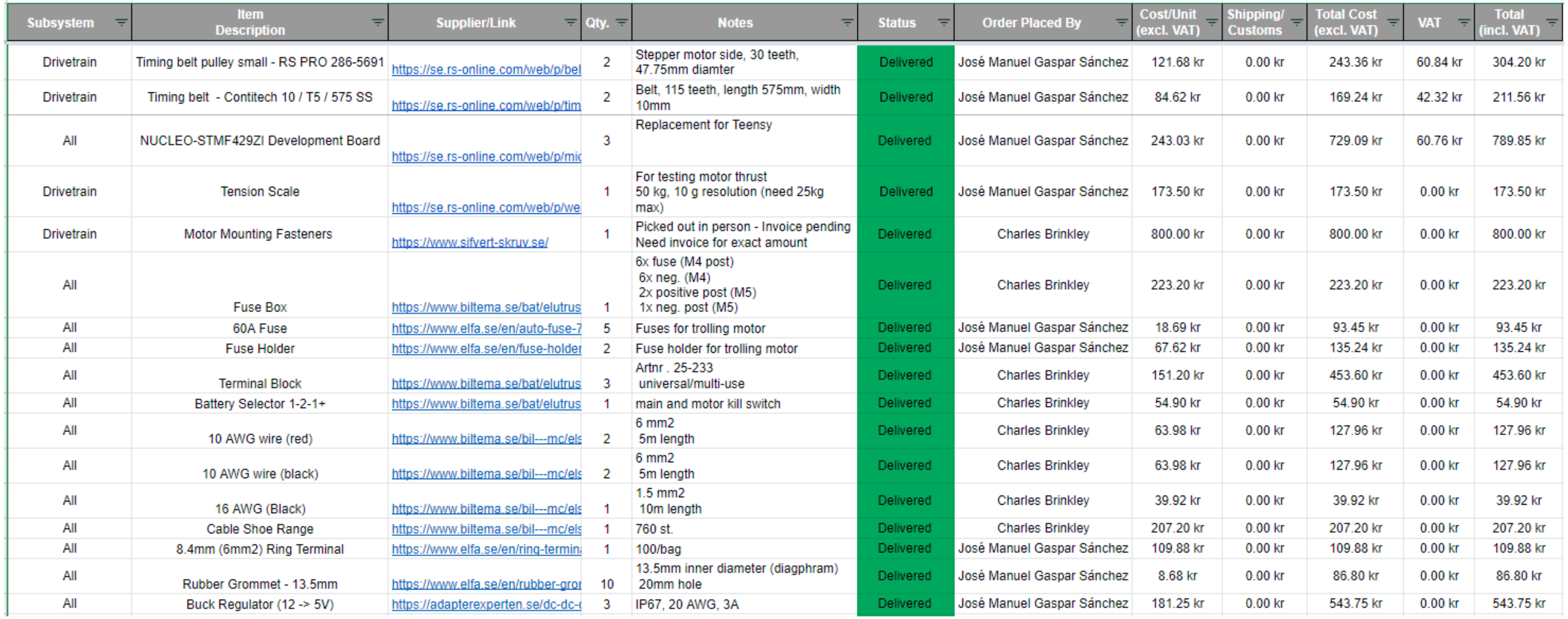
PO Tracker and BOM

TOTAL: (Excluding VAT)	**51,043.72 kr**
TOTAL VAT:	**11,296.21 kr**
TOTAL COSTS: (Incl. All Expenses)	**61,610.39 kr**
EST. WORKSHOP HOURS:	**1440**

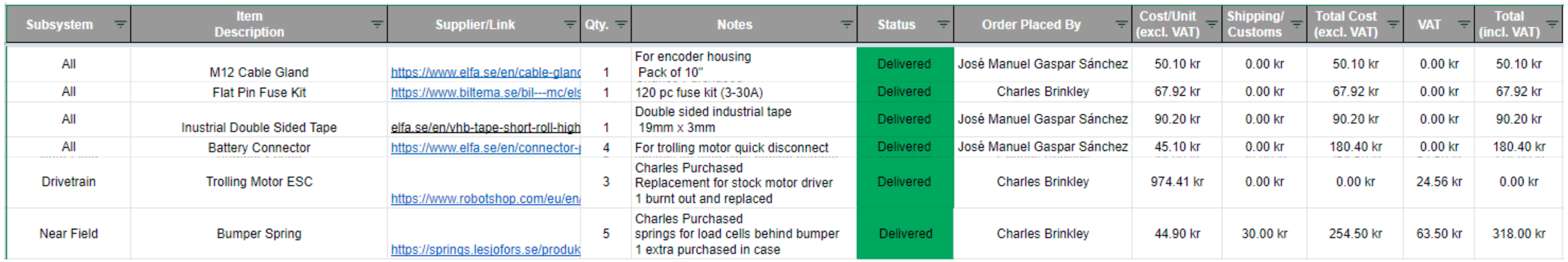
Subsystem	Item Description	Supplier/Link	Qty.	Notes	Status	Order Placed By	Cost/Unit (excl. VAT)	Shipping/ Customs	Total Cost (excl. VAT)	VAT	Total (incl. VAT)
Control and Position	C099-F9P-1	https://www.digikey.se/product-de	1	GNSS + RTK, U-Blox Purchased with Charles cash Customs Paid by Anton H.	Delivered	Charles Brinkley	1,623.00 kr	818.00 kr	2,441.00 kr	405.75 kr	2,846.75 kr
Control and Position	HMC6343 Breakout	https://www.elfa.se/sv/hmc6343-k	1	Magnemometer 21-7-26: Jose submitted	Delivered	José Manuel Gaspar Sánchez	1,475.00 kr	28.00 kr	1,503.00 kr	375.75 kr	1,878.75 kr
Control and Position	STM32H747I-DISCO	https://www.elfa.se/sv/discovery-s	1	Microcontroller 21-7-26: Jose submitted	Delivered	José Manuel Gaspar Sánchez	1,318.00 kr	28.00 kr	1,346.00 kr	336.50 kr	1,682.50 kr
Obstacle Detection	Zed 2	https://www.mybotshop.de/Stere	1	Stereo Camera 21-7-26: Jose submitted 21-8-19: Provided alt suppliers 21-8-20: Costing provided by Service Center (dont know costing specifics)	Delivered	José Manuel Gaspar Sánchez	4,503.40 kr	0.00 kr	4,503.40 kr	0.00 kr	4,503.40 kr
Obstacle Detection	NVIDIA Jetson Xavier NX	https://www.elfa.se/sv/nvidia-jetso	1	Microcontroller 21-7-26: Jose submitted (Elfa, 4869 kr) 21-8-23: **ON BACK-ORDER UNTIL JAN 2022 - Order canceled** ~~21-8-23: Charles purchased 1- to be delivered mid October (8097.50 kr~~) 21.8.26: Able to borrow unit from Ocado (Needing returned)	Delivered	José Manuel Gaspar Sánchez	0.00 kr	0.00 kr	0.00 kr	0.00 kr	0.00 kr
Drivetrain	Minn Kota Endura Max 55 (36" L)	https://www.muziker.se/minn-kot	2	Motor 21-7-26: Jose submitted 21-8-19: Provided alt suppliers 21.8.31: They are waiting on invoice	Delivered	José Manuel Gaspar Sánchez	3,546.75 kr	0.00 kr	7,093.50 kr	2,364.50 kr	9,458.00 kr
Drivetrain	Angler 330 Wide	http://www.bcmarine.se/index.php	1	Aluminum Boat Hull 21-7-26: Jose submitted 21-8-21: Alternative being found 21-8-26: Alternative: BC Marine	Delivered	José Manuel Gaspar Sánchez	13,425.00 kr	2,000.00 kr	15,425.00 kr	4,475.00 kr	19,900.00 kr
Drivetrain	AGM Batteries	https://www.batteriexpressen.se/p	3	12V @ 100 Ah/each. 21-7-21: Bjorn requested change from Amazon to Batterie Xpressen 21-7-26: Jose submitted	Delivered	José Manuel Gaspar Sánchez	1,950.75 kr	619.00 kr	6,471.25 kr	1,684.00 kr	8,155.25 kr
Drivetrain	12 V AC/DC Charger	https://www.batteriexpressen.se/	1	3x 12V 30A battery charger 21-7-26: Jose submitted	Delivered	José Manuel Gaspar Sánchez	1,801.50 kr	0.00 kr	1,801.50 kr	600.50 kr	2,402.00 kr



Subsystem	Item Description	Supplier/Link	Qty.	Notes	Status	Order Placed By	Cost/Unit (excl. VAT)	Shipping/ Customs	Total Cost (excl. VAT)	VAT	Total (incl. VAT)
Near Field	A02YYUW Waterproof Ultrasonic Sensor	SEN0311 DFRobot | Mouser Sver	2	Front ultrasonic sensors	Delivered	Bjorn Moller	165.74 kr	0.00 kr	331.48 kr	82.87 kr	414.35 kr
Near Field	TE FX29K0-100A-0100-L Load Cell	FX29K0-100A-0100-L Measurem	2	Compression load cells I2C output	Delivered	Bjorn Moller	275.23 kr	0.00 kr	550.46 kr	137.62 kr	688.08 kr
Near Field	ARM Teensy 3.6 - DEV14058	DEV-14058 SparkFun | Mouser S	1	Microcontroller for load cells & front ultrasonics	Delivered	Bjorn Moller	328.15 kr	0.00 kr	328.15 kr	82.04 kr	410.19 kr
Near Field	Rubber Block 200x113x42	https://www.svith.se/gard-maskine	2	Front rubber blocks	Delivered	José Manuel Gaspar Sánchez	120.00 kr	125.00 kr	365.00 kr	60.00 kr	425.00 kr
Drivetrain	24HS39-4204D - Stepper	https://www.omc-stepperonline.co	2	Stepper Motor 4 Nm, 4.24 A, -> .175 sec response *SHIP FROM GERMANY* 21.8.26: Waiting on Quote 21.8.28: Charles purchased personally cost includes VAT	Delivered	Charles Brinkley	425.50 kr	158.51 kr	1,009.51 kr	0.00 kr	1,009.51 kr
Drivetrain	DM556T - Stepper Driver	https://www.omc-stepperonline.co	2	Stepper Motor Drivers 200 khz max input, 20-50 VDC, 4A RMS *SHIP FROM GERMANY* 21.8.26: Waiting on Quote 21.8.28: Charles purchased personally shipping cost included with stepper	Delivered	Charles Brinkley	224.26 kr	0.00 kr	448.52 kr	0.00 kr	448.52 kr
Drivetrain	AS5048A-TS_EK_AB	https://www.mouser.se/ProductDe	2	Magnetic Encoder 14 bit resolution (.0219 deg), SPI Comm.	Delivered	Bjorn Moller	140.09 kr	0.00 kr	280.18 kr	70.05 kr	350.23 kr
Drivetrain	H20-12-24A	https://adapterexperten.se/dc-dc-c	1	DC / DC boost converter 12V (11-16V) to 24V 20A 500W	Delivered	Bjorn Moller	960.00 kr	63.20 kr	1,023.20 kr	255.80 kr	1,279.00 kr
Drivetrain	ARM Teensy 3.6 - DEV-14058	https://www.digikey.se/products/e	2	321.62 kr /ea.	Delivered	Charles Brinkley	321.62 kr	0.00 kr	643.24 kr	0.00 kr	0.00 kr
Drivetrain	CAN Transceiver - MCP2562-H/P	https://www.digikey.se/product-de	6	CAN Transceivers for all subsystems 10.29 kr/ ea.	Delivered	Charles Brinkley	10.29 kr	0.00 kr	61.74 kr	0.00 kr	0.00 kr
Drivetrain	Timing belt pulley large - RS PRO 286-5720	https://se.rs-online.com/web/p/bel	2	Trolling motor side, 60 teeth, 95.5mm diameter	Delivered	José Manuel Gaspar Sánchez	227.71 kr	0.00 kr	455.42 kr	113.86 kr	569.28 kr
Drivetrain	Timing belt pulley small - RS PRO 286-5691	https://se.rs-online.com/web/p/bel	2	Stepper motor side, 30 teeth, 47.75mm diamter	Delivered	José Manuel Gaspar Sánchez	121.68 kr	0.00 kr	243.36 kr	60.84 kr	304.20 kr



Subsystem	Item Description	Supplier/Link	Qty.	Notes	Status	Order Placed By	Cost/Unit (excl. VAT)	Shipping/ Customs	Total Cost (excl. VAT)	VAT	Total (incl. VAT)
Drivetrain	Timing belt pulley small - RS PRO 286-5691	https://se.rs-online.com/web/p/bel	2	Stepper motor side, 30 teeth, 47.75mm diamter	Delivered	José Manuel Gaspar Sánchez	121.68 kr	0.00 kr	243.36 kr	60.84 kr	304.20 kr
Drivetrain	Timing belt - Contitech 10 / T5 / 575 SS	https://se.rs-online.com/web/p/tim	2	Belt, 115 teeth, length 575mm, width 10mm	Delivered	José Manuel Gaspar Sánchez	84.62 kr	0.00 kr	169.24 kr	42.32 kr	211.56 kr
All	NUCLEO-STMF429ZI Development Board	https://se.rs-online.com/web/p/mic	3	Replacement for Teensy	Delivered	José Manuel Gaspar Sánchez	243.03 kr	0.00 kr	729.09 kr	60.76 kr	789.85 kr
Drivetrain	Tension Scale	https://se.rs-online.com/web/p/we	1	For testing motor thrust 50 kg, 10 g resolution (need 25kg max)	Delivered	José Manuel Gaspar Sánchez	173.50 kr	0.00 kr	173.50 kr	0.00 kr	173.50 kr
Drivetrain	Motor Mounting Fasteners	https://www.sifvert-skruv.se/	1	Picked out in person - Invoice pending Need invoice for exact amount	Delivered	Charles Brinkley	800.00 kr	0.00 kr	800.00 kr	0.00 kr	800.00 kr
All	Fuse Box	https://www.biltema.se/bat/elutrus	1	6x fuse (M4 post) 6x neg. (M4) 2x positive post (M5) 1x neg. post (M5)	Delivered	Charles Brinkley	223.20 kr	0.00 kr	223.20 kr	0.00 kr	223.20 kr
All	60A Fuse	https://www.elfa.se/en/auto-fuse-7	5	Fuses for trolling motor	Delivered	José Manuel Gaspar Sánchez	18.69 kr	0.00 kr	93.45 kr	0.00 kr	93.45 kr
All	Fuse Holder	https://www.elfa.se/en/fuse-holder	2	Fuse holder for trolling motor	Delivered	José Manuel Gaspar Sánchez	67.62 kr	0.00 kr	135.24 kr	0.00 kr	135.24 kr
All	Terminal Block	https://www.biltema.se/bat/elutrus	3	Artnr . 25-233 universal/multi-use	Delivered	Charles Brinkley	151.20 kr	0.00 kr	453.60 kr	0.00 kr	453.60 kr
All	Battery Selector 1-2-1+	https://www.biltema.se/bat/elutrus	1	main and motor kill switch	Delivered	Charles Brinkley	54.90 kr	0.00 kr	54.90 kr	0.00 kr	54.90 kr
All	10 AWG wire (red)	https://www.biltema.se/bil---mc/els	2	6 mm2 5m length	Delivered	Charles Brinkley	63.98 kr	0.00 kr	127.96 kr	0.00 kr	127.96 kr
All	10 AWG wire (black)	https://www.biltema.se/bil---mc/els	2	6 mm2 5m length	Delivered	Charles Brinkley	63.98 kr	0.00 kr	127.96 kr	0.00 kr	127.96 kr
All	16 AWG (Black)	https://www.biltema.se/bil---mc/els	1	1.5 mm2 10m length	Delivered	Charles Brinkley	39.92 kr	0.00 kr	39.92 kr	0.00 kr	39.92 kr
All	Cable Shoe Range	https://www.biltema.se/bil---mc/els	1	760 st.	Delivered	Charles Brinkley	207.20 kr	0.00 kr	207.20 kr	0.00 kr	207.20 kr
All	8.4mm (6mm2) Ring Terminal	https://www.elfa.se/en/ring-termin	1	100/bag	Delivered	José Manuel Gaspar Sánchez	109.88 kr	0.00 kr	109.88 kr	0.00 kr	109.88 kr
All	Rubber Grommet - 13.5mm	https://www.elfa.se/en/rubber-gro	10	13.5mm inner diameter (diagphram) 20mm hole	Delivered	José Manuel Gaspar Sánchez	8.68 kr	0.00 kr	86.80 kr	0.00 kr	86.80 kr
All	Buck Regulator (12 -> 5V)	https://adapterexperten.se/dc-dc-	3	IP67, 20 AWG, 3A	Delivered	José Manuel Gaspar Sánchez	181.25 kr	0.00 kr	543.75 kr	0.00 kr	543.75 kr



Subsystem	Item Description	Supplier/Link	Qty.	Notes	Status	Order Placed By	Cost/Unit (excl. VAT)	Shipping/ Customs	Total Cost (excl. VAT)	VAT	Total (incl. VAT)
All	M12 Cable Gland	https://www.elfa.se/en/cable-glanc	1	For encoder housing Pack of 10"	Delivered	José Manuel Gaspar Sánchez	50.10 kr	0.00 kr	50.10 kr	0.00 kr	50.10 kr
All	Flat Pin Fuse Kit	https://www.biltema.se/bil---mc/els	1	120 pc fuse kit (3-30A)	Delivered	Charles Brinkley	67.92 kr	0.00 kr	67.92 kr	0.00 kr	67.92 kr
All	Inustrial Double Sided Tape	elfa.se/en/vhb-tape-short-roll-high	1	Double sided industrial tape 19mm x 3mm	Delivered	José Manuel Gaspar Sánchez	90.20 kr	0.00 kr	90.20 kr	0.00 kr	90.20 kr
All	Battery Connector	https://www.elfa.se/en/connector-i	4	For trolling motor quick disconnect	Delivered	José Manuel Gaspar Sánchez	45.10 kr	0.00 kr	180.40 kr	0.00 kr	180.40 kr
Drivetrain	Trolling Motor ESC	https://www.robotshop.com/eu/en/	3	Charles Purchased Replacement for stock motor driver 1 burnt out and replaced	Delivered	Charles Brinkley	974.41 kr	0.00 kr	0.00 kr	24.56 kr	0.00 kr
Near Field	Bumper Spring	https://springs.lesjofors.se/produk	5	Charles Purchased springs for load cells behind bumper 1 extra purchased in case	Delivered	Charles Brinkley	44.90 kr	30.00 kr	254.50 kr	63.50 kr	318.00 kr