Master's thesis

NTNU
Norwegian University of Science and Technology
Faculty of Engineering
Department of Mechanical and Industrial Engineering

Lars Digerud

Utilizing vision-based techniques to aid unmanned cargo ships in the auto-docking scenario

Master's thesis in Mechanical Engineering
Supervisor: Martin Steinert
Co-supervisor: Anna Olsen, Kim Alexander Christensen
January 2022

NTNU
Norwegian University of Science and Technology

Lars Digerud

Utilizing vision-based techniques to aid unmanned cargo ships in the auto-docking scenario

Master's thesis in Mechanical Engineering
Supervisor: Martin Steinert
Co-supervisor: Anna Olsen, Kim Alexander Christensen
January 2022

Norwegian University of Science and Technology
Faculty of Engineering
Department of Mechanical and Industrial Engineering

NTNU
Norwegian University of
Science and Technology

Abstract

The focus on autonomy in the maritime industry has in recent years increased. Autonomous vessels have the potential to reduce costs and improve safety in the maritime industry. Unmanned Surface Vehicles (USVs) are reaching the commercial market, e.g., Maritime Robotics, SailDrone, and Kongsberg. However, further enchantment of the field requires more product development to make it possible for USVs to perform more complex operations without any human intervention. This project is a collaboration between NTNU, Fosenregionen, Maritime Robotics, and SFI Autoship. The project's grand vision is to connect urban and remote parts of the Fosen region with autonomous cargo boats. Here, the crew and port handling is considered to constitute a significant cost associated with short-range shipping.

The objective of this master thesis is to develop solutions for *autonomous cargo handling* to connect small remote places, e.g., islands and coastal towns. This master thesis combines theory and practice to verify any predictions. By looking at the theoretical aspects of autonomous cargo handling, in addition to testing, the understanding of the theoretical methods, strengths, and shortcomings is in this project reviewed.

This thesis describes a vision-based approach of using fiducial markers to aid USVs while auto-docking. Recent applications of USVs have shown increased ease and increased efficiency of cargo shipping at precise locations. However, further field enhancement requires operations over long periods, e.g., days or weeks of which the vessel must be precisely docked to be refueled/recharged or for complex cargo handling in transit. Utilizing vision-based techniques allows USVs to orient themselves to their environment from their perspective and may represent an efficient method for such vessels to accurately orientate themselves in the docking scenario. In addition, fiducial markers can give USVs six degrees of freedom orientation relative to a target, allowing more complex interactions with a potential pier or a floating docking station.

RTK GNSS with a base station on the pier was used to validate the estimated camera positions to obtain reliable additional positioning. The experiments show

that essential computer vision (CV) techniques using fiducial markers perform well in obtaining accurate outdoor position estimates in optimal conditions. However, in more adverse conditions, the results demonstrated that the performance decreased significantly. Finally, suggestions for further development are given.

The following topics are addressed in this master:

1. Challenges with autonomous cargo handling.
2. An overview of available technologies.
3. A edge case for autonomous cargo handling: The docking process.
4. Development of a vision-based system using fiducial markers to precisely orientate a USV in the docking scenario.

Sammendrag

Denne masteroppgaven er en del av et større forskningsprosjekt i samarbeid med NTNU, Fosenregionen, Maritime Robotics og SFI Autoship. Målet med forskningen er å binde mindre urbane regioner av Fosen med autonome cargo fartøyer. For at ubemannede marine cargo systemer kan nå det kommersielle markedet, er det mange mulige utfordringer som må identifiseres og løses.

Autonome fartøyer kan enten fortøye til flytende brygger eller til stasjonære kaier. Vi mennesker har evnen til å enkelt kunne justere våre bevegelser til omgivelsene, mens autonome fartøyer har i dag ikke den samme evnen. I tillegg har mennesker evnen til å lokalisere og posisjonere et fartøy til samme lokasjon og utføre vanskelige løfteoperasjoner mellom fartøyet og kaien. For at autonome fartøyer som skal kunne gjøre den samme prosessen uten noen menneskelig medvirkning, kreves det ulike løsninger som sammen gir autonome fartøyer en lignede evne.

Denne oppgaven beskriver en kamera-basert metode som identifiserer objekt-markører på en kai og kalkulerer fartøyets orientering relativt til disse når fartøyet nærmer seg. Markørene gir fartøyet informasjon om dens orientering i seks frihetsgrader relativt til kaien i et lokalt referanse system. Objekt-markørene er uavhengige om det er en flytende brygge eller en stasjonær kai. Metoden gjør at fartøyet kan estimere en lokal presis posisjon av kaien uten bruk av ekstern kommunikasjon til fartøyet. Ettersom kamerasystemet har evnen til å gi informasjon over seks frihetsgrader, kan et slikt system gi autonome fartøyer evner til mer komplekse interaksjoner med en docking stasjon, f.eks. robotarmer på fartøyet som flytter cargo mellom fartøy og en docking stasjon.

For å verifisere posisjonsestimatene fra kamerasystemet ble det brukt realtids kinematiske måling med en lokal basestasjon. Denne metoden ga en realistisk og et presist sammenligningsgrunnlag til posisjonsestimatene fra kamerasystemet mot fartøyets globale posisjoner. Posisjonene fra kamera systemet ble i tillegg Kalman filtert i et forsøk på å oppnå bedre nøyaktighet og ytelse, men også integrert som et steg nærmere for å bli en del av navigasjonssystemet til et autonomt fartøy. Resultatene fra de gjennomførte eksperimentene viste at systemet under gode forhold

fungerte godt, men i mer utfordrende situasjoner, f.eks. mørke eller reflektert lys, gikk ytelsen til systemet ned. Til slutt, blir det gitt forslag til forbedringer.

Følgende temaer blir diskutert i denne masteroppgaven:

1. Utfordringer med autonom cargo håndtering.
2. Oversikt over tilgjengelige teknologier.
3. En spesialutfordring ved autonom cargo håndtering: Dockingprosessen.
4. Utviklingen av et kamera-basert system som bruker fidusielle markører for å gi presise posisjons estimater.

Preface

This master's thesis is part of the Master of Science degree in Mechanical Engineering at the Norwegian University of Science and Technology (NTNU) in Trondheim. The following thesis was written in its entirety by Lars Digerud during the autumn of 2021, and the workload is equivalent to 30 ECTS. The thesis is an exploratory research study focusing on the concept of autonomy and uses vision-based techniques to orientate the USV in the docking scenario.

The motivation for this thesis is an urge to explore different fields of technology and understand how different fields of technology can be combined and applied to problems that revolve around autonomous vessels. The author has a bachelor's degree in mechatronics/mechanical engineering and has worked part-time for Maritime Robotics with designing, prototyping, and building autonomous systems. Working at Maritime Robotics has motivated the author to take a more thorough look at existing technology to solve unsolved problems.

Additionally, I would like to thank my advisors, Kim Alexander Christensen, Martin Steinert, and Anna Olsen, for their excellent guidance through the thesis work. A special thank goes to Øystein Volden and Petter Solnør for extensive support for developing the camera system and helping with performing the experiments conducted in this thesis.

Finally, I would like to thank my family for all support and my girlfriend, Sandra, for bearing any frustration through the work and still sticking with me. I love you!

Contents

Nomenclature

ASV	Autonomous Surface vehicle
CoG	Center of gravity
CPU	Central Processing unit
CV	Computer Vision
DUNE	Distributed and Unified Numerics Environment
EO	Electro-Optical
GNSS	Global Navigation Satellite System
GPS	Global Positioning System
IMU	Inertial Measurement Unit
INS	Inertial Navigation System
IR	Infra-Red
KF	Kalman Filter
NED	North East Down
NTNU	The Norwegian University of Science and Technology
OpenCV	Open Source Computer Vision Library
PnP	Perspective-n-Point
RMSE	Root Mean Square Error
ROS	Robot Operating System
ROV	Remotely Operated Vehicle
RTCM	Radio Technical Commission for Maritime Services

RTK	Real Time Kinematics
$SLAM$	Simultaneous Localisation and Mapping
USV	Unmanned Surface vehicle
UWB	Ultra Wide Band
VO	Visual Odometry

Chapter 1

Introduction

1.1 Motivation and background

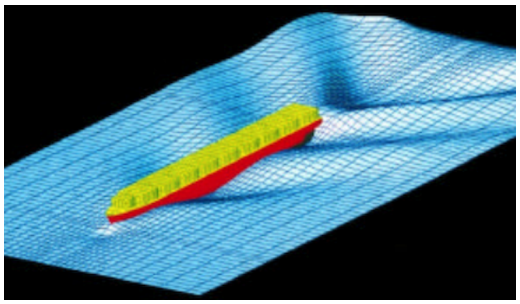
This master project is part of a larger research project in a collaboration between NTNU, Fosen region inter-municipal political council (IKPR), Maritime robotics, and SFI Autoship. The project's grand vision is to connect remote parts of the Fosen region with autonomous cargo boats. A significant cost of short-range shipping comes from the crew and port handling. Here, autonomous shipping is assumed be significant part of reducing these costs. In addition to this, autonomous shipping can enable new transport and supply chain opportunities that, thus far, have not been economically feasible, like individualized transport of packages.

The use and interest of autonomous systems has in recent years increased significant, ranging from robotic lawn mowers, autonomous cars and buses, to name a few. In 2017, Yara and Kongsberg revealed that they would partner up to build the world's first autonomous and fully electric container vessel, Yara Birkeland [1]. The aim is to eliminate 40 000 truck drives between Yara's Porsgrunn fertilizer plant in the south of Norway and the port of Brevik and Larvik, expectantly reducing greenhouse gas emissions. The initial plan for the vessel was to begin testing in 2019 and be certified as a fully autonomous cargo ship by 2021. However, as of the thesis being written the land-based system which interacts with the vessel become too complicated [2]. Nevertheless, Yara and Kongsberg aims to put the ship in action by 2022 [3].

The are several potential advantages of making cargo ships that can navigate and transport cargo from A to B without any human intervention, e.g., lower transport costs, more environmentally friendly transport of cargo, better space distribution of cargo on the ships. Autonomous cargo ships are therefore considered to be a pivotal element to achieve a much more competitive and sustainable shipping industry.

Figure 1.1: Concept illustration of Yara Birkeland. Image courtesy to [1].

(a) Vision for master-slave concept (Schönknecht et al., 1973) [4].



(b) Unmanned Container Vessel Vision (Kasai et al., 1996).

Figure 1.2: Early visions of unmanned concepts on cargo ships [5].

The idea of autonomous cargo shipping is not new. The ideas presented in figure 1.2 illustrate early visions for the future on concepts of cargo ships, here, ideas about a master-slave configuration are shown in figure 1.2 a) from Schönknect et al., 1983 [4]. A master ship is used to control several other cargo ships to, e.g., reduce the amount of crew members. While figure 1.2 b) illustrates a fully unmanned container vessel. A short excerpt from Schönknect et al., 1983 [4] with their predictions about future cargo ships:

> "In this age of rationalization and automation it would not be difficult to imagine a ship without a crew. [...] It is indeed quite possible that at some distant future date the captain will perform his duties in an office building on shore. In his place he will leave a computer on board the ship which will undertake all tasks of the navigator's art, [...] controlling the ship, and will in fact perform the task much more efficiently."

By describing a captain performing his duties from an office and an onboard computer undertaking all necessary actions to control the ship, seem Schönknect et al.

to be quite accurate with their predictions about future cargo ships. Despite recent advances in autonomous marine vessels, several challenges remain before fully autonomous cargo ships can be a part of the shipping industry.

The marine environment is considered a harsh environment with a dynamic climate that changes its form throughout the day due to, i.e., tide, currents, and waves causing any marine craft to move along different directions and axis—making the docking phase of a ship a complex and high-risk task. A task that demands high precision, well-adopted control algorithms to precisely predict the ship's path and capability to align the ship to the dock. A vital part of this particular task is to feed the ship with reliable position estimates, of which the ship can perform docking in a dynamic environment.

1.2 Objective and research questions

This thesis overall goal is to research methods to connect remote regions using autonomous cargo boats, to bring cargo from location A to location B. Specifically, the thesis has mainly focused on the auto-docking phase. This phase is seen as a limitation and, among others, e.g., path planning, collision avoidance, constraints to make autonomous cargo shipping feasible. Recent applications of USVs have shown increased ease and increased efficiency of cargo shipping at precise locations, e.g., Yara Birkeland. However, further enhancement of the field requires operations over long periods of time, e.g., days or weeks of which the vessel must be precisely docked to be refueled/recharged or for complex cargo handling.

Utilizing vision-based techniques allows USVs to orient themselves to their surroundings from their perspective and may represent an efficient method for such vessels to precisely localize and position themselves in the docking scenario. The work in this thesis is a novel attempt to develop vision-based methods to independently localize USVs locally to a dock in order to optimize the cargo handling process. In addition, the value of the vision-based method presented in this thesis is deployable to be implemented with other types of use cases, e.g., visual controlling of robot arms, visual monitoring of cargo, visual communication to a USV. Thus, based on the thesis objective, the following research questions were formulated:

1. Can traditional vision-based techniques be efficiently utilized to locally localize and estimate the position of a USV relative to a dock?

2. Where are the boundaries of a vision-based navigation system to be used in outdoor conditions?

1.3 Main contributions

The thesis demonstrates how relative low-cost cameras can aid USVs in obtaining a precise relative position of a pier or a floating docking station. The main objective is to develop an independent vision-based positioning system to increase the redundancy and accuracy of autonomous vehicles' navigation systems during the docking phase. Compared to previous work with vision-based navigation for USVs, the method focuses primarily on only using traditional computer vision and filtering techniques to allow the method to be computability efficient to be utilized in real-time systems. The thesis also addresses some challenging weather conditions that a fully developed camera system must overcome in a marine environment.

1.4 The Master‘s Thesis organization

The thesis is structured as follows:

- **Chapter 1:** The motivation and the background of the thesis is introduced. Followed by the objective and research questions, and the main contributions.
- **Chapter 2:** Introduces the reader to essential aspects of docking of marine ships, challenges with autonomous cargo handling, and the current regulations on docking of autonomous vessels with a brief introduction to some aspects on autonomy. Further, related work of existing docking approaches of autonomous ships is briefly assessed. Then, topics on navigation and communication are presented, followed by some theory on relevant sensors for USVs. Then, the pinhole theory is presented, followed by theory on the Kalman filter. Lastly, theory on fiducial markers is thoroughly presented with a description of the method to link these to global position systems.
- **Chapter 3:** Describes the methodology, essential hardware, and required software used in this thesis. Here, details about the different product development stages during the master project are also presented.
- **Chapter 4:** Presents the results from the different scenarios that the vision-based system achieved from the conducted experiments.
- **Chapter 5:** Discusses the diverse topics that have been presented and are reflected upon in relation to the results from the experiments in order to answer the research questions.

- **Chapter 6:** Concludes the thesis. Additionally, the research questions are addressed, and recommendations for further research and work is given.

Chapter 2

Theory

2.1 Aspect of marine docking

The purpose of a pier or a floating dock is to moor a vessel safely and make sure that the vessel does not drift away. Due to the tide, waves, wind, and current, the mooring of a marine vessel is a complicated task, especially if no humans are to be involved in any part of the operation. Consequently, one can divide the dockings in two different types of docks, a stationary pier or a floating dock. What type of dock the vessel meets can play a significant role in the vessel's capability to interact with the dock.

1. **Stationary pier:** The vessel is the only thing that moves throughout the day due to external factors, e.g., tide, currents, waves, and wind. These forces can also tear on the mechanisms involved in keeping the vessel in its place. Specifically, it is only the vessel that will fluctuate in its position relative to the stationary pier throughout the day due to external factors.
2. **Floating docking station:** A floating docking station will follow the sea movements. The vessel and the floating docking station can move in various directions and change their positions throughout the day due to e.g., tide, currents, waves, and wind.

These two possible docking interactions may cause difficulties in designing a solution that can suit all types of vessels. Consequently, different system designs for different types of dockings may be developed for certain types of vessel, but may also play a significant role on how the vessel plans it route to the dock.

For an unmanned surface vessel, a dock is where it is being moored, but probably also where the USV for instance is being charged/refueled, loaded/unloaded with

cargo, or loaded/unloaded with passengers. Therefore, it is important that the USV can easily identify and localize the dock to initiate an automatic docking sequence, in which the vessel can safely maneuver to the dock and in a final stage moor itself. For a human being, this operation may be easy because humans can adjust their actions based on the input from the surroundings in a more efficient way than a robot.

The motions of a marine craft is considered to take place in 6 degrees-of-freedom (DoF), whereas the equations of motions can be derived using the Newton-Euler or Lagrange equations [6]. The three first degrees are defined as the translational motions, heave, sway and surge. Heave is the vessel's vertical motion (up/down), whereas sway is the vessel's linear transverse motion (side to side) and surge is linear longitudinal movements (front/back). The three last degrees are defined as the rotational motions, pitch, roll, and yaw. Pitch is the up and down rotation along its transverse direction (side-to-side or port-starboard), while the roll is the tilting rotation about along its longitudinal axis (front-back or bow-stern) and yaw is the turning rotation of the USVs vertical axis.

Because a USV can move in 6 DoF, the development of automatic latching mechanisms to moor a USV becomes increasingly tricky. This process might be more manageable when working with stationary piers. Floating docks, however, which also can move within 6 DoF may make the process of mooring become more arduous. However, unlike a marine craft, a floating dock is probably permanently anchored. The movements of the floating dock can therefore in most situations be assumed to be less than the movements of a marine craft. This difference between a stationary dock and a floating dock may severely affect the USVs ability to align itself.

2.1.1 A short on vessel control

Vessel control is usually divided into three independent blocks called *guidance, navigation and control* [6]. These systems interact with each other through data and signal transmission. The different blocks can further be extended. *Guidance* is the action or a system that continuously computes a marine craft's desired position, velocity, and acceleration. Essential components for a guidance system are motion sensors, external data such as weather, wave height, current speed, and direction, and a computer to process this data. Navigation is the marine crafts' ability to determine its position/attitude, course and distance traveled. Finally, control, specifically motion control, determines the necessary control of forces and moments to satisfy a specific control objective.

In addition, there are two conditions a marine craft can operate under as defined by [6]:

- **Fully actuated marine craft.** Consequently, a fully actuated marine craft working in 6 DoF must be equipped with actuators that can provide independent forces and moments in all directions.
- **Underactuated marine craft.** In many cases, having a fully actuated vessel is not practical. For example, a marine vessel can be equipped with a single rudder and a propeller, which means the control system cannot satisfy a 6 DoF control objective. However, it is still possible to control the ship with fewer DoF.

2.1.2 A short on the docking process

The objective of a cargo vessel in the docking process is to navigate to the dock safely, latch/lock itself with the dock, load/unload cargo, and then eventually navigate back to the sea. A typical scenario for an autonomous cargo boat can unfold as follows:

1. The cargo boat arrives at the dock and positions itself with sensors and thrusters.
2. The cargo boat latches or secures itself with the dock.
3. The hatches and doors open, allowing humans, or automatic onshore solutions, to pick up cargo.
4. Humans and onshore solutions load cargo onto the cargo vessel.
5. The cargo boat un-docks and starts navigating out of the harbor in to the sea.

It is essential to keep people, materials, and the surroundings safe during all stages. Therefore, the docking process must be repetitive with high precision and accuracy.

USVs usually navigate with GPS in "waypoint mode,", i.e., a predefined path. The waypoints are usually set with global coordinates. A GPS point will only give the vessel information about where the dock is in global coordinates. However, it is crucial for safety and long-term use to have systems locally on the vessel to provide precise relative position estimates about the dock, as if the vessel is to moor to floating jetty, this position will change throughout the day. As for mooring mechanisms to moor a marine craft without any human intervention, it is crucial that the vessel can closely align itself to the same point. Otherwise, the locking

mechanism will probably have to extensively move its position to the vessel to be capable to moor the USV.

Last but not least, it is essential to note that the conditions of a USV or a floating dock can change due to fuel, passengers walking on/off, or cargo being loaded on/off the USV or dock, inducing movements on both objects.

2.1.3 Challenges with autonomous cargo handling

Typically, during the docking process, the cargo crew aligns the vessel to the harbor, ensuring that easy maneuvering to the dock is possible. The captain usually takes manual control over the thrusters and moves the vessel slowly in place, while the deck crew throws mooring lines onshore to the onshore crew to moor the vessel safely. This method is often used by larger vessels. For smaller vessels that only need to make a quick stop to load/unload cargo, aligning the vessel to the dock is usually enough, with thrusters pushing the vessel against the dock. This method reduces personnel and loading time because the whole docking process is much simpler. However, the method is not feasible for cargo vessels to be moored for more extended periods or lifting on/off heavy/complex cargo. Additionally, depending on the schedule, fueling/charging method, the vessel may need to be connected to a precise point on the dock. In short, the following points give a brief overview of the challenges an autonomous cargo vessel must meet in the docking scenario:

- **Navigation:** Because there are many different layouts of harbors and potential blockages, a cargo vessel must be capable of identifying these in order to avoid dynamic and static obstacles. This requires sensors that can give valuable data to the vessel motion control system so that the programmed algorithms can make precise decisions and move the vessel with high accuracy to avoid these.
- **Weather conditions:** Weather can heavily affect the sensors' performance, by disturbing the sensor data. Rain, fog, darkness, glare, heat, coldness, and dew can dramatically decrease the sensors' line of sight. Therefore, the limits of the sensors must be found, in addition to figuring out how well the sensors work in difficult weather conditions to ensure safe autonomous operations.
- **Hacking/spoofing:** The threat of hackers. Manipulating sensors to give erroneous estimates to fool the onboard systems will always be challenging. Cyber-attacks can be done in different ways. In addition, there are also other ways of performing cyber attacks, such as spoofing, jamming, bitstream manipulation, or physically blocking the vessel from entering the dock. Such

attacks can manipulate the vessel to do unwanted maneuvers, harming the vessel, its cargo, or the surroundings.

- **Mechanical design - mechanisms:** In order to make autonomous systems safer, it may also be necessary to understand what effects an autonomous vessel can have on the environment and what effects the environment can have on the vessel. Such as having a mechanism to lock the vessel to the dock. Alternatively, make sensors/or relevant systems perform better.

- **Interacting with the dock:** The loading/unloading of passengers, cargo or fuel may change the vessel dynamics or the position of the vessel, which is a process a mooring solution must handle. For instance, heavy cargo, which is not along the vessel's baseline but on the furthest side of the transverse direction, may impose massive movements on the vessel once these are lifted onto a floating dock.

- **Loss of external communication:** A significant challenge for autonomous vessels is loss of communication with external services during the docking phase. Because communication with external services, especially in a docking phase, is critical for the autonomous vessel to achieve safe dockings. If the vessel loses external communication, it could lose its positioning, thus becoming a danger for itself and its surroundings. Furthermore, if the vessel does not have a backup solution, it could crash into a quay or other surrounding vessels—several possible reasons for such failures, e.g., hacking and software failure.

For recent years there have been autonomous cars, buses, and trucks, in addition to self-supported sailing drones [7], and drones being flown across the globe. However, one significant difference between autonomous vessels and autonomous vehicles is that in the event of loss of communication or propulsion power, a vessel may drift and crash into whatever may be in the vessel's direction. In contrast, a vehicle will stay in the same position.

2.1.4 Current regulations on the docking of autonomous vessels

Due to the rapid advancement of making land-based vehicles and drones autonomous, the shipping industry for transporting cargo has gained more interest in optimizing the way cargo is being transported around the globe. The benefits of autonomous shipping may be numerous and include lower operational costs, greener

transport, and increased safety for both crew and the vessel. There is no longer a question of whether one can make autonomous vessels. The question is now when one can start making feasible autonomous vessels. However, before autonomous vessels can become a reality, certain technologies must be developed further before being implemented safely on vessels. Bureaus Veritas and DNV are two independent classification agencies that set industry standards to make different industries safe. The following note from Bureau Veritas gives an indication of what guidelines such classification agencies believe autonomous vessels must follow [8].

The note divides automation into several degrees. In general, any vessel which is to be fully autonomous can be divided into:

- **A0: Human operated:** Automated or manual operations are under human control. Human decisions and controls all functions.
- **A1: Human directed:** Decision support: system suggests actions. Humans make decisions and actions.
- **A2: Human delegated:** No System invokes functions. Humans must confirm decisions. Humans can reject decisions.
- **A3: Human supervised:** System invokes functions without waiting for human reaction. System is not expecting confirmation. Humans are always informed of decisions and actions.
- **A4: Full automation:** System invokes functions without informing the human, except in case of emergency. The system is not expecting confirmation. Humans are informed only in case of emergency.

This thesis does not go further into the various levels of automation but points out that it is vital to have several solutions that can work together to form a better output in a docking scenario for an autonomous vessel. Having several solutions is better than having one single system to tackle different types of sources of failure. This is especially important if a vessel is to reach level A4: Full automation.

As of for docking and un-docking of vessels, the note from Bureau Veritas says: *"2.2.4 The docking, un-docking, mooring, un-mooring and anchoring operations, as well as the harbor navigation or port approach and the assistance in distress situations should be controlled or remotely supervised in case the degree of automation does not allow a full automation for these operations".*

"2.5.4 When the degree of automation requests supervision during operations in harbor (e.g. docking and un-docking) or heavy traffic conditions near shore, land-based

communication networks should be used to provide a maximum availability and a minimum latency".

"2.6.1 The docking and un-docking procedures should be monitored by sensors (e.g pressure sensors, radar...) to confirm that there are no obstacles for the safe progress".

"2.6.2 A device should be available to stop the sequence of docking or un-docking at any time in the event that the system has not been able to detect a hazardous situation".

As of for cargo on autonomous vessels; the note from Bureau Veritas says in general:

- **2.12 Cargo:** Cargo should be carefully loaded, stowed, and monitored at all times and for all operations. The stowage of the cargo should be ensured at port, since the vessel could have few or means (less or no crew and equipment) to ensure proper cargo securing at sea.
- **2.9 vessel status and dynamics:** Cargo monitoring should be a part of the vessel status
- **5 Cargo management system:** In general, have cargo management system that overall ensures that the cargo does not compromise the safety of the vessel or not degrade the environment.

Following the guidelines, a final docking sequence for a fully autonomous vessel will require a robust system made up of several subsystems working together. As for other regulations, a final fully-automated vessel would also have to follow the regulations in Norway and internationally, the rules of the road at sea ("Sjøveisreglene") [9], the maritime code ("Sjøloven") [10], the harbour act ("Havne- og farvannslovene") [11], the international convention of life at sea (SOLAS), and convention on the International Regulations for Preventing Collisions at Sea (COLREG) [12]. These regulations cover topics that a final fully autonomous system will follow.

2.1.4.1 Positioning requirements for autonomous vessels

Research by C. Kooij et al. from 2018 explores the question of when autonomous vessels will arrive and tries to predict the threshold of certain technologies necessary to make vessels fully autonomous [13]. The study looked at navigation, diverse propulsion fuels, and the cost of making the cargo handling of vessels fully automatic. In short, their findings for navigation were that for general navigation at sea, an accuracy of 10 m was required, but in ports, the absolute accuracy needs to be at

2.5 m, and for an automatic docking sequence, an absolute accuracy of 0.10 m is required. These values are only based on GPS accuracy and do not say anything about IMU, cameras, lidars, proximity sensors, or other types of sensors could be a part of the solution to better give accurate position data.

Research from Smart, 2013 [14] have shown that a requirement of 0.01 meter in general horizontal positioning accuracy is needed to keep the integrity in a docking sequence until the system would have to alert an external operator in an unwanted event. However, DNV GL [15], states in their guidelines for autonomous and remotely operated vessels that vessels must reach an absolute positioning accuracy of 0.1 meter with 95 % probability during the docking phase, confirming C. Kooij et al. research. It must be mentioned that Bureau Veritas has currently not set any positioning requirements in their guidelines for autonomous shipping [8].

2.1.5 The Autonomy Engine

Marine autonomous systems that can change their behavior due to unanticipated events during an operation are called "autonomous" [16]. As described earlier, for a marine system to be fully autonomous, it must be capable of detecting various scenarios, understanding these, and acting on the data the system has at that particular time. This implies that the system must have several sensors for sensing different scenarios and simultaneously analyze the data the system is retrieving and act on this information. To make autonomous systems safe; redundant solutions must be developed if one sensor/capability is lost during the operation.

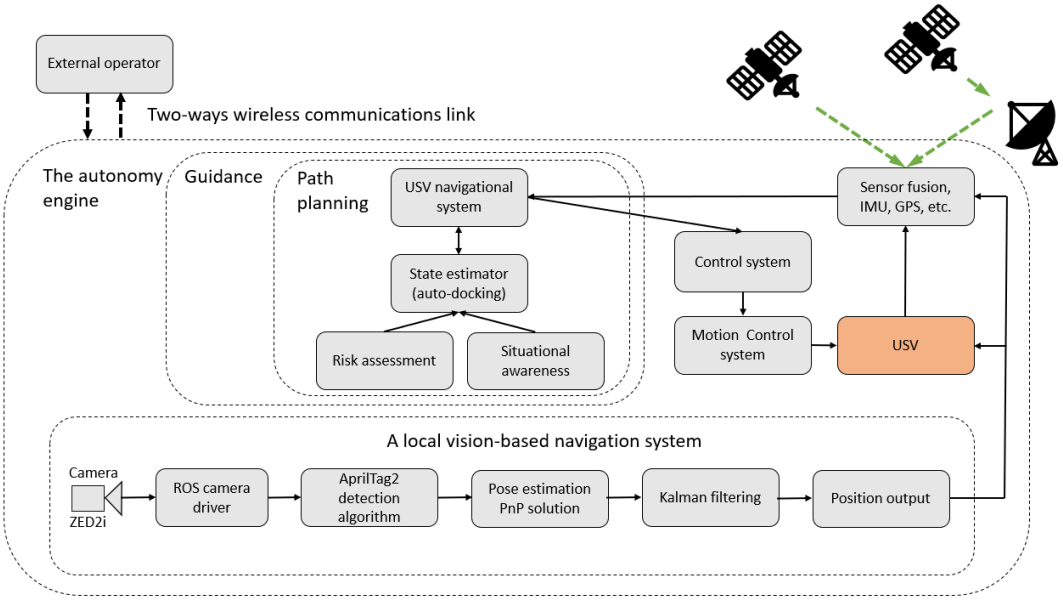


Figure 2.1: High-level sensor and software system block diagram, with a potential vision-based navigation approach to fed the navigation system with position estimates.

Figure 2.1 shows an illustration of a possible layout of an autonomous system with auto-docking of the vessel. A vision-based approach is also included to feed the navigation system with position estimates. Here, guidance, navigation, and a control system with auto-docking are presented. Concepts such as situation awareness, risk assessment, path planning, sensor fusion, the external operator are also presented in the layout. A short note on these concepts:

- **Situational awareness** is crucial for the system to identify dangerous threats as soon as possible to perform any operation safely. Being aware of the surroundings will make the vessel capable of changing its behavior in relation to the environment and to notify the remote control in the case of an unwanted event.
- **Risk assessment** is to be capable of assessing what risk different situations possess. Thus, making the safest choice in that situation.
- **Path planning,** the vehicle capability to plan a route from A to B. There has been much work done in this field in recent years, developing more intelligent and more efficient methods for vessels to plan more optimized and robust trajectories [17]. However, such systems rely on precise position data from the navigation systems, and in scenarios where GNSS or vital position data are not available, such methods may no longer function properly. It is critical to have several solutions from which the path planning methods can derive

information to address this issue. In addition, in the event of a failure, one example being the loss of external communication, it is essential to have local on-board systems that these path planning methods can work with to make autonomy more robust.

- **Sensor fusion** is the ability to bring together inputs from multiple sources of sensors, for instance, radars, cameras, and lidars, to form a single model or an image of an environment. By fusing data from different sensors, one may get more insight into the environment or more precise data.
- **External operator** is personnel ready to take control in situations where the vehicle is not capable of taking action itself—implying that the vessel must have a strong communication link to the remote control room.

2.2 Related work: Demonstrated marine dockings

2.2.1 Rolls-Royce and Wärtsilä

In 2018 Rolls-Royce demonstrated a fully autonomous trip with a large ferry from A to B, including a fully automated docking sequence without any human intervention[18]. The operation was carried out between Parainen and Nauvo in Finland. At around the same time, another company called Wärtsilä demonstrated an auto-docking sequence of a passenger ferry in the fjords of Norway [19]. These two companies demonstrated that with the current technology available are fully capable of docking larger passenger vessels. However, a captain must be on-board ready to take control if an unpredictable failure happens.

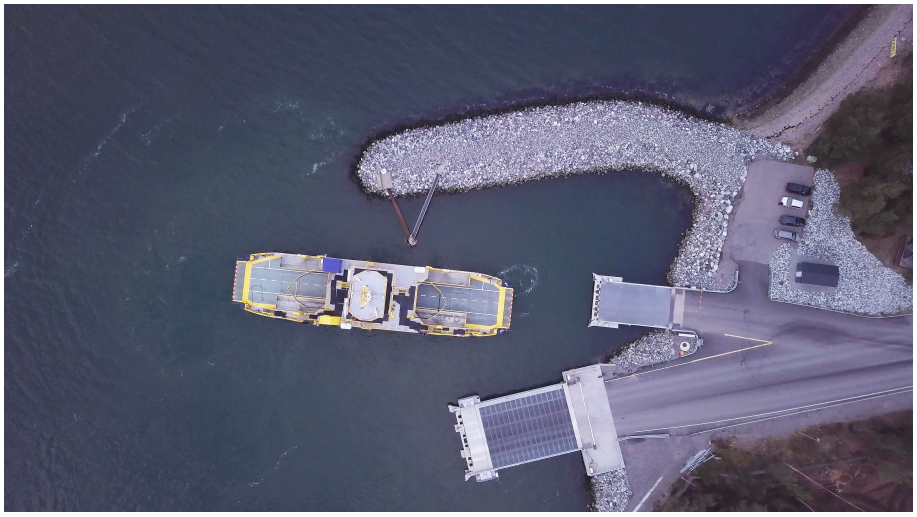


Figure 2.2: The 3. December 2018 Rolls-Royce demonstrates the world first fully autonomous ferry. Image courtesy to [18].

Another company that has demonstrated automatic docking is Volvo, with the Volvo Pentas system. This system is developed for smaller vessels and is designed to assist the boat driver [20] during the docking phase. The system uses sensors onboard the vessel and on land to assist the boat driver. Here, as with the systems from Rolls-Royce and Wärtsilä, a captain must be present in case of any failure.

Figure 2.3: milliAmpere2 in the docking phase. Developed by NTNU as a research project for autonomous system. The overall goal of the project is to develop intelligent solutions to bring passengers from place a to a place b.

The milliAmpere2 is another vessel currently designed to reach autonomy level A3: Supervised Current regulations on the docking of autonomous vessels, and later to reach autonomy level A4. The vessel is currently on the research stage and uses RTK GNSS fused with an IMU integrated into an INS as the primary navigation source and near distance sensors to give accurate distance measurements to the floating dock. When writing this thesis, the lidars, the radar, and the cameras are not used in the docking sequence. The vessel has four azimuth thrusters, giving the vessel great maneuverability. In short, the docking sequence for milliAmpere2 can briefly be described as follows:

1. **Approach - (navigation)**, the milliAmpere2 navigates to the dock with 0.50 - 2.00 m distance from the dock, with ca. 0.01 - 1.00 m precision (data not confirmed). Depending on the accuracy of the GNSS system.

2. **Docking - (sensing the dock)**, uses near field sensors to aid the navigation system to precisely estimate the distance to the floating dock.

3. **Mooring - (pushing the vessel against the dock)**, the docking sequence initiates once the near field sensors sense the dock. The milliAmpere2 moves slowly to the dock and makes a small controlled ”crash” into the dock while using propulsion power to keep the vessel at position.

4. **Un-docking - (rotating the thrusters in the opposite direction)**, once the loading/offloading of the passengers are done. Then the un-docking sequence initializes, which rotates the thrusters before the thrusters push the vessel to the other side of the channel.

This approach to dock the vessel is considered the simplest method to moor a vessel for a shorter time automatically. The method avoids the problem of working with mechanisms that latches and secures the vessel in place, a process that may require a higher level of positioning accuracy. Another advantage of the method is that it allows the passengers to board and depart quickly, making the transit time shorter for the passengers. Development of latching mechanisms to secure the milliAmpere2 is under development to lock the vessel to a docking station in an attempt to allow the vessel to be shut down and e.g., charge the onboard batteries for more extended periods without any human intervention.

2.2.2 Vision-based docking

In 2019, Ø. Volden researched whether vision-based systems could be utilized to successfully dock a small unmanned vessel. His thesis explored different "vision-based" techniques and whether these could be used to estimate relative positions between a dock and a vessel. In his research, he used lidars, stereo-cameras, and mono-cameras in order to recognize fiducial markers [16]. One of his approaches was to use deep learning techniques to recognize fiducial markers on land, acting as markers intended to be identified and classified, which the vessel then could use to orient itself with. The main contribution of this research was to develop systems/techniques to position the vessel in events of loss of external communication, making the navigational system more redundant and more robust, thus making the vessel fully self-supported in a docking sequence.

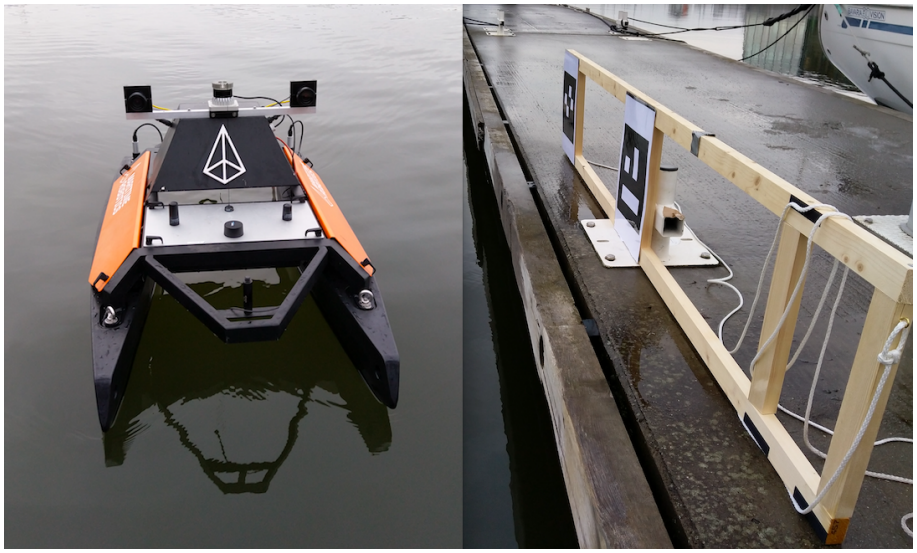


Figure 2.4: Illustration of Ø. Volden experiment with a Otter USV and fiducial markers on the pier. Image courtesy to Ø. Volden [16].

The research results showed that vision-based systems could be utilized to get position estimates to align the vessel. The overall results of the experiments showed that lidars, mono- and stereo-cameras could be utilized to get position estimates relative to a pier/dock. However, localizing a dock with a lidar can be quite computationally heavy. It may also be challenging to extract the needed information from the lidar to localize or use the information to precisely get accurate positional estimates of the dock. However, a significant benefit of using lidar is that one may get precise positioning information about the surroundings, the shape of possible obstacles, and the distance in varying weather conditions. The thesis compared mono- and stereo-camera capabilities, and the main findings were that the mono-camera delivered almost equally the same performance as the stereo-camera.

2.2.3 Roboat

Roboat is another company that is developing smaller autonomous vessels. Their primary focus is developing urban solutions to transport cargo and passengers and is researching on new possible usages of urban autonomy[21].

Figure 2.5: Roboat docking platform using fiducial markers. Image courtesy to [21].

In the figure 2.5 one can see the Roboat using fiducial markers to create a local coordinate system on the docking. The Roboat uses this system to locally localize the docking. This method may give the vessel capability to move to the same position relative to the docking. As a part of the research prior to Roboat, the team also researched latching mechanisms to latch several vessels together and develop methods to latch the vessel to a dock. The purpose is to develop autonomous systems in the Amsterdam canals. Figure 2.6, shows a fiducial marker, specifically,

AprilTags, in use to create a local reference coordinate system on each vessel. These tags offer 6-DoF position estimates between the vessels, which comes in handy when the vessels can have motion in around six axes. Moreover, Mateos et al. [22], mentions that by using fiducial markers as local reference systems, one can minimize the impacts on canal walls, additionally one does not need any power supplies to power the markers. However, localizing such markers in the dark can thus be a challenge for a camera system.

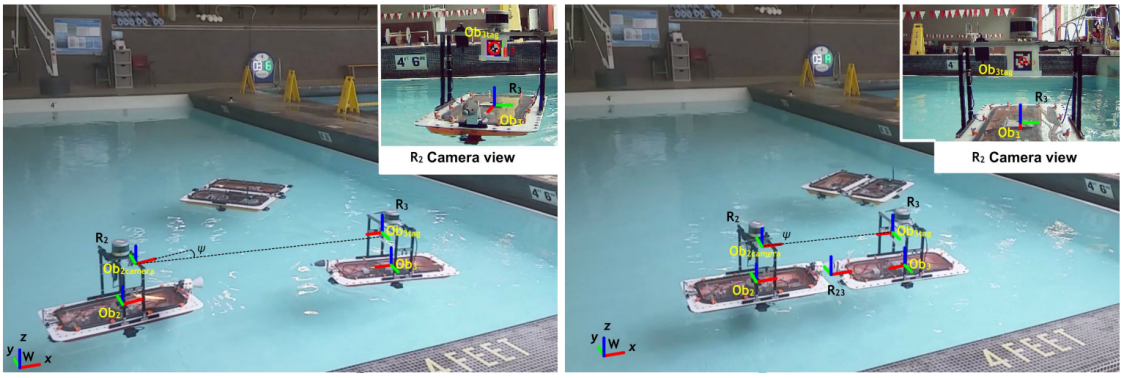


Figure 2.6: Roboat latching mechanism using AprilTags to give each USV orientation estimate about each other. Roboat uses the same approach as a part of the solution to latch to a dock[22].

2.3 Navigation and communication

Navigation and communication are essential technologies for making unmanned vessels feasible. Navigation is essential for a USVs ability to determine its location within an environment and determine its path to a final destination without any human intervention. There are different types of technologies the USV can use to navigate, but USVs mainly use GNSS signals fused with an onboard IMU integrated into an INS to position themselves and navigate from A to B. Therefore, this thesis will briefly introduce some critical navigation and communication technologies.

2.3.1 Global Navigation Satellite System (GNSS)

The Global Navigation Satellite System (GNSS) is a system developed to provide geolocation and time information to a GNSS receiver anywhere on or near the Earth. Global Positioning System, GPS, is, for instance, a part of the GNSS system. For a GNSS system to fully work, there must be an unobstructed line of sight to four or more GNSS satellites or any base stations. Obstacles such as buildings, mountains, or GNSS jamming devices can block or weaken the GNSS signal. Onboard marine vehicles, there is also a potential for internal interference with onboard equipment [23]. These sources of interference can result in no or poor geological position accuracy. To increase the accuracy and reliability of GNSS signals, one can set up a base station or several base stations with a known position. This method is called real-time kinematic positioning (RTK) positioning, which under certain circumstances can obtain centimeter precision [24].

2.3.1.1 RTK GPS (Real Time Kinetic Global Positioning System

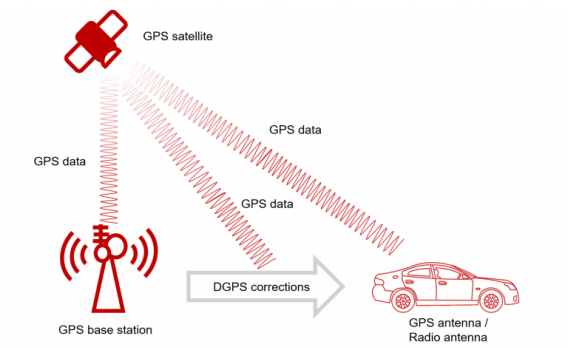


Figure 2.7: Concept illustration of how RTK GPS works. Illustration courtesy to [25].

RTK GPS provides higher positioning accuracy for systems that demands higher position precision. For safe operations, utilizing RTK GPS for autonomous vessel can be seen as vital part of the USV navigation system. With one or several RTK ground station, an information flow through the Ultra High Frequency (UHF) band exist. RTK GPS are designed to give centimeter precision when it is correctly setup. Figure 2.7 gives an brief illustration of the setup. RTK GPS was later used in the experiments carried out in this thesis as a "ground-truth" to compare the vision-based position estimates to global position estimates.

2.3.1.2 Other systems

Ultra Short Baseline positioning (USBL) and Long Baseline positioning (LBL) are examples of two other systems, primarily used in underwater positioning systems for Remotely Operated Vehicles (ROVs). With a USBL system, the vessel sends out an acoustic impulse (ping) received by an ROV transponder. This system is capable of giving distance and bearing position estimates. An LBL system is based on having beacons on the seabed, and the ROV position is triangulated based on these beacons.

2.3.2 The North-East-Down (NED) coordinate system

GPS coordinates describe positions on Earth. These coordinates can be expressed in several ways. In order to compare estimated camera positions relative to GPS positions, this thesis uses the North-East-Down (NED) coordinate system. This is because this coordinate system is defined as a local tangent plane to the Earth's surface, relative to a specific geographical position, defined by latitude, longitude, and altitude. Furthermore, the coordinate system is defined with the first dimension pointing North, the second pointing East, and the third axis pointing Down towards the center of Earth. This thesis will evaluate the results using this coordinate system to precisely compare the results between the camera detections and the achieved GPS positions. The calculations converting from the global system to the local system, NED, are done with the Python module pymap3d.

2.3.3 Inertial Measurement Unit (IMU)

IMU is a device made up of a three-axis gyroscope, three-axis accelerometer, and, sometimes, three-axis magnetometers, as well as a one-axis barometric pressure (altitude) sensor [6], used to measure forces, angular rate, and orientation of a

system. These devices usually deliver a precise estimate about a system's orientation and play a central role in aiding a navigation system for manned- or unmanned vehicles to provide heading and positioning information. However, a major challenge with IMUs is that they can quickly lose accurate position data. IMUs typically suffer from accumulated error, also called "drifting." This comes from the fact that the IMU's positioning data continue integrating the acceleration estimations to calculate velocity and position. Due to this integration, one gets a measurement error that is accumulated over time. This so-called drifting is an ever-increasing difference between where the system thinks it is located and the actual location.

As for autonomous vessels' ability to navigate from one target to another, one of the most important systems on-board is the GPS fused with an IMU. The benefit of having a GPS fused with an IMU, which gives global and time information, is that the combination of the two systems often can give an exact position estimate. To avoid or to keep the accumulated error in the IMU at the lowest; the IMU is corrected by the GPS position estimates. A significant weakness of this system is its dependency on external communication. If the GPS signal is blocked or obstructed, the vessel can become fully inoperative. In a docking situation, a loss of external communication could impose a significant threat as the vessel would not be able to position itself.

2.3.4 Visual and odometry navigation

Odometry uses motion sensors to determine the vessel's position over time and localize itself within the environment. This allows the vessel to orientate itself within the environment from its perspective. Such sensors can, for instance, be mono- or stereo-cameras, which gives a position estimate about its surroundings. A great benefit of using visual aid and odometry in robotic navigation is that cameras can generally give more information about their surroundings than what an IMU or a GPS can. In addition, cameras are less expensive than standard navigation systems, and utilizing machine vision to navigate USVs may be an intelligent method to achieve safer autonomous operations. A camera can, for instance, calculate distances and determine what objects are in the field of view to give essential data to a navigation system. However, vision-based navigation systems can easily be obstructed or interfered with, and therefore it has their weaknesses.

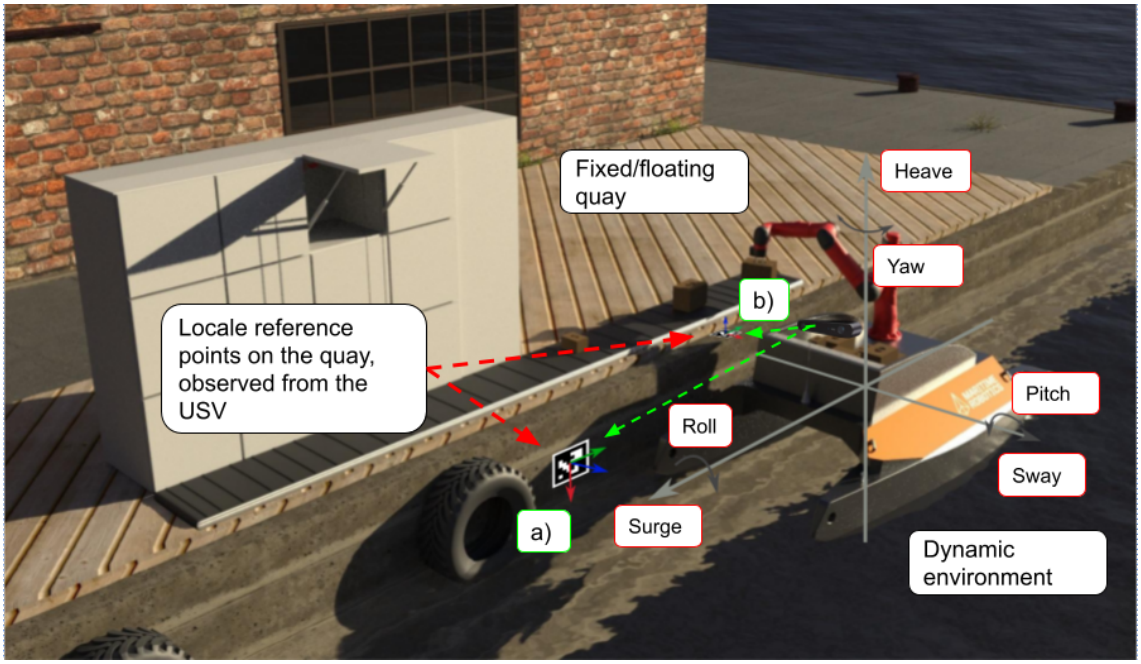


Figure 2.8: Concept illustration of a Otter USV in a docking scenario with fiducial markers, which can give 6-DoF position estimates of the dock to the USV. Here, the Otter USV can use marker a) to get precise position estimates and necessary information for the USV to preciserly align itself with the dock. While marker b) can be used by a robot arm to put cargo on the dock no matter if the dock is a floating/or a fixed dock.

Figure 2.10 illustrates a concept of using fiducial markers to create a local coordinate system on the docking. These markers can create a 6-DoF local coordinate system on each marker, these reference systems can also be used to give the USV estimates about its orientation in 6-Dof relative to the marker. Which may make it more convenient for underactuated and fully actuated vessels to orient themself to a pier or a docking station. Marker a) can, for instance, be used to locally create a reference system to feed the USV with local positions of the dock, while marker b) can be used for instance by a onboard robot arm to put cargo on dock safely.

2.3.4.1 SLAM

SLAM stands for simultaneous localization and mapping and is a computational system for constructing and updating a map of an unknown environment - while also keeping track of a robot's position in a map. For several years algorithms and methodologies have been developed to optimize this system. SLAM uses several different sensors to construct a map and orientate a robot within the map. Consequently, if one sensor fails, the system should still keep functioning.

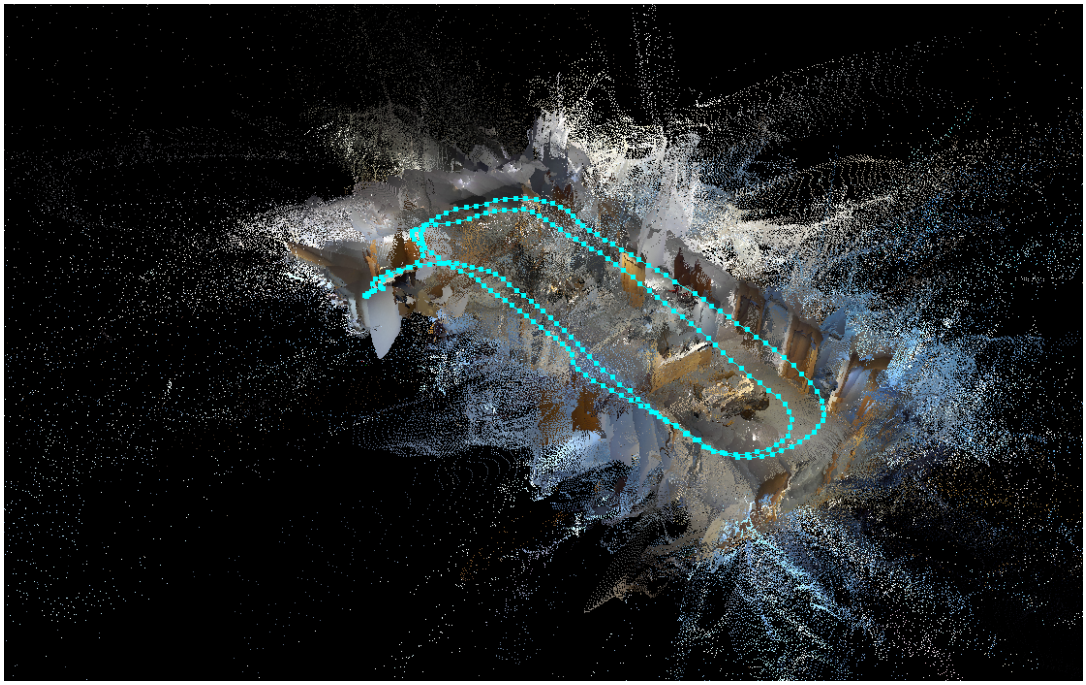


Figure 2.9: Illustration of a point cloud of the Safe Autonomous ships (SFI) Autoship office, created with the ZED2i camera using its built-in IMU and stereo-camera capability.

Figure 2.9 shows a visual SLAM map created with the ZED2i camera and the ZED API with ROS. The map is created by using the ZED2i built-in IMU and its stereo vision capability. However, it must be specified that this SLAM algorithm heavily depends on the camera built-in IMU and uses odometry to build a point cloud that uses "loop closure" to correct the IMU. This means the underlying algorithms recognize features from the captured images and correct the camera's built-in IMU position to avoid drift. With an accurate camera position, it is possible to create an accurate 3D map of the surroundings seen by the camera. The loop closure is performed using depth and visual information, and relays on the environment to be mapped is colorful. Such methods are considered computatively costly.

2.4 Sensors

In its broadest definition, a sensor is a device that can detect changes in its environment and send information to other electronic devices. This section will briefly introduce relevant technologies that could be utilized in the docking phase of USVs.

2.4.1 Distance Sensors

Sensors that can sense the distance of an object without being in contact [26] is often associated with being called distance sensors. Such sensors can be inductive, capacitive, ultrasonic, laser, or IR-based. The advantage of such sensors is that they can robustly calculate the distance to particular objects with relatively high accuracy, in varying conditions, and are quite cheap. On the other hand, providing more information about the surroundings may be limited to such sensors. For robotic applications, such sensors are usually necessary for specific tasks.

2.4.2 Lidar

Compared to Electro-Optical and IR cameras, a lidar is not passive and has similar features as certain types of distance sensors in that Lidar is laser-based and emits a pulse reflected by the surroundings. The time from the pulse is sent to its return is used to create a map of the surroundings. Some lidars spin a beam in a circle, which emits a pulse at regular intervals, while others can emit a pulse within a smaller range. Lidars for building maps of their surroundings are precise because of how a lidar works. Using filtering methods, materials and unwanted noise, such as rain and snow, can be filtered away with Lidar technology.

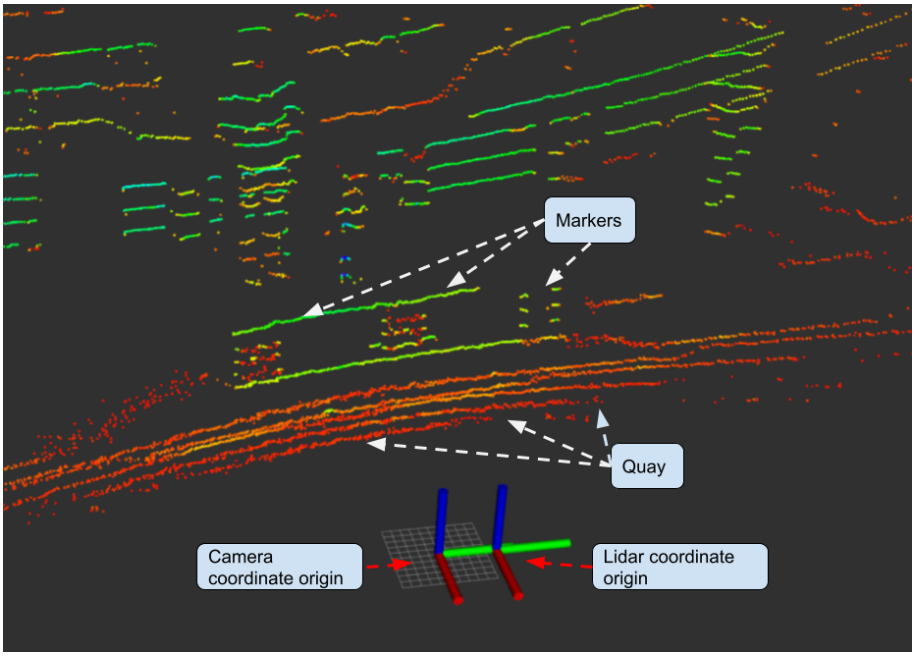


Figure 2.10: Point cloud image of a docking scenario produced by a lidar. A pier and three fiducial markers can be seen in the image. Image courtesy to [16].

A considerable disadvantage with lidar technology is that the technology is currently considered costly and computatively demanding. Estimating orientation for robots, e.g., 6-DoF, from Lidar data may be challenging. Lidars are excellent at giving precise position estimates about their surroundings, even in more demanding weather conditions, but are currently limited to giving orientation estimates to a robot. Recent applications of Lidars are usually for collision avoidance, recognizing particular objects, and distance measurement between the lidar and the object along with a few directions.

2.4.3 Radar

Radars are designed to emit high-intensity radio waves in pulses spinning in a beam around 360 degrees. These radio waves can travel long distances until they reach an object that reflects the radar signal. This method gives a precise position estimate and an estimate of the object's shape. Radars are of great use in many different fields, including the naval and aviation field and meteorology, where it is being used for weather forecasting. Compared to lidars, a radar sends a pulse with a longer wavelength, resulting in a greater range and the ability to penetrate heavier objects, e.g., snow, birds, fog, which can be filtered away.

2.4.4 Infrared camera

Infrared cameras (IR), also called thermal cameras, measure heat energy in the sensors' field of view. This heat energy is defined as light waves with a wavelength of 1000 - 1400 nm, while conventional cameras can detect lights in the 400 - 700 nm spectrum. This difference allows IR cameras to distinguish objects such as living creatures because they emit(s) heat to their surroundings. An important note about IR cameras is the way they measure resolution. Optical cameras count pixels to measure resolution, while IR cameras usually follow a Johnson criterion by estimating the number of line pairs across a target [27]. In practice, this criterion, combined with the fact that IR cameras measure longer wavelengths, makes thermal cameras better at detecting specific targets from further distances than conventional optical cameras.

2.4.5 Electro-Optical Camera

Electro-Optical (EO) cameras are passive sensors that capture visual images of scenes in front of them and work quite similarly to how IR cameras work. At a basic level, electro-optical cameras measure the reflected light emitted from a light source in the 400 - 700 nm spectrum (lenses and filters can affect this range). The images are then built up by zeros and ones in an array containing information about the colors captured and where they are present. This array of information makes it possible to extract, recognize, and estimate distances to the objects captured by the camera. There are several methods for recognizing features in images, such as the classical techniques that sort information based on colors and then recognize the geometrical shapes, to learning-based approaches (i.e., deep neural networks).

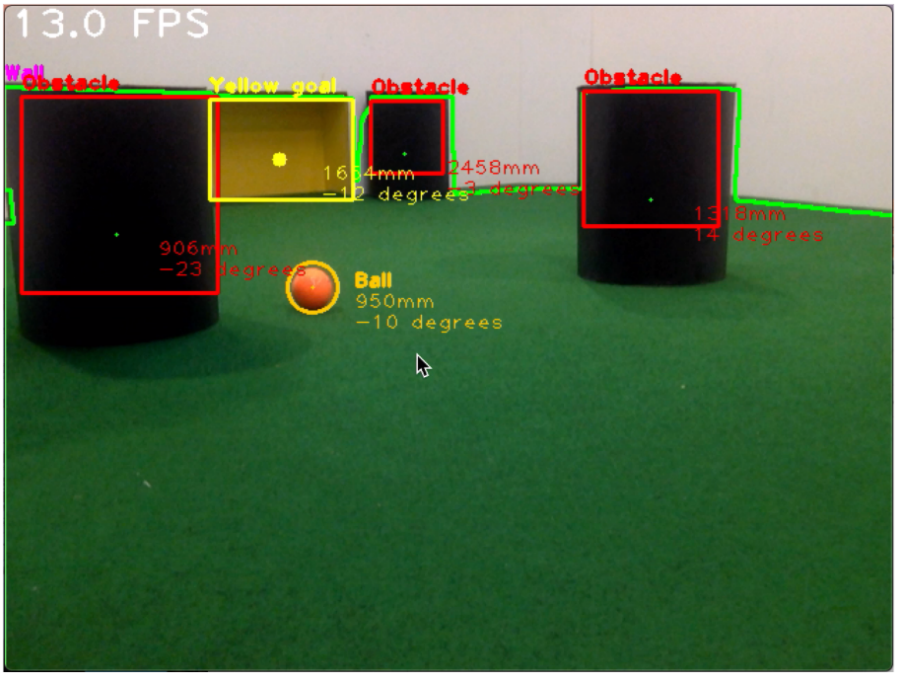


Figure 2.11: Object detections with position estimates with traditionally computer-vision techniques. Used for an small soccer robot to navigate and shoot a ball into a goal. Detection of ball, goal, obstacles and wall is based on the each objects colour. The system were run on a raspberry pi 2. The navigation between the obstacles where done by applying a potential field telling the soccer robot where it potential were a obstacle. Developed by the author at the Queensland University of Technology during his exchange in 2018.

Cameras can be a significant source of information for an autonomous cargo vessel. However, utilizing all the information a camera can capture might be difficult. Much research has been done on this topic, and more research will be carried out in the future.

Despite the benefits cameras possesses, the main drawback that affects the performance of a conventional camera is the amount of light in the scene which can be captured. In the darkness, there is less light which the camera sensor can capture, and therefore less information about the scene will be captured. This results in lower detailed images with less information that can be extracted. There is no information to be retrieved in absolute darkness, and a conventional camera may become useless. The information particularly influences the Vision-based algorithms a captured image possesses to extract any context. Changes in the light, shadows, motion blur, texture, rain, snow, and fog are environmental conditions that influence a camera's performance. Camera-based algorithms will therefore have problems in such situations.

2.5 Pinhole camera geometry theory

The pinhole camera model is a mathematical simplification of the relationship between 2D image coordinates and 3D coordinates, and its projection onto the image plan of an *ideal* pinhole camera.

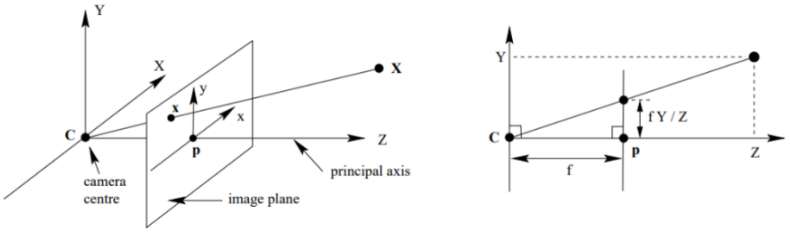


Figure 2.12: The left part illustrates the pinhole camera model, while the right part maps points to the image point. Illustration courtesy to [28].

In figure 2.12, a 3D point with the coordinate $\mathbf{X} = (X, Y, Z)^T$ is mapped to an image point where a line join the point $\mathbf{x}$ to the center of where the projection meets the image plane. Due to similar triangles, the point $(X, Y, Z)^T$ can be mapped to the point $fX/Z, fY/Z, f)^T$ on the image plan. As the resulting image coordinate represents a constant distance between the camera and center and the image plane, one can therefore ignore this constant that gives the following relationship:

$$\mathbf{X} = (X, Y, Z)^T \rightarrow (fX/Z, fY/Z)^T \tag{2.1}$$

which describes the relationship from 3D world coordinates to 2D image coordinates. The equation 2.2 can further be extended with homogeneous coordinates and be written as the matrix:

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \rightarrow \begin{bmatrix} fX \\ fY \\ Z \end{bmatrix} = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} = diag(f, f, 1)[\mathbf{I}^{3x3}\mathbf{0}^{3x1}] \begin{bmatrix} X \\ Y \\ Z \\ 0 \end{bmatrix} \tag{2.2}$$

In equation 2.2, the $diag(f, f, 1)$ is the diagonal matrix, $\mathbf{I}^{3x3}$ is the identity matrix and $\mathbf{0}^{3\times1}$ is a 3×1 zero vector. An element $\mathbf{X}$ can now be introduced as the world point, given by the coordinate vector $(X, Y, Z, 1)^T$ and $\mathbf{X}_c$ for a corresponding point in the image plane represented by a homogeneous coordinate 3-vector. Further, the matrix $\mathbf{P}$ is being introduced as the homogeneous 3×4 camera projection matrix. Equation 2.2 can compactly be written as,

$$\mathbf{X}_c = \mathbf{P}\mathbf{X} \tag{2.3}$$

where

$$\mathbf{P} = diag(f, f, 1)[\mathbf{I}^{3x3}\mathbf{0}^{3x1}] \tag{2.4}$$

The transformation done in equation 2.2 assumes that the origin of the image plane coincides with a principal point, meaning the point where the principal axis intersects an image plane. In practice this results in,

$$(X, Y, Z, 1)^T \rightarrow (fX/Z + p_x, fY/Z + p_y)^T \tag{2.5}$$

where $(p_x, p_y)^T$ are the coordinates of the principal point. With a principal offset, this equation can be written in homogeneous coordinates as,

$$\begin{bmatrix} fX + Zp_x \\ fY + Zp_y \\ Z \end{bmatrix} = \begin{bmatrix} f & 0 & p_x & 0 \\ 0 & f & p_y & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} = \mathbf{k}[\mathbf{I}^{3x3}\mathbf{0}^{3x1}] \begin{bmatrix} X \\ Y \\ Z \\ 0 \end{bmatrix} \tag{2.6}$$

where

$$\mathbf{K} = \begin{bmatrix} f & 0 & p_x \\ 0 & f & p_y \\ 0 & 0 & 1 \end{bmatrix} \tag{2.7}$$

is defined as the camera matrix. The last step involves a change of coordinates, from meters to pixels, which can be performed with the pixel density of the imaging sensor. The number of pixels per unit distance in image coordinates along horizontal and vertical directions can be defined as $m_x = n_x/s_x$ and $m_y = n_y/s_y$. n_x, n_y, s_x and s_y, are in this case representing the number of pixels in the imaging sensor and the physical size of the sensors in meters, respectively. The camera matrix is further used to calculate the orientation, pose, of fiducial markers and map the fiducial markers into global 3D coordinates.

2.6 The Kalman filter

Kalman filtering is a filtering method that provides estimates of some unknown variables measured and observed over time. The filter is named after Rudolf E. Kalman, who was one of the primary developers of the theory. Some of the first filter equations were formulated and published during the 1960s. The method is widely used in numerous applications, but particular aircraft, spacecraft, and ships are positioned dynamically. For example, filtering and fusing GPS signals and IMU data to obtain a more precise position estimate. This thesis will go briefly into the Kalman

filtering algorithm, formulated by Y. Kim and H. Bang [29] and implemented with the pykalman package [30], briefly based on the implementation from [31]. The Kalman filter can be modeled with the following equation 2.9:

$$x_k = \mathbf{F}x_{x-1} + \mathbf{B}u_{k-1} + \mathbf{w}_{k-1} \tag{2.8}$$

where $\mathbf{F}$ is the state transition matrix applied to the previous state vector $\mathbf{x}_{k-1}$, $\mathbf{B}$ is the control input matrix applied to the control vector $\mathbf{u}_{k-1}$, and $\mathbf{w}_{k-1}$ is the process noise vector assumed to be zero-mean Gaussian with the coveriance matrix $\mathbf{Q}$, i.e $\mathbf{w}_{k-1} \sim \eta(0, Q)$. The process model is paired with the measurement model that describes the relationship between the state and the measurements at the current time step k as:

$$z_k = \mathbf{H}x_k + \nu_k \tag{2.9}$$

where z_k is the measurement vector, H is the measurement matrix, and ν_k is the measurement noise vector assumed to be zero-mean Gaussian with the coveriance matrix R, i.e., $\nu_k \sim \eta(0, R)$.

The purpose of the Kalman filter is to provide a $\mathbf{x}_k$ at time given an initial estimate of $\mathbf{x}_0$, the series of measurement, $z_1, z_2, \ldots, z_k$, and the information of the system described by F, B, H, Q, and R.

The Kalman filter algorithm can be expressed of two stages: Prediction and update. The hat operator, $\hat{}$, means an estimate of a variable, and the superscripts $-$and $+$ denote predicted (prior) and the updated (posterior) estimates, appropriately. The prediction state estimate is defined as:

$$\hat{\mathbf{x}}_{\mathbf{k}}^{-} = F\hat{\mathbf{x}}_{\mathbf{k-1}}^{+} + B\mathbf{u}_{\mathbf{k-1}} \tag{2.10}$$

and its predicted error covariance,

$$P_k^{-} = FP_{k-1}^{+}F^T + Q \tag{2.11}$$

The update stages are then defined as,

$$\tilde{y}_k = z_k - H\hat{\mathbf{x}}_{\mathbf{k}}^{-} \tag{2.12}$$

$$K_k = P_k^{-}H^T(R + HP_k^{-}H^T)^{-1} \tag{2.13}$$

$$\tilde{\mathbf{x}}_{\mathbf{k}}^{+} = \tilde{\mathbf{x}}_{\mathbf{k}}^{-} + K_k\tilde{\mathbf{y}} \tag{2.14}$$

$$P_k^{+} = (I - K_kH)P_k^{-} \tag{2.15}$$

Equations 2.12, 2.13, 2.14 and 2.15 are defined as the measurement residual, Kalman gain, updated state estimate and the updated error covariance, respectively. The predicted state estimate is evolved from the updated previous state estimate. In the update stage, the measurement residual $\tilde{y}_k$ is computed first. The measurement residual is the difference between the true measurement, z_k, and the estimated measurement $H\hat{x}_k^-$. The filter estimates the current measurement by multiplying the predicted state by the measurement matrix. $\tilde{y}_k$ is later then multiplied by the Kalman gain, K_k, to provide the correction $K_k\tilde{\mathbf{y}}_\mathbf{k}$ to the predicted estimate $\hat{\mathbf{x}}_k^-$. Once the updated state estimate has been calculated, then the error covariance, p_k^+, is calculated to be used in the next time step.

To properly function, the Kalman filter needs an initial value, $\hat{\mathbf{x}}_0^+$, and an initial guess of the error covariance matrix, P_0^+. Lastly, the Kalman filter implements the prediction and the update stages for each time step, $k = 1, 2, 3, ...$, after initializing the estimates. Importantly, the Kalman filter is derived on the assumption that the process and measurement models are linear and be expressed with the matrices F, B, and H, whereas the process and measurement noise are additive Gaussian.

2.7 Fiducial markers

2.7.1 Why use fiducial markers?

To use vision-based detection systems to locate a dock, one must have distinguishable features on the dock that the vision system can recognize. Such features can be specific markers that distinguish themselves from the environment, making it easier for the vision system to identify and estimate the orientation. There may be many visual features a vision-based navigation system could detect and try to estimate its position from. Briefly, these can be summarized as:

- **Natural landmarks**, such as already installed marine lights or features on the specific dock, which a vision system can be trained or programmed to detect. The challenge here is that features can come in any shape, and there is no concise way natural markers can be described for a vision system. Therefore, making it hard to create a robust algorithm that could be utilized.
- **Fiducial markers** is another option, which is a standardized marker compared to natural landmarks, developed for robotic applications to give robots 6 DoF orientation estimate to their surroundings. Such markers are pre-defined with a specific shape, have an internal code, and developed to be easily detected by a camera.

Fiducial markers are designed for robotic vision applications and provide 6-DoF orientation estimates. Additionally, fiducial markers may also be a cheap option to realize autonomous cargo transport to connect remote places because making a local reference system with fiducial markers may make it feasible to dock smaller USVs on floating docking stations. Consequently, fiducial markers were chosen to explore how well such markers perform in marine environments. Furthermore, utilizing fiducial markers to get 6 DoF position estimates for a USV may be more efficient to precisely control a USV to a dock. It may also open up the possibility of designing mechanisms to lock the vessel or other complex systems to interact with the USV.

2.7.2 What is a fiducial marker?

A fiducial marker is an object meant to be placed within the field of view of a camera, designed for robotic applications, that is computationally effective and can provide position and orientation data to a robot. A fiducial marker is composed of an external black border and an inner region that encodes a binary pattern. The

inner region is unique and identifies each marker, consisting of more or fewer bits. The figure 2.13 shows a brief overview of different types of fiducial markers developed in recent years.

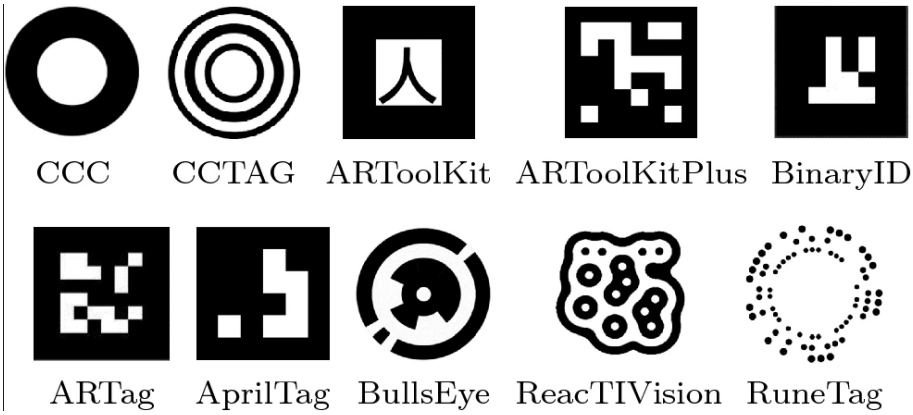


Figure 2.13: Illustration of different types of fiducial markers families, illustration from [32].

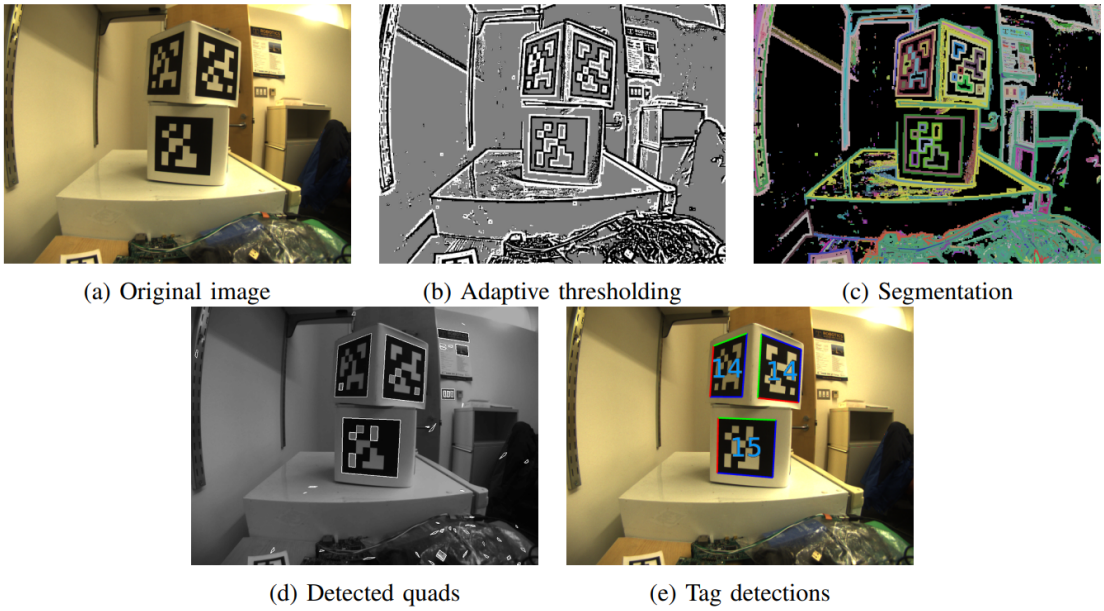
Fiducial markers have been developed for different purposes and can contain a certain amount of information, dependent on the type and size of the markers. The figure 2.13 shows a brief overview of different types of fiducial markers developed in recent years. Depending on the type of family, some fiducial marker families, such as the AprilTag3 family, Standard52h13, can have up to 48 714 unique marker IDs [33]. An AprilTag marker, as shown in figure 2.13, is a specialized fiducial marker developed by Olson [34], renowned for its detection speed, robustness, and extremely low false positive detection rates. Wang and Olsen improved the algorithm to improve computational efficiency further and to enable detection of smaller tags [35]. The AprilTag fiducial markers can, for instance, be seen in landing zones in video content from Amazon [36], Google delivery drones [37] and for Boston Dynamics warehouse robots [38], where these markers aid the robots with accurate position estimates. A fiducial marker is an marker meant to be placed within the field of view of a camera, designed for robotic applications, that is computationally effective and can provide position and orientation data to a robot. A fiducial marker is composed of an external black border and an inner region that encodes a binary pattern. The inner region is unique and identifies each marker, consisting of more or fewer bits.

2.7.3 The detection pipeline

As this thesis mainly uses AprilTag markers, the thesis gives, therefore, a brief introduction to the AprilTag detection algorithm developed from [35]. The detection algorithm of AprilTag markers attempts to detect and find all possible AprilTags

markers in a grayscale image. Firstly, the image is binarized to grayscale using an adaptive thresholding algorithm. Then all the connected black and white regions are segmented into connected components. Then any quads are fit to each cluster border pixel. Finally, poor quad fits and undecodable tags are discarded, and only a valid output is allowed and returned.

Then markers are identified by projecting the image into 2D homogeneous points from the AprilTag coordinate system, called the homography matrix, and is then estimated by the Direct Linear Transform (DLT) algorithm formulated by [28]. Then the internal code is identified by dividing the marker into a 6×6 grid or another size, dependent on the specific marker to be detected, of which the internal cells contain the ID information.



(a) Original image (b) Adaptive thresholding (c) Segmentation

(d) Detected quads (e) Tag detections

Figure 2.14: The illustration shows the different steps of the AprilTag detector. The input image (a) is binarized using an adaptive thresholding algorithm, (b) then the connected black and white regions are segmented into connected components (c). Next, component boundaries are segmented using a novel algorithm, which efficiently clusters pixels that border the same black and white region. Lastly, the quads are fit to each cluster border pixels (d), bad quad fits, and undecodable tags are discarded, and only valid detections are returned (e). Illustration from [35].

Each valid detection contains the marker *id number*, which is the marker identification number, *centre*, which is the marker center, *corners*, which is a list of pixel coordinates defined in the following order: left bottom, right bottom, right bottom, right top and left top, *Rvec* which is the 3D rotations and *Tvec* which is the 3D translations.

2.7.4 Pose estimation of AprilTag markers

Estimating the orientation, specifically, the marker pose, is being done with a method called Perspective-n-Point (PnP), which uses four points (e.g., four corners) that lays on planar surface [39]. The AprilTag algorithm in this thesis is using the cv2.solvePnp() algorithm, based on the Levenberg-Marquardt optimization[40], which inputs object points, image points, the camera matrix and the camera distortion matrix to output *Rvec* and *Tvec* of the marker. A challenge of this method is that it is only using four coplanar points, which is subject to a *ambiguity* problem, see figure 2.15.

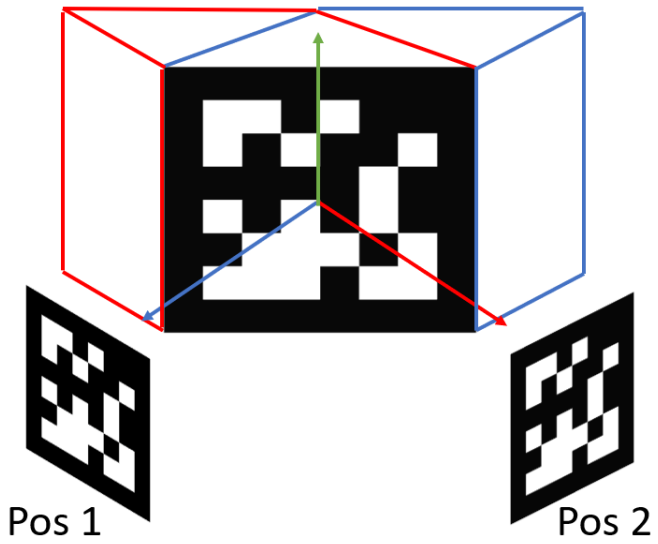


Figure 2.15: Illustration of the ambiguity problem. The same marker can be projected into two different locations.

The ambiguity problem comes from the fact that only four coplanar points (corners of the marker) are used to calculate the marker's 6-DoF orientation and pose. The output of this problem can be seen as that the estimated orientation is being flipped around one or several axes. This causes that either the marker is flipped in respect to the camera or the camera is flipped in respect to the marker. Using several markers, which combined gives a position, may solve this problem, developed by [23] [41], and [42], as a method to precisely land drones with great accuracy. This method uses several pre-defined markers with its location to each other in a bundle configuration to output precise position estimates of the drone relative to the marker. Developed for Unmanned Aerial vehicles (UAVs) to create a local reference point to enable more precise landing.

2.7.5 The link between a vision-system to global position systems

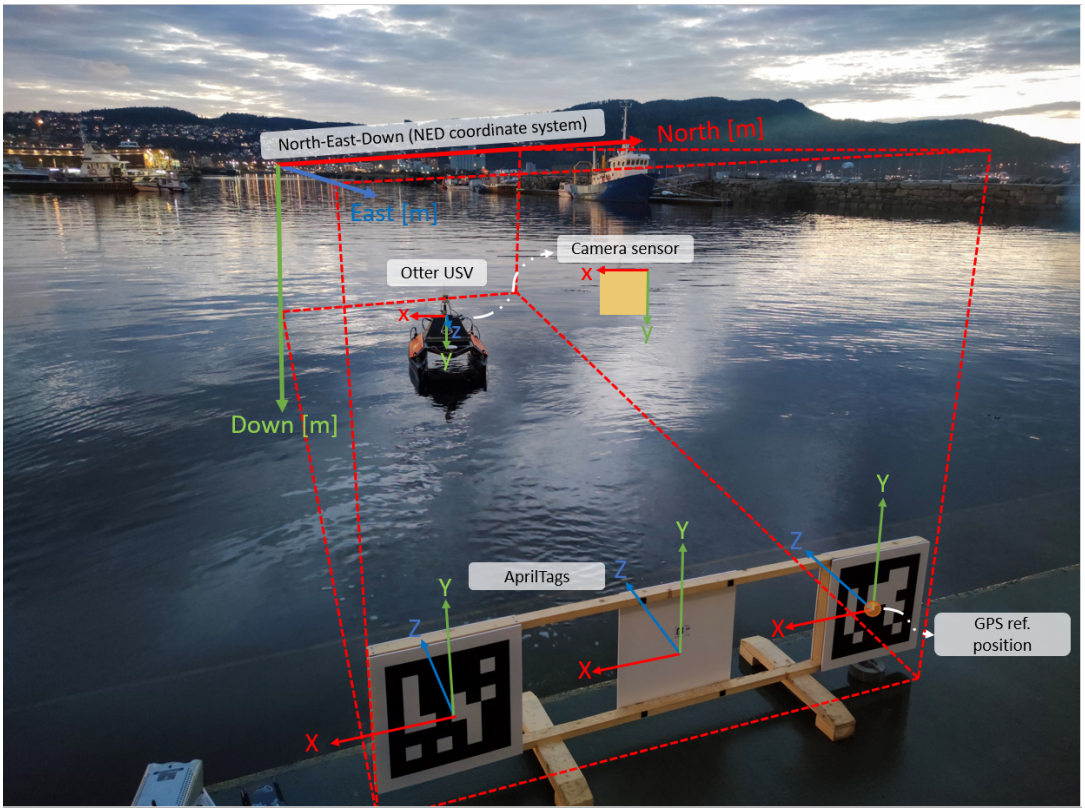


Figure 2.16: Illustration of the link between AprilTag to NED coordinates. Here, three fiducial markers are set up on the pier. Each marker creates a local reference system seen from the Otter perspective. A GPS reference point must be set in the center of the fiducial marker origin to link GNSS positions to fiducial markers.

The figure 2.16 visualizes the link between the camera system to NED coordinates. Here the GNSS origin is set in the marker origin, making it possible to compare the GNSS positions against a vision-based navigation system. The Apriltag algorithm computes a 3×3 homography matrix that project 2D points in homogeneous coordinates from the tag's coordinate system using a Direct Linear Transform (DLT) algorithm [28]. Computing the tag's position and orientation requires additional information, such as the camera focal length and physical size of the tag. The 3×3 homography matrix computed from the DLT algorithm can be written as a product of the 3×4 camera projection matrix P, which is assumed to be known. The joint rotation matrix [R—t], 4×3 is called a matrix of extrinsic parameters and is used to describe the camera motion around a static scene, or vica versa, the rigid motion of an object in front of a camera. The camera global orientation is then calculated

with the following equations from [43];

$$sm' = K[R|t]M' \tag{2.16}$$

Fully extended, equation 2.16 can be written:

$$s\begin{bmatrix} u_x \\ v_y \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}. \tag{2.17}$$

Here, s is the marker pixel coordinates on the camera sensor, K is the camera matrix as defined with equation 2.7, X, Y and Z are the locale coordinates of the marker in the camera frame, and $R|t$ is the rotation translation matrix. Finding the global camera position of the camera is found by calculating the inverse of the $R|t$ matrix,

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = [R|t^{-1}] \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}. \tag{2.18}$$

The absolute distance between camera and marker can therefore be found with the following equation,

$$|d_{distance2marker}| = \sqrt{x^2 + y^2 + z^2}. \tag{2.19}$$

The marker's pitch, yaw, and roll angles can be found by only focusing on the 3×3 rotation elements of the R—t matrix, also denoted as the rotation matrix.

$$\phi_x = \arctan^2(r_{32}, r_{33}) \tag{2.20}$$

$$\phi_y = \arctan^2(-r_{31}, \sqrt{r_{32}^2 + r_{33}^2}) \tag{2.21}$$

$$\phi_z = \arctan^2(r_{21}, r_{11}) \tag{2.22}$$

Using ϕ_x,ϕ_y and ϕ_z further one finds the yaw, pitch and roll angles represented in radians:

$$yaw = \begin{bmatrix} \cos\phi_x & -\sin\phi_x & 0 \\ \sin\phi_x & \cos\phi_x & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2.23}$$

$$pitch = \begin{bmatrix} \cos\phi_y & 0 & \sin\phi_y \\ 0 & 1 & 0 \\ -\sin\phi_y & 0 & \cos\phi_y \end{bmatrix} \tag{2.24}$$

$$roll = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi_z & -\sin\phi_z \\ 0 & \sin\phi_z & \cos\phi_z \end{bmatrix} \tag{2.25}$$

Chapter 3

Design and implementation

This section briefly describes the methodology used in this thesis, an overview of the hardware and the software architecture of the Otter USV, the development of the vision-system and the experiment setup.

3.1 Methodology: Rapid prototyping

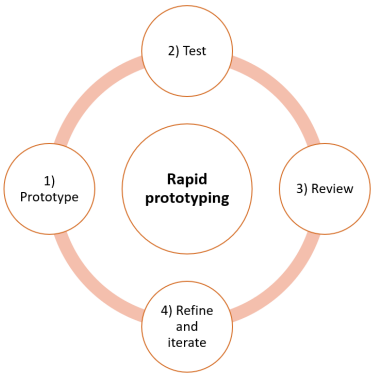


Figure 3.1: Stages of rapid prototyping.

Rapid prototyping is an agile strategy used throughout the product development process and was used in this thesis to develop the camera system. The method allows to test technologies to build an deeper understanding of certain technologies. The methodology is based on multiple iterations generated during a short period based on feedback and analysis. In addition, the process is a way to validate the hypothesis that a product will solve the problem it is intended to solve. More on this can be found in [44].

The "rapid" part comes from the speed at which the prototypes are produced, how quickly feedback is gathered and synthesized, and how fast iterations can go through the same process. For this thesis, this has meant testing ideas and algorithms to gather a deeper insight to develop an solution that can solve a particular problem.

3.2 Hardware

This section describes the primary hardware used in this project.

3.2.1 The Otter USV

The Otter is a small underactuated Unmanned Surface Vehicle (USV) currently used as a test platform in various experiments at NTNU. The vehicle is forth while set up with two Torqeedo thrusters, four Torqeedo batteries, an Ouster OS2 lidar, two GNSS-antennas, and a ZED2i camera. Figure 3.2 shows the Otter USV and the setup of the experiment carried out in this thesis.

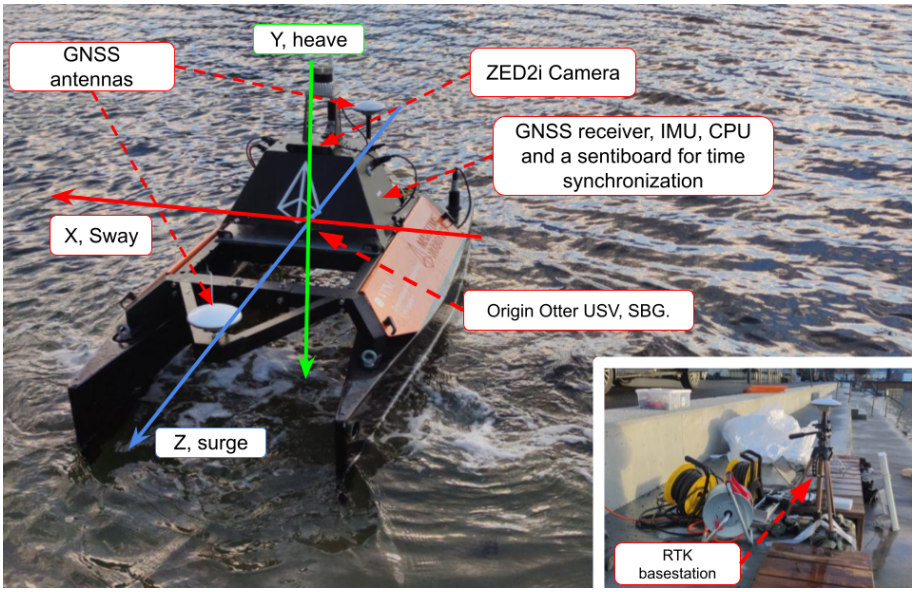


Figure 3.2: Brief overview of the Otter setup and a RTK base station on the quay.

3.2.2 Illuminated AprilTag markers

The motivation behind illuminated markers as seen in figure 3.3 was to create markers that could be utilized in absolute darkness and/or in more challenging weather conditions. The figure shows the produced markers in the TrollLab workshop. The backlight was created by some roof light panels from the workshop. A hand controller could easily change the light brightness and light temperature. The markers internal code were laser cut in fins and then painted black. Figure 3.3, d) shows the marker's code, c) is the marker without any background light, while b) is with background light on, and a) is showing the final marker illuminating in darkness. The final marker was tested indoors at TrollLabs, as seen in figure 3.4 to test if

the fiducial marker could be detected and decoded. The full size of the markers where 0.5 $m \times 0.5$ m, while the distance between the corners of the markers was 0.412 $m \times 0.412$ m. The chosen marker size made it possible to compare the camera system performance to other results achieved in the literature, e.g., [45].

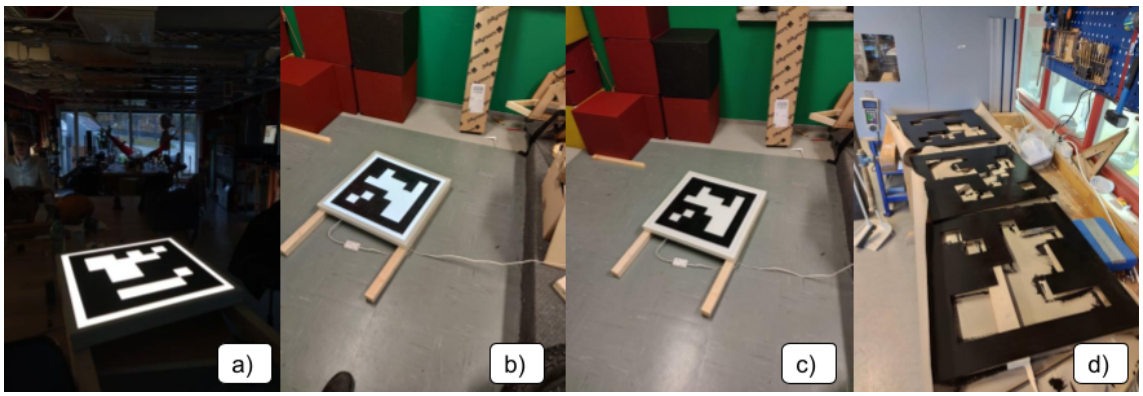


Figure 3.3: Production of the prototype illuminated markers.

Figure 3.4: In-door test of one illuminated marker with the aruco-detection algorithm at TrollLabs. Image captured with the ZED2i camera in 2202 x 1242 pixel resolution. A local reference point can be seen on the marker. Here, the camera system was not adjust to be used on the Otter USV. Therefore colors are included in the image.

3.2.3 The ZED2i camera

The ZED2i camera was chosen as it could record images in 2K (2208×1242) resolution, was IP66 certified, had polarized lenses, and had an easy-to-integrate Application Programming Interface (API), making it easy to communicate with the camera. Furthermore, the ZED2i camera also came with several built-in sensors, such as an IMU, barometer, and magnetometer. This made it easier to test and build applications as one had easy access to all its sensors through the Stereolabs ZED API. Another benefit of the camera is that the camera is widely used in different open-source communities, which means that there is information available on the internet on how to implement features or debug occurring errors. Lastly, the ZED2i camera is precisely pre-factory-calibrated, so one does not need to re-calibrate the camera to get the camera matrix. Equation 3.1 shows the zed2i camera matrix used in this thesis,

$$K = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1069.38 & 0 & 1103.6696 \\ 0 & 1069.38 & 664.69 \\ 0 & 0 & 1 \end{bmatrix}. \tag{3.1}$$

Figure 3.5: The ZED2i camera mounted on the Otter USV.

3.2.4 The Otter USV hardware set-up

The GNSS antennas mounted on the Otter were two GPS-1000 GNSS survey antennas mounted on the bow and on the stern. Both were connected to their own Ublox F9P-ZED GNSS receiver. The stern receiver was configured as a "moving base," sending RTCM (Radio Technical Commission for Maritime Services) corrections to the bow, which is set up in a "rover" configuration. This setup makes the bow receiver capable of precisely estimating the USV heading.

The Otter USV was also equipped with an ADIS16490 IMU, currently only used to log acceleration and angular rate and not fused with the GNSS system to aid the positioning system. Both GNSS receivers and the IMU is connected through a SentiBoard to ensure precise timing. However, the Otter USV is equipped with an SBG Ellipse2-D, INS (Inertial Navigation System), which is being fed with GNSS signals from the Ublox antennas. The GPS signals and the IMU positions are fused, and Kalman filtered in order to get precise position estimates. The SBG is interfaced with ROS through the SBG ROS Driver. The data from the SBG is also sent to a Unified Navigation Environment (DUNE), software, and for logging through a ROS IMC bridge. More information about the Otter setup can be found in [46].

3.3 Software

This section describes the primary software used in this project.

3.3.1 Robot Operating System (ROS)

The Robot Operating System, also called ROS, is an open-source collection of software, modules, libraries, and tools used in robotic systems [47]. The software is built to be modular in terms of nodes. Where each different node is publishing data (sensors, etc.) at different frequencies and other nodes are listening. This software was explicitly used for logging data and visualizing the data recorded on the Otter USV. All data was logged in Rosbags, which is a format that can contain data from all sensors from the Otter. However, to precisely analyze the logged data, additional software from StereoLabs, ZED API [48] that is using CUDA version 11.0 from Nvidia [49] with self-developed scripts, had to be utilized.

3.3.2 Choice of programming language

To be able to iterate quickly, the Python programming language was selected. Mainly because it is considered an easy language to understand and has excellent support, with a significant binding to languages like C in case of speed bottlenecks. In addition to this, it can be written to work with many varying platforms, such as, e.g., ROS, StereoLabs API. All code was written in Ubuntu version 18 as Ubuntu is a modern, open-source platform based on the Linux kernel and supports many different core technologies [50].

3.4 Development

This section briefly shows the development, testing, and progression of the development of the algorithms used to develop the vision-based navigation system.

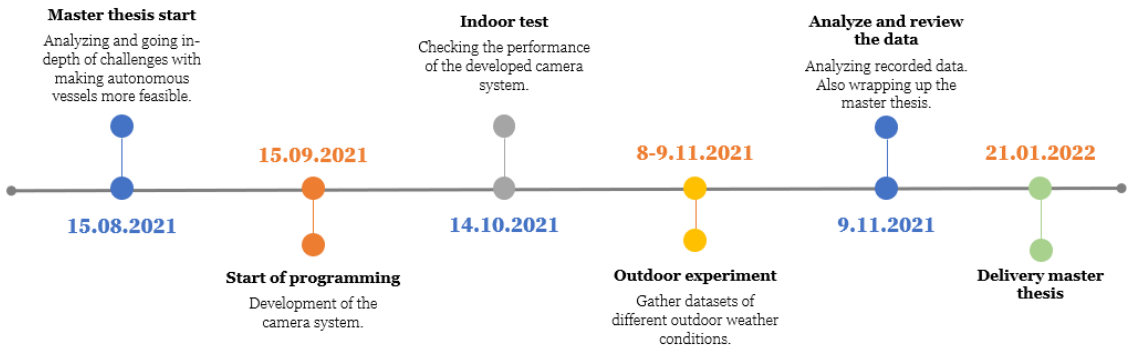


Figure 3.6: A simple timeline of the development and the work in this thesis.

3.4.1 Stage 1: Detect markers

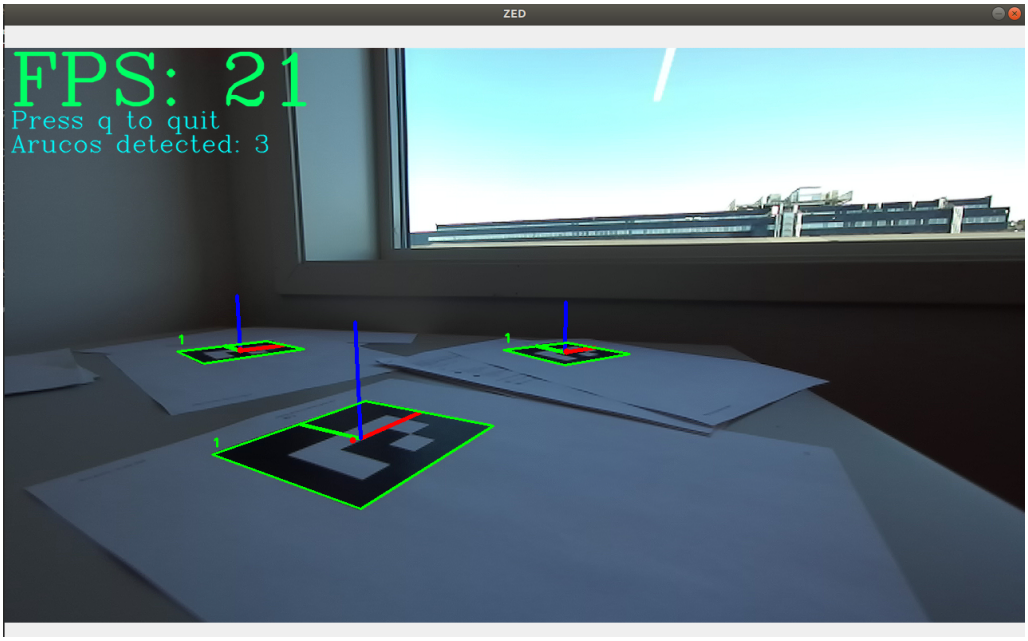


Figure 3.7: Detection of three fiducial markers with a locale coordinate system on each. Here the Aruco detection algorithm and library in use.

The first stage was to develop code to detect fiducial markers. Figure 3.7 shows three fiducial markers that are detected in a image. The orientation of each is then calculated and used to draw the marker respective reference system on each. Each

marker has a local coordinate system, which can be seen with the red, green, and blue lines pointing in x-, y- and z-direction, respectively.

3.4.2 Stage 2: Extract position and orientation information

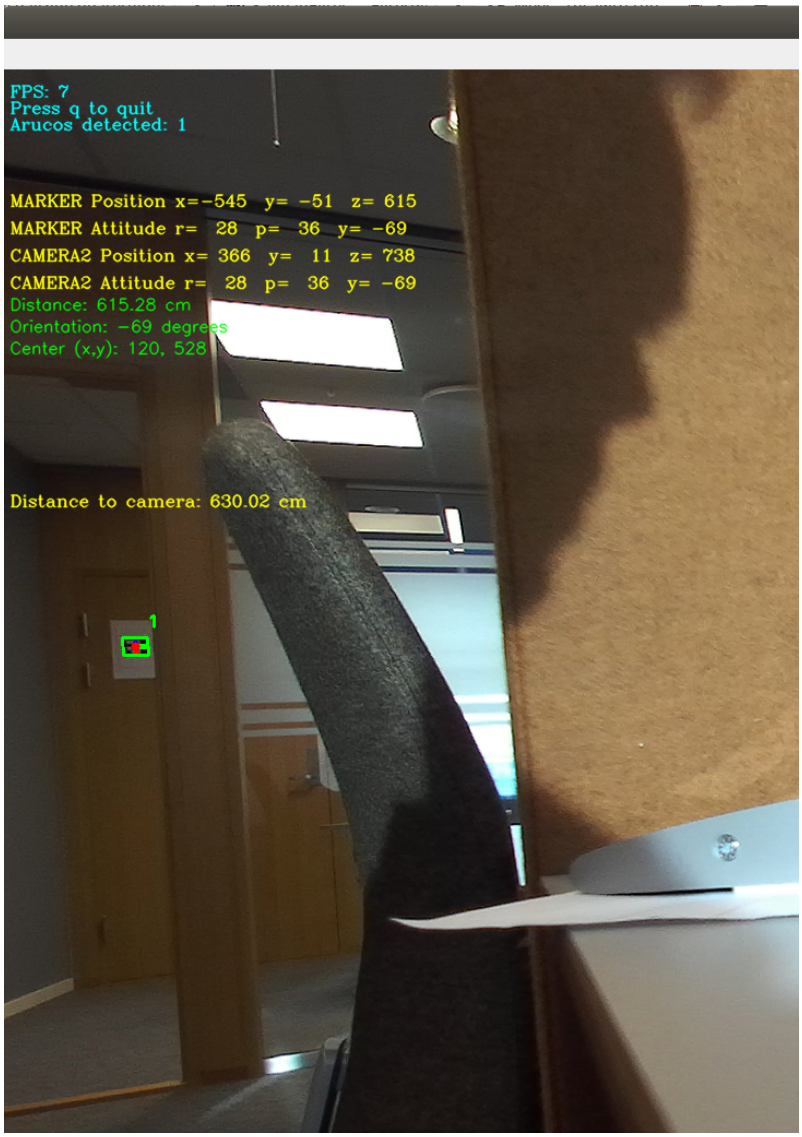


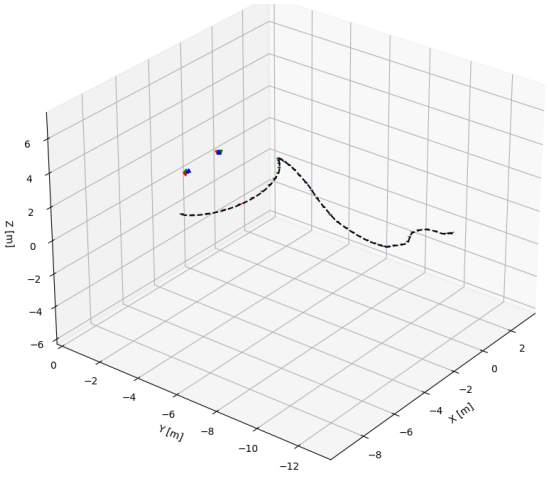
Figure 3.8: Detection of a $10cm \times 10cm$ aruco marker from a distance of 6.30 m. Calculation of the camera orientation to the marker is also displayed in the image (image screenshot by the computer screen).

The next stage was to extract the orientation estimates of the marker locale coordinate system and then calculate the camera's orientation relative to the marker. Figure 3.8 shows the orientation of the camera relative to the camera. This information was displayed on the screen as seen in the figure 3.8, here the distance to the marker and the marker position on the camera is estimated. Different equations and methods are here being tested to validate the progression.

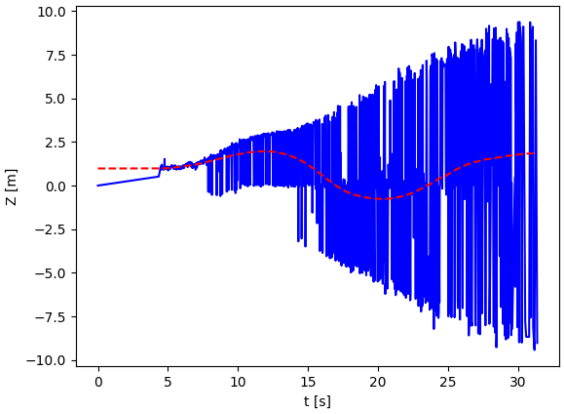
3.4.3 Stage 3: Perform indoor tests

Figure 3.9: Detection of two $0.412m \times 0.412m$ markers with the aruco detection algorithm, with 1280×720 resolution. Link to recording: https://www.youtube.com/watch?v=EGs43mpN0DE.

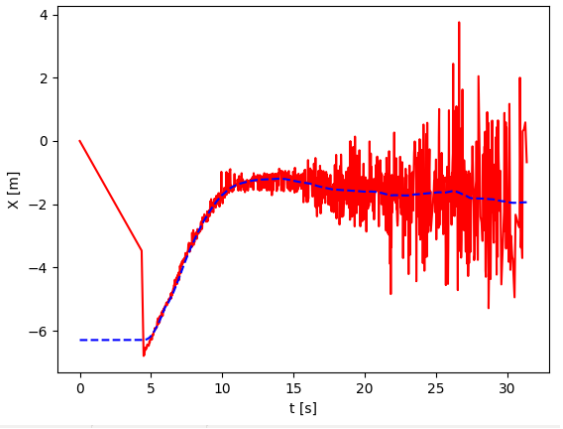
The third stage was to perform larger indoor tests to predict the system performance and how well the system could estimate the camera's position relative to the markers on longer ranges, but also to figure out methods to validate the position estimates achieved with the vision-system. These were carried out on the 14th of October 2021. The figure 3.10 shows the results of the indoor tests with plots showing the position estimates as a function of time for each axis. In this test, a Kalman filter is also included to filter out wrong position estimates and estimate a more precise camera position.



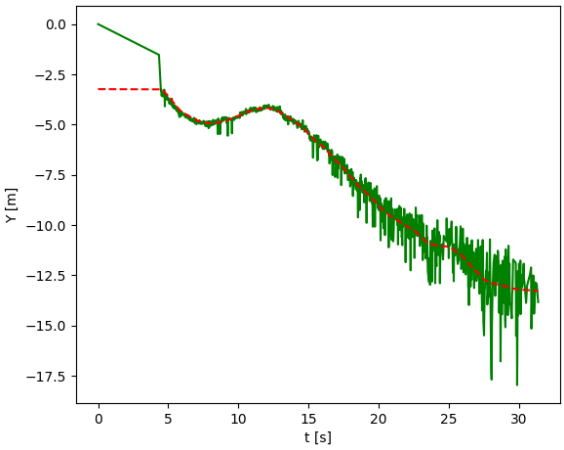
(a) 3D view of the camera path, with the two markers plotted.



(b) Position estimates along the Z-axis.



(c) Position estimates along the X-axis.



(d) Position estimates along the y-axis.

Figure 3.10: Results of the pre-testing with the aruco detection algorithm before the outdoor experiments, with a Kalman filter implemented by [51].

Figure 3.10 shows the results of the indoor testing with the camera position estimates. The camera system can estimate the position of the markers and their orientation relative to the camera. The plots were plotted with algorithms from [51], repository, which is developed to use AprilTags with the Aruco detection algorithm [52] to navigate an indoor drone. As seen in the figure, the camera system is getting many wrong position estimates along the z-axis, while this seems quite accurate around the x- and y-axis. The Kalman filter seems to smooth out the results.

Notice that in figure 3.10 b) it seems that the camera system is getting the ambiguity problem around the Z-axis. Around the x- and y-axis, this does not seem to not occur. However, the position estimates are quite inaccurate at a greater distance from the markers. The Kalman filter seems to filter out the outliers and smooth out

the position estimates.

The indoor tests gave no actual estimates on how well the system was performing outdoors in different scenarios or how accurate the system was to predict the camera positions. Therefore, it was decided to do perform some outdoor experiments to validate the positions from the vision-system. Using the USV INS system as a ground truth to benchmark the positions estimates from the camera system was considered a good method to test the system performance with outdoor conditions.

3.4.4 Stage 4: Perform outdoor experiments

The experiments conducted in this thesis were necessary because there were no relevant data-sets with challenging outdoor scenarios available. Consequently, there were no data-sets that could be utilized to benchmark the developed camera system. The experiments were also needed to raise awareness of the typical outdoor challenges a vision-based system needs to overcome if it is to be used in the future as an additional independent system or as a backup system to enable complex outdoor operations for autonomous marine vessels.

3.4.4.1 Planning of the experiments

The planning of the experiments was done in cooperation with Ø. Volden and P. Solnør who were interested in gathering data for their work. Therefore, the experiments were carried out on the 8.-9. November 2021 in front of the Maritime Robotic office in Trondheim, Norway. They helped the team with power the RTK GPS and with waterproof boots to launch and recover the Otter. The cooperation was mutually beneficial, as multiple people were necessary to ensure everything was working correctly and that all data was appropriately recorded. It must be mentioned that the marker tag detection algorithm was changed to the AprilTag algorithm from [53], to get comparable results with Ø. Volden work.

It must also be mentioned that the advisors, K. A. Christensen, S. Kohtala and M. Steinert also advised on relevant scenarios to be tested for.

3.4.4.2 Testing scenarios

Different testing scenarios were specified to analyze the vision system performance of estimating the USV positions to a dock. The testing scenarios were set to establish the boundaries of a vision-based system and achieve a deeper understanding of

potential threats a vision-based navigation system must overcome if a vision system should work along with other positioning systems, as a part or entirely independently. It must be mentioned, that if any smaller cargo boats are to be capable to dock to a floating dock without any human intervention, these vessels must be capable to localize the docks precisely. The different testing scenarios were set to:

- **Optimal conditions** - getting a benchmark of the vision-based system performance. This scenario was assumed to be in cloudy weather with even light.
- **Smoke/fog**, to simulate fog and to see of this affects the system performance.
- **Water droplet on lens**, to demonstrate how rain affects the system performance.
- **Harbor in darkness**, to test the system performance in low light.
- **Illuminated markers**, to test if background-lightened fiducial markers could be utilized to make fiducial markers more detectable in the dark by the vision-system.
- **Counter light and mirrored counter light,** to see how this affects the camera performance and the predicted position estimates.

In addition to these scenarios, tests were also carried out to get range estimates and precision estimates about the camera system compared with GNSS data. However, no data-sets for any scenarios with heavy rain, snow, or homogeneous fog were carried out, though tests for such weather conditions are necessary if a robust vision-based navigation system is to be fully developed.

Chapter 4

Results

This section presents the most important observations and results from the experiments carried out in this thesis. All position estimates between the vision-system and the onboard INS were precisely time sampled, making it possible to compare each system against each other carefully. Furthermore, because the RTK GPS positions estimates had a higher frequency was this data post-processed and time-synchronized with the estimated positions from the camera. Only estimates calculated from one single marker are used to analyze the vision-system performance. Comparing estimates from one single marker may give a better picture of the vision system performance to RTK GNSS positions.

4.1 Overview

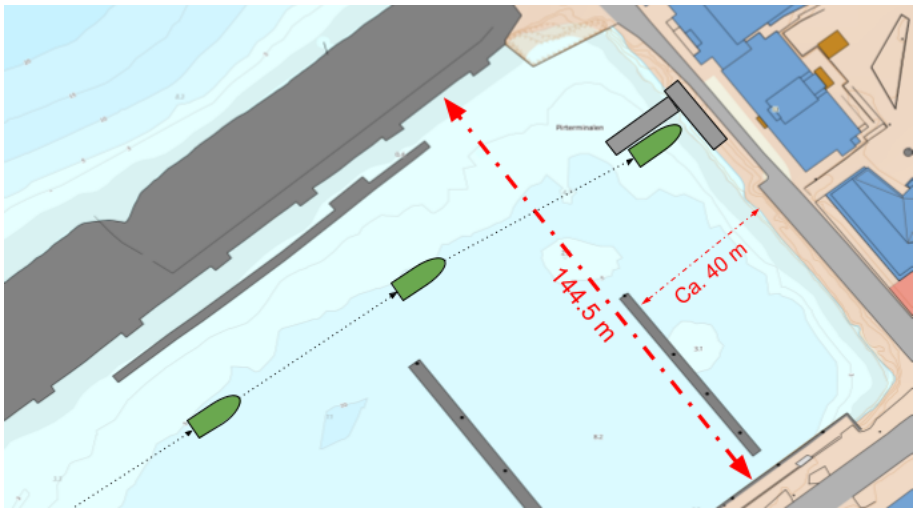


Figure 4.1: Map of the harbor where the experiments were carried out. Illustration of a USV approaching a conceptional quay.

A illustration of the Trondheim harbor with a potential cargo vessel approaching a conceptional dock can be seen in figure 4.1. The location of the conceptional dock is, in this figure, approximately the exact location where the following experiments were carried out. The objective of the experiments was to collect data sets of different weather conditions and different vessel approaches from different angles.

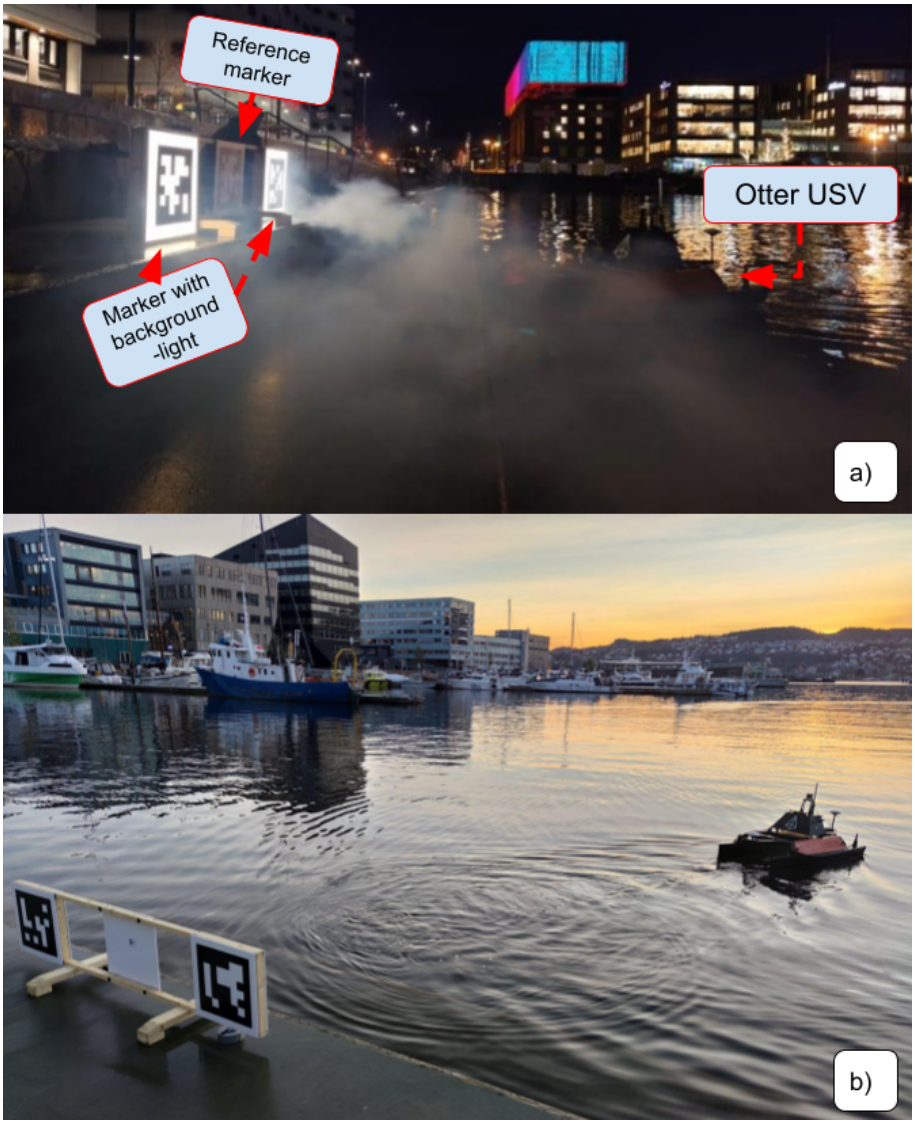


Figure 4.2: Illustrations to give a context to the experiments.

Figure 4.2 a) shows the Otter USV approaching illuminated AprilTag markers in the darkness with smoke to illustrate fog. A marker without any illuminated light was used in the middle to get a benchmark for the illuminated markers. Figure 4.2 b) shows the vessel approaching the markers in "daylight". It must be specified that this image do not illustrate the lightning during the "optimal condition" experiment. Here, the light was slightly more brighter, with

4.2 Camera observations

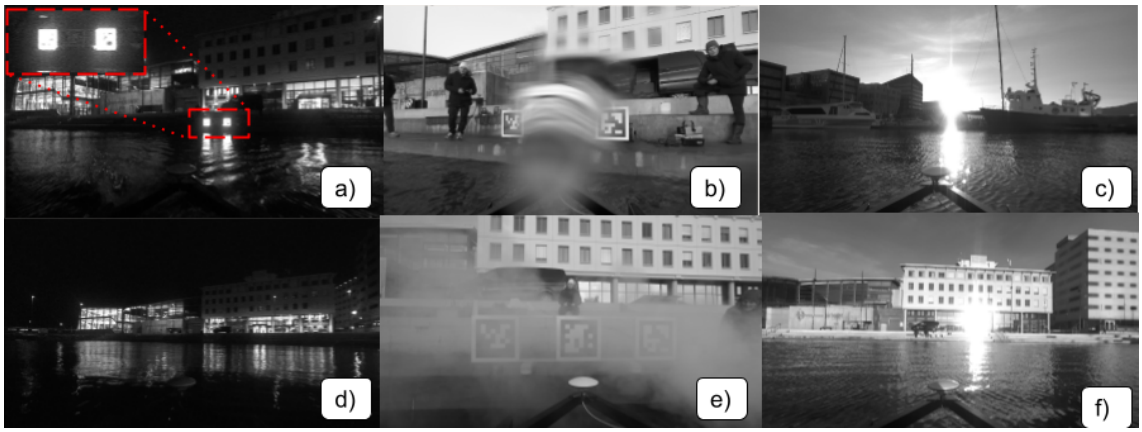


Figure 4.3: Camera observation of the different scenarios. a) the lightened up markers, b) water droplet on the camera lens, c) counter-light, d) darkness, e) fog and d) mirrored counter-light. Link to the dataset with the illuminated marker: https://www.youtube.com/watch?v=XhEYvEJEDaY&ab_channel=LarsDigerud.

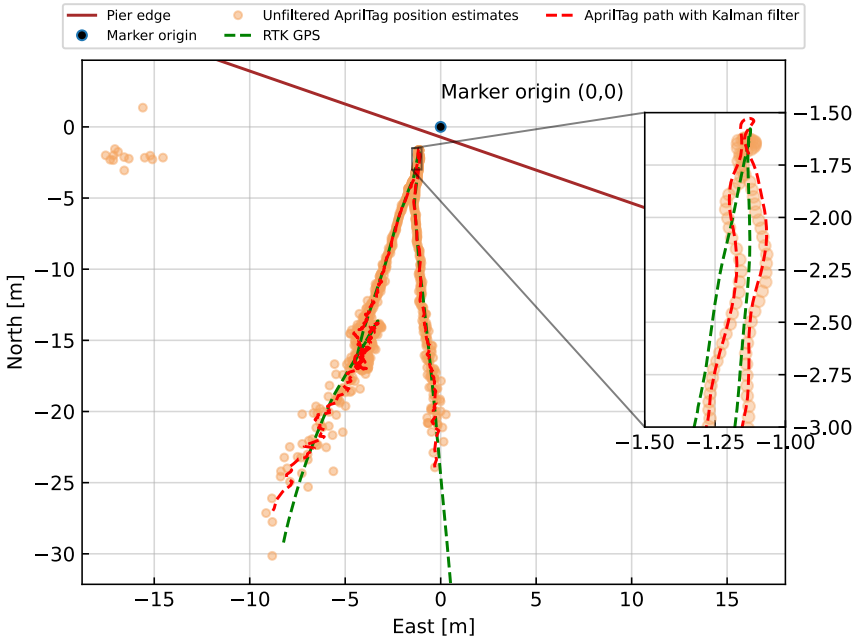
Figure 4.3 shows the challenges a robust camera system must overcome if cameras are to be utilized to navigate USVs in the docking scenario. The figure shows captured images of the worst scenarios that were captured during the experiment. All images were captured in "gray-scale" to save bandwidth, but also because the detection algorithms are using gray-scale images as an input to detect the markers.

Figure 4.3 a) shows the illuminated AprilTag markers. The IKEA light panels were on the lowest brightness setting in this scenario [1]. However, the illuminated markers become too bright, causing the vision-system not capable to detect the markers. The edges around the markers become also non-straight, making the markers impossible to be identified by the camera system. In figure 4.3 b), one can see that the third marker in the middle of the two observed markers is also almost not possible to decipher due to the water droplet on the lens. Figure 4.3 c) and 4.3 f), shows the challenges with counter-light and mirrored counter-light made by the sun. The counter-light creates larger white areas on the image, making the camera system incapable of detecting any markers. The figure 4.3 d) illustrates the challenge with a harbor in the darkness. Figure 4.3 e) shows how the images are influenced by smoke, supposed to simulate fog.

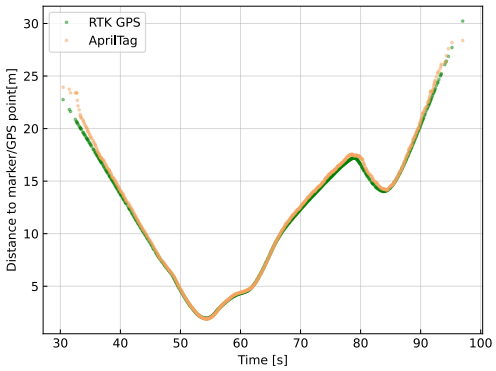
[1]Link to the dataset with the illuminated markers: https://www.youtube.com/watch?v=XhEYvEJEDaY&ab_channel=LarsDigerud

4.3 Scenario 1: Optimal conditions

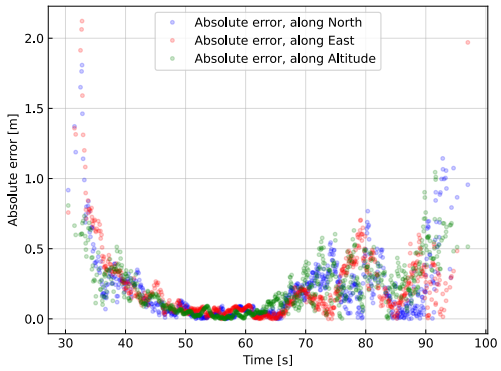
To establish a benchmark and test the system performance, the first scenario was carried out in what was considered to be the most optimal condition: Cloudy weather with even light.



(a) Experiment 1 Overview



(b) USV absolute distance to marker.



(c) Absolute error.

Figure 4.4: The results of experiment 1, link to dataset: https://www.youtube.com/watch?v=gno73QVMDfY&ab_channel=LarsDigerud. Position from only marker ID 227 is in this dataset calculated.

Figure 4.4 a), b) and c), shows the overview results of the vision-based system. Here, one can see that the vision-based system is capable of detecting the markers and estimate the vessel positions from a short to a great distance close to the RTK

GPS ground-truth. Figure 4.4 a) gives an overview of the calculated path from the camera system, unfiltered and Kalman filtered position estimates against the "ground truth" RTK GPS path. The figure shows that the camera system could detect and estimate the vessel position to the marker from a great range. Figure 4.4 b) shows the absolute distance between the USV and the marker, calculated by the vision-system and the RTK GPS positions with the equation 2.19. The figure 4.4 c) shows the magnitude of the absolute error of the measured camera positions to RTK GPS along North, East, and Down/altitude as a function of time.

The ambiguity problem seems to be present in figure 4.4 a), as outliers in the upper left corner. The ambiguity problem is more clearly described in section 2.7.4. However, it seems that the Kalman filter is capable to handle this, and filter out these outliers and still estimate the correct position of the USV. Notice, in figure 4.4 b) there are none position outliers. The magnitude of the position errors from figure 4.4 seems to be down to the centimeter range.

Summary, the results in figure 4.4 shows that the camera system is capable of estimating the USV position relative to the marker versus the USV GNSS system. The furthest distance from which the camera system was capable of estimating the USV position was from ca. 28 m range, as seen in figure 4.4 b), with an absolute error of 0.10 m along all axes as seen in figure 4.4 c).

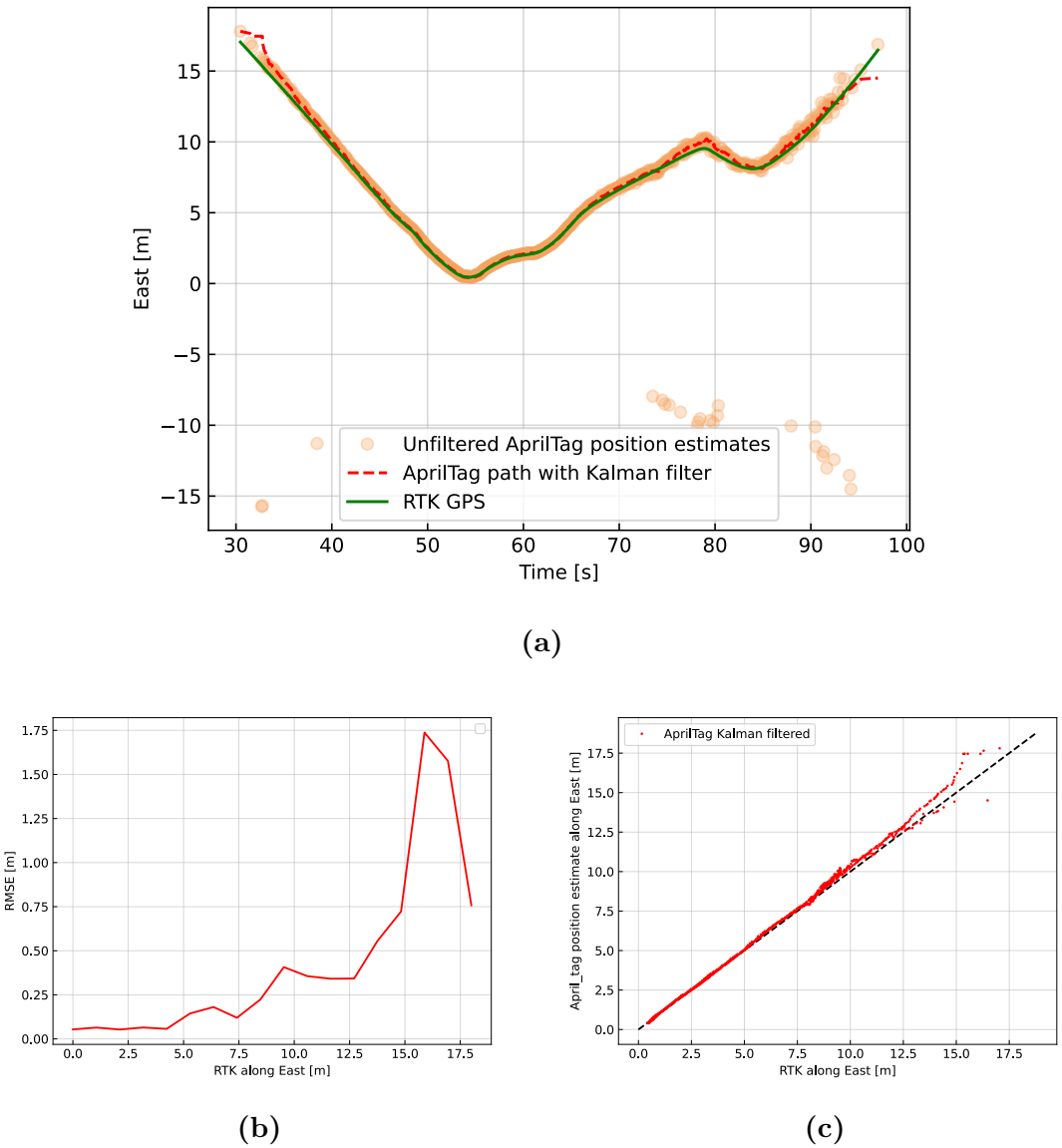


Figure 4.5: (a) Shows the AprilTag camera position estimates unfiltered and Kalman filtered to RTK GPS along the east axis as a function of time, while (b) illustrates the RMSE error and (c) illustrates the covariance.

Figure 4.5 shows the AprilTag position estimates along the East axis versus time, while figure 4.5 b) shows the Root Mean Square Error (RMSE) of the Kalman filtered camera positions estimates versus the RTK GPS path. Figure 4.5 c) shows the RTK GPS versus AprilTag position estimates, an indication of the covariance of the position estimates from both camera and RTK GPS.

Pay close attention to figure 4.5 a), and the unfiltered AprilTag position estimates below zero. These wrong position estimates are maybe a result of the orientation is being flipped around the y-axes on the local marker coordinate system or, in this plot, around the North axis, indicating the ambiguity problem as explained in

section 2.7.4. Plot b) shows that the camera system can calculate the USV position along the East axis with an RMSE accuracy lower than 10 cm within a 4,8 m range from the marker. Up to a 15 m range, the accuracy seems to be within 0.75 m. A large deviation can be seen at ca 16 m, where the RMSE error was calculated to 1.75 m. However, this deviation was much lower at 17.5 m.

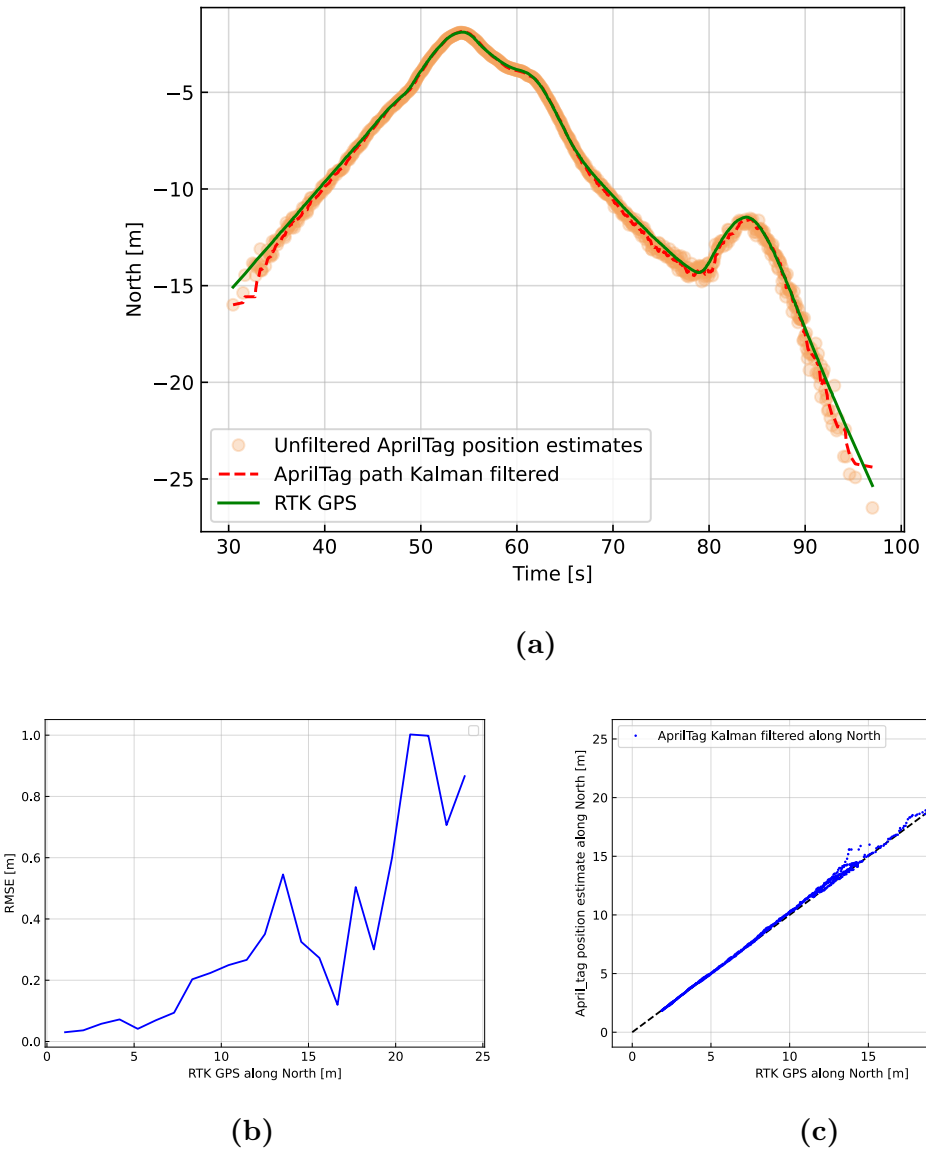
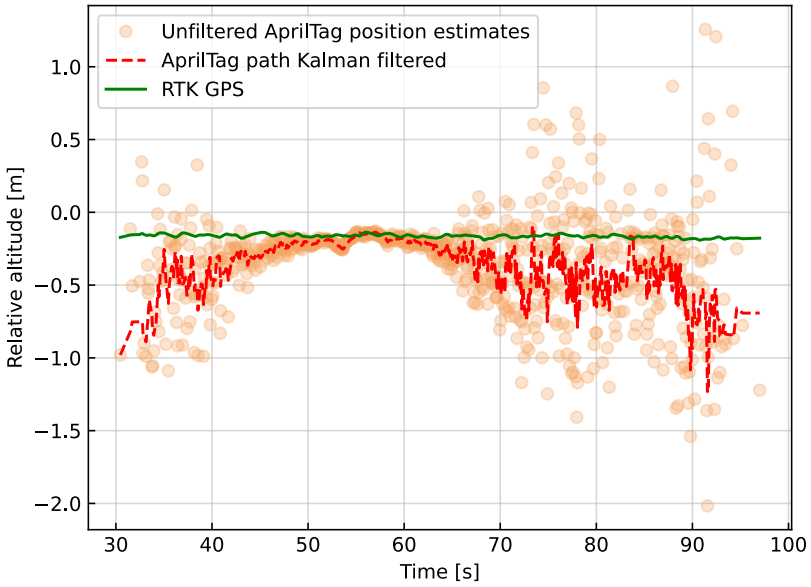


Figure 4.6: (a) Shows the AprilTag camera position estimates unfiltered and Kalman filtered to RTK GPS along North as a function of time, while (b) illustrates the RMSE error and (c) illustrates the covariance.

Figure 4.6 shows the AprilTag position estimates along the North axis versus time, while figure 4.6 b) shows the Root Mean Square Error (RMSE) versus the RTK GPS track and figure 4.6 c) shows the RTK GPS versus AprilTag position estimates, an indication of how accurate the position estimates are. Plot b) shows that the camera system can calculate the USV position along the East axis with an RMSE accuracy lower than 10 cm within a 7 m range from the marker. Up to 20 m range, the accuracy seems to be within 0.60 m, before the accuracy increases to 1 m from 20 to 25 m range.



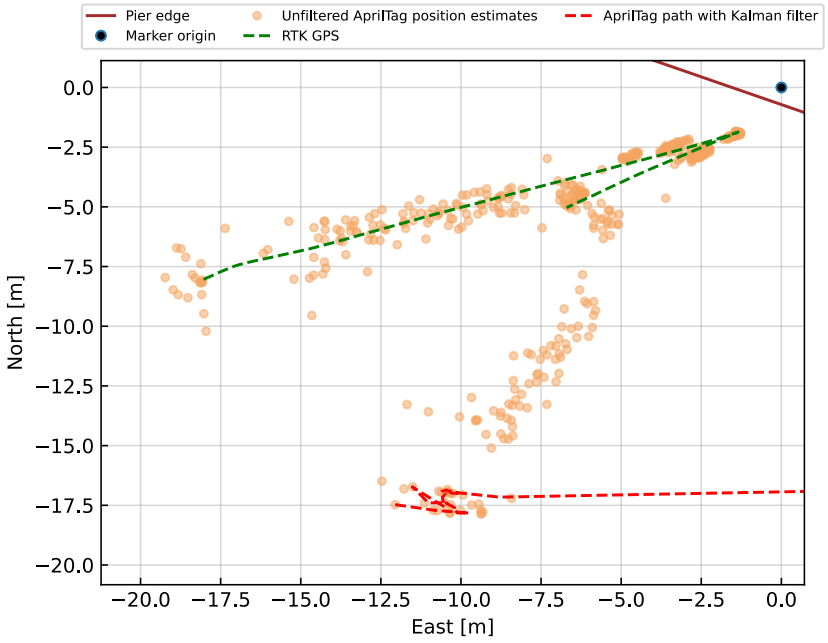
(a)

Figure 4.7: Shows the AprilTag camera position estimates unfiltered and Kalman filtered to RTK GPS along the altitude/Down axis.

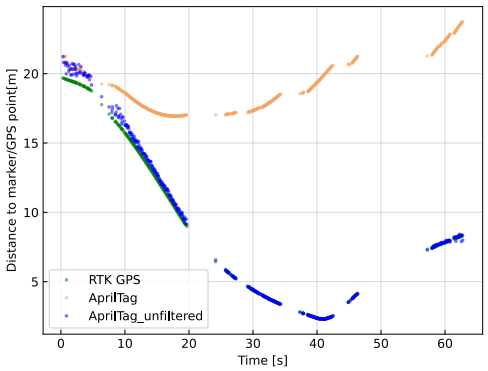
Figure 4.7 shows the AprilTag position estimates along the down/altitude axis versus time. The results along this axis were neglected as the East and North axes were assumed to be of the most importance. However, the figure 4.7 shows that the camera system was capable of estimating the altitude with relatively high accuracy, around time 50 - 60 s, where the USV was at the closest range to the marker. Still, the filtered AprilTag path estimate seems not as smooth as along the other axes, East and North axes.

4.4 Scenario 2: Fog/smoke

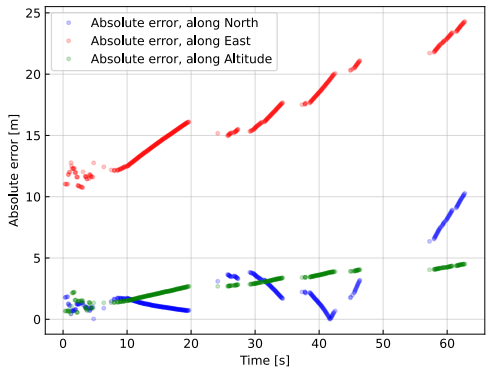
The purpose for this scenario was to identify how fog degrades the camera system's performance to detect markers and estimate the USV position.



(a) Experiment 2 overview



(b) USV distance to marker.



(c) Absolute error

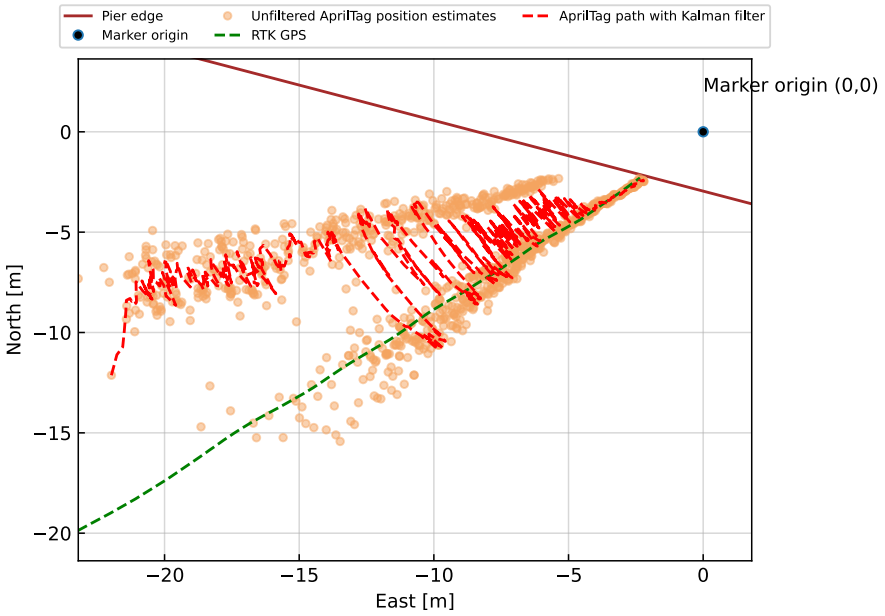
Figure 4.8: Results of experiment 2: Fog: Link to dataset: https://www.youtube.com/watch?v=IdBrYB4kcN4&ab_channel=LarsDigerud. Position from only marker ID 227 is in this dataset calculated.

Figure 4.8 a) shows all the predicted position estimates by the camera system, with the predicted Kalman filtered path against the RTK GPS path. The figure indicates that the unfiltered position estimates are largely subjected to the ambiguity problem around the y-axis on the marker. The ambiguity problem may be the cause of

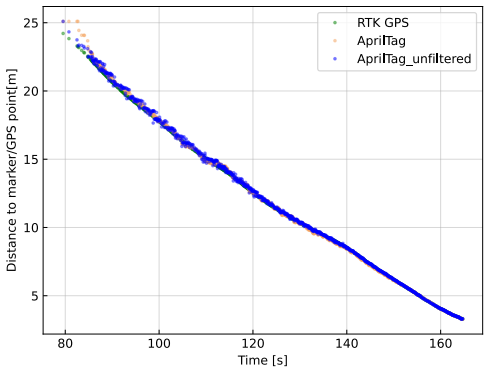
why the Kalman filter is failing in this scenario. The initial position estimates that are fed to the Kalman filter seem to be entirely wrong and deviate from the actual position estimates as seen from the RTK GPS. This causes the Kalman filter to predict and produce wrong position estimates, and once the valid position estimates are fed to the Kalman filter, these are discarded because they deviate too much from the real ones. The result is that the Kalman filter fails to estimate the USV position. However, as seen in figure 4.8 b), the absolute unfiltered distance estimates to the marker seem to be quite accurate, calculated with equation 2.19. Furthermore, the absolute error calculated with the Kalman positions against RTK GPS along the East- and the altitude-axis seems to be relatively accurate along with the Altitude/down and North axes, while the error along East is quite distinct.

4.5 Scenario 3: Mirrored counter-light

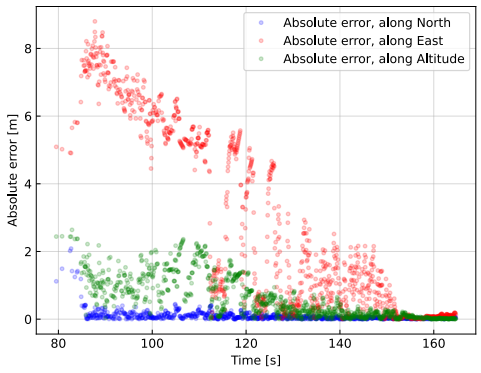
The objective of this scenario was to identify the issues counter-light may present. In order to get comparable results with the other scenarios and due to a specific GPS location; The results from this scenario uses only mirrored-light.



(a) Experiment 3 Overview



(b) USV distance to marker.



(c) Absolute error

Figure 4.9: Results of experiment 3: mirrored counter-light results. Link to dataset: https://www.youtube.com/watch?v=ikBzOHf1T-E&ab_channel=LarsDigerud. Position from only marker ID 227 is in this dataset calculated.

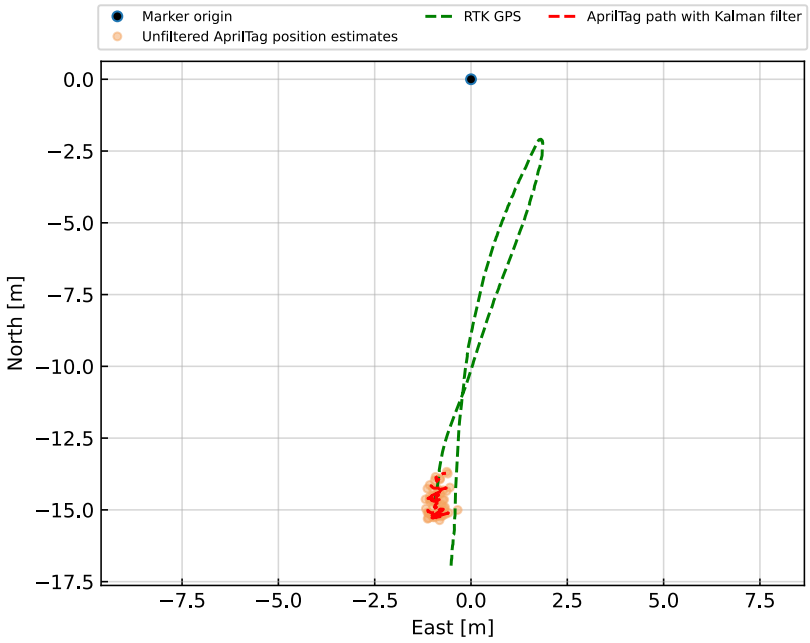
Figure 4.9 shows that mirrored counter-light made the camera system to estimate many positions outliers. Notice that many of the positions estimates seems to be subjected to the ambiguity problem. The Kalman filter is also struggling with

filtering the actual position estimates. A reason for this, as explained in the previous scenario 4.4, may be that the first position estimates fed to the Kalman filter seem to be entirely wrong, causing the Kalman filter to start with a very wrong initial positions. Compared to scenario 2, the Kalman filter seems to compensate and estimate more correct positions once the USV is on a closer range to the marker.

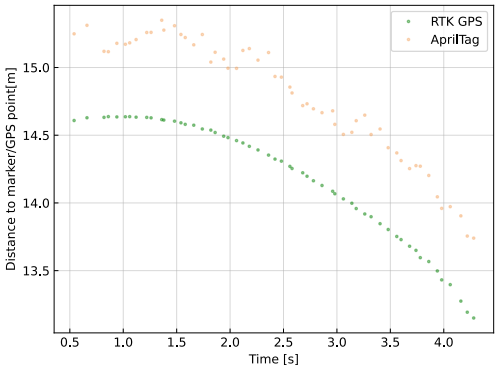
However, the USVs absolute distance to the marker seems to be quite accurate according to the RTK GPS, as seen in figure 4.9. Which indicates that the camera system can calculate the absolute distance to the marker with high accuracy, but not the position of the USV relative to the marker.

4.6 Scenario 4: Water droplet on the lens

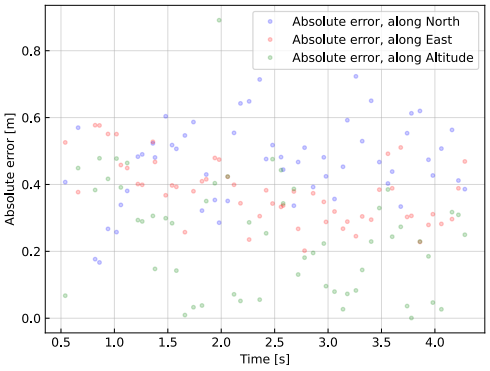
The objective of this scenario was to illustrate how a water droplet affects the captured images, and thus the camera positioning system performance.



(a) Experiment 4 Overview



(b) USV distance to marker.



(c) Absolute error

Figure 4.10: Results of experiment 4: Water drop on the camera lens. Link to dataset: https://www.youtube.com/watch?v=syAfvDmEmEE&ab_channel=LarsDigerud. Position from only marker ID 227 is in this dataset calculated.

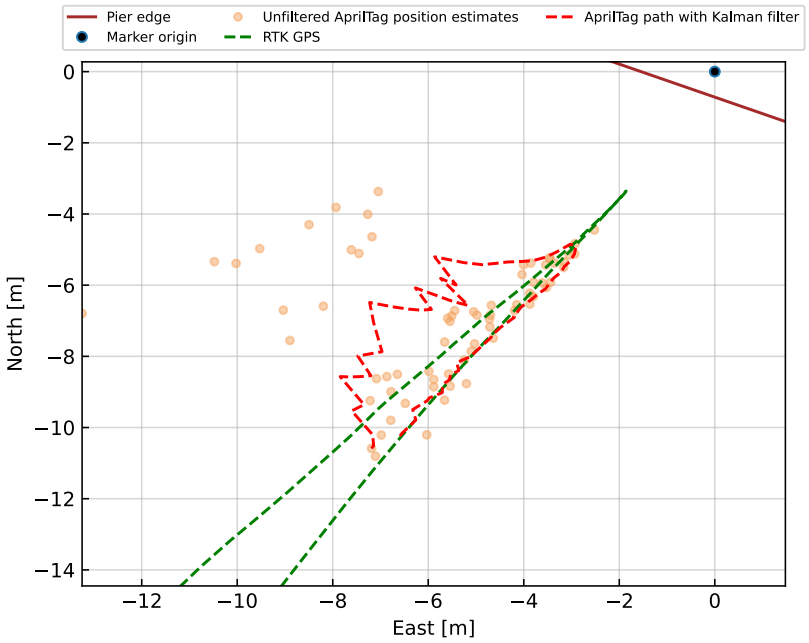
There was only a tiny water droplet on the lens during this experiment to simulate rain. Still, its position made it impossible for the camera to calculate the USVs position relative to the marker as seen in fig 4.10 a). The USVs position is only estimated at the beginning of its journey to the marker, but these position estimates

seem to be relative accurate according to figure 4.10. Figure 4.10 b) and c) only show the absolute distance and the absolute error of where the marker is detected. However, as seen in figure 4.11, the camera is capable of detecting the markers on the side of the marker that is concealed by the water droplet. As mentioned at the beginning of this section, the camera system is only calculating the orientation of the USV from one single marker, therefore positions estimates from other the detections are discarded.

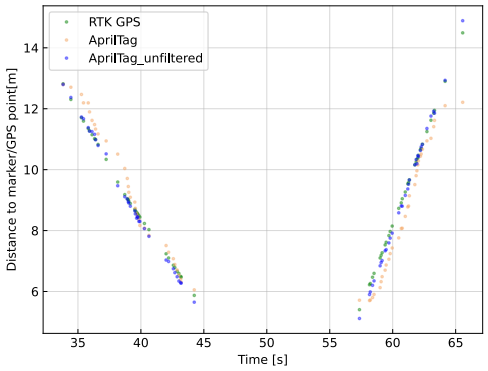
Figure 4.11: Marker detections with water droplet on the camera lens. Here marker ID 227, which is used to estimate the USV position is concealed by a water droplet.

4.7 Scenario 5: Harbor in darkness

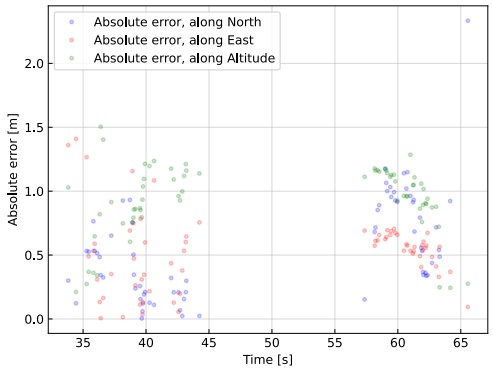
The objective of this scenario was to test how well the camera system was to detect the markers in a harbor in darkness. However, it must be specified that absolute darkness was not achieved due to light noise from the surroundings. Thus, this may set a more realistic harbor setting where such markers have to be detected.



(a) Experiment 5 Overview



(b) USV distance to marker.



(c) Absolute error

Figure 4.12: Results of experiment 5: Harbor in darkness. Link to dataset https://www.youtube.com/watch?v=7BlAfv6XNX0&ab_channel=LarsDigerud. Positions from only marker ID 227 is being calculated.

As seen in figure 4.12 a) one can see that the camera detections were relatively few. In figure 4.12 a) one can see the camera system is getting several false detections

which seems to be due to the ambiguity problem. However, in figure 4.12 the camera system is capable of calculating the distance to the marker with relatively high accuracy, while in figure 4.12 c) the absolute error of each marker is between zero to approximately 2.8 m along the north axis. The camera system was not capable of detecting any marker when the USV was at the closest range to the marker, between time 45 s and 57 s as seen in 4.12 b). However, the camera system was in this scenario capable of detecting and estimating the USV distance to the marker from a distance of ca. 12 m.

4.8 Summary of the results

Scenarios	Absolute error	Max detection range	The ambiguity problem	Observations
1) *Optimal lightening	< 10 cm	28 m	Minimal	Seems to be equally good as the positions from RTK GNSS
2) Fog/smoke	n/d	22 m	Few	Difficulties with obtaining accurate position estimates.
3) Mirrored counterlight	n/d	25 m	Significant	Difficulties with obtaining accurate position estimates.
4) **Water droplet	n/d	16 m	n/d	Difficulties with obtaining accurate position estimates.
5) Dark	n/d	16 m	Few	Difficulties with obtaining accurate position estimates.
6) Illuminated markers	n/d	n/d	n/d	Camera was not capable to detect any illuminated markers

Figure 4.13: Summary of the results. *Estimated absolute error below 10 cm within 4.8 m range. **Detection only when the marker was visible.

The table 4.13 briefly summarizes the results from the different testing scenarios. The results from the optimal condition scenario showed that the vision-based system was capable to detect the markers from a range of 28 m, with a estimated accuracy below 0.1 m within 4.8 m range. The system was also capable to detect the markers in most of the other scenarios, but not with the illuminated markers. However, the vision-based system was not able to robustly estimate any accurate position estimates in any of these scenarios. The ambiguity problem seemed also to be present in all scenarios, with varying effects.

4.8.1 Lessons learned

The outdoor experiments posed several new challenges that are absent in an indoor environment. Firstly, light and darkness may be for a vision-based navigation system be the most effective performance factors. Additionally, external factors, such as water droplets on the lens, fog, and mirrored counter light, demonstrated how quickly the vision-based system's navigation ability to estimate positions deteriorated. Next, the way the images were captured seemed to deteriorate the amount of information in each captured frame. A root of this cause may be in the ROS driver, which saves the images to the Rosbags. Finally, the indoor pre-test to check if the camera system could detect the illuminated markers indicated that the prototype illuminated markers were easily detectable indoors with environmental light. This may imply two things; either that the outdoor lighting made it more challenging for the detection algorithm to detect the marker or how the frames were captured

during the experiment. The author believes a change of settings in ROS may be a solution to this problem.

The experiments also assured the quality of the accurateness of the navigational system onboard the Otter USV. Looking at the optimal scenario, using RTK GNSS as a ground truth can be considered successful to asses the vision-based position estimates.

Chapter 5

Discussion

A novel vision-based approach for estimating a USV position relative to a dock was implemented and tested for different scenarios in this thesis. Vision-based navigation aims to give the USV capability to orient from its perspective and not rely on external services for safer and more reliable operations, e.g., autonomous cargo handling. Furthermore, estimating a USV orientation in 6 DoF to a stationary pier or a floating docking station may be a part of the solution to give USVs abilities to achieve more complex interaction with a dock, e.g., latching, charging, on/unloading cargo, in addition, to serve as an independent positioning system in a safety-critical operation.

The vision-based system used fiducial markers to estimate the USV position relative to a pier by creating a locale reference point on the dock seen from the USV perspective. The estimated position estimates were then Kalman filtered and compared against position estimates from the USVs GNSS system. Here, an RTK base station onshore was additionally used to increase the GNSS position accuracy. The method was considered a robust and reliable method to compare a vision-based navigation system performance for different outdoor scenarios in a dynamical environment, to establish the edge-cases, a potential vision-based navigation system must overcome.

Different testing scenarios were set and conducted to test the vision-based system integrity and performance for various weather conditions. All the results were calculated from one single marker using the AprilTag detection algorithm by [42], although several markers were on the pier. This approach was considered to give fewer sources of errors in the evaluation of the results. The achieved results in the optimal condition scenario, section 4.3, showed that the vision-based navigation system was capable of estimating the USV position from 28 m range from the marker. However, compared to the USV INS system, the absolute error seemed to be lower than 0.10 m within 4.8 m range. In a thermal docking phase, this might be within

the limits from which research from C. Kooij et al. 2018 [13] have set and within guidelines from Det Norske Veritas [15] as discussed in section 2.1.4.1. Furthermore, the optimal scenario confirmed Volden et al. research from 2021 [45] that fiducial markers can be utilized to give position estimates to a USV in an outdoor environment. An essential core difference between the Volden et al. approach and the presented method, is that this method focuses purely on traditionally computer vision and filtering techniques to allow the method to be more computationally lightweight and be utilized in real-time systems.

Despite the noticeable results from the optimal scenario, the other testing scenarios exposed the vision-based system for more challenging conditions. Fog/smoke gave a shorter range and fewer detections, with many position outliers. Here, the outliers caused the Kalman filter to fail. The outliers probably deviated too much from the true positions making the Kalman filter discard any valid positions. Mirrored counter-light by sun created large white regions on the images; however, an interesting result from the vision system was that many of the positions estimates also seemed to be subjected to the ambiguity problem. This caused the Kalman filter to predict entirely wrong position estimates in this scenario. However, at close range, the Kalman filter estimated more correct position estimates.

With water droplets on the lens, the vision-based system could not detect the specific marker as long as the marker was concealed, as seen in figure 4.8. Feeding the vision system with position estimates from several markers could compensate for this problem to a certain point, as seen in figure 4.11, where the surrounding markers are still detected. A harbor in the darkness gave the vision-based system less light to extract context from the images. This scenario gave significantly fewer position estimates with several outliers, which seems also to be subjected to the ambiguity problem. Absolute darkness was not reached, but light disturbance from the surroundings gave a realistic scenario for a potential operation in a dark harbor. A possible solution to increase the detection rate and range for a harbor in darkness could be possible by using illuminated markers. However, no position estimates were achieved with the illuminated markers in the outdoor experiment, despite the positive indoor pre-test in section 3.4. This may be caused by the ROS driver that receives and saves the images from the camera onboard the USV. Nevertheless, such fine-tuning is beyond this initial research and development scope.

All results from all scenarios seemed to indicate position outliers subjected to the ambiguity problem. To overcome this problem, position estimates from several markers could be used together to form a more robust output. Malyuta et al. 2019, suggest a method of pre-configuring the markers in a known pattern [23]. If one or several markers are concealed, such a method would probably keep the integrity of the

vision-based navigation system if one or several markers are not detected. Further, fusing the AprilTag position estimates with an IMU could potentially increase the position accuracy and precision. This approach is explored by Kayhani et al. 2019 [54] with positive results. However, if this solves the ambiguity problem is unknown. For practical applications, the Kalman filter can be initialized with GNSS positions to compensate for positions subjected to being ambiguous to aid the Kalman filter to overcome this issue. To avoid sudden jumps in GNSS positions, it is also essential to check that the difference in estimated position between the camera system and the INS is minimal. Therefore, utilizing vision-based positioning could alert any external operators if the primary positioning system is out of its valid range. Lastly, using cameras with a larger dynamic range and greater resolution will probably increase the system's performance.

In summary, vision-based techniques may, in certain conditions, overcome some limitations of traditional positioning methods. However, vision-based positioning has several challenges to be addressed before serving as an independent positioning system in safety-critical docking operations. Many of these challenges are believed to be solved by further hardware and software development. Nevertheless, the presented vision-based navigation method herein is believed to accommodate autonomous cargo handling.

Chapter 6

Conclusion

The thesis demonstrates that vision-based positioning using fiducial markers can be utilized to aid in the auto-docking scenario of a USV in creating a locale reference point on dock seen from the vessel perspective. In terms of detection accuracy, the results demonstrated that traditional computer vision algorithms were appropriate to estimate the USV position with a concise accuracy in optimal weather conditions. However, the vision-based system might fail due to the ambiguity problem in more adverse conditions, e.g., mirrored counter-light and a harbor in darkness. Furthermore, experiences throughout the experiments showed that several external factors, e.g., illuminated markers, a water droplet on the lens, notably influence the vision-based system performance and must be addressed before such a system can be included as an independent navigation system. Lastly, using cameras with a larger dynamic range, and greater resolution will probably increase the system's range. Using several markers together may form a more independent vision-based navigation system.

6.1 Future Work

Experiences with developing the vision-system from the ground, highlighted multiple features for future development. The suggestions are presented and discussed below:

- Use the AprilTag algorithm with bundle features developed by [23]. This may help improve the accuracy, range, and system robustness, as one or several markers could be non-visible and still get position estimates.
- Use a high-dynamic camera instead of the ZED2i camera. This will probably help improve the camera performance in situations as counter-light, wrong

exposure from the camera, rain, snow, etc.

- Use a camera with higher resolution. This will probably improve the range and accuracy.
- Explore the possibilities for using illuminated markers with very low backlight to enable the possibilities for using the marker in darkness. Using a high-dynamic camera may also improve the performance of using such markers.
- Implement deep-learning algorithms, such as the deep-tag algorithm [55], to improve the detection rate in difficult weather conditions. May be more computatively costly.
- Explore at what range such markers should be detected on and then choose a suitable marker size. For example, is it necessary to detect the marker at 30 m range for smaller USVs?
- Explore the possibilities for using markers to communicate with the vessel in events when the vessel has lost all its external communication.
- Collect more data sets from challenging weather conditions such as heavy snowing and more "homogeneous" fog.
- Explore the computational cost of having a camera-detection system used to navigate the vessel, both in terms of a backup solution and in the last phase of the docking as the primary positioning source.
- Explore the possibilities for doing more advanced operations between the dock and the vessel, as one can achieve 6-DoF position estimates (e.g., putting a robot-arm on the vessel to put cargo from the vessel onto the quay), as illustrated in the figure 2.10.
- Use Kartverkets position services to get better ground-truth precision, as one will have access to more RTK base stations, which may give a better positioning accuracy than achieved in this thesis [56]. One may also get rid of setting up a base station to get a precise global position ground truth.

As listed, several features can further be developed to improve the vision-based navigation system.

Bibliography

[1] Kongsberg. Autonomous ship project, key facts about yara birkeland. [Online]. Available: https://www.kongsberg.com/no/maritime/support/themes/autonomous-ship-project-key-facts-about-yara-birkeland/

[2] T. Ukeblad. Yara birkeland: Autonomiprosjekt på land ble for komplisert. [Online]. Available: "https://www.tu.no/artikler/yara-birkeland-autonomiprosjekt-pa-land-ble-for-komplisert/502268"

[3] Y. I. ASA. Yara birkeland. [Online]. Available: https://www.yara.com/news-and-media/press-kits/yara-birkeland-press-kit/

[4] J. L. M. S. Schönknecht, R. and H. Obenaus, *Schiffe und Schiffahrt von Morgen*, 1973.

[5] C. Andrews, *Robot Ships and Unmanned Boats*, 2016.

[6] T. I. Fossen, *Marine Craft Hydrodynamics and Motion Control*, 2021.

[7] SailDrone. What is a saildrone. [Online]. Available: https://www.saildrone.com/technology/vehicles

[8] B. Veritas, "Guidelines for autonomous shipping," 2019. [Online]. Available: https://marine-offshore.bureauveritas.com/ni641-guidelines-autonomous-shipping

[9] Lovdata, "Forskrift om forebygging av sammenstøt på sjøen (sjøveisreglene)," 2018. [Online]. Available: https://lovdata.no/dokument/SF/forskrift/1975-12-01-5

[10] ——, "Lov om sjøfarten (sjøloven)," 2020. [Online]. Available: https://lovdata.no/dokument/NL/lov/1994-06-24-39

[11] ——, "Lov om havner og farvann (havne- og farvannsloven)," 2021. [Online]. Available: https://lovdata.no/dokument/NL/lov/2019-06-21-70

[12] IMO. International convention for the safetly of life at sea (solas). [Online]. Available: https://www.imo.org/en/About/Conventions/Pages/International-Convention-for-the-Safety-of-Life-at-Sea-(SOLAS),-1974.aspx

[13] C. Kooij, A. Colling, and C. Benson, "When will autonomous ships arrive? a technological forecasting perspective," 2018.

[14] A. Smart, "Precise positioning services in the maritime sector)," 2013. [Online]. Available: http://www.ignss.org/LinkClick.aspx?fileticket=b%2F3x6KEaFS4%3D&tabid=56

[15] DNV, "Class guideline: Autonomous and remotely operated ships," 2018. [Online]. Available: https://www.dnv.com/maritime/autonomous-remotely-operated-ships/class-guideline.html

[16] Volden, "Vision-based positioning system for auto-docking of unmanned surface vehicles (usvs)," 2020.

[17] G. Bitar, "Optimization-based trajectory plannnig and automatic docking for autonomous ferries," 2021.

[18] R. Royce. Rolls-royce and finferries demonstrates world's first fully autonomous ferry. [Online]. Available: https://www.rolls-royce.com/media/press-releases/2018/03-12-2018-rr-and-finferries-demonstrate-worlds-first-fully-autonomous-ferry.aspx

[19] W. Corporation. Wärtsilä offers the world first commercially available auto-docking system. [Online]. Available: https://www.wartsila.com/media/news/03-06-2019-wartsila-offers-the-world-s-first-commercially-available-auto-docking-system-246

[20] Volvo. Assisted docking. [Online]. Available: https://www.volvopenta.com/assisteddocking/

[21] Roboat. A testbed for autonomy in city waters. [Online]. Available: https://roboat.org/technology

[22] L. A. Mateos, W. Wang, B. Gheneti, F. Duarte, C. Ratti, and D. Rus, "Autonomous latching system for robotic boats," in *2019 International Conference on Robotics and Automation (ICRA)*, 2019, pp. 7933–7939.

[23] D. Malyuta, C. Brommer, D. Hentzen, T. Stastny, R. Siegwart, and R. Brockers, "Long-duration fully autonomous operation of rotorcraft unmanned aerial systems for remote-sensing data acquisition," *Journal of Field Robotics*, p. arXiv:1908.06381, Aug. 2019. [Online]. Available: https://doi.org/10.1002/rob.21898

[24] Novatel. Real-time kinematic. [Online]. Available: https://novatel.com/an-introduction-to-gnss/chapter-5-resolving-errors/real-time-kinematic-rtk

[25] OXTS. What is rtk. [Online]. Available: https://www.oxts.com/rtk/

[26] Seedstudio. All about proximity sensors: Which type to use. [Online]. Available: https://www.seeedstudio.com/blog/2019/12/19/all-about-proximity-sensors-which-type-to-use/

[27] R. Mesnik. Detection, recognition, and identification - thermal vs. optical ip camera. [Online]. Available: https://kintronics.com/detection-recognition-and-identification-using-thermal-imaging-vs-optical-ip-camera/

[28] A. Z. R. Hartley, *Multiple View Geometry in Computer Vision*, 2011.

[29] H. B. Y. Kim, "Introduction to kalman filter and its applications," 2018. [Online]. Available: https://www.intechopen.com/chapters/63164

[30] Python. pykalman. [Online]. Available: https://pykalman.github.io/

[31] T. Szepessy. Optical control of tello drone. [Online]. Available: https://github.com/TamasSzepessy/DJITelloOpticalControl

[32] M. I. . B. M. Araar, O., "A framework for fast, robust, and occlusion resilient fiducial marker tracking." [Online]. Available: https://doi.org/10.1007/s11554-020-01010-w

[33] D. Nugent. Designing the perfect apriltag. [Online]. Available: https://optitag.io/blogs/news/designing-your-perfect-apriltag

[34] E. Olson, "Apriltag: A robust and flexible visual fiducial system," 06 2011, pp. 3400 – 3407.

[35] J. Wang and E. Olson, "Apriltag 2: Efficient and robust fiducial detection," 10 2016, pp. 4193–4198.

[36] Amazon. Amazon prime airs first customer delivery. [Online]. Available: https://www.youtube.com/watch?v=vNySOrI2Ny8&ab_channel=amazon

[37] J. Vincent. Google project wing has successfully tested its air traffic control systems for drones. [Online]. Available: https://www.theverge.com/2017/6/8/15761220/google-project-wing-drone-air-traffic-control-tests

[38] B. Dynamics. Handle robot reimagined for logistics. [Online]. Available: https://www.youtube.com/watch?v=5iV_hB08Uns&ab_channel=BostonDynamics

[39] R. Szeliski, *Computer Vision: Algorithms and Applications*, 2021.

[40] OpenCV. solvepnp(). [Online]. Available: https://docs.opencv.org/3.3.1/d9/d0c/group__calib3d.html#ga549c2075fac14829ff4a58bc931c033d

[41] C. Brommer, D. Malyuta, D. Hentzen, and R. Brockers, "Long-duration autonomy for small rotorcraft UAS including recharging," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, oct 2018, p. arXiv:1810.05683. [Online]. Available: https://doi.org/10.1109/iros.2018.8594111

[42] J. Wang and E. Olson, "AprilTag 2: Efficient and robust fiducial detection," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, oct 2016, pp. 4193–4198.

[43] J. G. et al, "A precision pose measurement technique based on multi-cooperative logo," *Journal of Physics*, Aug. 2020. [Online]. Available: https://iopscience.iop.org/article/10.1088/1742-6596/1607/1/012047/pdf

[44] K. Mcelroy, *Prototyping for Designers*, 2017.

[45] T. I. F. . A. S. O. Volden, "Vision-based positioning system for auto-docking of unmanned surface vehicles (usvs)." [Online]. Available: https://www.researchgate.net/publication/353813259_Vision-based_positioning_system_for_auto-docking_of_unmanned_surface_vehicles_USVs

[46] P. S. O. Volden. Optical control of tello drone. [Online]. Available: https://otter.itk.ntnu.no/doku.php?id=cyberotter

[47] ROS. Ros - robot operating system. [Online]. Available: https://www.ros.org/

[48] StereoLabs. Introduction. [Online]. Available: https://www.stereolabs.com/docs/

[49] Nvidia. Cuda c++ programming guide. [Online]. Available: https://docs.nvidia.com/cuda/cuda-c-programming-guide/index.html

[50] Ubuntu. Ubuntu. [Online]. Available: https://ubuntu.com/

[51] T. Szepessy. Djitelloopticalcontrol. [Online]. Available: https://github.com/TamasSzepessy/DJITelloOpticalControl/tree/master/src

[52] OpenCV. Detection of aruco markers. [Online]. Available: https://docs.opencv.org/4.x/d5/dae/tutorial_aruco_detection.html

[53] M. Zucker. apriltag. [Online]. Available: https://github.com/swatbotics/apriltag

[54] W. Z. M. N. B. M. N. Kayhani, A. Heins and A. P. Schoellig, "Improved tag-based indoor localization of uavs using extended kalman filter," 2019.

[55] Z. Zhang, Y. Hu, G. Yu, and J. Dai, "DeepTag: A General Framework for Fiducial Marker Design and Detection," 2021.

[56] Kartverket. Brukerveiledning posisjonstjenester. [Online]. Available: https://www.kartverket.no/til-lands/posisjon/brukerveiledning-posisjonstjenester

Appendix A

The detection algorithm

This appendix shows only parts of the AprilTag detection algorithm necessary to detect AprilTag markers and estimate the camera position relative to the AprilTag marker in x-,y- and z-coordinates:

```
from AprilTag.scripts import apriltag

while frame == True: # simplification
    gray = cv2.cvtColor(frame, cv2.COLOR_RGB2GRAY)
    options = apriltag.DetectorOptions(families='tag36h11',
    border=1, nthreads=4, quad_decimate=1.0, quad_blur=0.1,
    refine_edges=True, refine_decode=False, refine_pose=True,
    debug=False, quad_contours=True)

    detector = apriltag.Detector(options)
    camera_params = (1069.38, 1069.38, 1103.6696, 664.64) #iZed2
    tag_size = 0.412 # [m]
    detections, frame = detector.detect(gray, return_image=True)

    for i, detections in enumerate(detections):
        # e0: initial error, e1: final error, pose: marker position
        pose, e0, e1 = detector.detection_pose(detection,
                                               camera_params, tag_size)
        # invert marker orientation to get relative camera
                                               orientation:
        pose = np.linalg.inv(pose)

        # camera orientation relative to AprilTag marker:
        x = float(pose[0,3])
        y = float(pose[1,3])
        z = float(pose[2,3])
```

Appendix B

Implementation of the Kalman filter

Implementation of the Kalman filter defined in section 2.6 was implemented with the Python package Pykalman from [30]. The purpose of implementing the Kalman filter was to demonstrate it use, filter out any unwanted position estimates and to get a more precise position estimates from the positions received from the camera system. Besides also to get a more fundamental knowledge of how Kalman filters works. The Kalman filter was hugely motivated by the results achieved with the algorithms from [51], which demonstrated its use. However, this implementation seemed from different testing that it might not be properly implemented. The indoor test results achieved in 3.4 demonstrated the Kalman filter usefulness to recover a more precise position estimate of the camera relative to the marker. The following Kalman filter was further used to estimate the camera positions plotted in chapter 4, results.

```
from pykalman import Kalmanfilter
import numpy as np
# xp, yp and zp contains position estimates along each axis
positions = np.column_stack((xp,yp,zp))
initial_position = np.array([xp[0], 0, yp[0], 0, zp[0], 0])
dt = 0.04 # timestep
state_transition_matrix = np.array([[1, dt, 0, 0, 0, 0],
        [0, 1, 0, 0, 0, 0],
        [0, 0, 1, dt, 0, 0],
        [0, 0, 0, 1, 0, 0],
        [0, 0, 0, 0, 1, dt],
        [0, 0, 0, 0, 0, 1]])

measurement_matrix = np.array([[1, 0, 0, 0, 0, 0],
            [0, 0, 1, 0, 0, 0],
            [0, 0, 0, 0, 1, 0]])
```

```
measurement_coveriance = 11*np.eye(3) # measurement coveriance
prev_state = initial_position.copy() # previous state matrix
prev_cov = np.eye(6) # previous coveriance

kf = KalmanFilter(transition_matrices = state_transition_matrix,
        observation_matrices = measurement_matrix,
        observation_covariance = measurement_coveriance)

allstates = []
for pos in positions:
    measurment_residual = pos - measurement_matrix@prev_state
    if np.linalg.norm(measurment_residual) < 2:
        prev_state, prev_cov = kf.filter_update(prev_state,
                                                prev_cov, pos)
    else:
        prev_state, prev_cov = kf.filter_update(prev_state,
                                                prev_cov)
    allstates.append(prev_state)

positions_kalman_filtered = np.array(allstates)

print("Kalman filter done")

xp = positions_kalman_filtered[:, 0] # filtered x positions
yp = positions_kalman_filtered[:, 2] # filtered y positions
zp = positions_kalman_filtered[:, 4] # filtered z positions
```

Appendix C

GPS coordinates to NED coordinates

This appendix shows how the global coordinates were changed to NED coordinates.

```
from pymap3d import geodetic2ned
import numpy as np

# initiate position /  position defined in the marker origin
lat0  = float()
ion0 = float()
h0 = float()

# arrays containing the respective positions
lat = np.array()
ion = np.array()
h = np.array()

gps2NED = np.zeros((0,3)) # NED coordinates
for i in range(len(lat)):
    north, east, down = geodetic2ned(lat[i], ion[i],h[i],lat0,ion0,
                                     h0,deg=True)
    gps2NED = np.append(gps2NED, [[north, east,down]],axis=0)
```

Appendix D

Paper

The appended conference paper "Vision-based positioning using Fiducial Markers to aid auto-docking of Unmanned Surface Vehicles" has been written as a result of the work in this thesis. It included a method of using fiducial markers to aid unmanned surface vehicles while auto-docking. Additionally, it highlights adverse conditions such a system must overcome and supplements this thesis.

Vision-based positioning using Fiducial Markers to aid auto-docking of Unmanned Surface Vehicles

Lars Digerud*, Øystein Volden, Kim A. Christensen*, Sampsa Kohtala*, Martin Steinert***

**Department of Mechanical and Industrial Engineering, Norwegian University of Science and Technology, 7491 Trondheim, Norway (e-mail: larsdi@stud.ntnu.no, Kim@fosenregionen.no, sampsa.kohtala@ntnu.no, martin.steinert@ntnu.no)*

*** Department of Engineering Cybernetics, Norwegian University of Science and Technology, 7491 Trondheim, Norway (e-mail: oystein.volden@ntnu.no)*

Abstract: This paper describes a method of using fiducial markers to aid Unmanned Surface Vehicles (USVs) while auto-docking. Recent applications of USVs have shown increased ease and efficiency of cargo shipping operations. During auto-docking, it is essential that the USV can orient itself precisely and in real-time for long periods of time without losing signal to external services. Utilizing vision-based techniques allows USVs to orient themselves to their environment from their perspective and may represent a new method for such vessels to precisely orient themselves in the docking scenario. The work in this paper is a novel attempt to develop and evaluate vision-based strategies to localize USVs to the dock. We used Real-Time kinematic GNSS with a base station on the dock to validate the vision-based position estimates. The experiment shows that traditional computer vision techniques using fiducial markers give accurate outdoor position estimates in good conditions. We also highlight adverse conditions where the performance decreases significantly.

Keywords: Ambiguity problem, Autonomous docking, Fiducial markers, Kalman filter, Object detection, Vision-based navigation, Pose estimation

1. INTRODUCTION

The most common source of positioning data for USVs is the Global Navigation Satellite System (GNSS) fused with an internal Inertial Measurement Unit (IMU), which is integrated into an inertial navigation system (INS) to increase positioning accuracy and precision. Precise and accurate positioning is required during the docking phase. Research from Kooji, Colling, and Benson (2018) and guidelines from Det Norske Veritas (DNV) (2018) have currently set the required positioning accuracy in docking scenarios to be within 0.1 meters for autonomous vessels. Current research works at fusing INS with proximity sensors such as lidar to get a precise estimate of the distance to the pier. GNSS and Real-Time Kinematic (RTK)-GNSS systems are dependent on external services and suffer from precision degradation and signal loss in occluded urban or canyon environments (Malyuta *et al.*, 2020). Furthermore, such signals can interfere with external objects and internal devices producing noise such as Wi-Fi or the vessel being jammed by cyber-attacks. To overcome some of these limitations, we propose a method that uses a fiducial marker system to estimate the USVs orientation with a monocular camera to aid the USV during the docking scenario.

USVs can either dock at a stationary pier or a floating docking station that will move throughout the day. The latter requires a local reference point from which the USV can orient itself to get a good location estimate. This paper argues that fiducial markers in the form of AprilTags can be used as reference points for vision-based localization to enable more reliable and safer docking operations.

1.1 Related work

Several approaches for unmanned vision-based docking have been developed. Yang et al. (2013) presented a monocular visual landing method based on the estimation of the 6 Degree of Freedom (DoF) orientation of a circled H-marker to land an Unmanned Aerial Vehicle (UAV). Malyuta et al. (2020) used AprilTags to make a UAV fly, dock, and charge itself autonomously with high accuracy and precision. Volden et al. (2021) suggest a method of estimating the USV position in the docking scenario using ArUco markers with Convolutional Neural Network testing both mono- and stereo camera configurations. The work focused on outdoor performance and demonstrated that using such techniques made it possible to estimate the USV position in a range of up to 16 m, both for mono- and stereo camera configurations. Mateos et al. (2019) and Mateos (2020) used fiducial markers to orient several USVs to each other to latch the USVs together or latch a USV to a docking station.

1.2 Main contributions

This paper demonstrates how relatively low-cost cameras can aid USVs in obtaining a precise relative position estimate of a pier or a floating docking station, using a high precision RTK-GNSS for validation. The main objective is to develop an independent vision-based positioning system to increase the redundancy and accuracy of autonomous vehicles' navigation systems during the docking phase. The method focuses on only using traditional computer vision and filtering techniques to allow the method to be computationally efficient and be utilized in real-time systems. The paper also addresses some

adverse weather conditions that a fully developed camera system must overcome.

1.3 Outline

The paper is organized as follows. Section 2 describes the vision-based detection and positioning system. Section 3 describes the experimental setup and procedure. Results and discussions are presented in section 4, with the conclusion in section 5.

2. DESIGN, ALGORITHMS, AND IMPLEMENTATION

The vision-based system uses a camera to feed images to an AprilTag detection algorithm, where PnP is used to estimate the relative position of the USV. A Kalman filter is applied to improve accuracy by removing outliers. An overview of the system is shown in Figure 1.

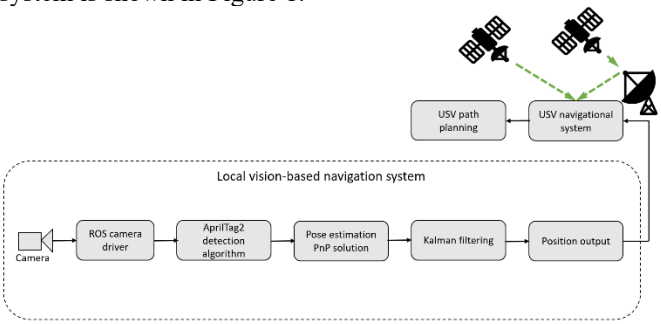


Figure 1. Block diagram describing our vision-based navigation system.

2.1 Fiducial markers

Fiducial markers are commonly used in computer vision applications for automatically detecting reference points in a physical space. The markers can be used to estimate 6 DoF relative to the camera or a robot, making fiducial markers valuable in dynamic environments where robots need exact position estimates. There are different types of markers, including circular, squared, or colored shapes. Each type consists of unique patterns which constitute the marker ID.

A fiducial marker system usually consists of a detection algorithm and a coding system. The detection algorithms are often based on traditional image processing techniques such as edge detection, blob detection, and image binarization. For instance, ARTag (Fiala, 2004), ArUco (Garrido-Jurado *et al.*, 2014), and AprilTag (Wang and Olson, 2016) use black-and-white cells in a checker-board system, whose quadrilateral boundary is detected by analyzing the lines. The markers' 6-DoF pose is then estimated using Perspective-n-Point (PnP) with four points corresponding to the markers' four corners. Under certain conditions, e.g., when the marker is viewed with low pixel resolution, the calculated pose from a single marker may be subjected to an ambiguity problem where there can be two unique solutions, i.e., the marker being projected in two different orientations, and consequently estimating two different camera locations. In general, this is not a problem as long as the marker is sufficiently close to the camera or multiple markers are used (Collins and Bartoli, 2014). The precision of such markers is usually down to a millimeter depending on the environmental lighting, camera resolution, distance to marker, and the marker size.

2.2 Fiducial Markers and Global Positions

Figure 2 visualizes the relationship between the AprilTags and the global position in our experimental setup. In order to compare the estimated camera positions to RTK GNSS positions, the North-East-Down (NED) coordinate system was used. The NED system is defined as a local tangent plane to the earth's surface, relative to a specific geographical position, represented by latitude, longitude, and altitude. Furthermore, the NED system is defined with the first dimension pointing north, the second pointing east, and the last, down, pointing towards the earth's center. Here, the RTK GNSS origin is set in the marker origin, thus making it possible to compare the RTK GNSS against the vision-based navigation system.

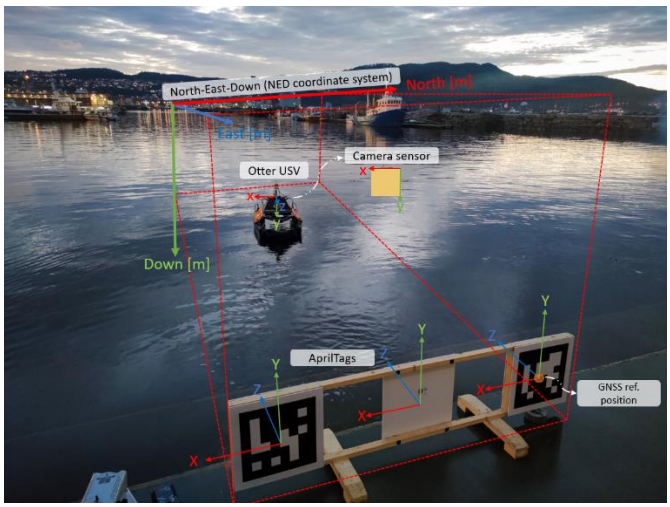


Figure 2. Experiment setup with the Otter USV and fiducial markers on the pier. The NED coordinate system is also visualized, showing the relationship between the markers and the global positioning system.

The AprilTag algorithm computes a 3×3 homography matrix that projects 2D points in homogeneous coordinates from the tag coordinate system using a Direct Linear Transform (DLT) algorithm (Hartley and Zisserman, 2004) to compute the tag position and orientation. DLT also requires the camera focal length and the physical size of the marker. The 3×3 homography matrix computed from the DLT algorithm can be written as a product of the camera projection matrix, which is assumed to be known, and the joint rotation matrix R|t, 3×4, which is called a matrix of extrinsic parameters, and is used to describe the camera motion around a static scene, or vice versa, the rigid motion of an object in front of a camera. The camera global orientation is then calculated with the following equations from Guo et al. (2020),

$$sm' = K[R|t]M' \tag{1}$$

where s is the marker pixel coordinates on the camera sensor, K is the camera matrix, M' is the local coordinates of the marker in the camera frame, and $R|t$ is the rotation translation matrix. Fully extended, (1) can be written as:

$$s\begin{bmatrix} u_x \\ v_y \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}, \quad (2)$$

Finding the global camera position of the camera is obtained by calculating the inverse of the $R|t$ matrix:

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = [R|t]^{-1} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}. \quad (3)$$

The Euclidean distance between the camera and the marker can then be obtained as,

$$|d_{abs}| = \sqrt{x^2 + y^2 + z^2}. \quad (4)$$

2.3 Kalman Filter

A Kalman filter can be utilized to provide a precise position estimate of an object (e.g., by fusing GNSS and IMU data). We use the Kalman filter to accurately determine the vehicle position relative to the pier for our vision-based system. The filter is modeled with the following equation as defined by (Kim and Bang, 2019):

$$x_k = Fx_{x-1} + Bu_{k-1} + w_{k-1} \quad (5)$$

where $\boldsymbol{F}$ is the state transition matrix applied to the previous state vector x_{k-1}, $\boldsymbol{B}$ is the control input matrix applied to the control vector u_{k-1}, and w_{k-1} is the process noise vector assumed to be zero-mean Gaussian with the covariance matrix $\boldsymbol{Q}$, i.e., $w_{k-1} \sim \eta(0, Q)$ found empirically. The process model is paired with the measurement model that describes the relationship between the state and the measurements at the current time step k as,

$$z_k = Hx_k + \nu_k \quad (6)$$

where z_k is the measurement vector, H is the measurement matrix, and ν_k is the measurement noise vector assumed to be zero-mean Gaussian with the covariance matrix R, i.e., $\nu_k \sim \eta(0, R)$.

The purpose of the Kalman filter is to provide a state x_k at time step k given an initial estimate of x_0, the series of k measurements, z_1, z_2, ..., z_k, and the system's information described by F, B, H, Q, and R.

The Kalman filter algorithm can be expressed by prediction and update stages. The hat operator, ˆ, means an estimate of a variable, and the superscripts $-$ and $+$ denote predicted (prior) and the updated (posterior) estimates, accordingly. The prediction state estimate is defined as,

$$\widehat{x_k^-} = F\widehat{x_{k-1}^+} + Bu_{k-1} \quad (7)$$

and its predicted error covariance as,

$$P_k^- = FP_{k-1}^+F^T + Q \quad (8)$$

The update stages are then defined as,

$$\widetilde{y_k} = z_k - H\widehat{x_k^-} \quad (9)$$

$$K_k = P_k^-H^T(R + HP_k^-H^T)^{-1} \quad (10)$$

$$\widetilde{x_k^+} = \widetilde{x_k^-} + K_{k\tilde{y}} \quad (11)$$

$$P_k^+ = (I - K_kH)P_k^- \quad (12)$$

where 9, 10, 11 and 12 are defined as the measurement residual, Kalman gain, updated state estimate and the updated error covariance, respectively. The predicted state estimate is evolved from the updated previous state estimate. In the update stage, the measurement residual $\widetilde{y_k}$ is computed first. The measurement residual is the difference between the true measurement, z_k, and the estimated measurement $H\widehat{x_k^-}$. The filter estimates the current measurement by multiplying the predicted state by the measurement matrix. $\widetilde{y_k}$ is later then multiplied by the Kalman gain, K_k, to provide the correction $K_{k\widetilde{y_k}}$ to the predicted estimate $\hat{x}_k^-$. Once the updated state estimate has been calculated, the error covariance matrix, p_k^+, is calculated to be used in the next time step.

To properly function, the Kalman filter needs an initial value, $\hat{x}_0^+$, and an initial guess of the error covariance matrix, P_0^+. Importantly, the Kalman filter are derived on the assumption that the process and measurement models are linear and can be expressed with the matrices F, B and H, whereas the process and measurement noise are additive Gaussian.

3. EXPERIMENTAL SETUP AND TESTING

Our system consists of an USV called Otter equipped with GNSS receivers and an IMU integrated into an INS. The Otter is developed by Maritime Robotics in collaboration with the Norwegian University of Science and Technology (NTNU) as an experimental test platform to conduct sea trials. We used a camera onboard the USV and an onshore base station for processing positioning data, and a wireless radio for communication. The USV and camera specifications can be seen in Table 1 and Table 2 respectively. All actions of the USV during the experiment were remote-controlled, and all data were recorded with the Robotic Operating System (ROS) for post-processing. The AprilTag markers' physical size were $0.412\ m \times 0.412\ m$ located on the pier, as seen in Figure 2.

3.1 Environmental effects on AprilTag detection

Six different scenarios were analyzed to test how the camera system performed in both good and adverse weather conditions. The scenarios can be seen in Figure 4 showing the following conditions: a) markers at night with backlight illumination (no detection), b) water droplets on the lens (no detection), c) optimal light conditions, d) the harbor in darkness, e) fog, and f) mirrored counter light from the sun.

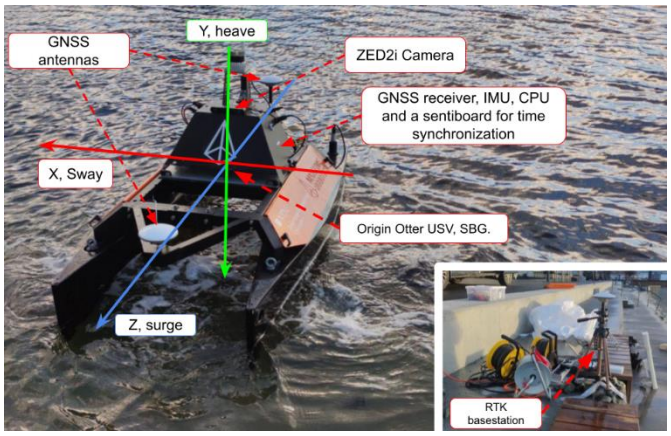


Figure 3. The fully equipped Otter USV and the onshore RTK GNSS base station.

3.2 Accuracy

Three scenarios were further analyzed to measure the accuracy of our vision-based positioning system. Scenario 1: Optimal conditions, scenario 2: Mirrored counter light and scenario 3: Harbor in darkness. All position estimates were calculated based on information from one single marker.

Table 1. Otter USV specifications

Dimensions	2 m by 1.08 m symmetric footprint
Position and heading reference system	Two GNSS receivers in bow and stern
IMU	ADIS 16490
INS	SBG Ellipse2-D

Table 2. Zed2i camera specifications

Model name	2 m by 1.08 m symmetric footprint
Pixel format	RGB
Resolution	2208 × 1242
Sample rate	15 Hz

4. RESULTS AND DISCUSSION

This chapter presents and discusses the positional accuracy of the vision-based system for each of the three scenarios.

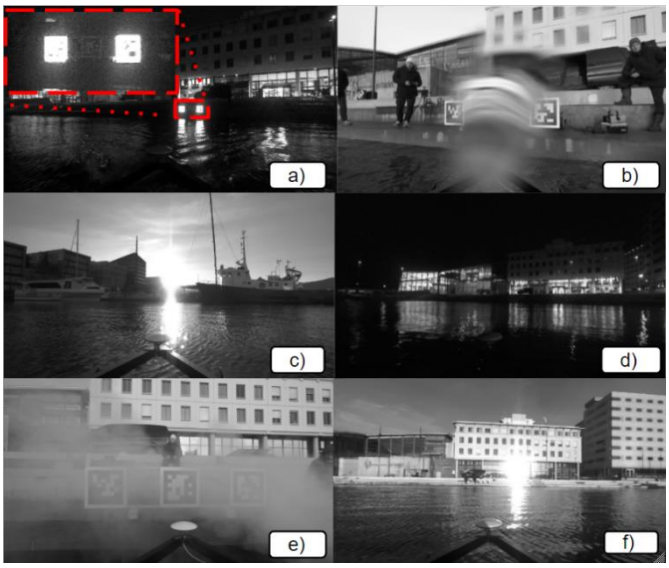


Figure 4. ZED2i camera observations as seen from the Otter USV, a) with illuminated markers, b) water droplet on the lens, c) optimal conditions, d) harbor in darkness, e) smoke to simulate fog, and f) mirrored counter light.

4.1 Scenario 1: Optimal conditions

Figure 5 shows the top-down view of the Otter approaching the marker in NED coordinates. The yellow points show the Otters' position estimates by the AprilTag system, while the Kalman filtered path can be seen as the red dotted line and the RTK GNSS path as the green dotted line.

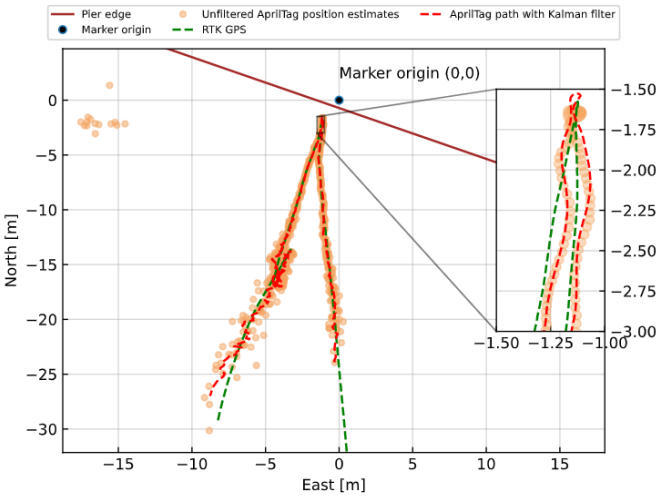


Figure 5. USV RTK GNSS path with raw and Kalman filtered positions from the camera system.

In the upper left corner of Figure 5, we can see a few erroneous position estimates, which are likely caused by the ambiguity problem (marker orientation predicted incorrectly). Moreover, the results show that the Kalman filter can remove these outliers and estimate an accurate USV position.

Figure 6 shows the absolute distance to the marker, calculated from the camera and the RTK GNSS positions with equation 4. Notice that there are no outliers for the unfiltered position

estimates in the figure, which indicates that the ambiguity problem does not affect absolute distance.

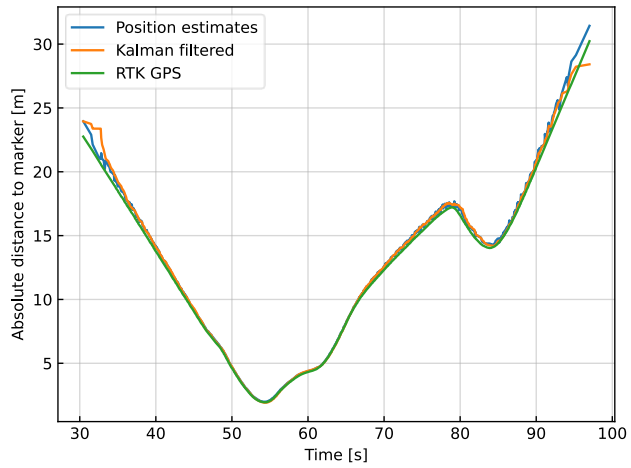


Figure 6. USV absolute distance to marker during its mission.

Figure 7 shows the AprilTag position estimates along the three axes: East, North, and Down versus time. Some position estimates seem to be flipped symmetrically in the East axis, which explains why the absolute distance remains accurate compared to the RTK GNSS. This problem does not appear along the North axis. However, the Down axis has multiple positional outliers, making it less precise than the Eastern and Northern axes.

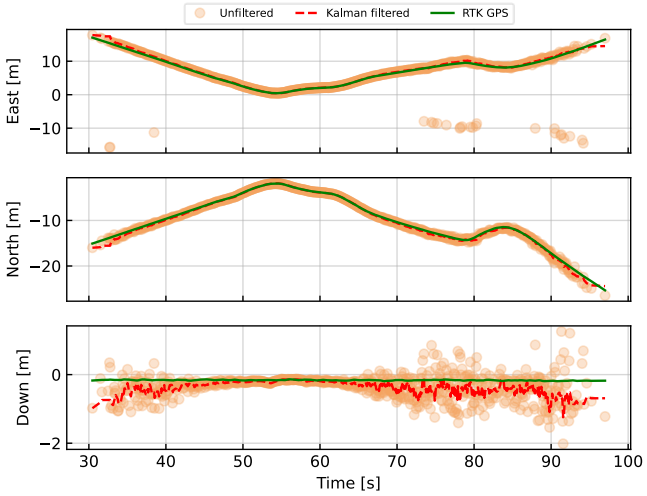


Figure 7. Positions along East, North, and Down as a function of time.

Figure 8 shows the absolute error of the raw position estimates to RTK GNSS along North, East, and Down as a function of time. The absolute error is below the 0.1-meter requirement set by DNV between 50 s to 63 s when the Otter is within a range of 5 meters to the marker.

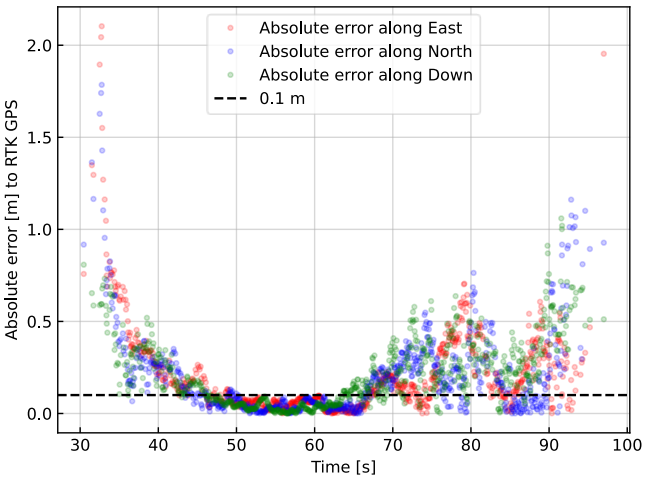


Figure 8. The absolute error of the raw position estimates.

Figure 9 shows the RMSE of the estimated camera positions by the AprilTag system compared to the RTK GNSS. The unfiltered and the Kalman filtered positions seem to have a high positive covariance, indicating that the Kalman filter may not significantly increase the absolute positioning accuracy. However, it does not reflect the ambiguity problem, which causes the coordinates along the East axis to deviate as shown in Figure 7.

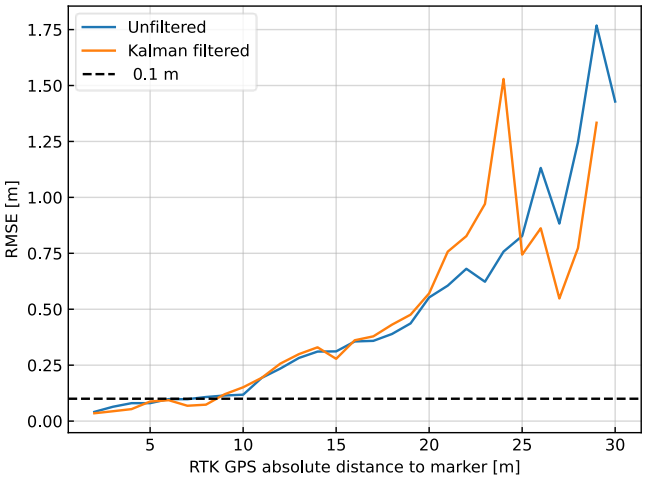


Figure 9. RMSE for unfiltered and Kalman filtered position estimates.

4.2 Scenario 2: Mirrored counter light

Figure 10 shows the results of testing with a mirrored counter light. The figure shows that the camera system estimates contain many outliers compared to the true RTK GNSS estimates. Additionally, the Kalman filter started with wrong initial conditions, which caused incorrect predictions. However, the Kalman filter accurately predicts the USV position when the distance to the marker is low.

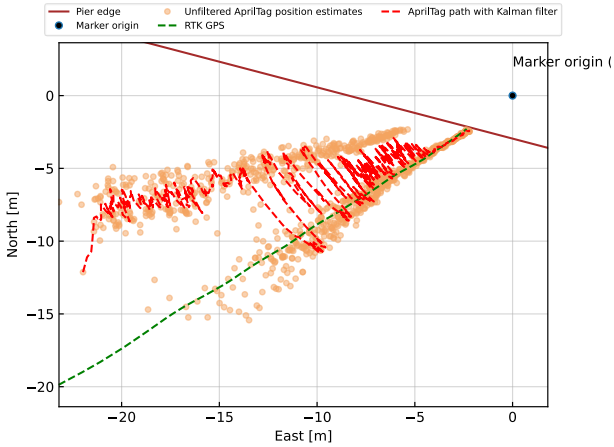


Figure 10. USV RTK GNSS path with estimated positions during the mirrored counter light scenario.

4.3 Scenario 3: Harbor in darkness

Figure 11 shows position estimates when the USV approached the pier at night. As seen in the figure, the camera system is getting significantly fewer position estimates compared to scenarios 1 and 2. In addition, it seems that many of the estimates are close to the RTK GNSS positions, but some are not. It is difficult to tell whether these position estimates are subjected to the ambiguity problem, but they cause the Kalman filter to estimate wrong positions.

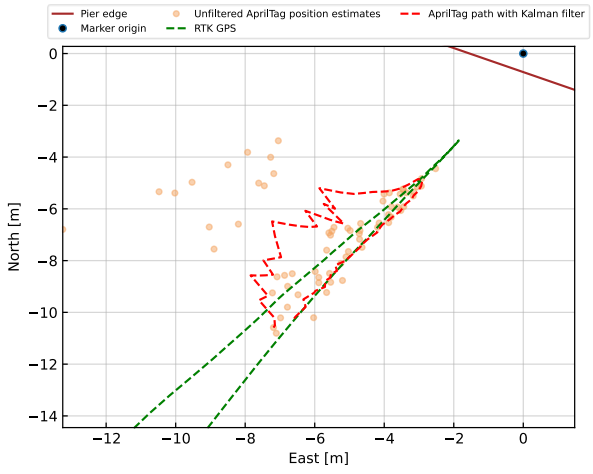


Figure 11. USV RTK GNSS path with estimated positions during the harbor in darkness scenario.

4.4 Discussion

All the results were calculated from one single marker using the AprilTag detection algorithm by Wang and Olson (2016). The camera system in scenario one successfully created a local reference point on the pier to estimate the USV position. The method can accurately estimate the USV positioning at a range of up to 5 meters, thus, demonstrating that the camera system can provide sufficient positioning accuracy as an independent positioning system. The method is robust in optimal weather conditions, despite a few outliers. However, system performance degraded significantly in more adverse environmental conditions (e.g., mirrored counter light and a harbor in darkness). This may be due to the ambiguity problem, which the Kalman filter also struggled to compensate.

Pre-configuring multiple markers in a known pattern may improve positioning accuracy and ensure the vision-based system integrity. Malyuta et al. (2020) suggest using several AprilTags in a bundle configuration to estimate a UAV position robustly. However, if all markers or only a single marker is detected, the estimated positions may be subjected to the ambiguity problem. Fusing the AprilTag position estimates with the IMU may improve the position estimates further (Kayhani *et al.*, 2019). For practical applications, the Kalman filter can also be initialized with GNSS positions to improve the vision-based system before the final docking phase. To avoid sudden jumps in position, it is also important to check that the difference in estimated position between the camera system and the INS is minimal.

In summary, vision-based techniques may overcome some limitations of traditional positioning methods in certain conditions, especially in the event of lost communication to external services or interference. Furthermore, vision-based positioning could also alert any external operators if the primary positioning system is out of its valid range. However, vision-based positioning has several challenges that must be addressed before serving as an independent positioning system in safety-critical docking operations.

5. CONCLUSION

This paper has demonstrated how vision-based positioning can be utilized to aid in the auto-docking scenario of a USV by creating a locale reference point between the USV and a docking station. In terms of detection accuracy, the results have shown that traditional computer vision algorithms can estimate the USV position with reasonable accuracy in optimal weather conditions. However, the vision-based system is limited during adverse conditions, including mirrored counter-light and darkness. These scenarios must be addressed before a vision-based system can be included as an independent navigation system. Lastly, we believe using cameras with a higher dynamic range, and greater resolution will increase performance. Using several markers may also improve the robustness before an independent vision-based navigation system can be deployed.

6. ACKNOWLEDGEMENTS

We are grateful to the NTNU Center of Autonomous Marine Operations and Systems (AMOS) at the Norwegian University of Science and Technology, who let us use one of their USVs to conduct experimental tests. They have also been accommodating with support and help when preparing the core functionality for the field experiments onboard the Otter USV.

REFERENCES

Collins, T. and Bartoli, A. (2014) ‘Infinitesimal Plane-Based Pose Estimation’, *International Journal of*

Computer Vision, 109(3), pp. 252–286. doi:10.1007/s11263-014-0725-5.

DNV (2018) 'Class guideline: Autonomous and remotely operated ships'. DNV. Available at: https://www.dnv.com/maritime/autonomous-remotely-operated-ships/class-guideline.html.

Fiala, M. (2004) *ARTag, An Improved Marker System Based on ARToolkit*. National Research Council Canada. doi:10.4224/5763247.

Garrido-Jurado, S. *et al.* (2014) 'Automatic generation and detection of highly reliable fiducial markers under occlusion', *Pattern Recognition*, 47(6), pp. 2280–2292. doi:10.1016/j.patcog.2014.01.005.

Guo, J., Wu, P. and Wang, W. (2020) 'A precision pose measurement technique based on multi-cooperative logo', *Journal of Physics: Conference Series*, 1607(1), p. 012047. doi:10.1088/1742-6596/1607/1/012047.

Hartley, R. and Zisserman, A. (2004) *Multiple View Geometry in Computer Vision*. 2nd edn. Cambridge University Press. doi:10.1017/CBO9780511811685.

Kayhani, N. *et al.* (2019) 'Improved Tag-based Indoor Localization of UAVs Using Extended Kalman Filter', in. *36th International Symposium on Automation and Robotics in Construction*, Banff, AB, Canada. doi:10.22260/ISARC2019/0083.

Kim, Y. and Bang, H. (2019) 'Introduction to Kalman Filter and Its Applications', in Govaers, F. (ed.) *Introduction and Implementations of the Kalman Filter*. IntechOpen. doi:10.5772/intechopen.80600.

Kooij, C., Colling, A.P. and Benson, C.L. (2018) 'When will autonomous ships arrive? A technological forecasting perspective', in. *14th International Naval Engineering Conference and Exhibition*, Glasgow, UK. doi:10.24868/issn.2515-818X.2018.016.

Malyuta, D. *et al.* (2020) 'Long-duration fully autonomous operation of rotorcraft unmanned aerial systems for remote-sensing data acquisition', *Journal of Field Robotics*, 37(1), pp. 137–157. doi:10.1002/rob.21898.

Mateos, L.A. *et al.* (2019) 'Autonomous Latching System for Robotic Boats', in *2019 International Conference on Robotics and Automation (ICRA)*. *2019 International Conference on Robotics and Automation (ICRA)*, Montreal, QC, Canada: IEEE, pp. 7933–7939. doi:10.1109/ICRA.2019.8793525.

Mateos, L.A. (2020) 'AprilTags 3D: Dynamic Fiducial Markers for Robust Pose Estimation in Highly Reflective Environments and Indirect Communication in Swarm Robotics', *arXiv:2001.08622 [cs]* [Preprint]. Available at: http://arxiv.org/abs/2001.08622 (Accessed: 18 January 2022).

Volden, Ø., Stahl, A. and Fossen, T.I. (2021) 'Vision-based positioning system for auto-docking of unmanned surface vehicles (USVs)', *International Journal of Intelligent Robotics and Applications* [Preprint]. doi:10.1007/s41315-021-00193-0.

Wang, J. and Olson, E. (2016) 'AprilTag 2: Efficient and robust fiducial detection', in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Daejeon, South Korea: IEEE, pp. 4193–4198. doi:10.1109/IROS.2016.7759617.

Yang, S., Scherer, S.A. and Zell, A. (2013) 'An Onboard Monocular Vision System for Autonomous Takeoff, Hovering and Landing of a Micro Aerial Vehicle', *Journal of Intelligent & Robotic Systems*, 69(1–4), pp. 499–515. doi:10.1007/s10846-012-9749-7.

Lars Digerud

NTNU
Norwegian University of
Science and Technology