

## **Sonar Based Autonomous Aquaculture Net Cage Inspection**

**Daniel Garcia Domingos Rosa**

Thesis to obtain the Master of Science Degree in  
**Electrical and Computer Engineering**

Supervisors: Prof. David Alexandre Cabecinhas  
Prof. Fausto Miguel Pascoal Ferreira

### **Examination Committee**

Chairperson: Prof. João Manuel de Freitas Xavier  
Supervisor: Prof. David Alexandre Cabecinhas  
Members of the Committee: Prof. António Pedro Rodrigues de Aguiar

**July 2023**



**Declaration**

I declare that this document is an original work of my own authorship and that it fulfils all the requirements  
of the Code of Conduct and Good Practices of the Universidade de Lisboa.

## Acknowledgements

Firstly, I would like to thank my parents, João and Susana, for all the love and education given to me throughout my life, for all the life teachings, and thank you for making me a hardworker, as said in our house: "Isto não é casa de mandriões!". I would also like to thank my brother, Fábio, for the all the patience that an older brother must have, for all the games that we played, for all the fights and for all the brother love.

Secondly, words cannot express my gratitude to my girlfriend. Mafalda. She has been a source of emotional support that kept me focused and confident in my capabilities. Sincerely, thank you!

Special thanks to my supervisors, Professor David Cabecinhas, and Professor Fausto Ferreira, for all the guidance and knowledge crucial for the completion of this thesis.

Moreover, I would like to thank the entire DSOR team and former members. Thank you João Quintas and David Souto for helping me in this new world and advise me in how to draw my path. Thank you Marcelo Jacinto, Francisco Branco, Manuel Rufino, Luís Sebastião, and Helena Santana for all the help in everything I needed to fit in the team. Lastly, a particular thanks to Eduardo Cunha that was always the first to offer me help and keep me motivated during this work, which also gave me tutorials directly from Goa.

Furthermore, I would like to thank all my friends "Amstardanenses" that I made along my journey in IST for all the laughs and unforgettable moments. Although, this journey comes to end, I hope to continue new ones with you all.

Lastly, I'm thankfull for all the people that are no longer in my life, but left a significant mark on my character. A special thanks to my highschol's Physics teacher, Catarina Caseiro.

## **Abstract**

In recent years, there has been a rising interest in aquaculture development, driven by the increasing demand for food, as a consequence of the population's growth. The aquaculture infrastructures require regular monitoring to assure their preservation and food safety for consumers. Traditionally, this monitoring is conducted by professional divers, which causes it to be a slow and risky process. To solve this issue, an autonomous aquaculture net inspection using an Autonomous Underwater Vehicle (AUV) is proposed. The AUV is equipped with a Forward-Looking Sonar (FLS), which provides acoustic images of the environment in front of the vehicle. These acoustic images are processed by image processing techniques, namely Bilateral and Binary Filters, to remove noise. Then, the vehicle's distance and orientation relative to the net cage are measured from the processed acoustic image. For measuring the relative orientation, the net cage's center is estimated, using a circle regression based on the least-squares method. After both distance and orientation are measured, velocity and orientation references are generated so the AUV can stay at a desired distance from the net cage, pointing directly to its center, while moving sideways, covering the whole net perimeter. Finally, several representative tests are performed to analyze the vehicle's behavior while encircling the net. Additionally, the controller's robustness is examined in the presence of currents.

**Keywords:** Autonomous Underwater Vehicle (AUV), Autonomous Aquaculture Net Inspection, Forward-Looking Sonar (FLS), Image Processing



## **Resumo**

Nos últimos anos, houve um grande aumento no interesse no desenvolvimento de aquacultura, causado pelo aumento da demanda de comida como consequência do aumento populacional. As infraestruturas de aquacultura necessitam de inspeções regulares, de forma a assegurar a sua preservação e a qualidade da comida. Tradicionalmente, esta inspeção é feita por mergulhadores profissionais, sendo um processo lento e perigoso. Para resolver este problema, uma inspeção autónoma de redes de aquacultura, com o uso de um Autonomous Underwater Vehicle (AUV), é proposta. O AUV é equipado com um Sonar de Varredura Frontal, que produz imagens acústicas do meio à frente do veículo. Estas imagens são processadas por um filtro bilateral e binário para remover o ruído. Depois, a distância e a orientação do veículo relativamente à estrutura é medida a partir da imagem processada. No caso da orientação, o centro da estrutura é estimado usando uma regressão circular, baseada no método dos mínimos quadrados. Com a medição da distância e da orientação relativas à estrutura de aquacultura, referências de velocidade e orientação são geradas, de modo a que o veículo se localize a uma distância desejada da estrutura, apontando diretamente para o centro da estrutura, enquanto se move lateralmente ao longo de todo o perímetro da infraestrutura de aquacultura. Finalmente, vários testes foram realizados para analisar o comportamento do veículo enquanto circula à volta da rede. Além disso, a robustez dos controladores são examinados na presença de correntes.

**Palavras-Chave:** Autonomous Underwater Vehicle (AUV), Inspeção Autónoma de Redes de Aquacultura, Sonar de Varrimento Frontal, Processamento de Imagem



# Contents

<b>Contents</b>	<b>ix</b>
<b>List of Figures</b>	<b>xii</b>
<b>List of Tables</b>	<b>xv</b>
<b>Acronyms</b>	<b>xvi</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Motivation . . . . .	1
1.2 Problem Statement . . . . .	2
1.3 Thesis Contribution . . . . .	3
1.4 Thesis Organization . . . . .	4
<b>2 State of the art</b>	<b>5</b>
2.1 Autonomous Net Cage Inspections . . . . .	5
2.2 Object Detection Based on Sonar . . . . .	7
<b>3 Vehicle Model</b>	<b>9</b>
3.1 Coordinate Frames and General Notation . . . . .	9
3.2 AUV Kinematics . . . . .	10
3.3 AUV Dynamics . . . . .	11
3.4 AUV Simplified Motion . . . . .	12
3.5 BlueROV vehicle . . . . .	13
3.6 Thruster Model . . . . .	14
3.7 System Model . . . . .	15
3.8 FAROL Stack . . . . .	16
<b>4 Sonar Image Net Detection</b>	<b>17</b>
4.1 Net Cage 3D Model . . . . .	17
4.2 Forward-Looking Multibeam Sonar . . . . .	19
4.2.1 Imaging Forward-Looking Sonar Geometry . . . . .	19
4.3 Image Processing . . . . .	21
4.3.1 Bilateral Filter . . . . .	21

4.3.2	Binary Filter . . . . .	22
4.4	Circle Regression . . . . .	23
4.4.1	Initial Condition . . . . .	25
4.4.2	Inner Points Filter . . . . .	26
4.4.3	Pole Detection . . . . .	27
4.4.4	Center Estimation Performance . . . . .	30
4.5	Distance Measurement . . . . .	31
<b>5</b>	<b>Inspection Controller</b>	<b>32</b>
5.1	Inner-loop controllers . . . . .	32
5.1.1	Yaw Controller . . . . .	32
5.1.2	Surge Controller . . . . .	33
5.1.3	Sway Controller . . . . .	34
5.1.4	Depth Controller . . . . .	34
5.2	Yaw Reference . . . . .	35
5.3	Distance Controller . . . . .	36
5.4	Sway Reference . . . . .	37
<b>6</b>	<b>Simulation and Results Discussion</b>	<b>38</b>
6.1	Implementation . . . . .	38
6.1.1	Multibeam FLS Plugin . . . . .	38
6.1.2	Integral Anti-Windup . . . . .	39
6.1.3	Yaw Reference Correction . . . . .	39
6.1.4	Yaw Reference Smoothing . . . . .	40
6.1.5	Yaw Error . . . . .	40
6.1.6	Yaw Reference Error . . . . .	41
6.1.7	Actual Distance . . . . .	41
6.1.8	Distance Correction . . . . .	41
6.2	Simulation Results . . . . .	43
6.2.1	AUV Inner-loops . . . . .	43
6.2.2	Inspection Tests . . . . .	43
6.2.3	Inspection Tests Summary . . . . .	69
<b>7</b>	<b>Conclusions and Future Work</b>	<b>72</b>
7.1	Future Work . . . . .	73
<b>Bibliography</b>		<b>75</b>
<b>A Additional Figures for test T2</b>		<b>78</b>
<b>B Additional Figures for test T3</b>		<b>79</b>

<b>C Additional Figures for test T4</b>	<b>80</b>
<b>D Additional Figures for test T5</b>	<b>81</b>

# List of Figures

1.1	Proposed conceptual approach to Autonomous Aquaculture Inspection. . . . .	3
3.1	Adopted Reference Frames, inertial $\{U\}$ and body $\{B\}$ , used like in [28] . . . . .	9
3.2	BlueROV heavy configuration 3D model used in simulation. (a) Front View. (b) Top View. . . . .	14
3.3	Inner-Outer Loop architecture. . . . .	16
4.1	Simulated Aquaculture Net Cage used in this project. This 3D model was designed by the Laboratory for Underwater Systems and Technologies (LABUST) of the Faculty of Electrical Engineering and Computing (FER) of the University of Zagreb. (a): Side view. (b) Top view. . . . .	18
4.2	Multibeam sonar used in this project. (a) Sound Metrics ARIS Explorer 3000 ([32]). (b) BlueROV with the multibeam sonar attached in the gazebo simulation. . . . .	19
4.3	Imaging Sonar Projection Model. . . . .	20
4.4	Sonar Image 10 meters away from the net cage. The supporting posts for the cage net are clearly visible in their circular configuration. The horizontal lines correspond to multiple sonar reflections at the water's surface. . . . .	21
4.5	Stages of Processing the image in Figure 4.4: (a) Gray Scale. (b) Bilateral Filter. (c) Binary Filter. . . . .	23
4.6	Circle regression for the Figure 4.5c. The green dots are the points for which the circle (in blue) was fitted. The computed center coordinates were (342.56, -39.51). . . . .	25
4.7	(a) Sonar image at 4 meters from the net. (b) Circle Regression for image (a), the blue circle is the solution before outlier detection, and the red one is after outlier detection. . . . .	26
4.8	Flowchart of the filter of the circle's inner points algorithm. . . . .	27
4.9	Flowchart of the pole detection procedure. . . . .	28
4.10	(a): Sonar image with the vehicle at 2 m away from the net. (b): Circle regression for image (a) without pole detection. (c): Circle regression for image (a) with pole detection. (d): Circle after the center's adjustment to intersect the outer poles. . . . .	29
4.11	Center Adjustment Diagram. . . . .	30
4.12	Examples of bad detections. . . . .	31
5.1	Yaw Reference Calculation Diagram. . . . .	35
6.1	Implementation Flowchart. . . . .	38

6.2	Measured Distance behavior when the vehicle is circulating the net. . . . .	42
6.3	(a): Diagram for angle $\gamma$ . (b): Variation of the error between the measured distance and the actual distance with the angle $\gamma$ . . . . .	42
6.4	Controller performance for some different given references: (a) Surge, (b) Sway. . . . .	43
6.5	Controller performance for some different given references: (a) Yaw, (b) Depth. . . . .	44
6.6	(a): Sonar image example during test T1. (b): Camera image example during test T1 with the sonar's field-of-view represented by the blue rectangle. . . . .	45
6.7	Trajectory of the vehicle on the $x$ - $y$ plane while performing test T1 at 0.2 m/s, at 2 m depth, and at 2.5 m from the net. . . . .	46
6.8	Distance behavior throughout test T1 at 0.2 m/s, at 2 m depth, and at 2.5 m from the net. . . . .	47
6.9	Surge behavior throughout test T1 at 0.2 m/s, at 2 m depth, and at 2.5 m from the net. . . . .	47
6.10	Sway behavior throughout test T1 at 0.2 m/s, at 2 m depth, and at 2.5 m from the net. . . . .	47
6.11	Yaw behavior throughout test T1 at 0.2 m/s, at 2 m depth, and at 2.5 m from the net. . . . .	48
6.12	Yaw reference error throughout test T1 at 0.2 m/s, at 2 m depth, and at 2.5 m from the net. . . . .	48
6.13	Trajectory of the vehicle on the $x$ - $y$ plane while performing test T2 at 0.2 m/s, at 5 m depth, and at 2.5 m from the net. . . . .	49
6.14	Distance behavior throughout test T2 at 0.2 m/s, at 5 m depth, and at 2.5 m from the net. . . . .	49
6.15	Surge behavior throughout test T2 at 0.2 m/s, at 5 m depth, and at 2.5 m from the net. . . . .	50
6.16	Yaw reference error throughout test T2 at 0.2 m/s, at 5 m depth, and at 2.5 m from the net. . . . .	50
6.17	Trajectory of the vehicle on the $x$ - $y$ plane while performing test T3 at 0.5 m/s, at 5 m depth, and at 2.5 m from the net. . . . .	51
6.18	Distance behavior throughout test T3 at 0.5 m/s, at 5 m depth, and at 2.5 m from the net. . . . .	51
6.19	Surge behavior throughout test T3 at 0.5 m/s, at 5 m depth, and at 2.5 m from the net. . . . .	52
6.20	Yaw reference error throughout test T3 at 0.5 m/s, at 5 m depth, and at 2.5 m from the net. . . . .	52
6.21	(a): Sonar image example during test T4. (b): Camera image example during test T4 with the sonar's field-of-view represented by the blue rectangle. . . . .	53
6.22	Trajectory of the vehicle on the $x$ - $y$ plane while performing test T4 at 0.2 m/s, at 5 m depth, and at 5 m from the net. . . . .	53
6.23	Distance behavior throughout test T4 at 0.2 m/s, at 5 m depth, and at 5 m from the net. . . . .	54
6.24	Surge behavior throughout test T4 at 0.2 m/s, at 5 m depth, and at 5 m from the net. . . . .	54
6.25	Yaw reference error throughout test T4 at 0.2 m/s, at 5 m depth, and at 5 m from the net. . . . .	54
6.26	Trajectory of the vehicle on the $x$ - $y$ plane while performing test T5 at 0.5 m/s, at 5 m depth, and at 5 m from the net. . . . .	55
6.27	Distance behavior throughout test T5 at 0.5 m/s, at 5 m depth, and at 5 m from the net. . . . .	56
6.28	Surge behavior throughout test T5 at 0.5 m/s, at 5 m depth, and at 5 m from the net. . . . .	56
6.29	Yaw reference error behavior throughout test T5 at 0.5 m/s, at 5 m depth, and at 5 m from the net. . . . .	56
6.30	Trajectory of the vehicle on the $x$ - $y$ plane while performing test T6 at 0.5 m/s, at 5 m depth, and at 5 m from the net. . . . .	57

6.31	Distance behavior throughout test T6 with a smooth approach. . . . .	58
6.32	Surge behavior throughout test T6 with a smooth approach. . . . .	58
6.33	Sway behavior throughout test T6 with a smooth approach. . . . .	59
6.34	Thrusters RPM command throughout: (a): Test T5; (b): Test T6. . . . .	59
6.35	Sway behavior throughout test T6 with a smooth approach. . . . .	60
6.36	3D trajectory of the vehicle in the full inspection test. . . . .	61
6.37	Depth behavior throughout the full inspection test. . . . .	61
6.38	Sway behavior throughout the full inspection test. . . . .	62
6.39	Yaw behavior throughout the full inspection test. . . . .	62
6.40	Distance behavior throughout the full inspection test. . . . .	63
6.41	Surge behavior throughout the full inspection test. . . . .	63
6.42	Yaw Reference Error throughout the full inspection test. . . . .	63
6.43	Trajectory of the vehicle on the $x$ - $y$ plane during test T8 in the presence of a 0.1 m/s current. . . . .	64
6.44	Current components on the body frame in Test T8. . . . .	65
6.45	Surge behavior throughout test T8 in the presence of a 0.1 m/s current. . . . .	65
6.46	Sway behavior throughout test T8 in the presence of a 0.1 m/s current. . . . .	65
6.47	Distance behavior throughout test T8 in the presence of a 0.1 m/s current. . . . .	66
6.48	Yaw behavior throughout test T8 in the presence of a 0.1 m/s current. . . . .	66
6.49	Trajectory of the vehicle on the $x$ - $y$ plane during test T9 in the presence of a 0.1 m/s current with new controllers parameters. . . . .	67
6.50	Current components on the body frame during test T9, with new parameters. . . . .	68
6.51	Surge behavior throughout test T9 in the presence of a 0.1 m/s current with new controllers parameters. . . . .	68
6.52	Sway behavior throughout test T9 in the presence of a 0.1 m/s current with new controllers parameters. . . . .	68
6.53	Distance behavior throughout test T9 in the presence of a 0.1 m/s current with new controllers parameters. . . . .	69
6.54	Yaw behavior throughout test T9 in the presence of a 0.1 m/s current with new controllers parameters. . . . .	69
A.1	Sway behavior throughout test T2 at 0.2 m/s, at 5 m depth, and at 2.5 m from the net. . . . .	78
A.2	Yaw behavior throughout test T2 at 0.2 m/s, at 5 m depth, and at 2.5 m from the net. . . . .	78
B.1	Sway behavior throughout test T3 at 0.5 m/s, at 5 m depth, and at 2.5 m from the net. . . . .	79
B.2	Yaw behavior throughout test T3 at 0.5 m/s, at 5 m depth, and at 2.5 m from the net. . . . .	79
C.1	Sway behavior throughout test T4 at 0.2 m/s, at 5 m depth, and at 5 m from the net. . . . .	80
C.2	Yaw behavior throughout test T4 at 0.2 m/s, at 5 m depth, and at 5 m from the net. . . . .	80
D.1	Sway behavior throughout test T5 at 0.5 m/s, at 5 m depth, and at 5 m from the net. . . . .	81
D.2	Yaw behavior throughout test T5 at 0.5 m/s, at 5 m depth, and at 5 m from the net. . . . .	81

# List of Tables

3.1	The SNAME notation for marine vessels . . . . .	10
3.2	BlueROV's parameters . . . . .	14
4.1	Net Cage Dimensions . . . . .	17
4.2	Parameters of the simulated Multibeam Sonar . . . . .	19
4.3	Center's Estimation Accuracy for different distances and depths. . . . .	31
6.1	Aquaculture Net Inspection Simulation Log . . . . .	44
6.2	Controllers Parameters . . . . .	45
6.3	Controllers New Parameters . . . . .	66
6.4	RMSE values for tests T1 and T6 . . . . .	70
6.5	Distance Steady-State Error for tests T1 and T6 . . . . .	70
6.6	RMSE and $K_I$ values for tests T8 and T9 . . . . .	71
6.7	RMSE for the measured and actual distance . . . . .	71

# Acronyms

AUV	Autonomous Underwater Vehicle
CG	Center of Gravity
DOF	Degrees of Freedom
DSOR	Dynamical Systems and Ocean Robotics Laboratory
FAO	The Food and Agriculture Organization of the United Nations
FLS	Forward-Looking Sonar
LARSyS	Laboratory for Robotics and Engineering Systems
LM	Levenberg-Marquardt
MSE	Mean-Squared Error
ISR	Institute for Systems and Robotics
IST	Instituto Superior Técnico
PID	Proportional-Integral-Derivative
ROS	Robotic Operating System
ROV	Remotely Operated Vehicle
RMSE	Root Mean Squared Error
SSS	Side-Scan Sonar
SAS	Synthetic Aperture Sonar
SSE	Sum Squared Error
ROI	Region of Interest
VFOV	Vertical Field-of-View

# Chapter 1

## Introduction

### 1.1 Motivation

The rise in the world population is the cause for the growing interest in new methods to increase global food production and its efficiency. According to The Food and Agriculture Organization of the United Nations (FAO) [1], there is a growth in the consumption of seafood in recent years. The consumption of aquatic food grew from an average of 9.9 kg *per capita* in the 1960s to a record high of 20.5 kg in 2019. Furthermore, the predictions estimate that, on average, the consumption will escalate to 21.4 kg *per capita* in 2030.

On top of that, fishery resources are continuing to decline due to overfishing and poor management. Overfishing is also closely tied with bycatch, that is, the capture of unwanted species, since fishing nowadays is predominated by the use of trawls. These bycatch can be dolphins or even juvenile fish, affecting the species' reproduction and leaving serious repercussions on marine ecosystems.

As a result of population growth and the rising demand for seafood, the need for further aquaculture development, both inshore and offshore, is becoming increasingly urgent in the upcoming years. Aquaculture provides a stable source of food, particularly in areas where wild-caught seafood is scarce or overfished, and it can generate economic opportunities in rural and coastal communities. Moreover, it can be implemented in such a way that minimizes negative impacts on the environment, such as overfishing or habitat destruction. Furthermore, the damaged ecosystems can be restored and it can provide habitat for endangered wild species.

Today, according to FAO, more than half of the fish consumed globally is produced in aquaculture facilities. These infrastructures require regular monitoring to assure their preservation and food safety for the consumers. Net inspection is a key procedure of the aquaculture monitoring process and looks for issues, such as biofouling or net holes. However, there are numerous problems regarding the inspection of the aquaculture net. Traditionally, this work is made manually by professional divers. As a consequence, manual inspections can be risky and time-consuming, because the divers can only be underwater for up to five minutes at a time. Furthermore, globally there is a shortage of divers, for example, in Australia [2] and in the USA [3].

More recently, other methods of net inspection have been explored, namely resorting to the large

industrial Remotely Operated Vehicles (ROVs), which allow for an extended dive time and are safe for the user. Despite this, large industrial ROVs are very expensive and require extensive training to operate. Moreover, these vehicles have low-quality setting recordings and are not suitable for inspecting compact areas, since they are attached to a ship by a tether cable and can get entangled in the nets. For those reasons, the identification of net issues can be challenging, tedious, and slow. In addition, since the vehicle is operated by a human, inefficiencies and human-driven errors can occur.

Autonomous Underwater Vehicles (AUVs) are another alternative for net inspection. As ROVs, AUVs allow for a safer inspection, but without the risk of human error. Moreover, AUVs do not need a connected tether or constant operator supervision, enabling them to access a wider range of areas and save valuable time for operators.

Motivated by the above considerations, this dissertation proposes an autonomous method for the inspection of aquaculture infrastructures, in which an AUV is used to move around the net cage for net inspection. Furthermore, as an innovative approach, the AUV is equipped with a Forward-Looking Sonar (FLS) that provides real-time information about the vehicle's relative position to the net, allowing it to navigate in the net's perimeter.

The work in this thesis was developed at Dynamical Systems and Ocean Robotics Laboratory (DSOR) of the Institute of Systems and Robotics (ISR) at Instituto Superior Técnico (IST), in Lisbon.

## 1.2 Problem Statement

The above motivation clearly shows the advantages of autonomous inspection of aquaculture facilities. The main goal of this thesis is to use an AUV to locate the net cage and move around it. Figure 1.1 displays a scheme of the entire project pipeline. Firstly, the vehicle detects the infrastructure net ( $t_0$ ). After that, it steers so it is pointing directly to the center of the net ( $t_1$ ). Then, it moves in the direction of the net and stops at a desired distance ( $t_2$ ). Finally, it starts to move around the aquaculture net, while inspecting it ( $t_3$ ).

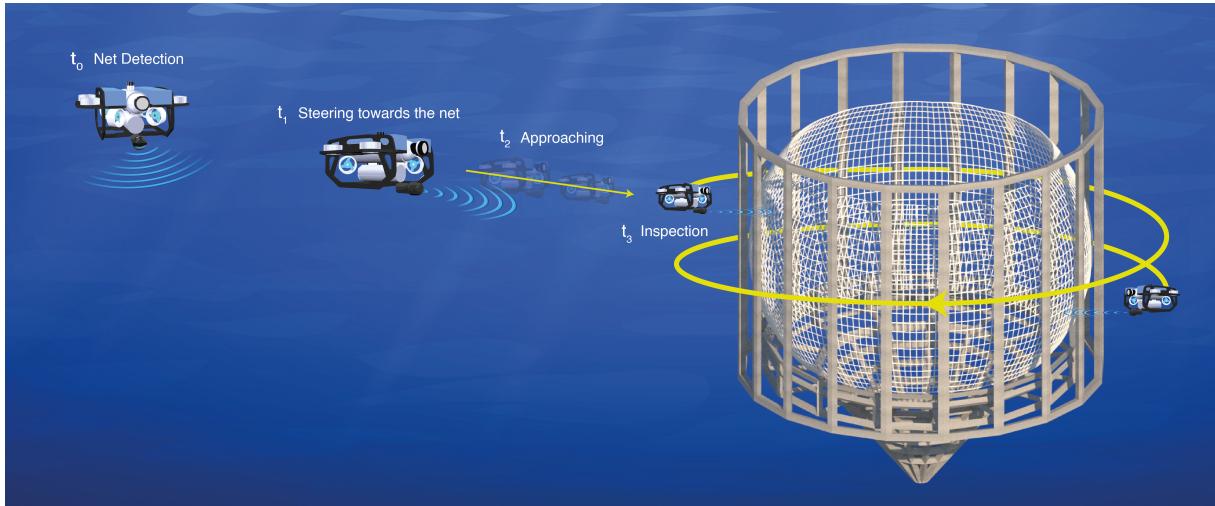
The experimental setup includes a BlueROV, manufactured by BlueRobotics, which is equipped with a variety of sensors for measuring the vehicle's orientation, velocity, and depth. The net cage used in this study consists of multiple poles arranged in a cylindrical structure, housing a spherical-shaped net inside.

Traditionally, cameras are used, not only for the net's inspection, but also to give the relative position of the vehicle regarding the net. However, in underwater environments, cameras can have visibility issues due to, mainly, low luminosity, water flow, and suspended particles.

Considering these problems, a solution is proposed where the vehicle is equipped with an FLS to measure its relative position to the net. Unlike cameras, sonars are preferred in this scenario due to their ability to operate effectively underwater without visibility issues. Moreover, sonars offer a larger detection range compared to cameras although with lower resolution.

Knowing that the FLS provides acoustic images of the environment in front of the vehicle, by employing image processing techniques, the net cage can be detected. Following the detection, the AUV's relative position to the net can be computed. The data regarding the relative position can then be used to control

the vehicle to be close and oriented toward the net. This approach ensures a reliable and accurate localization of the net cage, enabling a precise control of the vehicle around the net.



**Figure 1.1:** Proposed conceptual approach to Autonomous Aquaculture Inspection.

### 1.3 Thesis Contribution

The main goal of this work is to design an autonomous procedure for the inspection of aquaculture infrastructures. For the navigation of the vehicle relative to the aquaculture net, a Forward-Looking Sonar is employed, providing acoustic images of the AUV's environment. These images are then processed for the detection of the infrastructure. With the location of the aquaculture net cage in the sonar image, and knowing its shape, it is possible to obtain measurements of the distance and orientation of the vehicle relative to the net. Then, with these measurements, the vehicle can be steered toward the net center, allowing it to approach the net until a desired distance, and circle the net cage for inspection purposes.

The work done addresses a diverse and yet complementary number of sub-problems, the most relevant of which are summarized below:

- **Vehicle Model:** a model that describes the motion of the AUV is shown as a set of mathematical equations.
- **Sonar Image Processing and Net Detection:** the acoustic image provided by the FLS is processed by applying bilateral and binary filters. Afterward, a circle regression is used to find the net cage's center giving the location of the structure.
- **Vehicle Control:** design controllers for the vehicle to follow desired velocities and orientations.
- **Vehicle Guidance:** design a set of laws to provide references for the vehicle to follow in order for it to circle the net.
- **Simulation tests:** carry out missions with the BlueROV vehicle, in a simulated 3D environment, to corroborate the proposed method.

## 1.4 Thesis Organization

This thesis is formulated in order for the reader to be led through the problems that arise from the main objective and then the proposed solutions. It is organized in the following way:

- **Chapter 2 - State of the Art:** makes a brief overview of the contributions in autonomous aquaculture inspections, such as vehicle navigation and control. This chapter also gives insights on tools for object detection in sonar images;
- **Chapter 3 - AUV Model:** introduces the SNAME notation used throughout this thesis and derives the equations of the kinematics and dynamics of the vehicle;
- **Chapter 4 - Sonar Image Net Detection:** reports on the net cage used in the simulation and details an image processing procedure to identify the structure on the sonar images;
- **Chapter 5 - Inspection Motion Control:** presents the controllers designed for the vehicle to reach desired velocities and orientations. Furthermore, it shows the calculations for the vehicle's references of velocity and orientation;
- **Chapter 6 - Simulation and Results:** depicts the simulation conditions as well as the obtained results;
- **Chapter 7 - Conclusions and Future Work:** draws conclusions from the obtained results and discusses the future work to be done.

# Chapter 2

## State of the art

In this chapter, the state of the art of autonomous inspection aquaculture methods and object detection out of sonar images.

### 2.1 Autonomous Net Cage Inspections

In recent years, there has been a significant increase in research focused on autonomous inspections of fish net pens. As a result, new techniques and methods utilizing ROVs/AUVs are being developed. The current studies in this field encompass various aspects, including net damage detection, vehicle control, and navigation during net inspections.

The net damage detection methods address hole detection, biofouling detection, deformation detection, etc. In [4], the authors describe a hole detection method. The image's region of interest (ROI) is processed by a bilateral filter [5] and then by an inter-class variance method. After that, a feature gradient histogram is computed corresponding to the mesh hole area. With this histogram, the local maximums are searched to find the damage's position. Additionally, in [6], it is proposed another hole detection method using image processing techniques. In this work, a combination of Hough Transform and statistical analysis methods were used to perform a local and global search for hole detection in the mesh. However, the method still needs to be tested on a real vehicle in real-time systems to verify its relevance since it was only tested on offline images.

Biau Su *et al.* [7] propose a numerical model for the monitoring of net cage deformation. Acoustic sensors are used to get positioning data for different current magnitudes and directions, used for the modeling. With this approach, the currents can be estimated from the deformation of the net cage.

In [8], a small-sized AUV is designed for fish-net cage inspection to find biofouling and net holes, while navigating autonomously. Furthermore, the distance to the net cage is estimated using the triangulation method based on the target detection in the camera image. The proposed work was tested in a real environment. However, the vehicle's movement was limited to only forward or backward based on distance estimation, and it lacked implementation of top-down movement control.

Aquaculture inspections based on ROVs/AUVs face localization challenges due to the limitations of GPS, which does not function underwater. With that in mind, the authors of [9] proposed a design of an

Omnidirectional Surface Vehicle (OSV) for fish net cage inspection. The choice of an OSV was driven by its ease of deployment and communication capabilities, as it operates at the water's surface. The authors designed a retractable camera that is adjustable by depth, so the OSV can capture pictures of the fish cage at different depths. Moreover, it was designed a small neural network, Tiny YOLO, for net hole detection. The same authors, in [10], developed an algorithm based on a hierarchical task network (HTN) for the vehicle to search for new fishnet holes promptly.

As in the previous method, Herman B. Amundsen *et al.* [11] suggests another approach to solve the localization underwater issue. The authors propose using a Doppler Velocity Log (DVL) directed to the net to approximate the geometry of the local region of the net pen. After that, the pose of the vehicle relative to the net is computed and then used in a guidance control law inspired by the Line-of-Sight (LOS) method. However, the velocity measurements obtained from the DVL require noise-handling techniques to improve the accuracy of the results.

In [12], a navigation scheme for AUVs utilizing camera-based techniques was proposed. For positioning, an object is attached to the net, and, using computer vision methods, the target is located in the image. Then, the orientation and distance of the AUV relative to the net are computed resorting to the location of the target. Finally, the vehicle adjusts its position and orientation if necessary. However, the bottleneck of this method is the need of placing beforehand of multiple targets in the net.

Magnus Bjerkeng *et al.* [13] propose a laser-camera triangulation method, consisting of two laser lines and one camera for pose estimation from one image. This work showed better results than the DVL method, but it only proposes solutions to the navigation and net tracking problems, not taking into account the control problem.

Christian Schellewald *et al.* [14] propose a vision-based solution regarding the pose estimation for autonomous operations in aquacultural fish farms. The fishnet is detected by searching for regular peaks in the Fast Fourier Transform of a camera video stream's ROI. Once the fishnet is identified, a single mesh is reconstructed from the regular peaks of the Fourier Transform. Then, the net's distance and orientation relative to the camera are computed knowing the real size of the mesh and the camera parameters. [15] describes yet a different method for pose estimation using a camera. In this work, a junction detector is proposed to detect the fishnet knots in the camera image. The knots and the parameters of the camera are used to estimate the pose of the net relative to the camera position.

In [16], the authors propose two methods for the vehicle's relative position to the net estimation. The first gets uses two cameras to collect image sequences. An ROI is selected in the images, and a "Canny" edge detector algorithm is applied. After that, a method based on SURF and K-Nearest Neighbor [17] algorithms is used for feature matching in the corresponding left and right cameras. Finally, triangulation is performed to compute the relative pose to the net. In the second method, the objective is to identify the two parallel ropes that support the net. The designed solution is handled by a single camera. The main idea is to use a Bilateral Filter, to reduce the effect of the low visibility in the image, and a "Canny" Edge Detector Algorithm, to highlight the ropes. Later, a Hough Lines transform is executed to identify the parallel lines. With the parallel lines identified, the distance to the net is computed using the camera's intrinsic parameters. An algorithm to control the vehicle's distance to the net is additionally designed.

In recent years, there has been a growing interest in using the Internet of Things (IoT) to continuously monitor aquaculture cages in real time. Chung-Chang Cheng *et al.* [18] developed an IoT-based smart fish net cage system. The system accommodates a cloud system, big data analysis, sensors, communication technology, and deep learning models for automatic feeding. This proposed architecture focuses on real-time monitoring of fish health, survival rate, and food residuals. The main emphasis of this work is on data collection and processing, rather than vehicle control and guidance.

Finally, the HEKTOR (Heterogeneous Autonomous Robotic System in Viticulture and Mariculture) project is being developed, carrying out many types of missions involving mariculture and viticulture [19]. In the viticulture scenario, an UAV (Unmanned Aerial Vehicle) is used for vineyard surveillance with the help of an all-terrain mobile manipulator. Regarding net cage inspection, in mariculture, the project aims to use a UAV for aerial inspection, an ASV (Autonomous Surface Vehicle), and an ROV for underwater inspection. The end goal of this project is to autonomously coordinate the UAV, ASV, and ROV for the inspection of the net cage. Goran Borković *et al.* [20] propose an autonomous scheme for fish cage inspection with an ROV. In this work, the ROV performed image processing of the net to detect two parallel ropes that are considered targets. After the detection, velocity references are computed so the ROV can scan the net. While the vehicle is scanning the net, a deep neural network, pre-trained on a dataset of real-life conditions from the CROMARIS fish farm, is responsible for the detection of biofouling.

This study introduces a novel approach to autonomous aquaculture net inspection, wherein navigation around the net cage is accomplished using a Forward-Looking Sonar instead of the conventional camera-based methods. By doing so, we can overcome the visibility challenges associated with underwater cameras.

## 2.2 Object Detection Based on Sonar

Sonar is a well-known acoustic tool that can provide images of underwater environments, even in null visibility conditions. This property is crucial for underwater object detection because regular cameras are impractical in low-visibility conditions.

There are three types of sonar technology providing image-link data: Side-Scan Sonar (SSS), Synthetic Aperture Sonar (SAS), and Forward-Looking Sonar (FLS). The first two are ideal to survey vast areas, owing to the long range of the signals. Both technologies excel at mapping and exploring underwater landscapes as well as detecting objects at significant distances. On the other hand, the FLS allows a more detailed inspection of nearer objects. FLS's higher resolution and real-time imaging afford a more detailed view of close objects. This real-time imaging capacity reveals to be crucial for decision-making during inspections in dynamic environments.

Recent advances in computer vision methods, particularly in deep neural networks, have led to a variety of proposals for many object detection methods using SSS and SAS images. Gerald J Dobeck *et al.* [21] developed a method for sea mines detection, using SSS images and a neural-network classifier based on the k-nearest neighbor method. In [22], a novel method is proposed to identify sea mine echoes in SAS images. In this method, the image segmentation of the echoes is performed by a threshold

criterion. This threshold criterion is set using the mean and standard deviation as well as a statistical model of the data (the Weibull law was shown to be the best fit for the data).

Despite that, if a closer and more detailed inspection is desired, both SSS and SAS are not fit for that task. Therefore, the use of an FLS is preferred. Lu and Sang [23] propose image processing techniques for target and shape recognition in a sequence of sonar images. In [24], continuous image sequences generated by an electronic scanning sonar are utilized to enable obstacle avoidance and visual navigation for an AUV. Additionally, a track-before-detect strategy and a dynamic programming algorithm are employed to estimate AUV dynamics and solve the detection problem with minimal computational costs.

Petillot *et al.* [25] describes a framework for the segmentation of sonar images, tracking of underwater objects, and motion estimation. Using a multi-beam FLS, an object avoidance and path planning procedure is created within this framework. The real-time data flow acquired with multi-beam FLS is initially segmented and the relevant features are subsequently extracted.

In [26], Enric Galceran *et al.* propose an algorithm for the detection of man-made objects in forward-looking sonar imagery. A Region of Interest (ROI) has to be selected to avoid certain noisy areas of the sonar image and to accelerate the detection by not processing the entire image. Next, utilizing the integral image of the selected ROI, two key components are constructed: the background, which represents the reverberation from the seabed, and the echo map, which identifies areas with high-intensity echo returns. Lastly, when the echo map has a greater value than the background map, a potential object is determined. Using *a priori* knowledge of the object the potential alarm will be filtered according to its geometry. This method is fast and object detection in real-time is possible for sonar data onboard an AUV. Thereafter, Fausto Ferreira *et al.* [27] designed a mosaicking algorithm method for FLS sonar images to improve the previous algorithm results. This approach can reduce the noise level and give a better contrast to the images to be processed. Additionally, creating mosaic images has proven to be helpful during post-processing and data analysis.

A method for detecting the structure of aquaculture nets, particularly targeting the identification of net cage poles, in FLS images is proposed.

# Chapter 3

## Vehicle Model

In this chapter, the general notation and coordinates frames are described. Additionally, the vehicle kinematics and dynamics for six Degrees-of-Freedom (DOF) are also detailed. A simplified vehicle model for 3 DOF is also characterized. Finally, the structure of the FAROL stack used in this thesis is introduced.

### 3.1 Coordinate Frames and General Notation

For modelling the AUV, two reference frames are adopted. Following the notation of Fossen et al. [28], we define an inertial frame  $\{U\}$ , composed by the orthonormal axes  $\{x_U, y_U, z_U\}$ , and a body-fixed frame  $\{B\}$ , whose origin coincides with the center of gravity (CG) of the vehicle, composed by the orthonormal axes  $\{x_B, y_B, z_B\}$ . For the Inertial Reference Frame,  $\{U\}$ , the North-East-Down Reference Frame (NED) convention is used, where the  $x_U$  axis is directed North, the  $y_U$  axis directed East and the  $z_U$  axis directed downwards. Now, for the body frame  $\{B\}$ , the  $x_B$  axis is the longitudinal axis, directed from aft to fore,  $y_B$  is the transverse axis, directed from port to starboard, and  $z_B$  is the normal axis, directed from top to bottom. Marine craft experience motion in all 6 DOF, that is, motion in all three axes and rotation around them, as shown in Figure 3.1. The six motion components are defined as *surge*, *sway*, *heave*, *roll*, *pitch* and *yaw*.

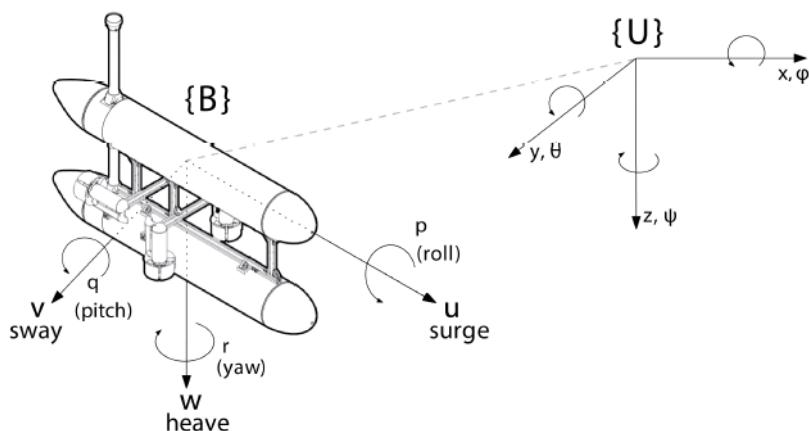


Figure 3.1: Adopted Reference Frames, inertial  $\{U\}$  and body  $\{B\}$ , used like in [28]

Furthermore, the SNAME nomenclature will be used to represent the vehicle's position, speed, torque and forces:

- $\eta_1 = [x, y, z]^T$  - position of the origin of  $\{B\}$  expressed in  $\{U\}$ ;
- $\eta_2 = [\phi, \theta, \psi]^T$  - orientation of the origin of  $\{B\}$  expressed in  $\{U\}$ ;
- $v_1 = [u, v, w]^T$  - linear velocity of the origin of  $\{B\}$  relative to  $\{U\}$ , expressed in  $\{B\}$ ;
- $v_2 = [p, q, r]^T$  - angular velocity of  $\{B\}$  relative to  $\{U\}$ , expressed in  $\{B\}$ ;
- $\tau_1 = [X, Y, Z]^T$  - actuating forces expressed in  $\{B\}$ ;
- $\tau_2 = [K, M, N]^T$  - actuating moments expressed in  $\{B\}$ .

In compact form, these entities can be written to obtain the augmented state vectors  $\eta = [\eta_1^T, \eta_2^T]^T$ ,  $v = [v_1^T, v_2^T]^T$  and forces  $\tau = [\tau_1^T, \tau_2^T]^T$ .

**Table 3.1:** The SNAME notation for marine vessels.

	Forces and Moments	Linear and Angular velocities	Positions and Euler Angles
Motions in the x direction (surge)	$X$	$u$	$x$
Motions in the y direction (sway)	$Y$	$v$	$y$
Motions in the z direction (heave)	$Z$	$w$	$z$
Rotation about x axis (roll)	$K$	$p$	$\phi$
Rotation about y axis (pitch)	$M$	$q$	$\theta$
Rotation about z axis (yaw)	$N$	$r$	$\psi$

## 3.2 AUV Kinematics

The AUV Kinematics describe the geometrical transformations from the body frame to the inertial frame or vice-versa. The analysis of the general AUV model follows the description of Fossen et al. in [28].

Using the notation described in Section 3.1, the kinematic equations are characterized by

$$\dot{\eta} = J(\eta)v, \quad J(\eta) = \begin{bmatrix} {}^U_B R(\eta_2) & 0_{3 \times 3} \\ 0_{3 \times 3} & T(\eta_2) \end{bmatrix}, \quad (3.1)$$

where  ${}^U_B R(\eta_2)$  is the rotation matrix from the body frame to the inertial frame. In this matrix, the  $z$ - $x$ - $y$  convention is used,

$${}^U_B R(\eta_2) = R_{z,\psi} R_{y,\theta} R_{x,\phi}, \quad (3.2)$$

with the rotation matrices for each axis such as

$$R_{z,\psi} = \begin{bmatrix} c\psi & -s\psi & 0 \\ -s\psi & c\psi & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad R_{y,\theta} = \begin{bmatrix} c\theta & 0 & s\theta \\ 0 & 1 & 0 \\ -s\theta & 0 & c\theta \end{bmatrix}, \quad R_{x,\phi} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & c\phi & -s\phi \\ 0 & s\phi & c\phi \end{bmatrix}, \quad (3.3)$$

so the  ${}^U_B R(\eta_2)$  is given by

$${}^U_B R(\eta_2) = \begin{bmatrix} c\psi c\theta & -s\psi c\phi + c\psi s\theta s\phi & s\psi s\phi + c\psi c\phi s\theta \\ s\psi c\theta + c\psi c\phi + s\psi s\theta s\phi & -c\psi s\phi + s\psi c\phi s\theta & -c\psi s\phi + s\psi c\phi s\theta \\ -s\theta & c\theta s\phi & c\theta c\phi \end{bmatrix}, \quad (3.4)$$

with  $c(\cdot)$  and  $s(\cdot)$  used as short-hand for the functions  $\cos(\cdot)$  and  $\sin(\cdot)$ , respectively.  $T(\eta_2)$  is an angular velocity transformation that represents the relation between  $[p, q, r]^T$  and  $[\dot{\phi}, \dot{\theta}, \dot{\psi}]^T$ , i.e, represents the transformation of the angular velocity expressed in  $\{B\}$  and the rate of Euler angles, defined as

$$T(\eta_2) = \begin{bmatrix} 1 & s\phi t\theta & c\phi t\theta \\ 0 & c\phi & -s\phi \\ 0 & s\phi/c\theta & c\phi/c\theta \end{bmatrix}, \quad (3.5)$$

with  $t(\cdot)$  representing  $\tan(\cdot)$ . One should notice that  $T(\eta_2)$  is undefined for  $\theta = \pm\frac{\pi}{2}$ . Nevertheless, this will not be a issue in this study, because the AUV will be only operated near  $\theta = 0$ . Hence, this constraint is not an issue for the inspection work considered in this thesis..

### 3.3 AUV Dynamics

As suggested in [28], the rigid-body dynamics of a generic rigid body can be described, in  $\{B\}$ , as

$$M_{RB}\dot{v} + C_{RB}(v)v = \tau_{RB}, \quad (3.6)$$

where  $M_{RB}$  is the rigid-body inertia matrix,  $C_{RB}$  the Coriolis and centrifugal terms and  $\tau_{RB}$  the external forces and moments applied.

The general vector of external forces and moments can be decomposed in several terms such as

$$\tau_{RB} = \tau + \tau_A + \tau_D + \tau_R + \tau_{dist}, \quad (3.7)$$

with

- $\tau$ : Vector of external forces and moments applied in the rigid-body from the thrusters or surfaces
- $\tau_A$ : Terms due to added masses

$$\tau_A = -M_A\dot{v} - C_A(v)v \quad (3.8)$$

- $\tau_D$ : Hydrodynamics terms due to lift, drag, skin friction, etc

$$\tau_D = -D(v)v \quad (3.9)$$

- $\tau_R$ : Restoring forces and torques due to gravity and buoyancy forces

$$\tau_R = -g(\eta) \quad (3.10)$$

- $\tau_{dist}$ : Vector represents external disturbances, for example currents, wind, waves, etc.

Now, replacing equations (3.6), (3.8), (3.9) and (3.10) in (3.7), the dynamics equation in vectorial form can be expressed as

$$\underbrace{M_{RB}\dot{v} + C_{RB}(v)v}_{\text{rigid-body forces}} + \underbrace{M_A\dot{v} + C_A(v)v + D(v)v}_{\text{hydrodynamic forces}} + \underbrace{g(\eta)}_{\text{hydrostatic forces}} = \tau + \tau_{dist}. \quad (3.11)$$

## 3.4 AUV Simplified Motion

To simplify the AUV motion, throughout this work it is assumed that the vehicle will move in the horizontal plane, therefore,  $\theta = 0$ ,  $\phi = 0$  and  $z = \text{constant}$ . Furthermore, it is also assumed that the vehicle is neutrally buoyant, i.e.,  $g(\eta) = 0$ . Now, the vehicle has only 3 DOF with  $[u, v, \psi]^T$ . Thus, the kinematics equations of motion can be given by

$$\begin{cases} \dot{x} = u \cos \psi - v \sin \psi \\ \dot{y} = u \sin \psi + v \cos \psi \\ \dot{\psi} = r \end{cases} \quad (3.12)$$

Furthermore, considering only forces in the x and y axes direction and the torque around the z-axis, then  $\tau = [\tau_u, \tau_v, \tau_r]$ . Hence, the dynamics equations in (3.11) can be described as

$$\begin{cases} m_u \ddot{u} - m_v v r + d_u u = \tau_u \\ m_v \ddot{v} + m_u u r + d_v v = \tau_v \\ m_r \dot{r} - m_{uv} u v + d_r r = \tau_r \end{cases} \quad (3.13)$$

It should be noted that not all vehicles have the capability to exert force  $\tau_v$ . However, in this work, a fully-actuated vehicle was chosen, which means it has control over the 3 DOF. This choice allows for greater freedom of movement, which is essential for addressing inspection-related challenges. The

coefficients used in (3.13) are computed such as

$$\begin{aligned} m_u &= m - X_{\dot{u}} & d_u &= -X_u - X_{|u|u}|u|, \\ m_v &= m - Y_{\dot{v}} & d_v &= -Y_v - Y_{|v|v}|v|, \\ m_r &= I_z - N_{\dot{r}} & d_r &= -N_r - N_{|r|r}|r|, \\ m_{uv} &= m_u - m_v \end{aligned}$$

where  $X_{\dot{u}}$ ,  $Y_{\dot{v}}$ , and  $N_{\dot{r}}$  represent the hydrodynamic added mass coefficients. Additionally, the terms  $X_u$ ,  $Y_v$ , and  $N_r$  correspond to the linear damping effects, while  $N_{|r|r}$  represents the axial drag. Moreover, the parameters  $X_{|u|u}$  and  $Y_{|v|v}$  are associated with the crossflow drag.

Considering ocean currents, some changes to the previous equations have to be made. For a constant irrotational ocean current  $\mathbf{v}_c$ , forming an angle  $\phi_c$  with respect to the inertial frame, the kinematics become  $u = u_r + u_c$ , where  $u_r$  and  $v_r$  are the components of the vehicle velocity with respect to the fluid and  $u_c$  and  $v_c$  are the components of the ocean current velocity in the body-fixed frame. The dynamics equations described in (3.13) must have some adjustments, being now defined by

$$\begin{cases} m_u \ddot{u}_r - m_v v_r r + d_u u_r = \tau_u \\ m_v \ddot{v}_r + m_u u_r r + d_v v_r = \tau_v \\ m_r \dot{r} - m_{uv} u_r v_r + d_r r = \tau_r \end{cases}, \quad (3.14)$$

where, in this case,  $d_u = -X_u - X_{|u|u}|u_r|$  and  $d_v = -Y_v - Y_{|v|v}|v_r|$ .

Finally, in this thesis, when the vehicle moves upwards or downwards, it does not move in any other direction, that is,  $u = v = r = 0$ . With that in mind, the movement in the  $z$ -axis can be decoupled from the horizontal movement. Therefore, the dynamics equation in the vertical axis is described as

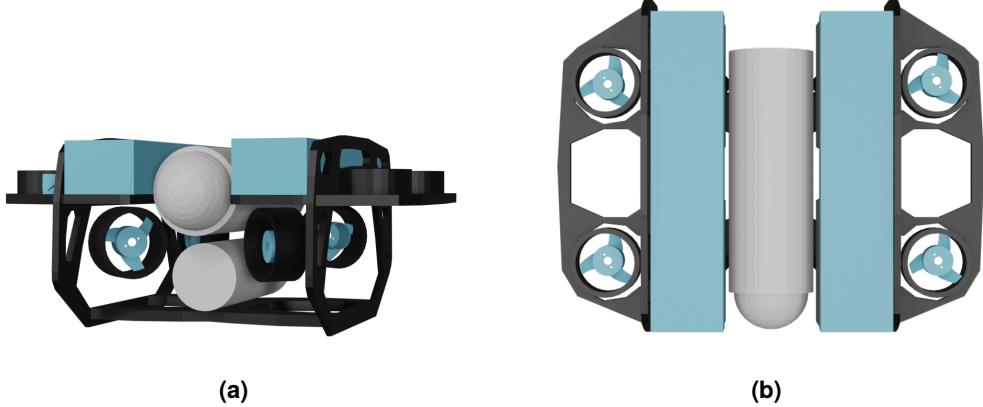
$$m_w \dot{w} + d_w w = \tau_w, \quad (3.15)$$

with  $m_w = m - Z_w$  and  $d_w = -Z_w - Z_{|w|w}|w|$ .

## 3.5 BlueROV vehicle

The BlueROV vehicle is an ROV manufactured by BlueRobotics, a marine robotics company. There are two configurations of this vehicle: the regular one (with 6 thrusters) and the heavy one (with 8 thrusters). The only difference is the number of thrusters on the vertical axis (two and four, respectively). The configuration used in this project is the heavy configuration.

This vehicle has four thrusters on the vertical axis and another four on the horizontal plane, distributed in a 45 degrees configuration, as seen in Figure 3.2. Furthermore, the vehicle possesses two horizontal tubes containing all the electronic devices and batteries, isolating them from the water. Lastly, the hydrodynamic parameters of the vehicle, as determined by DSOR's research team, are presented in Table



**Figure 3.2:** BlueROV heavy configuration 3D model used in simulation. (a) Front View. (b) Top View.

3.2.

**Table 3.2:** BlueROV's parameters

$X_{\dot{u}}$	-27.08 kg	$X_u$	-1.17 kg/s	$X_{ u u}$	-46.27 kg/m
$Y_{\dot{v}}$	-25.95 kg	$Y_v$	-1.17 kg/s	$Y_{ v v}$	-46.27 kg/m
$Z_{\dot{w}}$	-29.91 kg	$Z_w$	-1.11 kg/s	$Z_{ w w}$	-50.28 kg/m
$N_{\dot{r}}$	-1.0 kg.m <sup>2</sup>	$N_r$	-0.5 kg.m/s	$N_{ r r}$	-1.0kg.m

The inertia about the z-axis and the vehicle's mass were also measured, respectively, as  $I_z = 0.245 \text{ kg} \cdot \text{m}^2$  and  $m = 11.5 \text{ kg}$ .

## 3.6 Thruster Model

The actuators, in the BlueRov case, thrusters, are devices responsible for generating forces on the vehicle. Joining up all the generated forces at each thruster gives the overall force applied to the body of the vehicle. Furthermore, knowing the position and orientation of the thrusters relative to the vehicle's center of mass, the overall torque can also be calculated.

The BlueROV vehicle used in this work is an overactuated vehicle, that is, the total number of actuators is more than the number of DOFs needed to reach the control objective. Let  $\tau_i \in \mathbb{R}^n$  be the vector of forces and torques applied to the vehicle by the thruster  $i$  and  $f_i = [F_x, F_y, F_z]^T$  being the vector of forces produced by the thruster  $i$  along the translational axes. Both can be related by

$$\tau_i = \begin{bmatrix} \mathbf{f} \\ I_i \times \mathbf{f}_i \end{bmatrix} = \begin{bmatrix} F_x \\ F_y \\ F_z \\ F_z I_{iy} - F_y I_{iz} \\ F_x I_{iz} - F_z I_{ix} \\ F_y I_{ix} - F_x I_{iy} \end{bmatrix}_i, \quad (3.16)$$

where  $I_i = [I_{ix}, I_{iy}, I_{iz}]^T$  are the moment arms of the thruster  $i$  relative to the center of mass.

For vehicles with several thrusters, a more convenient notation is utilized. The vector of forces applied to the body of the vehicle  $\tau \in \mathbb{R}^n$  is given, in matrix notation, by

$$\tau = T\mathbf{f}, \quad (3.17)$$

where  $\mathbf{f} \in \mathbb{R}^k$  is the vector of forces produced by all the thrusters and  $T \in \mathbb{R}^{n \times k}$  is the thruster configuration matrix (in this case  $n = 6$  and  $k = 8$ ).

### 3.7 System Model

Taking into account the dynamics equations given in Equations (3.14) and (3.15), the vehicle model can be represented by

$$\left\{ \begin{array}{l} \dot{u}_r = \frac{1}{m_u} (\tau_u + m_v v_r r - d_u u_r) \\ \dot{v}_r = \frac{1}{m_v} (\tau_v + m_u u_r r - d_u v_r) \\ \dot{r} = \frac{1}{m_r} (\tau_r + m_{uv} u_r v_r - d_u r) \\ \dot{\psi} = r \\ \dot{w} = \frac{1}{m_w} (\tau_w - d_w w) \\ \dot{z} = w \end{array} \right. . \quad (3.18)$$

Since the obtained model is not linear, one can not employ linear control tools. So, one needs to linearize the model around the equilibrium point  $u_{eq} = [u_{r0}, v_{r0}, r_0, \psi_0, w_0, z_0] \in \mathbb{R}^6$ , the succeeding the model is obtained

$$\left\{ \begin{array}{l} \dot{u}_r = \frac{1}{m_u} (\tau_u + m_v v_{r0} r + m_v v_r r_0 - d_u u_r) \\ \dot{v}_r = \frac{1}{m_v} (\tau_v + m_u u_{r0} r + m_u u_r r_0 - d_u v_r) \\ \dot{r} = \frac{1}{m_r} (\tau_r + m_{uv} u_{r0} v_r + m_{uv} u_r v_{r0} - d_u r) \\ \dot{\psi} = r \\ \dot{w} = \frac{1}{m_w} (\tau_w - d_w w) \\ \dot{z} = w \end{array} \right. . \quad (3.19)$$

In the scenarios taken into account, the vehicle will circle the net pointing towards the net cage's center, while moving sideways. Therefore, the yaw rate  $r$  and sway velocity  $v$  values will be significant. However, the surge velocity will be minimal, having values of approximately zero. With this in mind, the cross terms in the sway  $v$  and yaw rate  $r$  can be neglected and assumed to be an internal disturbance attenuated by

the control loop. So, the equations can be rewritten in

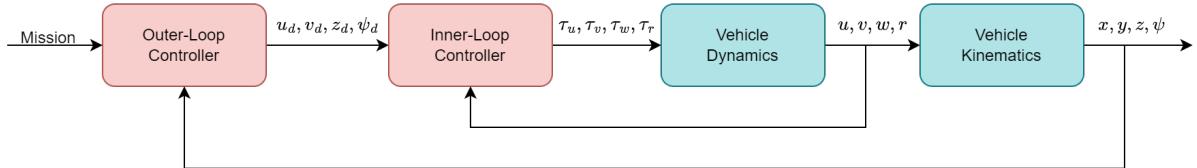
$$\begin{cases} \dot{u}_r = \frac{1}{m_u} (\tau_u + m_v v_r r - d_u u_r) \\ \dot{v}_r = \frac{1}{m_v} (\tau_v - d_u v_r) \\ \dot{r} = \frac{1}{m_r} (\tau_r - d_u r) \\ \dot{\psi} = r \\ \dot{w} = \frac{1}{m_w} (\tau_w - d_w w) \\ \dot{z} = w \end{cases} . \quad (3.20)$$

## 3.8 FAROL Stack

The FAROL stack is a group of Robot Operating System (ROS) packages built in C++ and Python created by the research team at DSOR, in ISR/IST, and by other robotics teams and companies. This stack aims to simulate the underwater environment and operate vehicles with great similarity to the real world, using the GAZEBO simulator. This is possible by taking into account the physical models of each vehicle and its behavior underwater. Furthermore, there are available many sensor models for GAZEBO, namely cameras, sonars, IMUs, AHRS, etc, which can be incorporated into the vehicles with ease. The stack is public and accessible for the scientific community in [29].

In addition, since the stack is built in ROS, one can use it for simulation but also to incorporate it in real vehicles. It is designed taking into account all the sensors and thrusters of the real vehicles, so that the simulation is as realistic as possible. Preliminary development and debugging in the simulation environment facilitate the transition to actual vehicles.

The control stack is built in an inner-outer loop architecture, where the dynamics and kinematics of the vehicle are decoupled, simplifying the software's development, as follows in Figure 3.3.



**Figure 3.3:** Inner-Outer Loop architecture.

Firstly, the inner-loop controller gives the necessary forces and torques for the vehicle to be able to reach given references, such as surge, sway, yaw, and depth. These controllers are mostly PIDs (Proportional-Integral-Derivative) or just PIs controllers. On the other hand, the outer-loop is responsible to give the references for the inner-loop to follow in such a way the desired mission is accomplished.

In this approach, it is assumed that the inner-loop is sufficiently faster than the outer-loop in order to ensure the stability of the system.

# Chapter 4

## Sonar Image Net Detection

The goal, as described in Section 1.2, is for the autonomous vehicle to be capable of detecting the aquaculture infrastructure, resorting to a Multibeam Forward-Looking Sonar, and moving around it, while performing an inspection.

In this chapter, a net cage detection algorithm is proposed. The aim is to detect the aquaculture net from the FLS images. Furthermore, the algorithm must be fast since it has to run in real time in an autonomous vehicle with limited computational resources.

The main idea of the approach is to filter the noise of the sonar acoustic image and locate echo highlights to get the position of the net support infrastructure. Then, those points can be used to find the location of the net cage center to help in the vehicle's navigation. To accomplish this, *a priori* knowledge about the fish cage is used.

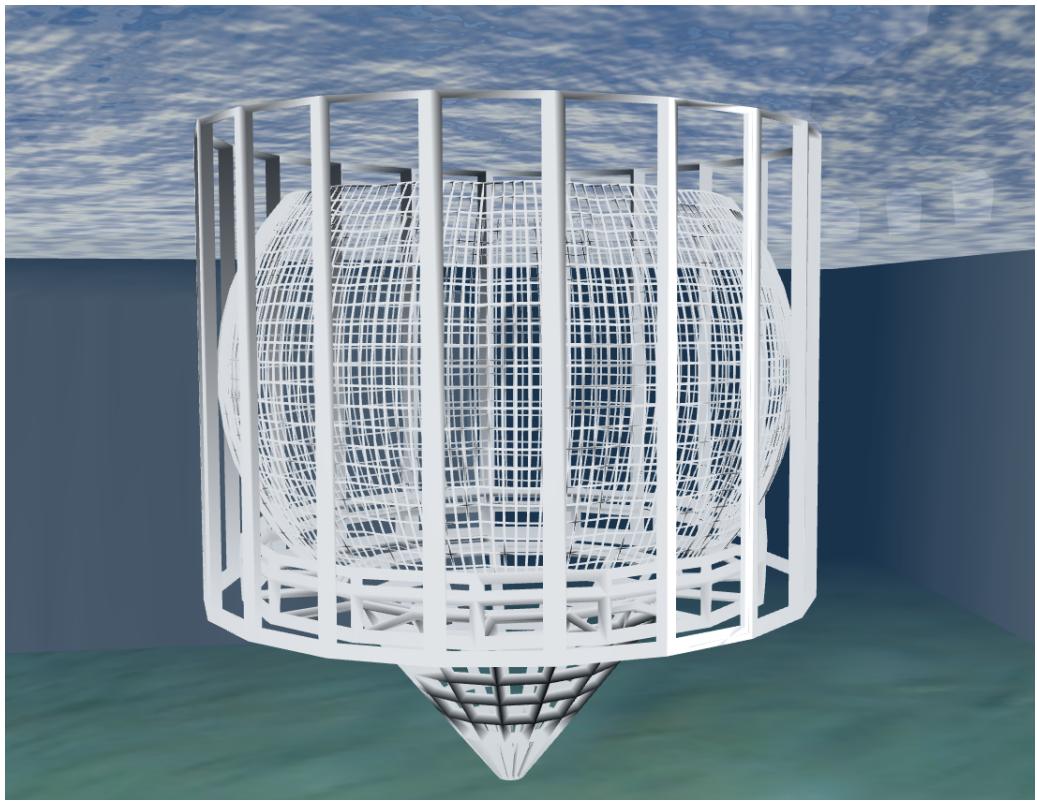
### 4.1 Net Cage 3D Model

In Figure 4.1, the 3D model of an Aquaculture Net Cage used in this thesis is shown. This model was designed by the Laboratory for Underwater Systems and Technologies (LABUST) of the Faculty of Electrical Engineering and Computing (FER) of the University of Zagreb and used in [16].

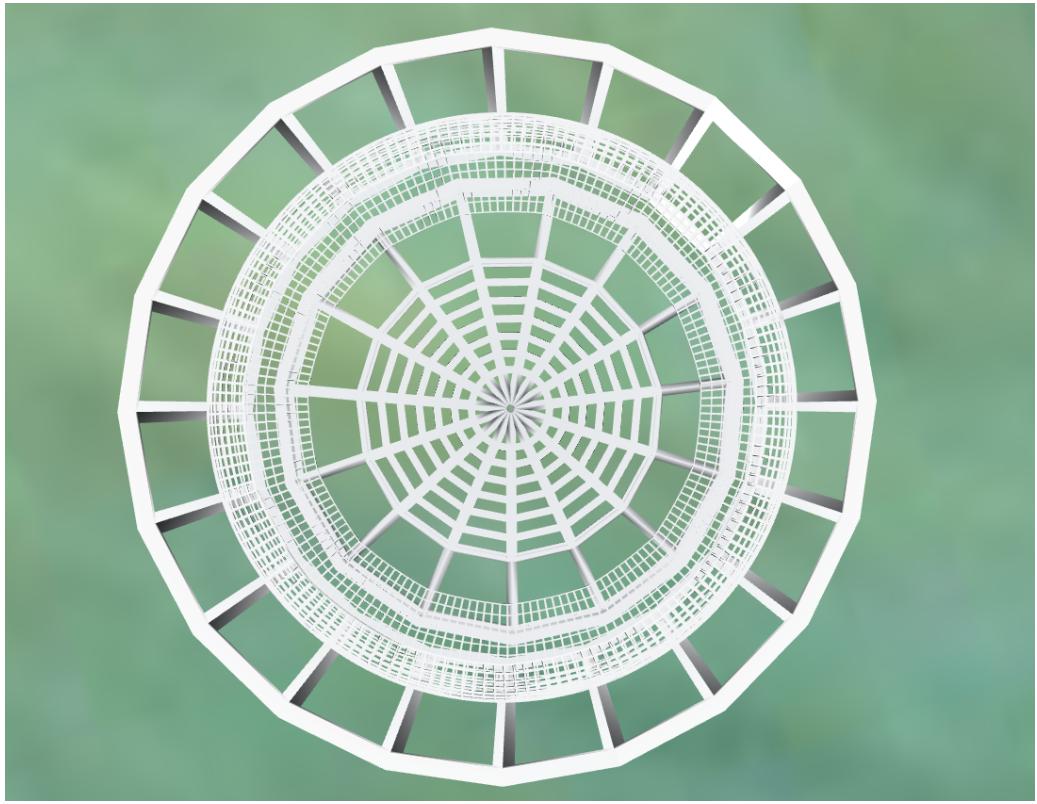
The net cage is composed of 20 outer posts that are responsible for holding the net. As one can observe in Figure 4.1a, the outer posts form a cylindrical shape, when seen sideways. Also, in the top view, in Figure 4.1b, the posts can be approximated by a circle. In addition, the fish net has a form of a sphere, placed inside the outer posts. The dimensions of the cage are represented in Table 4.1.

**Table 4.1:** Net Cage Dimensions

Diameter	12.80 m
Outer Posts Height	9.63 m
Distance between Posts $d_{\text{pole}}$	2.03 m
Sphere Height	7.12 m



(a)



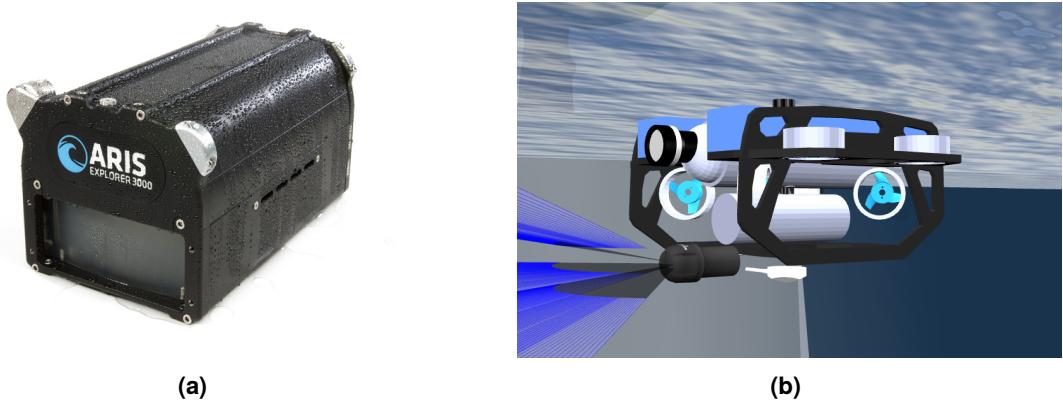
(b)

**Figure 4.1:** Simulated Aquaculture Net Cage used in this project. This 3D model was designed by the Laboratory for Underwater Systems and Technologies (LABUST) of the Faculty of Electrical Engineering and Computing (FER) of the University of Zagreb. (a): Side view. (b) Top view.

## 4.2 Forward-Looking Multibeam Sonar

In this work, a multibeam echosounder plugin is used for the simulation. This plugin was developed by Woen-Sug Choi *et al.* [30] and implemented as a plugin in the Gazebo framework. The source code and description of the method used for simulation are available in [31].

The sonar is attached to the vehicle, as one can see in Figure 4.2b. Considering  $\{M\}$  being the sonar reference frame, it has the same orientation as the  $\{B\}$  frame relative to the inertial frame  $\{U\}$ . In addition,  $\{M\}$  frame is a translation from  $\{B\}$  described by the position of the sonar in the body frame represented by  ${}^B S = [{}^B x_s, {}^B y_s, {}^B z_s]^T$ .



**Figure 4.2:** Multibeam sonar used in this project. (a) Sound Metrics ARIS Explorer 3000 ([32]). (b) BlueROV with the multibeam sonar attached in the gazebo simulation.

The FLS plugin allows for the specification of multiple parameters. Table 4.2 indicates the parameters used in this thesis, corresponding to the ARIS Explorer 3000 [32], shown in Figure 4.2a. The Field-of-View is the maximum area that the sonar beams can cover. However, in order to have a broader view of the net structure, a horizontal field-of-view twice the ARIS horizontal field-of-view was considered. This roughly estimates the work of two ARIS sonars beside each other. Note that the overlapping of both sonars is not considered.

**Table 4.2:** Parameters of the simulated Multibeam Sonar

Operating Frequency	1.8 MHz
Horizontal Field-of-View	60°
Vertical Field-of-View (VFOV)	10°
Number of Beams	512
Beam Spacing	0.18°
Range ( $R$ )	15 m

### 4.2.1 Imaging Forward-Looking Sonar Geometry

All the 3D points of an FLS ensonified area can be represented in spherical coordinates (bearing  $b$ , elevation  $e$ , and range  $R$ ). Figure 4.3 shows the imaging sonar projection model. Considering the point  ${}^M \mathbf{P} = [{}^M x, {}^M y, {}^M z]^T$  in the sonar coordinate frame. This point  ${}^M P$  can be represented in spherical coordinates as shown in (4.1).

$${}^M\mathbf{P} = \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} R \cos b \cos e \\ R \sin b \cos e \\ R \sin e \end{bmatrix} \quad (4.1)$$

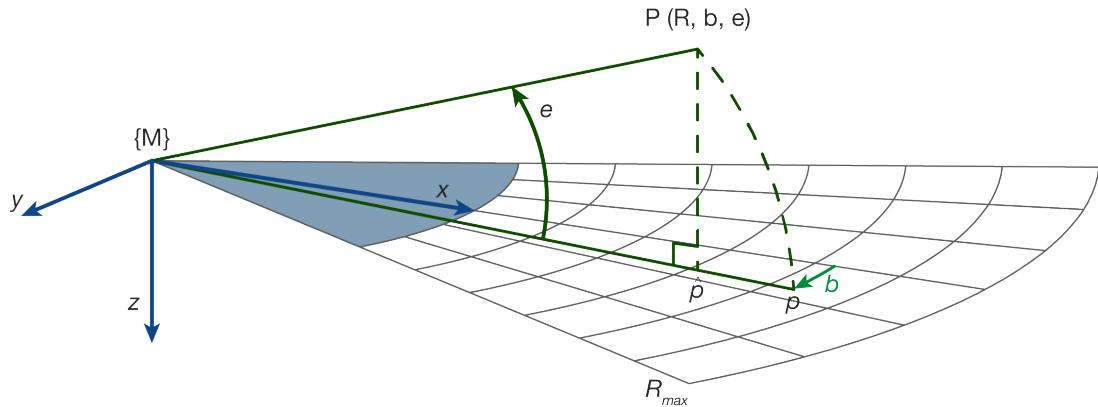
In order to represent the point  ${}^M\mathbf{P}$  in an image, an approximation is made by projecting the 3D point onto a 2D plane. Equation (4.2) shows the projection of the 3D point  ${}^M\mathbf{P}$  onto the 2D point  $\hat{\mathbf{p}}$  using an orthographic projection as an approximation. A study on the effect of this approximation in image registration can be found in [33].

$$\hat{\mathbf{p}} = \begin{bmatrix} m \\ n \end{bmatrix} = \begin{bmatrix} R \cos b \\ R \sin b \end{bmatrix} \quad (4.2)$$

This approximation is valid only if:

- the FLS is mounted with a small tilt angle;
- the vertical beam width should be less than or equal to  $10^\circ$ .

In the case stated above, the elevation angle  $e$  is small, thus  $\cos(e) \approx 1$  and  $\sin(e) \approx 0$ , which validates (4.2). Since during this project, both conditions are met, this approximation will be used.



**Figure 4.3:** Imaging Sonar Projection Model.

Furthermore, with this approximation, the image height can correspond to the sonar range in meters. With this correspondence, distances in the  $x$ - $y$  plane can be computed and converted from meters to image pixels and back and forth in the following manner

$$d_m = d_{px} \frac{R_{\max}}{h}, \quad (4.3)$$

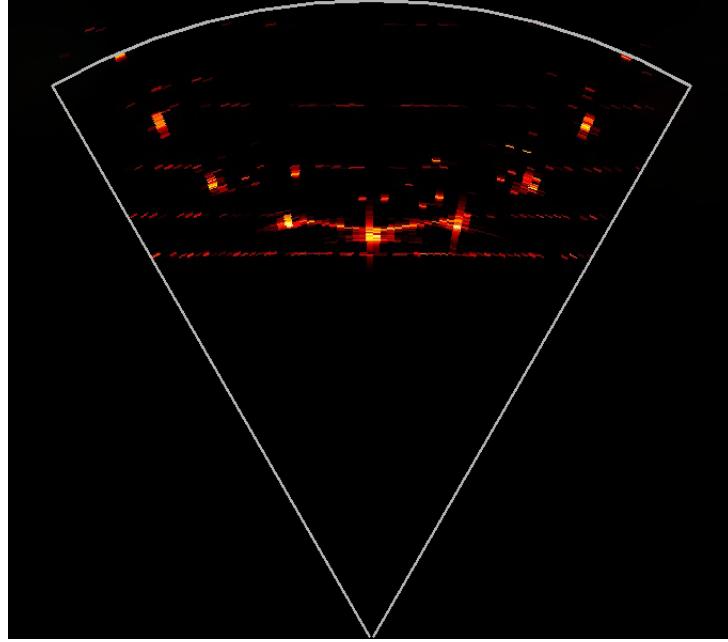
where  $d_{px}$  is a distance in pixels to be converted,  $R_{\max}$  the max range of the sonar in meters and  $d_m$  the distance  $d_{px}$  in meters.

In the sonar image, the pixels' positions are indexed starting at the top left corner, being ordered from left to right (considering the  $x$ -axis) and from top to bottom (considering the  $y$ -axis). Furthermore, in the

image, the sonar is located in the middle of the image's bottom border described by

$$\text{img } P_{\text{sonar}} = \begin{bmatrix} \text{img } x_{\text{sonar}} \\ \text{img } y_{\text{sonar}} \end{bmatrix} = \begin{bmatrix} w/2 \\ h \end{bmatrix}, \quad (4.4)$$

with  $w$  and  $h$  being the width and height of the image, respectively. With this being said, since the sonar is an FLS with the same orientation as the vehicle, it is aligned with the  $x_B$ -axis direction of the  $\{B\}$  frame. Then, the image corresponds to the  $x$ - $y$  plane in front of the vehicle.



**Figure 4.4:** Sonar Image 10 meters away from the net cage. The supporting posts for the cage net are clearly visible in their circular configuration. The horizontal lines correspond to multiple sonar reflections at the water's surface.

## 4.3 Image Processing

Firstly, the image given by the sonar must be processed to reveal the relevant features of the net and filter the noise. In this section, Figure 4.4 will be used as an example to demonstrate the image processing pipeline.

The acoustic image is provided in a red colormap. To simplify the algorithm and make it lighter computationally, the image is changed to a grayscale image, which is represented in Figure 4.5a.

### 4.3.1 Bilateral Filter

After that, the gray image is run through a bilateral filter. The result can be seen in 4.5b. As it is shown in [5], the bilateral filter can be formulated by

$$BF[I]_p = \frac{1}{W_p} \sum_{q \in S} G_{\sigma_s}(\|p - q\|) G_{\sigma_r}(|I_p - I_q|) I_q, \quad (4.5)$$

where  $I$  is the gray image,  $I_p$  denotes the image value at pixel position  $p$ . The set  $S$  denotes all possible image locations. With

$$W_p = \sum_{q \in S} G_{\sigma_s}(\|p - q\|) G_{\sigma_r}(|I_p - I_q|), \quad (4.6)$$

being a normalization term, so the pixel weights sum to 1.0. The terms  $G_{\sigma_s}$  and  $G_{\sigma_r}$  are 2D Gaussian Kernels with such form

$$G_\sigma(x) = \frac{1}{2\pi\sigma^2} \exp\left\{-\frac{x^2}{2\sigma^2}\right\}. \quad (4.7)$$

The parameters  $\sigma_s$  and  $\sigma_r$  are adjustable to establish the desired type of filtering. In (4.6), the term  $G_{\sigma_s}(\|p - q\|)$  is a spatial Gaussian weighting which gives less importance to distant pixels from  $p$ . On the other hand, the term  $G_{\sigma_r}(|I_p - I_q|)$  is responsible for reducing the importance of pixels with intensity values that diverge from  $I_p$ .

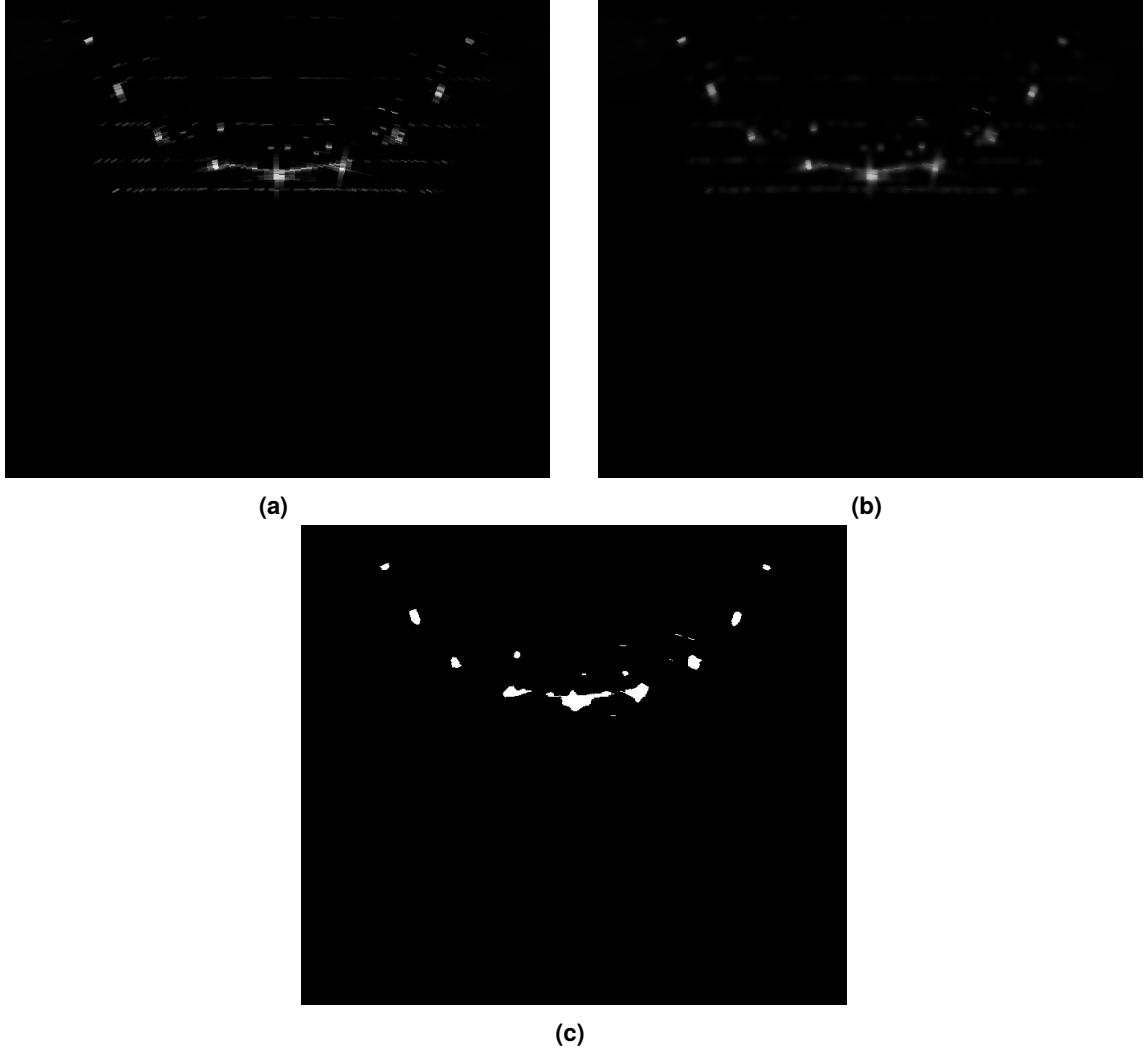
With the correct adjustment of the parameters  $\sigma_s$  and  $\sigma_r$ , the filter smooths the image, reducing the effect of the noise while maintaining the sharp contours. Furthermore, it can be computed quite fast, which was a deciding factor in choosing this type of filter.

### 4.3.2 Binary Filter

Finally, binary thresholding is performed in the resulting image of the bilateral filter. This kind of filtering be defined by

$$TF(BF(I_p)) = \begin{cases} 255 & \text{if } BF(I_p) \geq \xi, \\ 0 & \text{if } BF(I_p) < \xi, \end{cases}, \quad (4.8)$$

where the parameter  $\xi$  is a fixed constant. As well as can be understood in (4.8), the values bigger than  $\xi$  are now 255, otherwise are 0. The resulting image is displayed in Figure 4.5c. One can see that the net cage structure stands out after the binary filtering, terminating the image segmentation.



**Figure 4.5:** Stages of Processing the image in Figure 4.4: (a) Gray Scale. (b) Bilateral Filter. (c) Binary Filter.

## 4.4 Circle Regression

Now, with the net cage points identified in the image, the next step is to find the center of the structure for the vehicle to be able to steer in its direction.

Considering that the net cage outer poles form a cylinder, the points can be approximated to a circle, since the sonar image could be viewed as a slice in the  $x$ - $y$  plane of the structure. Therefore, the chosen approach was to fit a circle to the data points. It is described by

$$(x - a)^2 + (y - b)^2 = r_{px}^2, \quad (4.9)$$

where  $(a, b)$  is the center coordinates and  $r_{px}$  the radius of the outer poles in pixels. For the simulated cage, the radius  $r_{px}$  is constant and can be converted from meters to pixels according to (4.3) by

$$r_{px} = r_m \frac{h}{R_{\max}}, \quad (4.10)$$

with  $r_m$  the outer poles radius in meters. Suppose  $n$  points were extracted from the binary image and are stacked in the vector  $t = [t_1, t_2, \dots, t_n]$ , with  $t_i = (x_i, y_i)$ . Defining  $\beta = [a, b]^T$ , the circle fitting problem can be formulated as a least-squares problem expressed by

$$\arg \min_{\beta} \text{SSE}(\beta), \quad (4.11)$$

where SSE is the sum squared error given by

$$\text{SSE}(\beta) = \sum_{i=1}^n \left[ r_{px} - \underbrace{\sqrt{(x_i - a)^2 + (y_i - b)^2}}_{f(t_i; \beta)} \right]^2. \quad (4.12)$$

Since  $r_{px}$  is constant and known (from Equation (4.10)), the only remaining parameters to find are the center's coordinates,  $(a, b)$ . To solve (4.11), Python's Scipy Function `optimize.least_squares` was used. This function uses the Levenberg-Marquardt (LM) Algorithm [34] to determine (4.11). The LM algorithm is an iterative method, which requires an initial condition  $\beta_0 = [a_0, b_0]$  to be established.

In each iteration,  $\beta$  is replaced by a new estimate  $\beta + \delta$ . The function  $f$  is linearized by

$$f(t_i; \beta + \delta) \approx f(t_i; \beta) + J_i \delta, \quad (4.13)$$

where

$$J_i = \frac{\partial f(t_i; \beta)}{\partial \beta} \quad (4.14)$$

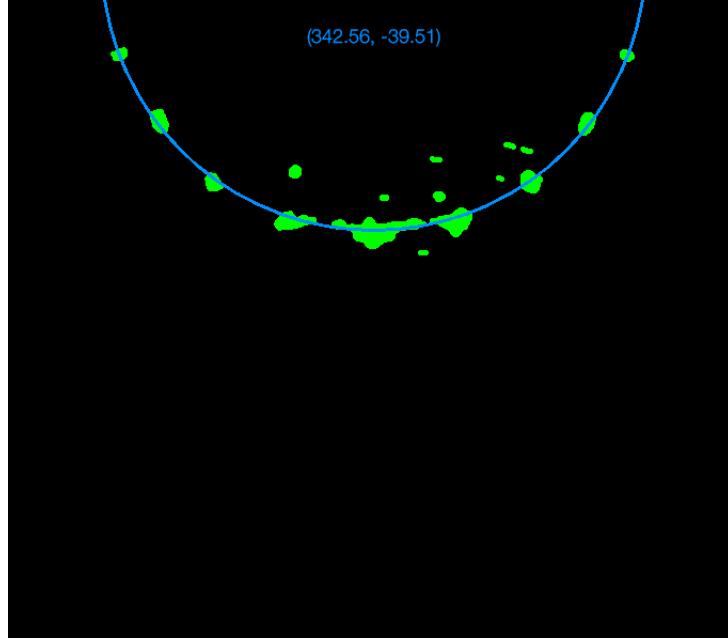
is the gradient of  $f$  with respect to  $\beta$ .

The increment  $\delta$  is computed by

$$[J^T J + \lambda \text{diag}(J^T J)]\delta = J(r - f(t; \beta)), \quad (4.15)$$

with  $\lambda$  being a damping factor and  $J$  being the Jacobian Matrix, whose  $i$ -th row equals  $J_i$ . The parameter  $\beta$  is updated to  $\beta + \delta$ , if  $S(\beta) > S(\beta + \delta)$ . The algorithm stops when certain conditions are met that are described in [34]. The damping factor is chosen according to multiple criteria depicted also in [34].

After solving (4.11) for Figure 4.5c, the solution is shown in Figure 4.6. One can see that the circle fits well with the green points of the net cage poles. Note: since the estimated center has a  $y$  coordinate value below zero, the center is beyond the image in this case.



**Figure 4.6:** Circle regression for the Figure 4.5c. The green dots are the points for which the circle (in blue) was fitted. The computed center coordinates were  $(342.56, -39.51)$ .

#### 4.4.1 Initial Condition

The LM algorithm is iterative and must have an initial condition, raising the problem of how to choose it. For the first detection, the real center's coordinates in the image  ${}^{\text{img}}C = [{}^{\text{img}}x_C, {}^{\text{img}}y_C]^T$  are computed and set as the initial condition for the LM algorithm.

At first, it is known *a priori* the center's position, described by  ${}^U C = [{}^U x_C, {}^U y_C]^T$  (the  $z$  coordinate is neglected since we are dealing with the  $x$ - $y$  plane), and the initial position of the vehicle, given by  ${}^U P = [{}^U x, {}^U y]^T$ . The center's location in  $\{B\}$  is provided by

$${}^B C = {}_U^B R \left( {}^U C - {}^U P \right) \quad (4.16)$$

and in  $\{M\}$  sonar frame is expressed by

$${}^M C = {}^B C - \begin{bmatrix} {}^B x_S \\ {}^B y_S \end{bmatrix} = \begin{bmatrix} {}^M x_C \\ {}^M y_C \end{bmatrix}. \quad (4.17)$$

Lastly, one can compute  ${}^{\text{img}}C$  with

$$\begin{cases} {}^{\text{img}}x_C = {}^M y_C \frac{h}{R_{\max}} + {}^{\text{img}}x_{\text{sonar}} \\ {}^{\text{img}}y_C = - {}^M x_C \frac{h}{R_{\max}} + {}^{\text{img}}y_{\text{sonar}} \end{cases}. \quad (4.18)$$

It is important to note that the rotation from  $\{M\}$  frame to the image frame is given by

$$\begin{bmatrix} {}^{\text{img}}x \\ {}^{\text{img}}y \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \begin{bmatrix} {}^Mx \\ {}^My \end{bmatrix}. \quad (4.19)$$

For the first detection, it is important to notice that the initial condition is computed by Equation (4.17), using the vehicle's initial position. However, after the first detection, the vehicle's position is unknown, so, the initial guess will be the previous estimation, for the next circle regressions.

#### 4.4.2 Inner Points Filter

Figure 4.7a represents a sonar image of when the vehicle is 4 meters away from the net. The center's estimation with the method described in the previous section is displayed in blue in Figure 4.7b.

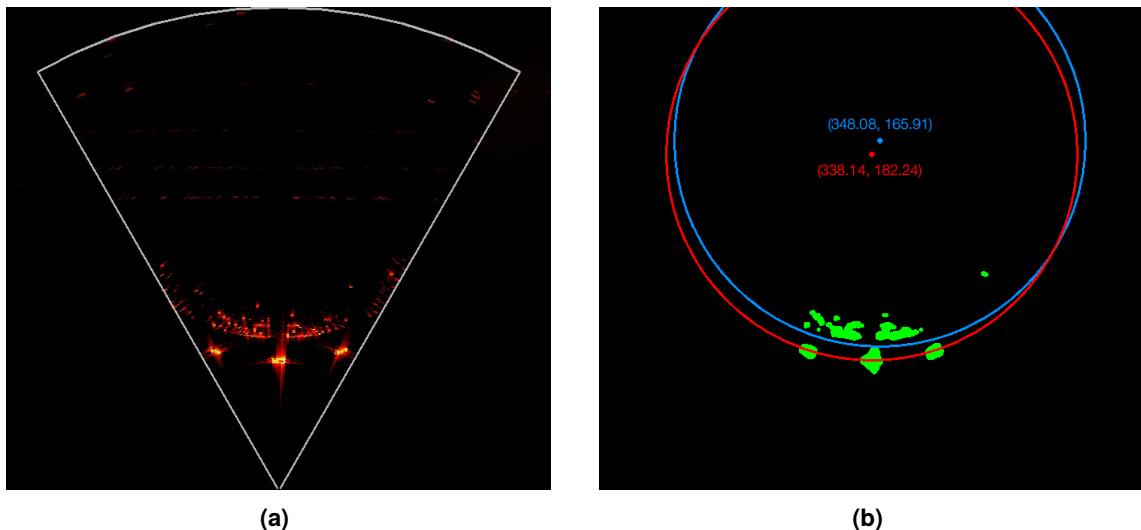
As one can notice, the blue circle does not intersect the outer blobs, which represent the outer poles of the net. Therefore, the circle can fit better the outer poles if the inner points of the blue circle are not considered.

To improve the circle estimate, we developed an algorithm for filtering the inner points on the image, corresponding to reflection on the cage net and not on the support poles. Its logic is depicted in Figure 4.8. So, after the first regression, the mean squared error (MSE) is computed. Considering MSE being expressed by

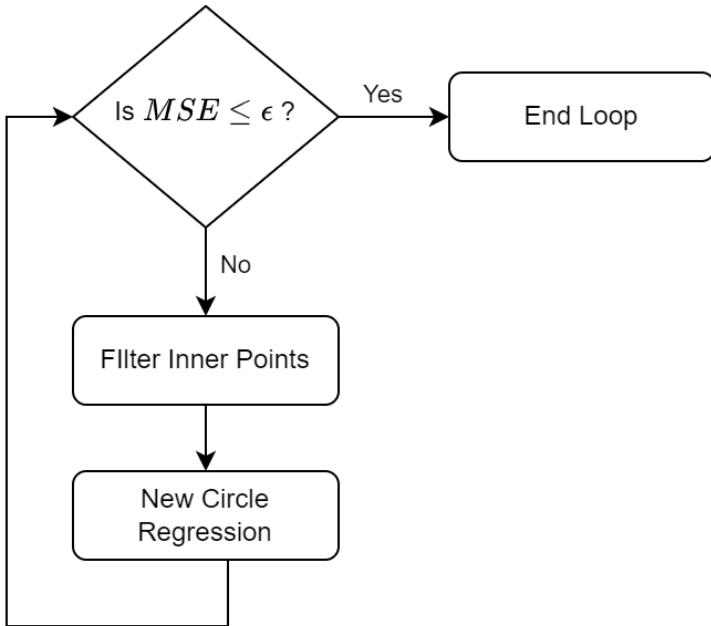
$$\text{MSE}(\beta) = \frac{1}{n} \sum_{i=1}^n [r - f(t_i; \beta)]^2. \quad (4.20)$$

Then, if MSE is bigger than a defined threshold value  $\epsilon > 0$ , the inner points of the circle, represented by  $f(t_i; \beta) < r$ , are taken out in the next regression. After this, the algorithm stops when there is a regression under the threshold  $\text{MSE} \leq \epsilon$ .

Finally, the red circle in Figure 4.7b is the new result after the filter, where the circle intersects the center of each of the outer poles.



**Figure 4.7:** (a) Sonar image at 4 meters from the net. (b) Circle Regression for image (a), the blue circle is the solution before outlier detection, and the red one is after outlier detection.



**Figure 4.8:** Flowchart of the filter of the circle's inner points algorithm.

#### 4.4.3 Pole Detection

However, when the vehicle is close to the net (less than 3.0 meters from the net), there is a different behavior from the previous examples. At those distances, the sonar can get only one of the poles in the image, just as exemplified in Figure 4.10a. During these specific moments, the resulting circle regression, depicted in Figure 4.10b, does not accurately represent the intended outcome.

Just a pole does not have enough data points to accurately estimate the center using circle regression. To address this issue, the points behind the pole, corresponding to the net, are considered. These points form an arc that can be used for the circle regression. Although the inner net does not have the same radius as the outer poles, the circle regression realized to the inner net points is considered to be a good estimation of the center, since the difference between radii is not significant and both the outer posts and inner net are concentric. Therefore, by detecting the pole, circle regression can be performed solely with the inner points.

Initially, the approach taken to locate the pole was to examine the momentums of the nearest blob. However, since the shape of the blob continually changes according to the sonar image, this method proved to be inadequate and was therefore abandoned. Subsequently, an alternative method was devised.

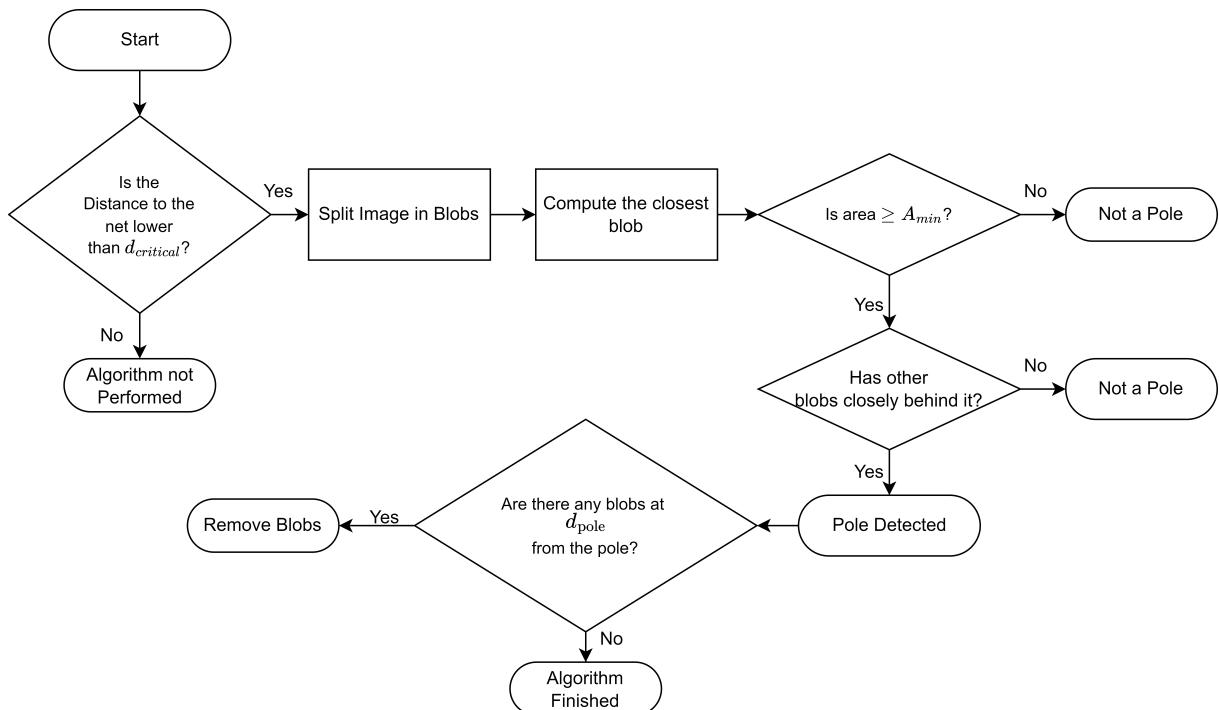
The new approach entails dividing the image points into various blobs to accurately pinpoint the pole. Once these blobs have been formed, the pole is then identified using three distinct criteria:

- the blob's centroid needs to be the closest to the sonar;
- the blob's area has to be larger than  $A_{\min} > 0$  in order to filter noise;
- there must be other blobs positioned closely behind the blob, with a distance  $d_b$  between at least one blob and this blob's centroid such that it is under the threshold  $d_b < d_{\max}$ .

If the blob fulfills the three criteria mentioned above, it is classified as a pole. Furthermore, as stated previously, the presence of poles influences the circle regression when the vehicle is at a particular distance from the net. Hence, this method is only applied when the distance relative to the net is below a predefined threshold value, denoted as  $d_{critical}$ . Succeeding the pole detection, the circle regression is conducted without taking the pole into account, with its resulting output presented in Figure 4.10c. As one can see, the detection was successful as the circle fits the points behind the pole, where the used points in the regression are in green.

However, it should be noted that the aforementioned method can solely detect the nearest pole to the sonar. In cases where sonar images display two poles instead of one, it becomes necessary to detect the remaining poles as well.

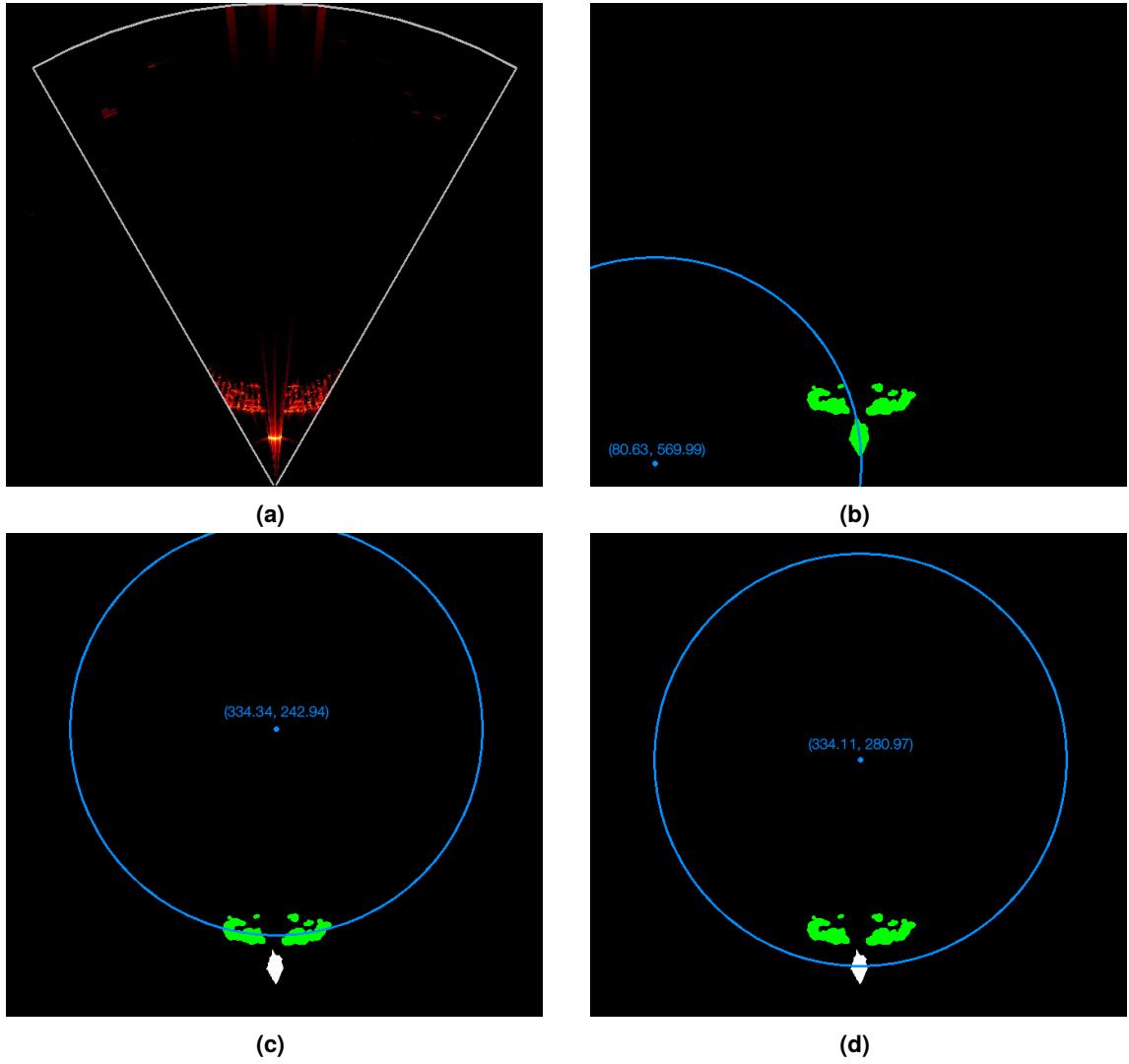
To detect the other poles, we use the known distance between poles, denoted as  $d_{pole}$  and presented in Table 4.1. By converting this distance to pixels using Equation (4.3), we examine if there are any blobs located at that specific distance from the detected pole. However, it should be noted that the detection of the other poles is not highly precise, as it captures all the blobs within the distance between the poles. Despite this limitation, extensive analysis has demonstrated that excluding these blobs from the regression does not significantly impact the accuracy of the estimations. Then, the detected poles are not considered in the circle regression. The overall procedure is shown in Figure 4.9.



**Figure 4.9:** Flowchart of the pole detection procedure.

### Center Estimation Adjustment

After the execution of the pole detection algorithm, the resulting circle of the regression aligns with the inner net rather than the outer poles, as can be seen in Figure 4.10c. Notably, the circle that fits the inner net has the same radius as the outer poles. Consequently, the circle that intersects the pole center can



**Figure 4.10:** (a): Sonar image with the vehicle at 2 m away from the net. (b): Circle regression for image (a) without pole detection. (c): Circle regression for image (a) with pole detection. (d): Circle after the center's adjustment to intersect the outer poles.

be considered as a translation of the circle that fits the inner net. Now, the main challenge is to determine the correct direction of the translation. Considering the centroid of the detected pole  ${}^{img}W$  expressed by

$${}^{img}W = \begin{bmatrix} {}^{img}x_W \\ {}^{img}y_W \end{bmatrix}, \quad (4.21)$$

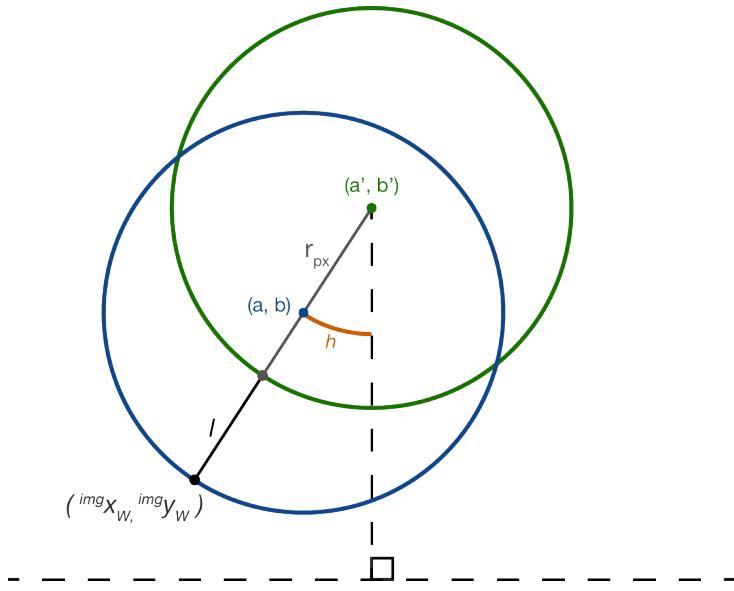
the distance of the translation is defined by

$$l = \sqrt{(a' - {}^{img}x_W)^2 + (b' - {}^{img}y_W)^2} - r_{px}, \quad (4.22)$$

with  $\beta' = [a', b']^T$  characterizing the estimated center fitting the inner net. Now, the desired center  $\beta = [a, b]^T$ , which intersects the outer poles, is computed by

$$\begin{cases} a = a' + l \cos h \\ b = b' + l \sin h \end{cases}, \quad h = \arctan \left( \frac{a' - \text{img}x_w}{b' - \text{img}y_w} \right), \quad (4.23)$$

where  $h$  is the angle of the translation. Figure 4.11 depicts a diagram of the calculations in Equation (4.23), where the green circle exemplifies the one fitting the inner net and the blue one intersecting the pole's centroid. Applying Equation (4.23) to the example in Figure 4.10c, the resulting center is represented in Figure 4.10d, where the circle intersects the pole's center.



**Figure 4.11:** Center Adjustment Diagram.

#### 4.4.4 Center Estimation Performance

In order to analyze the center estimation performance, a set of tests for different distances and depths were conducted. The algorithm was run 2000 times, while the vehicle was moving around the net. Table 4.3 presents the accuracy of the center's estimation for each case. For evaluation purposes, any detection with an error exceeding 0.5 m from the real center was considered inaccurate.

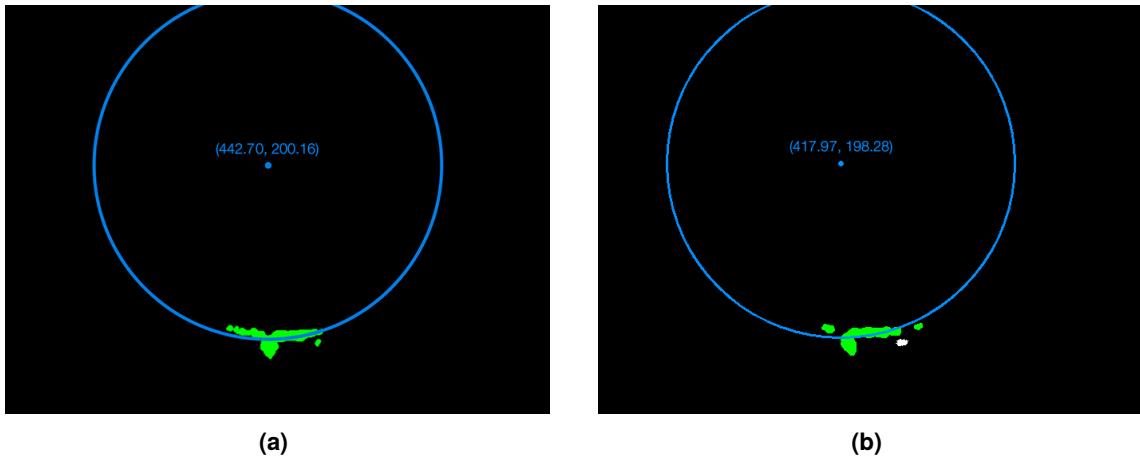
Firstly, at a distance of 2.5 m, the pole's detection algorithm was performed to estimate the center. As one can see, the accuracy is around 90% for depths at 2 and 5 m. However, at a depth of 4 m, the accuracy decreases to 79.9%. This decrease in accuracy is attributed to the close proximity of the pole and the net at that depth, and, after the image filtering process (bilateral and binary filters), the data points of the pole and the inner net merge, forming a single blob, as Figure 4.12 exemplifies. Nevertheless, an accuracy of 79.8% still indicates a reasonable performance. Furthermore, when the distance is varied from 4 m to 10 m while maintaining the same depth, the accuracy remains around 90% for most cases, except at a distance of 10 m, where it is 83.60%.

In conclusion, the center estimation demonstrates satisfactory performance across multiple distances

**Table 4.3:** Center's Estimation Accuracy for different distances and depths.

Distance [m]	Depth [m]	Accuracy [%]
2.5	2.0	92.8%
2.5	4.0	79.8%
2.5	5.5	88.65%
4.0	2.5	88.95%
6.0	2.5	96.45%
8.0	2.5	91.25%
10.0	2.5	83.60%

and depths. The lowest accuracy obtained was 79.80%, indicating that the estimation remains reliable in most scenarios.



**Figure 4.12:** Examples of bad detections.

## 4.5 Distance Measurement

After the image processing, the resulting image is split in blobs. With this, the closest blob to the sonar in the image is computed using

$${}^{\text{img}}d_{\text{net}} = \sqrt{({}^{\text{img}}y_{\text{sonar}} - x_{\text{blob}})^2 + ({}^{\text{img}}y_{\text{sonar}} - y_{\text{blob}})^2}. \quad (4.24)$$

After that, the  ${}^{\text{img}}d_{\text{net}}$  is converted to meters with (4.3).

# Chapter 5

## Inspection Controller

In this chapter, vehicle motion control will be addressed. The motion control is organized in an inner-outer loop approach. In addition, the velocity and orientation references are generated from the sonar image measurements, in order for the vehicle to move around the net cage.

### 5.1 Inner-loop controllers

As mentioned in Section 3.8, the FAROL stack is structured in an inner-outer loop architecture, where the outer loop is responsible for generating references for the inner loop to follow. In this case, the inner loop controls the forces and torques, represented in Equation (3.13), so the vehicle can reach the velocity or orientation references given by the outer loop.

More commonly, PIDs are used for the inner-loop controllers since they are easy to develop and tune. In this thesis, PIDs or PID variations were selected as the controllers to regulate the forces and torques of the vehicle.

#### 5.1.1 Yaw Controller

The inner-loop controller for the vehicle's yaw  $\psi$  was designed to converge to a desired heading  $\psi_d$ . Considering this, the yaw error can be defined as

$$\tilde{\psi} = \psi - \psi_d. \quad (5.1)$$

The yaw error time derivative is expressed by

$$\dot{\tilde{\psi}} = \dot{\psi} - \dot{\psi}_d. \quad (5.2)$$

For controller design, it is considered that the  $\psi_d$  is constant, resulting in  $\dot{\psi}_d = 0$ . From Equation (3.20), Equation (5.2) can be represented by

$$\dot{\tilde{\psi}} = \dot{\psi} = r. \quad (5.3)$$

Now, applying the second-time derivative of the error and using Equation (3.20),  $\ddot{\psi}$  can be described as

$$\ddot{\psi} = \dot{r} = \frac{1}{m_r} (\tau_r - d_u \dot{\psi}). \quad (5.4)$$

The proposed control law is

$$\tau_r = \left( -K_P \tilde{\psi} - K_I \int \tilde{\psi} - K_D \dot{\psi} \right) m_r, \quad (5.5)$$

with  $K_P, K_I, K_D > 0$ .

Replacing Equation (5.5) in (5.4) gives

$$\ddot{\psi} = -K_P \tilde{\psi} - K_I \int \tilde{\psi} - \left( K_D + \frac{d_r}{m_r} \right) \dot{\psi}. \quad (5.6)$$

Knowing that  $d_r/m_r \geq 0$ , from PID theory, the error converges to zero if  $K_P$ ,  $K_I$ , and  $K_D$  are positive values.

### 5.1.2 Surge Controller

Here, the inner-loop controller for the surge velocity  $u$  is formulated. The goal is for the velocity to converge to a desired speed  $u_d$ . The surge error is defined by

$$\tilde{u} = u - u_d. \quad (5.7)$$

The time derivative of the error is described in Equation (5.8). Considering that the desired speed is constant, then  $u_d = 0$ .

$$\dot{\tilde{u}} = \dot{u} - u_d \quad (5.8)$$

Using Equation (3.20), Equation (5.8) can be written as

$$\dot{\tilde{u}} = \dot{u} = \frac{1}{m_u} (\tau_u + m_v v r - d_u u). \quad (5.9)$$

Now, proposing the control law with a feed-forward term as

$$\tau_u = m_u \left( -K_P \tilde{u} - K_I \int \tilde{u} \right) + d_u u - m_v v r, \quad (5.10)$$

with  $K_P, K_I > 0$ . Replacing this law in Equation (5.9), the time derivative of the surge error is expressed by

$$\dot{\tilde{u}} = -K_P \tilde{u} - K_I \int \tilde{u}. \quad (5.11)$$

From PID theory, the error converges to zero if  $K_P$  and  $K_I$  are positive, making the surge velocity reach the desired value.

### 5.1.3 Sway Controller

The sway inner-loop controller was designed to make the sway converge to a desired value  $v_d$ .

The sway error is defined by

$$\tilde{v} = v - v_d, \quad (5.12)$$

and its time derivative is described as

$$\dot{\tilde{v}} = \dot{v} - \dot{v}_d, \quad (5.13)$$

where the desired speed is considered constant and, therefore,  $\dot{v}_d$  can be assumed to be zero.

Applying Equation (3.20), the time derivative of the sway error is given by

$$\dot{\tilde{v}} = \frac{1}{m_u} (\tau_v - d_v v), \quad (5.14)$$

and replacing it in Equation (5.13), it is obtained

$$\dot{\tilde{v}} = \dot{v} = \frac{1}{m_v} (\tau_v - d_v v). \quad (5.15)$$

Now, it is proposed a control law, with a feed-forward term, represented by

$$\tau_v = \left( -K_P \tilde{v} - K_I \int \tilde{v} \right) + d_v v, \quad (5.16)$$

with  $K_P, K_I > 0$ . Replacing the proposed control law in Equation (5.15), it results in

$$\dot{\tilde{v}} = -K_P \tilde{v} - K_I \int \tilde{v}, \quad (5.17)$$

from which one can conclude that the error will reach zero if  $K_P$  and  $K_I$  are positive values, according to PID theory.

### 5.1.4 Depth Controller

Finally, the depth controller was designed. The depth error is characterized as

$$\tilde{z} = z - z_d, \quad (5.18)$$

where  $z_d$  is the desired depth and  $z$  the vehicle's current depth. The time error derivative is expressed by

$$\dot{\tilde{z}} = \dot{z} - \dot{z}_d, \quad (5.19)$$

and, since the desired depth is constant,  $\dot{z}_d$  is zero. Taking from Equation (3.20), one gets

$$\dot{z} = w, \quad (5.20)$$

$$\dot{\psi} = \frac{1}{m_w} (\tau_w - d_w \dot{z}). \quad (5.21)$$

Now, using both Equations (5.20) and (5.21), the error second-time derivative is expressed by

$$\ddot{z} = \dot{\psi} = \frac{1}{m_w} (\tau_w - d_w \dot{z}). \quad (5.22)$$

Lastly, the proposed control law is

$$\tau_w = \left( -K_P \tilde{z} - K_I \int \tilde{z} - K_D \dot{\tilde{z}} \right), \quad (5.23)$$

with  $K_P, K_I, K_D > 0$ . Thereafter, substituting the control law in Equation (5.22), the second-time derivative can be given by

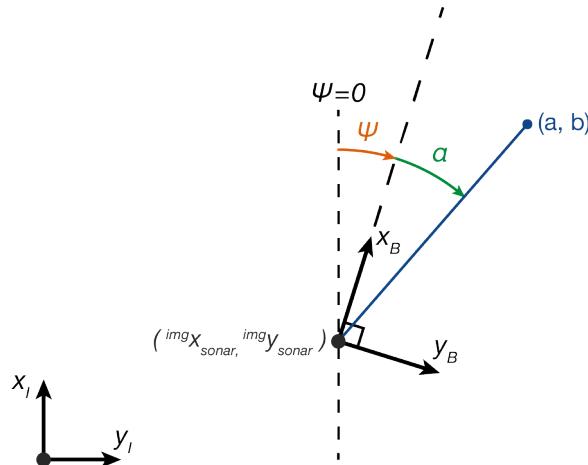
$$\ddot{z} = -K_P \tilde{z} - K_I \int \tilde{z} - \left( K_D + \frac{d_w}{m_w} \right) \dot{\tilde{z}}. \quad (5.24)$$

Knowing that  $d_w/m_w \geq 0$ , then, with  $K_P, K_I$ , and  $K_D$  positive, the depth error goes to zero, according to PID theory.

## 5.2 Yaw Reference

As stated previously, the main goal was for the vehicle to steer and be oriented toward the net cage's center, during the inspection. Therefore, the desired vehicle's heading can be computed using the coordinates of the net cage's center, estimated from the sonar image (see Section 4.4).

Figure 5.1 shows a schematic diagram for the determination of the vehicle's desired orientation, where the sonar position on the image is  $[{}^{img}x_{sonar}, {}^{img}y_{sonar}]^T$  and the net cage's estimated center is  $[a, b]^T$ . To simplify the calculations, the sonar was considered to be at the origin of the body frame. This assumption was made based on the understanding that this translation does not significantly impact the control. If such an influence were to exist, it could be taken into account, with the cost of increasing the complexity of the calculations. This way, for the vehicle to be pointing directly to the center  $[a, b]^T$ , it has to rotate the angle  $\alpha$  in the body frame's  $z$ -axis.



**Figure 5.1:** Yaw Reference Calculation Diagram.

Following the stated reasoning, the yaw reference is given by

$$\psi_{\text{ref}} = \psi + \alpha, \quad (5.25)$$

with

$$\alpha = \arctan 2 \left( -\left( {}^{\text{img}}x_{\text{sonar}} - a \right), {}^{\text{img}}y_{\text{sonar}} - b \right). \quad (5.26)$$

### 5.3 Distance Controller

For the vehicle to approach the net and converge to a desired constant distance, a distance controller was designed. For the development of this controller, it was considered that the vehicle was pointing directly to the net cage's center, under the action of the yaw controller. This way, the distance can only be adjusted by moving forward or backward, i.e., positive or negative surge velocity, respectively. As a result, the time derivative of the distance is the surge velocity itself. Therefore, the main goal of the distance controller is to compute the desired surge velocity of the vehicle.

Defining the error distance by

$$e_d = d - d_{\text{desired}}, \quad (5.27)$$

the time derivative of the error is

$$\dot{e}_d = \dot{d} - \dot{d}_{\text{desired}}, \quad (5.28)$$

with  $\dot{d}_{\text{desired}} = 0$ , since  $d_{\text{desired}}$  is constant.

Now, since the derivative of the distance is the surge velocity, the time derivative of the error can be described by

$$\dot{e}_d = u. \quad (5.29)$$

Therefore, a PI controller was implemented to calculate the desired surge, as

$$u_d = -K_P e_d - K_I \int e_d, \quad (5.30)$$

where  $K_P$  and  $K_I$  are proportional and integral gains, respectively, and  $u_d$  is the desired surge velocity. Since it is assumed that the surge velocity's inner loop controller is sufficiently fast, following the general guideline that the inner loop should possess 5 to 10 times the bandwidth of the outer loop, the surge velocity is  $u \approx u_d$ . Consequently, Equation (5.29) can be reformulated as follows:

$$\dot{e}_d = u_d = -K_P e_d - K_I \int e_d, \quad (5.31)$$

from which can be concluded from the PID theory that the error converges to zero if  $K_P$  and  $K_I$  are positive values.

## 5.4 Sway Reference

The main goal is for the vehicle to be oriented to the net cage's center, and to maintain a desired constant distance to the net. For that purpose, the vehicle's orientation is controlled by the yaw controller, and the distance to the net is controlled by the distance controller. This way, the sway velocity is the only DOF remaining and, so, it is used for the vehicle to move around the net. The sway velocity follows a predefined desired value. This desired velocity is generated in a way that depends on the distance error  $e_d$  and yaw error  $\tilde{\psi}$ , since the AUV should only move if it is correctly oriented towards the net and at a desired distance. Defining the sum of squared errors  $e_t$

$$e_t = c_d e_d^2 + c_\psi \tilde{\psi}^2, \quad (5.32)$$

where  $c_d$  and  $c_\psi$  are positive constants.

For the sway velocity to vary smoothly with the  $e_t$  value, i.e., according to an infinitely continuously differentiable function, one defines

$$W(e_t) = \begin{cases} \exp\left\{1 - \frac{1}{1 - x^2}\right\} & , \text{if } e_t \in [0, 1[ \\ 0 & , \text{otherwise} \end{cases}, \quad (5.33)$$

based on the bump function. Then, the desired sway is given by

$$v_d = v_n W(e_t), \quad (5.34)$$

where  $v_n$  is the nominal speed. The desired sway speed  $v_d$  is  $v_n$  when  $e_t$  is zero.

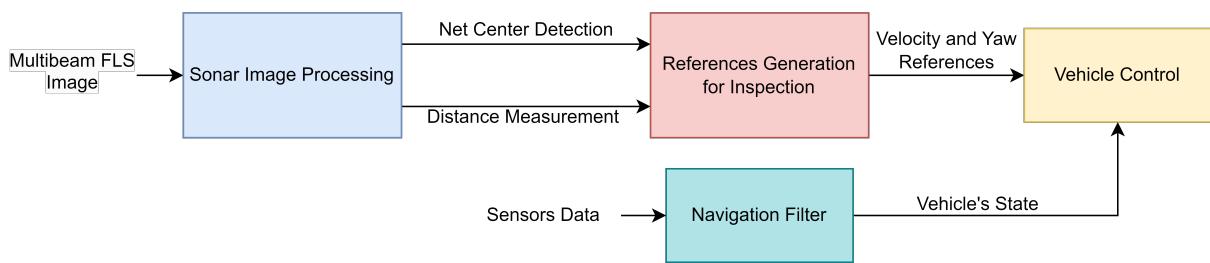
# Chapter 6

## Simulation and Results Discussion

### 6.1 Implementation

As stated in Section 3.8, the implementation of this work was made in the FAROL stack, which is a set of ROS packages in Python and C++ created and developed by the DSOR group. This allows for the simulation of the controllers' behavior and of the sonar image detection, with the BlueROV vehicle in an environment close to reality.

Figure 6.1 displays how the implementation in ROS was made. Firstly, the Multibeam sonar plugin generates the sonar image of the environment in front of the vehicle. Then, the image is processed, using the techniques discussed in Chapter 4, giving the center location on the image and also the distance of the vehicle relative to the net. After that, the velocity and orientation references are generated in order for the vehicle to approach and move around the net structure. The Navigation Filter implemented in the FAROL stack gives the information of the vehicle's state in the inertial frame. With this, one can control the vehicle autonomously.



**Figure 6.1:** Implementation Flowchart.

#### 6.1.1 Multibeam FLS Plugin

As stated before, the used multibeam sonar plugin was developed by Woen-Sug Choi *et al* [31]. To simulate the beams and rays of the Multibeam FLS within the Gazebo framework, a point-based scattering model was adopted. Additionally, noise was incorporated and the reflectivity of targets/objects was taken into account to ensure the beam intensity measurements closely resemble realistic conditions.

Additionally, to decrease the multibeam's process time, GPU parallelization using NVIDIA CUDA was performed. Regarding the output image, it has a resolution of  $512 \times 598$  pixels, with an update rate of approximately 0.8 seconds.

### 6.1.2 Integral Anti-Windup

While using integral control to manipulate a particular state variable, an occurrence known as windup might happen. This occurs when the integral term of the controller accumulates a significant quantity of error before it reaches zero, which conducts the variable to a considerable overshooting. The anti-windup technique implemented in the stretch of this work is known as Back-Calculation, proposed by Fertik and Ross [35].

Defining a state variable saturation as

$$a = \begin{cases} a_u & , \text{ if } a_r > a_u \\ a_l & , \text{ if } a_r < a_l \\ a_r & , \text{ otherwise} \end{cases} \quad (6.1)$$

where  $a_u$  and  $a_l$  are the output's upper and lower bounds, respectively. The back-calculation approach aims to prevent the accumulation of the integral term up to a large value, when the controller's output is saturated, i.e., when  $a \neq a_r$ . This prevention is done by using extra compensation which feedbacks to the integral term. The compensation is defined by the error between the controller output and the system input  $e_p = a_r - a$  multiplied by a constant  $K_a$ . So, the integral term can be defined by

$$K_i \int (e + K_a e_p). \quad (6.2)$$

The constant is defined by  $K_a = 1/\delta_t$ , where  $\delta_t$  determines how fast the integral term will reset. In this case,  $\delta_t$  is defined to be the inverse of the vehicle's controllers' frequency, which is 10 Hz.

### 6.1.3 Yaw Reference Correction

As declared in Section 5.2, the yaw reference is computed based on the measurements of the net cage's center on the image. However, the image provided by the sonar represents the reality in front of the vehicle, but with a certain delay. This delay is due to the time taken from the sonar plugin to generate the image, the image processing techniques, and, lastly, the circle regression algorithm. Thus, the yaw reference calculated solely with the information given from the sonar image is not accurate enough.

In the context of this problem, the yaw reference will only vary when the vehicle is circulating the net infrastructure, i.e., when the sway reference is not zero. So, to estimate a more accurate value for the yaw reference, one calculates a correction value  $\psi_c$  with

$$\psi_c = r_n \Delta t, \quad (6.3)$$

where  $r_n$  is the nominal yaw rate, that is, the yaw rate that the vehicle should have in order to move in the desired trajectory (at the desired velocity and distance). Furthermore, the  $\Delta t$  is the time interval between receiving two consecutive center estimations.

The nominal yaw rate is described by

$$r_n = -\frac{P}{v_n}, \quad (6.4)$$

where  $P$  is the perimeter of the vehicle's desired trajectory, defined by

$$P = 2\pi(r_m + d_{\text{desired}}), \quad (6.5)$$

with  $r_m$  as the net radius and  $d_{\text{desired}}$  as the desired distance. The  $v_n$  is the nominal sway speed.

Lastly, the yaw reference should be updated as

$$\psi_{\text{ref}} = \psi_{\text{ref}0} + \psi_c, \quad (6.6)$$

with  $\psi_{\text{ref}0}$  being the yaw reference computed by Equation (5.25) after the center estimation.

#### 6.1.4 Yaw Reference Smoothing

Each time the yaw reference changes, the yaw controller tries to reach it. However, in order to reach the new yaw reference as soon as possible, the vehicle gains a sudden spike in yaw rate  $r$ . With this, the cross terms in Equation (3.14) have a more significant effect in the surge  $u$  and sway  $v$  velocities.

To soften that undesired effect, instead of the yaw reference changing abruptly with the update given by each detection, as shown in Equation (6.6), the desired yaw is now computed to change linearly as

$$\psi_d(t + \delta_t) = \frac{\psi_{\text{ref}}(t_k + \Delta t) - \psi_{\text{ref}}(t_k)}{\Delta t} \delta_t + \psi_d(t), \quad (6.7)$$

where  $\psi_{\text{ref}}(t_k)$  is the yaw measured in the last detection at the instant  $t_k$ ,  $\psi_{\text{ref}}(t_k + \Delta t)$  is the yaw measured in the detection at the instant  $t_k - \Delta t$  and  $\psi_d(t)$  is the previous yaw desired. The desired yaw  $\psi_d$  is computed with Equation (6.7) for all the tests realized, being the value given for the yaw controller to follow.

#### 6.1.5 Yaw Error

Moreover, in the FAROL stack, the yaw angle is contained in a specific interval,  $\psi \in [0, 360]^\circ$ . Consequently, the calculation of the yaw error must be adjusted to handle this discontinuity. Therefore, the yaw error  $\tilde{\psi}$  in degrees is redefined as

$$\tilde{\psi} = \begin{cases} \tilde{\psi}_0, & |\tilde{\psi}_0| < |\tilde{\psi}_0 - \text{sgn}\tilde{\psi}_0 \times 360| \\ \tilde{\psi}_0 - \text{sgn}\tilde{\psi}_0 \times 360, & |\tilde{\psi}_0| > |\tilde{\psi}_0 - \text{sgn}\tilde{\psi}_0 \times 360| \end{cases}, \quad (6.8)$$

with  $\tilde{\psi}_0 = \psi - \psi_d$ . This ensures that the error is always in the interval  $[-180, 180]^\circ$  and, in this way, the vehicle steers the minimum required.

### 6.1.6 Yaw Reference Error

To analyze the accuracy of the yaw measurements, it is noteworthy to characterize the error between the yaw measurements and the ideal yaw, that is, the yaw angle that the vehicle must have to be pointing towards the net cage's center.

The ideal yaw  $\psi_{\text{ideal}}$  is described as

$$\psi_{\text{ideal}} = \arctan 2(^U y_C - y, ^U x_C - x), \quad (6.9)$$

with  $[x, y]^T$  being the vehicle's location, given by the GAZEBO Simulator, and  $^U C = [^U x_C, ^U y_C]^T$  as the coordinates of the net cage's center in the inertial frame.

Then, the yaw reference error is defined by

$$\tilde{\psi}_d = \psi_d - \psi_{\text{ideal}}. \quad (6.10)$$

It is important to notice that this value is computed just to evaluate the yaw reference, and, so, this value is not controlled.

### 6.1.7 Actual Distance

In addition, to analyze the accuracy of the distance measurements by the sonar, the actual distance between the vehicle and the net cage is calculated. In other words, this value is the ground truth for the measured distance.

For the calculation of the actual distance, the coordinates of the vehicle and the net cage center  $^U C$  in the inertial frame are used. The actual distance is defined by

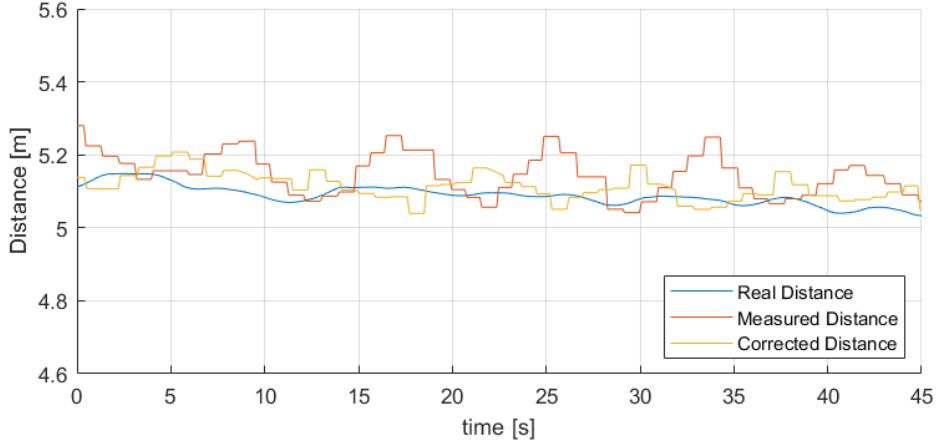
$$d_{\text{actual}} = \sqrt{(x - ^U x_C)^2 + (y - ^U y_C)^2} - r_m, \quad (6.11)$$

with  $r_m$  as the net radius.

### 6.1.8 Distance Correction

As stated previously in Section 4.5, the measurement of the distance between the vehicle and the net cage is done by computing the Euclidean distance to the sonar's nearest blob on the image and, then, converting it to meters.

However, the nearest blob is always an outer pole, since they are always the closest object to the vehicle. Then, the distance is measured to the nearest pole, not to the closest point to the circumference that intersects all the net cage outer poles. Thus, as the vehicle is circulating the net cage, the measured distance oscillates, as Figure 6.2 shows, despite the actual distance remaining approximately constant.



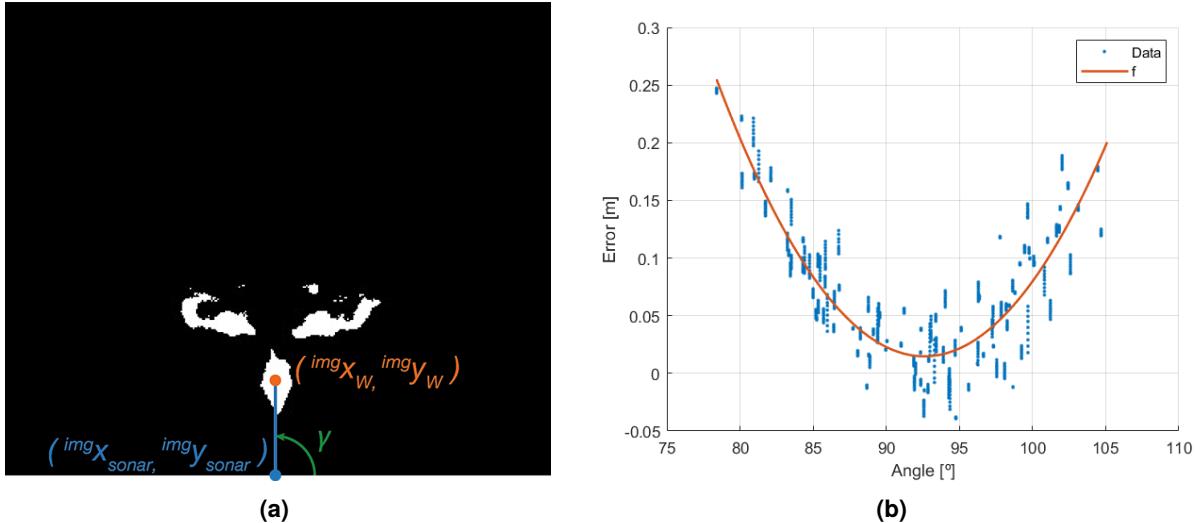
**Figure 6.2:** Measured Distance behavior when the vehicle is circulating the net.

In order to correct this unwanted phenomenon, we checked for patterns in its behavior. Thus, the pattern was evaluated using the error between the measured distance and the actual distance, as well as the angle  $\gamma$  between the line that unifies the post on the image and the sonar location  ${}^{img}P_{sonar}$ , and the image  $x$ -axis, as shown in the diagram in Figure 6.3a. The error is defined by  $e = d - d_{actual}$  and the angle  $\gamma$  is given by

$$\gamma = \arctan 2 \left( {}^{img}x_{sonar} - {}^{img}x_w, {}^{img}y_{sonar} - {}^{img}y_w \right), \quad (6.12)$$

where  $[{}^{img}x_w, {}^{img}y_w]^T$  are the pole coordinates on the image.

Figure 6.3b presents the variation of the error values with the angle  $\gamma$ . It can be seen that there is a correlation between the  $\gamma$  value and the error, which resembles a quadratic relation. Therefore, a second-degree polynomial  $f$  fit the data points to estimate how the error changes with the angle  $\gamma$ , as shown in Figure 6.3b.



**Figure 6.3:** (a): Diagram for angle  $\gamma$ . (b): Variation of the error between the measured distance and the actual distance with the angle  $\gamma$

In this fashion, after finding the function  $f$  parameters with the regression, the corrected distance  $d'$

has the following form

$$d' = d - f(\gamma), \quad (6.13)$$

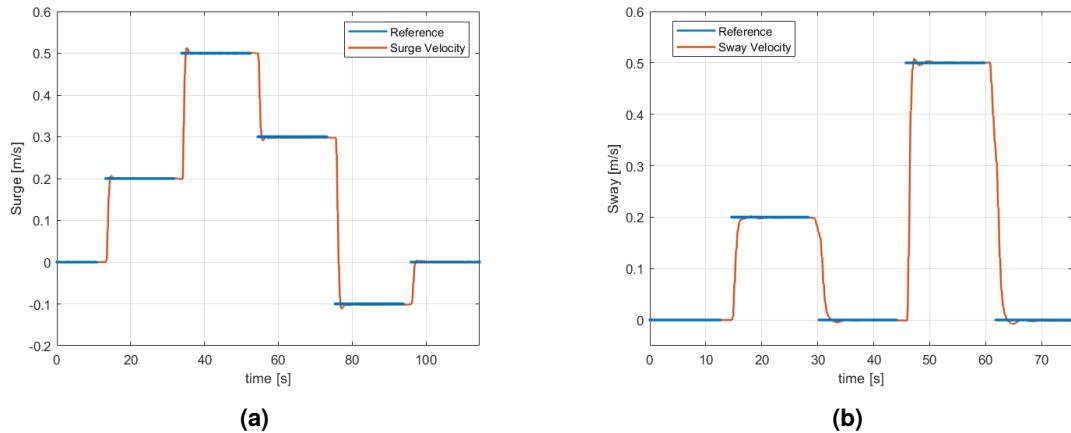
and it is represented by yellow in Figure 6.2. As shown in the figure, the corrected distance follows better the actual distance than before, having a Root Mean Squared Error (RMSE) of 0.0441 m, when the measured distance has an RMSE of 0.0787 m.

Furthermore, observing Figure 6.3b, the data has a lot of variance, with a standard deviation from the function  $f$  of 2.77 cm. When looking at the sonar image dimensions, which are  $512 \times 598$  pixels, from Equation (4.3), each pixel corresponds approximately to 2.5 cm. Thereby, the standard deviation is explained by the uncertainty of the measurements.

## 6.2 Simulation Results

### 6.2.1 AUV Inner-loops

As preliminary tests, the inner-loops defined in Section 5.1 are tested. The vehicle is expected to follow and converge to determined references. The tested controllers are the following: the surge and sway velocity, the yaw angle and the depth controllers. Figure 6.4 represents the vehicle's response when applying different surge and sway velocities references. Both surge and sway can follow and converge rapidly to the desired values with a slight overshoot.

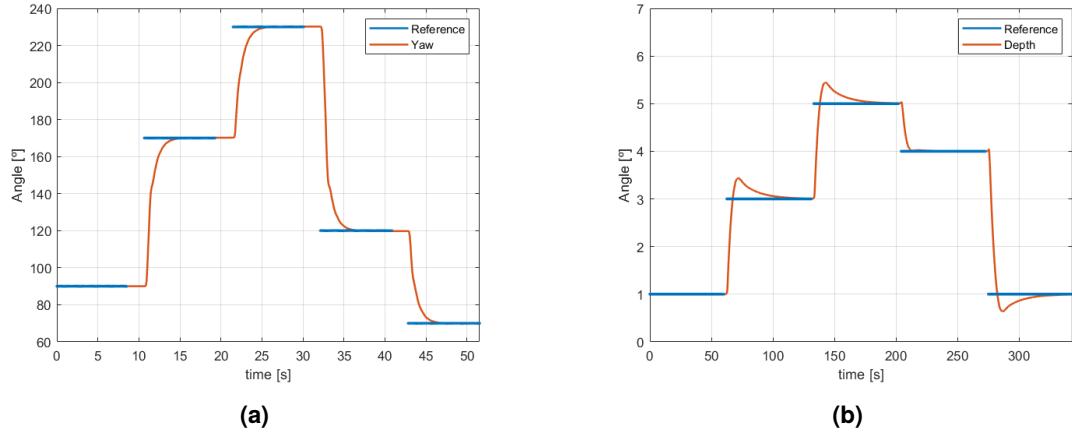


**Figure 6.4:** Controller performance for some different given references: (a) Surge, (b) Sway.

In Figure 6.5, one can see how the yaw and depth controllers behave with different desired values. The yaw controller is able to quickly reach the desired reference values without any overshooting. However, the depth controller experiences overshooting as soon as the vehicle reaches the desired depth due to the integrator gain. But, this allows the vehicle to converge quickly to the desired depth reference.

### 6.2.2 Inspection Tests

In this section, multiple tests were executed to examine the impact of depth, distance to the net, and inspection velocity on the vehicle's performance. Table 6.1 displays the parameters of these tests, where



**Figure 6.5:** Controller performance for some different given references: (a) Yaw, (b) Depth.

each test focuses on changing one aspect to evaluate the influence of these parameters on the inspection process. In addition, in these tests, the vehicle's side movement only starts when it is at the desired condition, being the desired sway speed equal to the nominal speed  $v_n$ .

**Table 6.1:** Aquaculture Net Inspection Simulation Log

Test	Desired Distance - $d_{\text{desired}}$	Nominal Sway - $v_n$	Depth - $z$
T1	2.5 m	0.2 m/s	2 m
T2	2.5 m	0.2 m/s	5 m
T3	2.5 m	0.5 m/s	5 m
T4	5 m	0.2 m/s	5 m
T5	5 m	0.5 m/s	5 m

Beyond those tests presented in Table 6.1, the following tests were performed:

- T6 - test T5 with a smooth approach;
- T7 - a full type inspection;
- T8 and T9 - test T5 in the presence of currents for different controllers parameters.

A smooth approach is tested in T6 with the desired sway given by Equation (5.16), where the value of the reference depends on the distance and yaw errors, allowing a more smooth approach towards the net. Also, an inspection of the whole net is performed in T7, to give a view of a real-life inspection, where the AUV goes through the whole surface of the net. Finally, to test the robustness of the controllers, a trial with a constant current value is done in T8 and T9 for different controllers' parameters.

The parameters used for the controllers are shown in Table 6.2. Also, the parameters used in the sonar image net detection for detecting posts and fitting the circle to the outer points are the following:  $d_{\text{critical}} = 2.9$ ,  $A_{\min} = 100$  and  $\epsilon = 0.5$ .

Throughout all the tests, the center of the net cage was located at  ${}^U C = [30, 0]^T$  m. Furthermore, the vehicle's initial position was  $[15, 0]$  m and its initial yaw angle  $\psi_0$  was  $0^\circ$ .

**Table 6.2:** Controllers Parameters

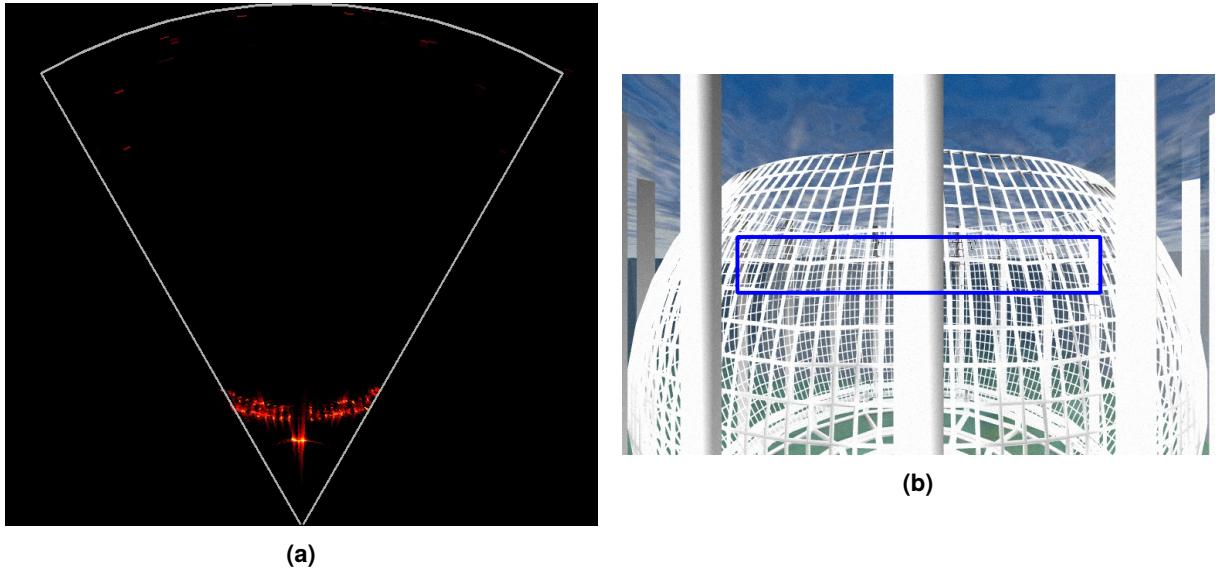
	Distance - $d$	Surge - $u$	Sway - $v$	Yaw - $\psi$	Depth - $z$
$K_P$	0.15	40	50	0.16	20
$K_I$	0.0001	0.42	0.42	0.0005	1.0
$K_D$	-	-	-	0.133	40.0

In order to compare the performance of the tests, the RMSE metric was computed for the measured distance, the actual distance and the yaw reference error.

#### T1 - Sway 0.2 m/s Depth 2 m Distance 2.5 m

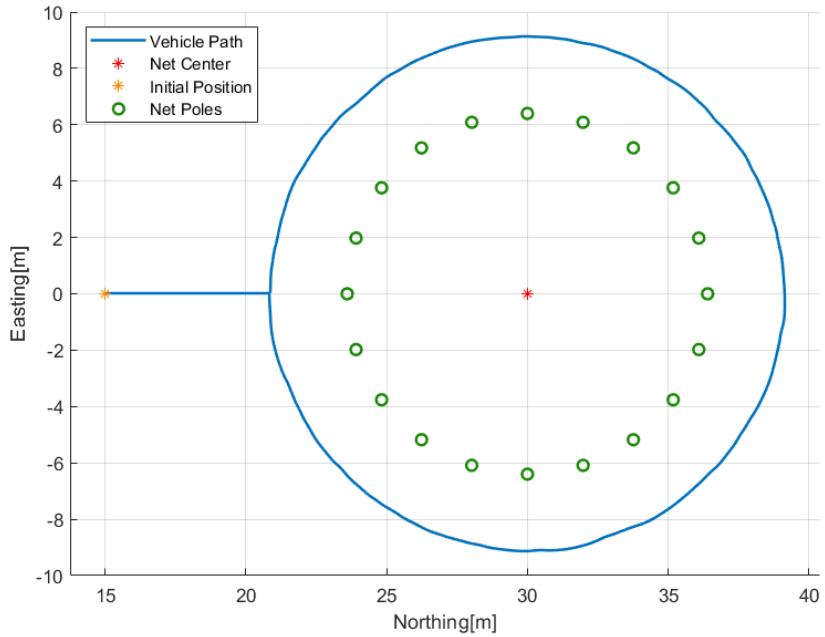
In test T1 the nominal sway speed  $v_n$  was 0.2 m/s, the desired inspection distance from the net  $d_{\text{desired}}$  was 2.5 m, and the depth was 2 m. Figure 6.6 shows examples of both the sonar image and camera view, regarding test T1.

Figure 6.7 presents the trajectory of the AUV in the  $x - y$  plane during this test. As it can be seen, the vehicle circulates the net successfully, completing a full circle.



**Figure 6.6:** (a): Sonar image example during test T1. (b): Camera image example during test T1 with the sonar's field-of-view represented by the blue rectangle.

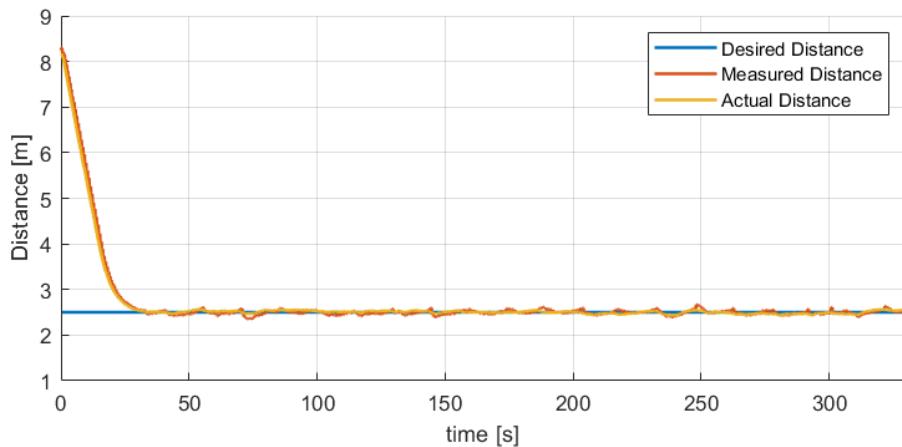
In Figure 6.8, both the measured and the actual distance reach and converge to the desired distance, as expected. Moreover, the actual distance and the measured distance from the sonar images match with a small error, which shows the accuracy of the sonar measurements of the distance. The measured distance for test T1 had an RMSE of 0.85 m and a steady-state error of 0.05 m, while the RMSE for the actual distance was 0.79 m with a steady-state error of 0.03 m. Note that the measured distance value is already the corrected value computed in Equation (6.13). This steady-state error is the RMSE of the distance after reaching the desired value, that is, the error of the oscillations around the reference. Additionally, one can notice that the measured distance has some oscillations, which can be explained by the measurements' uncertainty as well as the fact that the distance is measured to the nearest pole.



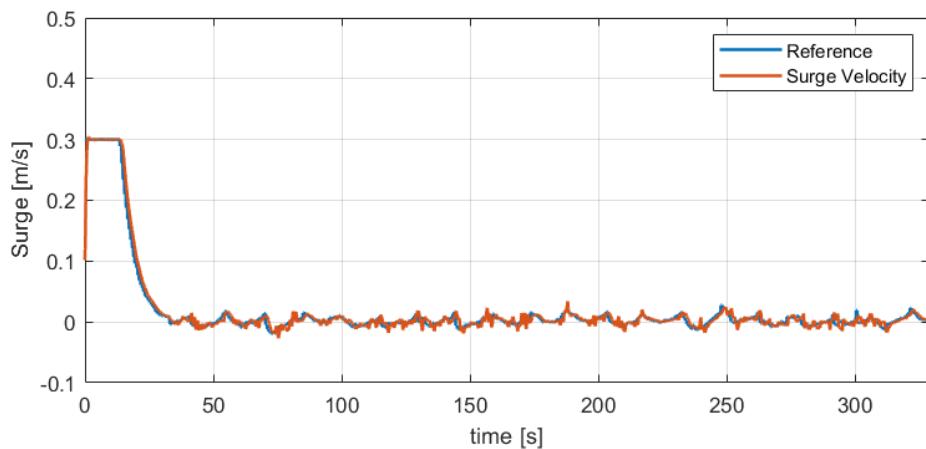
**Figure 6.7:** Trajectory of the vehicle on the  $x$ - $y$  plane while performing test T1 at 0.2 m/s, at 2 m depth, and at 2.5 m from the net.

As declared previously, the distance to the net is controlled through the surge velocity. The vehicle's surge reference depends on the distance error. Figure 6.9 reveals the surge velocity during the inspection. The output of the distance controller is saturated at 0.3 m/s, so the vehicle does not get to move too quickly, as well as it has enough time to stop and not collide with the net. Analyzing the experimental run, at  $t \approx 30$  s, the velocity gets close to zero, which coincides with the moment the vehicle reaches the desired distance. Since the distance oscillates around the desired value with a slight error, the surge velocity is also small, having values in the order of magnitude of  $10^{-2}$  m/s. In addition, there is some small chattering after the time instant  $t \approx 30$  s, which can be explained by the existence of cross terms in the vehicle dynamics, described in Equation (3.13). The Figures 6.10 and 6.11, showing the sway velocity and yaw angle throughout the inspection, respectively, corroborate the previous statement since both the sway and the yaw rate are not equal or close to zero after  $t \approx 30$  s. Moreover, this result also shows that the side movement started at that time instant.

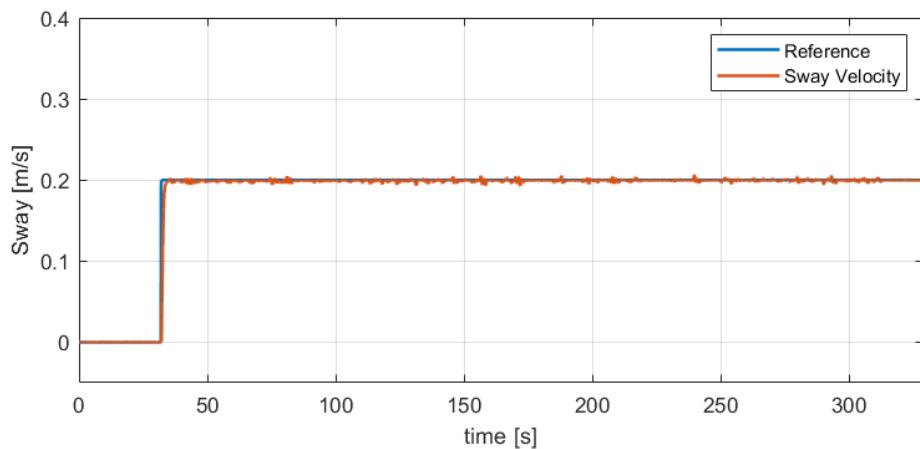
Continuing the analysis of Figure 6.10, the sway follows the desired velocity and converges without any visible oscillation. As for Figure 6.11, yaw follows the reference with minimum error, with an insignificant RMSE of  $0.31^\circ$ . Nevertheless, the yaw reference oscillates and does not have a linear variation after  $t \approx 30$  s. Seeing that the yaw reference is obtained from the detection of the net cage's center, and there is a level of uncertainty associated with each measurement, one should predict a random variation for the yaw reference. For further examination, Figure 6.12 presents the yaw reference error, with an RMSE of  $1.96^\circ$ , as a means to correlate the used yaw reference with the ideal yaw reference (obtained from the ground truth localization), defined in Equation (6.10).



**Figure 6.8:** Distance behavior throughout test T1 at 0.2 m/s, at 2 m depth, and at 2.5 m from the net.



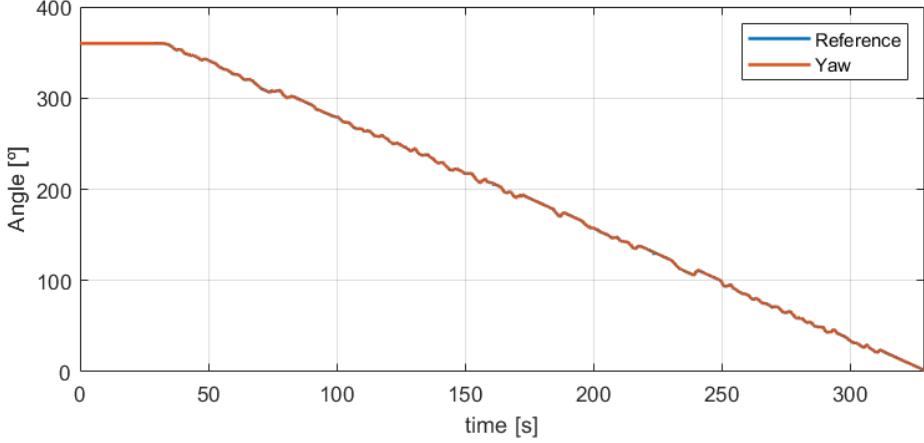
**Figure 6.9:** Surge behavior throughout test T1 at 0.2 m/s, at 2 m depth, and at 2.5 m from the net.



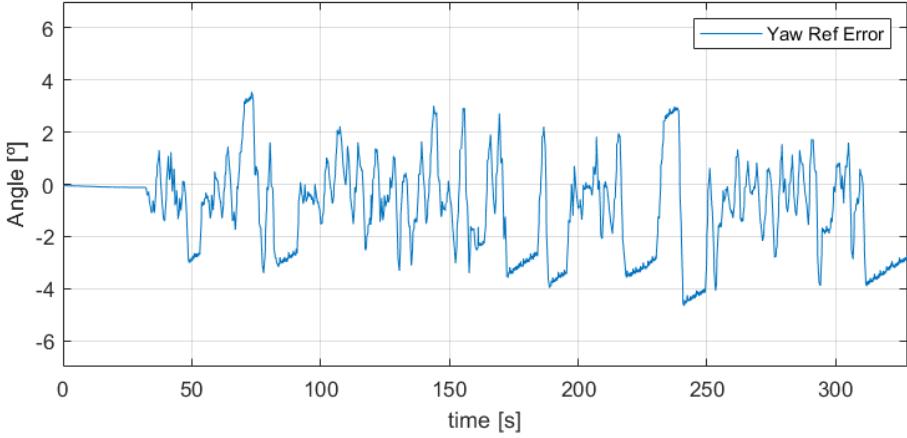
**Figure 6.10:** Sway behavior throughout test T1 at 0.2 m/s, at 2 m depth, and at 2.5 m from the net.

## T2 - Sway 0.2 m/s; Depth 5 m; Distance 2.5 m

To examine how depth affects the movement of the vehicle, test T2 was conducted at a depth of 5 m, maintaining the values of sway and distance from test T1.



**Figure 6.11:** Yaw behavior throughout test T1 at 0.2 m/s, at 2 m depth, and at 2.5 m from the net.

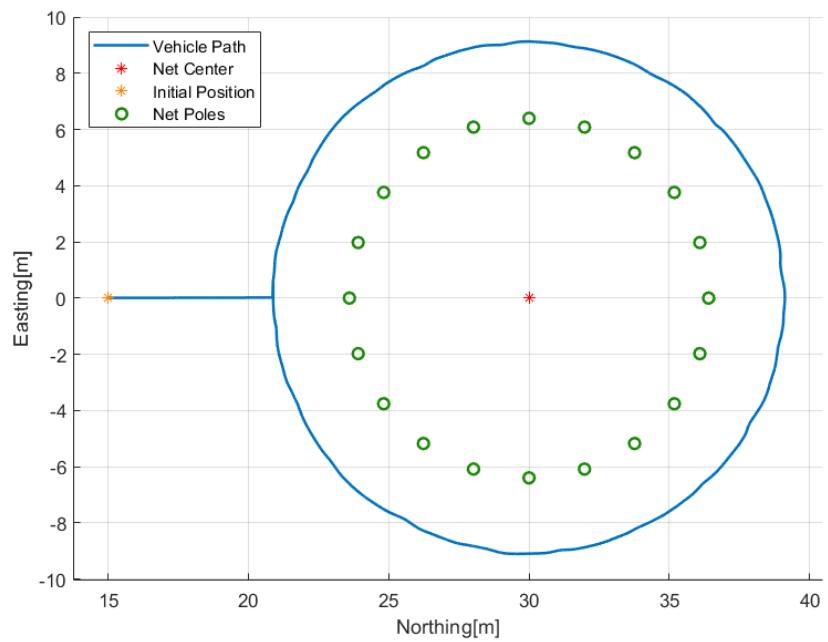


**Figure 6.12:** Yaw reference error throughout test T1 at 0.2 m/s, at 2 m depth, and at 2.5 m from the net.

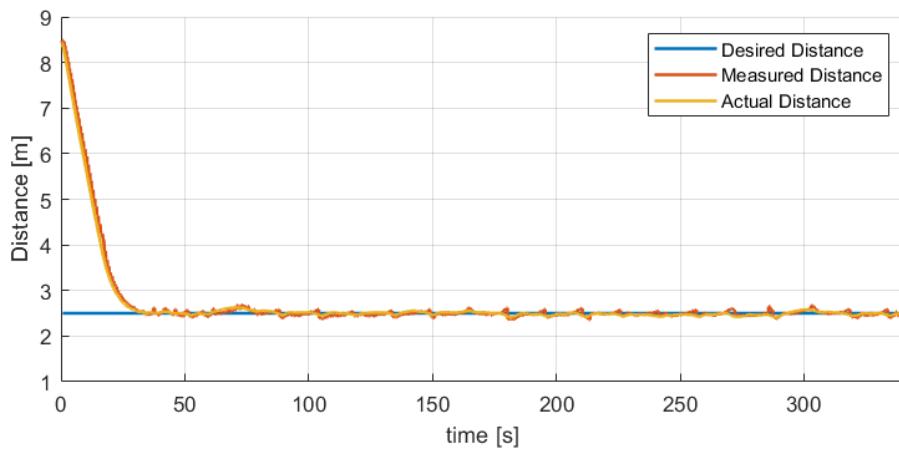
The vehicle's trajectory is presented in Figure 6.13. Similarly to test T1, the AUV encircles around the net with good performance, as expected. Additionally, Figures 6.14 and 6.15 illustrate the vehicle's distance to the net and surge velocity, respectively. Both the distance and the surge perform similarly to test T1, where the first converges to the desired value  $d_{\text{desired}}$  while having minor oscillations due to the measurements' uncertainty. The measured distance had an RMSE of 0.89 m and a steady-state error of 0.06 m, while the RMSE for the actual distance was 0.83 m and with a steady-state error of 0.04 m.

Regarding the sway velocity and the yaw angle, both follow the reference successfully as the last test (as can be seen in Figures A.1 and A.2 in Appendix A). Furthermore, the yaw reference error is displayed in Figure 6.16, having an RMSE value of  $2.21^\circ$ .

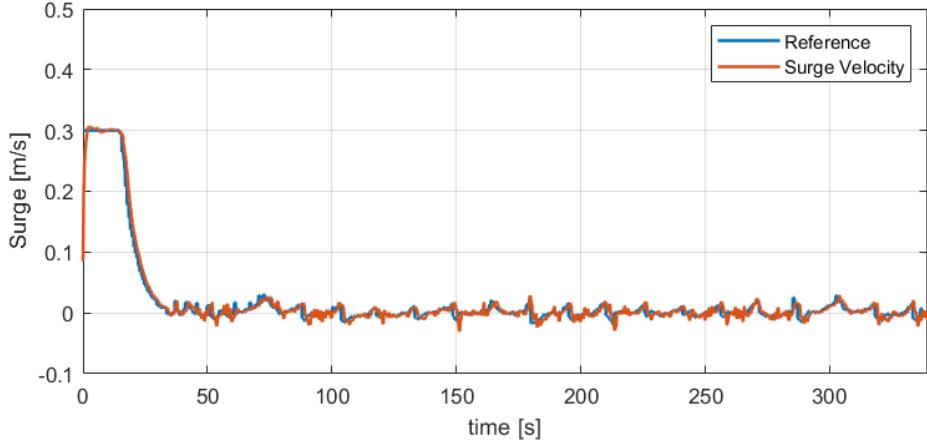
In conclusion, as expected, the distinction between T1 and T2 is not noticeable, so the variation of the depth does not have a major impact on the vehicle's behavior, even though the distance from the cage net to the poles varies significantly.



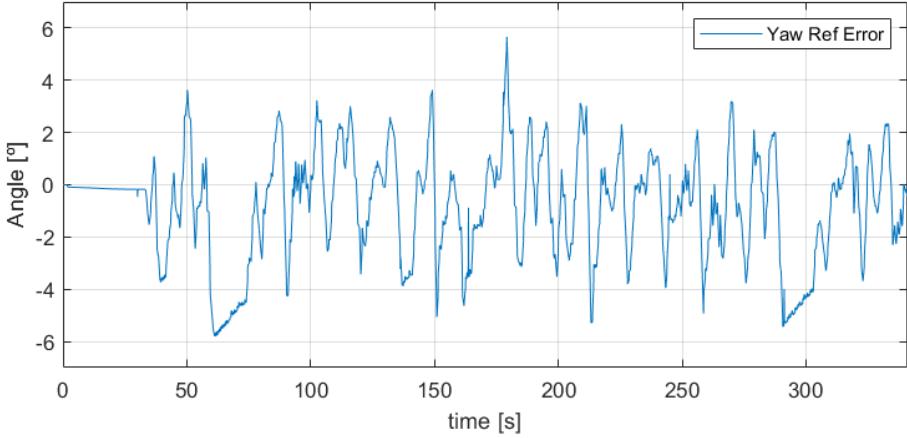
**Figure 6.13:** Trajectory of the vehicle on the  $x$ - $y$  plane while performing test T2 at 0.2 m/s, at 5 m depth, and at 2.5 m from the net.



**Figure 6.14:** Distance behavior throughout test T2 at 0.2 m/s, at 5 m depth, and at 2.5 m from the net.



**Figure 6.15:** Surge behavior throughout test T2 at 0.2 m/s, at 5 m depth, and at 2.5 m from the net.



**Figure 6.16:** Yaw reference error throughout test T2 at 0.2 m/s, at 5 m depth, and at 2.5 m from the net.

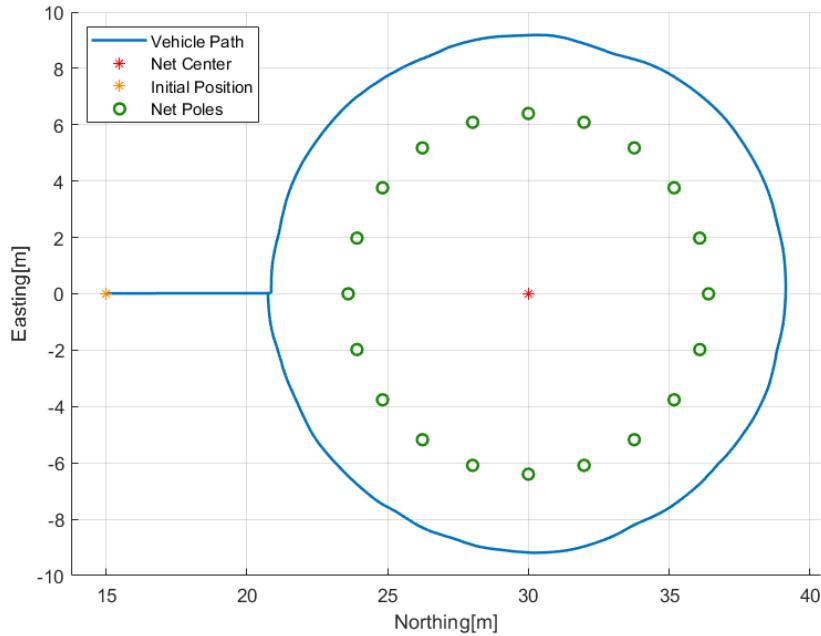
### T3 - Sway 0.5 m/s Depth 5 m Distance 2.5 m

Test T3 was performed with a nominal sway speed  $v_n$  of 0.5 m/s, in order to investigate the effects that the speed has on the inspection performance. The other parameters remain unchanged from test T2.

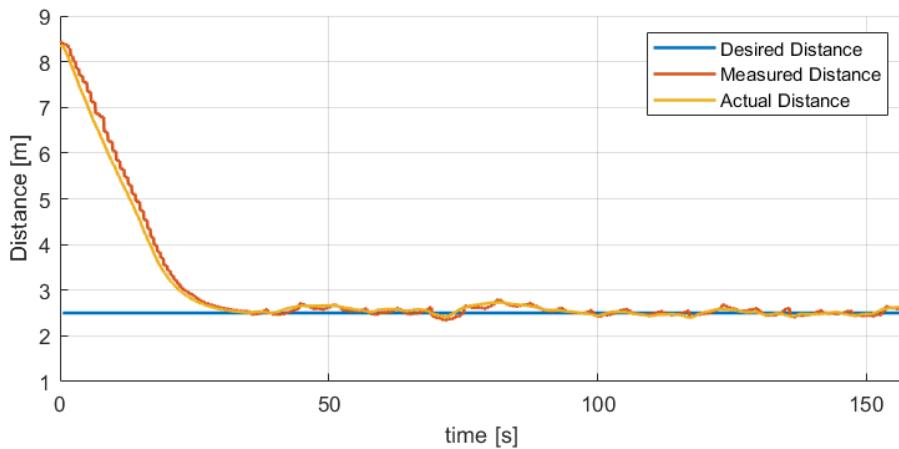
Figure 6.17 illustrates the vehicle's trajectory, revealing its circular path around the net, indicating a consistent behavioral trend observed in previous tests. The vehicle's distance to the net during test T3 is presented in Figure 6.18, showing that the distance converges to the desired value, as in the other tests. However, during the time interval from  $t \approx 70$  s to  $t \approx 90$  s, there is a noticeable deviation from the reference distance. By referring to Figure 6.20, which displays the yaw reference error, it becomes apparent that during this interval, the error revolves around  $-6^\circ$ . Therefore, the deviation from the desired distance can be attributed to the vehicle not aligning directly with the center of the net and maintaining a significant yaw reference error while moving sideways. The measured and actual distance's RMSE are 1.35 m and 1.25 m, respectively, each with a steady-state error of 0.09 m. Furthermore, the yaw reference's RMSE is  $2.50^\circ$ .

In addition, the surge velocity, represented in Figure 6.19, has not had a major influence on the phenomenon during  $t \approx 70$  s to  $t \approx 90$  s, since the values are already small, around 0.04 m/s. Furthermore, the chattering in surge is more significant than in tests T1 and T2 due to the increase in the sway velocity.

The sway velocity and the yaw angle (shown in Figures B.1 and B.2 in Appendix B) exhibit similar patterns to the prior tests, where both follow their references with good performance.



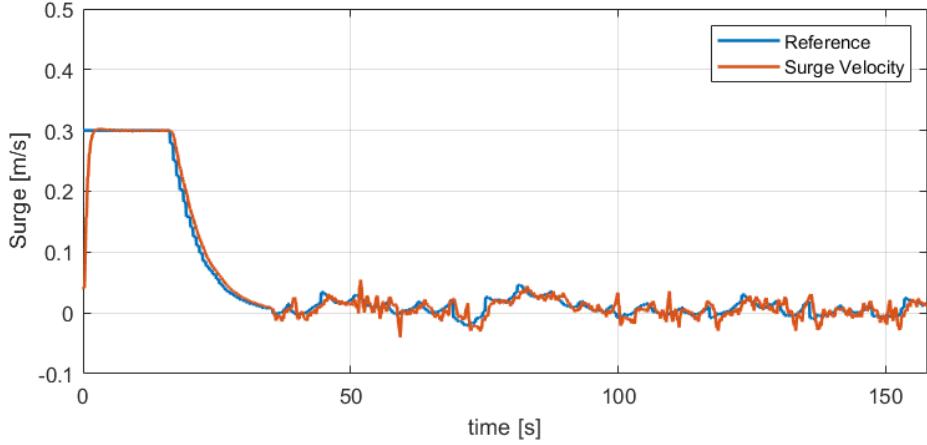
**Figure 6.17:** Trajectory of the vehicle on the  $x$ - $y$  plane while performing test T3 at 0.5 m/s, at 5 m depth, and at 2.5 m from the net.



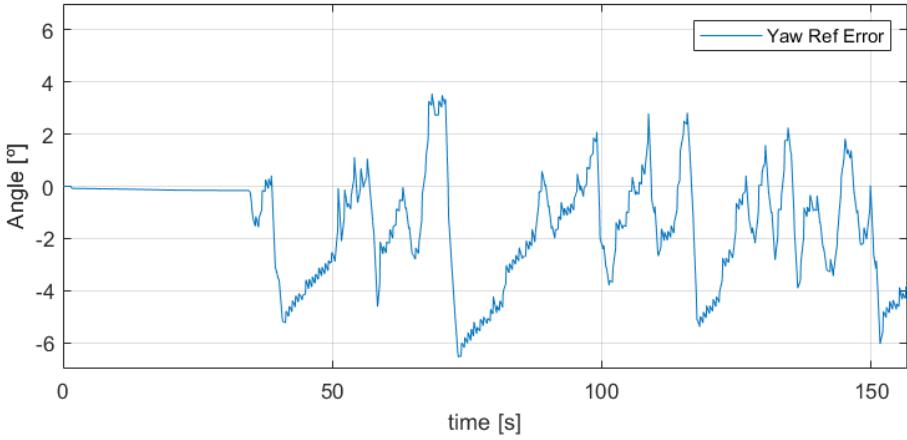
**Figure 6.18:** Distance behavior throughout test T3 at 0.5 m/s, at 5 m depth, and at 2.5 m from the net.

#### T4 - Sway 0.2 m/s Depth 5 m Distance 5 m

To assess the effect of distance on the inspection, test T4 was carried out at a distance of 5 m, with a sway velocity of 0.2 m/s, and at a depth of 2 m, the same velocity and depth as test T2. Figure 6.21



**Figure 6.19:** Surge behavior throughout test T3 at 0.5 m/s, at 5 m depth, and at 2.5 m from the net.



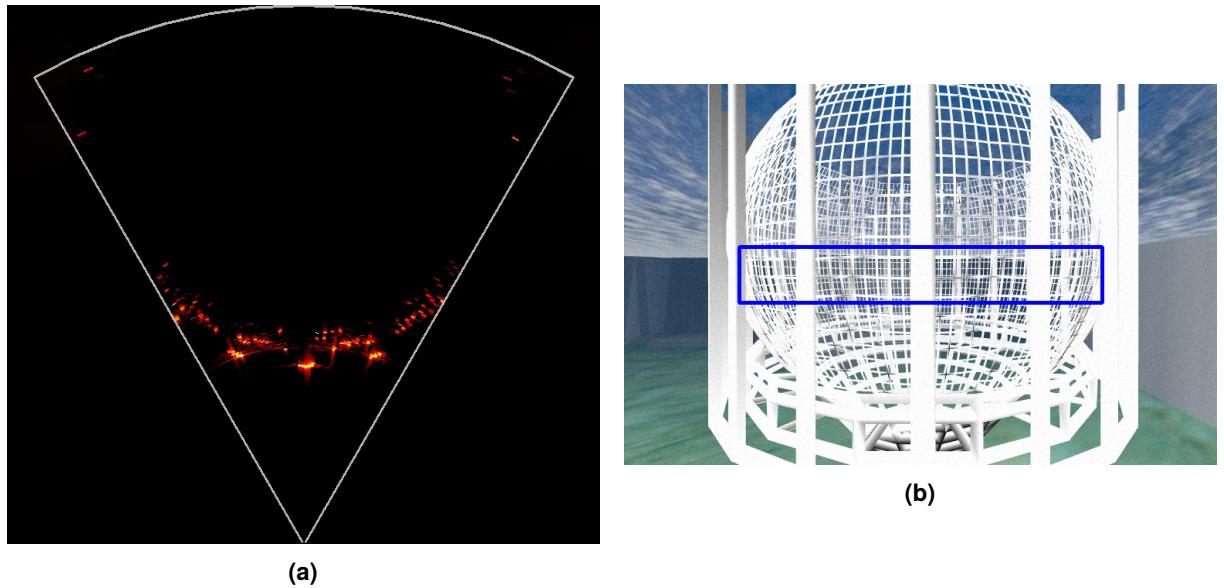
**Figure 6.20:** Yaw reference error throughout test T3 at 0.5 m/s, at 5 m depth, and at 2.5 m from the net.

depicts both sonar image and camera view at a distance of 5 m.

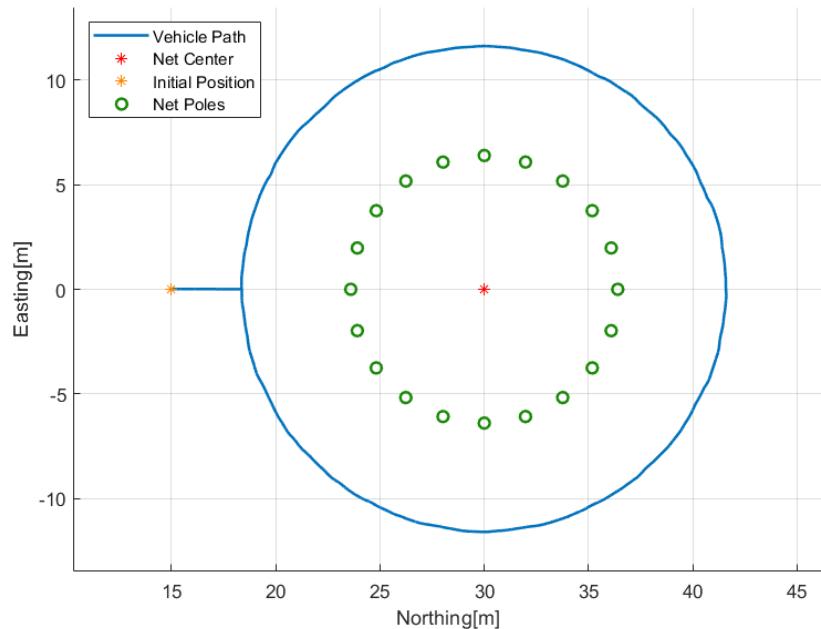
Figure 6.22 shows the vehicle's circular path around the net, corresponding to the outcomes of earlier tests. As Figure 6.23 displays, the distance converges to its reference, with minor oscillations due to the measurements' uncertainty. The measured distance for test T4 had an RMSE of 0.41 m and a steady-state error of 0.06 m, while the RMSE for the actual distance was 0.36 m with a steady-state error of 0.04 m.

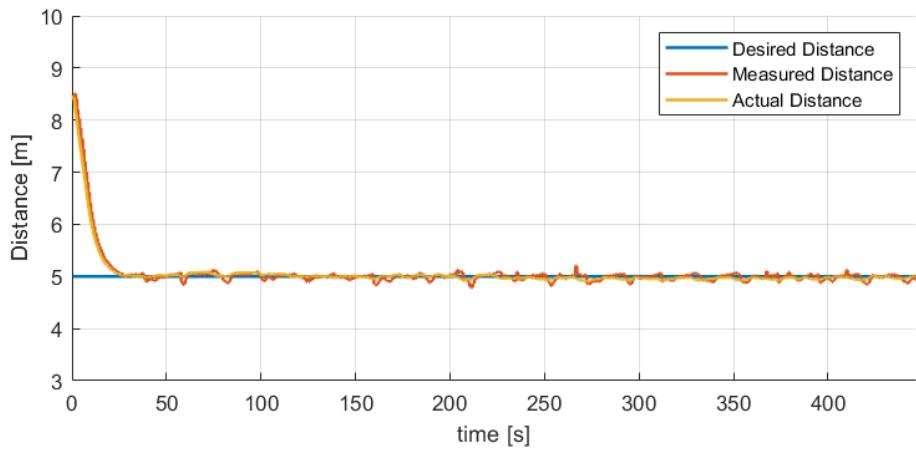
The surge velocity is captured in Figure 6.24, tracking its reference accurately, as the test T2, however, the reference is saturated at 0.3 m/s for a shorter period than test T2 since the vehicle is closer to the desired distance this time. Also, both sway and yaw have the same behavior as the prior tests (represented in Figures C.1 and C.2 in Appendix C). Regarding the yaw reference error showcased in Figure 6.25, it oscillates between  $-4^\circ$  and  $4^\circ$ , having an RMSE of  $1.60^\circ$ .

In general, when comparing tests T2 and T4, there are no significant differences between them.

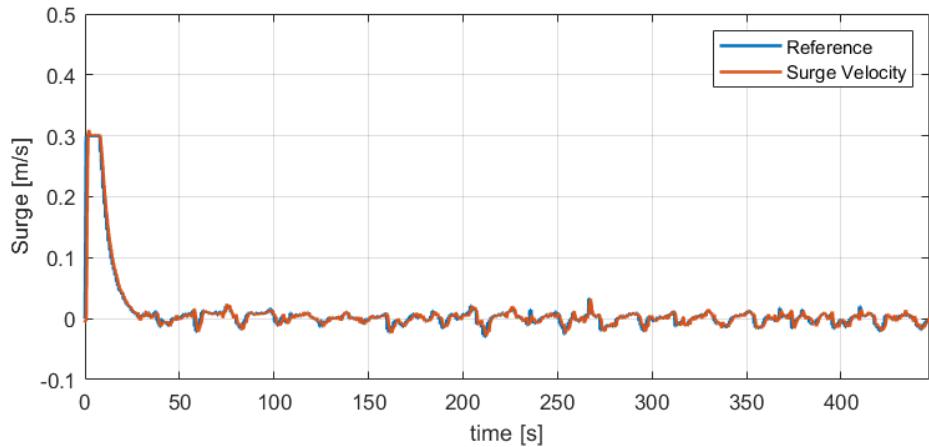


**Figure 6.21:** (a): Sonar image example during test T4. (b): Camera image example during test T4 with the sonar's field-of-view represented by the blue rectangle.

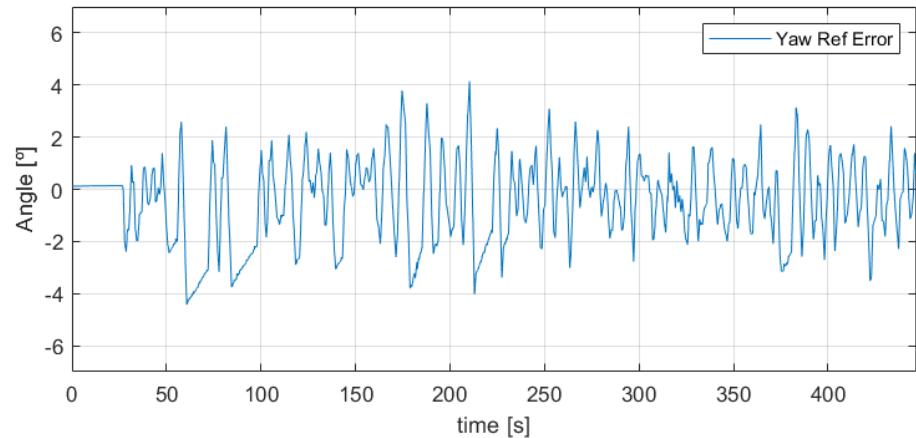




**Figure 6.23:** Distance behavior throughout test T4 at 0.2 m/s, at 5 m depth, and at 5 m from the net.



**Figure 6.24:** Surge behavior throughout test T4 at 0.2 m/s, at 5 m depth, and at 5 m from the net.



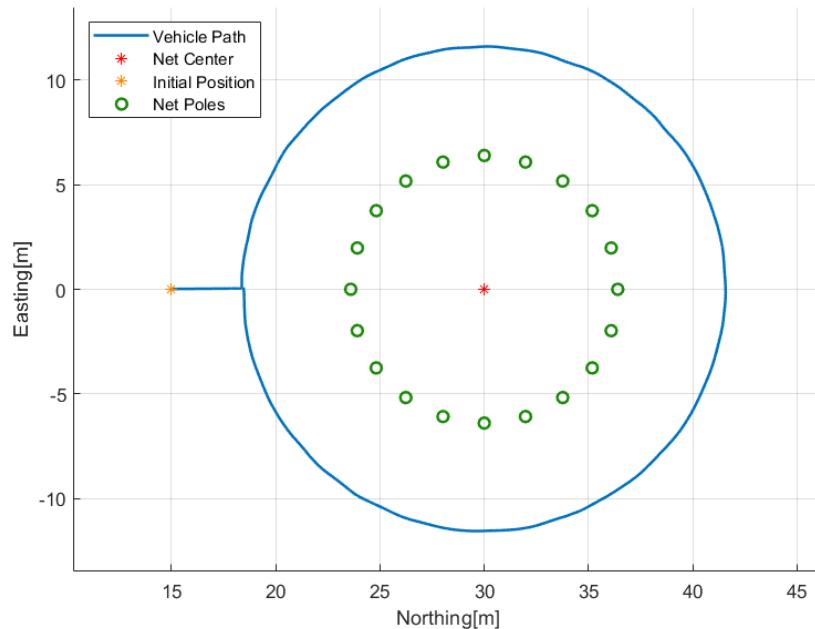
**Figure 6.25:** Yaw reference error throughout test T4 at 0.2 m/s, at 5 m depth, and at 5 m from the net.

## T5 - Sway 0.5 m/s Depth 5 m Distance 5 m

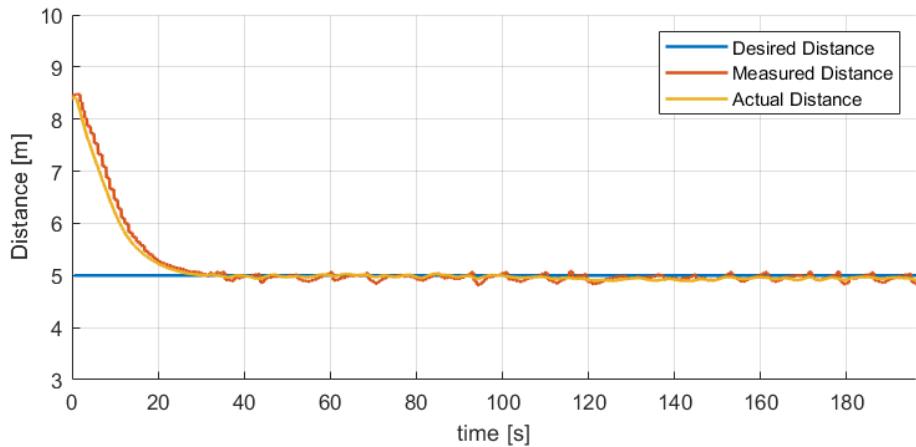
Test T5 was undertaken with a sway velocity of 0.5 m/s, at the same distance and depth as test T4, that is, 5 m.

In Figure 6.26, the  $x$ - $y$  plane vehicle's trajectory around the net is represented. Its trajectory has the expected outcome since it completes a full circle around the net center. Figure 6.27 displays the distance during the test and Figure 6.28 shows the surge velocity. Regarding the distance, it converges to the desired value, with some oscillation due to measurements' uncertainty, as explained previously. In terms of the measured distance, there was an RMSE of 0.62 m and a steady-state error of 0.07 m. As for the actual distance, the RMSE was 0.55 m, with a steady-state error of 0.05 m. Concerning the surge velocity, it has a similar behavior as the previous tests, with chattering still present, as shown in Figure 6.28.

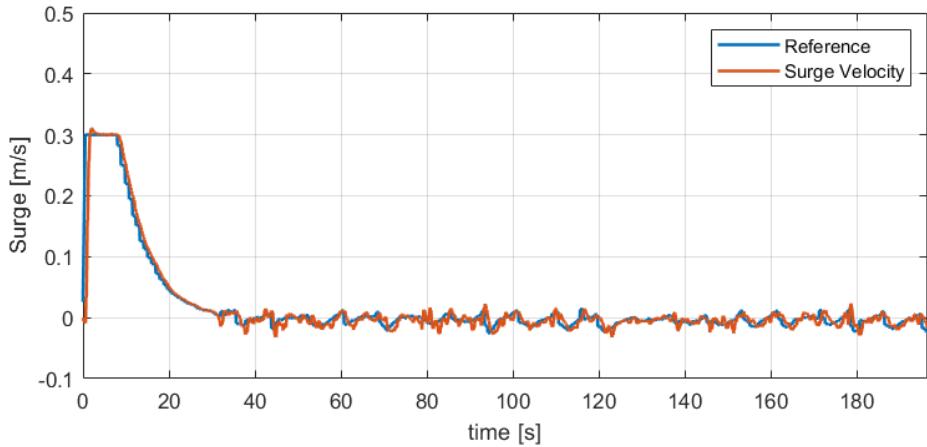
The sway velocity has the same behavior as the previous test, where the vehicle can reach and converge to the desired velocity with no overshoot. For the yaw, the same applies and the yaw rate is higher since the vehicle is moving much faster. Figures D.1 and D.2 in Appendix D provide insights into the sway and yaw characteristics. Furthermore, the yaw reference error is portrayed in Figure 6.29, oscillating from  $-3^\circ$  to  $3^\circ$ , while having an RMSE of  $1.13^\circ$ .



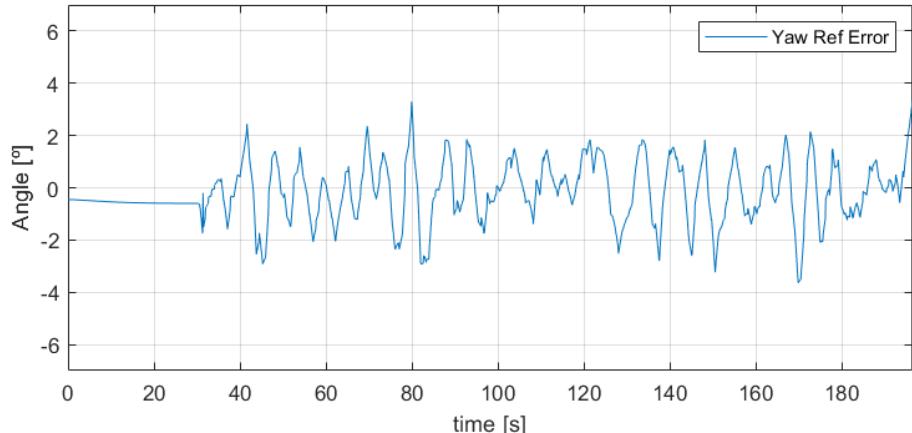
**Figure 6.26:** Trajectory of the vehicle on the  $x$ - $y$  plane while performing test T5 at 0.5 m/s, at 5 m depth, and at 5 m from the net.



**Figure 6.27:** Distance behavior throughout test T5 at 0.5 m/s, at 5 m depth, and at 5 m from the net.



**Figure 6.28:** Surge behavior throughout test T5 at 0.5 m/s, at 5 m depth, and at 5 m from the net.

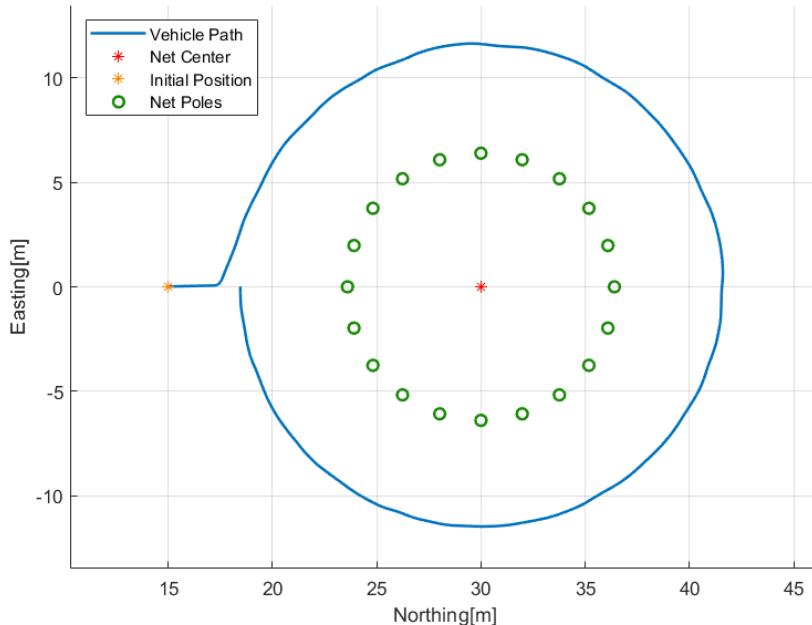


**Figure 6.29:** Yaw reference error behavior throughout test T5 at 0.5 m/s, at 5 m depth, and at 5 m from the net.

## T6 - Smooth Approach

Test T6 was conducted with a vehicle's smooth approach to the net cage. The sway reference is computed according to the distance error and the yaw error, as shown in Equation (5.33). The nominal sway speed  $v_n$  was 0.5 m/s. The  $c_d$  and  $c_\psi$  values are set to 1.0 and 0.0044, respectively. The gain  $c_d$  regulates the importance given to the distance error, that is, the bigger its value, the smaller the error needs to be for the vehicle to start its side movement. Hence, reducing the  $k_d$  will cause a side movement sooner and, because of that, farther from the desired distance. The same reasoning applies to the  $k_\psi$  value for the yaw angle.

Figure 6.30 displays the vehicle's path around the net with a smooth approach. As in the previous tests, the vehicle also makes a circle around the net, but at this time the AUV starts to move earlier around the net, starting to have sway velocity at the point  $[17.5, 0]^T$  m.



**Figure 6.30:** Trajectory of the vehicle on the  $x$ - $y$  plane while performing test T6 at 0.5 m/s, at 5 m depth, and at 5 m from the net.

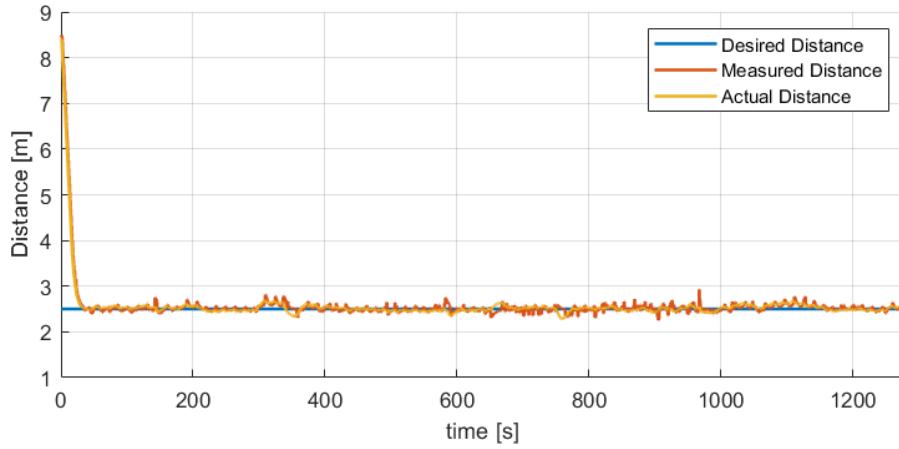
Moving forward, Figure 6.33 illustrates the sway behavior. The sway reference changes from 0 to 0.1 m/s at  $t \approx 10$  s and slowly increases until it reaches the defined nominal speed  $v_n = 0.5$  m/s. Also, checking it simultaneously with Figure 6.31, which depicts the distance over time, one can also remark that the distance error is approximately 1 m at the time instant  $t \approx 10$  s, which corroborates that the side movement starts before the vehicle reaches the desired distance. Additionally, the distance converges to the reference value with minor oscillations around it. The measured distance exhibits an RMSE of 0.65 m, along with a steady state error of 0.08 m. Furthermore, the RMSE for the actual distance is 0.57 m, with a steady-state error of 0.08 m.

Furthermore, Figure 6.34 depicts the RPM command from each of the horizontal thrusters during test T5 (Figure 6.34a) and T6 (Figure 6.34b). As can be seen, the actuation for test T5 in time instant  $t \approx 30$  s

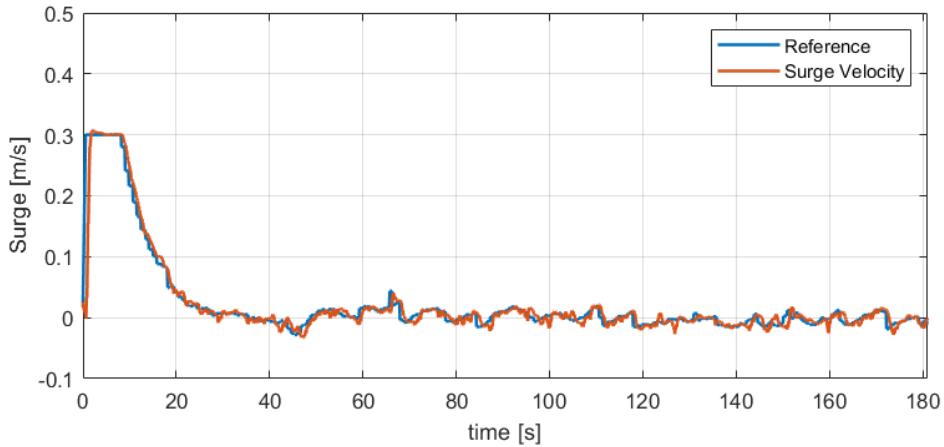
has a peak, which coincides with the sudden change in the sway reference. On the other hand, for test T6, since the sway reference gradually increases since time instant  $t \approx 10$  s, the thrusters actuation does not have a sudden peak when the reference reaches 0.5 m/s. This way, with a smooth approach, we can have a more controlled actuation.

Figure 6.32 exhibits the surge velocity in test T6. It can closely track the reference. It is worth noting that in comparison to other tests, the occurrence of slight chattering starts earlier. The sway and yaw rate values being different than zero earlier can explain the earlier chattering.

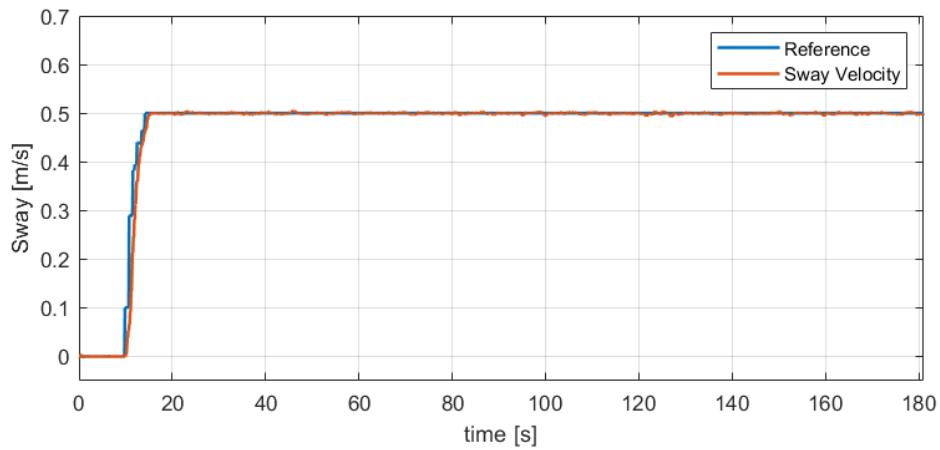
In addition, the yaw angle still effectively tracks the reference, clearly visible in Figure D.2 in Appendix D. Also, the yaw reference error is illustrated in Figure 6.35, with oscillations from  $-4^\circ$  to  $4^\circ$  and an RMSE value of  $1.76^\circ$ .



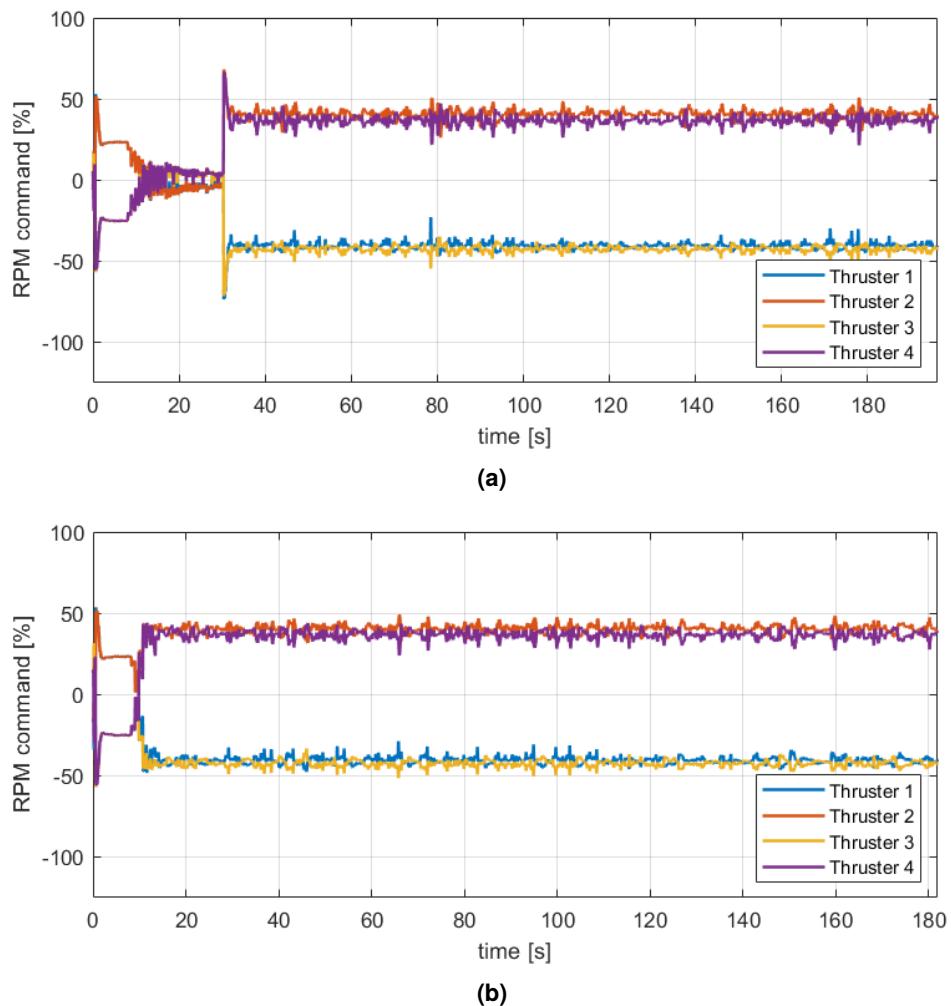
**Figure 6.31:** Distance behavior throughout test T6 with a smooth approach.



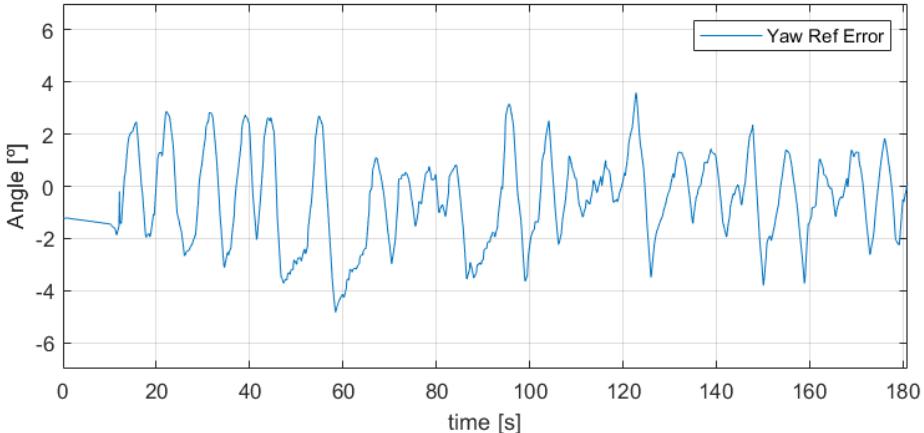
**Figure 6.32:** Surge behavior throughout test T6 with a smooth approach.



**Figure 6.33:** Sway behavior throughout test T6 with a smooth approach.



**Figure 6.34:** Thrusters RPM command throughout: (a): Test T5; (b): Test T6.



**Figure 6.35:** Sway behavior throughout test T6 with a smooth approach.

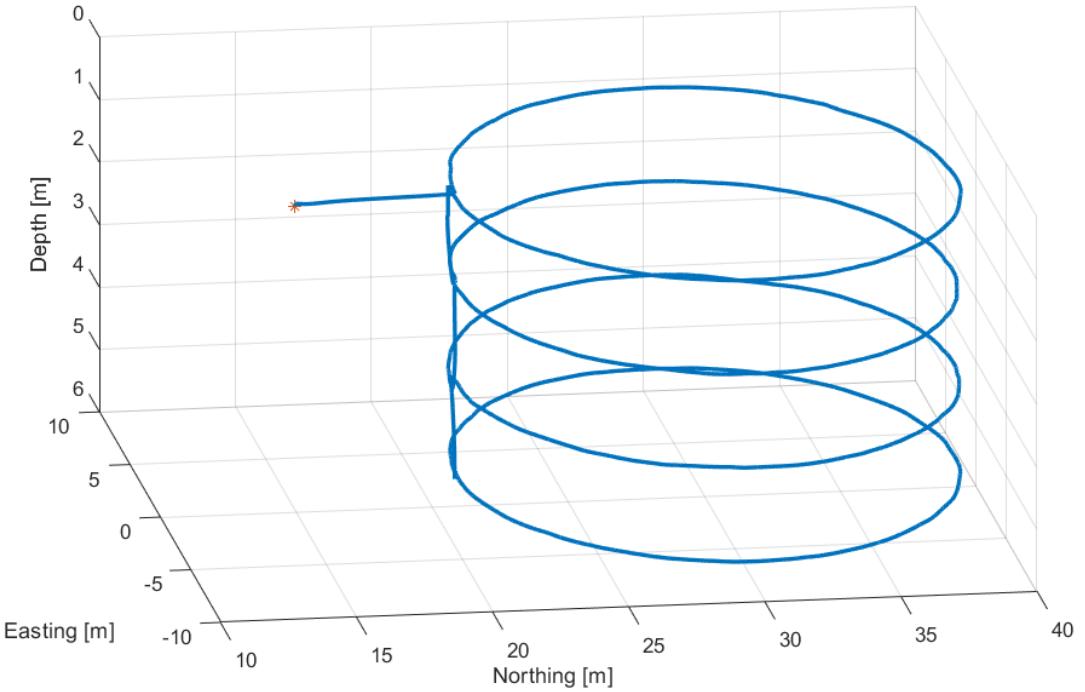
## T7 - Full Inspection

The previous tests were fulfilled at a constant depth. To conduct a comprehensive inspection of the net, it is necessary to perform a test that covers different depths. During the inspection, the vehicle maintains a nominal sway speed of 0.2 m/s and a desired distance of 2.5 m from the net. This allows the vehicle to proceed at a slow speed and stay close enough to the net to analyze its conditions. The vehicle is equipped with a camera with a Vertical Field-of-View (VFOV) of 64°. Therefore, the height of the net that can be covered at a distance of 2.5 m can be calculated as follows

$$\text{height} = 2 \tan\left(\frac{64^\circ}{2}\right) d = 2 \tan\left(\frac{64^\circ}{2}\right) \times 2.5 \approx 3.12 \text{ m.} \quad (6.14)$$

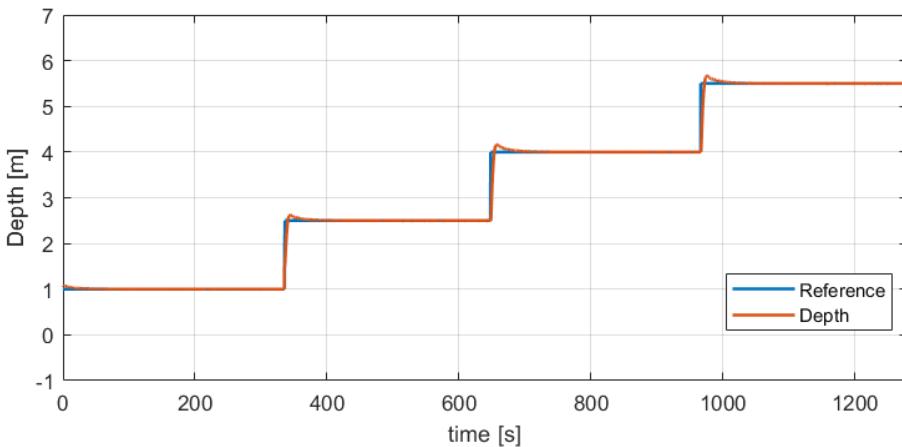
With that in mind, for the vehicle to cover the entirety of the net, which has a respective height of approximately 7 m, the test was conducted at the following depths: 1.5, 2.5, 4, and 5.5 m. In this way, we ensure sufficient coverage and overlap of the images for different depths.

In Figure 6.36, the path of the AUV is illustrated. As seen in this figure, the vehicle approaches the net and moves around, as before, but in this case, after completing one lap, it changes its depth and does another lap around the infrastructure, until covering all the net.



**Figure 6.36:** 3D trajectory of the vehicle in the full inspection test.

Figure 6.37 shows the depth behavior throughout the whole test. When the reference changes, the vehicle follows it, having a minor overshoot. Nonetheless, the depth converges without any oscillations around the reference. Also, it is noticeable that each lap around the net lasts roughly the same since each step has approximately the same duration.

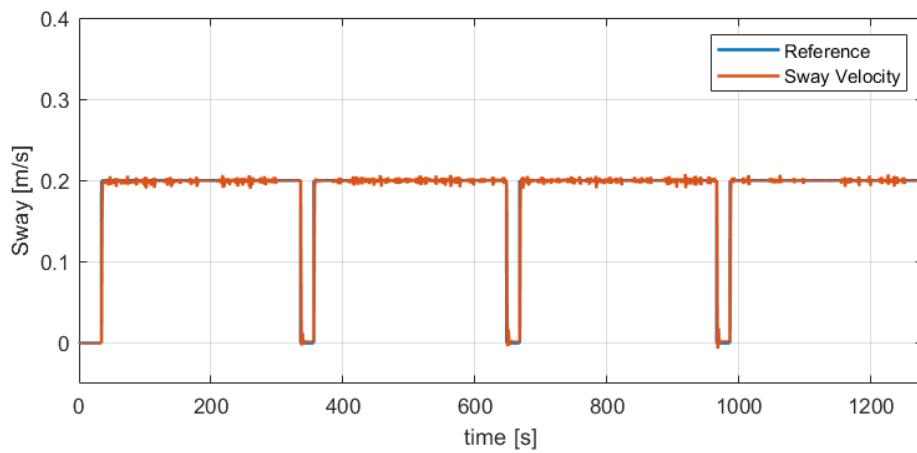


**Figure 6.37:** Depth behavior throughout the full inspection test.

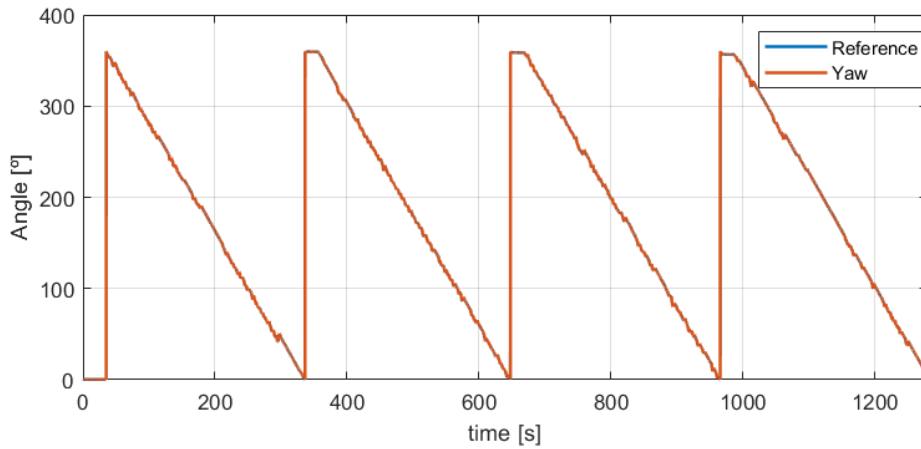
In addition, it is worth mentioning that the side movement stops when the depth reference changes, as can be seen in Figure 6.38 that the sway velocity goes to zero at the instant  $t \approx 350$  s for around 20 seconds until the depth is close enough from the reference. Figure 6.39 presents the yaw angle evolution through the mission. It also stays constant, around  $360^\circ$ , for about 20 seconds at  $t \approx 350$  s denoting the descending phase. Additionally, the yaw angle goes through all angles in the  $[0, 360]^\circ$  range four times

since it does four laps around the net cage.

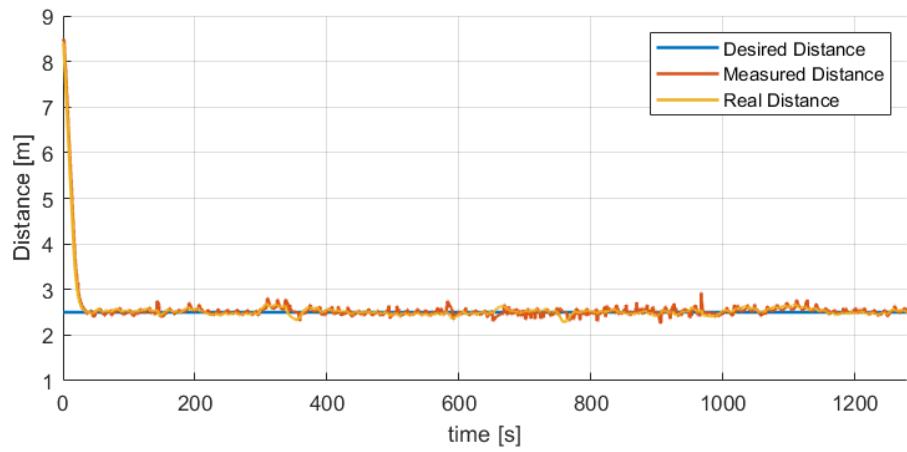
For further analysis, the vehicle's distance to the net and the surge velocity throughout the mission are shown in Figures 6.40 and 6.41, respectively. The distance after converging to the desired value, has some minor oscillations justified by measurement errors. The measured distance has an RMSE of 0.48 m and a steady-state error of 0.07 m, while the actual distance has an RMSE of 0.45 m and a steady-state error of 0.06 m. It is important to notice that from around instant  $t \approx 300$  s to  $t \approx 350$  s, the distance moves away from the reference. Examining Figure 6.42, which illustrates the yaw reference error, it is observed that the error remains approximately  $8^\circ$  within that interval. Since the vehicle is not perfectly aligned with the net cage's center, it departs from the distance's reference. When the yaw reference error reaches  $-4^\circ$  the distance converges to its reference. The RMSE of the yaw reference is  $3^\circ$ .



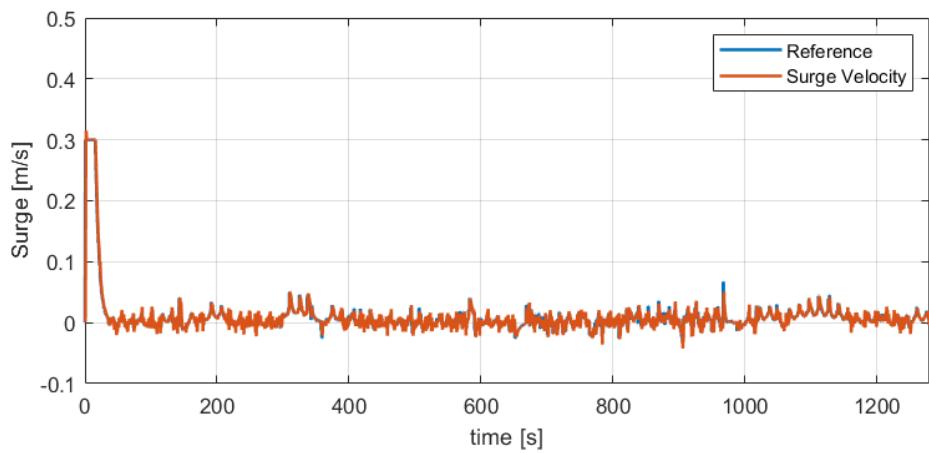
**Figure 6.38:** Sway behavior throughout the full inspection test.



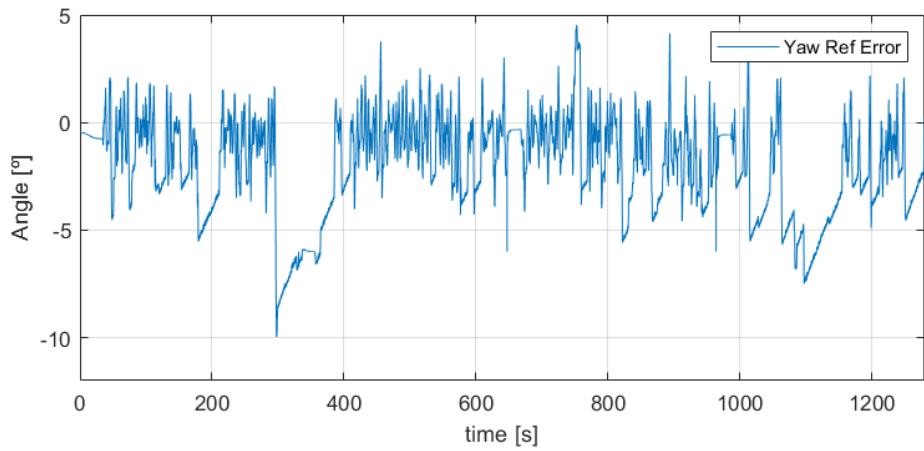
**Figure 6.39:** Yaw behavior throughout the full inspection test.



**Figure 6.40:** Distance behavior throughout the full inspection test.



**Figure 6.41:** Surge behavior throughout the full inspection test.



**Figure 6.42:** Yaw Reference Error throughout the full inspection test.

## T8 and T9 - Currents

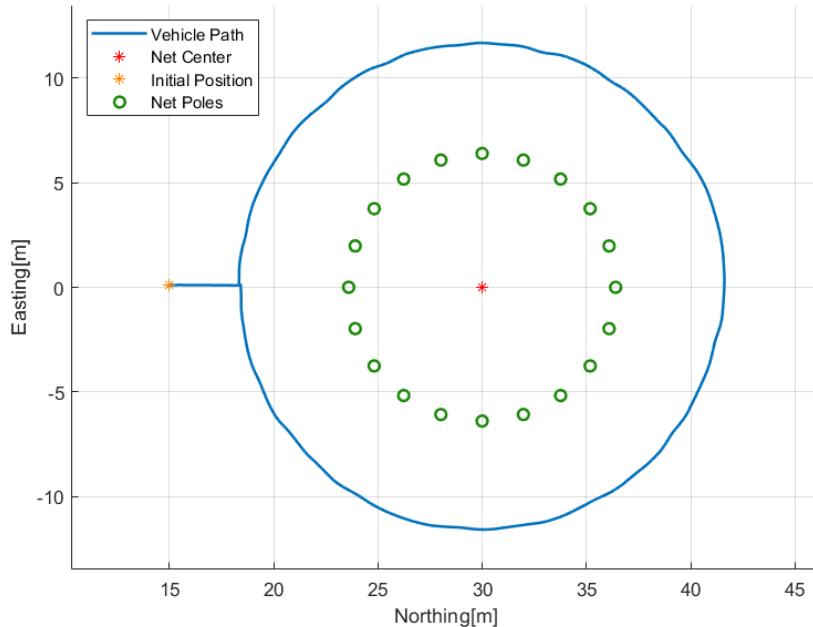
To test the controller's robustness in the presence of disturbances, a test was made with a constant irrotational current velocity defined in the inertial frame by  ${}^U \mathbf{v}_c = [v_{cx}, v_{cy}^T]$ . The current speed used in the test T8 was  ${}^U \mathbf{v}_c = [0, 0.1]^T$  m/s. From now on, this test will be addressed as T8.

For this test, the depth remained constant at  $z = 5$  m and the desired distance also at 5 m. Furthermore, the nominal sway speed was 0.5 m/s.

The vehicle's trajectory in the  $x - y$  plane is represented in Figure 6.43. Although the currents are constant and irrotational, since the vehicle adjusts the yaw as it moves, the currents hit the vehicle in all directions throughout the movement. To verify this statement, the velocity of the current in the body frame is illustrated in Figure 6.44, where  $u_c$  is the current component on the surge velocity and  $v_c$  the current component on the sway velocity. As can be seen, both velocities have a sinusoidal form and can also be represented by

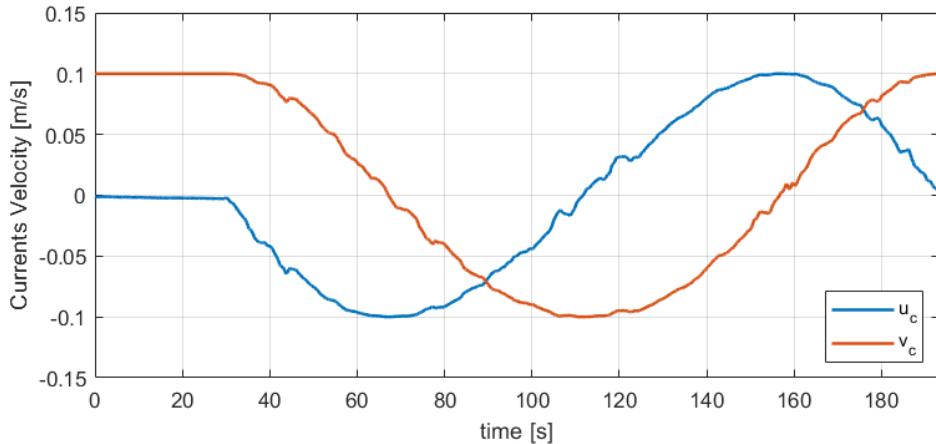
$${}^B \mathbf{v}_c = {}_B R^T(\psi) {}^U \mathbf{v}_c, \quad \begin{cases} u_c = v_{cx} \cos \psi + v_{cy} \sin \psi \\ v_c = -v_{cx} \sin \psi + v_{cy} \cos \psi \end{cases}, \quad (6.15)$$

where  ${}^B \mathbf{v}_c = [u_c, v_c]$  is the current velocity in the body frame.



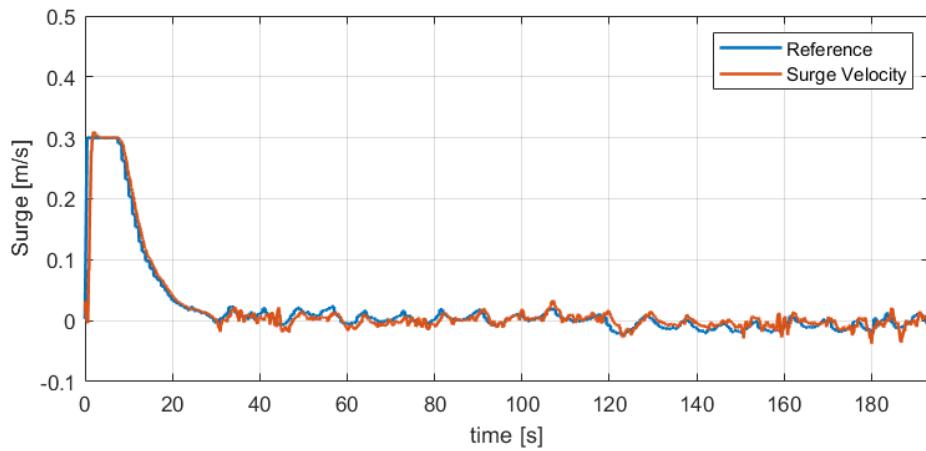
**Figure 6.43:** Trajectory of the vehicle on the  $x$ - $y$  plane during test T8 in the presence of a 0.1 m/s current.

In Figures 6.45 and 6.46, both surge and sway velocities can be observed throughout the inspection. Analyzing the surge velocity, around the time instants  $t \approx 50$  s and  $t \approx 140$  s, it has a small error due to the current value being at its max absolute value. Secondly, examining the sway velocity, until  $t \approx 30$  s, it has difficulty in converging to the reference, since  $v_c$  is 0.1 m/s. Thereafter, its reference changes to 0.5 m/s, and the velocity have a significant overshoot, since the current has the same direction as the movement. Furthermore, one can inspect that the sway is never capable of converging to the reference,

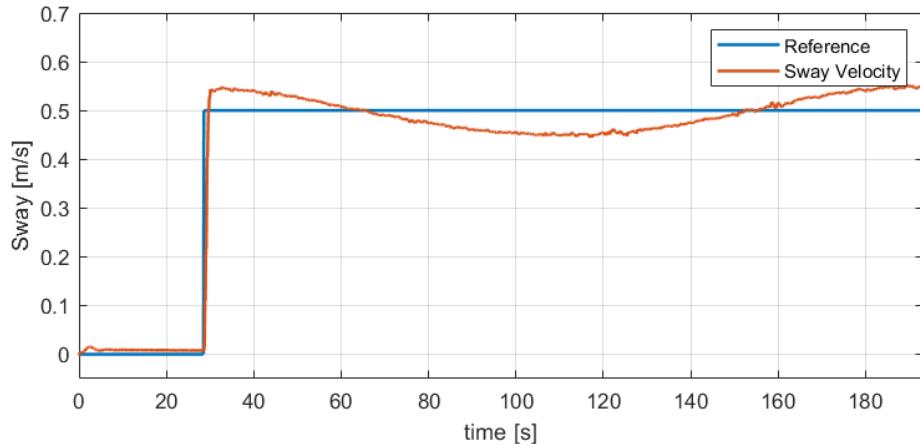


**Figure 6.44:** Current components on the body frame in Test T8.

and also, the sway error is positive when the  $v_c$  value is positive and vice-versa, making sway oscillate around 0.5 m/s with a peak-to-peak error of approximately 0.1 m/s



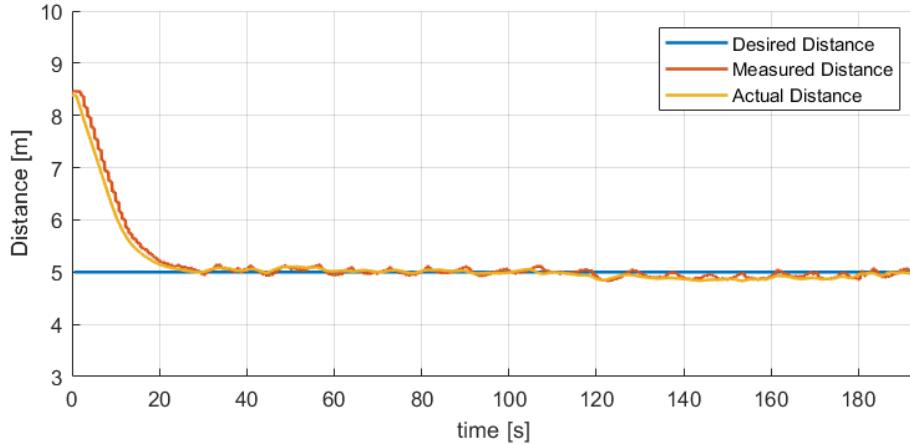
**Figure 6.45:** Surge behavior throughout test T8 in the presence of a 0.1 m/s current.



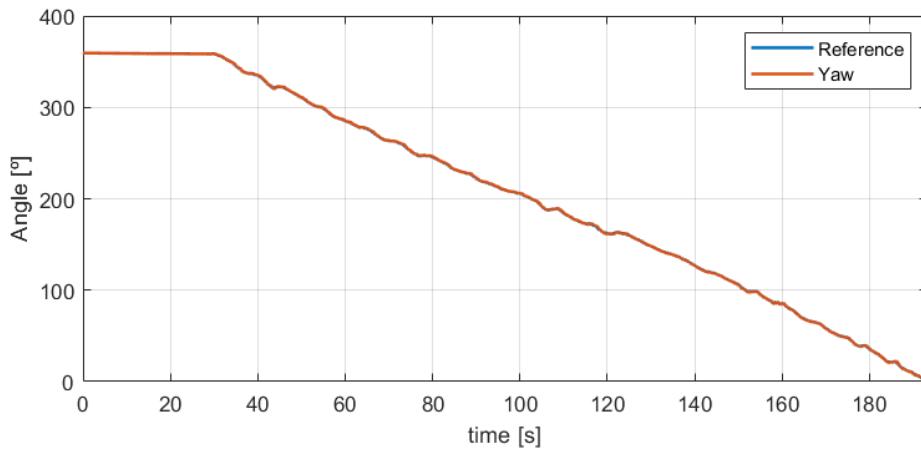
**Figure 6.46:** Sway behavior throughout test T8 in the presence of a 0.1 m/s current.

For further analysis, both Figures 6.47 and 6.48 display the distance and yaw course over time for test

T8. The measured distance had an RMSE of 0.63 m and a steady state error of 0.07 m, while the RMSE for the actual distance was 0.55 m with a steady-state error of 0.08 m.



**Figure 6.47:** Distance behavior throughout test T8 in the presence of a 0.1 m/s current.



**Figure 6.48:** Yaw behavior throughout test T8 in the presence of a 0.1 m/s current.

Finally, in order to address the issues related to the inability of surge and sway velocities to reach their desired values when currents are present, it was conducted an additional test T9. This test involved modifying the controller parameters, which are listed in Table 6.3.

The process behind the solution was to increase the integral gains of the three controllers: distance, surge, and sway. The integral term of the controllers is capable of mitigating the effect of the disturbances. This happens because the integral controller responds to accumulated past error, and, so, when there is a disturbance, it opposes it.

**Table 6.3:** Controllers New Parameters

	Distance - $d$	Surge - $u$	Sway - $v$
$K_P$	0.2	40	50
$K_I$	0.0005	5.0	7.0

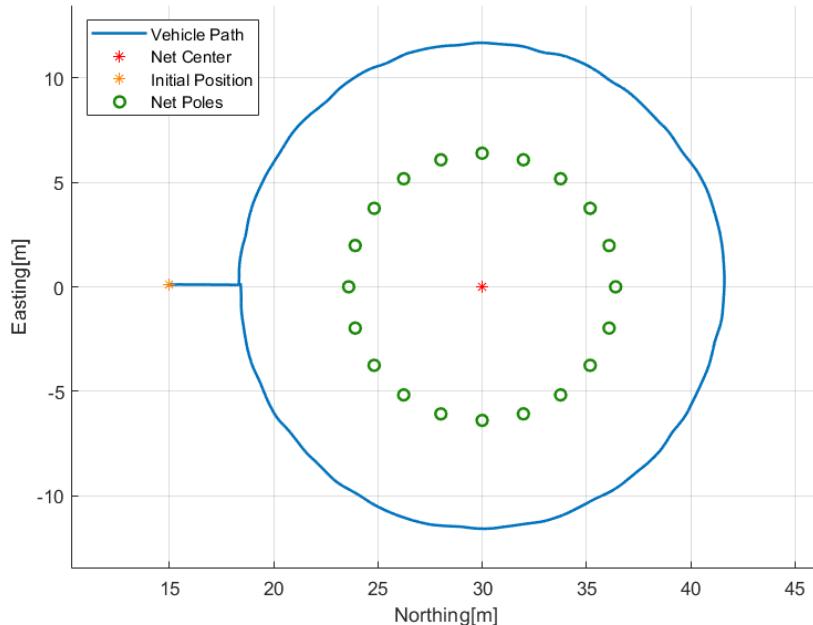
Figure 6.49 presents the trajectory of the vehicle in the presence of 0.1 m/s currents with the new

parameters. Setting it side by side with the previous test, in Figure 6.43, the trajectory has roughly no considerable differences.

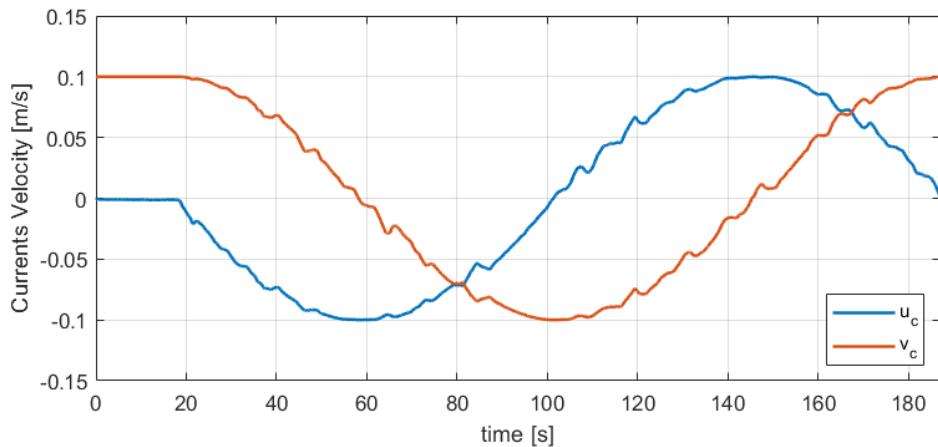
Considering the surge velocity represented in Figure 6.51, one can notice that it can reach the reference better regarding the last test. However, with the increase of the integral gain, the surge consequently gains an overshoot.

Next, through analysis of the sway speed pictured in Figure 6.52, it is clear that the sway improved in resisting the current and converging to the desired value. The oscillation around the reference is approximately 0.04 m/s peak-to-peak, which is an improvement. Also, the current component  $v_c$ , shown in Figure 6.50, does not have an immediate effect as the previous test, i.e., at  $t \approx 100$  s, when  $v_c$  is at its minimum, the sway velocity is at the reference. This shows that the current disturbance is being canceled by the integral term.

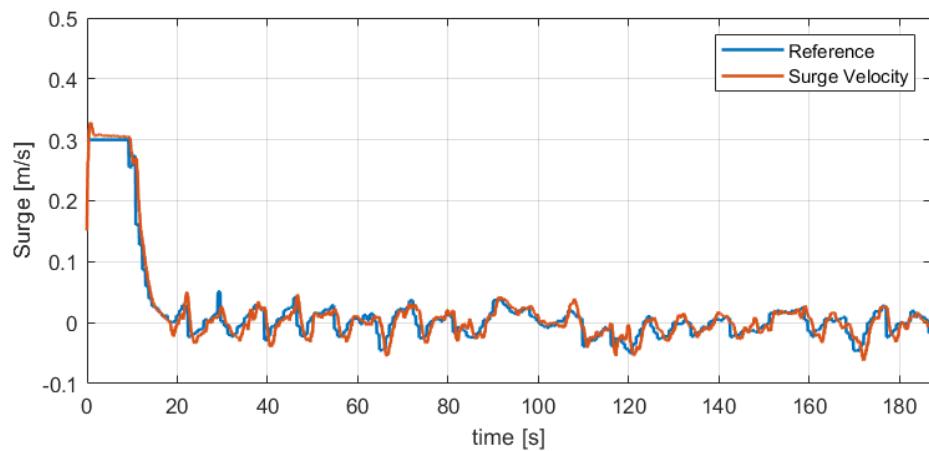
For deeper examination, both Figures 6.53 and 6.54 display the distance and yaw course over time for test T9. Furthermore, the measured distance had an RMSE of 0.61 m and a steady-state error of 0.08 m, while the RMSE for the actual distance was 0.54 m and with a steady-state error of 0.07 m.



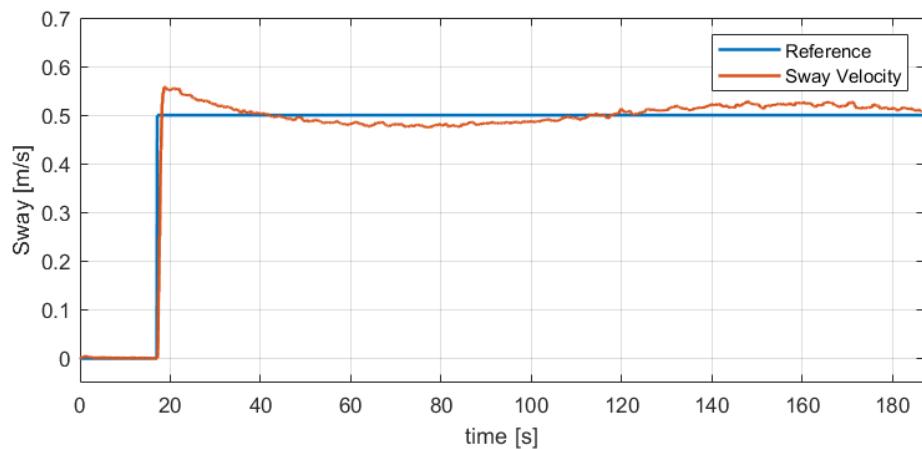
**Figure 6.49:** Trajectory of the vehicle on the  $x$ - $y$  plane during test T9 in the presence of a 0.1 m/s current with new controllers parameters.



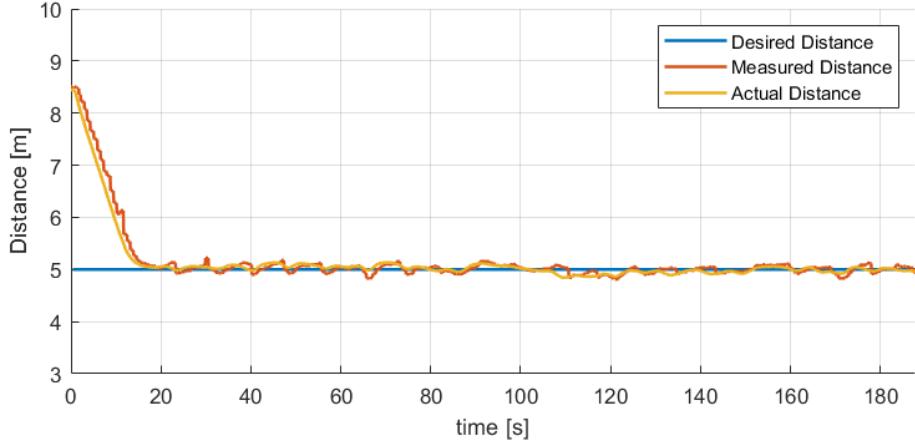
**Figure 6.50:** Current components on the body frame during test T9, with new parameters.



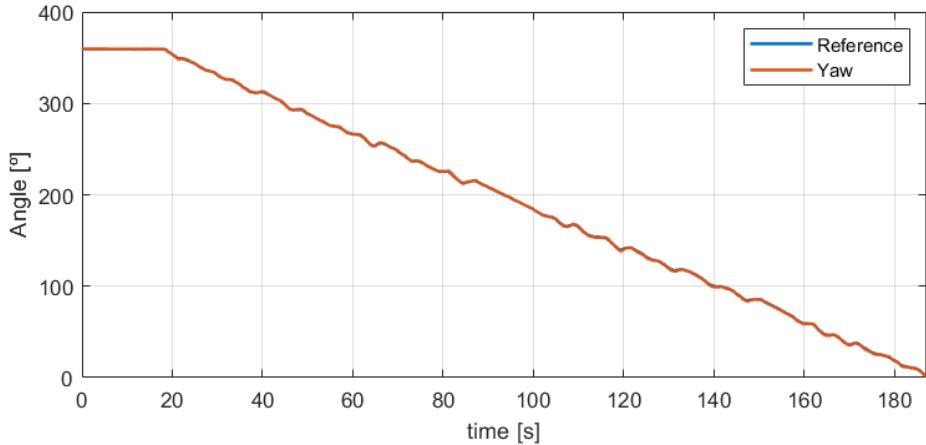
**Figure 6.51:** Surge behavior throughout test T9 in the presence of a 0.1 m/s current with new controllers parameters.



**Figure 6.52:** Sway behavior throughout test T9 in the presence of a 0.1 m/s current with new controllers parameters.



**Figure 6.53:** Distance behavior throughout test T9 in the presence of a 0.1 m/s current with new controllers parameters.



**Figure 6.54:** Yaw behavior throughout test T9 in the presence of a 0.1 m/s current with new controllers parameters.

### 6.2.3 Inspection Tests Summary

This section summarizes the results obtained from the aquaculture net inspection tests. This summary presents an overview of the comparative performance of tests T1 to T6. The tests were compared by evaluating the Root Mean Squared Error (RMSE) of certain variables, namely the measured distance  $d$ , the actual distance  $d_{actual}$  and the desired yaw  $\psi_d$ , displayed in Table 6.4.

Firstly, comparing tests T1 and T2, there is a variation of about 4 cm in the distance error. However, this value is not significant in the experimental context. This result was expected since the depth changes between tests T1 and T2, while the sway speed was set at 0.2 m/s and the desired distance 2.5 m for both. Regarding the yaw reference error, its variation between both tests is roughly  $0.35^\circ$ , which is almost null. Therefore, depth is not a determinant factor for the inspection.

Now, looking at tests T2 and T3, where there was an increase in speed from 0.2 (T2) to 0.5 m/s (T3), it is noticeable that the distance error rises about 45 cm. This can be explained by the fact that at a higher speed, the vehicle must turn more quickly, that is, have a higher yaw rate. However, the sonar detection

update rate remains the same, which can lead to a higher chance that the yaw reference has a larger error and therefore the vehicle departs from the desired distance. The RMSE for the yaw reference in T3 is  $0.30^\circ$  away from T2. Even though the yaw error variation is not significant, since the vehicle is moving at a considerable speed, the impact of this yaw reference is substantial in the distance. For example, in one second, the vehicle in T2 would move 20 cm, while 50 cm in T3.

When examining tests T4 and T5, carried out at speeds of 0.2 and 0.5 m/s, respectively, while maintaining a constant distance and depth of 5 m for both, the distance error experiences an approximate increment of 20 cm from T4 to T5. This increase in error demonstrates a similar trend to the results observed in tests T2 to T3 when the velocity is increased. Regarding the yaw reference error, it decreases from T4 to T5, in contrast with the comparison of T2 and T3.

Next, we compare the results of tests T5 and T6. Both experiments were conducted under identical conditions, including speed, depth, and distance. The only difference lies in the approach method, with T6 employing a smoother approach by initiating lateral movement earlier. The distance RMSE values for both tests are nearly identical, as expected given the identical conditions. Regarding the yaw reference error, the difference between both tests is roughly  $0.6^\circ$ , which is not relevant.

**Table 6.4:** RMSE values for tests T1 and T6

Test	$d_{actual}$ RMSE [m]	$\psi_d$ RMSE [ $^\circ$ ]
T1	0.79	1.96
T2	0.83	2.21
T3	1.25	2.50
T4	0.36	1.60
T5	0.55	1.13
T6	0.57	1.76

Table 6.5 shows the steady-state error of the distance from test T1 to T6. In general, the errors are in the order of magnitude of  $10^{-2}$  m, which indicates that the vehicle moves at the desired distance with good performance for all tests. Furthermore, the steady-state error has the same tendency as the RMSE, since it increases when the speed increases, even if barely. Comparing tests conducted at the same speed but with different distances, T2 and T4, or, T3 and T5, we can observe that the desired distance does not influence the distance's steady-state error.

**Table 6.5:** Distance Steady-State Error for tests T1 and T6

Test	$d_{actual}$ [m]
T1	0.03
T2	0.04
T3	0.09
T4	0.04
T5	0.05
T6	0.08

Finally, to analyze the performance of tests T8 and T9, both conducted in the presence of currents, the errors regarding the distance and the velocity are presented in Table 6.6. Initially focusing on the distance error, there is no significant difference between the two tests. However, when examining the velocity error, a more noticeable disparity emerges. While the difference in surge velocity is not significant, there is a clear distinction in the sway velocity of test T9. The variation in sway is approximately 0.01 m/s. By referring to Figures 6.46 and 6.52, it is apparent that the controllers with new parameters in test T9 perform better in the presence of currents. Concluding, tests T8 and T9 exhibit no distance error differences, but T9 demonstrates slightly better performance in terms of velocity.

**Table 6.6:** RMSE and  $K_I$  values for tests T8 and T9

<b>Test</b>	$d_{\text{actual}}$		$u$		$v$	
	RMSE [m]	$K_I$	RMSE [m/s]	$K_I$	RMSE [m/s]	$K_I$
T8	0.55	0.0001	0.02	0.42	0.04	0.42
T9	0.54	0.005	0.02	5.0	0.03	7.0

Comparing the RMSE from the actual distance and the measured distance, presented in Table 6.7, we can observe that the measured distance had the same behavior and similar values to the actual distance, indicating a decent accuracy in the distance measurements.

**Table 6.7:** RMSE for the measured and actual distance

<b>Test</b>	T1	T2	T3	T4	T5	T6	T7	T8	T9
$d$ RMSE [m]	0.85	0.89	1.35	0.41	0.62	0.65	0.48	0.63	0.61
$d_{\text{actual}}$ RMSE [m]	0.79	0.83	1.25	0.36	0.55	0.57	0.45	0.55	0.54

In conclusion, the higher the inspection velocity, the more significant the distance error is. Concerning the yaw reference angle, it seems that neither the distance nor the velocity exerts a significant influence on this value, as it primarily depends on the accuracy of estimating the net center from the sonar image. Nevertheless, a significant yaw reference error can result in the vehicle deviating from the desired distance, as it is not directly aligned with the center. Overall, across all the tests conducted, the vehicle demonstrated satisfactory performance.

## Chapter 7

# Conclusions and Future Work

This work addressed the development of an autonomous method for inspecting Aquaculture Nets using an Autonomous Underwater Vehicle (AUV) equipped with a Forward-Looking Sonar (FLS). The FLS was responsible for providing acoustic images of the net cage. The acoustic images were then processed, giving measurements about the vehicle's distance and orientation relative to the net.

Firstly, a comprehensive examination of the current state of the art was conducted, including an analysis of various methods and strategies for Autonomous Aquaculture Net Inspection, along with several techniques for sonar image detection. Secondly, the fundamental notation and reference frames used in the vehicle's model were elucidated, with a particular emphasis on the kinematics and dynamics of a general AUV.

As mentioned previously, the FLS produces acoustic images that represent the environment in front of the vehicle. These images needed to be processed to get any relevant information. Foremost, image processing techniques, namely bilateral and binary filters, were applied for noise removal. Then, both the vehicle's distance and the orientation relative to the net were measured. To measure the relative orientation, the net cage's center on the image was estimated.

The center estimation on the image was conducted by doing a circle regression fitting of the net cage data points, which were extracted from the image processing techniques. Furthermore, the net cage poles had significant importance on the estimation's validity, depending on the vehicle's distance to the net. Therefore, the poles were identified to improve the center estimation. With the estimated center, the vehicle's orientation relative to the aquaculture net cage was trivial to compute. The distance from the sonar to the nearest blob was measured in pixels, and this value was then converted to meters to calculate the distance to the net.

With the measurements of both relative distance and orientation, velocity, and orientation references were calculated. Following those references, the vehicle could circulate the net cage, covering the whole net cage's perimeter.

Finally, several realistic simulations were conducted to test the proposed solution's robustness. Those simulations were performed using ROS and the Gazebo Simulator. Initially, some tests were realized to examine the influence of distance, depth, and velocity on the inspection. Subsequently, to simulate a realistic inspection scenario, the vehicle moved continuously around the net for various depths. Then,

simulations with currents were executed to check the vehicle's behavior in the presence of external disturbances. The obtained results showed that the vehicle can inspect the net cage with a satisfactory degree of robustness, having a slight error due to the measurements' inaccuracies and modeling assumptions.

## 7.1 Future Work

Despite the satisfactory results obtained, there are still some unresolved issues that require further attention. This section will discuss and address these issues while providing suggestions for improvement.

Firstly, there are some issues regarding the net cage center's estimation. A drawback of the used method is that an initial guess for the first circle regression must be given, and an inaccurate guess can compromise the result. To address this issue, we used the initial position of the vehicle and the net cage center's coordinates to compute the initial guess for the first circle regression. One option for not using the vehicle's position would be to initially detect three or more poles. Then, knowing that the center is at the same distance to all the poles, the initial guess could be computed with triangulation. However, this is only reasonable if the poles can be detected accurately.

Secondly, the least-squares algorithm is sensitive to outliers, so an outlier could jeopardize the estimation. Consequently, the existence of unforeseen objects, such as debris or fish, could result in inaccurate estimations. Therefore, one possible solution is to identify objects that are not part of the net cage and not consider them in the regression. An additional concern arises from employing a fixed thresholding approach across all images. To address this, a recommendation is made to implement an adaptive thresholding technique for each individual image. One possible approach is to construct a histogram using all pixel values and subsequently normalize them to determine the optimal threshold value.

Additionally, the current pole detection method is not optimal, being inaccurate, particularly when detecting poles other than the nearest one. To address this issue, one approach is to analyze the intensity and shape of the echo lights. If the echo lights returned by an object have a higher intensity than their surroundings, it suggests the presence of a potential pole. Then, by filtering the detected objects based on their shape, the poles could be identified. However, one must be careful while filtering objects by shape, since objects can have different shapes depending on the viewpoint.

Another challenge arises when the vehicle is near the net, resulting in only detecting the inner net and in the absence of detected poles in the sonar image. In such cases, if the AUV fails to track the poles' position, there is a risk of collision. A potential solution is to track the pole positions based on previous detections and actively avoid them when they are obstructing the vehicle's path. By adopting this strategy, the vehicle can navigate safely and avoid potential collisions with the poles.

The acoustic images provided by the simulated sonar plugin have low resolution. This can result in inaccuracies in the measurements, for example, in the center estimation, to be translated in noticeable deviations, in particular at larger distances from the cage. Therefore, obtaining higher-resolution acoustic images is crucial to achieving higher measurement accuracy. Furthermore, utilizing a Kalman-based filter, such as the Extended Kalman Filter, to estimate the distance and relative orientation to the net would

also enhance the accuracy of the measurements. This is important because the measured distance and yaw values in this study are susceptible to outliers, which can compromise the navigation of the vehicle. By implementing a robust filtering approach, the reliability of the measurements can be significantly increased.

The tuning of the controllers was performed in a try-and-error manner. A improvement for this is to use the Ziegler-Nichols method, which is also an empirical method that finds first the proportional gain that causes the control loop to oscillate indefinitely at steady state. Then, with the period of the oscillations, the integral and derivative gains are computed. In the case of the presence of currents, an observer could be implemented to estimate their velocity and then apply the necessary forces to oppose them.

# Bibliography

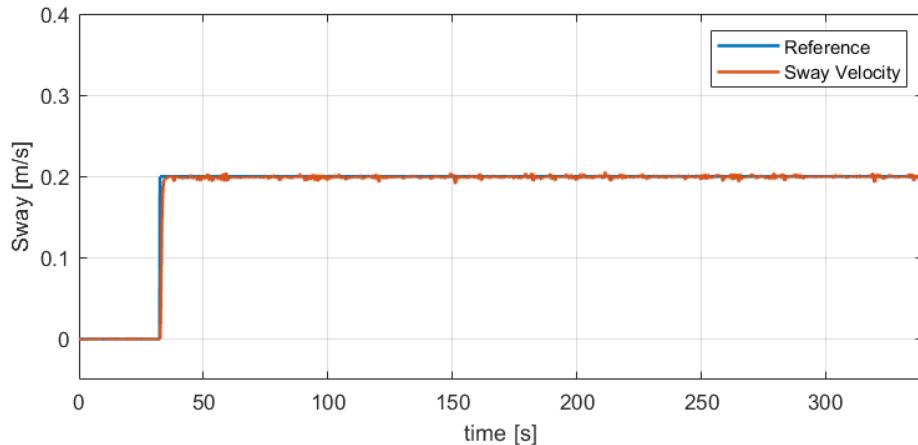
1. Of the United Nations. Fisheries Department, A. O. *The State of World Fisheries and Aquaculture. Towards Blue Transformation, 2022* (Food & Agriculture Org., 2022).
2. Bennett, S. *Dive operators hit by a shortage of staff* <https://xray-mag.com/content/dive-operators-hit-shortage-staff>.
3. Colleges, C. E. & Universities. *Shortage of Skilled Commercial Divers* <https://www.career.org/releases-and-statements/shortage-of-skilled-commercial-divers>.
4. Zhao, Y.-P., Niu, L.-J., Du, H. & Bi, C.-W. An adaptive method of damage detection for fishing nets based on image processing technology. *Aquacultural engineering* **90**, 102071 (2020).
5. Paris, S., Kornprobst, P., Tumblin, J., Durand, F. *et al.* Bilateral filtering: Theory and applications. *Foundations and Trends® in Computer Graphics and Vision* **4**, 1–73 (2009).
6. Paspalakis, S., Moirogiorgou, K., Papandroulakis, N., Giakos, G. & Zervakis, M. Automated fish cage net inspection using image processing techniques. *IET Image Processing* **14**, 2028–2034 (2020).
7. Su, B. *et al.* An integrated approach for monitoring structural deformation of aquaculture net cages. *Ocean Engineering* **219**, 108424 (2021).
8. Chalkiadakis, V. *et al.* *Designing a small-sized autonomous underwater vehicle architecture for regular periodic fish-cage net inspection* in *2017 IEEE International Conference on Imaging Systems and Techniques (IST)* (2017), 1–6.
9. Tao, Q. *et al.* *Omnidirectional surface vehicle for fish cage inspection* in *OCEANS 2018 MTS/IEEE Charleston* (2018), 1–6.
10. Lin, T. X., Tao, Q. & Zhang, F. *Planning for Fish Net Inspection with an Autonomous OSV* in *2020 International Conference on System Science and Engineering (ICSSE)* (2020), 1–5.
11. Amundsen, H. B., Caharija, W. & Pettersen, K. Y. Autonomous ROV inspections of aquaculture net pens using DVL. *IEEE Journal of Oceanic Engineering* **47**, 1–19 (2021).
12. Livanos, G. *et al.* *Intelligent navigation and control of a prototype autonomous underwater vehicle for automated inspection of aquaculture net pen cages* in *2018 IEEE International Conference on Imaging Systems and Techniques (IST)* (2018), 1–6.
13. Bjerkeng, M. *et al.* ROV navigation in a fish cage with laser-camera triangulation. *Journal of Marine Science and Engineering* **9**, 79 (2021).

14. Schellewald, C., Stahl, A. & Kelasidi, E. Vision-based pose estimation for autonomous operations in aquacultural fish farms. *IFAC-PapersOnLine* **54**, 438–443 (2021).
15. Duda, A., Schwendner, J., Stahl, A. & Rundtop, P. *Visual pose estimation for autonomous inspection of fish pens* in *OCEANS 2015-Genova* (2015), 1–6.
16. Akram, W., Casavola, A., Kapetanović, N. & Miškovic, N. A visual servoing scheme for autonomous aquaculture net pens inspection using ROV. *Sensors* **22**, 3525 (2022).
17. Kramer, O. & Kramer, O. K-nearest neighbors. *Dimensionality reduction with unsupervised nearest neighbors*, 13–23 (2013).
18. Chang, C.-C. *et al.* Applying artificial intelligence (AI) techniques to implement a practical smart cage aquaculture management system. *Journal of Medical and Biological Engineering* **41**, 652–658 (2021).
19. Goričanec, J. *et al.* *Heterogeneous autonomous robotic system in viticulture and mariculture-project overview* in *2021 16th International Conference on Telecommunications (ConTEL)* (2021), 181–188.
20. Borković, G. *et al.* *Underwater ROV software for fish cage inspection* in *2021 44th International Convention on Information, Communication and Electronic Technology (MIPRO)* (2021), 1747–1752.
21. Dobeck, G. J., Hyland, J. C. *et al.* Automated detection and classification of sea mines in sonar imagery in *Detection and Remediation Technologies for Mines and Minelike Targets II* **3079** (1997), 90–110.
22. Maussang, F., Chanussot, J., Hétet, A. & Amate, M. Mean–standard deviation representation of sonar images for echo detection: Application to SAS images. *IEEE Journal of Oceanic Engineering* **32**, 956–970 (2007).
23. Lu, Y. & Sang, E. Underwater target's size/shape dynamic analysis for fast target recognition using sonar images in *Proceedings of 1998 International Symposium on Underwater Technology* (1998), 172–175.
24. Guo, J., Cheng, S.-W. & Liu, T.-C. AUV obstacle avoidance and navigation using image sequences of a sector scanning sonar in *Proceedings of 1998 International Symposium on Underwater Technology* (1998), 223–227.
25. Petillot, Y., Ruiz, I. T. & Lane, D. M. Underwater vehicle obstacle avoidance and path planning using a multi-beam forward looking sonar. *IEEE journal of oceanic engineering* **26**, 240–251 (2001).
26. Galceran, E., Djapic, V., Carreras, M. & Williams, D. P. A real-time underwater object detection algorithm for multi-beam forward looking sonar. *IFAC Proceedings Volumes* **45**, 306–311 (2012).
27. Ferreira, F., Djapic, V., Micheli, M. & Caccia, M. Improving automatic target recognition with forward looking sonar mosaics. *IFAC Proceedings Volumes* **47**, 3382–3387 (2014).
28. Fossen, T. I. *Handbook of marine craft hydrodynamics and motion control* (John Wiley & Sons, 2011).
29. Team, D. R. *Farol* 2023. <https://github.com/charlespwd/project-title>.

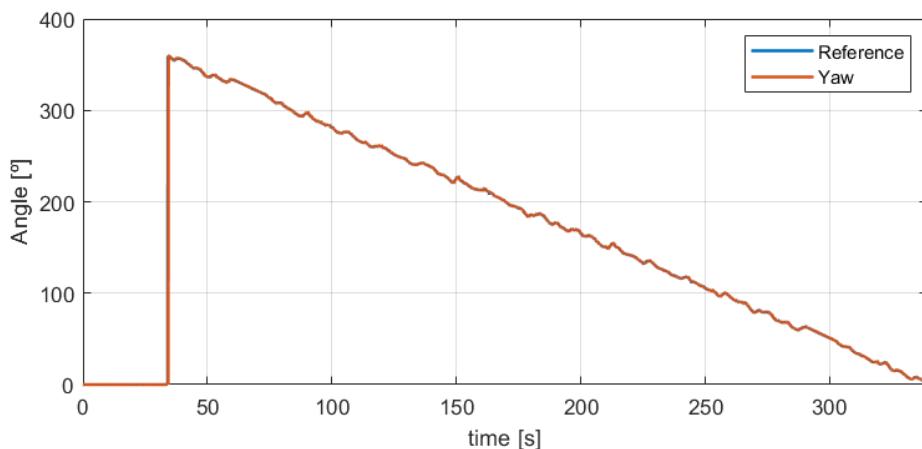
30. Choi, W.-S. *et al.* Physics-based modelling and simulation of multibeam echosounder perception for autonomous underwater manipulation. *Frontiers in Robotics and AI* **8**, 706646 (2021).
31. Choi, W.-S. *et al.* *nps\_uw\_multibeam\_sonar* 2021. [https://github.com/Field-Robotics-Lab/nps%5C\\_uw%5C\\_multibeam\\_sonar](https://github.com/Field-Robotics-Lab/nps%5C_uw%5C_multibeam_sonar).
32. *Multibeam Echosounder ARIS Explorer 3000*. Sound Metrics (). <http://www.soundmetrics.com/products/arис-sonars/ARIS-Explorer-3000/016621-A-ARIS-Explorer-3000-Specifications>.
33. Johannsson, H., Kaess, M., Englot, B., Hover, F. & Leonard, J. *Imaging sonar-aided navigation for autonomous underwater harbor surveillance* in *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems* (2010), 4396–4403.
34. Gavin, H. P. *The Levenberg-Marquardt method for nonlinear least squares curve-fitting problems* c © in (2013).
35. Fertik, H. & Ross, C. Direct digital control algorithm with anti-windup feature. *ISA Transactions* (1967).

## Appendix A

### Additional Figures for test T2



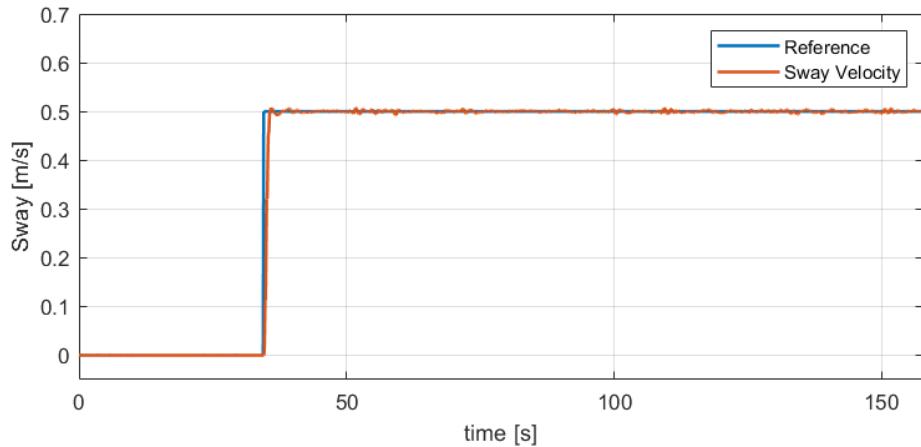
**Figure A.1:** Sway behavior throughout test T2 at 0.2 m/s, at 5 m depth, and at 2.5 m from the net.



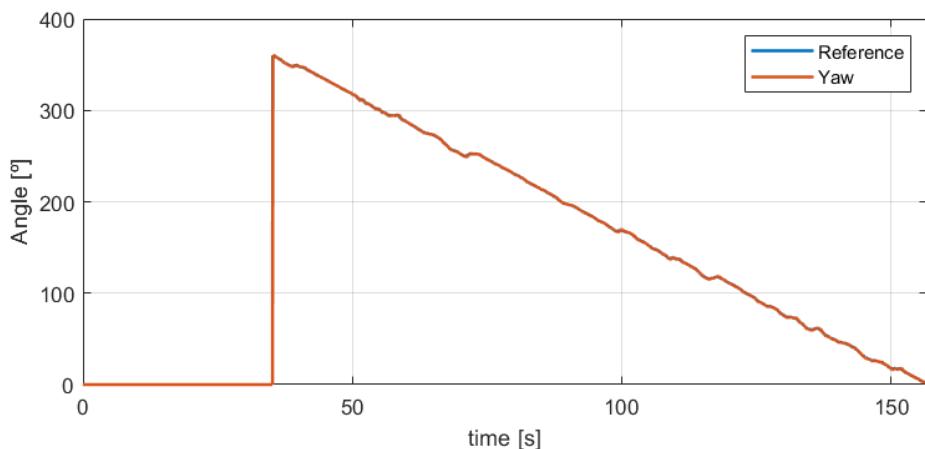
**Figure A.2:** Yaw behavior throughout test T2 at 0.2 m/s, at 5 m depth, and at 2.5 m from the net.

## Appendix B

### Additional Figures for test T3



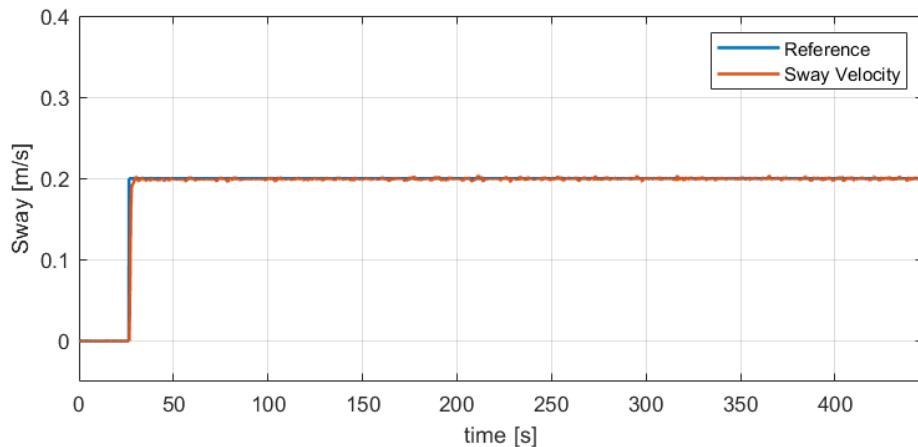
**Figure B.1:** Sway behavior throughout test T3 at 0.5 m/s, at 5 m depth, and at 2.5 m from the net.



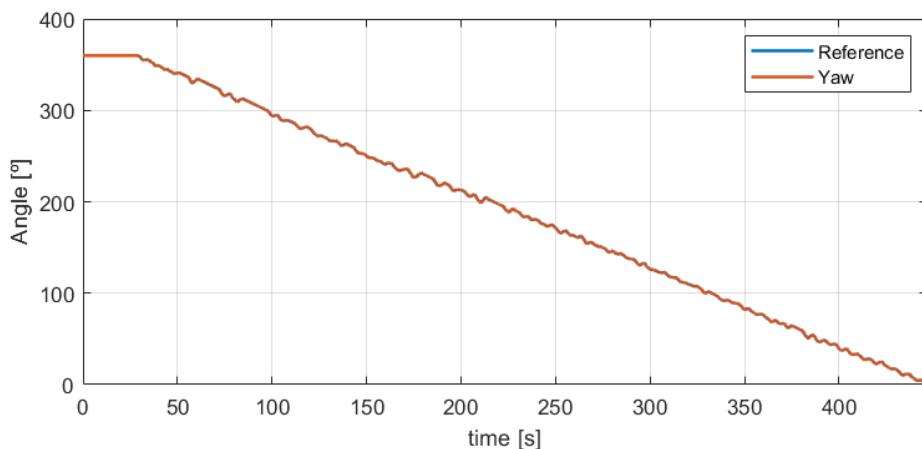
**Figure B.2:** Yaw behavior throughout test T3 at 0.5 m/s, at 5 m depth, and at 2.5 m from the net.

## Appendix C

### Additional Figures for test T4



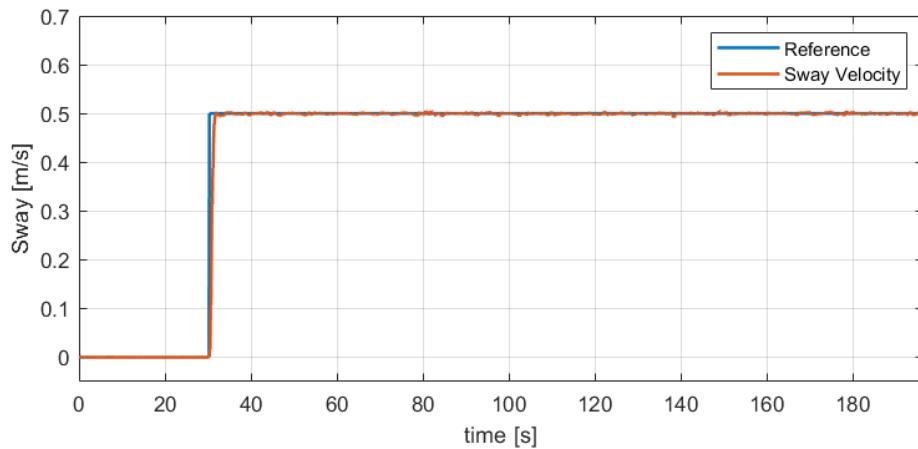
**Figure C.1:** Sway behavior throughout test T4 at 0.2 m/s, at 5 m depth, and at 5 m from the net.



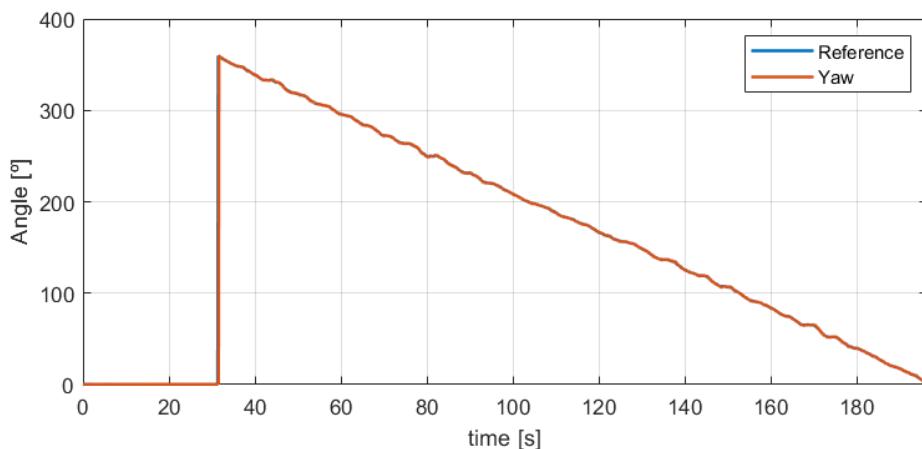
**Figure C.2:** Yaw behavior throughout test T4 at 0.2 m/s, at 5 m depth, and at 5 m from the net.

## Appendix D

### Additional Figures for test T5



**Figure D.1:** Sway behavior throughout test T5 at 0.5 m/s, at 5 m depth, and at 5 m from the net.



**Figure D.2:** Yaw behavior throughout test T5 at 0.5 m/s, at 5 m depth, and at 5 m from the net.

**Sonar Based Autonomous Aquaculture Net Cage Inspection**

Daniel Garcia Domingos Rosa