FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

U.PORTO

FEUP FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO

A Self-Guided Docking Architecture for Autonomous Surface Vehicles

Pedro Nuno Barbosa Leite

MASTER'S DEGREE IN ELECTRICAL AND COMPUTERS ENGINEERING

Supervisor: Prof. Andry Maykol Gomes Pinto

Co-Supervisor: Prof. Aníbal Castilho Coimbra de Matos

July 17, 2019

Resumo

Os Veículos Autónomos de Superfície (ASVs) providenciam a plataforma ideal para explorar ainda mais a fundo as muitas oportunidades provenientes da indústria marítima. Estas embarcações podem ser utilizadas para realizar tarefas que hoje em dia se apresentam como monótonas ou até mesmo perigosas para o ser humano. Contudo, uma vez que a maioria do comércio mundial utiliza o ambiente marítimo como meio de transporte de carga, esta mostra-se como a maior oportunidade para a aplicação destes navios autónomos.

Atualmente, uma das operações com a maior quantidade de riscos associados é a de atracagem, uma falha neste processo pode, não só, levar a grandes perdas monetárias, mas também a catástrofes ambientais e até mesmo casualidades entre membros da tripulação. Considerando que atualmente a maioria dos acidentes marítimos são causados por erro humano, a inclusão de navios capazes de conduzir esta operação autonomamente permitirá minimizar a probabilidade de ocorrência deste tipo de erros, reduzir os custos desnecessários com tripulação excessiva e ainda, aumentar a eficiência dos espaços de carga, tornando assim a indústria mais segura, fiável e rentável.

Com este propósito em mente, esta dissertação propõe uma arquitetura auto-guiada para a atracagem de ASVs, composta por três módulos distintos, capaz de conduzir em segurança esta operação em múltiplos cenários e condições marítimas. Como parte desta arquitetura, é implementado um sistema de controlo e supervisão capaz de dirigir a operação, escolhendo qual módulo deverá operar a cada instante. A aproximação à doca deverá apenas ter início quando todas as condições que garantem uma navegação segura são atingidas. Para tal, dois dos módulos funcionam como suporte, o primeiro encarregue de conduzir o ASV para uma área predefinida onde a embarcação pode usar os sensores que tem disponíveis a bordo para identificar e reconhecer a doca, por sua vez, o segundo módulo é responsável por garantir que o ASV está orientado com a plataforma, assegurando que informação fiável sobre a mesma possa ser recolhida. O módulo central desta arquitetura é então composto por um sistema de perceção que visa o reconhecimento da plataforma, combinando informação proveniente de múltiplos sensores, e por um módulo de manobra que assegura uma navegação segura para o local de atracagem através de uma heurística de verificação de viabilidade.

Um cenário simulado de teste que permite ao utilizador configurar parâmetros ambientais, como vento e ondas, foi utilizado para validar a arquitetura proposta em situações distintas e avaliar o seu comportamento e desempenho. Os resultados experimentais demonstram a capacidade desta arquitetura detetar a doca com bons valores de fiabilidade, provando a eficácia da abordagem tomada para o sistema de perceção. O módulo de manobra demonstra-se funcional, conduzindo a operação de atracagem de forma segura, atingindo erros de posição e orientação até 0.107 m e 0.007 rad, respetivamente.

Abstract

Autonomous Surface Vehicles (ASVs) provide the ideal platform to further explore the many opportunities in the maritime industry. These vessels can be utilized to perform tasks that are currently monotonous, or even dangerous to human beings. However, as the majority of world's trade uses the maritime environment as a means of cargo transport, this proves the biggest opportunity of application for ASVs.

Nowadays, one of the operations that has the most associated risks is the docking one, a failure in this process can lead not only to great monetary losses but also, to drastic environmental pollution and even harm to crew members and alike. Taking into consideration that most of the shipping insurance losses are caused by human error, the inclusion of ships that are able to autonomously perform this operation will minimize the likelihood of such errors, reduce unnecessary costs with excessive crew and, increase the efficiency of cargo spaces, leading to a safer and more reliable industry.

With this purpose in mind, this dissertation proposes a self-guided architecture for docking ASVs, composed by three distinct modules, that is capable of safely conducting this operation in multiple scenarios and maritime conditions. As part of this architecture, a supervisory control system oversees the operation, choosing which module should operate at each moment in time. The approach towards the dock should only begin when all the conditions that ensure a safe navigation are met. Two supportive modules were then implemented, the first tasked with navigating the ASV to a predefined area where it can make use of its suite of on-board sensors to identify the dock and, the second, responsible for guaranteeing that the vessel is oriented with the platform so that reliable data can be retrieved from it. The central module of the architecture is then composed by a Situational Awareness system that aims to identify and recognize the platform by combining information from multiple sensors, and by a Maneuver module that ensures a safe navigation towards the dock through a viability check heuristic.

A simulated scenario that allows for user configuration of multiple environmental parameters, such as waves and wind, was used to validate the proposed architecture in distinct scenarios and evaluate its behaviour. The experimental results demonstrate the architecture's ability for detecting the docking platform with good fidelity values, proving the effectiveness of the taken approach for the Situational Awareness System. The Maneuver module also proves reliable, conducting a safe operation towards berth, achieving errors up to 0.107 m in position and 0.007 rad in orientation.

Agradecimentos

Ao meu orientador, Prof. Dr. Andry Maykol Pinto, agradeço todos os seus conselhos, *pep-talks* e fundamentalmente todo o tempo e esforço despendido para que os frutos deste trabalho alcançassem tanto as suas como as minhas expetativas. Não poderia deixar de agradecer ao Prof. Dr. Aníbal Matos, pela oportunidade de integrar a equipa de investigação do CRAS assim como as suas palavras e *inputs* para o meu futuro profissional. De entre os vários que me acolheram, integraram e ajudaram prontamente, o meu especial obrigado ao Daniel Campos cujas inúmeras contribuições foram fundamentais para o desenvolvimento deste projeto.

Aos de sempre, Diogo Duarte, Bruno Miranda e Joana Gomes, um grande obrigado por todos estes anos de amizade e suporte, independente de distâncias e culturas. Ao Daniel Barbosa, David Lamas e Isabel Geriante um eterno obrigado, pelas aventuras, momentos e, sobretudo, por serem a minha *safety net*. Ao Renato Cruz, pelas conversas sem filtros sobre tudo e sobre nada, pelos projetos utópicos que surgem dos nossos *brainstorms* e por toda a amizade que construímos desde os tempos do CIC, um obrigado.

Àqueles que marcaram estes 5 anos cheios de memórias inesquecíveis, Francisca Pereira, a melhor parceira no Party&Co e a responsável por todos os rabiscos nos meus cadernos; José Caires, o meu pupilo do humor negro e descabido; Maria Alexandra, a dedicação em pessoa, portista ferranha e a que (não) sabe programar; Vitor Carneiro e Renato Silva, 3AM, a melhor equipa de estudo, os alvos de *bullying* construtivo e acima de tudo os pilares de todo este percurso; a todos, o mais sincero dos obrigados.

À C.O.P.A. e a todos os seus integrantes, um especial agradecimento por me ajudarem a manter a sanidade mental ao longo deste período e proporcionarem inúmeras gargalhadas e histórias para contar.

Finalmente, à minha mãe, por todo o amor e carinho, pelo imensurável esforço e dedicação, por acreditar em mim e fazer os possíveis e impossíveis para dar corda aos meus sonhos, o maior dos agradecimentos. Aos meus avós e restante família, agradeço o suporte, confiança e todas as boas palavras que ajudaram a moldar aquilo que sou hoje.

Esta dissertação é o culminar de 5 anos de esforço e dedicação, a todos os professores e colegas que contribuíram, direta ou indiretamente, para o meu percurso e formação, um sentido obrigado.

Pedro Nuno Barbosa Leite

“Strive for perfection in everything you do.
Take the best that exists and make it better.
When it does not exist, design it.”

Sir Henry Royce

Contents

List of Figures

List of Tables

Acronyms and Symbols

2D	Two-Dimensional
3D	Three-Dimensional
AAWA	Advanced Autonomous Waterborne Applications Initiative
AI	Artificial Intelligence
AMS	Amsterdam's Advanced Metropolitan Solutions Institute
APS	Automatic Plug-in System
ASV	Autonomous Surface Vehicle
AWARE	All Weather All Roads Enhanced vision
CFAR	Constant False Alarm Rate
COLREG	International Regulations for Preventing Collisions at Sea
DIMECC	Digital, Internet, Materials & Engineering Co-Creation
EO	Electro-Optical
EPV	Expected Present Value
FOV	Field Of View
GNSS	Global Navigation Satellite System
GPS	Global Positioning System
HD	High Definition
HSV	Hue Saturation and Variance
ICP	Iterative Closest Point
IMO	International Maritime Organization
IMU	Inertial Measurement Unit
IR	Infra-Red
LIDAR	Light Detection And Ranging
LWIR	Long-Wave Infra-Red
MIT	Massachusetts Institute of Technology
MOL	Mitsui O.S.K. Lines
MRB	Maritime Broadband Radio
NIR	Near-Infra-Red
ODE	Open Dynamics Engine
PCA	Principal Component Analysis
PLY	Polygon File Format
PRM	Probabilistic Roadmap
RGB	Red, Green and Blue (aditive color model)
ROI	Region-of-interest
SA	Situational Awareness
SAC-IA	Sample Consensus Initial Alignment
SAR	Synthetic-Aperture Radar
SVM	Support Vector Machine
SWIR	Short-Wave Infra-Red

TEB	Timed-Elastic-Band
UAV	Unmanned Aerial Vehicles
UPS	Uninterruptible Power Supply
USV	Unmanned Surface Vehicle
UTM	Universal Transverse Mercator

Chapter 1

Introduction

1.1 Context and Motivation

An ASV – Autonomous Surface Vehicle can be defined as a vehicle capable of moving and orienting itself in multiple environments, relying on sensors to gain understanding of its spatial awareness, as well as using control algorithms that make all the maneuver operations autonomous, ranging from obstacle avoidance to the docking sequence.

A natural step in the maritime industry would be to adopt the utilization of ASVs since these provide the ideal platform to substitute human labor in tasks and services with potential associated danger, such as war mine clearance, environmental monitoring in case of a disaster and off-shore platform maintenance, dull tasks like coastal patrol and reconnaissance or even in scientific research purposes [1]. In spite of these applications, the most valuable opportunity is the maritime cargo shipping, according to the International Maritime Organization[1] (IMO) around 90% of the world's trade is carried by the sea, in a process that is by far the most cost-effective when compared to road transport, yet it still shows an immense room for improvement and optimization via the use of autonomous ships.

Operation costs relative to the shipping industry differ from ship to ship, depending on various factors, among them company policies or the ship's type and age. In a statistical analysis conducted by Fraunhofer's Center for Maritime Logistics and Services [2], an average for ships' operating costs was obtained by taking a large number of *panamax* bulk carriers as a reference for structuring these costs.

Table 1.1: Operating Costs of the reference vessel [2].

Cost Type	Includes	Average Cost per year	Share of Operating Costs
Crew Costs	Wages, Travel, Provisions, etc.	833.149 €	45%
Stores & Consumables	General Stores (e.g. deck, cabin, engine) and lubricants	254.793 €	14%
Maintenance & Repair	Routine basic work on-board & more complex work during port calls	236.279 €	13%
Insurance	Hull & Machinery, Protection & Idemnity	275.953 €	15%
General Costs	Expenses associated with administration &amp; management	237.161 €	13%
Periodic Maintenance	Dry dockings	370.288 €	-

[1] https://business.un.org/en/entities/13

Taking a closer look at table 1.1, the larger share of the costs related to the operation of a cargo ship is associated to the crew, accounting to 45% of the costs. Once eliminated the necessity of human labor aboard the ship, the design of the vessel can be completely revamped to improve its efficiency and aerodynamics. Systems and compartments once needed to make the ship livable would become irrelevant and could be completely removed making space for more cargo aboard the ship, increasing each trip's efficiency and reducing all monetary costs related to the crew (833.149 €/year), life supporting systems such as medical and safety equipment (59.000 €/year), representing up to around 900.000 €/year in savings [2], these operating costs portray 20% of the total cost structure, as seen in figure 1.1. The introduction of autonomous shipping will not only directly impact the operation's cost structure, but the immense stream of accessible data will revolutionize the industry, allowing companies to manage their fleet by looking at data from individual ships as *"... they will be able to identify the best combination of route, cargo, maintenance schedule ... getting the maximum value from a set of very expensive assets"* [3].

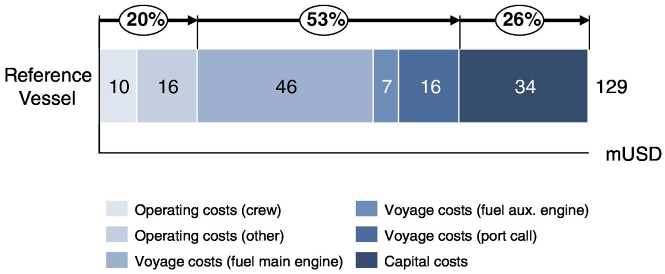


Figure 1.1: EPV (Expected Present Value) of costs over lifetime for reference bulk carrier [2].

Another key issue to be mitigated is the amount of accidents in the shipping industry, according to Allianz Global Corporate & Specialty [4], about 75% to 96% of all accidents related to the maritime shipping industry could be attributed to human error. The use of a completely autonomous boat would directly impact these numbers, greatly decreasing errors (not eradicating these, since the monitoring of the ship is still man made therefore being subjected to value judgments and potential errors) would consequently cut liability costs, losses of cargo and the implied pollution.

One of the biggest challenges faced by Autonomous Shipping is the lack of legislation, these vessels can't sail in international waters since the IMO enforces the COLREG (International Regulations for Preventing Collisions at Sea) [5], which is directed at regular ships. Once proven the viability and safety of the autonomous ships, the adaptation of these regulations would be rather straight forward since most of them are applicable to ASVs nonetheless, however, clarification would be needed for rules that directly imply the existence of a crew aboard the vessel [6]. The liability insurance related to potential accidents is also a major concern, this topic is still being vividly discussed within this industry. Subjects linked to AI (Artificial Intelligence) decision making in case of a potential crash and who is responsible for these decisions, either the ship's owner himself, the control room operator or even the system provider, are still up for debate. The fact that no crew is needed aboard the ship also raises some questions as this matter brings some potential risks for the vessel in case of an accident or hardware failure at sea, a particular example would be a fire aboard the ship that could become catastrophic to the vessel's structure. Preventive measures to avoid and mitigate the odds and impact of such specific accidents should be taken, arising new design and system requirements. In a highly connected Era, the reliability of the communication link to shore is mandatory for autonomous ships.*" Such communication will need to be bidirectional, accurate, scalable and supported by multiple systems creating redundancy and minimizing risks"*[3]. Matters like cybersecurity are also extremely important, if hacked the ship can be re-routed causing immense losses, such regulatory concerns are expressed in detail by the Danish Maritime Authority [7].

Ports and harbours are usually located in close proximity to urban centers, posing a very important factor for a country's economy. This fact makes accidents related to said facilities even more alarming as these tend to be associated with water pollution and soil contamination. According to a historical analysis [8], 56.5% of the accidents in seaports are related to transport operations (entering or leaving the port). From these, 43.6% are caused by an impact or collision between ships, or between a ship and the physical infrastructure. The latter, corresponds to 45% of the impact accidents, making the docking operation one of the most dangerous ones. For ASVs to secure the aforementioned regulatory approval, these need to be at least as safe as current vessels, the inclusion of ships that are able to autonomously perform this operation will minimize the likelihood of such errors, reduce unnecessary costs with excessive crew, and increase the efficiency of cargo spaces, leading to a safer and more reliable industry.

1.2 Objectives

The main objective of the present dissertation is to study and propose a new architecture for scene understanding, composed by complementary perception techniques that combine multiple sensors to detect a structure that can be used as a mooring facility (whose model is previously known), as well as, to calculate the relative pose of the ASV to a specific target point within the docking platform, with high levels of fidelity. Therefore the objectives of this work include:

- Propose an architecture capable of docking the ASV in multiple scenarios and diverse sea-state conditions. This architecture is composed by three operation modes, namely the Catch-Zone Approach, the Reorientation module and the Docking Approach;

- Supervise and ensure the safe navigation of the ASV towards the berth through a viability check heuristic, by evaluating the estimated target's associated confidence value and checking the surrounding scene representation for possible collisions;

- Combine information from distinct sensors such as LIDAR, Camera, GPS and IMU, in a complementary and reliable manner that enforces the capability of an ASV to understand the surrounding environment;

- Study the impact of the environment (weather conditions, tidal waves, wind) on the architecture's performance.

1.3 Structure

Besides the introductory section (chapter 1), the present document contains a chapter dedicated to the state of the art (chapter 2), which is divided is three main sections. The first, section 2.1, offers some context about the autonomous shipping industry, section 2.2 presents a rough comparison between several sensors used in maritime applications and how the information gathered by them can be fused to make the ASV aware of its surroundings and section 2.3 documents several planning algorithms used in related applications. To provide some closure to the State of the Art chapter, section 2.4 provides a critical analysis upon the conducted research and its main withheld thoughts. Chapter 3 describes the proposed architecture, detailing each operating mode. The results obtained from validating said architecture are presented and discussed in chapter 4. Finally chapter 5 offers a brief overview over the developed work and how it fulfills the proposed objectives. Future work is also elaborated upon in this chapter.

1.4 Scientific Contributions

This work originated two articles that were presented at the IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC) 2019:

- Pedro Leite, Renato Silva, Aníbal Matos and Andry Pinto, "An Hierarchical Architecture for Docking Autonomous Surface Vehicles", *IEEE International Conference on Autonomous Robot Systems and Competitions*, 2019 [9].

- Renato Silva, Pedro Leite, Daniel Campos and Andry Pinto, "Hybrid Approach to Estimate a Collision-Free Velocity for Autonomous Surface Vehicles", *IEEE International Conference on Autonomous Robot Systems and Competitions*, 2019 [10].

Chapter 2

State of the Art

2.1 Contextualization

There are currently no fully-scaled autonomous ships travelling the sea, most ASVs are either proofs of concept or rather small and used for very specific tasks such as the ones stated in Chapter 1, however the *"... scope of newly designed autonomous vessels is shifting from small ASVs for scientific researches to bigger cargo crafts..."* [11]. Some of these large scale cargo ships have their first missions planned for 2020, such as *Yara Birkeland* from Kongsberg, which will be expanded upon on this chapter alongside other relevant initiatives. Thematics like the changes needed in the infrastructures of ports and harbours will also be discussed in chapter 2, ranging from the construction of control centers to the integration of newly developed technologies that help and simplify the mooring operation, many are the changes that ports should attend to. The creation of test areas exclusive for ASVs is also another topic of discussion for this chapter, as these offer a valuable opportunity for real and thorough testing. Finally, some of the future projects that involve ASVs are presented in sub-section 2.1.4, giving emphasis to the fact that this is very much a growing industry, with projects all over the world with the most varied applications.

The input from sensors is heavily relied upon to provide information that allows the robot to recognize the docking platform. During section 2.2 of the present chapter, a comparison between the sensors used in maritime environments is provided, as well as multiple approaches on how the information these gather is combined to make the vessel aware of its surroundings. The complementary part of the perception system is the maneuvering towards the dock itself, hence section 2.3 presents the conducted research on planning algorithms and how these would fare during a docking operation.

2.1.1 Current Initiatives for Autonomous Vessels

MARINER$^{\text{TM}}$

Maritime Robotics has developed an Unmanned Surface Vehicle (USV) named the *MARINER*, this vehicle is currently being used for applications connected to the military as well as within geophysical exploration and environmental monitoring. It is a great USV for data acquisition and patrols in a 24/7 time basis as the vessel has proven to be quite stable, able to transport payloads up to 1 m^3 and has long-range capabilities, making it nearly maintenance free.Along with the vessel, Maritime Robotics also offers a monitoring system for the remote operator of the Mariner featuring information about the navigation, engine and electronics status and the ability to inspect the surroundings of the ship via EO (Electro-Optical) and IR (Infra-Red) cameras.

Volvo Penta's Self Docking Yatch

This Volvo Penta prototype has its focus on making the docking process as autonomous as possible, however it requires human supervision and action. The developed system undertakes three main phases to its process, in a preliminary stage the system identifies the dock and signals the captain that it is ready to perform the operation. Once the human command has been given, the boat uses GPS signals to move into a "docking ready position", the final stage must be initiated via the ship's captain as the system uses the GPS and on-board sensors as well as sensors placed on the dock to berth the boat safely. Although it relies on human interaction, this docking technology is considered a pioneer on the matter, stating that its sensors and on-board computers can react to hostile sea conditions like wind and tides in a matter of milliseconds.

Rolls-Royce

One of the cornerstones of the industry and major player on the Advanced Autonomous Waterborne Applications Initiative (AAWA) is Rolls-Royce. The AAWA's project brings together companies from a diverse area of operation such as satellite communications, automation and control, ship design and even partnering up with universities, the latter tasked with research about technology, legal and business aspects. According to the AAWA, the technology necessary to build an autonomous ship does exist, it is just a matter of finding the *"... optimum way to combine them reliably and cost effectively"*[3]. Their vision for a future time-line is presented in figure 2.1, in early development the ship will rely on a reduced crew and will only operate certain functions, shifting focus to a fully autonomous vessel as the technology evolves and gets fault proofed, aiming at sometime in the next decade for a finished product.

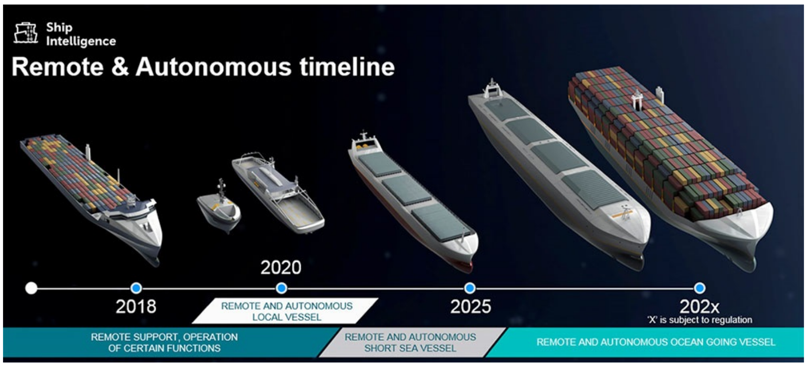

Ship Intelligence
Remote & Autonomous timeline
2018
2020
REMOTE AND AUTONOMOUS LOCAL VESSEL
2025
202x
'X' is subject to regulation
REMOTE SUPPORT, OPERATION OF CERTAIN FUNCTIONS
REMOTE AND AUTONOMOUS SHORT SEA VESSEL
REMOTE AND AUTONOMOUS OCEAN GOING VESSEL

Figure 2.1: Rolls-Royce's remote and autonomous operations time-line[1].

[1] https://www.rolls-royce.com/products-and-services/marine/ship-intelligence

Yara Birkeland

Yara provides the perfect example of how this technology will be implemented in the near future. As a leading force in the fertilizer market, Yara pretends to transport their goods utilizing the first ever fully electric autonomous ship, with zero emissions. The vessel, named *YARA Birkeland*, will utilize Kongsberg's integrated control and monitoring systems featuring an automatic mooring system that requires no special preparations on the port's physical infrastructure to to enable an autonomous docking operation which makes use of the on-board proximity sensors (radar, camera, etc) and communication systems such as the Maritime Broadband Radio (MRB) and the Global Positioning System (GPS) to gain perception of the docking platform and to guarantee a smooth maneuver. Operations will begin as a manned vessel around 2018, moving to remote operation in 2019 within a fixed route between the Brevik, Herøya and Larvik ports alongside the Norwegian coast, as shown in figure 2.2. Finally, in 2020, the vessel is expected to perform fully autonomous operations [12]. Shore control centers will be responsible for the handling of all aspects of operations, such as monitoring, decision support and surveillance. An interface towards Yara's logistical operation will also be implemented at the Herøya's center.

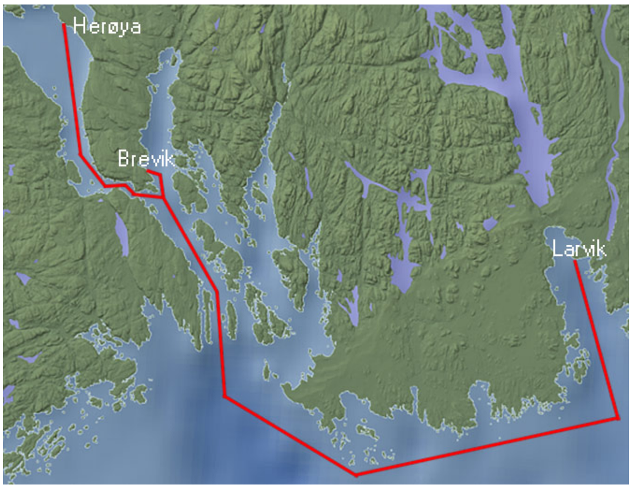


Figure 2.2: Yara Birkeland's defined route[2].

Folgefonn

A fully autonomous operation of a full-sized vessel such as the ferry *Folgefonn* has been conducted by Wärtsilä on one of Norway's maritime routes. Human intervention was only needed to define the destination of said route and once given the order, the ferry autonomously left the harbor, sailed towards the next port, maneuvered through the harbor and docked alongside the terminal. The navigation throughout the route was controlled via the use of multiple way points, and Wärtsilä's Dynamic Positioning system is used to assess and supervise the vessel's velocity and position within the predefined trajectory, using the Global Navigation Satellite System (GNSS) as its primary sensory tool. Its auto-docking prototype system has also been successfully tested both

[2] https://www.kongsberg.com/maritime/solutions/ship-types/autonomous-ships/

on a simulated platform and on a real harbour, to assure that this maneuver is conducted as safely and as efficiently as possible the system is provided with all the necessary inputs such as wind and wave conditions, the vessel's size and weight and finally its current orientation. *Folgefonn* is also capable of operating fully electric as it is equipped with and hybrid powering system that resorts to Wärtsilä's own technology for wireless inductive charging of the batteries.

2.1.2 Port Side Infrastructure

While the maritime industry waits further development of these autonomous ships, ports around the world are starting to become more intelligent, focusing on operational efficiency, interaction between stakeholders and even environmental concerns. A prime example of this, is the Port of Rotterdam as it started developing a centralized dashboard application to compile real-time information about water, weather conditions and communication data. This data is used to determine optimal times for ships to dock, load and unload, reducing wait times and upping the profitability. Still on the subject of obtaining information about the port's operation, the Port of Valencia installed a network of sensors on its cranes and carrier vehicles, collecting useful data about power consumption that can be used to compute better ways to schedule them, reducing idle times.

With these new autonomous ships, ports and harbours will need to adapt their physical and digital infrastructures so that ASVs can safely conduct all the necessary operations. As the port environment grows more complex, considerations for safe and successful autonomous operations start to arise, SmartPort, a venture between the Port of Rotterdam Authority and several knowledge institutes is one of the main forces in fomenting scientific research on this area, also providing financial help for these studies. Some of the key aspects for smart and autonomous shipping operations inside a port include a shore control center for monitoring and navigation control of the ASVs, a well defined role for the port authorities (would these be responsible for all the port operation or just provide the resources) and finally the need for investment in infrastructure [13].

Physical Infrastructure

A vital aspect to consider is the need to adapt the port's physical infrastructures to ensure the safe mooring and unmooring of autonomous ships. This operation can often be rather dangerous, possibly causing harm to the vessel, port infrastructure and even the mooring crew, a great innovative technology to solve these problems is automated berthing, allowing the ships' mooring without the use of ropes, it consists in attaching vacuum pads to the vessel which pull it closer to its berth [14]. Currently on the market, there are two main companies providing these solutions, being Cavotec and Trelleborg. The first product, from Cavotec, MoorMaster aims to optimize the interface between the vessel and the control unit on shore, allowing ports to economize in infrastructure costs. The company states[3] that berthing can be completed in 30 seconds, and that any ship can be berthed, even the ships larger that the berth itself. This system also offers real-time

[3] http://www.cavotec.com/en/your-applications/ports-maritime/automated-mooring

monitoring of the process. The latter solution, AutoMoor from Trelleborg, makes use of vacuum-based technology to secure the vessel at berth, being suitable for a wide range of environmental conditions. Operation can be supervised by using SmartPort technology to monitor the process. As a rather new technology there are associated risks. These systems require a constant source of power available at shore, even if operation is secured by a UPS (Uninterruptible Power Supply), there is no back-up plan that accommodates a supply failure. Another pertinent concern is the fact that vessels will most likely not have a smooth surface to attach to, fruit of wear and tear or corrosion, creating quite a challenge for a *"... safe and continuous vacuum connection"* [14].

A final aspect that ports need to take into consideration, is that future ships aim to be fully electric (e.g. *Yara Birkeland*, section 2.1.1), therefore, ports must be prepared to handle these ASVs. Cargotec also offers a solution, the Automatic Plug-in System (APS), that helps charging the ship at the port, *"... it can be done without any human intervention with a few added advantages namely, short charging time, minimal infrastructure cost and minimal maintenance costs"* [15].

Digital Infrastructure

As stated before, a shore control center is a must at port side, as it would *"... ensure complete monitoring of the vessels, including, efficient path-planning operations to achieve maximum efficiency"* [15]. When it comes to the digital infrastructure, ports will need to improve their communications systems to provide better connectivity between the ASV and the onshore control center [16], to achieve this, the Inmarsat technology is taken as the best bet [15], as it provides a global coverage in satellite communications and has a long history of maritime applications. Some other relevant maritime communication systems have been developed at INESC TEC, the BLUECOM+ solution, for instance, *"... enables cost-effective broadband Internet access at remote ocean areas using standard wireless access technologies..."* [17]. In 2018, André Coelho *et. al* [18] experimented on multiple Wi-Fi bands for communications between a base station placed on shore and an USV, evaluating the influence of factors such as distance and orientation on the Wi-Fi's quality link. From this work's experimentation, 5 GHz bands prove more suitable than the 2.4 Ghz for shore-to-USV Wi-Fi communications [18].

2.1.3 Test Areas

Autonomous vessels are rapidly growing and aim to take maritime operations by storm in the near future. For this to happen, exhaustive testing in real sea conditions is critical to ensure the functionality and durability of systems and technology. With this purpose in mind, two large testing sites are currently under construction.

One of them has started construction in 2018, in the South China Sea coast of Guangdon, it extends over roughly 772 km^2 and it is the first in Asia. This large area will allow the testing of navigation and colision avoidance technologies [19]. The other test area is located in the coastal area of Finland and it is hosted by DIMECC (Digital, Internet, Materials & Engineering Co-Creation) as part of the ONE-SEA collaboration project, that will be expanded upon in section 2.1.4. The Jaakonmeri Test Area, as it is named, is open to all companies or research institutes that wish to test their technologies in this open water area. It is to be noted that ice conditions may also be tested during the winter, making this quite a valuable perk for this testing area.

2.1.4 Industry's Next Steps

Until the end of the decade autonomous boats will be travelling the coasts in small fixed routes, with a smaller crew [4]. As the technology is fault proofed these routes can become bigger and more autonomy can be given to the vessel until a fully autonomous ship can be achieved. The long term implications imposed by the lower cost of cargo transportation would affect a large array of industries, creating new opportunities for businesses that once were not viable, such as some food types, clothing and other lower value items that were not worth shipping to certain locations, could become profitable [12]. As stated above, some fully scaled cargo ships are planned to set sail in the near future, being perfected and fully autonomous during the next decade like the ones discussed in section 2.1.1, however as discussion on this matter progresses, new projects arise and join the race to rule the sea. In the beginning of 2017, Japan's Minister of Land, Infrastructure, Transportation and Tourism unveiled a joint project between some of the biggest ship building companies in Japan like MOL (Mitsui O.S.K. Lines), Mitsui Engineering and Shipbuilding and many others. The project aims for the year 2025 and is set to "... *develop the technological concept for autonomous vessels, drawing upon the strengths of each participating company and organization, setting a course toward development of the technology needed to realize autonomous vessels that can provide reliable, safe, and efficient ocean transport"* [20].

China is ranked as the top three largest marine shipping countries, having the chance to be the most dominant by 2030 [19]. This may be supported by the fact that the country is currently developing a fleet of autonomous ships and a testing area (section 2.1.3). Once proven ready to operate autonomously, these ships will not only be used in the shipping industry, cementing even more China's position as a big player, but also in maritime patrolling tasks.

Another big force behind the growth in the autonomous shipping area is the ONE-SEA[5] Finnish collaboration, joining the efforts from maritime related technology suppliers, shipyards and ship owners as well as research organizations. It aims to introduce a new industrial standard for maritime traffic autonomous ships by 2025, as well as lead the way to the commercial viability of these ships. Some other smaller scale projects are also in motion, e.g. in Amsterdam researchers from the AMS Institute (Amsterdam's Advanced Metropolitan Solutions Institute) and the MIT (Massachusetts Institute of Technology) have joined forces in the *Roboat Project* that aims to reduce the traffic in the streets by utilizing the city's canals and populating them with autonomous vessels for transportation, food delivering and even garbage collection.

[4] https://www.ship-technology.com/projects/yara-birkeland-autonomous-container-vessel/
[5] https://www.oneseaecosystem.net/

2.2 Situational Awareness

The autonomous docking process can be divided in two distinct stages: Perception and Maneuver. The first stage ensures that the ship is aware of its surroundings, highlighting possible hazards for the vessel and recognizing the harbor or possible docking location, the latter stage of the operation is tasked with guaranteeing that the vessel follows the correct trajectory from its current position into the goal location, the previously identified dock.

During the current section some of the most relevant technologies used for Situational Awareness (SA) will be presented and briefly compared, as sensor fusion is the key aspect of the this stage of operation to provide real-time perception of the vessel's surroundings, some approaches used to process the information retrieved by the sensors will also be discussed.

2.2.1 Sensor Technology

Studies conducted for the autonomous driving sector combine information from multiple sensors, such cameras, LIDARs (Light Detection And Ranging) and radars [21, 22]. Accurate measurements of ranges and velocities can be obtained from a LIDAR, on the other hand cameras provide a cheap and easy way to capture visual information that allows for object classification. However, night time and weather conditions such as rain and fog do pose a harsh challenge. Different types of cameras can be used to mitigate these conditions and, combining their information with radar based sensors allows for the system to keep operating in most of these conditions. Such can be extrapolated to the maritime sector whose environment can be considered even more challenging for the SA technology, due to the constant bobbing of the ASV caused by the waves that leads to more unstable and unreliable sensor information.

An obvious choice to provide input about the ship's surroundings would be a camera. A normal visual spectrum HD (High Definition) camera is small, durable, relatively cheap and able to provide a very high spatial resolution that allows for object recognition and segmentation. However, these have some limitations when it comes to bad weather conditions, such as fog and heavy rain. Night time also presents a challenge being that its seeing distance, obviously, drops very quickly. This is the reason why these cameras are seen as an important asset to be fused with other sensor's data because by themselves they show far too many limitations [3]. Considering these conditions, better performances can be achieved using IR cameras. Long-Wave IR (LWIR) cameras can be used for situational awareness in total darkness, these are sensitive to an IR wavelength of 8-14 µm, being that thermal IR radiation is passively emitted by every object at a wavelength of around 10 µm, LWIR sensors are suitable independently of light conditions. Another option is a Short-Wave IR (SWIR) camera, its sensors operate in the 1-3 µm spectrum, forming images that are easy to interpret by using the light reflected by the objects. As stated by J. Wallace in [23], LWIR reconnaissance has some drawbacks, one being that reflecting objects are not easily recognized and depicted, complicating the interpretation of the produced images. Another drawback is the substantial attenuation in hot weather and high humidity, in these conditions SWIR provides the better performance, however, it can be advantageous to fuse the information from both sensors in order to operate in both wavelength ranges.

Further comparison can be made between the presented technologies by looking at N. Pinchon's *et al.* [24] project AWARE (All Weather All Roads Enhanced vision), whose aim is to enable vision in all poor visibility conditions such as night, fog, rain and snow. As most of these conditions are very relevant to the ship's SA, the presented inferences are rather interesting. The conducted experiments, presented in figure 2.3, show that LWIR cameras have a better capability to see through fog than the SWIR ones and, as stated before, the normal visual spectrum camera has the lowest fog piercing capability. The presented NIR (Near-IR) camera requires an active illumination font in order to be effective, this wouldn't be practical for an autonomous ship, for this very reason it was discarded as a relevant technology. When it comes to strong glares the LWIR camera provided the better output image, however for detecting these the SWIR camera shows the better solution as it is very sensitive to the glare in foggy conditions.

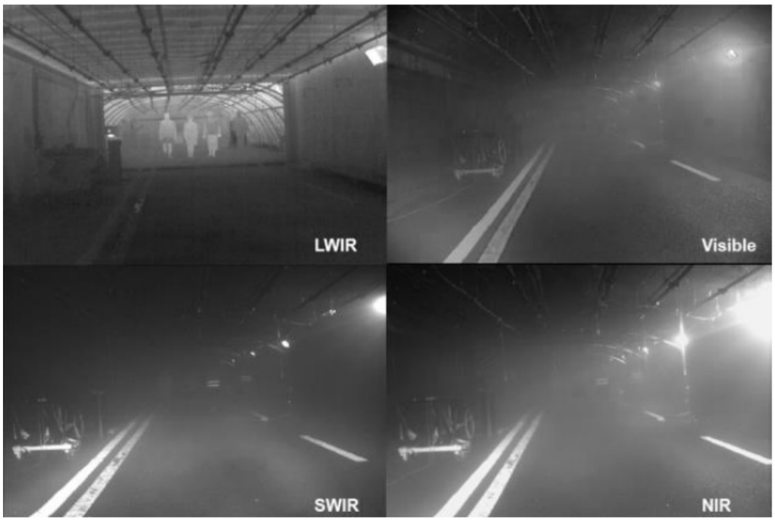


Figure 2.3: Project AWARE's conducted experiments for different cameras, in foggy conditions [24].

To improve a system's reliability and accuracy, it is necessary to add some redundancy and complementary characteristics, radars are the perfect complement to cameras, as they are more robust to weather effects and provide the depth analysis required to extract a 3D map of the visual scene. A combination of two monocular cameras can be used to implement stereo imaging to generate a 3D map, however it requires extensive processing performance and high-bandwidth transmission in order to solve the matching algorithm. A much better performance can be obtained by using radars or LIDARs. Radar technology has been used in maritime operations for a long time, typically using S- or X- bands, which are proven to be more robust in different weather conditions by S. Heuel (2013) [25]. However, its operating frequency resolution may not be good enough when it comes to reactive collision avoidance [3], take for instance an approach to the docking platform, the radar resolution must be accurate in short range. Bands developed for automotive applications like Ka and W- could be very useful as they offer a much better angular and distance resolution, at cost of reduced range.

A LIDAR is a sensing device that measures distances between itself and its targets by the means of a laser. By analyzing the reflected light, sampling the returning signal and measuring its travel time it generates a point cloud that can be utilized for mapping and localization. One of its advantages is being independent from ambient light, however as it typically relies on a IR laser beam, it is affected by adverse weather conditions. LIDARs have proven themselves worth it in multiple areas of application, obviously in autonomous vehicles for obstacle detection and safe navigation, prime examples of these are the the *Waymo* (formerly knows as Google Self-Driving Car Project) and the Singapore MIT Alliance (SMART) as the latter actively developing technologies for autonomous LIDAR vehicles. As these sensors are normaly used for 3D Laser Scanning, applications in areas such as geography, geology and forestry are very common [26, 27].

As previously stated, the usage of 3D LIDARs for high-resolution mapping is rather common, therefore this sensor is quite the perfect fit to detect the geometric features of the dock and generate a 3D point cloud, take this cloud as a matrix with three dimensions X, Y and Z. Figure 2.4 presents an example of a point cloud returned by the LIDAR sensor, taken from the simulation used for this project. The dock geometry is quite recognizable for the human eye, however there is a lot of noise involved in a maritime scenery, such as waves and weather conditions, which makes processing more difficult. For this very reason it is recommended to process the point cloud by means of filtering the unintended data [28].

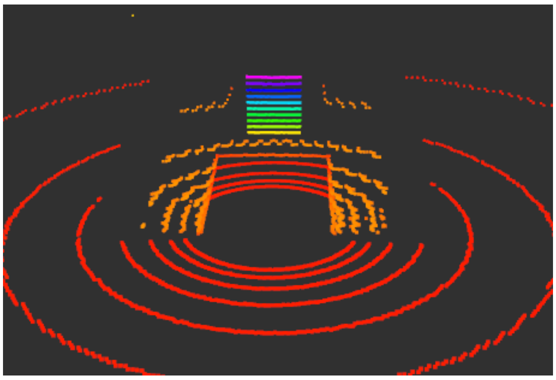


Figure 2.4: Example of a point cloud generated by the 3D LIDAR in a simulation environment.

Once established a solid input point cloud, one of the possible approaches is to project this 3D information into a plane, this information can then be processed as an image with computer vision algorithms, such as Harris Corner Detection [29], this method would allow for a clear identification of the dock's geometry and by computing the distance between corners, obtain a good target point for the final berth location. One other possible method is to use Hough's Transform to reconstruct the 2D shape of the dock [30], identify it's vertices and finally generate a target point. These 2D based algorithms are quite more efficient than processing a 3D point cloud, presenting itself as a great solution if the available technology has low processing power, however, completely rely on the sensors' resolution, and as seen before during the current section, LIDARs and other laser based technologies suffer from weather conditions such as fog, and also the limited range affects the quality of the generated data. Figure 2.5 is representative of a projection of the data into a 2D plane, both scenarios are given context with figure 2.6 a) and 2.6 c). The data collected within range of the docking platform with minimal waves and perfect weather conditions is depicted in figure 2.6 b), while 2.6 d) represents the constraints but with a variation on the distance from the dock. The quality of the data is affected in quite a degree, making the recognition process rather more complicated. We can then conclude that this method is capable of producing good results in close range applications, having the upside of needing a low end processing system.

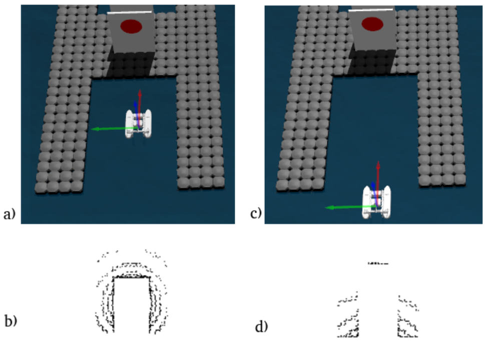


Figure 2.5: Comparison between two frames taken at different distances, obtained from a 2D perception algorithm. a), c) Context of the presented example for both close and further distance. b), d) Frames generated from the presented method, for close range and further distance, respectively.

A solution to bypass these resolution problems is the method proposed in 2014 by Joel M. Esposito *et al.* [28] that uses as input the 3D point cloud returned by the LIDAR sensor and a target cloud generated *a priori*. This target, is representative of a known dock model created from a 3D modeling software, or obtained via the LIDAR itself, under perfect conditions preferably, from a real environment. If the latter option is used, the density of the cloud might not be sufficient, leading to the need to concatenate multiple clouds generated in identical conditions [28]. The next step is to use the Iterative Closest Point (ICP) algorithm, it aims to determine correspondent pairs between the input cloud and the target one and estimate a transformation that aligns the correspondences, returning a homogeneous matrix that represents the rotation and translation between clouds that can later be used to estimate an optimal target point for the docking maneuver. The ICP has several criteria to terminate its iteration and return a final solution, these can be adjusted by the user to obtain a better and more faithful transformation. These criteria are the maximum number of iterations run by the ICP, a given convergence target (sum of the differences between the current and the last transformation) and finally a given fitness score (sum of the squared errors is smaller than a threshold) [6]. Figure 2.6, presents an example of the result from applying the presented method within the studied scenario. Represented in black is the target cloud, generated from a 3D model, the green color represents the input cloud, as we can see it is not aligned at all

[6] http://www.pointclouds.org/assets/icra2013/registration.pdf

with the target. On the right, presented in red is the output cloud from the ICP, already perfectly aligned with the target.

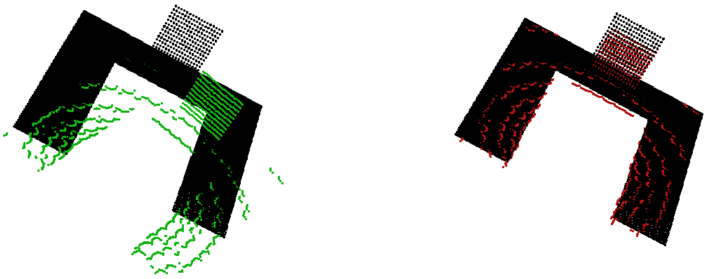


Figure 2.6: Example of the computed result from the ICP algorithm. On the left, the loaded template (black) and the input point cloud (green); On the right, the cloud is aligned with the target (red), after the ICP iteration.

The repeating geometry of the docks may lead to false correspondences from the ICP algorithm, for this reason Esposito *et. al* [28] propose using the Sample Consensus Initial Alignment (SAC-IA) method in order to *"... compute a rough estimate of the spatial transformation between the datasets"* [28], and feeding this result to the ICP algorithm. This method of processing the 3D LIDAR data is clearly more processing power demanding, however its results are quite impressive as it provides an easy way to calculate an optimal target point for the docking, combining this method with a camera in order to mitigate the laser's weaknesses would be a great solution for the current problematic.

2.2.2 Sensor Fusion

As stated before, the key to achieve an optimal perception of the robot's surroundings is sensor data fusion, each sensor has its flaws and weaker points. By adding sensory redundancy individual errors and weaknesses can be averaged out and better overall performance can be achieved, false positives and negatives can also be prevented this way. This section will present a rough comparison between multiple sensors that are used commonly in maritime applications and expose some of the approaches and algorithms that aim to recognize the docking platform and extract its important features by fusing information from some of the discussed sensors.

Table 2.1 presents a rough comparison between the technologies discussed in section 2.2.1, *"... based on the review of existing solutions reported in various fields of autonomy and the testing of sensor technologies already done in AAWA"*[3]. It is important to note the uncertainty of some of the presented parameters, mostly when the marine robustness is referred, as some of these technologies are still rather new and untested in maritime applications. A complete and reliable Situational Awareness system can be built around the data gathered from multiple sensors, radars are tolerant to harsh weather conditions and can easily provide tracking of a target's distance and its short-range version can be trusted for object detection in short ranges. The detected object data can be augmented by the visual HD cameras, these can provide detailed information and help identify the object through image processing such as segmentation and separation from the background. As discussed before, IR cameras are very useful in total darkness and foggy conditions providing valuable data, furthermore the fusion of all the camera and radar data in bound to increase detection robustness.

Table 2.1: Comparison of the presented SA sensors [3].

	HD camera	IR camera	Radar	Short-Range Radar	LIDAR
Spatial Accuracy	++	+	–	-	++
Field of View	+	-	++	-	+
Distance of Measurement	-	-	++	++	++
Object Identification	++	+	–	–	+
All Weather Operation	–	+	++	++	+?
Computational Load	–	-	++	++	–
Marine Robustness	++	++	++	+?	?

Situational Awareness systems usually fuse information from multiple on-board sensors such as cameras and LIDARs, assuring the needed redundancy in data that makes a perception system robust and reliable. Diverse configurations have been experimented on, using multiple cameras and even more than one LIDAR to guarantee that the gathered data is dense enough. However, some approaches rely only on LIDARs to obtain real-time information about the surroundings, matching it with templates built based on known information about the platform's geometry such as the ones presented in section 2.2.1. A completely different approach to detect the platform is based on off-board sensors, such as Synthetic-Aperture Radar (SAR) images obtained via satellite which are then compared with knowledge based models of the dock/harbor.

One of the more common approaches when it comes to the design of ASVs' perception systems is the fusion of information from cameras and LIDAR sensors, this aims to negate some of the weaker points of each sensor by adding redundancy. The work developed by J. Lee *et al.* [31] in 2017, utilizes a 2D LIDAR in order to locate the dock, and the camera to locate its features, in his work the considered dock for testing is that of the RobotX Challenge, therefore it consists of a combination of three docking bays with colored symbols associated with them. After the initial approach to the dock, when it is close enough to carry out image processing, the algorithm starts by identifying the correct target dock, a color thresholding method is used and the image is then down-sampled by using LIDAR-camera calibration data, this is done to improve the performance of the image processing and mitigate noises caused by glare and other environmental disturbances [31]. A region of interest (ROI) for the symbol is then calculated using this same data, once processed, shape identification is achieved by template matching and the color is classified by comparing the Euclidean distance between hue and saturation values of the ROI with those of the thresholds for the predefined colors. Once the correct docking platform is identified the ASV performs another scan of the area, calculating its desired position and carrying out the needed lateral position alignment.

Andrew Webb *et al.* [32] in 2016, utilizes a 3D LIDAR and five different cameras to identify the docking platform. The perception system presented by Webb runs two different nodes, one responsible for mapping and segmenting objects of interest from the point cloud, and other for shape and color classification. Given an input point cloud, the unwanted data is removed, for this a polygon is defined around the working area of the ASV and then a point-in-polygon algorithm is run, discarding any points that fall outside [33]. The remaining points are then filtered out based on a minimum intensity threshold in order to remove the water surface. Once filtered, the point cloud is separated in clusters based on distance thresholding, then a bounding box is defined around each cluster and used to update the current map. The ICP algorithm presented before is then iterated for each cluster looking for the best fit for the templates previously defined. After the object identification, the bounding box defined by the first node is transformed into the camera's frame, using this data to define a region of interest. From this region, the blob area, enclosing circle area and convex hull area are deduced, the relationship between these three attributes is then used to identify its shape. Segmentation of the image in the Hue Saturation and Variance (HSV) color space is then applied leading to a final shape and color.

A different approach for the same problematic was also presented in 2016, by Christopher M. Ash *et al.* [34], using four LIDARs and two wide-angled cameras equipped with IR and polarized filters, this configuration provides the sense of depth since they are angled differently from each other. The first step taken by Christopher and his team is to filter the data from the LIDAR that corresponds to the water surface, this is done by using an intensity threshold *"... as water is an IR black body absorbing most of the laser light"* [34]. All the data is then converted into a fixed frame, fused together and then used to fill a occupancy grid, this grid is then flattened into a 2D plane and used as a binary image from which contours can be extracted, resulting in a list of polygons from which multiple features can be computed (such as length, width, perimeter, etc...). Once the object has been detected and identified, its information is passed to the cameras for identification of visual properties such as color, the LIDAR information is transformed into the camera's frame and a ROI is defined, afterwards a color threshold is applied resulting in a binary image which is then compared with a shape classifier to generate a score for that color and shape combination.

A slightly different method is proposed by Joohyun Woo *et al.* in [35], as his work utilizes learning based recognition algorithms. The ASV used for testing is equipped with a single 2D LIDAR and a camera, the data provided by the LIDAR is saved into a grid map, representing the detected objects. To extract the dominant features of each shape, Woo utilizes Hu invariant moment for interpretation of the image's properties and the Principal Component Analysis (PCA) in order to convert this set of observations of possibly correlated variables into a smaller set that still contains most of the information in the larger one [36]. For shape classification, a Support Vector Machine (VSM) classifier was used, generating a hyper-plane used for classification of each shape. Once the shape is computed, the ASV must estimate the location of the dock. The data from both the camera and the LIDAR are used for this purpose, knowing the X, Y and Z position of each pixel of the designated shape and by using the lateral distance between the ASV and the dock, the bearing angle can be estimated, once determined, the corresponding distance to the dock can be measured by the LIDAR and a target point can be easily calculated. The fusion of these two sensors shows great results in all the presented approaches, even if most of these algorithms have only been tested for makeshift versions of the RobotX Challenge docks, the application in real life docking bays is only a step away with only minor adjustments needing to be made. The detection phase of the algorithms could be applied right away, although the templates for comparison would need to be reworked, using a method like the one discussed before (generating a target point cloud *a priori* with the LIDAR, in perfect weather conditions) proposed by [28]. The utilization of the cameras would have to be adjusted, since the features it would have to recognize wouldn't be as linear and straight forward, however the logic behind the system could still be utilized in a real environment.

An alternative approach for the problematic is the one present by Qi Chen *et al.* [37] in 2010, this work is based on detailed research about the harbor as an object to establish a harbor model and a general knowledge basis used for the detection algorithm. As harbors vary in shapes, numbers and types of docking platforms, establishing a uniform model would be a key point [37]. It is common knowledge that a harbor must be located in a coastline and, to protect it from the impact of the waves, a breakwater is normally constructed in front of the docking platforms. Qi Chen formulated the model presented in figure 2.7 as groundwork for analysis, with it a knowledge basis that serves an input for harbor detection algorithm was established.

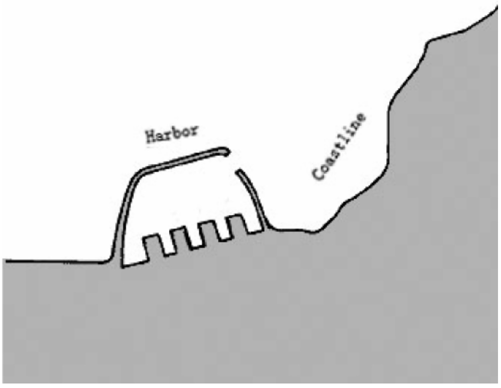


Figure 2.7: Formulated harbor model [37].

This algorithm is suitable for an ASV that has access to SAR images, or similar, as it needs an image of the sort as input along side the knowledge basis previously defined. Firstly the image is segmented, removing both the land and sea based on the knowledge that a *"... gray value of water is often lower and has better uniformity, while gray value of land is higher relatively and has worse uniformity"*[37], the result of an outline extraction of this newly segmented image is taken as the coastline. Next, candidates for the breakwater and dock regions are extracted based on the idea that these structures would have water on both sides, these are then discriminated recurring to the object features previously defined (length-width-ratio, for the matter). Afterwards, the structures that belong to the same harbor are merged together and enlarged and discriminated. An approach based on the same principle was presented by Zhu Bing *et al.* [38] in 2005, a knowledge basis is also established by studying the harbor's physical features and its common geographic placement. The proposed algorithm takes into consideration the possible difference in sizes between harbors, for that matter, two methods for extracting a candidate region from the input images are utilized. For a big harbor, it is assumed that the whole input image can be a candidate region, to make the recognition process less demanding, down-sampling is applied to the high-resolution image, the result is then segmented for land and sea removal based on thresholding. The resulting image is then used for identification of the harbor, the knowledge basis previously compiled is now used for feature detection on the semi-closed region. On the other hand, the image resulting from the big harbor segmentation serves as input for the medium and small harbor detection, as it is taken as a rough segmentation for later use. The method used for medium and small harbor detection, segmentation is achieved by using a block based approach, the input image is down-sampled and partitioned into blocks, from these, only the ones that represent the coastline are kept while the rest are filtered out. A candidate region is then defined by clustering these blocks using the nearest neighbor clustering algorithm, at the same time using the information provided by the image resulting from the big harbor segmentation to combine the segmentation results and filter out scattered clusters. In 2011, a slightly different algorithm is proposed by Y. Sui *et al.* in [39], Sui proposes a template-based detection method for harbors. This algorithm makes use of a previously assembled database, to generate these template maps, optical, multi-spectral or SAR images are selected and a ROI is defined. Important features from this region are then computed and binarized, and saved in the database. When a detected image arrives, a corresponding template is loaded (according to the harbor information), then a ROI is extracted by calculating the spatial transformation between the template and the input image. Segmentation is then carried out by fusing different input images, giving these different weights, and comparing them with the template information. The Constant False Alarm Rate (CFAR) algorithm is then used for extraction of the harbor targets, this was the chosen algorithm due to its proven worth in maritime applications [39]. Despite its great performance in extracting targets in sea backgrounds, some robust geometrical features were identified and fed to the CFAR algorithm, such as *"... length-width ratio, area and shape complexity etc, together with the center coordinates of targets to do change analysis"*[39], this was done to avoid some of its known difficulties in processing SAR images, e.g. the speckles present in these images. An ASV with access to satellite type images, could use the presented methods in this section to gain a general perception of where the docking bay or harbor is and direct itself towards it. However, these algorithms are obviously not intended for close range operation, for this matter, a combined use of these methods while the ASV is far away from the coast and plans to start a rough first approximation to the harbor, and one of the situational awareness algorithms that relies on on-board sensors in order to operate in closer proximity, could prove quite effective.

2.3 Planning Algorithms

The guidance subsystem is a key part of the autonomous ship, it takes into consideration several elements such as nearby obstacles, route from origin to destination and other navigational aspects (global path planner), as well as information provident from other vessels. This data is then combined to generate the ship's path, this can be achieved via a enormous number of algorithms, therefore only the most relevant for the problematic are going to be introduced in this section.

A-star

The A* algorithm was first introduced by Nilsson N. *et al.* [40] in 1982, it consists in computing the shortest path in a graph between the starting node and the final one. A heuristic function, f(n), is utilized to choose which nodes are searched through first, to obtain the least costly path. This function is given by the following equation:

$$f(n) = g(n) + h(n) \tag{2.1}$$

Take g(n) as a function representative of the cost associated from the selected node until the n^{th} node, and h(n) is the heuristic function that is used to estimate the cost to reach the final node. The algorithm works by expanding the initial node and compiling a pile of all the neighbours, ordering these for the value of their f(n) function (leaving the lowest score on top). The correspondent node of the value on top of the pile is then taken as the current node and expanded upon, iterating the process until a final solution if achieved [41]. The A* algorithm searches the most promising nodes first, based on the information of the closest ones and also of how much it's still left to reach the final goal, it is a complete and optimal algorithm and its complexity depends intrinsically of the quality of the chosen heuristic function [42]. However, the way it is firstly presented, it does not have a good efficiency, being that it visits the same node multiple times [42]. This leads to multiple modified versions of the A* algorithm presented in 2014 by Duchoň F. *et al.* [43] as well as by Costa [42] in 2011. Campos [44] also presents a modified heuristic for the A* in 2014, applying it towards the trajectory planning of autonomous robots. One can then conclude that the A* algorithm is rather flexible, allowing for modifications by adjusting its heuristic function, some other aspect that can be constraints to its efficiency are obviously hardware limitations such as a low computational speed or even limited memory.

Timed-Elastic-Band

Take the Elastic-Band approach as a way to deform a generated path, treating it as a rubber band that does not break, this band is then subjected to internal and external forces that shape up the path, keeping a distance from the obstacles and taking into consideration the shortest path towards the target point. The main problem with this algorithm is that it does not take into consideration the dynamic constraints of the robot, and, the fact that the band is not breakable may lead to local minima solutions [45]. To solve this problem surges the Timed-Elastic-Band (TEB) as an augment to the original Elastic-Band, this is an online method, meaning that the path is is optimized during run time, that takes into consideration the *"...temporal aspects of the motion in terms of dynamic constraints such as limited robot velocities and accelerations"* [45]. The generated path is resultant of a multi-objective optimization problem, computing a trajectory that satisfies the maximum velocities and accelerations possible by the vehicle and minimizes the execution time [46]. Figure 2.8 is representative of an example of how the algorithm works bending the said rubber band around the obstacles and generating an optimal path.

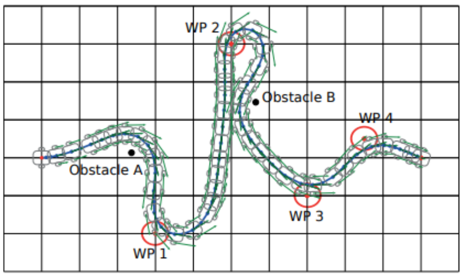


Figure 2.8: Example of a trajectory generated by the TEB algorithm [45].

A few years later in 2017, after proposing the TEB approach, C. Rösmann *et al.* [47, 48] also proposed an extension for the algorithm. This optimization takes care of the local minima issue, as the algorithm is now able to transit between different trajectories, "breaking" the rubber-band. This is done by optimizing a set of possible trajectories in parallel and switching between them in order to not get stuck in a local optimal solution, said is presented in figure 2.9 where multiple trajectories are computed and the algorithm chooses an optimal one.

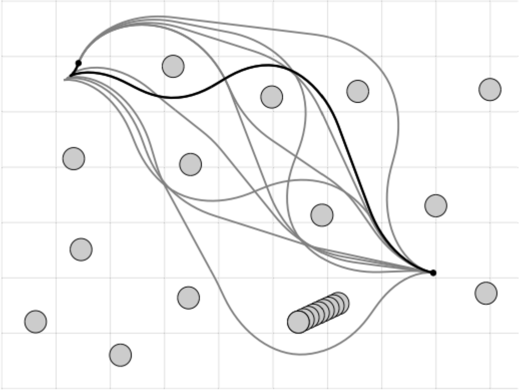


Figure 2.9: Example of the parallel trajectories computed by the TEB algorithm [48].

In its current state, the TEB algorithm presents itself as a great solution for the problematic as it computes in real-time a path between the given way-points and is able to adjust itself, bending around obstacles. The fact that it also respects the given kinematic constraints of the ASV is also a key factor.

Probabilistic Roadmap

A roadmap algorithm consists in nodes and the connections between them, these types of algorithms represent the robot's free space as a network of uni-dimensional curves [44]. The Probabilistic Roadmap (PRM) is constructed in two separated phases, the Learning Phase, where the actual map is built and the Query Phase where the the path is computed, from the initial node to the destination.

The Learning Phase can be sub-divided in three different stages, the first one is the construction of the map, starting with an empty one, a random configuration is loaded as a node, afterwards its neighbour nodes are iteratively selected, within a maximum distance, to each verifying the connection between nodes. This stage is done repeatedly until the graph is occupied with enough nodes. During the second stage, the map is expanded in the regions where there are rather few free spaces, this is done to improve the roadmap, since few spaces means very few nodes and connections [42]. The last stage is optional, and aims to simplify the roadmap by removing the redundant nodes. The latter phase, the Query Phase can also be parted in three stages. Firstly, a free path between the starting node and a node from the roadmap is created, then the same is done for the target point and finally a path that connects the starting point to the final one is computed [44]. The PRM algorithm is a proven one in multiple areas of application, such as path planning [49] for car-like robots and also Unmanned Aerial Vehicles (UAV) [50]. Good practical results can be achieved with this algorithm, however this method is not a complete one, as a probabilistic method, it may only converge and find a trajectory if the given time tends to infinite. If there is no possible trajectory, the algorithm may get stuck in a loop [42].

Potential Field

The potential field approach in trajectory planning is based on the physics of electric potentials as an heuristic to find a trajectory. The ASV is represented by a point in a space where there are obstacles, this very point is considered a positive charge in the field. For the artificial electric camp to be repulsive with the obstacles, these also behave as positive charges. The goal target, will obviously have a negative charge creating a attractive force towards the ASV. The artificial potential field can then be defined as the sum of all these electric camps, and the forces created by the attractions and repulsions will, in a ideal scenario, generate the path between the vehicle and the goal, avoiding obstacles in between. Figure 2.10 is representative of a path generated by the potential field method, as discussed, the obstacle creates a repulsive force driving the path to avoiding it, the closer the robot is to point, the stronger the repulsive force is, decreasing its influence on the robot as the distance to it increases. The path directs itself towards the goal point, the force generated by this point is greater with distance, being null when the robot reaches this point.

Both the mathematical model representative of the $U(q)$ and the $F(q)$ can be found in [42].

One of the most discussed issues with this algorithm is the fact that it is prone to being stuck in local minimum, [42, 44]. Figure 2.11 is representative of said situation, depending on the obstacle's shape (if concave) and size, this is situation is more susceptible to happen. Some other

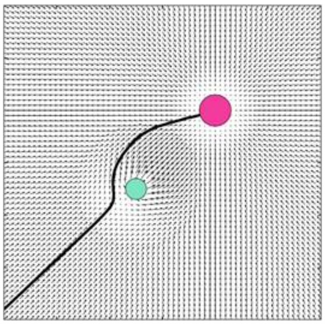


Figure 2.10: Example of a path generated by the Potential Field algorithm [51].

issues have been documented by Koren Y. *et al.* [52]. These include the one already presented, and also the fact that the algorithm can't compute a solution that passes through two close obstacles. Some fluctuations are also to be noted in the presence of obstacles that may lead to an unstable solution.

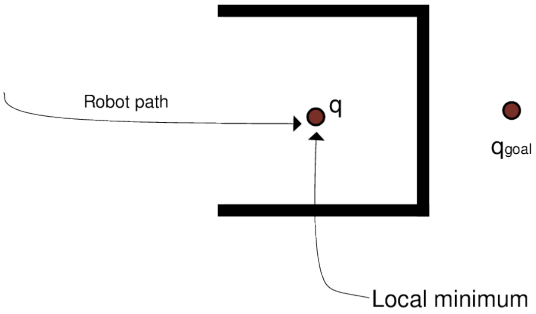


Figure 2.11: Local minimum in the Potential Field algorithm [53].

However, for the current problematic the local minima may not be an issue since the objective is to dock the ASV inside an object that is very similar to the ones that cause this very problem. For this reason the Potential Field algorithm is considered a valid option for implementation.

Velocity Obstacles

This algorithm has been quite utilized and expanded upon during the years, even being implemented in actual real environments by Kufoalor *et al.* [54] in 2018. The first step is to define the obstacles and an associated collision cone for each one, taking into consideration their relative velocity towards the ASV. Once the obstacles are represented, a set of admissible velocities is defined taking into considerations the vessel's dynamic constraints, within a preset temporal space. Figure 2.12 is representative of the described representation of the collision cones and the admissible set. The following step is then to use an heuristic to obtain the velocity that best suits the application and that belongs to the admissible set.

In 1993, Fiorini *et al.*[55] proposes some heuristics that can be applied to obtain a suitable velocity, among them:

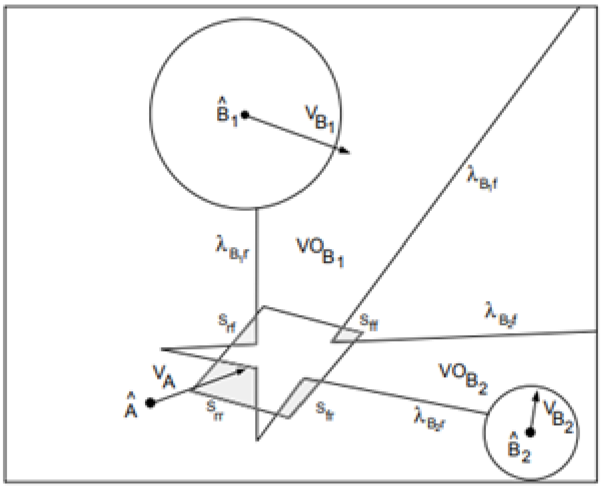


Figure 2.12: Representation of the collision cones and the set of admissible velocities [55].

- Choose the highest velocity that stays out of the collision zone and maintains alignment with the goal;
- Choose the highest velocity, outside the collision zone, with an associated maximum angle relative to the goal line;
- Choose the highest velocity that avoids obstacles according to their associated risk.

As the algorithm is rather flexible, allowing for any heuristic that satisfies the purpose, different applications are also possible such as the work presented in 2008 by van der Berg *et al.*[56] scaling the concept to multi-agent navigation.

2.4 Critical Analysis

The current section aims to bring some closure to the State of the Art chapter, presenting the main withheld thoughts from the conducted research. As perception systems rely heavily on aspects such as fidelity, accuracy and robustness, the use of a single sensor might seems like a great risk as the lack of redundancy can leave the ASV lost and unaware of its surroundings. However, solo LIDAR based systems do exist and prove themselves reliable enough to be used in a maritime environment. The data retrieved by this sensor can then be processed in 2D, which is more efficient and a good solution for low computational power systems but completely dependent on the sensors' resolution, or 3D which is the more usual solution and the one that provides better results even if it requires heavier computational resources and the effort to model the platform's geometry since the registration process requires a target template. The vast majority of perception systems use cameras to further improve the robustness of the algorithm, as both sensors compensate each other's weaknesses. From the conducted research this is the most viable solution, therefore the Situational Awareness System developed as integrating part of the Self-Guided Docking Architecture will be based on these two sensors, combining information from both in order to recognize the platform and detect its features. A third possible approach is the usage of satellite images and knowledge based models, however such technology is not available hence these approaches are not considered viable for the present application.

As planning algorithms go, the ones most fit for the docking operation are those that take into consideration time instants and the ASV's dynamic constraints, as well as the obvious geometrical constraints of the trajectory. From the studied approaches, the Velocity Obstacles concept is the one that stands out the most as it is very flexible and tested in multiple applications, being easily adaptable to the maritime environment by the means of any heuristic. From the remaining studied algorithms, the Potential Field could be considered useful as its main downside, to get stuck in a local minima, could actually be used as a positive for the docking operation. On the other hand, the PRM algorithm proves to be unreliable for this application since it might get stuck in an infinite loop, leaving the ASV without a planned trajectory to follow.

Chapter 3

A Self-Guided Docking Architecture

3.1 Problem Characterization

To secure the previously discussed regulatory approval, as well as industry support and public acceptance, ASVs need to be at least as safe as the existing vessels. One of the operations that has the most associated risks is certainly the docking one. A failure in this process can lead to monetary losses, by damaging the vessel or the physical infrastructure of the port, fatalities or even environmental disasters such as water pollution or soil contamination [57]. Making this operation autonomous would translate into a significant contribution for an autonomous ship safety, greatly reduce these risks.

Sea-state constraints, such as tidal waves and wind, are prone to hinder the docking operation and, therefore, the development of an architecture that is viable and applicable in a real environment must take these conditions into consideration. Same can be said for the vessel's inertia and kinematics. To accurately simulate these values, a model of the Zarco[1] ASV, developed at INESC TEC [58], is used within a Gazebo simulated scenario, which makes it possible to configure relevant environmental parameters like the ones presented above. Figure 3.1 presents both the real ASV, on the left, and the simulated model, on the right. This vessel is 1.5 m by 1 m and has approximately 1.5 m of height, these dimentions are to be used for analyzing if the docking platform is a suitable one or if the vessel would not fit inside the berth. Since the vehicle weights 50 kg (up to 75 kg with payload), the associated inertia is considerable, for this matter, its predictive capability must be high to guarantee that the response time of the systems is low, avoiding possible collisions with the platform.

Kinematic constraints are also to be considered, the vessel utilizes a differential drive that consists of two electrical thrusters mounted on a common axis, working independently. Its nominal velocity is $1\,\mathrm{m\,s^{-1}}$ granting it an autonomy of 4 to 6 hours at this speed, however the vessel is capable of achieving $2\,\mathrm{m\,s^{-1}}$ at maximum velocity. The Zarco ASV is equipped with a 3D LIDAR, a pair of stereo cameras, an Inertial Measurement Unit (IMU) and a GPS. The developed Situational Awareness System will then rely upon these sensors to provide data about the surroundings

[1] https://oceansys.fe.up.pt/?section=tech

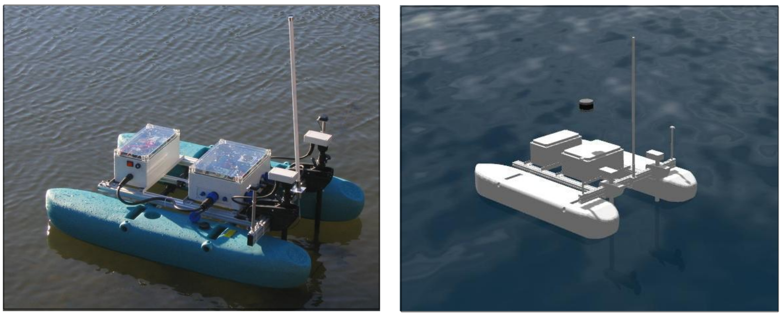


Figure 3.1: Zarco ASV, vessel used in simulation.

of the ASV, as well as pint-point its location relative to the dock. The spatial configuration of the described sensors is illustrated in figure 3.2 and the most relevant specifications from each of the sensors are as follows [59]:

- **IMU** - *MTi-30 Xsens* - Frame Rate: 200 Hz, Angular Accuracy: 0.2°/0.5°;
- **GPS** - *Swift Navigation* - Frame Rate: 20 Hz, L1/L2 RTK, Accuracy: 0.01 m horizontal and 0.015 m vertical;
- **Stereo Cameras** - *Mako G-125* - Frame Rate: 30 Hz, Resolution: 1292 x 964, Field of View: 80° horizontal;
- **LIDAR** - *VLP-16* - Frame Rate: 4 Hz, Range: 100 m, Range Accuracy: 0.03 m, Field of View: 360° horizontal and 30° vertical.

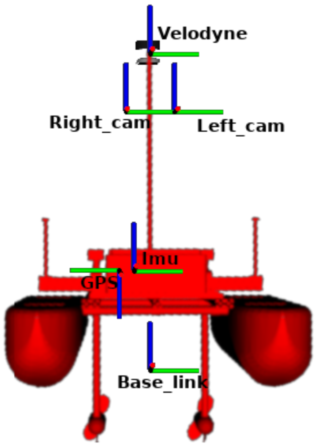


Figure 3.2: Sensor configuration used for the Zarco ASV.

The docking platform used as subject for testing is the one provided by the RobotX Challenge[2], a competition that supports innovation for autonomous surface vehicles. Figure 3.3 presents the Gazebo simulated version of said platform, as it is noticeable from the geometry of the dock itself,

[2] https://bitbucket.org/osrf/vmrc

the operation must be conducted frontwards. For this reason, the developed architecture is aimed at this kind of platform and type of approach.

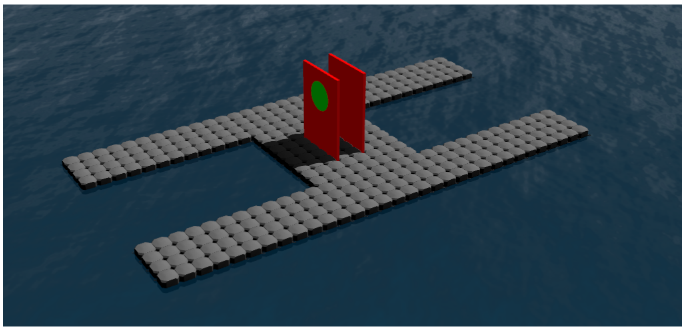


Figure 3.3: Docking platform model used in simulation.

3.2 Supervisory Control System

To accomplish a fully autonomous docking operation, the proposed architecture must be able to conduct this procedure in diverse scenarios, as to overcome the challenges induced by the harsh weather and environmental conditions. The diagram presented in figure 3.4 represents an overview of the developed architecture, depicting its main integrating modules.

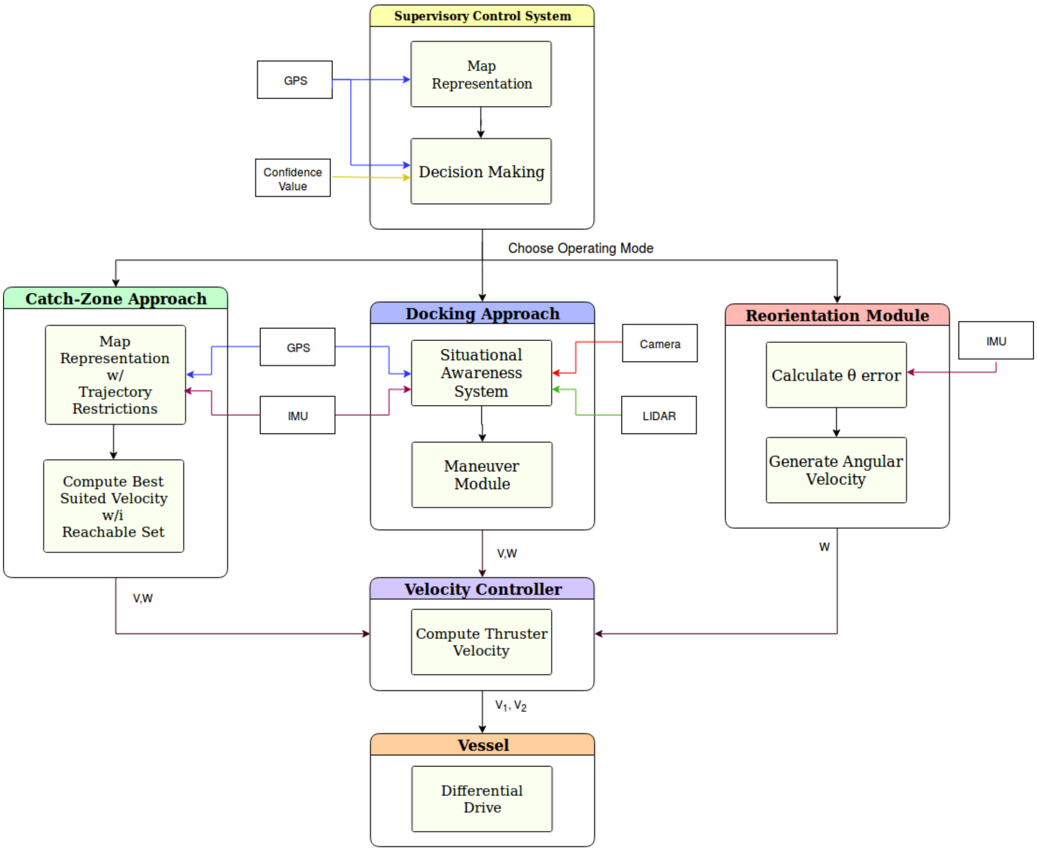


Figure 3.4: General overview of the self-guided docking architecture.

The docking operation should only begin when the necessary conditions (position and orientation of the ASV) are met, which contribute to a safe maneuver. Two main modules are then designed to guarantee these conditions:

- the **Catch-Zone Approach** - responsible for navigating the ASV towards a predefined area where the vessel can make use of its suite of on-board sensors to identify and recognize the dock - the catch-zone;
- the **Reorientation module** - that guarantees that the vessel is orientated towards the dock, improving visibility and consequently the quality of the retrieved data.

Once all the conditions are full-filled the **Docking Approach** may be initiated, this module is the most important one as it conducts the approximation towards the platform. To avoid damages to both the vessel and the physical infrastructure, the vessel needs a robust perception system that makes it aware of the surrounding scenario, by identifying the docking platform and even possible hazards. When in close proximity to the mooring facilities, the ship follows a given path that leads to a safe docking maneuver. To supervise these modules and actively choose which one should be enabled at each moment, a **Supervisory Control System** has been designed, operating within a closed loop. Each main module ends its iteration by generating linear and angular velocities. As specified in section 3.1 of the present chapter, the vessel used for testing has a differential drive, as so, the speed affected to each thruster (V_1,V_2) can be determined from equations 3.1 and 3.2 [60].

$$V_1 = \alpha \cdot v + \beta \cdot w \tag{3.1}$$

$$V_2 = \alpha \cdot v - \beta \cdot w \tag{3.2}$$

where α and β are constants representative of thruster placement and configuration, v is the linear velocity and the angular velocity (w) is considered to be proportional to the deviation from the docking platform (w $\propto$ θ).

By observing the interactions between modules provided by figure 3.4, the importance of the existing Supervisory Control System is noticeable as it manages the whole operation by deciding which system should be active at each moment. To do so, the controller must first obtain a rough map of the scenery where it will then represent the catch-zone, figure 3.5 serves as example of this representation. The first step to obtain this catch-zone is projecting the centroid point that represents the dock (D) forward (along the positive X axis), as a safety measure, gaining some more room to operate without the danger of collision. This projection distance is user-defined, and by applying it to D, the point D' can be obtained. From the latter point, the first vertex of the zone can be obtained by adding the working distance of the LIDAR sensor, this value represents the maximum distance from where the sensor can operate. This vertex is marked as V_{Top} in figure 3.5. The camera's FOV is used as the angle from which the remaining coordinates of the vertexes are calculated from. By defining the region based on these parameters, one is able to ensure that within this region and these orientation values, the ASV can obtain valid data that can be used for the docking operation. To obtain said vertexes, V_{Left} and V_{Right}, equations 3.3, 3.4 and 3.5 are used.

$$V_{Left^D_y} = D'_y - W_{Dist} \cdot \tan(\frac{FOV_{Cam}}{2}) \tag{3.3}$$

$$V_{Right^D_y} = D'_y + W_{Dist} \cdot \tan(\frac{FOV_{Cam}}{2}) \tag{3.4}$$

$$V_{Left^D_x} = V_{Right^D_x} = D'_x \tag{3.5}$$

where V_{Left} and V_{Right} represent two of the three vertexes of the catch-zone, W_{Dist} is the maximum range from the LIDAR sensor, FOV_{Cam} is the maximum angle where the camera can retrieve information and D' is the projection of the dock's centroid point.

These vertexes are obtained in reference to the Dock (D), they must now be transformed to World (W) coordinates. Such is done by applying to said vertexes a homogeneous matrix ([R(θ),T]) that represents the rotation ($R_{3x3}(\theta)$) and translation (T_{3x1}) between both sets of coordinate axis. Equations 3.6 and 3.7 are used in said transformation.

$$V_{Left^W} = V_{Left^D} \cdot [R_{3x3}(\theta)|T_{3x1}]^W_D \tag{3.6}$$

$$V_{Right^W} = V_{Right^D} \cdot [R_{3x3}(\theta)|T_{3x1}]^W_D \tag{3.7}$$

where V_{Left} and V_{Right} represent two of the three vertexes of the catch-zone referenced in two different coordinate spaces, the Dock's (D) and the World's (W), both the rotation and translation between both sets of coordinate axis are represented by $R_{3x3}(\theta)$ and translation (T_{3x1}), respectively.

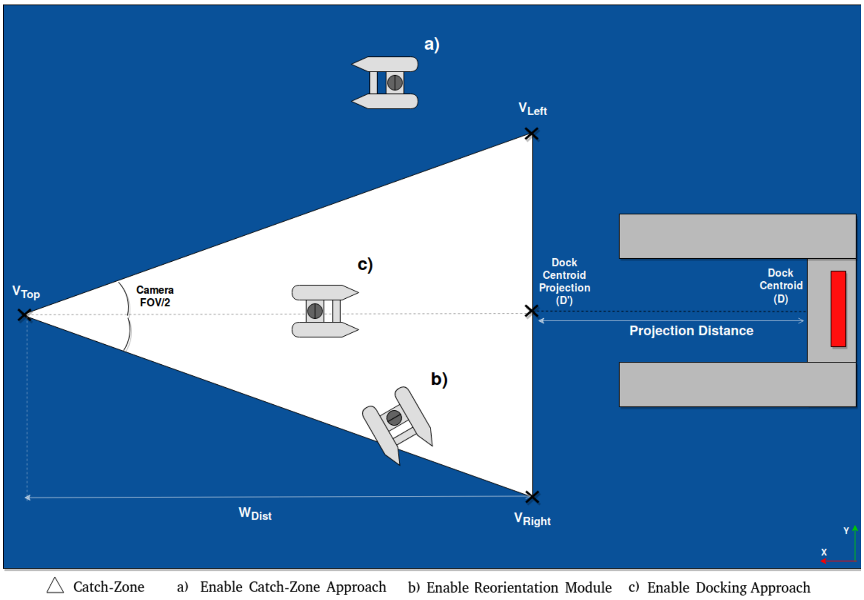

a)
V_{Left}
c)
V_{Top}
Camera FOV/2
Dock Centroid Projection (D')
Dock Centroid (D)
Projection Distance
b)
W_{Dist}
V_{Right}
Y
X
△ Catch-Zone
a) Enable Catch-Zone Approach
b) Enable Reorientation Module
c) Enable Docking Approach

Figure 3.5: Example of the map representation within the Supervisory Control System.

Once the zone has been defined, the Supervisory Control System's decision making process resorts to the state machine presented in figure 3.6 to choose which operating mode should be active. If the vessel is outside the Catch-Zone, the controller must then enable the Catch-Zone Approach module, conducting navigation towards this area. This condition is depicted in situation a) of figure 3.5 where the ASV is clearly outside the white region outlined by the triangle.

To calculate this area, the docking platform's position and orientation are previously known, with a certain uncertainty. So that one is able to determine if the ASV is in fact within the established region, a radius check is performed at the ASV's current location. Such approach is taken in order to compensate the error associated with the coordinates obtained from the GPS, therefore this radius is user-defined, taking into consideration both the dimensions of the vessel and this error associated to each measure.

However, being inside the Catch-Zone is not enough to guarantee a safe and accurate docking maneuver. Situation b) of figure 3.5 represents an example where this is the case. The ASV is inside the region, yet, its orientation towards the dock is completely incorrect, which means that the berth infrastructure is out of the FOV of the camera. The Supervisory Control System is then tasked with enabling the Reorientation Module until it corrects the relative orientation between robot and dock. Regardless of this adjustment, if a confidence value, associated to the data captured by the camera, does not meet the defined threshold (Confidence > 70 %), the ASV's orientation should keep being corrected until this threshold is achieved. This confidence is associated to the quality of the segmentation that can be obtained from the camera's image and will be expanded upon in later sections.

Finally, if all the conditions are met, situation c) in figure 3.5, the ASV being oriented, inside the zone and with a good confidence value from the camera, the Docking Approach can be initiated, ending the loop when the mission is carried out successfully.

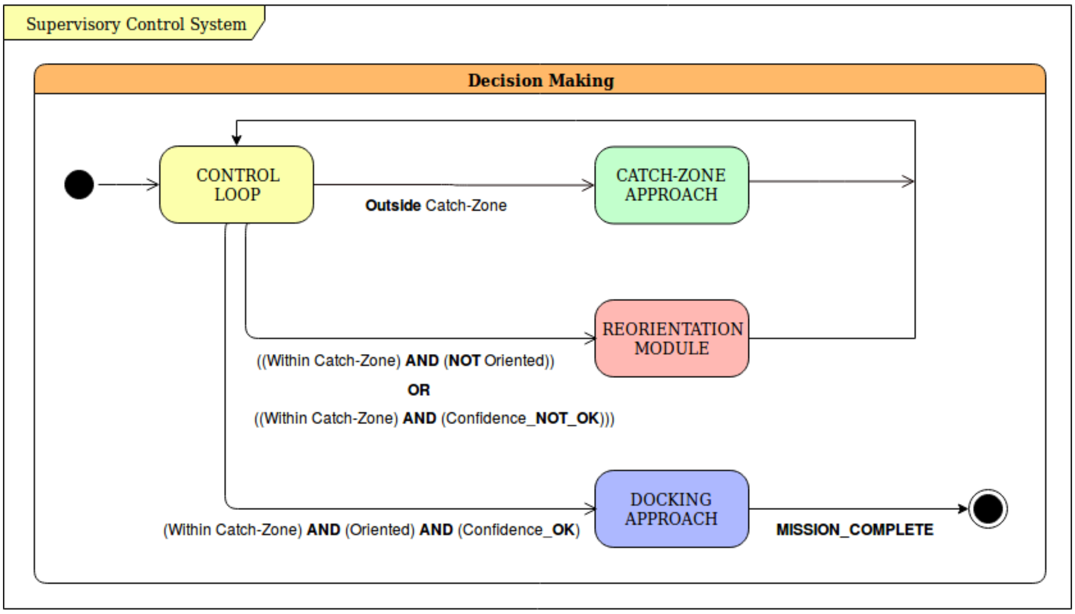

Supervisory Control System
Decision Making
CONTROL LOOP
Outside Catch-Zone
CATCH-ZONE APPROACH
REORIENTATION MODULE
((Within Catch-Zone) AND (NOT Oriented))
OR
((Within Catch-Zone) AND (Confidence_NOT_OK)))
DOCKING APPROACH
(Within Catch-Zone) AND (Oriented) AND (Confidence_OK)
MISSION_COMPLETE

Figure 3.6: State Machine implemented for the Supervisory Control System.

3.3 Catch-Zone Approach

The importance of starting the docking operation from a certain region where viable and rich data can be gathered has been previously alluded. When outside this defined catch-zone, the data retrieved from the sensors is unreliable and may lead to an insecure operation. The current section aims to expand on the topic by detailing how the Catch-Zone Approach module generates a trajectory and conducts the navigation by iteratively updating this trajectory and computing the necessary linear and angular velocities. The interactions within this module are expressed in figure 3.7, the catch-zone representation is received from the Supervisory Control System and combined with the data retrieved from the GPS and IMU sensors to map the scenario. This scene representation is redefined with the restrictions imposed by the trajectory that should be followed by the vehicle to reach the catch-zone.

Based on the presented kinematics of the vessel (section 3.1), the set of velocities that a vehicle can reach at a given instant can be visually represented by a polygon that is then projected on top of the previously obtained acceptable area. The interception of both areas generates a set of reachable velocities from which the best suited one is chosen, recurring to a heuristic that minimizes the error in norm and direction, relative to the target point.

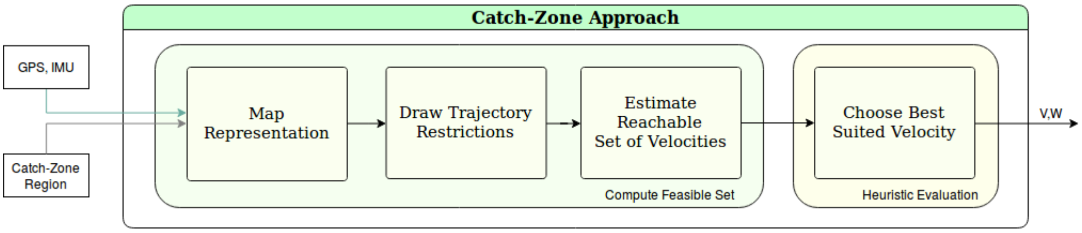
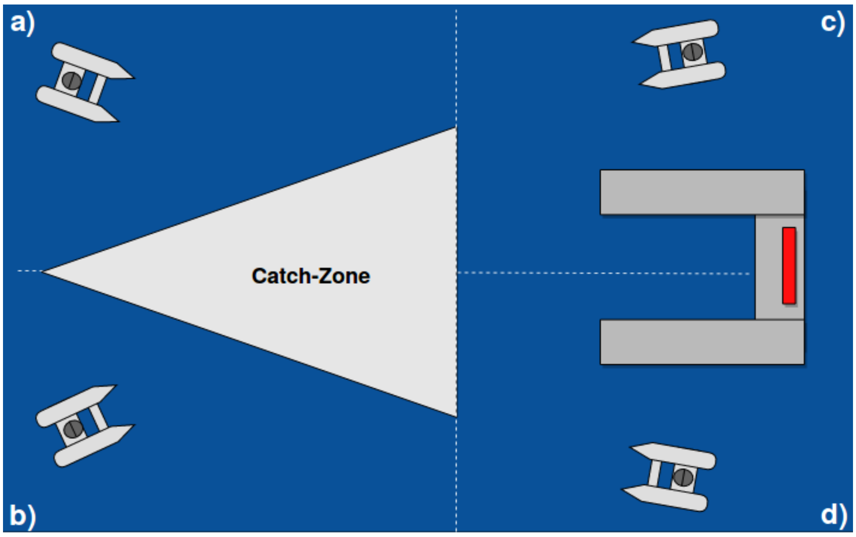


Figure 3.7: Diagram representative of the intrinsic interactions within the Catch-Zone Approach module.

This module should be capable of conducting the navigation of the ASV in different scenarios and multiple environmental conditions. Four main scenarios were considered while developing the Catch-Zone Approach module, a couple of scenarios where the ASV is in front of the dock's projection centroid (represented as D' in figure 3.5) and a couple more where the ASV starts behind said point. Each of these is then split into two approaches: from the left and from the right. Figure 3.8 offers visual context for the described scenarios. The estimation of the acceptable zone for the trajectory is based on the calculation of ellipses, each of the presented quadrants defines different start and end angles for the arc that creates a restriction for this zone.

The position and orientation of the ASV, retrieved from the GPS and IMU sensors, respectively, allows for an easy representation of the robot's pose within the scenario. By drawing a line between V_{Left^D} and V_{Right^D} one is able to define the first restriction for the acceptance zone, represented in figure 3.9. Projecting this line into the ASV's referential, points $V_{Left^{ASV}}$ and $V_{Right^{ASV}}$ can be obtained and consequently a new restriction when these are connected. This projection is achieved by rotating the first restriction around the D' point with the θ_{ASV}, which represents the ASV's orientation referenced to the world frame, then translating it to the vessel's current position.



a) Front and Left b) Front and Right c) Back and Left d) Back and Right

Figure 3.8: Four different scenarios for the Catch-Zone Approach.

The next step is to define the last two restrictions for the acceptance area, these are calculated using ellipses. This is the adopted method as it can be quite adaptive to trajectories that require a certain curvature, allowing the estimation of an area that accommodates such navigation. However, for straight lined trajectories an ellipse would simply cause an unnecessary curvature to the navigation, lowering the efficiency of the algorithm. For this reason, a simple heuristic has been developed to adjust the ratio between the minor and major axis of each ellipse, compensating the presented problem by narrowing down the area of the corresponding restriction.

To calculate the centers of each ellipse, a line is generated connecting the points from the ASV to the Dock, V_{Left^D} to $V_{Left^{ASV}}$ defining an upper bound restriction, and V_{Right^D} to $V_{Right^{ASV}}$ defining a lower bound restriction. The middle point of each line is taken as the center for the corresponding ellipse and the length of the segment from center to one of the end points defines the major axis radius of each one. Figure 3.9 presents a visual representation of the described ellipses. As discussed above, the minor axis for each ellipse is adjusted according to the trajectory's necessary curvature, to estimate its length the heuristic presented in equation 3.8 is used:

$$r_{min} = tanh(|m|) \cdot R_{MAJ} \tag{3.8}$$

where r_{min} represents the radius of the minor axis, R_{MAJ} is the radius of the major axis and m is the slope value correspondent to the major axis of each ellipse.

By using a normalized value of each slope and multiplying it by the major axis radius, one becomes able to adjust the curvature of the restriction. The more the slope becomes becomes horizontal, the lower is the value given to the minor axis of the ellipse, adjusting the restriction to allow a straight line trajectory. On the other hand, when a curved trajectory is needed, the slope will be almost if not completely vertical (infinite slope), after the heuristic the minor axis (r_{min}) will then be equal to the major one (R_{MAJ}), drawing out a circle that accommodates the needed curvature.

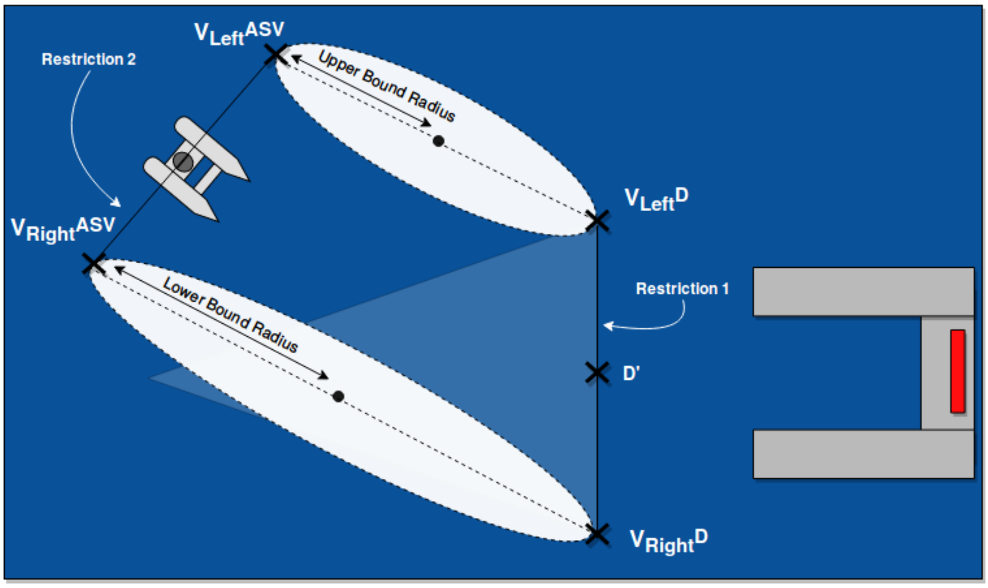


Figure 3.9: Representation of the map computed inside the Catch-Zone Approach module.

The last step to define the remaining restrictions is to choose which part of the ellipse's arc is actually part of the restriction. Such is done by defining a starting and ending angle from the ellipse that corresponds to the pretended arc, figure 3.10 provides context on the notation and a visual example of how the restriction is computed. These angles are predefined and change according to which quadrant from figure 3.8 the trajectory's starting point belongs to.

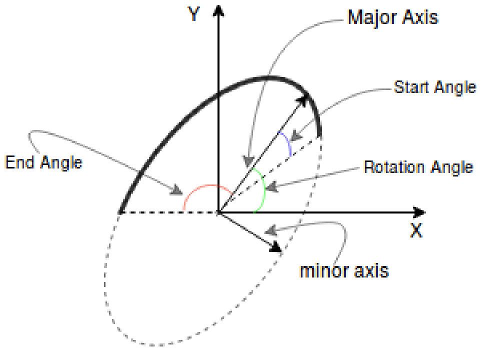


Figure 3.10: Visual representation of the notation used to calculate each ellipse restriction.

Once a rough estimation of the acceptable zone is generated, its area is filled and the resulting contours are computed. Depending on the trajectory, multiple contours may obtained. To sort out which is the the one corresponding to the actual acceptance zone, the one with the biggest associated area is kept. Figure 3.11 depicts the process step-by-step for a trajectory such as the one represented by d) in figure 3.8:

- Step 1 corresponds to the mapping of the scenario where both the ASV's position and orientation are represented as well as the dock projection. As the trajectory demands a big curvature, the ellipses are adjusted though the heuristic into becoming circle like;
- Step 2 depicts how the starting and ending angles define the arc of the ellipse that is actually the wanted restriction;
- Step 3 filters the contours;
- Step 4, the region is filled and taken as the acceptance zone.

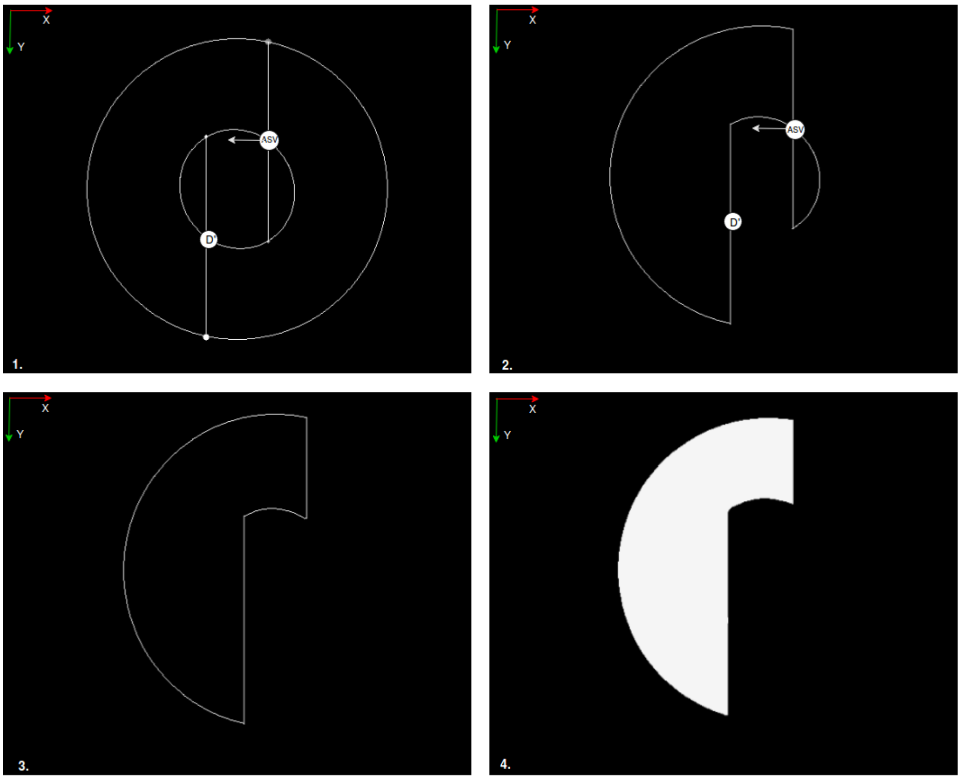


Figure 3.11: Step-by-step demonstration of how the acceptance zone is defined.

Now that the acceptance zone for the trajectory is defined, the set of velocities that belong to this area and also respect the robot's kinematics must be estimated. This set is calculated by representing within a polygon all the velocities that the ASV can achieve. As discussed in section 3.1, the Zarco ASV is a differential vessel and as such it cannot move in the direction along the axis - this is a singularity. This constraint and the maximum linear and angular velocities it can achieve are used to compute the polygon mentioned above, figure 3.12 a) depicts this set of velocities that can be met by the vessel.

By performing a *bitwise AND* operation between the resulting set and the acceptable zone, figure 3.12 c). The points that do remain from that operation are eligible velocities that respect the geometric constraints of the trajectory and also the kinematic ones, illustrated in figure 3.12 d).

The most suitable velocity from the established set of reachable velocities is chosen via the following heuristic that has been adapted from the HyCVEL algorithm [10]. As the Catch-Zone

Approach is based on the Velocity Obstacles concept, which allows for any heuristic that satisfies a given purpose, the adaptation is rather straight forward. Its practical result when applied to the set presented in figure 3.12 d) is illustrated in figure 3.13, where the chosen velocity is represented by the gray circle.

- Select the velocity that minimizes the error in norm and direction, relative to the target point D', belonging to the set of reachable velocities.

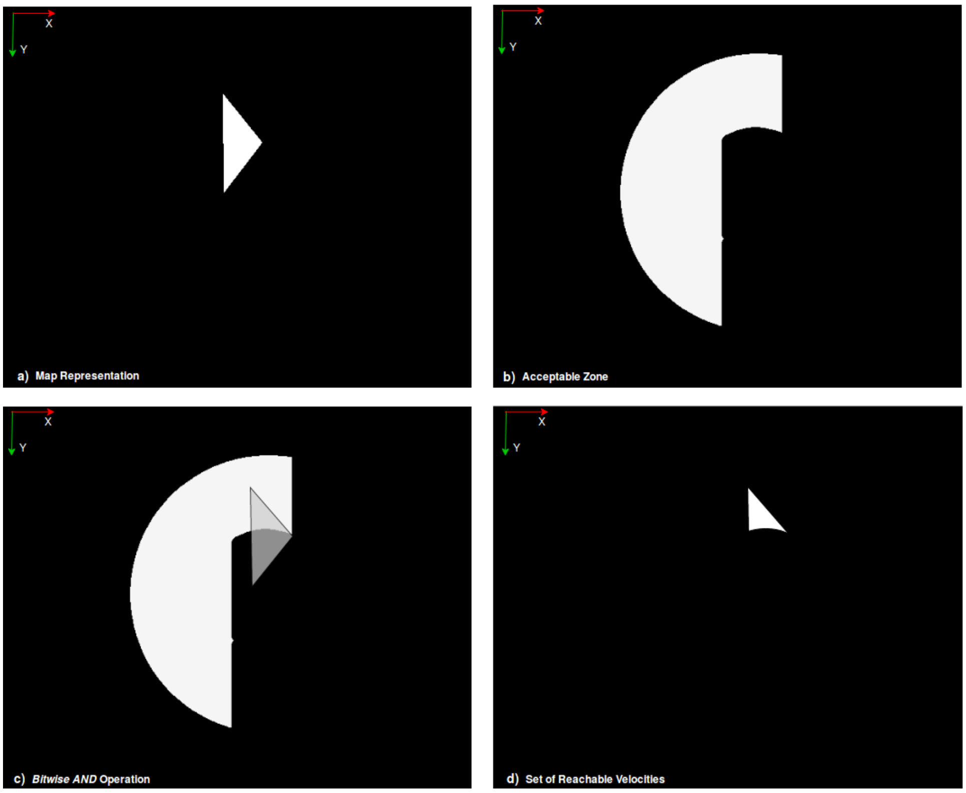


Figure 3.12: Step-by-step estimation of the reachable set of velocities.

This point must now be converted into its corresponding linear and angular velocities, such is done by taking into account the distance and orientation errors between the chosen point and the actual pose of the robot. The velocities' references are estimated with equations 3.9, 3.10 and 3.11:

$$\Delta Dist_{ASV}^{Point} = \sqrt{(X_{Point} - X_{ASV})^2 + (Y_{Point} - Y_{ASV})^2} \tag{3.9}$$

$$V = tanh(\Delta Dist_{ASV}^{Point}) \cdot K_1 \tag{3.10}$$

$$W = atan(\Delta Dist_{ASV}^{Point}) - \theta_{ASV} \tag{3.11}$$

where X_{Point} and Y_{Point} represent the coordinates of the point that represents the suitable velocity, X_{ASV}, Y_{ASV} and θ_{ASV} the position and orientation of the robot, respectively, and K_1 is a given gain to adjust the speed value.

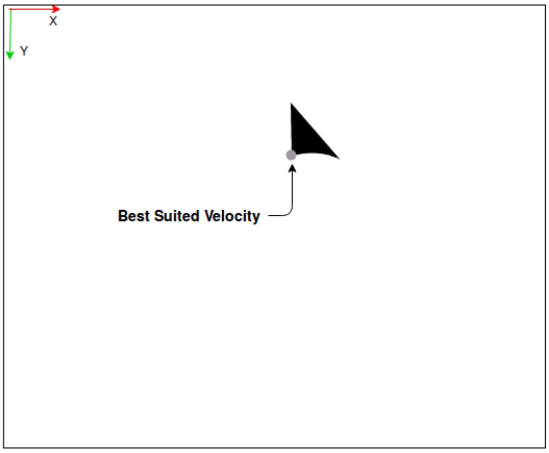


Figure 3.13: Result of the heuristic [10] application to the set of reachable velocities.

3.4 Docking Approach

The Docking Approach is obviously the central module of the architecture as it is responsible for not only recognizing the platform but also conducting safe navigation within close proximity to it. Its importance is even more evident as the remaining nodes are designed to support it, ensuring that all the conditions are met so that it can safely operate.

This system can be divided into two stages, as depicted in figure 3.14: a Situational Awareness (SA) System and a Maneuver module. The first stage combines the data retrieved from a 3D LIDAR, a camera, an IMU and a GPS to map the vessel's surroundings and recognize the dock. As this stage targets the identification of expected geometrical shapes, there is no need to modify the dock or surrounding waters by adding beacons or transponders.

The maneuver module is the complementary part of the SA system, taking into consideration the reconstruction of the scenario obtained by said Situational Awareness system, the route from origin to destination and other navigational aspects (e.g. specific actions to perform in each way point) to generate a suitable path. A key part of this operation is ensuring that the planned trajectory is a safe one, through a viability check procedure.

Information from the sensors is gathered and used by the SA module, which is responsible for making use of the information gathered from sensors, to estimate a target point for the docking operation. Once computed, the target is validated inside the maneuver module, generating linear and angular velocities that will then be converted into specific velocities (V_1,V_2) for each thruster and fed to the vessel's differential drive.

From the Camera perception subsystem a confidence value is computed, this value has been previously discussed as it is a fundamental part of the decision making process within the Supervisory Control System, section 3.2. This confidence represents the fidelity associated to the reconstruction of the scenario that can be computed from the gathered data at a given instant.

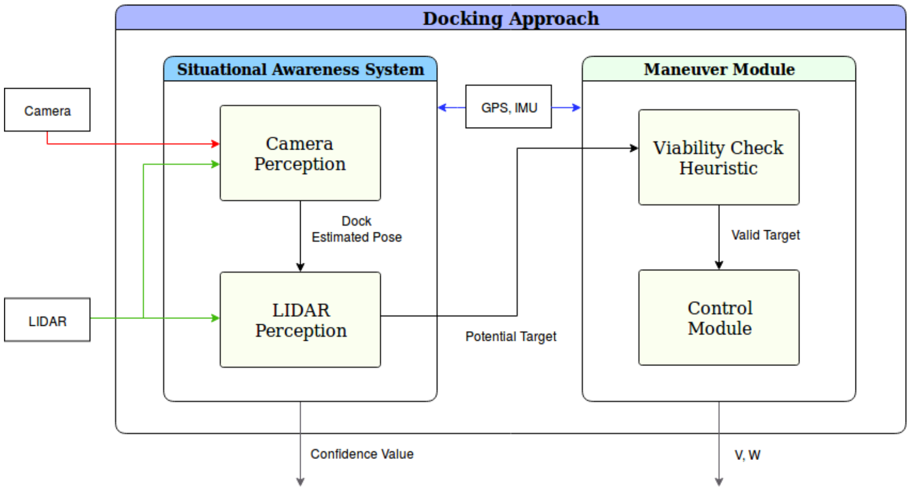


Figure 3.14: Docking Approach internal architecture.

3.4.1 Situational Awareness System

For the docking operation to be successful, making the vessel aware of its surroundings is a vital part of the process. This fact makes the SA system the most important part of the architecture presented in 3.14. A division into two subsystems can be noticed in said figure, a Camera and a LIDAR perception subsystem. Both to be detailed during the present section.

Using a camera together with a LIDAR is a very common option as per the research conducted during the State of the Art chapter. Both sensors compensate each other's weaknesses and by fusing data from both, one can obtain a more robust SA system. By taking a look at the interactions between both perception systems, figure 3.14, the information gathered from the camera is combined with the data retrieved from the LIDAR so that one may estimate the position of the dock relative to the ASV. This pose is then used for both the pre-processing nodes, and as a seed for the registration process that integrates the LIDAR perception system, to compute a potential target point that serves as the final destination for the operation.

Camera Perception

As the docking platform has a known geometry, figure 3.3, the Camera Perception subsystem takes the image obtained by the camera and segments it to extract its upper structure, since it is the most reliable part of said platform, being observable from a greater distance. Taking advantage of the structure's color, which is red and easily depicted from the remaining maritime background, image segmentation is conducted by color and then the geometry is computed to obtain a general region of interest that resembles the upper structure. This information is then used to reconstruct the structure and compute a bounding box around it, then by comparing its area with the one obtained from the actual extracted geometry, a confidence value is assessed.

At the same time, the 3D point cloud gathered from the LIDAR is projected onto a 2D plane where the region of interest is then applied, filtering the cloud that corresponds to the dock's upper structure. From this information one is able to compute its centroid and project it relative to the ASV, estimating a pose for the docking platform. Figure 3.15 presents a summary of the interactions between nodes inside the subsystem.

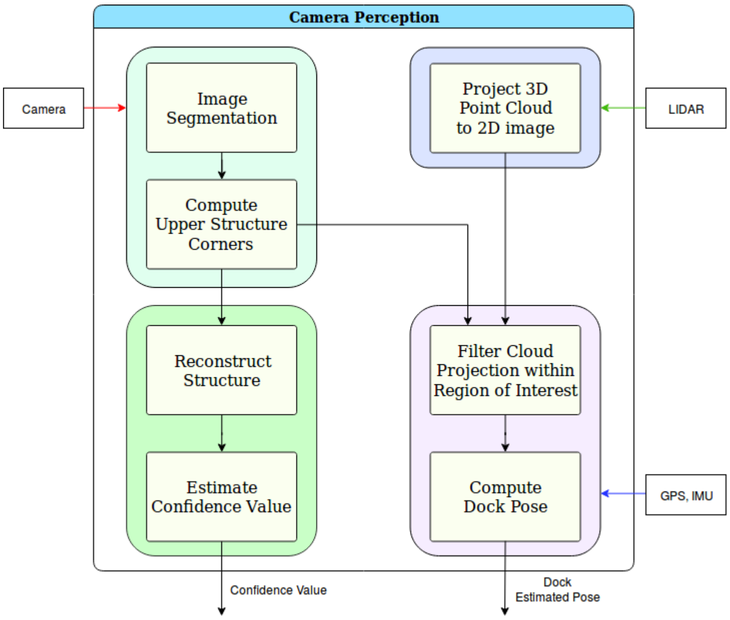

Camera Perception
Camera
Image Segmentation
Compute Upper Structure Corners
Project 3D Point Cloud to 2D image
LIDAR
Reconstruct Structure
Estimate Confidence Value
Filter Cloud Projection within Region of Interest
Compute Dock Pose
GPS, IMU
Confidence Value
Dock Estimated Pose

Figure 3.15: Diagram representative of the camera perception subsystem.

The first step of any perception system is pre-processing the gathered information, filtering out unwanted data. The raw image obtained from the camera needs to be segmented so that the geometry of the dock can be extracted. However, segmenting the platform from the background isn't the only challenge. As mentioned above, the upper structure of the dock has the red color, so, a simple approach would be to split the raw RGB image into its three channels and keep the information relative to the red one. Such approach is indeed effective if the light conditions are controlled, as that is not the case within a maritime environment, this process becomes rather unreliable since the intensities relative to each pixel vary with different lighting, therefore the RGB color space *"... is not preferred for color based detection and color analysis because of mixing of color (chrominance) and intensity (luminance) information and its non uniform characteristics."*[61].

A comparison between indoor and outdoor light conditions for the RGB color space is presented in figure 3.16, as the raw image is composed by the addition of three different channels which are affected with the same amount of lighting, the resulting colors are very sensitive to the scenario's light conditions. The blue channel is clearly the most influenced by the variation on the lighting, although visible changes are also present in the red and green channels.

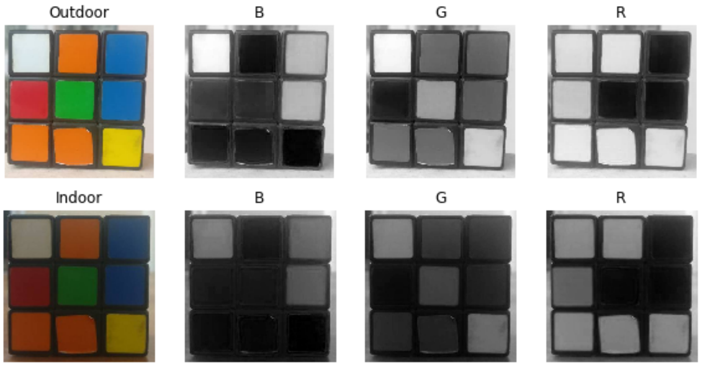


Figure 3.16: Effects of different light conditions on the RGB color space [3].

To contradict this dependence on the light conditions, a different color space was experimented on - the YCbCr color space. As it is derived directly from the RGB color space, the conversion is rather direct. Its three components represent the Luminance component of the RGB image (Y), and how far the red and blue components are from said Luminance (Cr and Cb, respectively). The luminance and chrominance components are then represented in different channels, such allows one to mitigate the effects of the light variances.

A comparison between light conditions for the YCbCr color space is presented in figure 3.17. The shift in light conditions is quite noticeable in the Y channel as expected but not in the chrominance ones, which proves the advantage of using said color space instead of the basic RGB one.

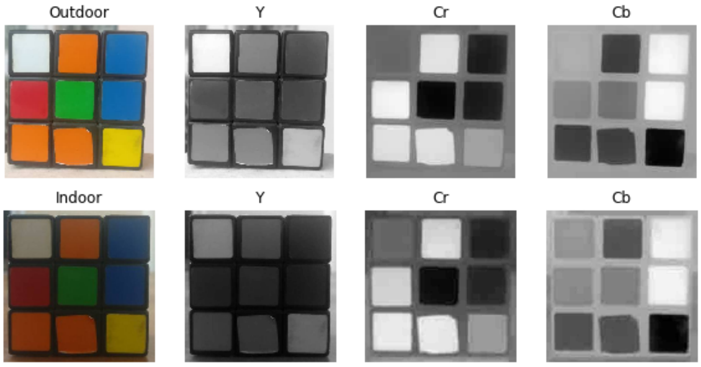


Figure 3.17: Effects of different light conditions on the YCbCr color space [3].

The equations used to transform the RGB color space to the YCbCr are presented in equations 3.12, 3.13 and 3.14 [62]:

$$Y = 0.299 \cdot R + 0.587 \cdot G + 0.114 \cdot B \tag{3.12}$$

$$Cb = (B - Y) \cdot 0.564 + \delta \tag{3.13}$$

[3] https://www.learnopencv.com/ color-spaces-in-opencv-cpp-python/

$$Cr = (R - Y) \cdot 0.713 + \delta \tag{3.14}$$

where R,G,B correspond to the red, blue and green components respectively, Y represents the Luminance and Cb, Cr the Chrominance in the new color space. Finally, δ varies with the number of bits present in the raw image, as it has 8 bits, δ = 128.

The step-by-step process for the color segmentation is presented in figure 3.18. Once the raw RGB image is converted to the YCbCr color space, figure 3.18 b), the three channels Y, Cr and Cb are split and the one correspondent to the Luminance is set to zero, the result of said operation is depicted in figure 3.18 c). This makes it so that the light conditions have reduced effects on later stages of the pre-processing. The image is then transformed back to RGB, by manipulating the equations presented above, figure 3.18 d), and the difference between it and the initial raw image is quite noticeable as the red color is now easily segmented from the remaining background.

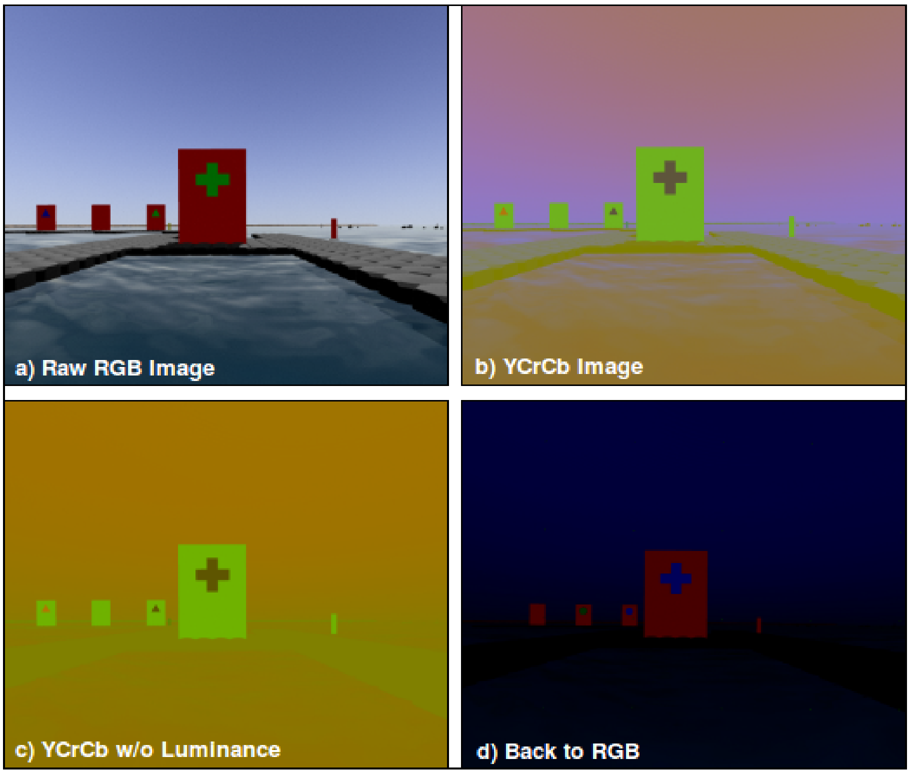


Figure 3.18: Step-by-step demonstration of the color segmentation process.

The final step for the color segmentation is now to separate the new RGB image into three separate channels and keep the information relative to the red component as it is the one that represents the upper structure of the docking platform. Figure 3.19 offers a visual comparison between the described operation applied to the raw image, before the color space transformation and afterwards. The difference between results is obvious, as the image resultant from the color space transformation is ready to be segmented, while the other one still has unwanted information that hinders such segmentation.

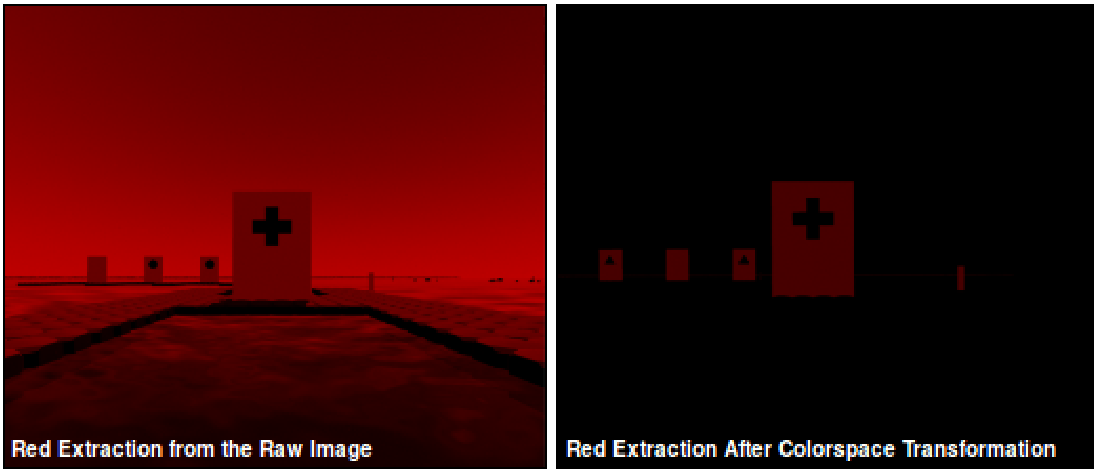

Red Extraction from the Raw Image
Red Extraction After Colorspace Transformation

Figure 3.19: Red component extraction for both the raw image and the processed one.

A median filter is then applied to further smooth out some existent noise such as the shoreline in the background. The image is now ready for the geometry segmentation. As color no longer serves a purpose said image is converted to grayscale to facilitate the upper structure geometry extraction.

Figure 3.20 depicts this geometry segmentation process, in step 1, the contours of the stucture present in the image are extracted, these are then approximated to a polygon and a bounding box is calculated for this polygon, describing step 2. As observed in the example presented by figure 3.20, multiple objects can be obtained from the geometric segmentation, the last step filters out the unwanted ones by assuming that the one with the biggest contour area is representative of the docking platform.

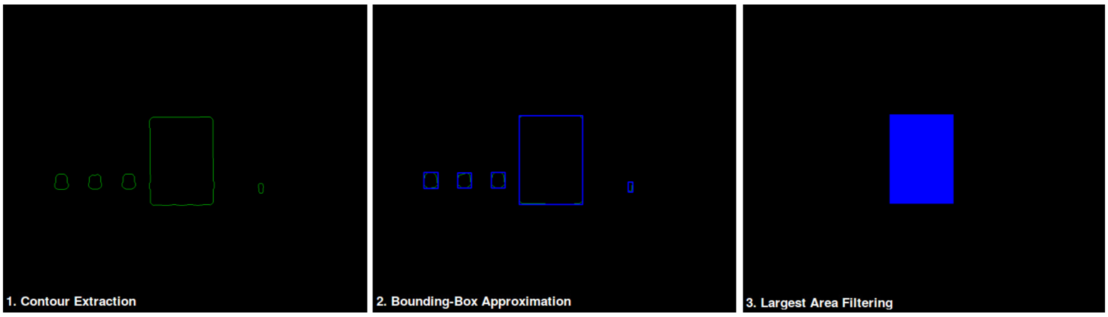


Figure 3.20: Step-by-step demonstration of the geometry segmentation process.

Once the geometry of the platform has been extracted one is able to compute from it a region of interest that represents the platform's upper structure, which will later be used to filter out the data gathered from the LIDAR, as well as reconstruct this structure within an image such as the one presented in step 3 of figure 3.20. By comparing how accurate this reconstruction is to the actual contour obtained from segmentation, the previously referred confidence value is calculated. Such value is obtained by subtracting both the contour area and the area obtained by approximating said contour to a polygon, equation 3.15 expresses how to estimate this confidence value, C(%):

$$C = 1 - \frac{A_{bbox} - A_{Contour}}{A_{bbox}} \cdot 100 \tag{3.15}$$

where C is the confidence value, A_{bbox} is the area of the reconstructed upper structure approximated to a bounding box and $A_{Contour}$ is the area resultant from the segmentation.

This confidence value is representative of the quality of the observed data from the camera. As expected this value decreases as the ASV gets further away from the dock. Another possible scenario that hinders this confidence is if the segmentation is affected by overlapping objects. The latter is exemplified by figure 3.21 where another dock in the background ends up affecting the geometry segmentation process, as the structures overlap only one contour is detected, enclosing both objects, therefore the polygon approximation results in a reconstruction that doesn't fully represent the real structure. The median filter previously implemented has been adjusted to smooth out these types of situations so that one may obtain more viable results.

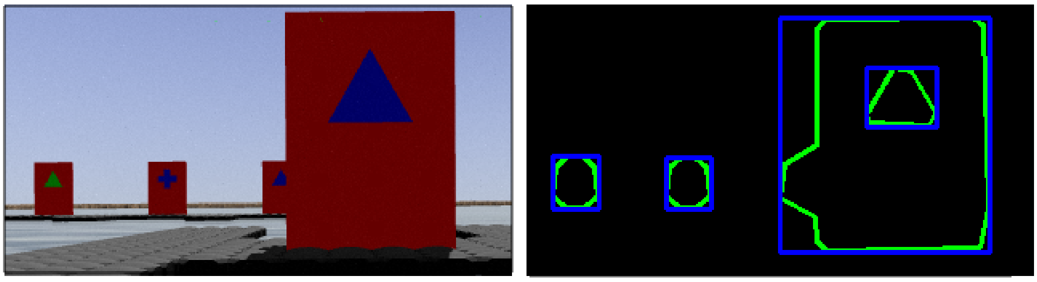


Figure 3.21: Example of a poor confidence situation.

The last step before the dock's pose relative to the ASV can be estimated is to project the 3D point cloud gathered from the LIDAR into a 2D plane and then filter from it the data that corresponds to the upper structure of the platform that has been segmented and reconstructed above in figure 3.20. This is done by approximating the camera by a projective model - the pinhole camera model, which expresses how a 3D point $P^{3D} = (X,Y,Z)$ is projected over an image plane where said point is discretized into a pixel $P^{2D} = (u,v)$. This model is represented in figure 3.22, before projecting the cloud's points, these must first be transformed into the camera's frame by using the extrinsic parameters matrix that represents the translation and rotation between this set of axis and the World frame.

Point C, visible in figure 3.22, is representative of the center of the camera (the pinhole). An image plane is projected by a distance defined as focal length, f = (f_x,f_y), on the line that is perpendicular to said plane and passes trough the pinhole, known as principal axis (Z). The intersection between this image plane and the principal axis defines the principal point, $p = (x_0, y_0)$. These parameters are expressed in the intrinsic matrix, allowing the model to take into consideration optic components by considering distortions introduced by the lens in the image.

Both these matrices that are alluded to are obtained from the simulated version of the camera in use, which is already calibrated and considers no lens distortions.

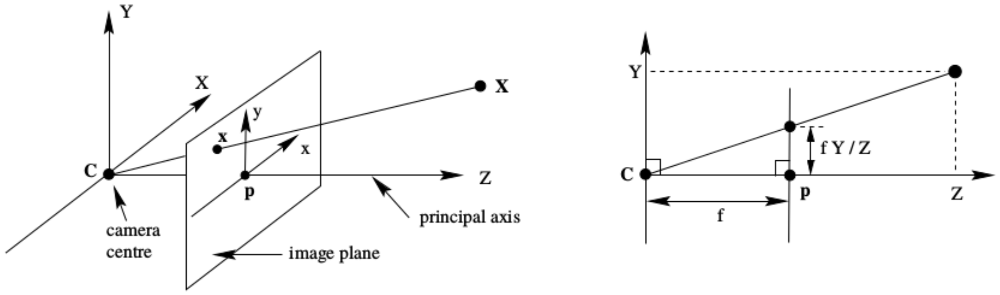


Figure 3.22: Pinhole camera model representation [63].

By resorting to the similar triangles created by the intersection of the principal axis with the ray that projects the point onto the image, such as the one represented in figure 3.22, one is able to compute the transformation from a $\mathbb{R}^3$ coordinate space to a $\mathbb{R}^2$ space using the equations presented in 3.16 and 3.17 [63]:

$$p_u^{2D} = \frac{f_x \cdot P_x^{3D}}{P_z^{3D}} + x_0 \tag{3.16}$$

$$p_v^{2D} = \frac{f_y \cdot P_y^{3D}}{P_z^{3D}} + y_0 \tag{3.17}$$

where p^{2D} is the projected point in pixels, P^{3D} is the point in X,Y,Z coordinates, f_x and f_y represent the focal length and finally x_0 and y_0 are the distances from origin to principal point.

These pixel values must now be filtered to ensure that these fit inside an image that has the same size as the ones taken by the camera, 800x800 pixels. Such is necessary so that one may use the reconstructed image obtained above to define a region of interest on top of this newly obtained image with the projected points. Before finally representing these pixels within a RGB image, their intensities are normalized according to the depth of their corresponding 3D points, using equation 3.18. This is done to correctly display the depth within this image by utilizing each pixel's intensity values, allowing for a better and more intuitive representation of the 3D scenario in the 2D plane.

$$I_{blue} = 255 \cdot \frac{P_z^{3D} - d_{min}}{D_{max} - d_{min}} \tag{3.18}$$

where I_{blue} is the value of the intensity given to the blue component of the RGB pixel, P^{3D} represents the point in X,Y,Z coordinates and D_{max}, d_{min} represent the maximum and minimum depth values of the points present in the cloud,respectively.

Figure 3.23 presents an example of application, the figure on the left shows the point cloud with the 3D points, and on the right one can see the said points projected onto the image and normalized, the brighter the intensity of the pixel, the further away it is from the vessel, on the contrary the darker it is, the closer. Points from the cloud that exceed the limits of the image are filtered out as mentioned above, such is the case for points that exist behind the ASV and outside the camera's FOV.

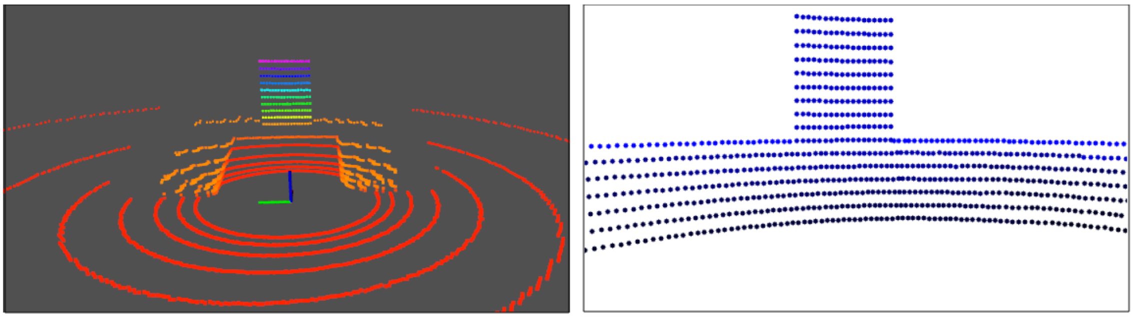


Figure 3.23: 3D point cloud projection onto a 2D plane.

The final step is now to use the reconstructed structure obtained in figure 3.20 3) as a region of interest, filtering the image resultant of the projection of all the points that exceed the limits defined by said ROI. The result is illustrated in figure 3.24.

The 3D coordinates corresponding to the remaining pixels are now used to estimate the centroid that represents the docking platform's position relative to the ASV, for this purpose, equations 3.19 and 3.20 are used.

$$P_{Dock_x} = \frac{\sum P_x^{3D}}{ROI_{Size}} - P_{ASV_x} \tag{3.19}$$

$$P_{Dock_y} = \frac{\sum P_y^{3D}}{ROI_{Size}} - P_{ASV_y} \tag{3.20}$$

where the P_{Dock} represents the estimated position of the dock relative to the ASV, P^{3D} represents the point in X,Y,Z coordinates, ROI_{Size} is the size of the region of interest to which said points must belong to, and finally P_{ASV} represents the current position of the ASV.

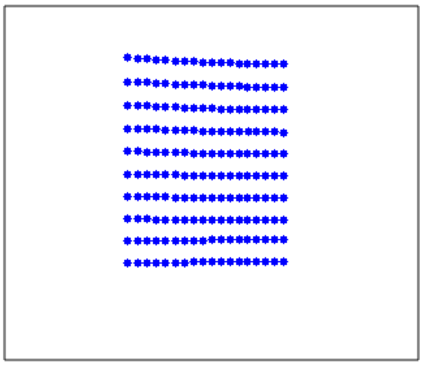


Figure 3.24: Projected image after ROI filtering.

LIDAR Perception

Now that the docking platform has been identified and its relative pose to the ASV is known, the target points that compose the mission can be calculated, figure 3.25 depicts this process. The data gathered by the LIDAR is highly affected by the conditions that the maritime scenarios impose, therefore it needs to be pre-processed before it can be used. Once filtered, a sliding window is implemented with the objective of providing a more complete and dense input point cloud to the registration method, by fusing multiple point clouds. Given input and target point clouds, said registration method estimates a transformation matrix that aligns the correspondences between both. As the ASV reaches the dock, the information retrieved by the sensors becomes more viable and complete, for this reason multiple templates are used as targets to match the increasingly more complex input data. The dock pose, estimated by the Camera Perception subsystem, is not only used for localization but also as an initial seed for the registration, so that one can obtain a faster convergence and also lower the computational demands of such process. Template points for the docking operation are defined *a priori* for each platform model, from these one is able to compute potential target points relative to the ASV by applying the inverse transformation matrix obtained from the registration method to said template points. These are later evaluated inside the Maneuver Module, which decides if the target is valid or not.

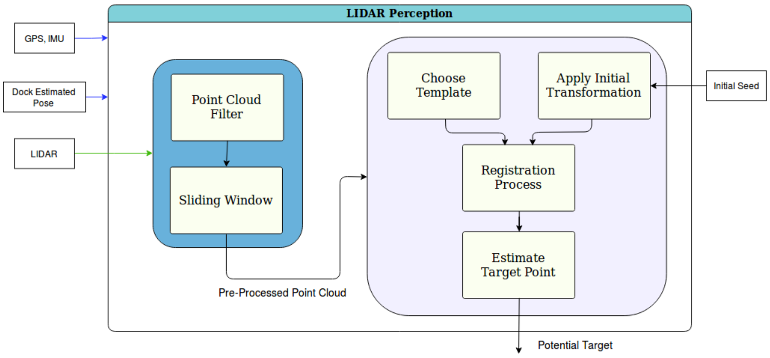


Figure 3.25: Diagram representative of the LIDAR perception subsystem.

The data obtained from the LIDAR is susceptible to the environmental conditions, the constant bobbing of the ASV caused by the tidal waves and wind can result in a very noisy point cloud, figure 3.26 on the left for instance. For the Situational Awareness System to work properly in a maritime scenario it must be able to interpret and analyze the reliability of the sensor information according to different environmental conditions at all moments. A simple but effective approach is to consider a ROI that can be adapted to the local conditions of the scenario. To mitigate the effects caused by the environment, this region is adjusted dynamically by using GPS and IMU data, resulting in a much cleaner point cloud such as the one presented in 3.26 on the right.

To achieve a point cloud filtered from the noise introduced by the environment, the LIDAR's FOV is adjusted according to the scenario, figure 3.27 depicts some example situations. The ideal scenario is presented in a), there are no waves and the platform can easily be detected by the default configuration of the sensor. Situations b) and c) present the best possible case where the

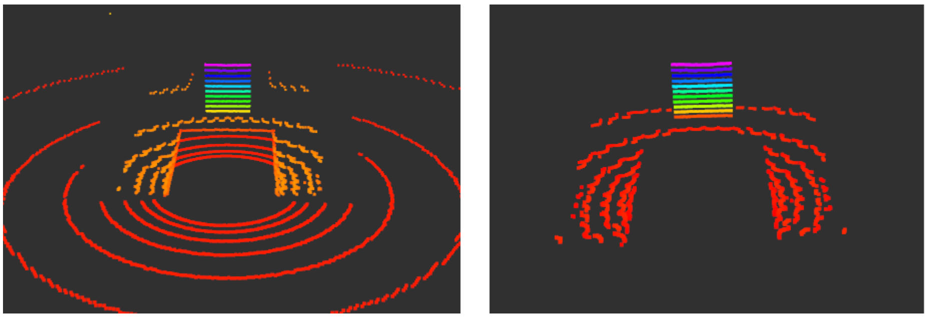


Figure 3.26: Comparison between point clouds before and after filtering.

FOV needs to be adjusted. Since there is no deviation of the vessel, induced by the waves, the field of view is adapted by simply adding or subtracting to it the robot's height relative to the waves, situations b and c, respectively. The biggest challenge is posed by situation d) as the LIDAR's FOV is not only misadjusted, but the ASV's orientation is deviated from the platform. To solve such problem, the IMU data is used so that one may correct the vessel's orientation by projecting it on the correct set of axis. Only then the relative height computed from said projection is used to readjust the field of view.

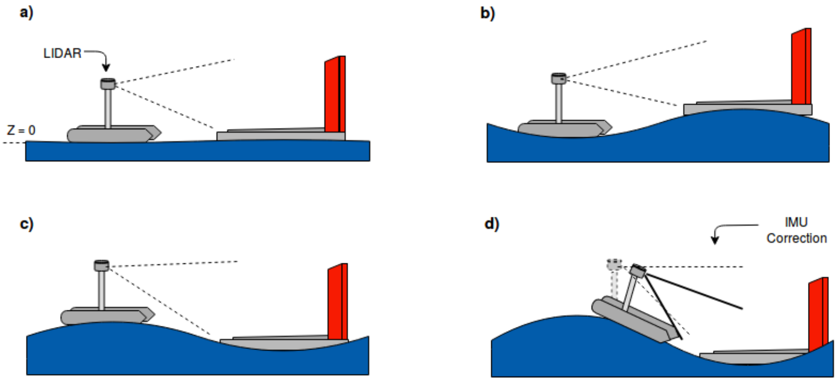


Figure 3.27: Point cloud filter adjusted for multiple scenarios.

Once the filter is adjusted to the wave conditions, the remaining dimensions can easily be computed by using the expected width of the docking platform as a parameter to filter horizontally, and the distance to the dock to filter out in depth. The resulting filter defines box type region that encloses the platform, discarding non relevant information from the surroundings. However the density of such information is very much dependent from the relative distance between the vessel and dock. By fusing together information from multiple data samples a more dense point cloud can be obtained, which is crucial for a better convergence of the registration method.

A sliding window is then implemented to achieve this purpose, adjusting its memory as the ASV approaches the platform. The further away the vessel is, more memory is allocated, therefore more point clouds are stored, increasing the window's size. Such is done due to the fact that the gathered data will be very sparse and incomplete and more samples are needed to compute a cloud dense enough to resemble the platform. As the ASV reaches closer to the platform, progressively less point clouds are kept in memory, as information gets more complete processing it becomes heavier, thus fusing multiple point clouds that are already dense enough would not only aggravate such computational demands but also provide no additional information. Another reason not to keep as much old data is the fact once the vessel is in close proximity to the dock, access to real-time data becomes essential to conduct a safe navigation. Figure 3.28 illustrates how the memory is allocated, every time a new point cloud is processed by the filter it occupies the first position of the window, shifting all the remaining clouds by a position. The oldest point cloud is therefore always stored in the maximum position, equivalent to the size of the window, and removed when a new cloud is inserted. As described above, as the ASV moves towards the platform, the size of said window is dynamically adjusted, decreasing its capacity as it reaches the goal. When capacity is shortened, the oldest data is removed and the new one is concatenated with the remaining stored point clouds.

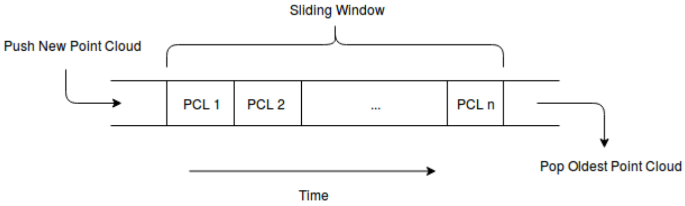


Figure 3.28: Memory allocation in the sliding window.

The application of the described sliding window is shown in figure 3.29. On the left is the point cloud after filtering, and on the right is the result of the sliding window, the density of the point cloud is clearly improved, providing a much more viable input cloud for the registration method.

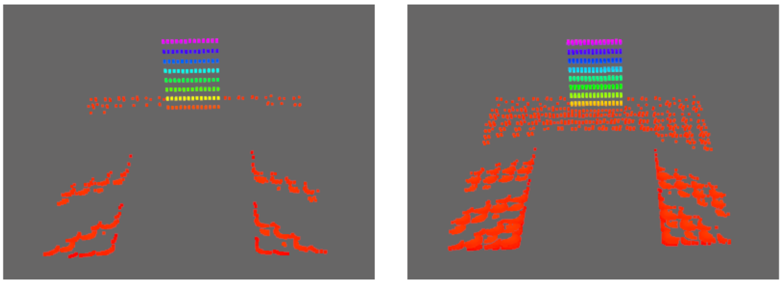


Figure 3.29: Comparison between the filtered point cloud before and after the Sliding Window implementation.

The distance from the dock affects the information gathered by the LIDAR, as the ASV gets closer to the floating platform, more reliable and precise data can be collected. Based on this premise, increasingly more complex templates should be used for matching by the registration method. This hierarchical approach can be observed in figure 3.30, the degree of detail can be extended to n levels, each correspondent model more complex than the previous [9]. Each template has been developed within a 3D modelling software, *Blender*, based on data gathered by the LIDAR so that one may achieve a good match between input and template. The model is then exported as a PLY (Polygon File Format) file and loaded as a target point cloud and down-sampled accordingly to match the density of the input data. The Iterative Closest Point (ICP) algorithm is used as the registration method. The algorithm iteratively revises the transformation (composed by a rotation and a translation $[R,T]$) needed to minimize an error metric, between a fixed reference point cloud, one of the models, and the input source cloud computed by the sliding window. Said error metric is denominated fitness value and corresponds to the sum of squared differences between matched pairs of points.

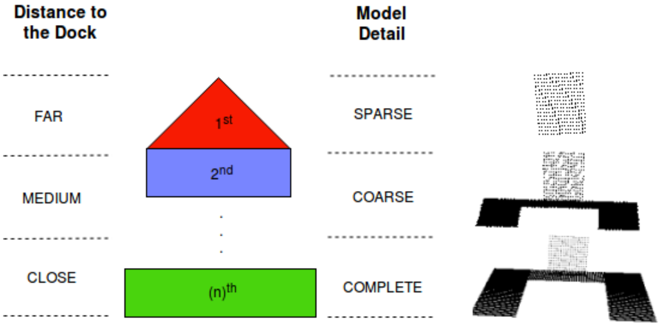


Figure 3.30: Representation of the hierarchical approach to template matching [9].

An example of application for the proposed approach, with 3 levels, is presented in figure 3.31 (only the two extremes are shown). Represented in black are the templates, the green and red point clouds are the input and output, respectively. When the ASV is rather far away from the docking platform, see figure 3.31 c), the most relevant part of the 3D scenario reconstructed by the SA module relies upon the upper structure of the dock. Once it gets closer, figure 3.31 a), a much more complete and reliable point cloud will be gathered, and thus, a template with more detail will be considered during the registration process. The transformations from the initial stage, see figures 3.31 a),c), to the final one, represented by figure 3.31 b) and d) respectively, are returned via the $[R,T]$ matrices previously presented.

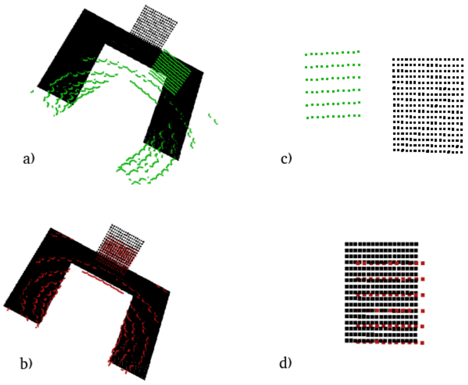


Figure 3.31: Example of application for the proposed hierarchical approach with ICP. a), c) Close and Far view templates (black), respectively, and input point cloud (green); b), d) Clouds are aligned with the targets (red) after the iteration [9].

As an attempt to lower the computational demands required to obtain the presented transformation, the estimated dock pose processed by the Camera Perception subsystem is taken as an initial seed for the ICP. Translating the source point cloud by the distance of the ASV to the dock will result in less necessary iterations for the ICP to obtain a convergence. Having both point clouds minimally overlapped, but keeping their relative orientations, the registration process only needs to compute a transformation that adjusts the rotation between both input and template clouds, and compensate the corresponding needed translation associated to said rotation. Figure 3.32 presents a visual comparison between both input point clouds before and after the initial seed is used (on the left and on the right, respectively).

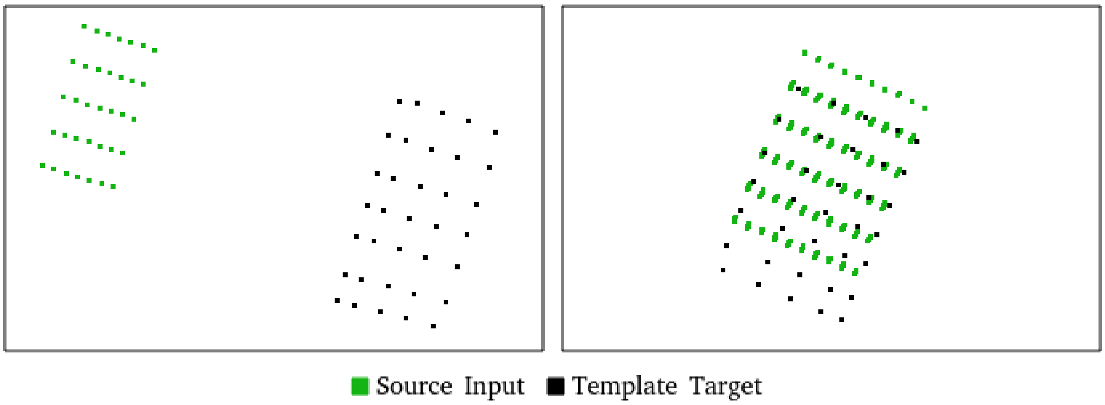


Figure 3.32: Comparison of the ICP input clouds before (left) and after (right) considering an initial seed.

The estimated translation and rotation obtained from the ICP convergence, $[R,T]$, is representative of necessary transformation to align both clouds, from input to target. The generated trajectory for the docking operation is based on this information as target points for the ASV are computed iteratively as the approach to the platform is taking place. To estimate a target point from this information, with basis on previously gathered knowledge of the platform's geometrical features (height and width of the berthing area) and the GPS data, UTM (Universal Transverse Mercator) coordinates of the dock, an optimal docking point is calculated and used as the reference for the algorithm. Take this target as a representation of an ideal position and orientation for the ASV at each iteration. Associated to each of the models used in the hierarchical approach, figure 3.30, is one of these reference targets. These are represented by the green arrow in figure 3.33, observing said figure one is able to conclude that as the models get more complex and complete, these reference points get more accurate and definitive leading up to the final docking location of the ASV when the n^{th} model is loaded in. Such is done to ensure the safety of the navigation, as the data retrieved from a far is associated with a certain degree of uncertainty, the reference point is placed further away from the platform to avoid possible collisions.

As the reference point is known in UTM coordinates and the transformation from input to template is already computed by the ICP, $[R|T]$, equations 3.21 and 3.22 are used to obtain these coordinates relative to the ASV's current position. Since the input source cloud has been previously translated to improve the registration's method performance, that same compensation must now be applied to the estimated target point so that an accurate value in obtained.

$$P_{target} = R(\theta)^{-1} \cdot P_{ref} - T - \xi \tag{3.21}$$

$$\begin{bmatrix} X_{target} \\ Y_{target} \end{bmatrix} = \begin{bmatrix} cos(\theta) & sin(\theta) \\ -sin(\theta) & cos(\theta) \end{bmatrix} \cdot \begin{bmatrix} X_{ref} \\ Y_{ref} \end{bmatrix} - \begin{bmatrix} t_x \\ t_y \end{bmatrix} - \begin{bmatrix} \xi_x \\ \xi_y \end{bmatrix} \tag{3.22}$$

where P_{target} is the target point after the transformation, P_{ref} is the point taken as reference (in UTM coordinates), R(θ) and T are the Rotation and Translation matrices returned by the ICP, respectively, and ξ represents the translation needed for compensating the initial seed used for the source input cloud.

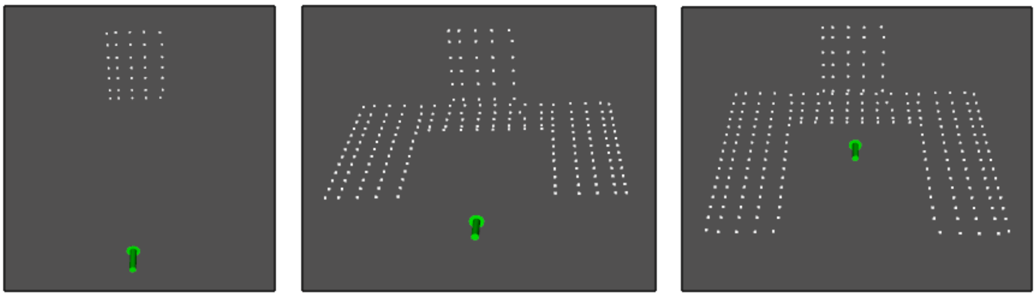


Figure 3.33: Reference target points associated to each template iteration.

The result of said transformation is presented in figure 3.34, where the obtained target is represented by the blue arrow, while the reference is depicted as the green one. Visible deviations in both orientation and position are easily perceived between both representations, as the potential target is now referenced to the ASV's coordinate space. This potential target point is now to be processed by the maneuver module, which is responsible for deciding whether this is a valid point or not.

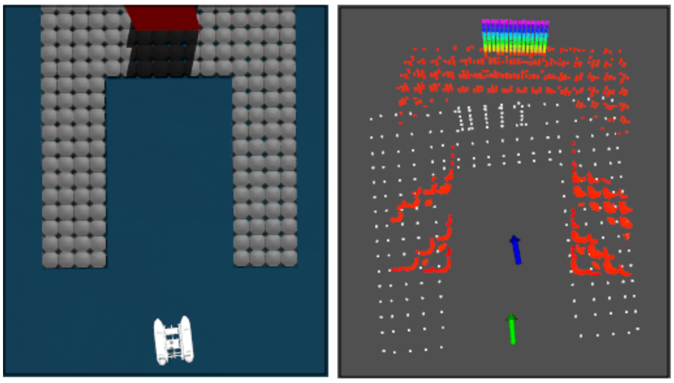


Figure 3.34: Visual representation of the reference target point (green arrow) and the potential target point (blue arrow).

3.4.2 Maneuver Module

The Maneuver Module is designed to act as a complementary system towards the SA algorithm, making the connecting it to the velocity controller, while ensuring the safe navigation of the vessel. Once a new potential target point for the docking is received, the heuristic presented in figure 3.35 is used to check if the given target is valid or not, potentially causing harm towards the vessel and/or mooring facilities. To do such evaluation, this viability check heuristic uses the quality metric of each convergence, calculated based on the registration process, and the spatial representation of the scenario. If the received target proves safe, a trajectory is generated from the ASV's current position towards the target, on the other hand, if said target is rejected, the vessel stays in place and rotates towards the platform. A simple control module is implemented to compute both the adequate linear and angular velocities.

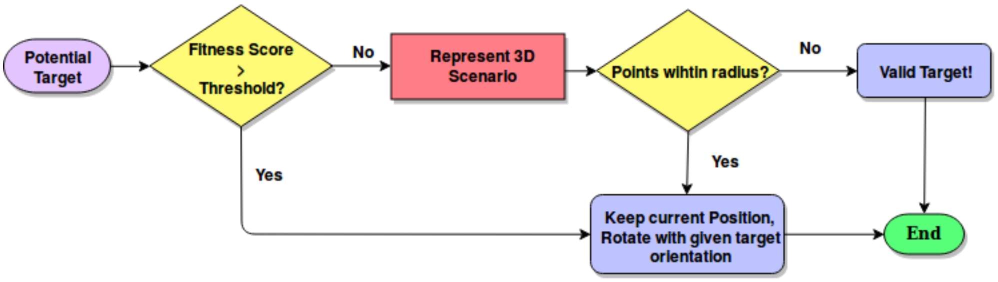

Potential Target
Fitness Score > Threshold?
No
Yes
Represent 3D Scenario
Points wihtin radius?
No
Yes
Valid Target!
Keep current Position, Rotate with given target orientation
End

Figure 3.35: Flowchart representative of the target validation heuristic.

Each time the ICP converges, a fitness score is also returned. This score is calculated by summing the squared distances from source to target, being indicative of the quality of the convergence. For this heuristic, said score is used as the first metric of evaluation. If the target point, correspondent to the latest convergence of the ICP, has an associated fitness score higher than a threshold defined *a priori* it is considered an unsafe target point. When this occurs, the estimated X and Y coordinates of the target are ignored, the ASV stays on its current location and uses the target's orientation to rotate under its own axis. As it aligns with the docking platform, the LIDAR should be able to provide better data which will consequently generate a more viable target point.

The navigation control is based on the center of mass of the ASV, if the threshold value of the first metric is respected, the 3D scenario is represented and a safe area around the ASV is defined with a radius that takes into consideration both the dimensions of the vessel as well as a safe margin distance towards the platform. A *KdTree* was then implemented to carry out said radius check as it proves an efficient and easily adaptable method, allowing for a used-defined radius that can be configurable for each vessel's dimensions. This aims to ensure that the given coordinates of the potential target do not lay inside of the input point cloud. If there are any points within radius, the target is considered an unsafe point and its position is discarded, following principle previously described. Otherwise, it is accepted as valid target point and both the position and orientation are kept and used to generate the adequate path. A visual representation of the radius check process is presented in figure 3.36. For this example, as there are points from the source point cloud that lay within the circumference computed around the target, there is potential for a collision and therefore the target is rejected.

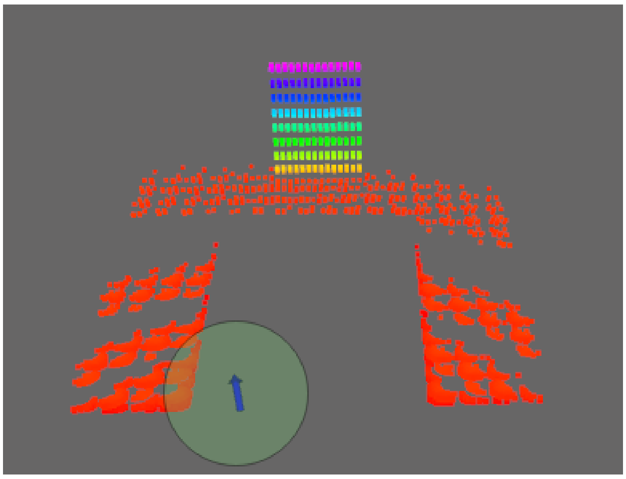


Figure 3.36: Representation of the radius check performed by the evaluation heuristic.

To navigate the vehicle towards its desired berthing location, a control module was implemented by taking into consideration the direction and distance to the desired target point. This path is discretized into small (free) sections and, once it is reached, the final orientation is corrected. The position error, utilized as an euclidean norm of the X and Y coordinates, was given a tolerance of $|0.1|$ m, while the final θ error was given $|0.1|$ rad, both these tolerances are user-defined and can be adjusted according to the needed precision. This module is also responsible for computing linear and angular velocities, feeding them to the ASV's velocity controller.

By enforcing trajectories towards target points whose viability has been checked via the heuristic presented in figure 3.35, while taking into consideration the kinematic constraints of the vessel when generating the necessary velocities, one is able to ensure the safe navigation of the ASV and consequently a secure and reliable docking operation.

Chapter 4

Results

The current chapter aims to present and discuss the results obtained from testing the described self-docking architecture. These tests are conducted to verify if the the proposed architecture is functional and how multiple environmental conditions affect its behavior and reliability. Both the Catch-Zone and Docking Approach modules will be analyzed, separately, in different conditions, the first will be tested for each of the scenarios presented in figure 3.8 of chapter 3, to ensure that the system is able to navigate the ASV towards the catch-zone, considering different orientations and positions. The latter will be tested within different trajectories, that start inside the predefined catch-zone, so that one can determine if a safe docking operation can be conducted. The ability of SA system to recognize the platform and detect its features will also be studied in different light conditions and for multiple wave scenarios. Finally, all the integrating modules of the architecture will be tested working simultaneously, so that one can assess if the developed Supervisory Control System is capable of choosing the correct operation mode for each moment in time and therefore conduct a successful mission.

As previously discussed in section 3.1, tests will be performed within a Gazebo simulation that resembles a maritime scenario, the models used for both the ASV and the docking platform are the ones presented in section 3.1. Visual context of the described world is offered in figure 4.1. Testing within a simulation allows for a reproducible scenario where elements such as the mechanical wear and battery autonomy of the vessel are non-factors, leading to a stable and constant environment from which reliable results and conclusions can be obtained. To accurately simulate such a scenario, the Open Dynamics Engine (ODE[1]) is used to model the physics of the world and its interactions with the robot, such as inertial forces which pose an important aspect to consider on a docking operation. Environmental constraints are perhaps the most important variable to take into consideration, therefore these must be introduced. Such is done via a plugin that allows for multiple parameter configuration.

To test the proposed architecture within various conditions, two test beds were defined. The first one with Minimal Environmental Constraints, in which no wind is considered and waves are

[1] https://www.ode.org/

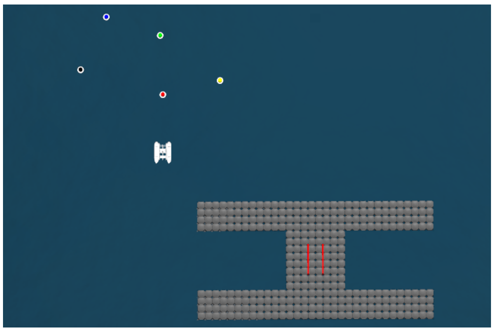


Figure 4.1: Simulated environment used in testing.

divided into three components that have configurable parameters such as amplitude, period and direction. The configuration of each wave component used for this test bed is as follows:

1. An amplitude of 0.06 m, a period of 12.6 s and a direction of ($\vec{x}$,$\vec{y}$) = (-1,0);
2. An amplitude of 0.04 m, a period of 3.7 s and a direction of ($\vec{x}$,$\vec{y}$) = (-0.7,0.7);
3. An amplitude of 0.03 m, a period of 6.3 s and a direction of ($\vec{x}$,$\vec{y}$) = (0.7,0.7);

The second test bed - Harsh Environmental Conditions, aims to simulate stronger weather constraints so that one is able to assess the robustness of the perception system when hindered by the effects that these pose. For said purpose, the amplitude of all the wave components presented above are scaled three times composing a wave of about a meter high, when all these add up. Wind was introduced with ($\vec{V}_x$,$\vec{V}_y$) = (0.7,0.7) m/s and (Roll,Pitch,Yaw) = (2,0.5,0.33) rad. These test beds can be adapted to each vessel in testing so that one is able to accurately simulate the wanted effects on the ASV. For a small vessel such as the Zarco these are considered to be strong enough waves to hinder the system's functionality, however, a cargo ship would obviously need larger scaled environmental conditions for the simulation to produce the same effects on the vessel.

4.1 Catch-Zone Approach

To ensure that the ASV is within an area where its sensors can retrieve complete and viable information about the platform, the Catch-Zone Approach module is tasked with generating an acceptable zone upon which a viable trajectory can be computed. Said trajectory is enforced by estimating linear and angular velocities that lay within a reachable set that respects both the vessel's dynamics as well as the zone's spatial constraints. The viability of the presented approach is evaluated by having the ASV start its course within each of the four quadrants presented in figure 3.8. The first objective is to validate if an acceptable zone for each trajectory can be generated and how this zone is updated during navigation to accommodate the vessel's dynamics as well as the trajectory's geometrical constraints.

The first iteration of the algorithm for each of the situations is presented in figure 4.2. The areas represented in white are the ones correspondent to the acceptable zones and define the spatial limitations of the trajectory, the Catch-Zone Approach module is then capable of generating constraints for the trajectory within any of the quadrants. As it is noticeable the areas correspondent to scenarios a)-b) and c)-d), have mirrored acceptable zones, which is to be expected as these represent the same starting conditions having only one of the axis inverted. To reduce redundancy, scenarios a) and d) are used to study the evolution of the trajectories resultant from this module since these cover all the starting conditions and the remaining ones would have a similar behavior.

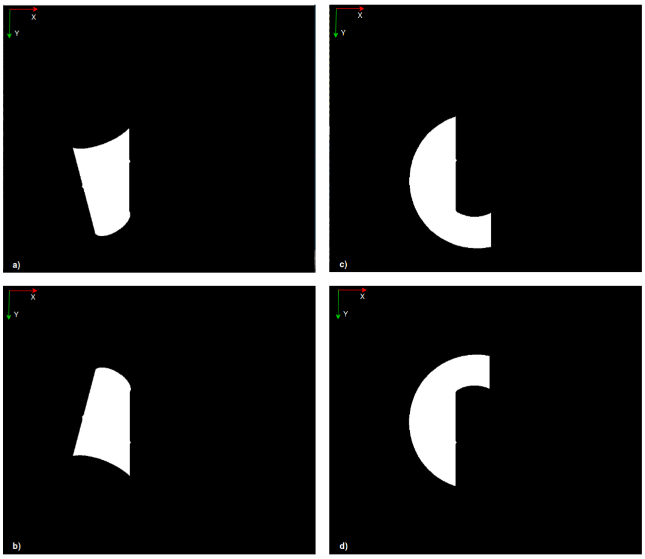


Figure 4.2: Computed acceptable zones for each of the scenarios.

The evolution of the acceptable zone obtained by the algorithm during the execution of trajectory a) is illustrated by figure 4.3. The black cross marks the target location which is resultant from the projection of the dock forward, the green zone represents the reachable set of velocities that the algorithm may choose from to conduct navigation and the red point is representative of the velocity being enforced at each moment. As described in section 3.3 the chosen velocity is estimated from the point that minimizes the distance to the target location that lays inside the reachable zone, as the ASV reaches the goal, both the point representative of the velocity and the target point overlap enforcing the vessel to stay in place, ending the trajectory. Such can be observed in figure 4.3 d).

A harsher challenge is posed by scenario d), since not only does it have its initial starting point close to the platform but also behind it. This is one of the reasons why the target location is projected forward and not directly next to the dock, allowing for a wider area to work with, leading to a safer trajectory. Figure 4.4 depicts the evolution of the acceptable and reachable zones for trajectory d), as expected a large curvature is necessary to conduct the navigation towards the catch-zone.

To prevent the fact that these zones are calculated based on ellipses from hindering the efficiency of the generated trajectory, an heuristic was proposed in section 3.3, that aims to adjust the minor to major axis ratio of said ellipses to mold the acceptable zone, allowing for straight lined trajectories and not adding to them unnecessary curvature. This heuristic is proven to work since, by looking at figure 4.3 and figure 4.4, one can see that the first follows a simple line trajectory towards goal and the latter is adjusted to the necessary curvature of the navigation.

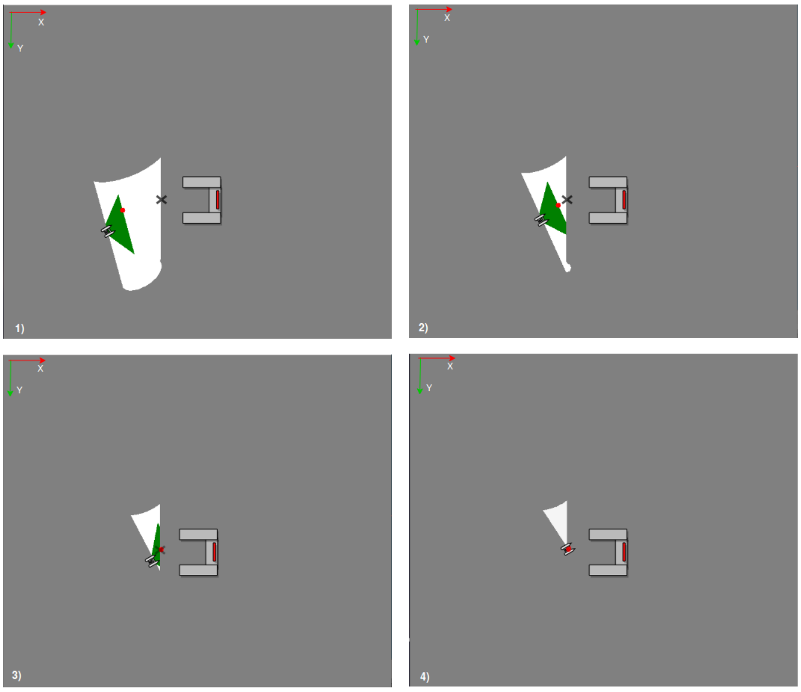


Figure 4.3: Evolution of the acceptable and reachable zones during trajectory a).

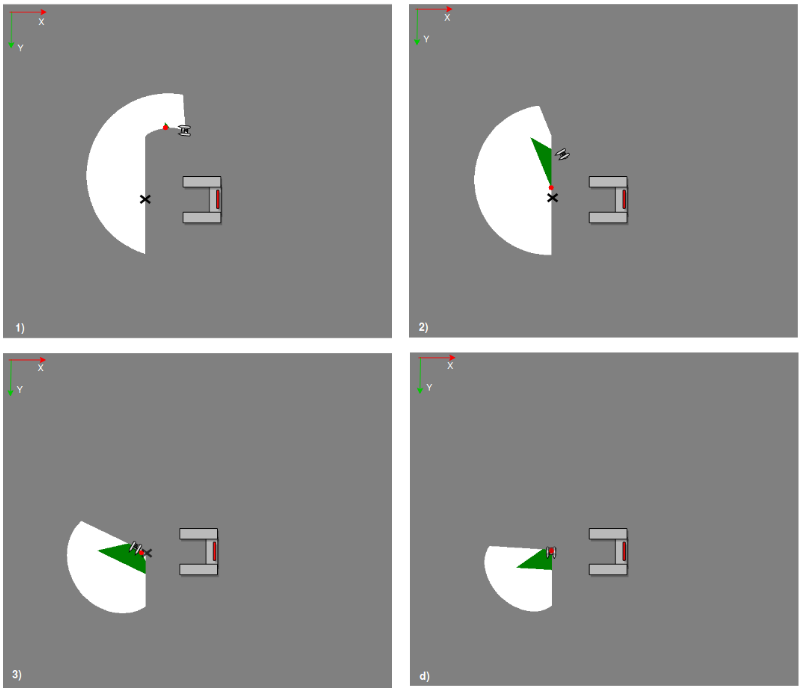


Figure 4.4: Evolution of the acceptable and reachable zones during trajectory d).

Both the Catch-Zone and Reorientation modules serve the purpose of guaranteeing that the situational awareness system is able to retrieve quality data from the platform. A noticeable deviation in orientation can be observed in both trajectories when the ASV reaches the goal, as expected, a trajectory such as scenario d) imposes would lead to a greater disorientation when compared to trajectory a), since the latter starts facing the platform. When the Catch-Zone Approach ends its mission, delivering the ASV into the catch-zone, the Reorientation module will be tested so that one can assess the evolution of the orientation error towards the dock during the trajectory and how it poses when said module is activated. As the information captured by the camera is highly dependent on the ASV's orientation towards the dock, the confidence value of the information reconstructed with basis on camera data is also studied as the ASV re-orientates.

The conclusions presented above about the evolution of both zones are independent of environmental conditions since these representations are updated iteratively and based on predefined geometry and vessel's kinematics, however the study of the performances of the planning and situational awareness algorithms are highly impacted by these constraints, being by shifting the vessel out of its planned path or even by hindering the ability of the sensors to function properly. For this reason, the remaining tests will be performed within the described test beds.

Minimal Environmental Constraints

The navigation of both trajectories towards the target within the catch-zone was conducted successfully, figure 4.5 presents an overview of said trajectories with minimal effects from the waves. Both are depicted in red, the trajectory from scenario a) is presented on the left, while the trajectory from scenario d) is presented on the right. The green area is representative of the catch-zone towards which the ASV navigates.

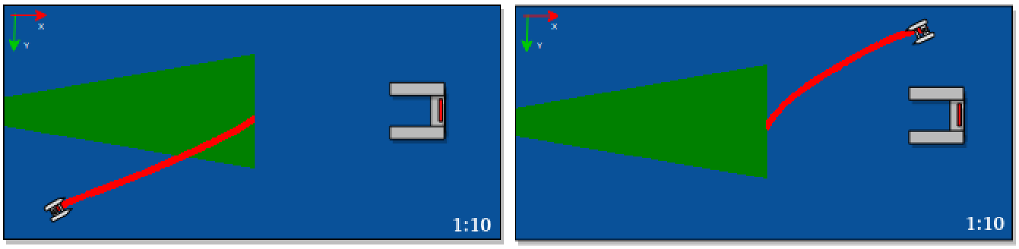


Figure 4.5: Trajectories generated for the approach towards the catch-zone with Minimal Environmental Conditions, scenario a) on the right and scenario d) on the left.

On average, the trajectory depicted in figure 4.5 on the left terminated with the ASV having an absolute deviation from the dock of 0.595 rad, while trajectory d)'s orientation is far worse when it reaches goal, averaging 1.797 rad. Such is expected from the nature of the starting points imposed on each trajectory.

Figure 4.6 presents the evolution of the deviation of the ASV relative to the platform during trajectory a), as well as the confidence associated with the data captured by the camera. These confidence values are always above 80% since the ASV is facing the platform and its deviation from it never hinders the ability of the camera to capture information from it. The variability of these values during course is due to the constant bobbing of the vessel which hinders the ability of the camera to obtain quality information, but not enough to generate bad results. The ASV reaches its goal, around the 330th point, at this moment the Reorientation module is activated to reduce the observed deviation in orientation. As it does so, the confidence value rises slightly as it stabilizes when the vessel stops moving.

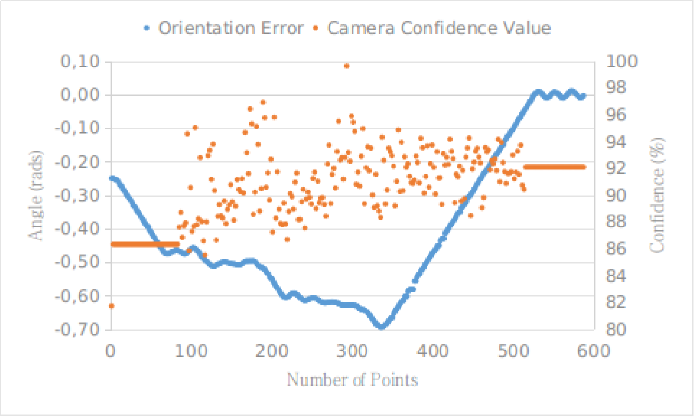


Figure 4.6: Evolution of both the orientation error and confidence value during trajectory a) with Minimal Environmental Conditions.

The trajectory associated with scenario d) ends its course completely deviated from the dock, in fact during navigation it is never facing the platform. Such is confirmed by the confidence values that are obtained and displayed in figure 4.7. This value is obviously zero as there is no visual information about the dock during the trajectory, however, as the goal is reached and the Reorientation module starts to actuate, the camera is able to capture information just about when it enters the defined working FOV (about 0.6 rad). As the deviation towards the dock is compensated, confidence values increases and stabilizes when the operation ends, around the 90% mark.

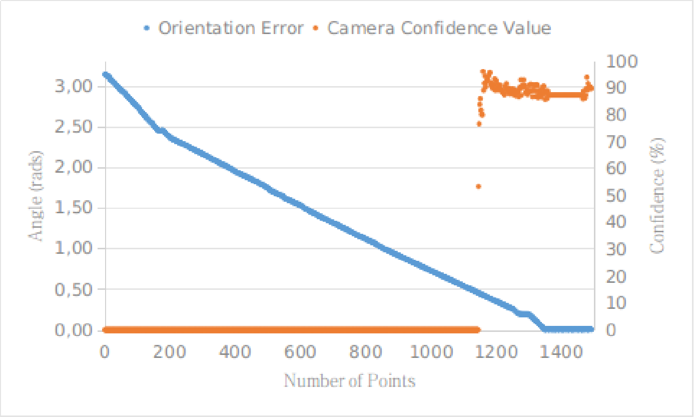


Figure 4.7: Evolution of both the orientation error and confidence value during trajectory d) with Minimal Environmental Conditions.

Harsh Environmental Constraints

The same course was then tested with harsh environmental constraints, trajectories generated from both scenarios a) and d) were still able to navigate the ASV towards the zone. Trajectory a) suffers very few effects from the stronger waves, while trajectory d) is slightly more impacted during course as the trajectory is noticeably not as smooth. When the latter trajectory reaches its goal, the effects are most obvious, as discussed previously the correction of deviation takes much longer for this trajectory, and as such, while the ASV aligns itself with the docking platform it is dragged out of the catch-zone as result of the waves' impact. Both trajectories are illustrated in figure 4.8, the trajectory from scenario a) is presented on the left, while the trajectory from scenario d) is on the right.

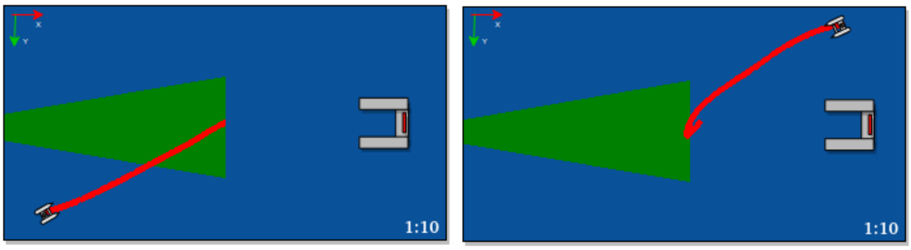


Figure 4.8: Trajectories generated for the approach towards the catch-zone with Harsh Environmental Conditions, scenario a) on the right and scenario d) on the left.

The orientation of the ASV relative to the platform when the goal is reached does not differ much from the values obtained with minimal constraints, trajectory a) ends with an absolute deviation of 0.626 rad and trajectory d) has an error of 1.843 rad, on average. By observing the graphic presented in figure 4.9 one can conclude that the stronger wave conditions directly impact the quality of the data captured from the camera, the constant shift on the vessel's roll and pitch angles will lead to a more unstable confidence value, hence the noticeable variability as the ASV approaches goal. When it starts to reorient towards the platform, not only does the vessel stay in place but the visibility of the platform improves, leading towards a much higher and more stable confidence value.

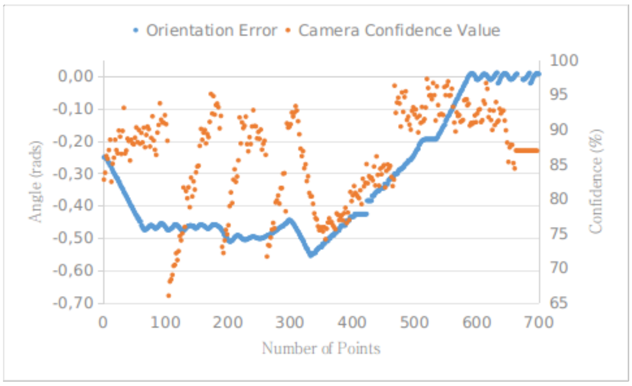


Figure 4.9: Evolution of both the orientation error and confidence value during trajectory a) with Harsh Environmental Conditions.

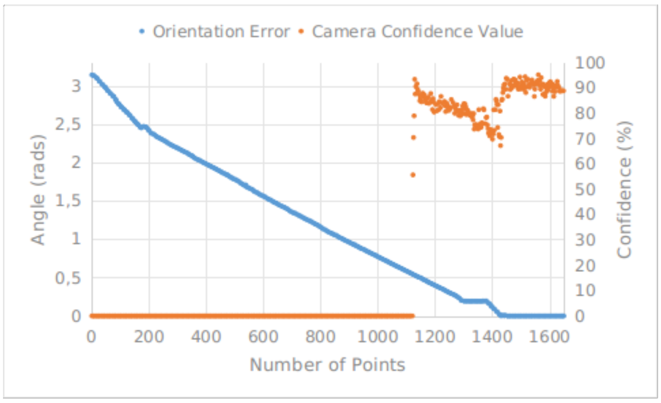


Figure 4.10: Evolution of both the orientation error and confidence value during trajectory d) with Harsh Environmental Conditions.

The evolution of this confidence value during the trajectory generated from scenario d) has a similar behavior independently from the environmental constraints, figure 4.10. When the target location is reached by the ASV, it starts aligning itself with the dock, as discussed from figure 4.8, the vessel gets pushed by the waves resulting in a greater variability of the confidence values between the time instants correspondent to the 1200th and 1400th points. As the ASV gets stable once again, the confidence values improve, reaching approximately 90%.

Conclusions

As conclusion one is able to observe that the course of operation on scenario a) is far shorter than scenario d), as the number of points in the graphics correspondent to each note so. Such is justifiable by the larger deviation in orientation correspondent to trajectory d), taking longer to achieve its desirable orientation. The minimal environmental conditions do not hinder the planned trajectory and its effects on the camera's confidence values are also minimal on both trajectories, averaging values above 88% in both (not taking into account the instants where no information from the camera is received). When compared to the harsh test bed, the obtained results don't vary much. Both trajectories take slightly longer as the waves hinder the vessel's movement and the final orientation of the ASV is similar in both cases. The camera confidence is the variable that is affected the most but even so it still is able to keep confidence above 80%, during both courses. As discussed, this is due to the fact that the vessel is more unstable as result from the environmental conditions. A summary of the obtained results is presented in table 4.1.

The Catch-Zone Approach module is then successful in maneuvering the vessel towards the previously defined catch-zone in all of the tested scenarios. Environmental conditions do affect the data captured by the camera, however it only slightly hinders navigation as the trajectory is updated iteratively and so are the reachable set of velocities that the ASV can reach at each moment, which helps mitigating any possible shift in trajectory caused by the waves.

Table 4.1: Comparison between the results obtained with both sets of environmental constraints.

Scenario		|Average Deviation from the Dock| (rads)	Number of Points	Average Confidence (%)
Minimal Constraints	Trajectory a)	0.595	587	90.6
	Trajectory d)	1.797	1492	88.5
Harsh Constraints	Trajectory a)	0.626	727	86.3
	Trajectory d)	1.843	1517	82.7

4.2 Docking Approach

The module responsible for the approach towards the dock is obviously the most important part of the self-guided architecture as it is not only responsible for detecting the platform and estimate its relative pose to the ASV, but also conduct navigation towards it. The reliability and precision of the operation will be tested in different trajectories, within each of the described test beds, using the position and orientation errors (e_{dist}, e_{θ}) as metrics to evaluate the final docking location, when compared to the reference goal. These are calculated using equations 4.1 and 4.2. Since the situational awareness system is key for a successful operation, the evolution of both the ICP's fitness score and the accuracy of the targets estimated with basis on the transformation matrix resultant from the registration method's convergence will be studied along said trajectories.

$$e_{dist} = \sqrt{(T_x - R_x)^2 + (T_y - R_y)^2} \tag{4.1}$$

$$e_{\theta} = |\theta_T - \theta_R| \tag{4.2}$$

where e_{dist} and e_{θ} are the position and orientation errors, respectively, T represents the final location for the docking operation and R is the reference target point.

As discussed in section 3.4.1, the camera is very sensible to light conditions changes. To assess if the color space transformation from RGB to YCbCr is in fact able to mitigate the effects caused by such luminance variations, the experimental setup presented in figure 4.11 was utilized within the Gazebo simulation. It consists of a light source, that emulates the sun's effects, projected onto both the camera and the platform. The first test aims to project the light rays onto the upper structure of the docking platform so that one can compare the results from the color segmentation algorithm, for both RGB and YCbCr color spaces. A second test would project the light source directly into the camera, causing the lens flare effect that would either create artifacts within the image or make it so that colors are less saturated. However, the latter was not possible to test since the physics of the simulated world do not reproduce the described effects, therefore no conclusion could be obtained on how the color segmentation algorithm fares when exposed to these conditions.

The results of the first test are presented in figure 4.12, on the left is the red component of the image in the RGB color space, the light conditions obviously affect the image, brightening it up and making it even harder for the upper structure geometry to be segmented. On the other hand,

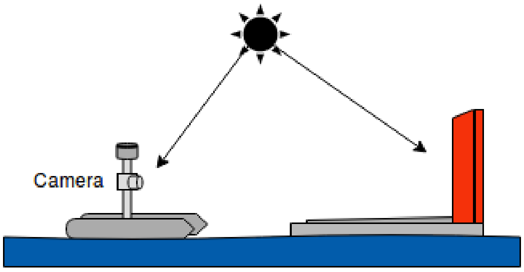


Figure 4.11: Experimental setup used in simulation to test how light conditions affect color segmentation.

the color space transformation does mitigate this light variation as the luminance component of the YCbCr is removed leading to a result that allows for an easy segmentation of the geometry from the background.

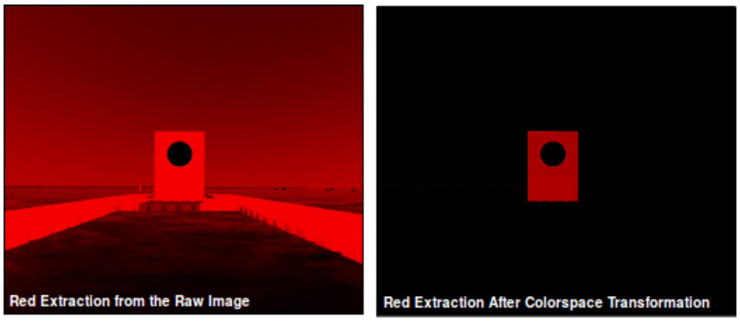

Red Extraction from the Raw Image
Red Extraction After Colorspace Transformation

Figure 4.12: Comparison the color segmentation in both color spaces, for the first test setup.

To test the docking approach, three trajectories were defined, all facing the platform with different orientations (ψ_{ASV}^{Dock}) and with starting points within the catch-zone. Trajectory 1 starting from the left with a deviation of $\psi_{ASV}^{Dock} = -0.35$ rad relative to the dock, trajectory 2 in front of the platform and aligned with it, and finally, starting from the right is trajectory 3 which is deviated by $\psi_{ASV}^{Dock} = 0.35$ rad. Each metric of evaluation, position and orientation, were given a small tolerance values which are presented in table 4.2 along with the remaining parameters used for the maneuver to take place, the nominal linear and angular velocities. Both tolerance parameters are user-defined for the desired precision and the velocities are set considering the vessels kinematics. Each trajectory will be conducted within both the test beds previously described, allowing one to assess the effects of the environmental conditions on the performance of both the maneuver module and the situational awareness system.

Table 4.2: Maneuver parameters used during testing.

Position Tolerance	Orientation Tolerance	Nominal Linear Velocity	Nominal Angular Velocity
(m)	(rads)	(m/s)	(rad/s)
|0.1|	|0.01|	1.2	2.5

Minimal Environmental Constraints

The evolution of the ASV's position during the presented trajectories, within the first test bed, are presented in figure 4.13. Dashed lines are representative of the docking platform's inner walls and the circle of the optimal target point that serves as a reference (with the given tolerance). The final position of the ASV is marked at the end of each trajectory with a cross. Even if minimal, the effects of the waves can still be noticed in both trajectories 1 and 3, represented with the red and the blue colors, respectively. The first gets slightly dragged by the waves, right at the start, as the ASV is adjusting its orientation towards the computed line, while the same happens to the latter while it adjusts its orientation to enter the dock. As expected, trajectory 2 (in green) shows no deviations from course, as it is already aligned with the platform and no reorientation is needed. From the set of tests run in these conditions, an average position error of 0.107 m and a 0.007 rad error in the orientation were obtained, both within tolerance. If given a tighter tolerance, the positional error could potentially be reduced even more, since the current induced error is not enough to trigger a response from the algorithm.

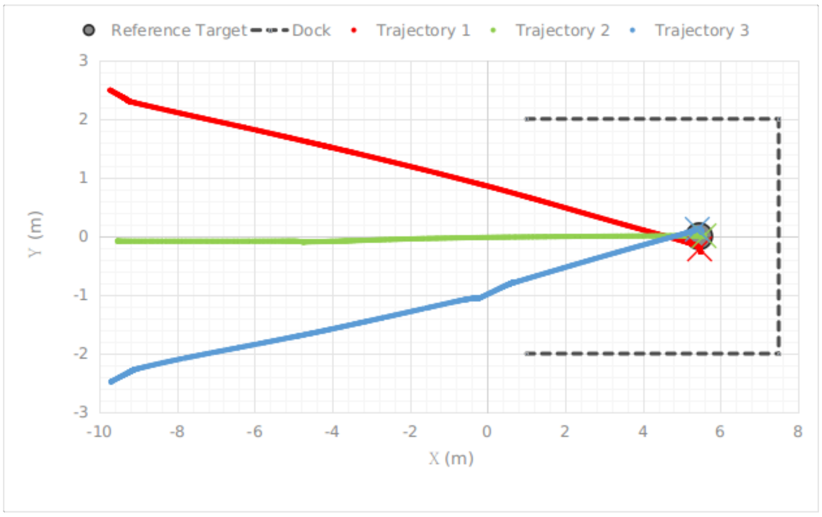


Figure 4.13: Evolution of the ASV's position in three different trajectories, with minimal environmental constraints.

The performance of the SA system was also tested for these environmental constraints. Figure 4.14 depicts the evolution of the fitness score during a trajectory such as the one presented previously as Trajectory 2. Take this score as a metric representative of the quality of the convergence from the registration method, being obtained from the sum of the squared differences between matched pairs of points. As alluded to in section 3.4.1, three layers are used for the hierarchical approach towards template matching, as so, each of the models and its correspondent fitness score are represented in figure 4.14. It is to be expected that when the ASV is further away from the dock, the retrieved data from the LIDAR would lead to a worse convergence and therefore a higher score. However such is not noticeable in figure 4.14, the fitness score relative to the far model is the lowest. This is due to the effects of the sliding window, increasing the density of points and also to the fact that the template model has considerably less points than the remaining two (far model - 7k points; medium model - 80k points; close model - 120k points, before down-sampling) resulting in the lowest error out of the three models. The increase in fitness score as one transitions from the far model to the medium one, is caused by the gradual change in the amount of data that is gathered from the geometry of the dock, as the vessel approaches, points belonging to side pillars of said platform start getting picked up by the LIDAR, since there are no such points in the model, the error increases. When the medium model is in fact loaded in as the template, the density of points of said model is far greater than the gathered point cloud, inducing a small spike in the fitness score. As the ASV reaches goal, the retrieved point cloud gets denser as sliding window processes information and it gets more complete, leading the fitness score to decrease until it stabilizes when in close proximity to the dock.

A solution to obtain a more uniform fitness score between model changeovers is obviously to expand the number of layers used in the template hierarchy, decreasing the point density discrepancy in templates and achieving a smoother transition since the progressively more complete and complex data would better match the reference models. To further minimize the variability within each model's fitness score values, a larger sized sliding window could be utilized leading to a even more complete input source cloud for the registration method, however, to store such quantity of information would have direct impact on the algorithms performance. As its current state, the size of the window has a good cost-effectiveness ratio.

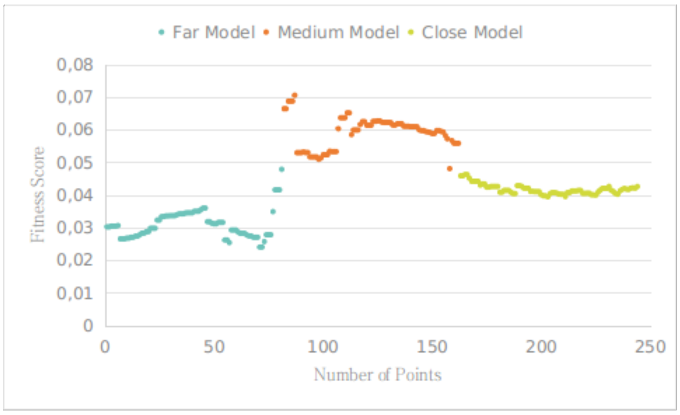


Figure 4.14: Evolution of the fitness score with Minimal Environmental Constraints.

Being able to estimate precise target points relative to each template is a rather important factor in conducting a safe and smooth navigation towards the berth. To achieve such, the transformation obtained from each convergence of the ICP is applied to a reference target point which is then translated into the estimated dock's position. So, the accuracy of these target points depends not only of a good convergence, and consequently a low fitness score, but also from the quality of the data captured from the camera which will produce an estimate for the dock's pose relative to the ASV. Figure 4.15 presents the accuracy of said targets relative to the reference ones, for each of the template models, during the same trajectory aforementioned. The validation check heuristic is seen in application for the points circled out in red as these are rejected, the ASV stays in place and rotates towards the platform until it gathers better information.

Once again, contrary to the expected, the far model has a better performance than the medium one. The targets generated when afar from the dock are quite accurate, excluding some outliers, and there doesn't seem to be any rejected points for the example trajectory. In comparison to the medium model, the target points are much more scattered from the reference and there is a good amount of rejected points. The reasoning behind this is the fact that the model used when far away from the dock does have less points to cause the radius check to reject said target and the reference is positioned far enough from the dock to account for the risks of the unreliable data being used, consequently a looser threshold is defined while deciding if a point is valid or not. On the other hand, the medium model not only has an associated higher fitness score as discussed above, but it is also loaded as model during the period of time that the ASV reaches the entrance of the dock. To ensure that the right orientation is achieved by the vessel while entering berth, the threshold used for rejecting targets within this model is much tighter, combined with the radius check now having a lot more point to iterate through, the number of rejected points increases. As the ASV is oriented and close to the platform, the close model generates very accurate target points.

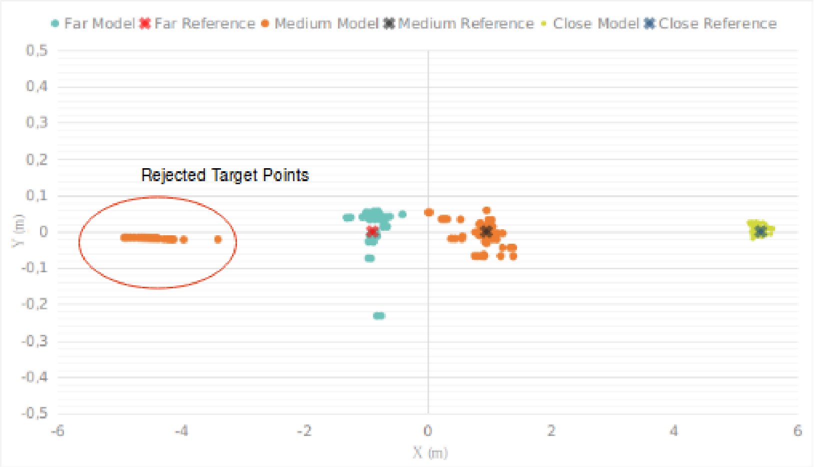


Figure 4.15: Accuracy of the estimated target points for each model, with Minimal Environmental Constraints.

Harsh Environmental Constraints

The results obtained from testing the same three trajectories on the harsh conditions test bed are presented in figure 4.16. From the initial instants of each trajectory, a clear influence of the waves and wind can be observed as these drag the vessel along the negative Y axis. During the first trajectory (represented in red), the ASV takes advantage of the the momentum generated from these constraints and reaches the target much faster, this can be seen as this trajectory has a visible lower number of points. However, the extra momentum ends up causing a small overshoot in the vessel's final position. The path that begins aligned with the dock, trajectory 2, suffers a clear deviation along its route when the ASV stops to rotate and adjust its orientation, as the vessel reaches its goal and stops to readjust the final orientation, a visible drag starts to occur, pushing it out of the given tolerance area. Finally, the trajectory represented in blue starts in the worse possible scenario, as the waves counter the desired action multiple times. Reflecting the behavior of the path in green, when the ASV stops to adjust its orientation, a visible drag occurs that leads to an overshoot on final position.

The tests performed in these conditions had an average position error of 0.427 m and a 0.009 rad error in the orientation. To achieve such an accurate orientation value, the ASV spends a good amount of time readjusting its final orientation. During this time period, the waves keep dragging the vessel out of the desired final position leading up to a considerable positional error. As the vessel has a differential drive, no possible lateral compensation can be given as the drag occurs.

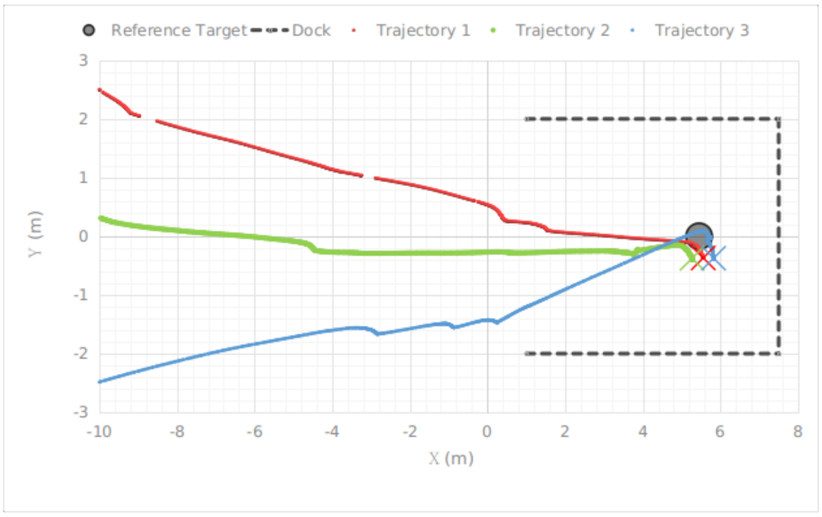


Figure 4.16: Evolution of the ASV's position in three different trajectories, with Harsh environmental constraints.

As the SA system heavily relies on the data gathered from the LIDAR and the camera, it is to be expected that these harsh conditions could hinder its efficiency and precision. The results obtained from testing how the system responds to these constraints, during a trajectory such as number 2, are presented in both figures 4.17 and 4.18. The evolution of the fitness score suffers from the same transition problem between models discussed before, however the higher error associated to these outliers can be associated to the less reliable data that is retrieved within these conditions, leading to worse convergences estimated by the registration method. Excluding these, the overall average fitness score, related to each model, is worse when compared to the previous set of tests (table 4.4), however only slightly, which confirms the effectiveness of the approach taken for the point cloud filter on mitigating these conditions and consequently improving the quality and reliability of the gathered data.

The higher variability of the camera confidence values, will definitely lead to less accurate target points as these depend on the estimation of the dock's pose relative to the ASV. Figure 4.18 corroborates this fact, the target points generated for the far and medium models are much more scattered around the reference, consequently a higher number of points are rejected which translates into more trajectory readjustments during navigation.

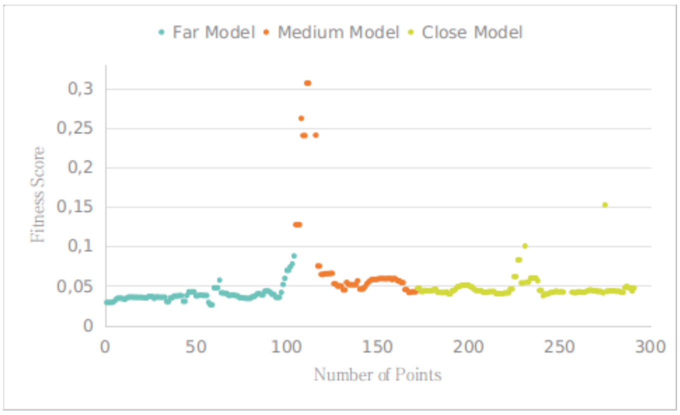


Figure 4.17: Evolution of the fitness score with Harsh Environmental Constraints.

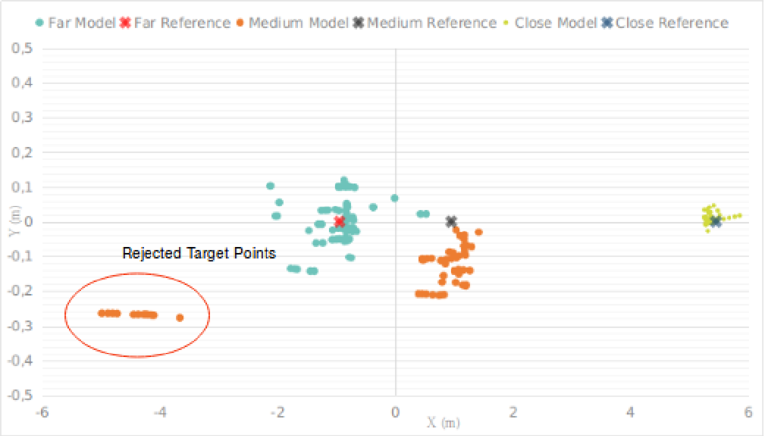


Figure 4.18: Accuracy of the estimated target points for each model, with Harsh Environmental Constraints.

Conclusions

The effects induced by the waves and wind are easily noticeable when comparing the results obtained from each set of tests, table 4.3. For minimal environmental conditions the navigation is conducted almost flawlessly, achieving very satisfactory distance and orientation errors, both within tolerance, on the other hand when tested with harsh constraints, the results are obviously affected. A safe docking is conducted regardless, however, to obtain a very precise final orientation one must sacrifice the positional accuracy since the ASV cannot move laterally to compensate the drag from the waves.

Table 4.3: Summarized comparison between trajectory precision, in both scenarios.

Scenario	Position Error Average (m)	Orientation Error Average (rads)
Minimal Constraints	0.107	0.007
Harsh Constraints	0.427	0.009

By observing the SA system's response to both test beds, one is able to conclude that both the implemented sliding window and the point cloud filter ensure a good convergence from the ICP, as its average fitness score does not vary much when tested in different conditions, table 4.4. The camera confidence values however have direct impact on the estimated target points accuracy, when the SA system is tested in harsh conditions, the confidence suffers from way more variability, inducing a more scattered set of target points. As such, a lot more of these points are rejected, mostly within the medium model as the percentage of rejected targets increases noticeably.

Table 4.4: Summarized comparison of the performance from the Situational Awareness system in both scenarios.

Scenario		Far Model	Medium Model	Close Model
Minimal Constraints	Average Fitness Score	0.04209	0.06031	0.04311
	#Rejected Targets (%)	5.56	14.58	2.01
Harsh Constraints	Average Fitness Score	0.06349	0.09552	0.04688
	#Rejected Targets (%)	16.13	35.60	6.61

4.3 Architecture Integration

The final section of the Results chapter aims to assess if the supervisory control system is able to oversee a complete docking operation and if its decision making is adequate for each moment in time. Starting points similar to the ones used to study the Catch-Zone Approach were used within the aforementioned test beds, starting one operation in front of the dock and another one behind it.

Minimal Environmental Constraints

The tests performed separately on the Catch-Zone and Docking Approach modules have proven successful within this test bed. Now working as a whole, the self-guided docking architecture is also capable of conducting a smooth operation. Once the ASV reaches the catch-zone, the supervisory control system changes the operating mode so that the vessel is reoriented and aligns itself with the platform. When all the conditions are met, the docking approach is enabled and the ASV safely reaches the berth. Figure 4.19 presents the vehicle's trajectory towards the dock (in red), for each of the starting conditions, and each part of the operation is labelled, providing visual context.

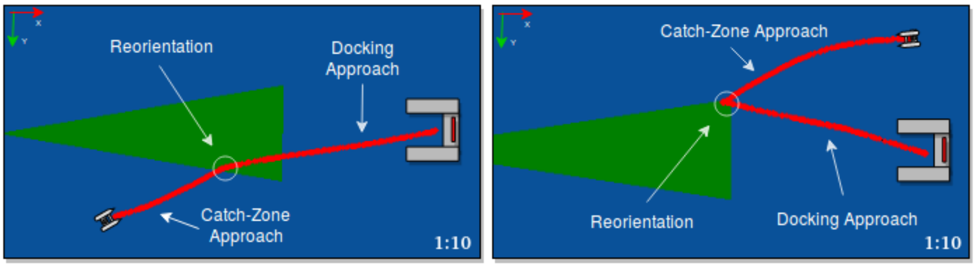


Figure 4.19: Complete docking operation for both scenarios, with Minimal Environmental Conditions.

Harsh Environmental Constraints

For the second set of tests, the architecture also proves able to conduct the docking operation. Figure 4.20 depicts the trajectories that lead the vessels towards berth, for both starting conditions. The effects of the stronger waves and wind are noticeable mostly when the Reorientation module is active, as the ASV stays in place and rotates towards the platform a visible drag affects both trajectories. Once the ASV reaches the docking location and adjusts its final orientation these effects are once again visible, as the vessel is pushed by the waves.

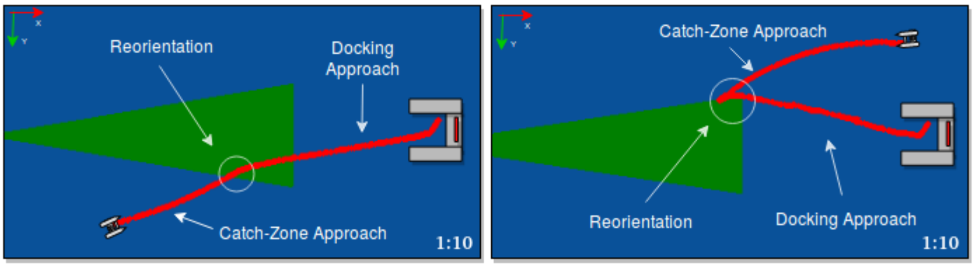


Figure 4.20: Complete docking operation for both scenarios, with Harsh Environmental Conditions.

Conclusions

The Supervisory Control System is proven to choose the correct operating mode at each moment, leading to a successful docking operation during both scenarios, regardless of the maritime conditions. Once the ASV reaches the catch-Zone, the Reorientation module is activated to correct the angular deviation relative to the platform. Finally when both conditions are met, position and orientation, the vessel begins its approach towards the dock.

Both the trajectories that are conducted with harsh environmental conditions take longer and have higher traveled distances when compared to the minimal constraints test bed. This is due to the fact that the Reorientation module takes much more time to readjust the ASV's orientation, since it is constantly hindered by the waves and wind, leading to a longer mission. The drag induced by the waves while this module is active is reflected on the traveled distance, as it is noticeably higher. A summarized comparison between the performance of the full docking operation for both scenarios, within both test beds, is presented in table 4.5.

Table 4.5: Summarized comparison of both missions' travel time and distances.

Scenario		Approach from the Front	Approach from the Back/Side
Minimal Constraints	Average Traveled Distance (m)	26.12	36.90
	Average Mission Time (s)	78	198
Harsh Constraints	Average Traveled Distance (m)	28.02	39.76
	Average Mission Time (s)	142	269

Chapter 5

Conclusions and Future Work

This dissertation proposes an architecture for the self-guided docking of ASVs, with the purpose of contributing towards a safer and more profitable cargo shipping industry, reducing the risk of impact with the platform and consequently preventing environmental catastrophes and possible casualties. The architecture is composed by three main modules, the Catch-Zone Approach, the Reorientation Module and the Docking approach, being supervised by a control system that oversees the operation and decides which one should be active at each moment in time. The Catch-Zone Approach and the Reorientation Module are responsible for ensuring that all the conditions are met so that a safe docking operation may occur, such positioning and orientating the vessel so that quality information from the platform can be retrieved, only then the Docking Approach may commence. This module is composed by a situational awareness system that combines information from sensors such as the LIDAR, camera, GPS and IMU to identify and recognize the dock and compute its relative pose to the ASV. A maneuver module is the complementary module to this system, ensuring a safe navigation through a viability check heuristic.

The proposed architecture has been tested within two sets of simulated experiments, one with minimal environmental effects, and another with harsh ones. The Catch-Zone approach is able to conduct the ASV towards the predefined area, where its on-board sensors are able to capture quality information from the dock, in all the tested scenarios, both in front and behind the platform. The different environmental constraints' effects are visible on the data captured by the camera, however these only slightly hinder navigation. The fact that the generated trajectory is updated iteratively, as well as the set of reachable velocities, helps mitigate any effects induced by the harsh environment. Evaluation of the Docking Approach for each set was based on the position and orientation errors of the final berth location reached by the ASV, when in comparison with the reference point previously calculated. A safe operation is conducted regardless of the environmental conditions, however these directly affect the trajectories conducted within the harsh test bed, as these cause visible drag both during navigation and when the ASV reaches the dock and stops to adjust its final orientation towards it. The Situational Awareness system was also tested during each trajectory, as stated before the data captured by the camera is highly dependent on the wave conditions, since these directly impact the vessel's stability and the consequently the quality of the captured image. On the other hand, both the sliding window and the dynamically adjusted point cloud filter prove to be a fitting solution, as the registration method shows a great convergence rate, regardless of the testing conditions.

Concluding, this dissertation contributes with a functional architecture that is viable for multiple environmental conditions, being able to conduct the operation in diverse scenarios, achieving errors in position and orientation up to 0.107 m and 0.007 rad, respectively. When such is the case, higher positional errors are obtained since the vessel used in testing has a differential drive and cannot compensate the drag induced from the waves and wind, as it reaches the desired final orientation relative to the berth. As future work, the architecture should be tested with the actual Zarco ASV in real maritime conditions so that one is able to assess its behaviour and performance. To further develop the proposed work and improve upon it, the following is proposed:

- Use a deep-learning approach to classify the platform's typology and adjust the docking approach accordingly;
- Adapt the architecture so that the ASV may be able to perform the docking operation towards a suitable enough platform in a non-structured environment;
- Study the possibility of using Inverse Perspective Mapping to obtain a bird's-eye view of the platform and fuse this information with the SA system;
- Implement a motion planning algorithm, for the docking approach module, that takes the vessel's kinematic constraints into account;
- Create a registration methodology to replace ICP and further improve its convergence by minimizing the error between matched points;
- Study the effectiveness of the hierarchical template matching approach on a real dock model, and its impact on the necessary computational power.

References

[1] Zhixiang Liu, Youmin Zhang, Xiang Yu, and Chi Yuan. Unmanned surface vehicles: An overview of developments and challenges. *Annual Reviews in Control*, 41:71–93, 2016.

[2] Lutz Kretschmann, Hans-Christoph Burmeister, and Carlos Jahn. Analyzing the economic benefit of unmanned autonomous ships: An exploratory cost-comparison between an autonomous and a conventional bulk carrier. *Research in Transportation Business & Management*, 25:76–86, 2017.

[3] Rolls Royce. AAWA Position Paper: Remote and Autonomous Ships - The next steps. 2016.

[4] Allianz Global Corporate & Specialty. Global Claims Review: Liability In Focus. pages 1–39, 2017.

[5] M Ventura. COLREGS -International Regulations for Preventing Collisions at Sea. pages 1–74, 2009.

[6] Sebastian Öhland and Axel Stenman. Interaction Between Unmanned Vessels and COLREGS. Bachelor's Thesis, Yrkeshögskolan Novia, 2017.

[7] Danish Maritime Authority. Analysis of Regulatory Barriers to the Use of Autonomous Ships. Technical report, 2017.

[8] Rosa Mari Darbra and Joaquim Casal. Historical analysis of accidents in seaports. *Safety Science*, 42(2):85–98, 2004.

[9] Pedro Leite, Renato Silva, Aníbal Matos, and Andry Maykol Pinto. An Hierarchical Architecture for Docking Autonomous Surface Vehicles. *IEEE International Conference on Autonomous Robot Systems and Competitions*, 2019.

[10] Renato Silva, Pedro Leite, Daniel Campos, and Andry M Pinto. Hybrid Approach to Estimate a Collision-Free Velocity for Autonomous Surface Vehicles. *IEEE International Conference on Autonomous Robot Systems and Competitions*, 2019.

[11] Matteo Schiaretti, Linying Chen, and Rudy R Negenborn. Survey on Autonomous Surface Vessels: Part II - Categorization of 60 Prototypes and Future Applications. In Tolga Bektaş, Stefano Coniglio, Antonio Martinez-Sykora, and Stefan Voß, editors, *Computational Logistics*, pages 234–252. Springer International Publishing, 2017.

[12] Jon Walker. Autonomous Ships Timeline - Comparing Rolls-Royce, Kongsberg, Yara and More. 2019.

[13] Tom van Dijk, Hans Moonen, Harmen van Dorsser, Rudy Negenborn, and Roy van der Berg. SmartPort White Paper - Smart ships and the changing maritime ecosystem. 2018.

[14] Berth control: looking at the potential of auto-berthing technology. *Ship Technology Global*, 2018.

[15] Abhilash Devaraju, Linying Chen, and Rudy R. Negenborn. Autonomous surface vessels in ports: Applications, technologies and port infrastructures. In Raffaele Cerulli, Andrea Raiconi, and Stefan Voß, editors, *Computational Logistics*, pages 86–105, Cham, 2018. Springer International Publishing.

[16] Joe Baker. How should ports prepare for autonomous shipping. 2018.

[17] Rui Campos, Tiago Oliveira, Nuno Cruz, Aníbal Matos, and José Miguel Almeida. BLUECOM+: Cost-effective broadband communications at remote ocean areas. *OCEANS 2016 - Shanghai*, pages 1–6, 2016.

[18] André Coelho, Mario Lopes, Bruno Ferreira, Rui Campos, and Manuel Ricardo. Experimental evaluation of shore to unmanned surface vehicle Wi-Fi communications. *IFIP Wireless Days*, pages 86–91, 2018.

[19] Ralph Jennings. China Is Developing Ships To Cover The Globe Without Captains. 2018.

[20] MOL Launches R & D on Autonomous Ocean Transport System - Selected for Japanese Government Transportation Research Program - | Mitsui O.S.K. Lines. URL: https://www.mol.co.jp/en/pr/2017/17031.html, [last visited on 2018-11-18].

[21] Thomas Herpel, Christoph Lauer, Reinhard German, and Johannes Salzberger. Trade-off between coverage and robustness of automotive environment sensor systems. *ISSNIP 2008 - Proceedings of the 2008 International Conference on Intelligent Sensors, Sensor Networks and Information Processing*, pages 551–556, 2008.

[22] Vítor Santos, J. Almeida, E. Ávila, D. Gameiro, M. Oliveira, R. Pascoal, R. Sabino, and P. Stein. ATLASCAR - Technologies for a computer assisted driving system on board a common automobile. *IEEE Conference on Intelligent Transportation Systems, Proceedings, ITSC*, pages 1421–1427, 2010.

[23] John Wallace. IR IMAGING: SWIR cameras spot unwanted activities through fog and at night. Laser Focus World, 2013.

[24] Nicolas Pinchon, M Ibn-Khedher, Olivier Cassignol, A Nicolas, Frédéric Bernardin, Patrick Leduc, Jean Philippe Tarel, Roland Bremond, Emmanuel Bercier, G Julien, N Pinchon, O Cassignol, F Bernardin, P Leduc, J-P Tarel, R Brémond, and E Bercier. All-weather vision for automotive safety: which spectral band? *International Conference Night Drive Tests and Exhibition*, 2016.

[25] Steffen Heuel. Radar Waveforms for A&D and Automotive Radar - White Paper. 2013.

[26] J.M. Wozencraft. Complete coastal mapping with airborne lidar. In *Oceans '02 MTS/IEEE*, volume 2, pages 1194–1198. *OCEANS '02 MTS/IEEE*, 2002.

[27] K. Omasa, F. Hosoi, and A. Konishi. 3D lidar imaging for detecting and understanding plant responses and canopy structure. *Journal of Experimental Botany*, 58(4):881–898, 2006.

[28] Joel M. Esposito and Mitchell Graves. An algorithm to identify docking locations for autonomous surface vessels from 3-D LiDAR scans. *IEEE Conference on Technologies for Practical Robot Applications, TePRA*, 2014.

[29] Guo Chenguang, Li Xianglong, Zhong Linfeng, and Luo Xiang. A Fast and Accurate Corner Detector Based on Harris Algorithm. In *2009 Third International Symposium on Intelligent Information Technology Application*, pages 49–52. IEEE, 2009.

[30] T. Hanif and M.B. Sandler. 2D shape reconstruction from the Hough transform. In *IET Conference Publication*, 1993.

[31] Jooho Lee, Joohyun Woo, and Nakwan Kim. Vision and 2D LiDAR based autonomous surface vehicle docking for identify symbols and dock task in 2016 Maritime RobotX Challenge. *2017 IEEE OES International Symposium on Underwater Technology, UT 2017*, 2017.

[32] Andrew Webb, Bradley Donnelly, Jesse Stewart, Jonathan Wheare, Michael Kossatz, Scott Hutchinson, Shane Geyer, and Tenzin Crouch. Development and Testing of the TopCat Autonomous Surface Vessel for the Maritime RobotX Challenge 2016. *RobotX Forum*, 2016.

[33] Yuan Ye, Fan Guangrui, and Ou Shiqi. An Algorithm for Judging Points Inside or Outside a Polygon. In *2013 Seventh International Conference on Image and Graphics*, pages 690–693. IEEE, 2013.

[34] Christopher M Ash, Nate D Bloom, Jeremy D Brown, Abby N Butka, Stephen P Cronin, Grady C Delp, Robert T Goring, Christopher J Hockley, Zachary R Joswick, Diego Lodato, Nicholas R Middlebrooks, Brandon F Mathews, Joshua L Pletz, Jefferson S Romney, Marco A Schoener, Nicholas C Schultz, Alena M Thompson, David J Thompson, Johazais K Wyble, Timothy A Zuercher, Charles F Reinholtz, Eric J Coyle, Brian K Butka, and Patrick N Currier. Design of the Minion Research Platform for the 2016 Maritime RobotX Challenge. *RobotX Forum*, 2016.

[35] Joohyun Woo, Induk Seo, Jooho Lee, Jongyeol Park, Ahmin Park, Midan Kim, Yoowon Jung, Jungho Park, Rokha You, Hujae Choi, and Nakwan Kim. ASV : MACS. *RobotX Forum*, pages 1–7.

[36] S. C. Ng. Principal component analysis to reduce dimension on digital image. *Procedia Computer Science*, 111:113–119, 2017.

[37] Qi Chen, Na Wang, Lingjun Zhao, Jun Lu, Canbin Hu, Yongmei Jiang, and Gangyao Kuang. Harbor detection of remote sensing images based on model. *Proceedings of the 2010 2nd International Conference on Future Computer and Communication, ICFCC 2010*, 1:322–325, 2010.

[38] Zhu Bing, Li Jinzong, and Cheng Aijun. Knowledge Based Recognition of harbor target. *Journal of Systems Engineering and Electronics*, 2005.

[39] Yinling Sui, Huanxin Zou, Lin Lei, and Zhiyong Li. A novel template-based change detection technique for harbor ship target. *Proceedings of the 2011 2nd International Conference on Digital Manufacturing and Automation, ICDMA 2011*, pages 97–100, 2011.

[40] Nils J. Nilsson. *Principles of artificial intelligence*. Springer-Verlag, 1982.

[41] Roland Siegwart, Illah R Nourbakhsh, and Davide Scaramuzza. *Introduction to Autonomous Mobile Robots second edition*. 2011.

[42] Pedro Luís Cerqueira Gomes da Costa. *Planeamento Cooperativo de Tarefas e Trajectórias em Múltiplos Robôs*. PhD thesis, Faculdade de Engenharia da Universidade do Porto, 2011.

[43] František Duchoň, Andrej Babinec, Martin Kajan, Peter Beňo, Martin Florek, Tomáš Fico, and Ladislav Jurišica. Path Planning with Modified a Star Algorithm for a Mobile Robot. *Procedia Engineering*, 96:59–69, 2014.

[44] Daniel Filipe Barros Campos. Planeamento Simultâneo de Trajetórias para Múltiplos Robôs Autónomos num Ambiente Industrial. Master's thesis, Faculdade de Engenharia da Universidade do Porto, 2014.

[45] Christoph Rösmann, Wendelin Feiten, Thomas Wösch, Frank Hoffmann, and Torsten Bertram. Trajectory modification considering dynamic constraints of autonomous robots. *German Conference on Robotics (Robotik)*, pages 74–79, 2012.

[46] Christoph Rösmann, Wendelin Feiten, Thomas Wosch, Frank Hoffmann, and Torsten Bertram. Efficient trajectory optimization using a sparse model. In *2013 European Conference on Mobile Robots*, pages 138–143. IEEE, 2013.

[47] Christoph Rösmann, Frank Hoffmann, and Torsten Bertram. Planning of Multiple Robot Trajectories in Distinctive Topologies. *2015 European Conference on Mobile Robots (ECMR)*, 2015.

[48] Christoph Rösmann, Frank Hoffmann, and Torsten Bertram. Integrated online trajectory planning and optimization in distinctive topologies. *Robotics and Autonomous Systems*, 88:142–153, 2017.

[49] Abraham Sánchez L., René Zapata, and J Abraham Arenas B. Motion Planning for Car-Like Robots Using Lazy Probabilistic Roadmap Method. In Carlos A Coello Coello, Alvaro de Albornoz, Luis Enrique Sucar, and Osvaldo Cairó Battistutti, editors, *MICAI 2002: Advances in Artificial Intelligence*, pages 1–10, Berlin, Heidelberg, 2002. Springer Berlin Heidelberg.

[50] Fei Yan, Yi-Sha Liu, and Ji-Zhong Xiao. Path Planning in Complex 3D Environments Using a Probabilistic Roadmap Method. *International Journal of Automation and Computing*, 10:525–533, 2013.

[51] Gregory Dudek and Michael Jenkin. *Computational principles of mobile robotics*. Cambridge University Press, 2000.

[52] Yoram Koren and Johann Borenstein. Potential Field Methods and Their Inherent Limitations for Mobile Robot Navigation. Technical report, 1991.

[53] Muhammad Zohaib, Mustafa Pasha, Raja Ali Riaz, Nadeem Javaid, M Ilahi, and Rahim Khan. Control strategies for mobile robot with obstacle avoidance. 3:1027–1036, 2013.

[54] D K M Kufoalor, E F Brekke, and T A Johansen. Proactive Collision Avoidance for ASVs using A Dynamic Reciprocal Velocity Obstacles Method. pages 2402–2409, 2018.

[55] P. Fiorini and Z. Shiller. Motion planning in dynamic environments using the relative velocity paradigm. *Proceedings IEEE International Conference on Robotics and Automation*, pages 560–565, 1993.

[56] Jur Van Den Berg, Ming Lin, and Dinesh Manocha. Reciprocal Velocity Obstacles for Real-Time Multi-agent Navigation Reciprocal Velocity Obstacles for Real-Time Multi-Agent Navigation. In *IEEE International Conference on Robotics and Automation*, 2008.

[57] Health and Safety Authority. Hazards in Port and Dock Operations. Technical report, 2015.

[58] Nuno Alexandre Cruz, Aníbal Matos, Sérgio Cunha, and Sérgio Silva. Zarco - An Autonomous Craft for Underwater Surveys. In *Proceedings of the 7th Geomatic Week*, 2007.

[59] Ana Rita Gaspar, Alexandra Nunes, Andry Maykol Pinto, and Aníbal Matos. Urban@CRAS dataset: Benchmarking of visual odometry and SLAM techniques. *Robotics and Autonomous Systems*, 109:59–67, 2018.

[60] Miguel Pinto, António Paulo Moreira, and Aníbal Matos. Localization of Mobile Robots using an Extended Kalman Filter in a LEGO NXT. *IEEE Transactions on Education*, 55(1):135–144, 2012.

[61] Khamar Basha Shaik, P. Ganesan, V. Kalist, B. S. Sathish, and J. Merlin Mary Jenitha. Comparative Study of Skin Color Detection and Segmentation in HSV and YCbCr Color Space. *Procedia Computer Science*, 57:41–48, 2015.

[62] Amanpreet Kaur and B.V Kranthi. Comparison between YCbCr Color Space and CIELab Color Space for Skin Color Segmentation. *International Journal of Applied Information Systems (IJAIS)*, 3(4):30–33, 2012.

[63] Richard Hartley and Andrew Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2003.