Tampere University

JOSE VILLA ESCUSOL

A Guidance, Navigation, and Control Architecture for a Co-operative Autonomous Offshore System

Tampere University Dissertations 471

Tampere University Dissertations 471

JOSE VILLA ESCUSOL

A Guidance, Navigation, and Control Architecture for a Co-operative Autonomous Offshore System

ACADEMIC DISSERTATION
To be presented, with the permission of
the Faculty of Engineering and Natural Sciences
of Tampere University,
for public discussion in the auditorium K1702
of Konetalo, Korkeakoulunkatu 7, Tampere,
on 1st of October 2021, at 12:15 PM.

ACADEMIC DISSERTATION
Tampere University, Faculty of Engineering and Natural Sciences
Finland

Responsible supervisor and Custos	Prof. Kari T. Koskinen Tampere University Finland	
Supervisor	Dr. Jussi Aaltonen Tampere University Finland	
Pre-examiners	Associate Prof. Kari Tammi Aalto University Finland	Prof. Carlos Efrén Mora Luis University of La Laguna Spain
Opponents	Associate Prof. Kari Tammi Aalto University Finland	Prof. Luc Jaulin University Bretagne Occidentale France

The originality of this thesis has been checked using the Turnitin OriginalityCheck service.

Cover design: Roihu Inc.

ISBN 978-952-03-2096-6 (print)
ISBN 978-952-03-2097-3 (pdf)
ISSN 2489-9860 (print)
ISSN 2490-0028 (pdf)
http://urn.fi/URN:ISBN:978-952-03-2097-3

PunaMusta Oy – Yliopistopaino
Joensuu 2021

PREFACE

The research presented in this thesis was carried out at the Hervanta campus at Tampere University (known as the Tampere University of Technology when the thesis first began) from 2017—2021. The research was principally funded by the Technology Industries of Finland Centennial Foundation and Jane and Aatos Erkko Foundation in the Autonomous and Collaborative Offshore Robotics (aColor) project under the Future Makers 2017 program. Moreover, this work was partly funded by the project MOCAV-G500 from EUMarineRobots at the European Union's Horizon 2020.

I want to express my greatest gratitude to my supervisors Professor Kari T. Koskinen and Dr. Jussi Aaltonen. I am truly thankful for the opportunity that they gave me to do my doctoral studies when I was an exchange student. Thank you for the feedback and recommendations throughout this thesis and all the inspiring discussions. I would also like to thank my colleagues from the Mechatronics Research Group (MRG) for their unconditional support throughout this research, particularly Tuomas Salomaa and Kalle Hakonen, for their help and advice in the designing and first implementation phases of the offshore vehicles. Furthermore, I would like to thank the company Alamarin Jet Oy, specially Sauli Virta, for their support during the development and implementation of the research vessel.

I want to thank the pre-examiners, Professor Carlos Efrén Mora Luis and Associate Professor Kari Tammi. Their comments helped to improve this thesis even further. Additionally, I would like to thank Professor Luc Jaulin and again Associate Professor Kari Tammi for accepting the invitation to act as the opponents in the public defense of my dissertation.

I am grateful to have had the support of all my friends in Tampere, who made living abroad feeling like home. They include Alberto Brihuega, Carlos Baquero, Carmen Cobos, Irene Martin, Laura Martin, Ruben Morales, Carlos Castillo, Anastasia Yastrebova, Sergio Moreschini, Pavel Marek, Maja Marek, Elena Peralta, Amir

Ahmadi, Jose Gonzalez, Laia Pi, Eneko Piedra, Aida Puente, and so many other friends whose names I have not mentioned here. Moreover, I want to be thankful to my great friends from *la peña Revolución & Trinkal* at my village, my studies at the University of Zaragoza, and my Erasmus studies at Tampere for their support from far away.

Lastly, yet importantly, I would like to thank my parents Ana and Jose Maria, as well as my brother Javier, my sister-in-law Anun, and my nieces Úrsula and Inés, for their unconditional long-distance love and encouragement. Their help and support have been crucial throughout my doctoral studies and the final writing of this thesis. I want to particularly thank my brother Javier for encouraging me to study industrial engineering and go for Erasmus, becoming both essential pieces of advice for my doctoral studies.

Tampere, August, 2021
Jose Villa Escusol

ABSTRACT

In recent years, researchers have widely used autonomous systems in marine environment exploration and exploitation. The main reasons for this are the amount of unknown and unexplored areas (in oceans, seas, and lakes) and the extensive range of autonomous vehicle applications. Autonomous offshore systems include unmanned surface vehicles (USVs) and autonomous underwater vehicles (AUVs) as primary offshore vehicles; their guidance, navigation, and control (GNC) architectures play a significant role in algorithm development. The ultimate goal of this research is to solve the design, modeling, and implementation challenges of a path-following algorithm with obstacle avoidance as the GNC architecture for a co-operative autonomous offshore system formed by a USV and AUV.

First, this thesis concentrates on developing a mathematical model based on nonlinear equations of motion, using system identification (SI) and parameter estimation techniques and validating the USV and AUV models with field test data. Second, this thesis also provides a comprehensive analysis of various guidance and control methods focusing on the path-following and obstacle avoidance algorithms. The GNC architecture uses a modular and multi-layer approach allowing for the fast check of the GNC algorithms for both USV and AUV platforms. This architecture includes all obstacle detection, path-following, and control algorithms. Then, the results show the implementation challenges in simulation and field test control scenarios. These results present the capabilities and adequate performance of the developed GNC architecture for an individual vehicle operation in the autonomous offshore system. Finally, a GNC architecture for the complete co-operative autonomous offshore system is designed and implemented based on the development of the USV and AUV. The co-operative system implementation includes decentralized control techniques, allowing for the fusion of information obtained from the individual vehicles. Additionally, the decentralized control allows for exchanging the necessary information with other components of the co-operative system.

Index terms:

Path-following, obstacle avoidance, model validation, GNC architecture, co-operative, autonomous.

CONTENTS

List of Figures

List of Tables

ABBREVIATIONS

1D	one dimension
2D	two dimensions
3D	three dimensions
AHRS	attitude and heading reference system
AOV	autonomous offshore vehicle
AUV	autonomous underwater vehicle
BODY	body-fixed reference frame
CAN	controller area network
COLREGs	convention on the international regulations for preventing collisions at sea
DOFs	degrees of freedom
DVL	doppler velocity log
ENU	east-north-up
ESC	electronic speed controller
GNC	guidance, navigation, and control
GPS	global positioning system
LOS	line of sight
MAE	mean absolute error
NED	north-east-down
PID	proportional-integral-derivative
PWM	pulse-width modulation

RMSE	root-mean-square error
ROS	robot operating system
SBB	safety boundary box
SD	standard deviation
SI	system identification
SLAM	simultaneous localization and mapping
STDB	starboard
UAV	unmanned aerial vehicle
USBL	ultra-short baseline
USV	unmanned surface vehicle

NOMENCLATURE

α_{box}	safety boundary box angle
α_{nozzle}	waterjet thrust force angle
α_{path}	predefined path angle
β	sideslip (drift) angle at the LOS guidance algorithm
$\boldsymbol{\eta}$	pose vector including position and orientation
$\boldsymbol{\nu}$	velocity vector including linear and angular velocities
$\boldsymbol{\tau}$	vector of generalized forces
$\boldsymbol{\tau}_{\text{wave}}$	environmental forces vector produced by the waves
$\boldsymbol{\tau}_{\text{wind}}$	environmental forces vector produced by the wind
$\boldsymbol{C}(\boldsymbol{\nu})$	Coriolis-centripetal matrix
$\boldsymbol{C}_{\text{A}}(\boldsymbol{\nu})$	Coriolis-centripetal added terms
$\boldsymbol{C}_{\text{RB}}(\boldsymbol{\nu})$	Coriolis-centripetal rigid-body terms
$\boldsymbol{D}(\boldsymbol{\nu})$	hydrodynamic damping matrix
$\boldsymbol{D}_{\text{lin}}$	linear damping matrix
$\boldsymbol{D}_{\text{nlin}}(\boldsymbol{\nu}_r)$	nonlinear damping matrix
$\boldsymbol{f}$	control forces and moments vector
$\boldsymbol{l}_n$	location vector for n thruster distance
$\boldsymbol{M}_{\text{A}}$	hydrodynamic added mass matrix

$\boldsymbol{M}_{\mathrm{RB}}$ rigid-body system inertia matrix

$\boldsymbol{M}$ mass matrix

$\boldsymbol{R}_{\mathrm{x}}$ rotation matrix about the x-axis

$\boldsymbol{R}_{\mathrm{z}}$ rotation matrix about the z-axis

$\boldsymbol{r}_{\mathrm{g}}^{b}$ distance vector from origin o_{b} to the centre of gravity

$\boldsymbol{T}$ thrust configuration matrix

χ_{p} path-tangential angle at the LOS guidance algorithm

χ_{r} velocity-path relative angle at the LOS guidance algorithm

δ rudder angle

$\Delta(t)$ look-ahead distance at the LOS guidance algorithm

$\varkappa$ scaling factor as the vehicle mass is not symmetrically distributed

$\mathscr{B}$ buoyancy force

$\mathscr{E}_{\mathrm{nozzle}}$ total thrust force efficiency

$\mathscr{I}_{\mathrm{scan}}$ every bin intensity from the mechanical imaging sonar

$\mathscr{W}$ submerged weight of the body

∇ fluid volume displaced by the AOV

ω shaft rotational speed of the waterjet engine

ω_{b} controller bandwidth

ω_{rpm} waterjet engine rpm

ϕ roll angle

ψ yaw angle

ψ_{d} course angle at the LOS guidance algorithm

ρ water density

τ_r torque produced by the AOV

τ_u thrust force produced by the AOV

$\boldsymbol{g}$ vector of gravitational/buoyancy forces and moments

θ pitch angle

θ_{scan} beam angle from the mechanical imaging sonar

ξ design parameter for controlling the notch magnitude

c_g relative center of mass point

$C_{pos}(s)$ position controller

$C_{vel}(s)$ velocity controller

D Down position

d_{dist_X} predefined constant parameters for safety distance for the x axis

d_{dist_Y} predefined constant parameters for safety distance for the y axis

E East position

$e(t)$ cross-track error at the LOS guidance algorithm

F thrust force per waterjet

F_{PORT} thrust force from the port waterjet

F_{STDB} thrust force from the starboard waterjet

F_{total} total thrust force from the waterjet propulsion system

g gravity

$h_n(s)$ filter structure

$h_{lp}(s)$ first-order low-pass filter

I_{cor} moment of inertia tuning factor at the USV

I_z moment of inertia about z axis

K_p process gain in a transfer function

K_D derivative gain from the PID controller

K_I integral gain from the PID controller

K_P proportional gain from the PID controller

K_{zn} tuning gain from the Ziegler-Nichols method

L_{box} safety boundary box length dimension

L_{obs} detected obstacle length dimension

l_{pivot} pivot point location

l_{pt} estimated powertrain mass location

L_{USV} total length of the USV

m vehicle mass

m_{hull} hull weight without the powertrain mass

m_{pt} estimated powertrain mass including the waterjets units, engines, fuel, etc.

N North position

N_r yaw linear damping coefficient

$N_{\dot{r}}$ yaw hydrodynamic added mass coefficient

$N_{|r|r}$ yaw nonlinear damping coefficient

o origin

p roll rate or roll angular velocity

P_{nozzle} nozzle position from the waterjet propulsion unit

q pitch rate or pitch angular velocity

R radius

r yaw rate or yaw angular velocity

T_d system input/output delay in a transfer function

T_z system zero in a transfer function

T_f time constant for the low-pass filter

T_p system pole in a transfer function

T_{zn} tuning period from the Ziegler-Nichols method

u surge linear velocity

v sway linear velocity

w heave linear velocity

W_{box} safety boundary box width dimension

W_{obs} detected obstacle width dimension

x x-axis position in the Cartesian coordinate system

$X_{\dot{u}}$ surge hydrodynamic added mass coefficient

$X_{|u|u}$ surge nonlinear damping coefficient

X_u surge linear damping coefficient

y y-axis position in the Cartesian coordinate system

$Y_{\dot{r}}$ sway hydrodynamic added mass force due to an angular acceleration $\dot{r}$

$Y_{\dot{v}}$ sway hydrodynamic added mass coefficient

$Y_{|v|v}$ sway nonlinear damping coefficient

Y_v sway linear damping coefficient

z z-axis position in the Cartesian coordinate system

$Z_{\dot{w}}$ heave hydrodynamic added mass coefficient

$Z_{|w|w}$ heave nonlinear damping coefficient

Z_w heave linear damping coefficient

ORIGINAL PUBLICATIONS

Publication I J. Villa, J. Aaltonen and K. T. Koskinen. Model-based path planning and obstacle avoidance architecture for a twin jet Unmanned Surface Vessel. *2019 Third IEEE International Conference on Robotic Computing (IRC)*. IEEE. 2019, 427–428. DOI: 10.1109/IRC.2019.00083.

Publication II J. Villa, G. Vallicrosa, J. Aaltonen, P. Ridao and K. T. Koskinen. Model-based Guidance, Navigation and Control architecture for an Autonomous Underwater Vehicle. *Global Oceans 2020: Singapore – U.S. Gulf Coast*. 2020, 1–6. DOI: 10.1109/IEEECONF38699.2020.9389247.

Publication III J. Villa, J. Aaltonen and K. T. Koskinen. Path-Following with LiDAR-based Obstacle Avoidance of an Unmanned Surface Vehicle in Harbor Conditions. *IEEE/ASME Transactions on Mechatronics* (2020). DOI: 10.1109/TMECH.2020.2997970.

Publication IV J. Villa, J. Aaltonen, S. Virta and K. T. Koskinen. A Co-Operative Autonomous Offshore System for Target Detection Using Multi-Sensor Technology. *Remote Sensing* 12.24 (2020), 4106. DOI: 10.3390/rs12244106.

Unpublished manuscript

Publication V J. Villa, G. Vallicrosa, J. Aaltonen, P. Ridao and K. T. Koskinen. Model-Validation and Implementation of a Path-following Algorithm in an Autonomous Underwater Vehicle. 2021.

Author's contribution

This thesis includes the scientific outputs from four published publications and one unpublished journal article. The summary (compendium) was prepared by the author of this thesis (later: the Author) Jose Villa Escusol, and it was reviewed by the responsible supervisor Professor Kari T. Koskinen and supervisor Dr. Jussi Aaltonen.

In all publications, the Author conceptualized and designed the methodology, developed the software and validation of the mathematical models, performed the necessary experiments, analyzed the data, and wrote the original manuscripts based on the test results. Furthermore, the Author is the main contributor in all five publications included in this thesis. The named co-authors supervised the study and partook in writing, reviewing, and editing the publications.

The Author and co-author contributions are presented as follows:

Publication I — The Author's contributions to this conference article involve developing and implementing the modular GNC architecture in a USV for a straight line path-following algorithm. The manuscript was written and revised by the Author. The co-authors provided their feedback and reviewed the final version of the manuscript.

Publication II — The Author was the main contributor to this conference article, developing a modular GNC architecture in an AUV for a wall-detection algorithm with waypoint following. This work was done in collaboration with the Universitat de Girona (Spain), where Dr. Guillem Vallicrosa assisted in performing the AUV experiments for this manuscript. The Author analyzed the results, and the co-authors provided their feedback for the final version of the manuscript.

Publication III — The Author's contributions to this journal article incorporate the conceptualization, development, and implementation for a path-following with obstacle avoidance in a USV based on the safety boundary box approach. The Author developed the mathematical model of the vehicle and wrote and revised the manuscript. The co-authors provided their feedback for the final version of the article.

Publication IV The Author's contributions to this journal article include the development and implementation of a target detection algorithm using multi-sensor technology in a co-operative offshore system. The Author analyzed the data and also wrote and revised the manuscript based on the test results. Mr. Sauli Virta assisted in performing the necessary USV experiments. The co-authors provided their feedback for the final version of this journal article.

Publication V The Author's contributions to this journal article include the development and implementation of a path-following algorithm for an AUV in an open environment. Also, the Author developed the model validation of the vehicle based on field test data. This work was done in collaboration with the Universitat de Girona (Spain), where Dr. Guillem Vallicrosa performed the necessary AUV experiments for this study. The manuscript was written and revised by the Author, and the co-authors provided their feedback for the final version of the article.

1 INTRODUCTION

Different aspects in society, such as climate change, environmental abnormalities, and personnel requirements, have led to a strong demand for the research and development of innovative autonomous systems that can be used in commercial, scientific, and military communities. In recent years, researchers have widely used autonomous systems in marine environment exploration and exploitation because of the amount of unknown and unexplored marine areas and extensive range of autonomous vehicle applications. Unmanned surface vehicles (USVs) and autonomous underwater vehicles (AUVs) as primary autonomous offshore vehicles (AOVs) and their guidance, navigation, and control (GNC) architectures play a significant role in algorithm development. In the GNC architecture, situational awareness and mission control are crucial for the operation of the AOV.

AOVs contain multiple sensors and actuators for situational awareness, positioning, or simple vehicle operation. The connectivity between these objects involves many different areas depending on which actuators or sensors are employed. With the possibility for more effective, affordable, and compact navigation sensors, numerous innovative research topics have appeared for autonomous system applications. The AOVs development covers a wide variety of potential applications in a profitable way, such as marine environment exploitation and exploration, scientific research, or military applications. Finally, co-operative systems allow direct interaction in a robotic net formed by several AOVs in the same workspace.

1.1 Objectives and Scope

The scope of this thesis covers the GNC architecture for a co-operative autonomous offshore system. The ultimate goal is to solve the design, modeling, and implementation challenges of a path-following algorithm that also contains an obstacle avoidance as GNC architecture for the co-operative system. This co-operative system employs

a USV and AUV. This implementation includes shared intelligence, situational awareness capabilities, and proper guidance system and control algorithms for a target detection scenario. Furthermore, the proposed GNC architecture combines all these capabilities in the same framework for each offshore vehicle, allowing their operation as a decentralized system. From the co-operative system application point of view, this thesis seeks to attain the following objectives:

- Develop a GNC architecture for shared intelligence in an autonomous offshore system, including situational awareness capabilities and mission control.
- Obtain mathematical models for all AOVs involved in this thesis, allowing an accurate simulation environment to design additional GNC algorithms of the autonomous offshore system.
- Implement the proposed GNC architecture in a co-operative system formed by a USV and AUV.

1.2 Hypothesis and Research Questions

The development of an analogous GNC architecture for AOVs will enable simple and easy connectivity and shared intelligence between multiple offshore vehicles, such as USVs and AUVs. This GNC architecture requires situational awareness capabilities and mission control algorithms to implement the co-operative system tasks. These algorithms need a comprehensive design before their final implementation. Hence, specific implementation methods are a crucial part of the final autonomous offshore system implementation. Furthermore, this offshore system can improve its performance with a co-operative system by using a multi-vehicle approach, including above- and below-water characterization. Thus, the GNC architecture demands a common framework between the offshore vehicles to obtain outstanding results.

To achieve the above-mentioned objectives, this thesis attempts to answer the following research questions for the proposed GNC architecture for a co-operative autonomous offshore system:

RQ1. What kind of implementation methods are needed for situational awareness and mission control in a system of multiple unmanned offshore vehicles?

RQ2. What kind of architecture should be used for multi-sensor networks and integration in offshore vehicle applications?

RQ3. What kind of co-operative framework is needed for shared intelligence in multiple autonomous robotic systems?

1.3 Contributions and Structure

The main contributions of this thesis are the following:

1. Development of a mathematical model based on nonlinear equations of motion for each offshore vehicle. To obtain accurate results from the simulation environment, this mathematical model needs an approximation of its hydrodynamic coefficients based on parameter estimation techniques. This model allows for proper GNC systems design with sensing, state estimation, and situational awareness capabilities. **Publications III and IV** explore the USV mathematical model based on two different parameter estimation methods. Similarly, **Publication V** presents the mathematical model for an AUV.
2. GNC architecture design, modeling, and implementation in the offshore vehicle applications, including collision avoidance capabilities in the guidance and control system. The GNC architecture uses a modular and multi-layer approach that provides a computationally cheap and easy implementation for the required autonomous capabilities. The modular approach allows these capability implementations individually, with the possibility to design and test each of the modules separately in both simulation and field test environments. **Publications I—V** include the development and implementation of this GNC architecture in the USV and AUVs employed in this thesis for different control scenarios.
3. GNC architecture implementation for a co-operative system formed by a USV and AUV. The co-operative system includes the necessarily shared intelligence between the vehicles, providing a solution for above- and below-water characterization. **Publication IV** presents the co-operative scenario where the AUV detects and locates an underwater target, and the USV carries out further inspection.

The thesis is organized as follows: Chapter 2 gives an overview of the state-of-the-art AOVs, which focuses on AOV sensors, AOV design and modeling, and GNC methods for single- and multi-vehicle systems. Then, Chapter 3 describes the

AOVs' mathematical model development and model validation based on parameter estimation procedures using field test data. After this, Chapter 4 presents the GNC architecture used in this thesis work for single-vehicle operations. This chapter describes the situational awareness methods, guidance systems, and control algorithms for each AOV application. This chapter also shows the experimental validation results to verify the proposed GNC architecture. Following this, Chapter 5 describes the GNC architecture for the co-operative system formed by a USV and AUV. This chapter describes the decentralized system employed for this system, along with the experimental results of the co-operative scenario. Finally, Chapter 6 presents the final discussion of this thesis, and Chapter 7 presents the conclusion and future research directions.

2 STATE OF THE ART

Offshore inspections and many other offshore operations remain very labor intensive and expensive activities. These activities typically require at least manned support surface vessels. However, the recent use of AOVs has increased the interest of research scientists, various maritime industries, and the military. Moreover, these vehicles are becoming a trend in the exploitation and exploration of the marine environment. This interest in offshore interventions includes numerous activities, such as search and rescue missions, seabed explorations, target detection, or offshore surveillance. The AOVs involve USVs and AUVs as the primary offshore vehicles. The main reasons for this are the amount of unknown and unexplored areas (in oceans, seas, and lakes) and the extensive range of autonomous vehicle applications. Thus, underwater research is currently a relevant topic in scientific research because of the ease of data gathering in remote and hazardous scenarios. Even though historically the focus has been on AUVs, research on USVs has become more relevant in recent years. Furthermore, co-operative systems can improve their performance by using multi-vehicle platforms, including above- and below-water characterization.

Remote operated vehicles and AUVs are the general classifications for underwater vehicles. There are numerous research topics for underwater vehicles, such as path planning [27], obstacle avoidance [74], or underwater manipulation [72]. Ribas *et al.* [63] described different methods for map-based localization and a novel approach for underwater simultaneous localization and mapping (SLAM), which has been a meaningful underwater research topic.

AUVs usually require manned surface vehicles for their operation. Thus, USV development as support platforms for AUVs will increase their autonomy and potential use cases. Curcio *et al.* [15] presented the first known implementation of USVs used to support AUVs. Then, Fallon *et al.* [19] performed AUV navigation and localization with a USV by including the primary heuristics for keeping observability and establishing the AUV survey by implementing various motion operations.

Additionally, Vasilijević *et al.* [80] presented an application for environmental monitoring and ocean sampling by utilizing a co-operative robotic system constituted by a USV and AUV. Other scenarios include the launch and recovery of an AUV from a station-keeping USV [69]. The co-operative system can also incorporate additional platforms to increase the offshore system capabilities with above-water inspection. Ross *et al.* [65] presented a heterogeneous system formed by a USV as the main platform, an AUV, and an unmanned aerial vehicle (UAV). This system aimed to achieve multi-domain awareness in a responsive ship or any floating structure as the floating target.

2.1 Sensors and Sensor Integration in Autonomous Offshore Vehicles

AOVs currently perform several offshore operations in maritime environments. These operations need accurate navigation and localization to ensure the accuracy of the data acquisition and processing. The navigational accuracy is the precision to reach a predetermined waypoint, while the localization accuracy relates to the error when localizing the AOV within a map.

The most common practices in autonomous systems above the water surface are the global positioning system (GPS) and spread spectrum or radio communications. However, these signals can only propagate in short distances while performing in an underwater scenario and are not suitable for autonomous underwater systems. Thus, acoustic-based sensors and communications are selected for underwater applications because they have better performance. Nevertheless, acoustic communications still suffer from many shortcomings, such as the low data rate, small bandwidth, high latency, or unreliability. Hence, the communication system must manage its transmissions without losing data.

2.1.1 Unmanned Surface Vehicles

Situational awareness is crucial in the design of high levels of autonomy in USVs. Wolf *et al.* [86] developed situational awareness during USV patrol missions based on change detection and object-level tracking method for detecting targets, establishing

their position, and identifying fluctuations in the nearby environment. For the USV state estimations, a vector GPS compass is commonly used as the primary sensor to obtain accurate heading and position for USV navigation. In this case, the GPS compass uses a satellite-based augmentation system for differential GPS position, providing a low-cost and highly accurate vehicle pose. Apart from the GPS compass, active ranging sensor methods, such as LiDAR and radar, can be utilized for state estimation. These methods are notably effective when there is a loss or jamming of GPS signals. These GPS signals may become weak and unreliable when the USV navigates near bridges or other covered environments. Additionally, a suitable choice for these scenarios is SLAM, which is becoming increasingly important in research applications due to the possibility to detected contours and employ them as landmark features [31].

USVs require the capabilities of obstacle detection and recognition, tracking targets, and mapping environments to accomplish real-world applications. There are two categories when grouping the environmental perception approaches for USVs based on the characteristics of the intended applications: passive perception methods, which adopt the infrared or visual sensors employed in numerous environment perception applications, and active ranging sensor methods, with LiDAR, radar, and sonar as the main sensors. LiDARs are the sturdiest sensors for acquiring depth data in obstacle detection techniques. Halterman and Bruch [29] studied the performance of three-dimensional (3D) scanning LiDAR installed in a USV. Another active perception sensor in USV applications is marine radar, which is the most used obstacle detection method for far-field applications [40]. The primary use of pulse radar sensors is still in the military area, but it is becoming more important in research applications. Zhuang *et al.* [89] developed an embedded collision avoidance system in a USV based on a marine radar sensor. Additionally, Han *et al.* [32] addressed the algorithm development for multiple target detection and tracking for a USV in the sensor fusion framework by integrating LiDAR and marine radar.

2.1.2 Autonomous Underwater Vehicles

Most underwater applications still use old technologies, such as long baseline and ultra-short baseline (USBL), requiring support infrastructure. However, dynamic multi-agent system approaches are more often being used in these applications because they

allow for flexibility and rapid deployment using minimal infrastructure. Regarding these dynamic approaches, underwater systems increasingly include the use of SLAM techniques based on above-ground robotics applications [63]. Thus, more accurate AUV navigation is becoming possible in a more cost-efficient way.

Accurate localization and navigation are essential in data acquisition and processing for autonomous applications. As mentioned above, most autonomous systems count on GPS systems and spread-spectrum communications above the water's surface. However, those signals only propagate over short distances because of the rapid attenuation of higher-frequency signals in the underwater environment. Thus, acoustic-based systems are used in AUV applications because their performance is better in the underwater scenario [58]. The underwater navigation and localization techniques are categorized as the following main categories [41]:

- Inertial/dead reckoning: Inertial navigation uses gyroscopes and accelerometers to disseminate the current AUV state. Nonetheless, each of these methods has unbounded position error growth.
- Acoustic transponders and modems: These navigation techniques measure the time of flight between signals from acoustic beacons or modems to the other platform.
- Geophysical: These techniques utilize external environmental information as references for the AUV's navigation. The underwater sensors need to detect, identify, and classify some surrounding environment features.

The underwater localization and navigation methods need specific navigation and survey sensors placed in the AUV platform. The most basic sensors for AUV navigation are the compass, which provides a globally bounded heading reference, and the barometer or pressure sensor, which measures the underwater depth of the AUV. Regarding acoustic navigation techniques, USBL navigation enables the underwater localization of the AUV relative to a support platform, offering an efficient and stable acoustic communication network [53]. The phase differencing across transceivers determines a relative bearing, while the time of flight determines the range. These transceivers, also known as model and transponder units, form the USBL navigation system, with its range being a major limitation. The modem is usually installed on the AUV's nose, while there is an acoustic transponder placed on a support platform that acts as the target because its position known and fixed. Batista *et al.* [3] proposed the

use of a USBL positioning system for sensor-based integrated guidance and control. This approach is used in the aColor AUV in the current thesis, employing the USBL navigation system for localization and possible communication with the support platform. Additionally, a mechanical imaging sonar allows underwater situational awareness capabilities.

Regarding AUV navigation algorithms, Miller *et al.* [48] considered the navigation problem for an AUV using an error state estimation based on a Kalman filter. The sensors used for this state estimation were a doppler velocity log (DVL), a pressure sensor, an long baseline system, and an attitude sensor. Ribas *et al.* [62] addressed the development of the Girona500 AUV that implements dead reckoning navigation based on a solid-state attitude and heading reference system (AHRS) and a DVL. Additionally, their study included the absolute position through a USBL system while the vehicle is underwater and using a GPS signal while it is on the water surface. The high-accuracy USBL system enables underwater localization and communication between the support and AUV platforms. This thesis employs the Girona500 AUV as the advanced underwater platform, which involves the sensor integration for localization and situational awareness capabilities of the AUV. Additionally, Font *et al.* [20] addressed a USBL-aided navigation method in an AUV. Their method included the state estimation based on a two-parallel extended Kalman filter with the data gathered from a pressure sensor, a GPS, a DVL, an inertial measurement unit, and a visual odometer.

2.2 Modeling and Simulation of Autonomous Offshore Vehicles

An accurate AOV model is essential for developing navigation algorithms, control methodology design, and simulation studies. The AOV model mainly involves two parts in the study of dynamics: kinetics, which analyzes the forces causing AOV displacement, and kinematics, which only handles the geometrical aspects of motion. The design and modeling of the AOV can use the theoretical six degrees of freedom (DOFs) dynamic model based on nonlinear equations of motion [25]. From the complete six DOFs dynamic model, USVs can reduce their order model to three DOFs involving surge, sway, and yaw motion control in the horizontal plane [25].

Additionally, the heading autopilot, which controls the yaw motion in the USV, can include the model representation of Nomoto [54]. Other studies, such as Han *et al.* [30], suggested using a nonlinear modeling system for a waterjet-propelled USV. Similarly, the development of an AUV dynamic model and simulation are also crucial for developing the GNC algorithms. An AUV model can help evaluate the structure, the thruster configuration, GNC algorithms, and environmental forces without losing the AUV platform. Evans and Nahon [18] formulated a dynamics model of a streamlined underwater vehicle, validating the model by using field test data. Nonetheless, most studies have used the theoretical six DOFs dynamic model for the mathematical model of the AUV [26, 61, 67].

Determining the mathematical model coefficients is a complex and laborious procedure because of the coupled terms and the nonlinear characteristics present in the AOV model. System identification (SI) could be convenient for defining an accurate model utilizing field test data for simulation study purposes [44]. Several studies have used the SI method for USV approaches, such as Moreno-Salinas *et al.* [50] with SI using the Nomoto model, Shin *et al.* [71] with SI based on particle swarm optimization, or Oh *et al.* [55] with a SI method for the three DOFs ship maneuvering model. Additionally, the parameter estimation methods [4] can also determine the required mathematical model coefficients. Some tools can accurately estimate these coefficients for the required transfer functions and dynamic model equations. In this case, the parameter estimation [76] and SI [77] tools from MATLAB-Simulink can develop the required AOV mathematical model utilizing field test data. Parameter estimation and SI procedures can also estimate the dynamic model coefficients utilizing field test data in the underwater platforms. Numerous research studies include these methods to develop AUV mathematical models. Furthermore, in this case, AUVs can reduce their order model to four DOFs involving surge, sway, heave, and yaw motions control. Kim *et al.* [38] proposed the estimation of the hydrodynamic coefficients based on the extended Kalman filter and sliding mode observer nonlinear observers. Additionally, Cardenas and de Barros [11] utilized an identification approach by combining an analytical and semi-empirical estimation method with a parameter estimator based on the extended Kalman filter.

The simulation tools are crucial in autonomous applications because several aspects need to be integrated and tested in the vehicle. There are numerous simulation tools available for offshore vehicles. These tools include the Gazebo simulator [39],

which simulates multiple robots in a 3D environment, including extensive dynamic interaction between objects, and the unmanned underwater vehicle simulator [45], which adds simple hydrodynamics features to the Gazebo simulator. Additionally, the Stonefish simulation tool [13] delivers advanced hydrodynamics based on actual vehicle geometry and offshore sensors and actuators simulation. Furthermore, the Marine Systems Simulator [23] is a MATLAB and MATLAB-Simulink library for marine systems. In general, the MATLAB-Simulink software can interactively simulate a system model and show the results on scopes and graphical displays, allowing for the design, modeling, and simulation in the same tool. The current thesis uses the MATLAB-Simulink tools for the design, modeling, and simulation of all studied AOVs, here using its SI and parameter estimation tools to estimate the hydrodynamic coefficients in the AOVs dynamic models. It provides a simple approach compared with the use of multiple simulation tools.

2.3 Guidance, Navigation, and Control Methods in Autonomous Offshore Vehicles

Marine interventions require an autonomous functionality, covering all AOVs navigational functions. Thus, it is necessary to select a specific degree of autonomy that commonly mixes human and system-operated tasks. Table 2.1 illustrates the levels of autonomy that these systems exhibit.

Table 2.1 Description of the levels of autonomy in navigation purposes [17].

Autonomy level	*Description of autonomy level*
M	Manually operated function.
DS	System decision supported function: the mission is executed by the human operator with support from the system.
DSE	System decision supported to function with conditional system execution capabilities. This level is referred to as "human in the loop" because it always requires a human before execution.
SC	Self-controlled function: the system will execute the operation despite that the person in charge can revoke the action. This level also refers to as "human on the loop".
A	Autonomous function: the system will execute the operation without any possibility for the operator to intrude on the functional level.

Regarding the AOVs' operation, the fundamental elements usually incorporate

the GNC subsystems, as follows [21, 24, 25]:

1. The guidance system generates and updates smooth, feasible, and optimal trajectory commands to the control system utilizing the data given by the predefined missions, the navigation system, environmental conditions, and AOV capability.

2. The navigation system identifies the AOV current and future states, which include the pose (position and orientation), velocity, acceleration, and the AOV's surrounding environment using the past and present states of the vehicle along with the environmental information gathered from its onboard sensors.

3. The control system determines the necessary control forces and moments to be delivered together with the instructions from the guidance and navigation systems while satisfying the desired control objectives.

The primitive guidance and control system of an AOV includes both an attitude and path-following control system. The attitude control system incorporates a heading autopilot where roll and pitch are usually left uncontrolled or regulated to zero. Its primary function is to keep the offshore vehicle in a desired attitude for the predefined path. The path-following controller tries to maintain the AOV on the predefined route, generating commands for the attitude control system. It commonly works as a heading controller with a surge controller in USVs, while AUVs also require a depth controller.

More sophisticated and hazardous applications require solving numerous technical challenges to improve the autonomy of the system. These challenges include more advanced collision avoidance capabilities within further AOV development. Unfortunately, current research has mainly involved the avoidance of stationary and slow-motion obstacles. Thus, the availability of more reliable, effective, and accurate methodologies to evade static and dynamic objects are a relevant interest for further investigation. The generated route needs to be obtained in real-time, integrating surrounding stationery and dynamical obstacles, AOV dynamics, and nautical chart data. Meanwhile, in a protocol-based case, the establishment and implementation of regulations in the USV obstacle avoidance approach present an enormous challenge because the navigation rules are only devised for human operators to steer marine crafts.

2.3.1 Path-following Algorithms

A guidance system is an indispensable component for increasing the AOVs autonomy. It provides advanced guidance capabilities in demanding scenarios under more complicated and strict constraints. The guidance laws for path following are highly relevant for the research and development of AOVs. The following planar guidance laws determine the path-following motion control and target tracking objectives [25]:

- Line-of-sight (LOS) guidance is listed as a three-point guidance system because it includes a generally continual reference point along with the target and interceptor.
- Pure pursuit guidance refers to the two-point guidance systems that only consider the target and interceptor in the engagement geometry. A vector pointing directly at the objective represents the pure pursuit guidance principle.
- Constant bearing guidance is another two-point guidance system, here with equal engagement geometry as the previous pure pursuit guidance. The difference is that the interceptor is assumed to align the LOS vector within the interceptor and the target along the interceptor-target velocity vector.

The LOS family of guidance laws has proven to be well suited for underactuated offshore vehicles. In short, the LOS algorithm mimics an experienced helmsman steering a ship by aiming toward a point that lies on the path ahead of the AOV. The LOS path-following law can also be directly applied to a curved route, making the vehicle steer toward the path tangential. Most studies for path-following in offshore applications have included a free obstacle scenario using a guidance-based algorithm [9] or the LOS algorithm [52]. Current progress on path-following mainly focuses on improving the control performance with external disturbances [82]. There are numerous studies for path-following using LOS algorithms, such as the enclosure-based LOS, integral LOS, and adaptive LOS.

In enclosure-based LOS, as described in [25], the vehicle is directed toward a point defined as one of the two intersection points between a circle centered on the platform and the desired path. It can be viewed as a lookahead-based approach with an implicitly time-varying lookahead distance, in which the cross-track error depends on the lookahead distance.

Notwithstanding the simplicity and effectiveness of the proportional guidance laws, their limitations appear when environmental elements, such as wind, waves, and ocean currents, expose an offshore vehicle to unknown drift forces. Underactuated offshore vehicles usually contain speed and heading control in the horizontal plane, and they present substantial cross-track errors throughout steady-state and path-following missions. These errors depend on the route shape, along with the direction and value of the drift force. Thus, the LOS guidance law needs to be modified to incorporate an integral action, which refers to integral guidance laws. In this case, Breivik and Fossen [10] confirmed that the integral guidance could remove the steady-state cross-track error in a straight line path-following scenario. Borhaug *et al.* [7] presented a more sophisticated approach with a globally stable nominal system for constant forward speed in a straight line path-following mission. They included the cascade of the integral guidance law and motion controller, ensuring asymptotic tracking and compensating for the drift caused by environmental disturbances. Fossen and Lekkas [22] presented a nonlinear adaptive path-following algorithm based on the classical LOS guidance method, here estimating and compensating ocean currents for marine crafts. Their algorithm produced a new conceptual integral LOS guidance law that adequately compensates for time-varying drift forces due to waves, wind, and ocean currents. The implementation of most of these studies occurs in a free obstacle path scheme. Thus, their guidance and control systems avoid obstacle avoidance capabilities. The integral LOS guidance law has been selected for the guidance and control system with situational awareness capabilities in the current thesis without environmental forces estimation. Furthermore, the guidance and control system includes simple position and velocity controllers for the AUV. The USV and AUV platforms incorporate this LOS guidance law because of its proven well-suited performance for underactuated offshore vehicles.

Other control techniques in AUVs can include a constrained self-tuning controller for the heading and diving motions [66] and a unified receding horizon optimization system for the integrated path planning and tracking control [70]. Additionally, Liang *et al.* [42] addressed a 3D path-following control for underactuated AUVs with parameter contingencies and external disturbances.

2.3.2 Obstacle Avoidance Algorithms

Obstacle avoidance is the process of avoiding collisions, in which the AOV follows its planned trajectory and avoids any possible physical contact. Liu *et al.* [43] and Tam *et al.* [78] presented various categorizations of collision prevention techniques, including path planning, route planning, and reactive obstacle avoidance. Additionally, Huang *et al.* [35] offered an overview of collision prevention techniques for either manned ships or unmanned ships, here based on conflict detection, conflict resolution, and motion prediction. Reactive obstacle avoidance aims at avoiding previously unknown or moving obstacles. The obstacle avoidance problem, particularly in two dimensions (2D), has been thoroughly studied by the scientific community. Heidarsson and Sukhatme [34] addressed the use of a forward-facing profiling sonar for obstacle avoidance on a USV.

An approach to combine both obstacle avoidance capabilities and path-following can be created using safety boundary boxes (SBBs) encompassing a static or moving obstacle. Simetti *et al.* [73] studied the inclusion of SBBs for collision avoidance, associating a boundary box for each detected target. They aimed to determine the optimal route while evading every box. Additionally, Wu *et al.* [88] included a multi-layer obstacle avoidance based on a single LiDAR; they presented an effective approach for USV path planning when sensor errors and collision risks appear. This was done by establishing a safety box for obstacle recognition. The SBB approach is selected for obstacle avoidance in the current thesis, allowing for fast decision-making capabilities because of its simplicity and low data transfer.

In a 3D environment, Wiig *et al.* [84] proposed a constant avoidance angle algorithm for evading moving obstacles in a 3D environment, here keeping a minimum safety distance from the moving object. Additionally, Vidal *et al.* [81] presented a novel motion planning framework that can generate trajectories involving the safety of an underwater vehicle and its dynamic constraints, as well as incorporating the conventional approaches of inevitable collision states.

USVs operating in populated area waterways should obey compliance with existing rules while also having safe and efficient control. These rules include the collision regulations established by the convention on international regulations for preventing collisions at sea (COLREGs) [16]. Concerning COLREGs in USV operations, Wang *et al.* [83] summed up the prefatory research outcomes of an innovative obstacle

avoidance strategy. Moreover, Moe and Pettersen [49] presented a collision avoidance algorithm for an underactuated USV in a simulation scenario, here ensuring a path-following mission while keeping to the COLREGs. The current thesis has the development of the GNC architecture for path-following and obstacle avoidance as the main focus. Thus, the COLREGs have not been implemented but will be considered in future research.

2.3.3 Guidance, Navigation, and Control Architectures

The software and hardware architectures of AOVs are similar to well-defined architectures because they allow for effective engineering development and deployment of comprehensive systems. Hence, the AOV architecture needs to be divided into particular levels of abstraction. These levels include the fundamental computing infrastructure, including processors and operating systems, the inter-application communications infrastructure and services, which are defined as middleware, and the secondary support infrastructure [14]. The adoption of suitable architectures enables the implementation of formal approaches for building reliability into autonomy. It allows for verification and certification of the AOVs' operations by implementing structural, mathematical, and algorithmic methods for modeling reliability and safety. Furthermore, suitable architectures evolve several approaches to increase the safety and reliability of AOVs.

The use of commercial off-the-shelf hardware for primary infrastructure components, such as the operating systems, communication protocols, and middleware, which ensures a degree of independence in the hardware and software of the AOV. The Robot Operating System (ROS) is an open-source middleware in robotics for writing robot software [60]. It is a compilation of libraries and tools that simplifies the mixed and robust robot performance across numerous robotic platforms. This tool can include data acquisition and processing from sensors, hence producing the required commands for the vehicle actuators. Regarding the case of system connectivity, Alberri *et al.* [2] designed and implemented a high-performance, low-cost, and nonexclusive multi-layer architecture based on ROS for autonomous systems.

Currently, MATLAB-Simulink is a software tool that enables C and C++ code generation from the MATLAB-Simulink models for deployment in several applications [46]. In general, MATLAB-Simulink is a block diagram environment commonly

employed for model-based and multi-domain simulation designs. This tool can assist in system-level design, simulation purposes, and automatic code generation adopting coding standards. MATLAB-Simulink can generate standalone ROS nodes to support the GNC architecture implementation. Thus, it provides the design, simulation, and implementation of the modular GNC algorithms using the same tool.

2.4 Guidance, Navigation, and Control Methods for Co-operative Systems

With the increment of robotic applications, it has become more common to involve multiple systems simultaneously in co-operation. Multi-vehicle systems can produce several advantages for perception systems compared with an individual vehicle implementation as more useful information may become available. Thus, it is required to fuse together information gathered from the single platforms to benefit from these advantages. Additionally, using varied robotic platforms often requires considering different sensor types among their specific measurements. Hence, there are some challenges in co-operative systems, such as task allocation and coordination, communications, information exchanged, or time synchronization. USVs usually co-operate with other autonomous vehicles, such as AUVs and UAVs, to accomplish more effective offshore missions. However, GNC methods can be relatively complex, so it is becoming crucial to fuse together the data gathered from individual vehicles.

The classification of multi-vehicle systems includes decentralized systems, with each AOV running an independent ROS master or centralized ones with the master node located at the ground control station. Decentralized systems are more effective and usually decrease the communication network conditions compared with centralized systems [6]. Nevertheless, decentralized systems are more complex because of contingencies and communication limitations, such as delays, noises, or simple failures. Hence, a multi-master approach can provide answers because each platform runs its ROS master and exchanges the required data with other components. Tiderko *et al.* [79] presented a ROS package that can accurately develop multi-master architectures. Insaurralde [36] proposed an intelligent control architecture to enable multiple offshore vehicles to perform autonomous underwater applications and used the control architecture in a case study where a USV and AUV work co-operatively toward

accomplishing complex activities. By using a decentralized modular and multi-layer GNC architecture with a multi-master approach, this allows for the testing of each offshore vehicle separately and the inclusion of new platforms, if necessary. Thus, the decentralized system improves the performance of the autonomous operation for the presented co-operative system.

3 MODELING AND MODEL VALIDATION OF THE AUTONOMOUS OFFSHORE VEHICLES

This chapter presents the mathematical models for the AOVs used in this thesis. This chapter also describes two approaches for estimating the dynamic model parameters. Obtaining an accurate simulation and performance are the two objectives of these mathematical models in the designed GNC algorithms. Additionally, this chapter provides the model validation in the USV and AUV mathematical models using field test data. The contents of this chapter are based on **Publications II—V**.

3.1 Overview of the Autonomous Offshore Vehicles

The design and modeling of the AOV needs six independent coordinates to define the pose for a moving AOV. The first three coordinates — consequently their time derivatives — correspond to the position and translational motion on the (x, y, z) axes. Similarly, the other three coordinates and their time derivatives define the orientation and rotational motion. Figure 3.1 shows the illustration of the surge u, sway v, and heave w linear velocities, along with roll p, pitch q, and yaw r angular velocities in the representation of the six motion components for an AOV. Furthermore, it is convenient to define the geographic reference frames used in the GNC subsystems [25]:

- North-east-down (NED) coordinate system: This coordinate system $\{n\} = (x_n, y_n, z_n)$ is determined relative to the earth reference ellipsoid with origin o_n. The x-axis looks towards true north, the y-axis aims towards east, and the z-axis points downwards normal to the earth's surface. The longitude and latitude angles determine the location of $\{n\}$ relative to the earth-centered earth-fixed reference frame $\{e\}$.

- East-north-up (ENU) coordinate system: This is similar to the NED coordinate system, but the x-axis looks east, the y-axis aims the true north, and the z-axis points upwards normal to the earth's surface.
- Body-fixed (BODY) reference frame: This reference frame $\{b\} = (x_b, y_b, z_b)$ is a moving coordinate frame fixed to the AOV with origin o_b. The longitudinal axis x_b is directed from the aft to fore, the transversal axis y_b is directed to starboard, and the normal axis z_b is directed from top to bottom.

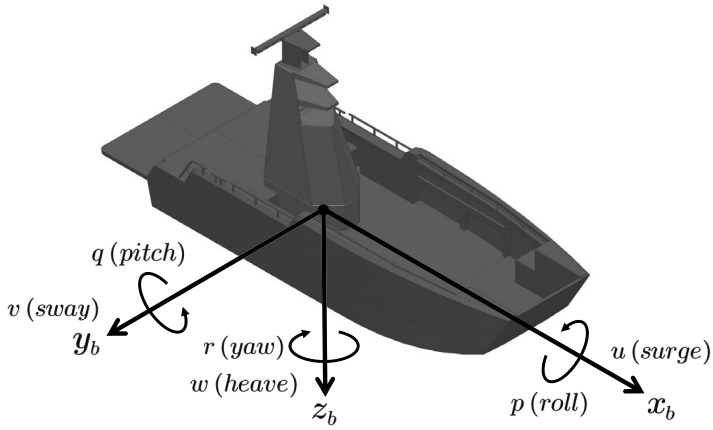


Figure 3.1 Six motion components for an AOV in the BODY reference frame.

The two AOVs used in the current thesis are a USV and AUV. Concerning the AUV, this thesis includes two particular platforms because of the possibility of using cutting-edge sensors technology in one of them. The USV utilized in the current thesis is a catamaran with an aluminum hull that includes a thrust-vectoring waterjet propulsion system. The USV has excellent maneuverability because of the twin waterjet propulsion system, facilitating the motion in all planar directions without using the bow and stern thrusters, as shown in Figure 3.2. This thesis work includes a simplification for the USV motion from six to three DOFs for planar maneuverability. The selected three DOFs are the surge, sway, and yaw motions, neglecting the roll, pitch, and heave ones.

Figure 3.3a shows the simplified USV model utilized in the autonomous offshore system. The twin waterjet-propelled vehicle employs the starboard (STDB) and port waterjet units to produce the necessary thrust forces to move in the 2D environment. The maneuvers include the forward, backward, and sideways motions and performing turns. Furthermore, Figure 3.3a shows the USV pose, including position and

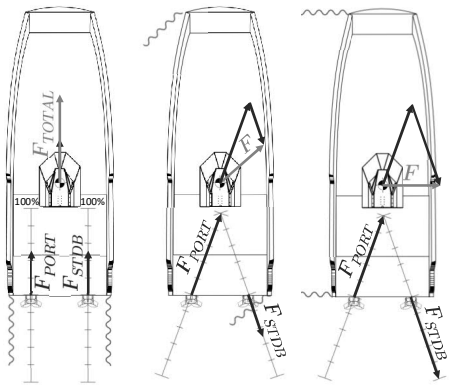


Figure 3.2 Twin waterjet USV maneuvering for surge, turning, and sideways motion. [**Publication III**]

orientation, here based on the NED coordinate system. The navigation system uses a GPS compass installed in the USV, requiring a coordinate transformation to calculate the USV's absolute position in the planar coordinate system. This transformation for the vehicle position is between World Geodetic System 84, which produces the longitude and latitude angles, and the ETRS-TM35FIN [56], which employs the NED coordinates system (N_{USV}, E_{USV}). The USV heading ψ is defined based on the attitude or Euler angles. Finally, the relative positioning uses the BODY reference frame represented by surge u or linear longitudinal motion, sway v or linear transverse, and yaw motion r or turning rotation around its z-axis.

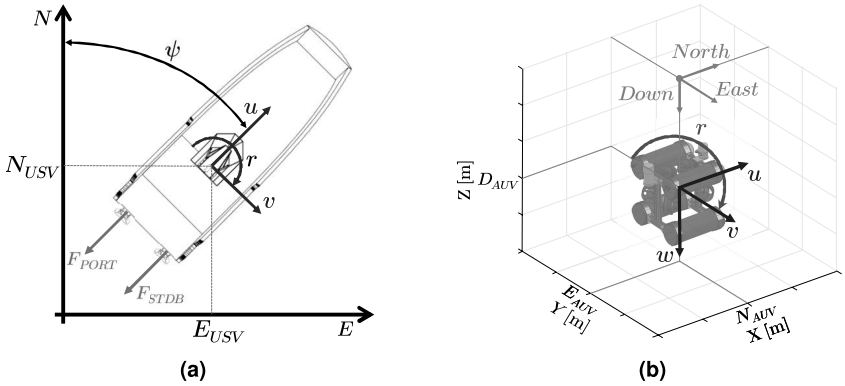


Figure 3.3 Simplified model of the considered offshore vehicles based on the NED coordinate system: (a) aColor USV. (b) aColor AUV. [**Publication IV**]

Regarding the AUVs, the current thesis uses two high-configurable underwater vehicles with diverse scientific instrumentation. The first platform is the aColor

AUV, which includes only essential instrumentation and sensors to perform AUV localization and target detection. The onboard sensors include an attitude sensor, a depth sensor, and a USBL system for underwater navigation and localization. Additionally, the AUV incorporates a mechanical imaging sonar as the main underwater active ranging sensor. Figure 3.3b illustrates the simplified model of the aColor AUV and the AUV position and velocities, whose AUV motion in six DOFs uses the NED local coordinate system. Additionally, Figure 3.4a illustrates the thrust forces for each thruster to perform the 3D maneuverability, and Figure 3.4b shows the distances between the application point for the thrust forces with the center of mass. This AUV employs a six-thruster configuration, hence providing the necessary thrust forces to move in the 3D environment. These distances define the thruster configurability matrix for further use in the AUV mathematical model and are denoted as $\boldsymbol{l_n} = [l_{xn}, l_{yn}, l_{zn}]$ for the n thruster.

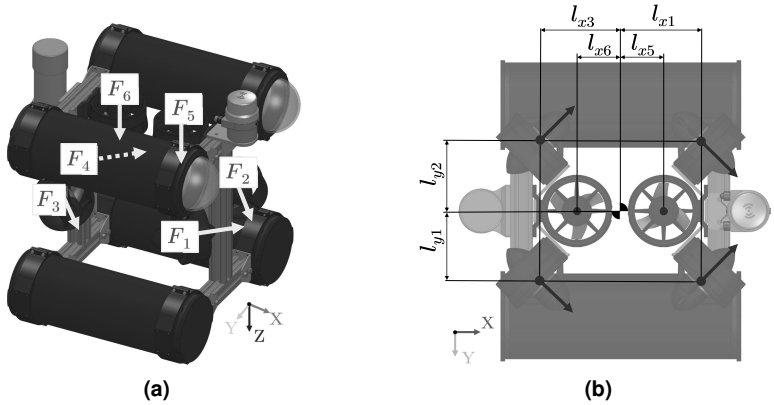


Figure 3.4 Six-thruster configuration in the aColor AUV: (a) Thrust forces indicating each thruster direction. (b) Distances from each thruster to the center of mass of the AUV.

The second underwater vehicle used in this thesis is the Girona500 AUV [62], which provides high configurability for cutting-edge scientific instrumentation and allows for the use of the ROS framework. The instrumentation and sensors utilized in this vehicle are DVL, depth sensor, and AHRS for the AUV navigation, along with a mechanical imaging sonar as chosen underwater perception sensor. Figure 3.5a illustrates the interpreted model of the Girona500 AUV. This AUV employs a five-thruster configuration to provide the necessary thrust forces when moving in a 3D environment.

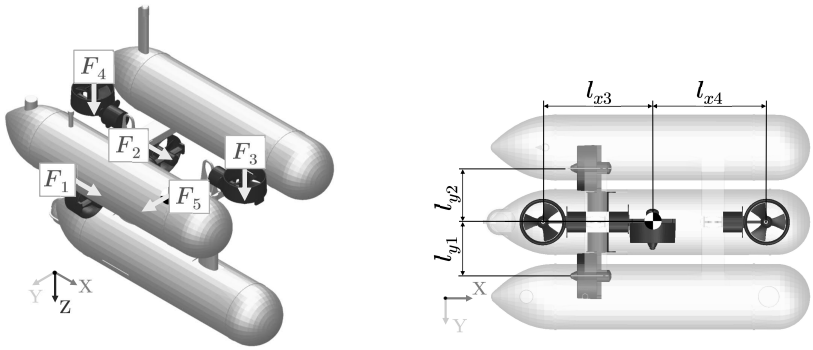


Figure 3.5 Five-thruster configuration in the Girona500 AUV: (a) Thrust forces indicating each thruster direction. (b) Distances from each thruster to the center of mass of the AUV. [**Publication II**]

In general, the AUV modeling in six DoF uses the NED local coordinate system, whose position and velocities definition are declared as follows:

$$\eta = [N, E, D, \phi, \theta, \psi]^\top, \nu = [u, v, w, p, q, r]^\top, \tag{3.1}$$

where $[N, E, D]$ indicate the NED positions in the earth-fixed coordinates, $[\phi, \theta, \psi]$ define the Euler angles, and then, based on the BODY reference frame, $[u, v, w]$ and $[p, q, r]$ denote the linear and angular velocities, respectively [25].

3.2 Mathematical Model of the Unmanned Surface Vehicle

As mentioned in Chapter 2, the development of an adequate mathematical model allows for the design and simulation simplification of the GNC algorithms. Thus, it is a crucial stage for the development of each of the AOVs. This section studies the USV mathematical model, which involves the propulsion, power system, and kinematics and dynamics models. First, **Publication III** considers the Nomoto autopilot model as a simple mathematical model for developing GNC algorithms. Then, the reduced three DOFs dynamic and waterjet propulsion system models are described based on **Publication IV**, increasing the accuracy and capabilities of the mathematical model for better development of future GNC algorithms.

3.2.1 Nomoto Autopilot Model

As mentioned above, a satisfactory maneuvering model will benefit the GNC algorithm development for the autonomous offshore system. Regarding the guidance system, the current thesis uses a path-following control, in which the vehicle travels by a constant surge velocity u. Meanwhile, the heading controller tries to reduce the cross-track error $e(t)$ to the requisite path. Concerning the use of heading controllers in offshore applications, numerous marine crafts utilize the model representation of Nomoto [54], deriving its autopilot model from the linearized USV maneuvering model. Nomoto's second-order model applies to the following:

$$\frac{r}{\delta}(s) = \frac{K_{\mathrm{p}}(1+T_{\mathrm{z}}s)}{(1+T_{\mathrm{p1}}s)(1+T_{\mathrm{p2}}s)}. \tag{3.2}$$

where δ is the rudder angle and r is the USV angular velocity. The transfer function includes a process gain (K_{p}), two poles (T_{p1}, T_{p2}), and one zero (T_{z}).

Nomoto's model representation is quite simple, which is its foremost benefit. Furthermore, field test data directly allows for the definition of Nomoto's model parameters. Thus, the model does not need to compute the hydrodynamic derivatives explicitly. **Publication III** uses Nomoto's model representation in the USV definition. The only difference from Nomoto's second-order model indicated in the Equation (3.2) is that **Publication III** substituted the rudder angle with the nozzle angle of the waterjet propulsion system.

3.2.2 Three Degrees-of-Freedom Dynamic Model

According to the six DOFs dynamic model established by nonlinear equations of motion, it is common to reduce its order to a three DOFs model in the horizontal plane for USV maneuvering. This model is used and described in **Publication IV**, comprising the following rigid-body kinetics [25]:

$$\boldsymbol{M}\dot{\boldsymbol{\nu}} + \boldsymbol{C}(\boldsymbol{\nu})\boldsymbol{\nu} + \boldsymbol{D}(\boldsymbol{\nu})\boldsymbol{\nu} = \tau + \tau_{\mathrm{wind}} + \tau_{\mathrm{wave}}, \tag{3.3}$$

where $\boldsymbol{\nu} = [u, v, r]^{\intercal}$ defines the velocity vector, including the surge, sway, and yaw motions. $\boldsymbol{M}$, $\boldsymbol{D}(\boldsymbol{\nu})$, and $\boldsymbol{C}(\boldsymbol{\nu})$, are the mass, damping, and Coriolis matrices, respectively.

Both $\boldsymbol{M}$ and $\boldsymbol{C}(\boldsymbol{\nu})$ include rigid body and added terms. Additionally, $\tau = [\tau_{\mathrm{u}}, 0, \tau_{\mathrm{r}}]$ is the forces and moments vector produced by the twin waterjet propulsion system, while τ_{wind} and τ_{wave} determine the environmental forces. The mass matrix $\boldsymbol{M}$ is determined by

$$\boldsymbol{M} = \boldsymbol{M}_{\mathrm{RB}} + \boldsymbol{M}_{\mathrm{A}} = \begin{bmatrix} m - X_{\dot{u}} & 0 & 0 \\ 0 & m - Y_{\dot{v}} & m x_{\mathrm{g}} - Y_{\dot{r}} \\ 0 & m x_{\mathrm{g}} - Y_{\dot{r}} & I_{\mathrm{z}} - N_{\dot{r}} \end{bmatrix}, \tag{3.4}$$

where m defines the vehicle mass, I_{z} the moment of inertia for z_{b} axis, $\boldsymbol{r}_{\mathrm{g}}^{b} = [x_{\mathrm{g}}, y_{\mathrm{g}}, z_{\mathrm{g}}]^{\intercal}$ regarding the distance vector from origin o_{b} to the centre of gravity and where $X_{\dot{u}}$, $Y_{\dot{v}}$, $Y_{\dot{r}}$, and $N_{\dot{r}}$ represent the hydrodynamic added mass. **Publication IV** includes the estimation of the moment of inertia I_{z} at the pivot point utilizing the moments of inertia calculation in the front $I_{\mathrm{z,front}}$ and rear $I_{\mathrm{z,rear}}$ of the USV. These moments of inertia are determined by

$$I_{\mathrm{z,rear}} = m_{\mathrm{pt}}\, l_{\mathrm{pt}}^{2} + \left(\frac{1}{3}\, m_{\mathrm{hull}}\, c_{\mathrm{g}}\right) l_{\mathrm{pivot}}^{2}, \tag{3.5}$$

$$I_{\mathrm{z,front}} = \frac{1}{3}\, m_{\mathrm{hull}} \left(1 - c_{\mathrm{g}}\right) \varkappa \left(L_{\mathrm{USV}} - l_{\mathrm{pivot}}\right)^{2}, \tag{3.6}$$

where

m_{pt} is the estimated powertrain mass including the waterjets units, engines, fuel, and so forth,

m_{hull} relates exclusively to the hull weight without the powertrain mass,

l_{pt} defines the estimated powertrain mass location,

c_{g} defines the relative center of mass point (equal to one in front of the USV),

l_{pivot} defines the pivot point location,

$\varkappa$ is a scaling factor because the vehicle mass is not symmetrically distributed,

L_{USV} is the total length of the USV.

Then, the total moment of inertia I_z is determined by Equation (3.7), where I_{cor} defines the moment of inertia tuning factor.

$$I_{\mathrm{z}} = \left(I_{\mathrm{z,rear}} + I_{\mathrm{z,front}}\right) I_{\mathrm{cor}}, \tag{3.7}$$

The parametrization of the Coriolis-centripetal matrix $\boldsymbol{C}(\boldsymbol{\nu})$ can ensure that $\boldsymbol{C}(\boldsymbol{\nu}) = \boldsymbol{C}^{\intercal}(\boldsymbol{\nu})$ [68]. Nonetheless, the linearization of these Coriolis-centripetal terms $\boldsymbol{C}_{\mathrm{RB}}(\boldsymbol{\nu})$ and $\boldsymbol{C}_{\mathrm{A}}(\boldsymbol{\nu})$ for an angular velocity equal to zero suggest that they can be removed from the above expressions [21]. Furthermore, the USV mathematical model is simplified to include only surge and yaw motions; thus, the Coriolis-centripetal terms are discarded at the reduced-order dynamic model.

The diverse damping terms add to both linear and quadratic damping elements of the dynamic model [25]. However, it is usually complicated to differentiate among those effects. Thus, the total hydrodynamic damping matrix $\boldsymbol{D}(\boldsymbol{\nu}_r)$ is the sum of the linear $\boldsymbol{D}_{\mathrm{lin}}$ and the nonlinear $\boldsymbol{D}_{\mathrm{nlin}}(\boldsymbol{\nu}_r)$ terms, ensuring the following:

$$\boldsymbol{D}(\boldsymbol{\nu}_r) = \boldsymbol{D}_{\mathrm{lin}} + \boldsymbol{D}_{\mathrm{nlin}}(\boldsymbol{\nu}_r), \tag{3.8}$$

where $\boldsymbol{D}_{\mathrm{lin}}$ is the linear damping matrix obtained from possible skin friction and potential damping and $\boldsymbol{D}_{\mathrm{nlin}}(\boldsymbol{\nu}_r)$ defines the nonlinear damping matrix resulting from the higher-order terms and quadratic damping, determined by

$$\boldsymbol{D}_{\mathrm{lin}} = -\begin{bmatrix} X_u & 0 & 0 \\ 0 & Y_v & Y_r \\ 0 & Y_r & N_r \end{bmatrix}, \tag{3.9}$$

$$\boldsymbol{D}_{\mathrm{nlin}}(\boldsymbol{\nu}_r) = -\begin{bmatrix} X_{|u|u} & 0 & 0 \\ 0 & Y_{|v|v} & 0 \\ 0 & 0 & N_{|r|r} \end{bmatrix} |\boldsymbol{\nu}_r|, \tag{3.10}$$

where X_u, Y_v, Y_r, N_r represent the linear damping coefficients and $X_{|u|u}$, $Y_{|v|v}$, $N_{|r|r}$ are the nonlinear damping coefficients.

3.2.3 Waterjet Propulsion System

Commonly, the number of actuators installed in the USV is less than the number of DOFs in motion. It represents a safe and precise control challenge in the operation of underactuated USVs. The underactuated vehicle control includes surge and yaw motions, which are provided by single propulsion systems formed by a rudder and

propeller or a waterjet. Other USV platforms, principally a catamaran hull as a structural element, are steered by differential thrusts produced by two independent engines installed in each hull. This last configuration is the one used in the USV for the current thesis.

The waterjet propulsion unit has its propulsion thrust created by the reaction force. The mass flow from the propulsion unit impeller produces this reaction force, which is caused by its kinetic energy. **Publication IV** describes the propulsion system with AJ245 waterjet units installed in the USV platform [1]. The nozzle position P_{nozzle} modifies the jet flow direction, hence generating the required force needed for turning. The total thrust force F_{total} definition involves the waterjet engine rpm ω_{rpm} and P_{nozzle}. The variable ω_{rpm} is obtained from the waterjet engine, and P_{nozzle} is chosen as a dimensionless variable from -10.000 to 10.000, with a value equal to zero in the neutral position for forwarding motion. Table 3.1 includes the manufacturer's data provided for the waterjet propulsion units at a particular operating point. These data include an operating point for engine rpm equal to 1800 rpm, bucket in the full-up position, and nozzle placed at the neutral position ($P_{\text{nozzle}} = 0$).

Table 3.1 Propulsion system data obtained for the specific operating point of the AJ245 waterjet propulsion unit [**Publication IV**].

Surge Speed [kt]	Thrust Force [kN]
2	2
4	1.85
6	1.7

The waterjet mathematical model calculates the thrust forces and torques based on the manufacturer's data and affinity law. To combine the data and affinity law, the mathematical model includes a 2D lookup table for the relation between the thrust force per waterjet F and the shaft rotational speed of the waterjet engine ω. The affinity law adopted to calculate the thrust force at each of the waterjet propulsion units is determined as follows:

$$\frac{F_1}{F_2} = \left(\frac{\omega_1}{\omega_2}\right)^2. \tag{3.11}$$

Figure 3.6 includes the results for this affinity law using the provided operating point data for a single waterjet engine. The waterjet engine range is defined as 600 to

2400 rpm, hence suiting the operational waterjet engine speeds in the USV control scenarios.

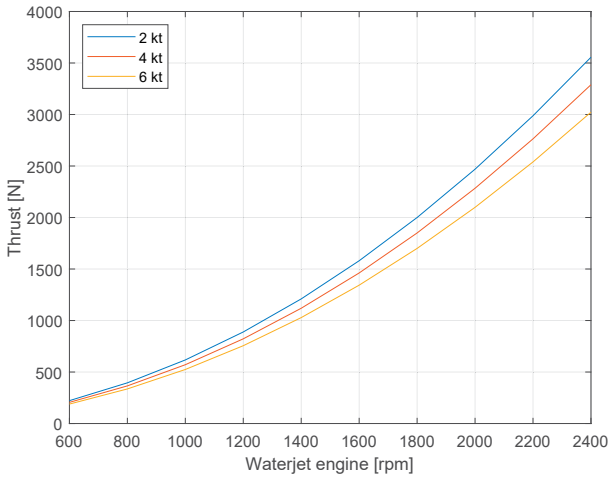


Figure 3.6 Affinity law results for the waterjet propulsion unit, where the generated thrust force F depends on the shaft rotational speed ω. [**Publication IV**]

Regarding the USV yaw motion, the total thrust force efficiency $\mathscr{E}_{\text{nozzle}}$ is determined by the waterjet nozzle position. This nozzle position refers to the waterjet thrust force angle α_{nozzle}. Based on the waterjet manufacturer experiments, this efficiency decreases exponentially to 30—40% of the maximum thrust force at the center position when the nozzle position is varied to the upper or lowest limit for the nozzle angle $\alpha_{\text{nozzle}} = \pm 25°$, which is equivalent to $P_{\text{nozzle}} = \pm$ 10,000. The adopted exponential function uses the general exponential model declared as

$$\mathscr{E}_{\text{nozzle}}(P_{\text{nozzle}}) = a \exp(b\, P_{\text{nozzle}}), \tag{3.12}$$

where $a = 1$ and $b = -9.163 \cdot 10^{-5}$. Based on the total efficiency $\mathscr{E}_{\text{nozzle}}$, the vector $\tau = [\tau_{\text{u}}, 0, \tau_{\text{r}}]$, which outlines the forces and moments produced by the waterjet propulsion system, is declared by the following:

$$\begin{cases} \tau_{\text{u}} = (F_{\text{PORT}} + F_{\text{STDB}})\mathscr{E}_{\text{nozzle}} \\ \tau_{\text{r}} = l_{\text{pivot}} \sin(\alpha_{\text{nozzle}})(F_{\text{PORT}} + F_{\text{STDB}})\mathscr{E}_{\text{nozzle}} \end{cases}. \tag{3.13}$$

3.3 Mathematical Model of the Autonomous Underwater Vehicle

The AUVs should include a theoretical six DOFs dynamic model for the correct design and modeling of underwater platforms. However, the dynamic model requires accurate navigation data to estimate its hydrodynamic coefficients. The AUV gathers the navigation data from a series of underwater scientific sensors, such as DVL, AHRS, pressure and attitude sensor, and USBL system. After acquiring the navigation data, the AUV mathematical model can include similar parameter estimation methods than what is described in the USV platform.

3.3.1 Four Degrees-of-Freedom Dynamic Model

The AUV design and modeling have been studied using the theoretical six DOFs dynamic model [26]. **Publication V** describes using a reduced-order dynamic model for the four DOFs control involving surge u, sway v, heave w, and yaw r while ignoring the roll p and pitch q motions. The dynamic model includes simplifying the later parameter estimation of its hydrodynamic coefficients, here considering only the diagonal elements from the theoretical six DOFs dynamic model. For an underwater vehicle, the hydrostatic forces and moments are related to weight and buoyancy, while the hydrodynamic forces and moments involve added mass and damping. The nonlinear equations of motion are similar to what is described in Equation (3.3), with the inclusion of the gravitational/buoyancy forces and moments vector $\boldsymbol{g}$, here ignoring the environmental forces τ_{wind} and τ_{wave}. Thus, it implies the following:

$$\boldsymbol{M}\dot{\nu} + \boldsymbol{C}(\nu)\nu + \boldsymbol{D}(\nu) + \boldsymbol{g}(\eta) = \tau. \tag{3.14}$$

The AUV mathematical model is analyzed considering only single motions similar to the three DOFs USV dynamic model, so Coriolis and centripetal terms $\boldsymbol{C}(\nu)$ are eliminated for the four DOFs dynamic model. The mass matrix $\boldsymbol{M}$ for the case of

four DOFs is defined by

$$\begin{aligned} \boldsymbol{M} &= \boldsymbol{M}_{\mathrm{RB}} + \boldsymbol{M}_{\mathrm{A}} \\ &= \begin{bmatrix} m - X_{\dot{u}} & 0 & 0 & 0 \\ 0 & m - Y_{\dot{v}} & 0 & 0 \\ 0 & 0 & m - Z_{\dot{w}} & 0 \\ 0 & 0 & 0 & I_{\mathrm{z}} - N_{\dot{r}} \end{bmatrix}, \end{aligned} \tag{3.15}$$

where m is the vehicle mass, I_{z} is the moment of inertia about z_{b} axis, and $X_{\dot{u}}$, $Y_{\dot{v}}$, $Z_{\dot{w}}$, and $N_{\dot{r}}$ describe the hydrodynamic added mass.

The actuator forces and moments relate to the control forces and moments by

$$\tau = \boldsymbol{T}\boldsymbol{f}, \tag{3.16}$$

where $\boldsymbol{T}$ is the thrust configuration matrix and $\boldsymbol{f}$ is the control forces and moments vector. There are two thrust configuration matrices $\boldsymbol{T}$ for each of the AUVs used in the current thesis: $\boldsymbol{T}_{\text{aColor-AUV}}$ for the six-thruster configuration in the aColor AUV and the other $\boldsymbol{T}_{\text{Girona500}}$ for the five-thruster configuration in the Girona500 AUV. The thruster configuration for the aColor and Girona500 AUV platforms are illustrated in Figures 3.4a and 3.5a, respectively. These two matrices are defined as

$$\boldsymbol{T}_{\text{aColor-AUV}} = \begin{bmatrix} -1.0 & -1.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.0 & 1.0 \\ 0.0 & 0.0 & -1.0 & -1.0 & 0.0 \\ -l_{\mathrm{y1}} & +l_{\mathrm{y2}} & 0.0 & 0.0 & 0.0 \end{bmatrix}, \tag{3.17}$$

$$\boldsymbol{T}_{\text{Girona500}} = \begin{bmatrix} -1.0 & -1.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.0 & 1.0 \\ 0.0 & 0.0 & -1.0 & -1.0 & 0.0 \\ -l_{\mathrm{y1}} & +l_{\mathrm{y2}} & 0.0 & 0.0 & 0.0 \end{bmatrix}, \tag{3.18}$$

where $\boldsymbol{l_n} = [l_{\mathrm{x}n}, l_{\mathrm{y}n}, l_{\mathrm{z}n}]$ is the distance from the centre of mass of the AUV to the thruster n. Regarding the control of the aColor AUV, thrusters T_1, T_2, T_3, and T_4

act in the surge, sway, and yaw motions, and thrusters T_5 and T_6 affect the heave and roll ones. Similarly, in the case for the Girona500 AUV, thrusters T_1 and T_2 affect the surge and yawing, thrusters T_3 and T_4 affect heave motion, and thruster T_5 affects the sway. Because thruster T_5 is located at the center of mass, it does not produce any rotational motion.

The damping terms involve both linear and quadratic damping [25]. As defined in the three DOFs USV dynamic model, the total hydrodynamic damping matrix $\boldsymbol{D}(\boldsymbol{\nu}_r)$ is the sum of the linear part $\boldsymbol{D}_{\text{lin}}$ and the nonlinear part $\boldsymbol{D}_{\text{nlin}}(\boldsymbol{\nu}_r)$. In the four DOFs dynamic model, these linear and nonlinear damping matrices are defined by the following:

$$\boldsymbol{D}_{\text{lin}} = -\begin{bmatrix} X_u & 0 & 0 & 0 \\ 0 & Y_v & 0 & 0 \\ 0 & 0 & Z_w & 0 \\ 0 & 0 & 0 & N_r \end{bmatrix}, \tag{3.19}$$

$$\boldsymbol{D}_{\text{nlin}}(\boldsymbol{\nu}_r) = -\begin{bmatrix} X_{|u|u} & 0 & 0 & 0 \\ 0 & Y_{|v|v} & 0 & 0 \\ 0 & 0 & Z_{|w|w} & 0 \\ 0 & 0 & 0 & N_{|r|r} \end{bmatrix} |\boldsymbol{\nu}_r|. \tag{3.20}$$

The gravitational force acts through the centre of gravity, while the buoyancy force does this through the center of buoyancy. The submerged weight of the body $\mathscr{W}$ and the buoyancy force $\mathscr{B}$ are declared by

$$\mathscr{W} = mg, \ \mathscr{B} = \rho g \nabla, \tag{3.21}$$

where ρ is the water density, g is the gravity, and ∇ is the fluid volume displaced by the vehicle. Because pitch and roll angles are declared as fixed and equal to zero, the vector of gravitational/buoyancy forces and moments $\boldsymbol{g}$ is defined by

$$\boldsymbol{g} = [0, 0, -(\mathscr{W} - \mathscr{B}), 0]^\top. \tag{3.22}$$

3.3.2 Thruster Propulsion System

The AUVs motions are produced by their thruster configuration, including six thrusters in the aColor AUV and five thrusters in the Girona500 AUV. The thrusters as propulsion systems are included in the mathematical models using one-dimensional (1D) lookup tables. These tables contain detailed performance charts for the thrust forces of each thruster. In the case of the aColor AUV, the thruster installed in the vehicle is the T200 Thruster from BlueRobotics [5]. Figure 3.7 shows the performance chart from the manufacturer, having the pulse-width modulation (PWM) in the electronic speed controller (ESC) as the input and the thrust force as the output. The reverse and forward thruster motions have different thrust forces because of the efficiency of the propulsion system. The same approach has been used in the installed thrusters at the Girona500 AUV.

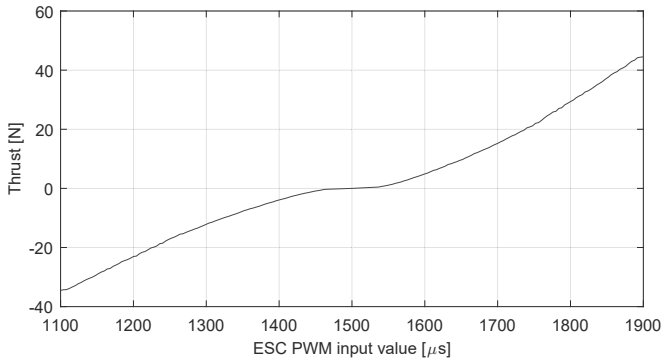


Figure 3.7 Performance chart for the thrust forces in the T200 Bluerobotics thrusters [5].

3.4 Estimation of the Dynamic Model Parameters

After the mathematical model definition for the AOVs, it is possible to estimate the hydrodynamic coefficients using SI or parameter estimation methods. These methods provide a realistic estimation of the necessary parameters by adjusting the mathematical model output to the field test data. This section provides an overview of the definition and procedures for the SI and parameter estimation methods in the USV and Girona500 AUV. The estimated parameters are calculated and identified via offline identification, here based on data gathered from onboard sensors

through extensive experiments. Moreover, this section includes the results for the hydrodynamic coefficients in each of the AOV.

3.4.1 System Identification Method

Publication III studies the Nomoto maneuvering model of the USV platform of the autonomous offshore system. The USV mathematical model includes a constant surge velocity with a variable yaw angle. The MATLAB SI tool [77] has been chosen for simplicity to estimate the necessary transfer functions for the current thesis, as well as for providing similar results to the least-squares support vector machines method [50].

The mathematical model based on the SI method involves two different transfer functions for each surge and yaw motion in the USV platform. After reviewing numerous SI models, the surge motion is determined in Equation (3.23) as a transfer function with a process gain (K_p), two poles (T_{p1}, T_{p2}), one zero (T_z), and an input/output delay (T_d). The waterjet engine rpm ω_{rpm} is defined as the input, while surge velocity u is the transfer function output.

$$\frac{u}{\omega_{rpm}}(s) = \exp(-T_d s)\frac{K_p(1+T_z s)}{(1+T_{p1}s)(1+T_{p2}s)}. \tag{3.23}$$

The yaw motion model uses the same Nomoto's second-order defined in Equation (3.2). Nonetheless, the model has as an input the waterjet nozzle position P_{nozzle} alternatively to the original rudder angle δ included in Nomoto's model. The nozzle position P_{nozzle} produces a particular angular velocity for a given engine rpm ω_{rpm}. The model output is the yaw angular velocity r, here calculating the USV heading angle ψ from its integration.

Table 3.2 comprises the identified transfer function coefficients of both surge and yaw USV motions using the SI method. The u and r variables are selected from their transfer functions from Equations (3.23) and (3.2), respectively. Thus, this mathematical model uses an alternative set of dynamics compared with a marine craft with a propeller and rudder, but the SI approach still achieves the required results.

Publication IV shows a more advanced study of the waterjet propulsion unit model based on the SI method. The USV mathematical model incorporates a 2D lookup table, including the surge USV speed and the waterjet engine rpm as the

Table 3.2 Transfer function coefficients for the surge and yaw USV motions [**Publication III**]

Motion	T_z	T_{p1}	T_{p2}	K_p	T_d
Surge	0.17563	4.08900	0.17299	2.930×10^{-3}	0.8
Yaw	0.09835	1.81108	0.00144	-3.177×10^{-5}	0.0

inputs; here, the total thrust generated by the waterjet propulsion unit is the output. Moreover, the model incorporates a 1D lookup table $f(Joy_u)$ that obtains the waterjet engine rpm based on the joystick controller input for surge motion. Then, a second-order transfer function computes the waterjet engine dynamics into the mathematical model. The MATLAB SI tool helps this transfer function estimation by using the USV field test data, which is similar to the previously described study. Hence, the engine rpm is computed by combining the 1D lookup table and the engine rpm transfer function, which is established by the following:

$$\omega_{\text{rpm}}(s) = \frac{0.317s^2 + 2.793s + 1.828}{s^2 + 3.499s + 1.828} f(Joy_u). \tag{3.24}$$

The waterjet nozzle position incorporates a 1D lookup table $f(Joy_r)$ with a first-order transfer function. Similar to previous transfer functions, the nozzle position of each waterjet uses the MATLAB SI tool to define its parameters compared with the field test data, and it is determined as follows:

$$P_{\text{nozzle}}(s) = \frac{-\exp(-0.25s)}{0.1s + 1} f(Joy_r). \tag{3.25}$$

Figure 3.8 illustrates the comparison between the SI tool transfer functions for both the waterjet engine rpm ω_{rpm} and nozzle position P_{nozzle} variables and USV field test data. This comparison shows the accurate performance of the estimated second-order transfer functions.

3.4.2 Parameter Estimation Approach

The parameter estimation in AOVs is an attractive research topic for many scientists. The main reason is that well-defined mathematical models lead to the optimal design of the system. **Publications IV and V** uses the parameter estimation tool from MATLAB-Simulink [76], here employing time domain methods for the mathematical

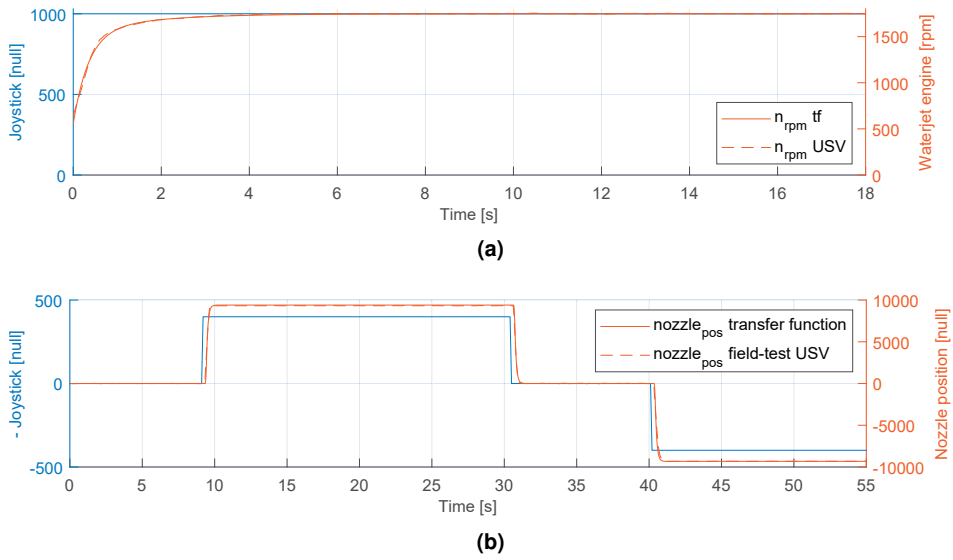


Figure 3.8 Comparison of the SI transfer functions with the USV field test data: (**a**) Waterjet engine rpm ω_{rpm}. (**b**) Nozzle position P_{nozzle}. **[Publication IV]**

models of the USV and Girona500 AUV, respectively. Over parametrization is the main problem when estimating parameters using SI or other nonlinear optimization methods. It is solved by exploiting theoretical information and by estimating a model of partially known parameters [26].

Concerning the three DOFs USV dynamic model, the parameter estimation tool in the MATLAB-Simulink model estimates the matrices $\boldsymbol{M}$ and $\boldsymbol{D}(\boldsymbol{\nu})$ by defining each of the matrices from their input values. After doing this, the estimation tool can determine these individual coefficients in each of the dynamic matrices. The USV mathematical model utilizes two separate parameter estimation runs involving the surge and yaw motions. Table 3.3 includes the fixed values shared in both of these experiments, while Table 3.4 presents the estimated coefficients with their corresponding results. The parameter estimation method considers only surge X_u, $X_{\dot{u}}$, $X_{|u|u}$ and yaw N_r, $N_{\dot{r}}$, $N_{|r|r}$ motion coefficients because the USV mathematical model involves these two single motions.

Regarding the four DOFs dynamic model in the Girona500 AUV, its hydrodynamic coefficients estimation requires four different parameter estimation runs related to the surge, sway, heave, and yaw motions. Table 3.3 includes the constant values shared in the experiments and computed in the vehicle assembly. Table 3.4 presents the hydrodynamic coefficients acquired from the parameter estimation tool with

their corresponding results. The dynamic coefficients X_u, $X_{\dot{u}}$, $X_{|u|u}$ relate to surge, Y_v, $Y_{\dot{v}}$, $Y_{|v|v}$ to sway, Z_w, $Z_{\dot{w}}$, $Z_{|w|w}$ to heave, and N_r, $N_{\dot{r}}$, $N_{|r|r}$ to yaw motion, incorporating the necessary components in the AUV mathematical model for the 3D environment.

Table 3.3 Principal characteristics of the autonomous offshore system.

USV		*Girona500 AUV*	
Parameter	**Value**	**Parameter**	**Value**
m	3500 [kg]	m	180.00 [kg]
m_{pt}	1100 [kg]	∇	0.1837 $\left[\text{m}^3\right]$
m_{hull}	2400 [kg]	I_z	40.70 $\left[\text{kg m}^2\right]$
L_{USV}	8 [m]	l_{y1}	0.2432 [m]
l_{pivot}	2.40 [m]	l_{y2}	0.2432 [m]
l_{pt}	2.16 [m]		
x	0.70		
c_g	0.30		
I_{cor}	0.6		
I_z from (3.7)	11,284.61 $\left[\text{kg m}^2\right]$		
x_g	0.0425 [m]		

Table 3.4 Dynamic coefficients of the autonomous offshore system using parameter estimation.

	USV	*Girona500 AUV*
Parameter	**Value**	**Value**
X_u	-10.586	21.750
$X_{\dot{u}}$	-3277	-250.184
$X_{\|u\|u}$	315.45	216.423
Y_v	-	6.192
$Y_{\dot{v}}$	-	-580.969
$Y_{\|v\|v}$	-	485.538
Z_w	-	92.657
$Z_{\dot{w}}$	-	-471.215
$Z_{\|w\|w}$	-	189.788
N_r	3907.9	15.560
$N_{\dot{r}}$	-36.555	-44.297
$N_{\|r\|r}$	3459.6	69.364

3.5 Model Validation Using Field Test Data

The estimation of the hydrodynamic coefficients for the USV and AUV mathematical models using field test data includes the vehicle geometry imperfections. These imperfections can be the onboard sensors and might affect the analytical model of the offshore vehicle. However, the model validation of each offshore vehicle requires enough scientific instrumentation. This instrumentation includes the actuators and sensors that could execute basic maneuvers and gather all the necessary field test data in offshore trials. This section shows the model validation using field test data for the USV and the Girona500 AUV because the aColor AUV does not contain enough scientific instrumentation to gather accurate position, velocity, and acceleration data.

3.5.1 Unmanned Surface Vehicle

The USV mathematical model combines the SI and parameter estimation methods for the waterjet and dynamics models, achieving a correct vehicle model validation. Figure 3.9 illustrates the schematic, including all required functions for the comprehensive three DOFs dynamic model. This schematic defines the USV from the joystick input values to the vehicle position and orientation outputs. The waterjet propulsion model subsystems incorporate a 1D lookup table, translating the joystick commands to the waterjet engine rpm, the estimated second-order transfer function for the dynamics response, and a 2D lookup table, including the relation with the thrust force in each waterjet propulsion unit. Moreover, it incorporates another 1D lookup table, translating the joystick commands to the waterjet nozzle position, the estimated first-order transfer function, the calculated thrust force efficiency based on the nozzle position, and the total torque calculation. This three DOFs mathematical model has the thrust force τ_u and torque τ_r as the inputs. The integration of the velocity vector ν produces the USV pose η. The 1D lookup table parameters involving the joystick commands are defined based on the USV field test data, as presented in **Publications III and IV**.

Publications III and IV include the USV mathematical models using the Nomoto's autopilot model and three DOFs dynamic model, both employing field test data for the estimation methods. First, the mathematical model includes the waterjet propulsion system as a second-order transfer function. Second, the propulsion system

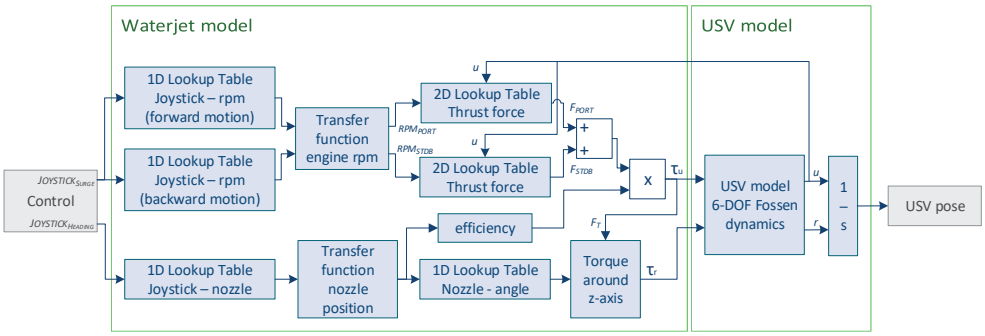


Figure 3.9 Schematic of the USV mathematical model, which incorporates the waterjet propulsion system and three DOFs dynamic models. [**Publication IV**]

model utilizes the data for a single operating point in a propulsion waterjet unit. Figure 3.10 shows the comparison between these two studies, where u_{SI} relates to the Nomoto representation and $u_{\text{3-DOFs}}$ defines the complete mathematical model described in the schematic shown in Figure 3.9. **Publication IV** also shows the three DOFs dynamic model, Nomoto representation, and field tests step response for a fixed Joy_{surge} input value and a zigzag pattern in Joy_{yaw}.

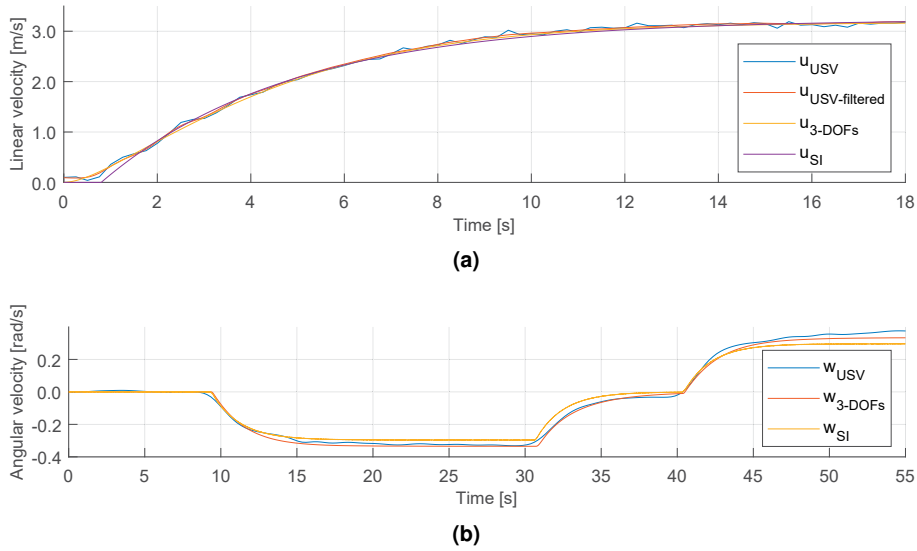


Figure 3.10 Comparison of the USV field test data with the parameter estimation method and SI tool: (**a**) Surge motion. (**b**) Yaw motion. [**Publication IV**]

3.5.2 Autonomous Underwater Vehicle

Publications II and V include the mathematical model for the Girona AUV. The aColor AUV mathematical model was not studied because of the lack of accurate onboard sensors to acquire enough SI or parameter estimation methods data. The first study does not include the estimation of the dynamic coefficients for the AUV mathematical model. Figure 3.11 illustrates the first results for the comparison between the simulation and field test controller variables. The simulation and field test results differ with an overshooting of the controlled position variables. Thus, an accurate AUV mathematical model is a fundamental part of the development of correct GNC algorithms.

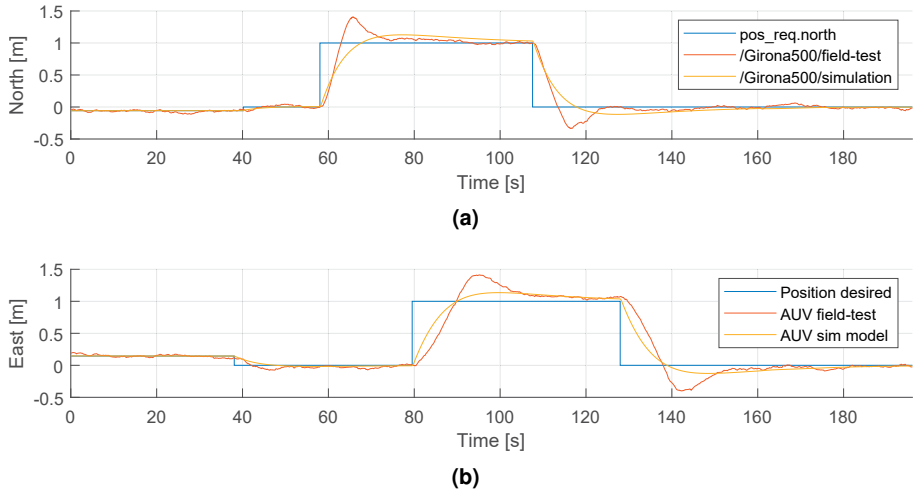


Figure 3.11 Comparison of the values from the GNC algorithm with the field test data from the Girona500 AUV: (a) north, (b) east. [adapted from **Publication II**]

Publication V includes the four DOFs dynamic model for an AUV using the parameter estimation method to improve the results from **Publication II**. The model validation for the Girona500 AUV has been performed based on separate simple motion implementations. Figure 3.12 includes the results for the surge and yaw motions, while **Publication V** presents the rest of the AUV's motions. The four DOFs mathematical model has separate variables for each movement, so each field test trial refers to a specific AUV motion. Thus, it is possible to make a straightforward approach for estimating the dynamic coefficients of the surge, sway, heave, and yaw motions.

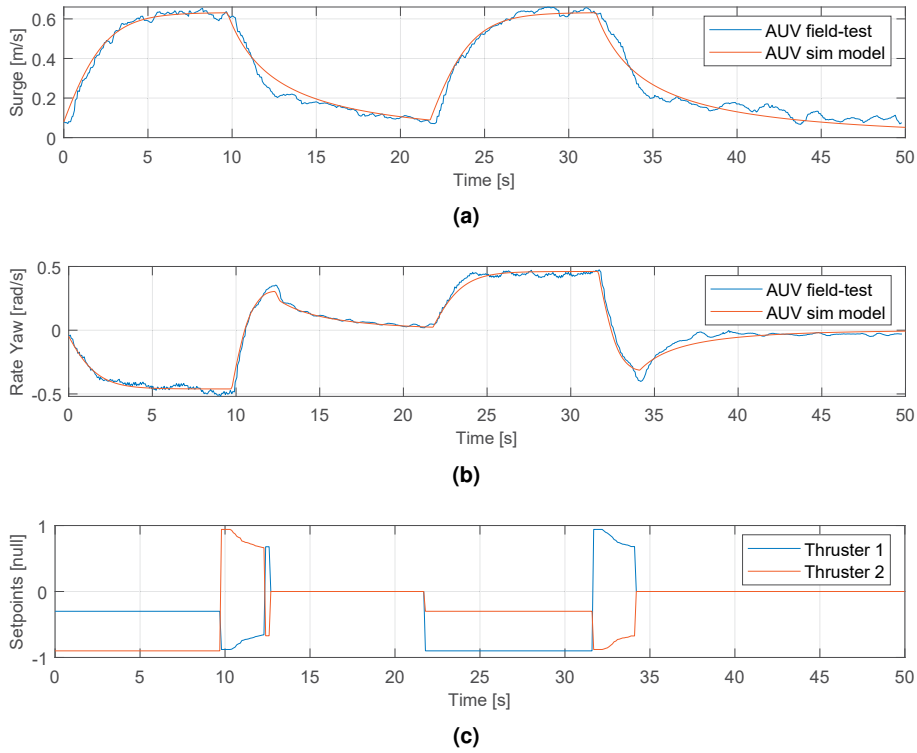


Figure 3.12 Model validation using parameter estimation for the Girona500 in a forward zig-zag motion: (a) Surge, (b) Yaw motion, (c) Thruster setpoints. [**Publication V**]

3.6 Discussion

This chapter has described the mathematical models for the autonomous offshore system and two approaches for estimating their dynamic model parameters. The mathematical models require the closest model to the real-world AOVs as possible in a way that it is possible to develop more advanced GNC algorithms. These methods use the SI and parameter estimation MATLAB-Simulink tools for simplification, providing a straightforward interface for design, modeling, simulation, and implementation for the autonomous offshore system.

The USV mathematical model uses two different approaches based on SI and parameter estimation. The SI method provides an easy and quick solution to generate the transfer functions of the model. However, these transfer functions can only contain a single model and do not include all the vehicle dynamics. Thus, the current thesis used the parameter estimation method to obtain the hydrodynamic coefficients

for the three DOFs dynamic model of the USV. As shown in the experimental results section, this approach provided a higher accuracy than the SI method. The reason was the combination of multiple motions based on nonlinear equations of motion, providing an accurate USV maneuvering compared with the field test results. All these methods used field test data to estimate their coefficients in both transfer function and dynamic matrices.

Concerning the AUV, this chapter has described the parameter estimation for the Girona500 AUV mathematical model. The Girona500 AUV used a similar approach to the USV mathematical model, estimating the hydrodynamic coefficients from field test data. The difference between these mathematical models is that the AUV uses the four DOFs dynamic model, including the heave motion for underwater operation. The mathematical model is very dependent on the configuration of the offshore vehicle, which is mainly related to the installation of scientific instrumentation. Thus, if there are any modifications to the platform, the mathematical model might change, resulting in non accurate simulation results. Nonetheless, these mathematical models considerably help the GNC algorithm's development, especially with the control system. The reason is its ease of providing a first approximation of the value of the controller parameters.

In general, the parameter estimation method was more accurate because it involved more dynamic coefficients in the simplified three and four DOFs mathematical models. However, the SI approach allowed for obtaining a transfer function more simply and quickly in case the mission requires a transfer function for the vehicle in a short period.

4 GUIDANCE, NAVIGATION, AND CONTROL METHODS FOR THE AUTONOMOUS OFFSHORE VEHICLES

This chapter presents the most significant GNC techniques for AOVs regarding path-following algorithms and situational awareness capabilities. Combining the USV and AUV mathematical models with these GNC techniques provides the ultimate solution for the autonomous offshore system's implementation. The contents of this chapter are based on **Publications I—V**, including the results from the simulated and implemented control scenarios.

4.1 Situational Awareness Methods

This section describes the three algorithms related to the situational awareness capabilities of the autonomous offshore system developed in this thesis. These algorithms start with the target detection with multi-sensor technology. Then, an SBB approach is defined based on the target origin position from the previous target detection algorithm. Finally, an additional algorithm for wall detection in an underwater environment is described based on mechanical imaging sonar data.

4.1.1 Target Detection Algorithm

The target detection algorithm, as studied and described in **Publication IV**, includes the application in the autonomous offshore system using LiDAR and mechanical imaging sonar, which was selected as the active ranging sensors for the USV and AUVs, respectively. Their use depends on the target position being over the water's surface or underwater. The underwater target algorithm, which employs the mechanical

imaging sonar data at both AUV platforms, analyzes the acoustic intensity received at each range bin to discover an object's presence. A Tritech Micron sonar [47] is the selected underwater perception sensor for both the aColor and Girona500 AUVs. It has an operating range of a minimum of 0.3 meters and a maximum of 75 meters, with a range resolution of around 7.5 mm. The Micron sonar provides one operating frequency of 700 kHz. **Publication IV** uses the mechanical imaging sonar with a maximum range of 10 meters, a forward-looking 90° field of view, and a 1.8° bin resolution as its predefined settings. It is worth mentioning the possibility to modify these predefined settings with other maximum ranges, field of view, or bin resolutions in a specific application with a higher operating range or narrower targets. Regarding the perception sensor installed on the USV, the SICK MRS1000 LiDAR is chosen for the autonomous operation [51]. This sensor contains simultaneous measurement on four far-reaching levels and a multi-echo technology that bypasses the possible environmental disturbances produced by rain, fog, or dust. Furthermore, because it has an operating range of a minimum of 0.2 meters and a maximum of 64 meters with an aperture angle of 275°, its use is convenient in small USVs.

The target detection algorithm requires the calculation of the target positions after gathering the scan array data. These data from the active ranging sensors use the BODY reference frame, and the data need to be translated toward an absolute coordinate system. This required translation is determined by the following:

$$\begin{bmatrix} x_{\text{obs}} \\ y_{\text{obs}} \end{bmatrix} = \boldsymbol{R}_{\text{z}}(\psi_{\text{AOV}}) \begin{bmatrix} x_{\text{scan}} \\ y_{\text{scan}} \end{bmatrix}, \tag{4.1}$$

where $\boldsymbol{R}_{\text{z}}(\psi_{\text{AOV}})$ is the rotation matrix about the z-axis based on the yaw angle ψ_{AOV} of the AOV. This rotation matrix changes from BODY to ENU coordinate systems. The 2D rotation matrix $\boldsymbol{R}_{\text{z}}(\psi_{\text{AOV}})$ is represented by

$$\boldsymbol{R}_{\text{z}}(\psi_{\text{AOV}}) = \begin{bmatrix} \cos(\psi_{\text{AOV}}) & \sin(\psi_{\text{AOV}}) \\ -\sin(\psi_{\text{AOV}}) & \cos(\psi_{\text{AOV}}) \end{bmatrix}. \tag{4.2}$$

After the perception sensor locates the target in the ENU coordinate system, the

origin position of this target (N_o, E_o) is declared by

$$\begin{bmatrix} N_o \\ E_o \end{bmatrix} = \begin{bmatrix} N_{AOV} \\ E_{AOV} \end{bmatrix} + \boldsymbol{R}_x(\phi_{AOV}) \begin{bmatrix} \frac{x_{obs,init}+x_{obs,end}}{2} \\ \frac{y_{obs,init}+y_{obs,end}}{2} \end{bmatrix}, \tag{4.3}$$

where $\boldsymbol{R}_x(\phi_{AOV})$ is the rotation matrix about the x-axis with roll angle $\phi_{AOV} = pi$ [rad], and positions $(x_{obs,init}, y_{obs,init})$ and $(x_{obs,end}, y_{obs,end})$ define the initial and ending positions of consecutive data bins. Every successive data becomes a single object for the target detection algorithm, allowing for multiple target detections in the same scan. The $\boldsymbol{R}_x(\phi_{AOV})$ matrix translates from the ENU to the NED coordinate system, which is employed in the final offshore navigation, and it is determined by the following:

$$\boldsymbol{R}_x(\phi_{AOV}) = \begin{bmatrix} 1 & 0 \\ 0 & \cos(\phi_{AOV}) \end{bmatrix}. \tag{4.4}$$

In general, the principle for the target detection algorithm is the same for both LiDAR and sonar perception sensors. The LiDAR sensor provides the measurement data as a point cloud in the 3D environment (see Figure 4.1a), which is conveniently transformed into Cartesian coordinates. The LiDAR sensor provides this point cloud data for each beam with a 275° aperture angle from its four scan planes. These scan planes allow for the use of a safety feature for collision avoidance because the target algorithm selects the closest detected point if there is an object on a slope. After gathering the 3D point cloud data, the target detection algorithm translates it into 2D by excluding the z-axis data (see Figure 4.1b). Finally, Figure 4.1c illustrates the target origin position in the NED coordinate system after implementing the target detection algorithm.

The mechanical imaging sonar data comprises the beam heading θ_{scan}, the particular point location in Cartesian coordinates (x_{scan}, y_{scan}), and every bin intensity $\mathscr{I}_{scan}$. After the imaging sonar data acquisition, the target detection algorithm involves the post-processing actions to detect the underwater object, which is defined in **Publication IV**. This algorithm includes the highest intensity value position for each bin in polar coordinates, filtering the data with a minimum working range to avoid any potential noise from the AUV structure. Figure 4.2 illustrates the target detection algorithm implementation for the AUV, starting from the scan data using the BODY reference frame until the final target origin position in the NED coordinate system.

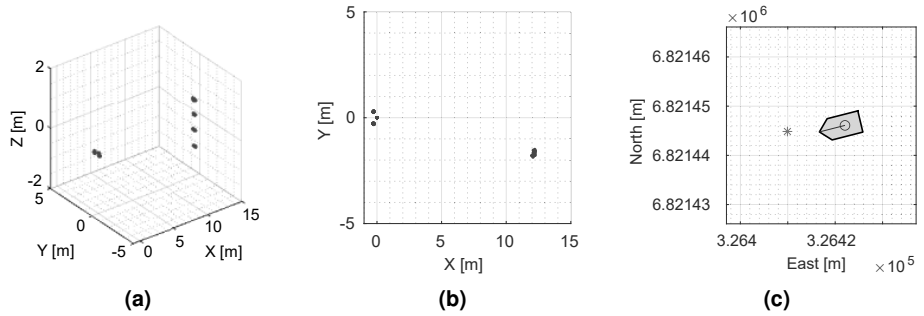


Figure 4.1 Target detection algorithm using the LiDAR active sensor: (a) LiDAR point cloud in 3D. (b) LiDAR point cloud in 2D. (c) Target absolute position in NED. [**Publication IV**]

Figure 4.2a shows the acquired raw data from the mechanical imaging sonar, while Figure 4.2b illustrates the filtering of the close-range values and selection of highest intensity positions. Finally, Figure 4.2c shows the target origin position in the NED coordinate system relative to origin [0,0], while Figure 4.2d illustrates the origin position located in the ETRS-TM35FIN absolute coordinates.

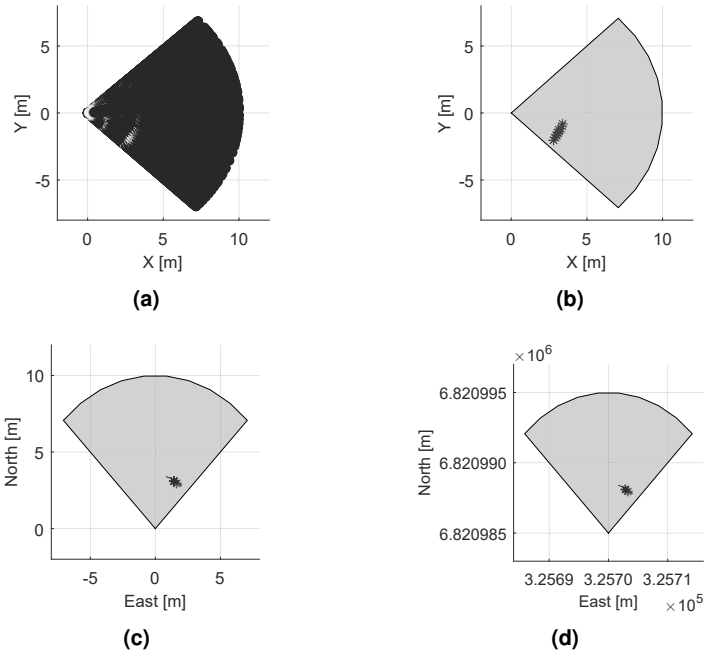


Figure 4.2 Target detection algorithm using the mechanical imaging sonar: (**a**) Data acquisition from sonar. (**b**) Post-processing with data filtering. (**c**) Target origin position with [0,0] origin in NED. (**d**) Target origin position in absolute coordinates. [**Publication IV**]

4.1.2 Obstacle Avoidance Using the Safety Boundary Box Approach

A complete GNC algorithm requires obstacle avoidance capabilities for safety purposes in the autonomous offshore system. After the target detection relative to the AOV, the current thesis proposes an SBB set around this object for obstacle avoidance. The SBB approach is studied and implemented in **Publication III** using LiDAR as the selected active ranging sensor. As mentioned in the previous section, LiDAR produces 3D point cloud data in Cartesian coordinates for each beam, providing a solution for the obstacle localization in absolute coordinates.

Figure 4.3 illustrates the implementation of the SBB approach. The boundary box is a rectangle-shaped area with length L_{box} and width W_{box} dimensions. These dimensions are defined in Equation (4.5) and depend on the obstacle length L_{obs} and width W_{obs}, along with predefined constant parameters for safety distance d_{dist_X} and d_{dist_Y} for the x and y axes, respectively.

$$\begin{cases} L_{\text{box}} = d_{\text{dist}_X} + \frac{L_{\text{obs}}}{2} = d_{\text{dist}_X} + \frac{|x_{\text{obs,init}} - x_{\text{obs,end}}|}{2} \\ W_{\text{box}} = d_{\text{dist}_Y} + \frac{W_{\text{obs}}}{2} = d_{\text{dist}_Y} + \frac{|y_{\text{obs,init}} - y_{\text{obs,end}}|}{2} \end{cases} . \tag{4.5}$$

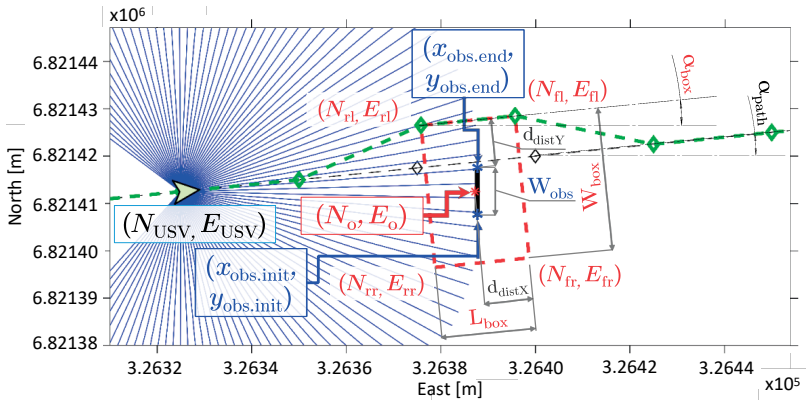


Figure 4.3 SBB approach for obstacle avoidance using LiDAR and the NED coordinate system: SBB (dotted red line) supports the new USV path (dotted green line) [adapted from **Publication III**].

The shape of this boundary box depends on the selected offshore application. In USV operations, the width of the harbor route defines the shape of the box. Furthermore, if the offshore vehicle enters a more restrictive area, the SBB approach decides the box's dimensions based on the vehicle position. The SBB origin (N_o, E_o)

uses the previously mentioned target detection algorithm utilizing the LiDAR data and USV position in absolute coordinates.

The SBB implementation allows a continuous path operation by defining the SBB angle α_{box} that is equal to the inclination of the predefined path α_{path}. Then, the SBB corners determine the waypoints for the LOS-based path-following controller. The closest possible trajectory of the vehicle defines the chosen left or right side waypoints. Rear-left (N_{rl}, E_{rl}) and rear-right (N_{rr}, E_{rr}) are the first possible waypoints, with the correspondent side of the front corner as the following waypoint. Equations (4.6) and (4.7) define the possible SBB corners, where a refers to rear/front, b to left/right, and i and j to the second element sign. $\boldsymbol{R}_{\text{z}}(\alpha_{\text{path}})$ refers to the xy-plane counterclockwise rotation matrix through α_{path}.

$$\begin{bmatrix} N_{ab} \\ E_{ab} \end{bmatrix} = \begin{bmatrix} N_{\text{o}} \\ E_{\text{o}} \end{bmatrix} + \boldsymbol{R}_{\text{z}}(\alpha_{\text{path}}) \begin{bmatrix} i\frac{W_{\text{box}}}{2} \\ j\frac{L_{\text{box}}}{2} \end{bmatrix}, \tag{4.6}$$

$$\begin{bmatrix} N_{ab} \\ E_{ab} \end{bmatrix} = \begin{cases} i = 1 & \text{if } a = \text{f} \\ i = -1 & \text{if } a = \text{r} \\ j = 1 & \text{if } b = \text{l} \\ j = -1 & \text{if } b = \text{r} \end{cases}. \tag{4.7}$$

The LOS-based path-following controller receives the new waypoints from the SBB approach to perform a smooth USV trajectory. The SBB approach is suitable for static and slow-motion objects because it continuously updates the SBB's position for the detected targets. Furthermore, if an obstacle is outside the USV's trajectory, the path-following controller does not receive its waypoints, and the AOV continues its predefined route. After avoiding the obstacle, the AOV continues its path until the vehicle detects another obstacle in its trajectory or reaches its final waypoint.

4.1.3 Wall-Detection Algorithm

Publication II studies a wall-detection algorithm, which defines the waypoints for the LOS-based path-following controller with a constant distance from the wall. The wall detection uses the mechanical imaging sonar point cloud and the split-and-merge algorithm, as shown in Figure 4.4. After identifying all the water tank walls, the

wall-detection algorithm selects one to perform the LOS guidance implementation. Finally, the line expression of the chosen side defines the two waypoints for the LOS-based path-following controller.

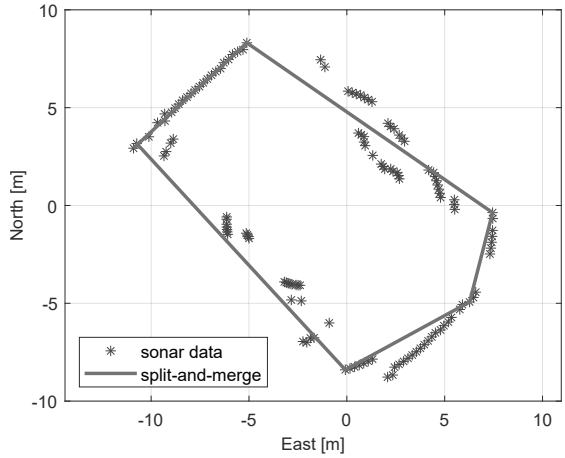


Figure 4.4 Wall-detection algorithm utilizing the mechanical imaging sonar and the split-and-merge line extraction. [**Publication II**]

4.2 Guidance System and Control Algorithms

The guidance system works for the path-following algorithm to reach each waypoint in a predefined path. This path following is assumed as a guidance system in this thesis because it is closer to practical engineering and is easier to perform than trajectory tracking. Mainly, the LOS-based guidance system forms the path-following algorithm of the autonomous offshore system. However, the present thesis includes other guidance systems because the Girona500 AUV includes a waypoint controller with position and velocity PID controllers.

4.2.1 Simple Position and Velocity Control

Publication II studies and implements diverse PID controllers as a guidance system in the Girona500 AUV. These controllers include a velocity control using a unique PID controller, a position control based on a single PID controller, and position control using a cascade controller with position and velocity PID controllers. The

mathematical model in MATLAB-Simulink provides an estimation for the PID parameters of all these controllers; this estimation uses a specific time response and transient behavior as the main controller specifications. After designing the PID controllers, the default COLA2 navigation module [37] can provide the AUV pose and velocity in the NED local coordinate system, collecting and merging the necessary data from the AUV navigation sensors.

The first case involves the position and velocity of unique PID controllers. They obtain the current vehicle pose (or velocity) from the COLA2 navigation module and desired pose (or velocity) from the GNC algorithm. Then, the controller computes the necessary force and torque τ to achieve the desired pose (or velocity) in each motion. For the position control using the cascade controller (position and velocity PID controllers), the position controller $C_{\text{pos}}(s)$ receives the current vehicle pose η from the navigation module and the desired pose request η_{ref} from the guidance system. The position controller output is the required velocities to achieve the requested AUV position. Then, the velocity controller $C_{\text{vel}}(s)$ includes a PID with an open-loop, model-based controller. The desired velocity is obtained from the position controller, while the COLA2 navigation module provides the current vehicle velocity. Finally, the velocity controller computes the necessary force and torque in each motion τ to achieve the desired velocity. After calculating the necessary force and torque τ in each motion, the necessary setpoints for each thruster $\boldsymbol{f}$ are calculated based on the received forces and torques using the pseudoinverse $\boldsymbol{T}^{+}$ of the thrust configuration matrix $\boldsymbol{T}$ [25], which is determined by the following:

$$\boldsymbol{f} = \boldsymbol{T}^{+}\tau. \tag{4.8}$$

4.2.2 Line-of-Sight Guidance

The current thesis uses the LOS-based guidance system that is used in most publications for controlling the autonomous offshore system [22, 25]. Its simplicity and convenient use for underactuated offshore vehicles are the main advantages of this guidance system. The heading control can steer the AOV by aiming towards the following waypoint in the path [25]. Figure 4.5 illustrates the LOS vector for path-following control in a USV. This LOS-based controller computes the course angle ψ_{d} utilizing the path-tangential angle χ_{p} and velocity-path relative angle χ_{r}. The

look-ahead-based steering implementation applies the transformation determined by

$$\psi_{\mathrm{d}} = \chi_{\mathrm{p}} + \chi_{\mathrm{r}} - \beta, \tag{4.9}$$

where β is the sideslip (drift) angle [25]. The current thesis does not include this variable in the AOV implementation to simplify the steering law. The velocity path relative angle χ_{r} establishes that the velocity direction faces a path location in a look-ahead distance $\Delta(t) > 0$ along the direct projection [57]. The velocity path relative angle χ_{r} and the path-tangential angle χ_{p} are declared as

$$\chi_{\mathrm{r}}(e) = \arctan(-K_{\mathrm{P}} e - K_{\mathrm{I}} \int_0^t e(\tau) d\tau), \tag{4.10}$$

$$\chi_{\mathrm{p}} = \mathrm{atan2}(E_{k+1} - E_k, N_{k+1} - N_k), \tag{4.11}$$

where $K_{\mathrm{P}} = 1/\Delta(t) > 0$ represents the proportional gain, $K_{\mathrm{I}} > 0$ defines the integral gain, and (N_k, E_k) and (N_{k+1}, E_{k+1}) are the positions of the passed and next waypoint, respectively. The cross-track error $e(t)$ is defined by the following:

$$e(t) = -[N_{\mathrm{AUV}}(t) - N_k]\sin(\chi_{\mathrm{p}}) + [E_{\mathrm{AUV}}(t) - E_k]\cos(\chi_{\mathrm{p}}). \tag{4.12}$$

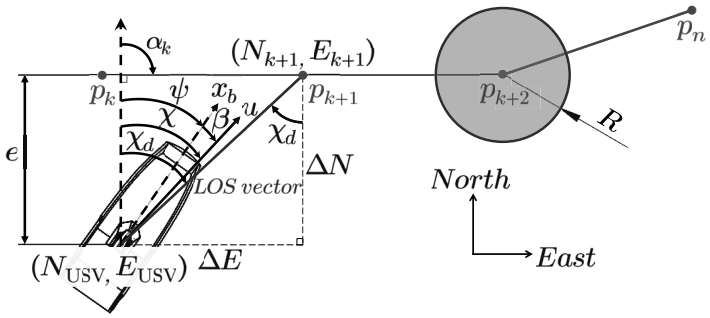


Figure 4.5 LOS guidance system and circle of acceptance in a USV using the NED coordinate system [adapted from **Publication III**].

The guidance system requires a switching mechanism to change the look-ahead waypoint in the predefined path. The switching mechanism incorporates a circle of acceptance for the USV [25] and a sphere of acceptance for the AUVs [33]. The circle of acceptance selects the next waypoint as a look-ahead point if the position of the

USV lies within a circle with radius R_{USV} around (N_{k+1}, E_{k+1}). Similarly, the sphere of acceptance selects the next waypoint as a look-ahead point if the AUV position lies within a sphere with a radius R_{AUV} around the location $(N_{k+1}, E_{k+1}, D_{k+1})$. Figure 4.5 illustrates the circle of acceptance in a USV determined by

$$[N_{\text{USV}}(t) - N_{k+1}]^2 + [E_{\text{USV}}(t) - E_{k+1}]^2 \leq R_{\text{USV}}^2, \tag{4.13}$$

where, if the time surface vehicle position $(N_{\text{USV}}(t), E_{\text{USV}}(t))$ satisfies Equation (4.13), the next waypoint (N_{k+1}, E_{k+1}) needs to be selected. Radius R_{USV} is equal to two USV lengths L_{USV} ($R_{\text{USV}} = 2L_{\text{USV}}$). Similarly, Figure 4.6 shows the sphere of acceptance switching mechanism, which is

$$[N_{\text{AUV}}(t) - N_{k+1}]^2 + [E_{\text{AUV}}(t) - E_{k+1}]^2 + [D_{\text{AUV}}(t) - D_{k+1}]^2 \leq R_{\text{AUV}}^2, \tag{4.14}$$

where, if the time AUV position $(N_{\text{AUV}}(t), E_{\text{AUV}}(t), D_{\text{AUV}}(t))$ satisfies Equation (4.14), the next waypoint $(N_{k+1}, E_{k+1}, D_{k+1})$ needs to be selected.

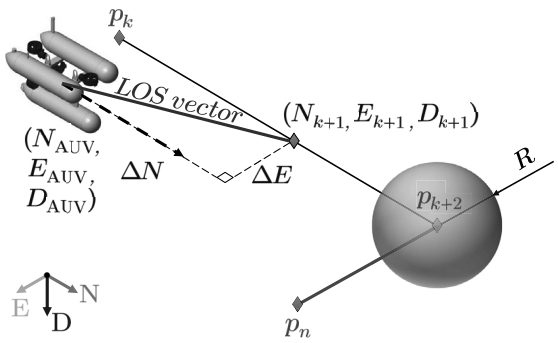


Figure 4.6 LOS guidance system and sphere of acceptance in an AUV using the NED coordinate system.

4.2.3 Directional and Attitude Control

As explained above, the USV uses a LOS-based, path-following controller by representing the desired path with waypoints (N_k, E_k). This controller sends the necessary heading commands to the yaw controller, aiming towards the following waypoint in the path. The directional control in the USV contains two separate PID controllers for the surge and yaw motions, which obtain their parameters based on rapid control prototyping during field tests. Additionally, the feedback loop incorporates low-pass

and notch filters to reduce the motions induced by waves [25]. A first-order, low-pass filter $h_{\mathrm{lp}}(s)$ with a time constant T_{f} is designed as follows:

$$h_{\mathrm{lp}}(s) = \frac{1}{1 + T_{\mathrm{f}} s}. \tag{4.15}$$

The controller bandwidth ω_b can be close to or within the range $\omega_{\mathrm{min}} < \omega_{\mathrm{e}} < \omega_{\mathrm{max}}$ of the wave spectrum for small USV platforms. The addition of a low-pass filter in cascade with a notch filter resolves this uncertainty, even though the notch frequency ω_{n} estimation might not be accurate. Therefore, the feedback loop incorporates a filter structure $h_n(s)$ formed by three cascaded notch filters with fixed center frequencies ω_i [28], which can be defined as follows:

$$h_n(s) = \prod_{i=1}^{3} \frac{s^2 + 2\xi\,\omega_i s + \omega_i^2}{(s + \omega_i)^2}, \tag{4.16}$$

where ξ is a design parameter for controlling the notch's magnitude.

Publication I studies the performance of this cascade control with low-pass and notch filters and the ROS implementation in the marginally stable system. The first-order, low-pass filter suppresses forces over the frequency $1/T_{\mathrm{f}}$. This time constant is difficult to specify in a USV, but $T_{\mathrm{f}} = 0.1$ s is a good estimation based on simulation and field tests. Similarly, the directional control assumes the design parameter $\xi = 0.8$, along with $\omega_1 = 0.1$ rad/s, $\omega_2 = 0.2$ rad/s, and $\omega_3 = 0.4$ rad/s as the center frequencies of the notch filters, here because of the lack of big waves in the field tests at Pyhäjärvi Lake (Tampere, Finland). Thus, steady outputs in the directional control help the drift force compensation from environmental elements, such as waves, wind, and ocean currents.

The PID parameters for the USV platform are shown in Table 4.1. The surge controller is in charge of the forward and backward movement of the USV. The USV mathematical model supports estimating the PID parameters by the time response and transient behavior parameters definition in the MATLAB-Simulink Control toolbox [75]. The yaw controller includes an integral part of the controller to avoid overshooting in the USV motion. Furthermore, the yaw controller includes only one-fifth of the estimated parameter controller from the control toolbox because of its better performance. Figure 4.7 illustrates a comparison from the USV's mathematical model output with a constant reference surge speed. The mathematical model output

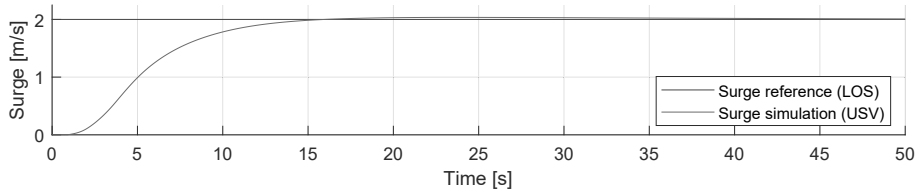


Figure 4.7 Comparison of surge speed from the USV mathematical model with a constant surge reference value.

is useful for estimating the PID parameters of the motion controllers because these parameter estimations allow for a steady response with a minimum overshoot of the offshore vehicle motions.

Table 4.1 Final PID controller parameters for the USV.

Controller	K_P	K_I	K_D
Surge	506.598	83.2	2.531
Yaw	5.236	0.071	15.084
LOS	0.1	0.001	0.0

Regarding the AUV platform, two different approaches are implemented depending on which AUV is being used. The aColor AUV incorporates surge, yaw, and heave motion controllers, while the Girona500 AUV uses several controllers, here depending on the control scenario. In the aColor AUV, once the LOS-based, path-following algorithm calculates the course angle, the yaw controller receives the necessary heading commands to reach the aimed path. Apart from the yaw controller, the heave controller maintains the AUV at a fixed depth, simplifying the AUV operation to a 2D environment. The rapid control prototyping using the Ziegler-Nichols PID tuning [90] throughout field tests determines the PID parameters for the heading controller. The tuning method calculates the amplitude K_{zn} and period T_{zn} for the AUV at a small water tank. Table 4.2 shows the PID parameters obtained from the Ziegler-Nichols tuning method. Additionally, the heave controller includes a simple proportional controller, and the surge motion has a constant PWM value as an input for the thrusters.

The Girona500 AUV incorporates the control in all translational motions (surge, sway, and heave) and in the yaw rotational motion, including a separate PID controller for each position and orientation of the NED coordinate system (north, east, down,

Table 4.2 Final PID controller parameters for the aColor AUV. [**Publication IV**]

Controller	T_{zn} [s]	K_{zn}	K_P	K_I	K_D
Yaw	2.10	5.80	0.580	0.276	0.812
Heave	-	-	300	0.0	0.0
LOS	-	-	0.333	0.0	0.0

and yaw). **Publication V** studies and implements this control approach. Regarding the heading control, it uses a LOS vector from the AUV position to the next waypoint of the path, which is similar to the USV's implementation. This AUV focuses on a 2D covering algorithm, so the heave motion remains steady by keeping a constant depth in the path-following algorithm. Table 4.3 shows the PID parameters obtained from a combination of simulation results and rapid control prototyping.

Table 4.3 Final PID controller parameters for the Girona500 AUV.

Controller	K_P	K_I	K_D
North	151.366	0.0106	1081.310
East	151.366	0.0106	1081.310
Down	151.366	0.0106	1081.310
Yaw	63.425	0.0037	271.85
LOS	0.0333	0.0	0.0

4.3 Experimental Validation

After studying and describing the GNC techniques involving situational awareness, guidance system, and control algorithms, correct autonomous offshore system operations require experimental validation in simulation and field test environments. This section includes the system implementation for all autonomous offshore vehicles used in the current thesis. Furthermore, this section shows the experimental results from several offshore control scenarios with the path-following control and target detection with obstacle avoidance as the primary use cases.

4.3.1 System Implementation

The GNC algorithms require specific mechatronics systems to implement obstacle avoidance, path following, and control for the autonomous offshore system. **Publication III** describes the system architecture as high-level control with the ROS computers, hence being able to perform intricate computations, and low-level control with the waterjet control units and sensors, hence forming the interface for basic vehicle operations. Additionally, the display computers constitute the intermediate-level control, linking the low-level and high-level systems to perform data acquisition and basic logic operations. Figure 4.8 illustrates all the scientific instrumentation included in the USV, which uses ROS as the middleware. ROS contributes the necessary tools for data acquisition and processing. Then, it is possible to generate the appropriate actuator responses to achieve autonomous operations.

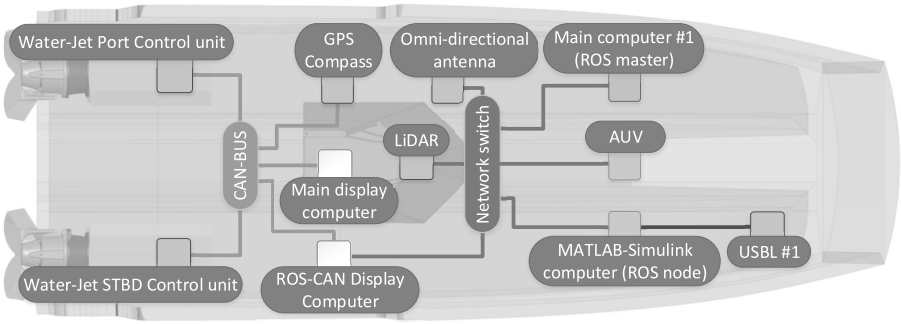


Figure 4.8 System implementation for the USV with ROS computers as high level, display computers as intermediate level, and low level control, including waterjet control units, GPS compass, and LiDAR. [**Publication IV**]

The high-level control includes a Linux computer acting as the ROS master and using a network switch to connect with the rest of the instrumentation via Ethernet. This Linux computer sends and receives the required commands for the USV operation. Then, another computer includes MATLAB-Simulink and acts as a ROS node during testing; it incorporates a standalone ROS node permitting a rapid control prototyping procedure while testing and providing a solution for the C++ programming because it skips numerous programming steps to fulfill the designed GNC algorithm [64]. The intermediate-level control incorporates two CCPilot VC display computers [12]. These computers are suitable for marine environments because of their IP66 class and are freely programmable. They contain a controller area network

(CAN) bus interface and Ethernet port, allowing for a connection to the low-level and high-level controls. The ROS-CAN display computer translates between CAN bus messages from the low-level control and the ROS messages from the high-level control. It uses rosserial for the connectivity within the USV platform, allowing for embedded systems to use ROS [8]. This display computer acquires the CAN bus messages and generates the required ROS messages for the high-level control, which contains the USV pose and velocity without requiring supplementary converters. The main display computer communicates with the ROS-CAN display computer by receiving the joystick commands generated from the high-level ROS computers. The main display computer also connects the two waterjet control units with the rest of the system by sending joystick commands to the waterjet control units, here by following the predefined priority levels. The three-axis joystick and steering wheel form the manual control and provide the safety feature for the USV's operation. The low-level control includes the scientific instrumentation for the USV's localization with the GPS compass and situational awareness capabilities with LiDAR as the primary active ranging sensor, providing collision avoidance capabilities besides 3D map construction of the surroundings. Furthermore, the waterjet control units define the actuators for the twin waterjet propulsion system of the USV platform.

Figure 4.9 illustrates the different scientific instrumentation and actuators installed in the aColor AUV. The connection between this AUV and the USV is performed via a neutrally buoyant tether, here with a straight Ethernet connection between the platforms, as shown in Figure 4.8. The AUV system's architecture is similar to the USV, including a high-level control with the ROS master computer and an intermediate-level control as a bridge between the main ROS computer located in the USV and companion computer installed in the AUV. The low-level control involves the actuators with six thrusters and their respective ESCs and scientific instrumentation. The AUV onboard sensors incorporate a mechanical imaging sonar as the underwater active ranging sensor and pressure sensor with a USBL acoustic system for positioning. This USBL system also allows for communication between the AUV and USV. Finally, the AUV includes a network switch, enabling the connection between the flight controller, the companion computer, and the ROS node computer. The ROS computer performs complex computations for the implementation of the GNC algorithms.

Finally, the system architecture at the Girona500 AUV also incorporates multi-

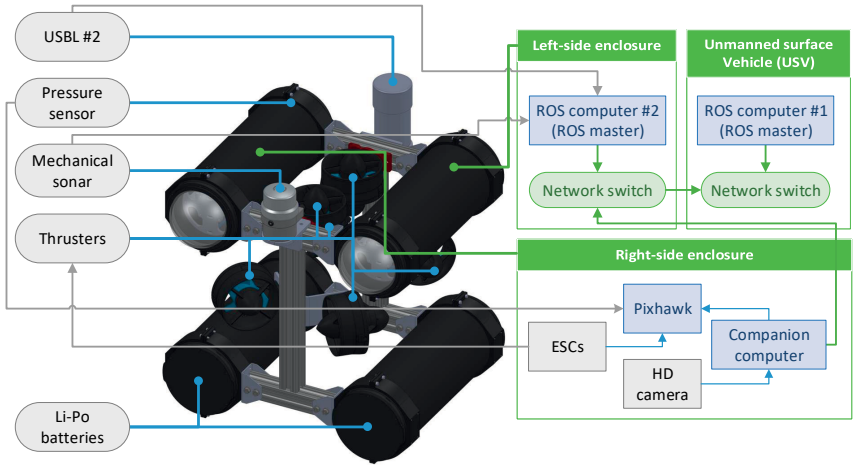


Figure 4.9 System implementation for the aColor AUV with ROS computer as high-level control, Pixhawk flight controller and companion computer as intermediate-level control, and low-level control including thrusters and installed scientific instrumentation. [**Publication IV**]

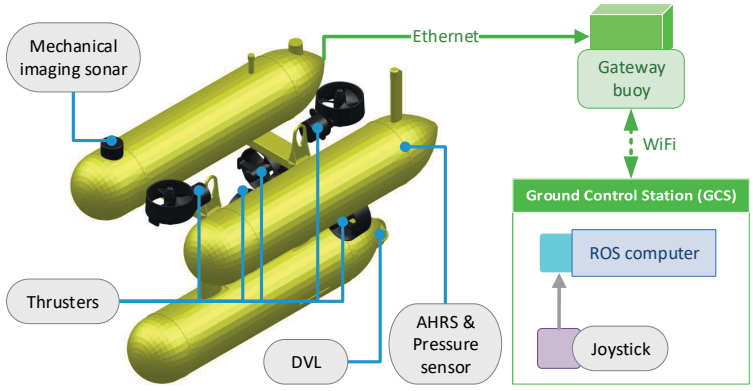


Figure 4.10 System implementation for the Girona500 AUV with ROS computer as high-level and low-level controls including onboard sensors and actuators. [adapted from **Publication V**]

ple mechatronic systems to control the surge, sway, heave, and yaw motions and successfully perform the path-following tasks. Figure 4.10 illustrates all installed mechatronic systems in the Girona500 AUV. This vehicle uses the same modular system architecture as the USV and aColor AUV. The high-level control involves the ROS computers, while the low-level control involves the DVL, AHRS, and depth sensors and actuators. An external Linux computer, acting as the ROS node and located onshore, runs the GNC algorithms. A gateway buoy connects the external computer to the AUV via WiFi and Ethernet. The system communication includes

the connection between the external computer and a gateway buoy via WiFi, and this buoy connects to the Girona500 AUV via Ethernet. The implemented GNC algorithms use the same standalone ROS node procedure discussed in the previous platforms. The ROS master runs the COLA2 navigation system to gather the necessary navigation and data of the surroundings from the AUV. Then, it enables the autonomous task performance by sending the required thruster setpoints commands from the GNC algorithm.

4.3.2 Experimental Results

The USV experimental results were all implemented for a field test at Pyhäjärvi Lake (Tampere, Finland) after testing the designed GNC algorithms in the simulation environment. **Publication I** studies and implements the first approach for the path-following algorithm with directional control in the USV platform. This first approach shows the correct implementation of the proposed modular GNC architecture based on the standalone ROS node generated from MATLAB-Simulink. The control scenario for this first approach includes a simple LOS-based, path-following algorithm without any obstacle avoidance capabilities. Figure 4.11b illustrates the USV trajectory for this first control scenario, including five waypoints to be followed in a straight-line path. Figure 4.12 illustrates the corresponding LOS cross-track error $e(t)$ for this straight line path-following algorithm with a random initial USV position. Figure 4.11a shows the USV trajectory for this first approach on the Tampere map. **Publication III** describes the map processing from an RGB to the black and white map, both including the ETRS-TM35FIN planar coordinate system. This map processing removes every noisy point with standard image processing software. Then, the black and white map is transformed into a binary occupancy grid in MATLAB for further use of the GNC algorithm's development in the simulation environment.

Publication III includes the second approach for the directional control in the USV. This approach focuses on the simulation and implementation of a path-following algorithm with obstacle avoidance of a USV in harbor conditions. It performs the obstacle avoidance of a static buoy located in the middle of Pyhäjärvi Lake (Tampere, Finland), as shown in Figure 4.13.

Figure 4.14 shows the satisfactory implementation for the path following with obstacle avoidance capabilities based on the SBB approach, illustrating the SBB created

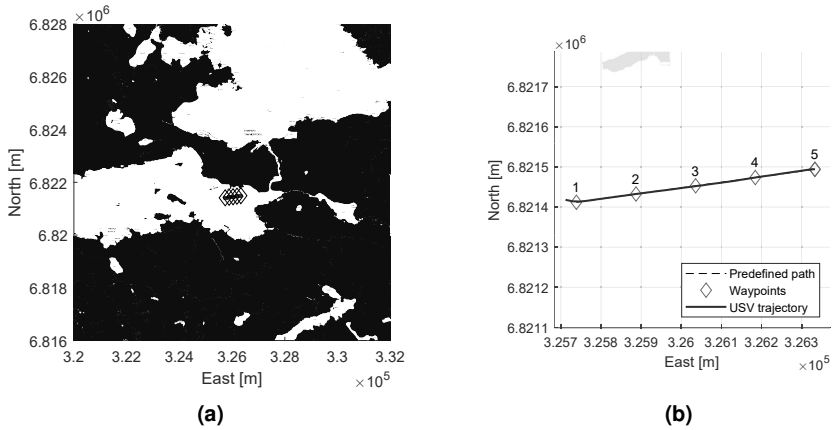


Figure 4.11 USV path-following implementation at Pyhäjärvi Lake (Tampere, Finland): (a) Map view [**Publication I**]. (b) Zoom view. Each waypoint is marked with its order number.

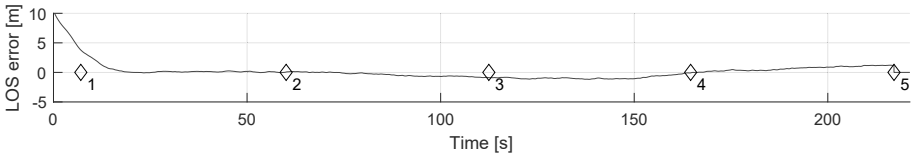


Figure 4.12 USV cross-track error at the LOS-based, path-following algorithm in a straight line path. Each waypoint is marked with its order number.

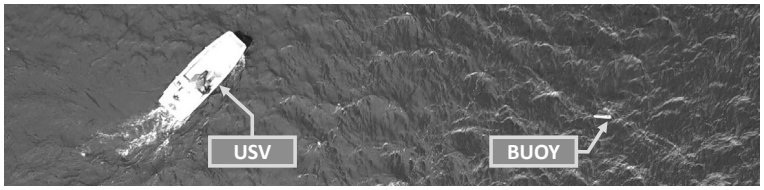


Figure 4.13 USV during the implementation of the path following with obstacle avoidance algorithm at Pyhäjärvi Lake (Tampere, Finland). [**Publication III**]

around the static obstacle (dashed red line), as well as the predefined and GNC generated paths (black and green dashed lines, respectively). This control scenario includes a predefined route with a certain number of waypoints with a random initial USV position. The USV tries to reach each waypoint using the directional control with the LOS-based, path-following algorithm. After detecting the blocking obstacle with LiDAR, the GNC algorithm generates the SBB and modifies the waypoints accordingly. Additionally, Figure 4.15 illustrates the corresponding LOS cross-track error $e(t)$ reducing the LOS error to the minimum for each waypoint, confirming

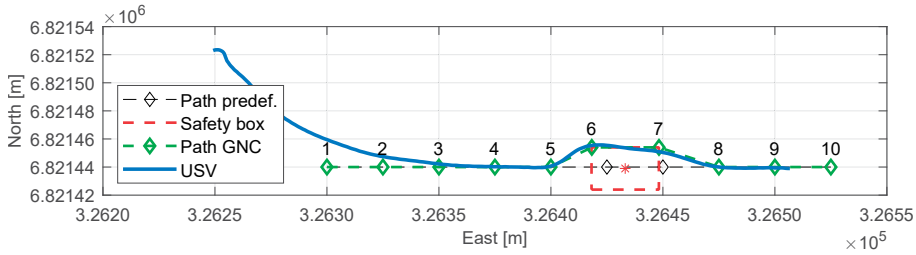


Figure 4.14 USV LOS-based, path-following with obstacle avoidance of a static obstacle using the left corners of the SBB (dotted red line) for the generated path (green line). Each waypoint is marked with its order number. [**Publication III**]

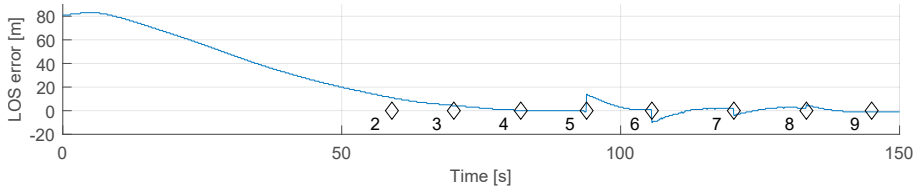


Figure 4.15 USV cross-track error at the LOS-based, path-following algorithm for the look-ahead-based steering law. Each waypoint is marked with its order number. [**Publication III**]

the correct performance of the proposed algorithm. The directional control includes a constant surge speed for the whole control scenario, while the USV heading is modified to reach each of the waypoints in the path.

Regarding the Girona500 AUV platform, **Publication II** studies and implements the position and velocity controllers in path following with a wall-detection algorithm. These control scenarios use the Girona500 AUV because of the availability of the cutting-edge sensors and instrumentation installed in the vehicle. Figure 4.16 shows the implementation of these control scenarios in a water tank. The position and velocity controllers implement the attitude control of the Girona500 AUV. The main purpose of this control scenario is to reach the two waypoints generated from the wall-detection algorithm. Figure 4.17 illustrates the AUV trajectory in 2D and 3D environments, showing the correct implementation of the attitude control. This control scenario includes the yaw motion with higher priorities than the other motions. Thus, the AUV first rotates parallel to the detected wall, and then, the AUV moves to the defined waypoints. Additionally, **Publication II** includes the plots for the surge, sway, heave, and yaw motions, comparing the simulation results from the Girona500 AUV mathematical model with the field test data.

Publication V describes the approach, including several tests in a sea environment

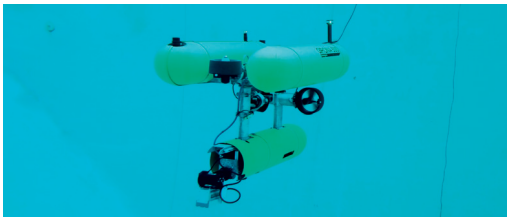
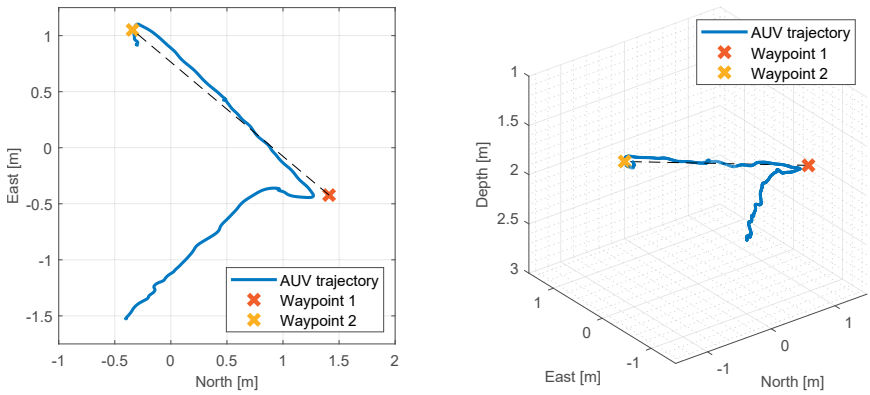


Figure 4.16 Girona500 AUV during the implementation of the wall-detection algorithm with waypoint following in the water tank at the Universitat de Girona (Girona, Spain). [**Publication II**]



(a) **(b)**

Figure 4.17 AUV trajectory for the path-following for the two waypoints generated from the wall-detection algorithm: (a) 2D trajectory (b) 3D trajectory. [**Publication II**]

at the harbor of Sant Feliu de Guíxols (Girona, Spain), as shown in Figure 4.18. These field tests include the complete path-following algorithm implementation of a planar coverage area. A predefined position of the corners of a rectangle underwater defines this coverage area, hence determining the depth for the 2D exploration. Figure 4.19 shows the successful implementation of the path-following algorithm, here the solid red line corresponds with the coverage path and the solid blue line with the actual AUV trajectory. This approach including a separate PID controller for each position and orientation of the NED coordinate system (north, east, down, and yaw) and improves the performance from the previous one because it includes the LOS guidance system with attitude control, allowing a continuous path operation. Figure 4.20 illustrates the corresponding LOS cross-track error $e(t)$ minimizing the LOS error for each waypoint. Based on this cross-track error $e(t)$, three non-dimensional indicators measure the performance for the designed GNC architecture: root-mean-square

error (RMSE), standard deviation (SD), and mean absolute error (MAE) [85]. Table 4.4 shows the three non-dimensional indicators for this path-following algorithm implementation of a planar coverage area. Additionally, **Publication V** incorporates the plots and AUV tracking position error for the controlled motions, comparing the field test results with the mathematical model of the Girona500 AUV.

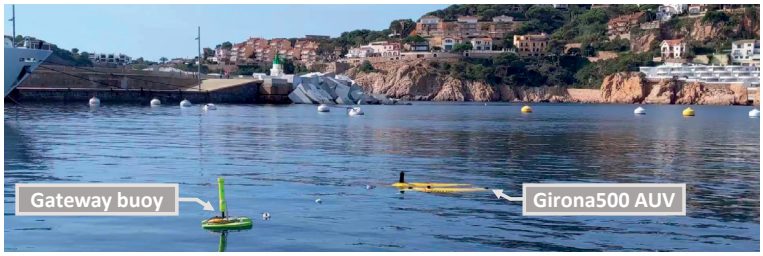


Figure 4.18 Girona500 AUV during the implementation of the path following algorithm at the harbor of Sant Feliu de Guíxols (Girona, Spain). [**Publication V**]

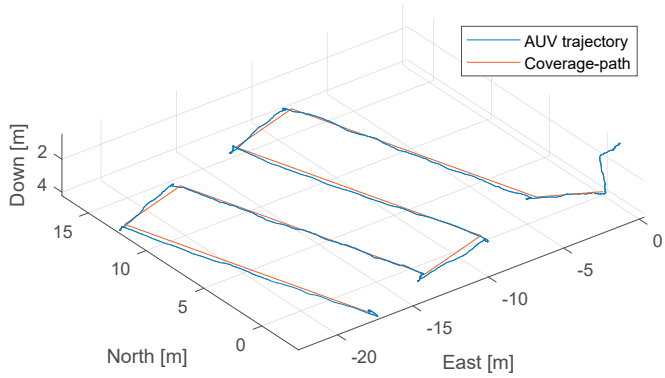


Figure 4.19 AUV tracking trajectory for the sea trials in NED coordinate system. [**Publication V**]

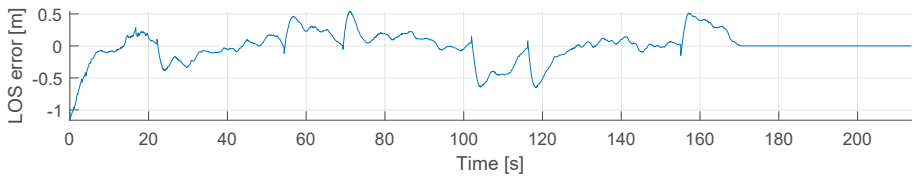


Figure 4.20 AUV cross-track error at the LOS-based, path-following algorithm for the sea trials.

Table 4.4 Comparison of non-dimensional indicators for the AUV tracking trajectory.

Vehicle	RMSE	SD	MAE
Girona500 AUV	0.2346	0.2345	0.1525

4.4 Discussion

This chapter has described the GNC architectures for AOVs involving situational awareness methods and guidance systems and control algorithms as the main components for the autonomous operation. The development of appropriate GNC algorithms allows for the correct performance in the autonomous offshore system. Furthermore, the experimental validation in this chapter has shown the proper implementation of the designed algorithms. Relevant research works in the scientific community studied the obstacle avoidance problem but mainly in 2D. Furthermore, several research works included the advanced guidance capabilities for path following for different planar guidance laws. The current thesis has used a self-controller function or "human on the loop" as level of autonomy, including the obstacle avoidance of static or slow-motion objects and attitude and path-following control in diverse control scenarios. This attitude control system includes a heading autopilot that has the main functionality of keeping the AOV in the desired yaw angle for a predefined path.

The USV control scenarios started with a simple LOS-based, path-following algorithm. The current work tested the control architecture implementation, where the standalone ROS node generated by MATLAB-Simulink and the modular architecture in the USV platform were under consideration. This basic implementation helped in understanding how the USV platform behaved in real-world conditions and to set the grounds for further development in the autonomous system. Then, the second approach included the obstacle avoidance of a static object based on the SBB approach. The SBB approach is a simple way to add the ability to effectively and reliably plan an optimal path, allowing for the implementation of the vehicle in a protocol-free application. The performance of these GNC algorithms fulfilled all requirements for the autonomous mission.

However, there is some needed improvements to raise the system into the next level. COLREGs are mandatory if the USV needs to operate in populated area waterways.

Thus, it would be necessary to research the inclusion of the SBB approach with COLREGs. The attitude control does not involve the compensation of environmental elements that considerably affect the USV's performance, such as wind or wave drift forces. All the control scenarios shown in the current thesis had good weather conditions. Thus, the compensation of these drift forces with the control system would remarkably increase the USV's capabilities.

The Girona500 AUV control scenarios involve two implementations, one in a water tank and another in an open sea environment. The first implementation included the wall-detection algorithm development with a waypoint following. This control scenario helped to understand how a fully functional AUV works underwater while testing the modular architecture approach for an AUV instead of USV. Despite the low number of waypoints to be followed, the control scenario was appropriate for gathering and processing data from a mechanical imaging sonar while verifying the attitude control in an AUV. The second control scenario tested the AUV's performance in a complete path-following mission based on the LOS-based guidance system. This second scenario involved a more advanced guidance control because it added heave and sway to the surge and yaw motions in the USV platform. It is noteworthy that there were differences in the total test time and steady-state behavior between the mathematical model and field test data, provoked by the reduced-order model. The non-inclusion of Coriolis and cross-terms of the dynamic matrices also affected the mathematical model.

All GNC algorithms had the same procedures from the design and testing in a simulation environment to the actual implementation in a real-world scenario. The modular approach in the GNC architecture allowed for the continuous testing of each algorithm. Furthermore, the standalone ROS node generated by MATLAB-Simulink provided a fast solution for C++ programming because it could skip numerous programming steps and fulfill the required programming standards. However, the autogenerated C++ code produced some uncertainties in the system when including several controllers simultaneously, which is what occurred with the Girona500 AUV in the second scenario. Thus, the GNC algorithm's simplification and the ROS topic frequency were mandatory to avoid system collapse.

5 GUIDANCE, NAVIGATION, AND CONTROL ARCHITECTURE FOR THE CO-OPERATIVE SYSTEM

This chapter presents the GNC architecture for the co-operative system that is formed by a USV and AUV, including the path-following and situational awareness capabilities in the same framework. The co-operative system evaluates the total situational awareness underwater and on the water surface, here based on the AUV's and USV's onboard sensors. The results and discussion within this chapter are based on **Publication IV**.

5.1 Multi-Vehicle Software Architecture

Multiple systems simultaneously in co-operation create several advantages for perception systems compared with individual vehicle implementation because more useful information may become available. However, the single platform's data need to be fused together to utilize the full benefit of these advantages.

The classification of multi-vehicle systems includes decentralized systems, with each AOV running an independent ROS master or centralized ones with the master node located at the ground control station. It is a well-known fact that most ROS applications use a centralized ROS master. The reason for this is the single robot application or that the system needs a common point for data processing. However, in a multi-vehicle system set-up, the naming of the diverse nodes, topics, and parameters involved in the ROS application becomes more complex with the possibility of high computing costs, duplicities, significant communication demands, processes delays, and other issues related to the system handling by an overloaded single ROS master. On the contrary, decentralized systems are more effective and usually de-

crease the communication network requirements compared with centralized systems, even though they are more complex because of contingencies and communication limitations, such as delays, noises, or simple failures. Hence, a multi-master approach can provide answers because each platform runs its ROS master and exchanges only the required data between them.

Different software architectures utilize the communication within various ROS masters, which commonly include one for the ground control station and the other for each vehicle in the system. Another relevant characteristic of this architecture is modularity. Hence, the multi-master schema helps in accomplishing the mission and decreases the required bandwidth for communications. Additionally, using a multi-master schema improves the communication between the base station and vehicles. The housekeeping data needed to control the high number of nodes and topics may introduce delays or data loss in relevant data (e.g., target detection, vehicle positions, or mission state) when managed by a single master.

There are a few challenges in co-operative systems, such as task allocation and coordination, communications, information exchanged, or time synchronization. USVs usually co-operate with other autonomous vehicles, such as AUVs and UAVs, to accomplish more effective offshore missions. However, GNC methods can be relatively complex because it becomes crucial to fuse together the data gathered from the individual vehicles.

5.1.1 Communication between the Offshore Vehicles

Communication becomes a relevant issue when working with co-operative systems because there is always a strong communication need between the robots and the ground control station in centralized or decentralized solutions. Most multi-vehicle systems use wireless communications to avoid constraints on vehicle movements. Thus, communication robustness becomes a vital part of the co-operative system when evaluating the overall system performance relating to bandwidth, range, and latency. A point-to-point scheme is the simplest way of communication because the co-operative system agents send information directly to each receiver. Nevertheless, this approach requires a reduced number of components in the system for suitable implementation.

WiFi is the most flexible way of communication because of its cost and charac-

teristics, and it is suitable for any configuration or size of a multi-vehicle system. A recommended alternative is long-term evolution because of its point—multi-point capabilities and its network-based nature for extensive scenarios where there is no LOS between the elements. **Publication IV** describes the communication between the USV and aColor AUV for the co-operative system, which was performed with a direct connection via Ethernet. Additionally, the USBL system provides reliable communication if configured accordingly.

5.1.2 Multi-Master Architecture

Publication IV studies and implements the multi-master-fkie approach for the GNC architecture in the co-operative system. This approach provides ROS compatibility and simplicity in the system, and it is an adequate multi-master implementation for topic and services transactions [79]. However, this implementation can produce some shortcomings because of the continuous master state scanning and delay during the advertising and exchanging of data. The GNC architecture requires only three ROS topics exchanged in the co-operative system implementation, so this package provides a simple plug-and-play solution for the USV and aColor AUV platforms.

Figure 5.1 shows the communication links between the USV and aColor AUV platforms, denoting the ROS nodes for the multi-master-fkie architecture. The schematic also illustrates all the connections between the high-level, intermediate-level, and low-level controls for system implementation in both offshore vehicles. The GNC architecture for the co-operative system contains the following exchanged topics:

- */target*: target origin location using the USBL BODY reference frame.
- */USV_GPS*: USV position gathered from the GPS compass.
- */USV_heading*: USV heading angle gathered from the GPS compass.

5.2 Experimental Validation

The co-operative system requires experimental validation to ensure the correct performance of the GNC architecture in each vehicle. This section includes the description

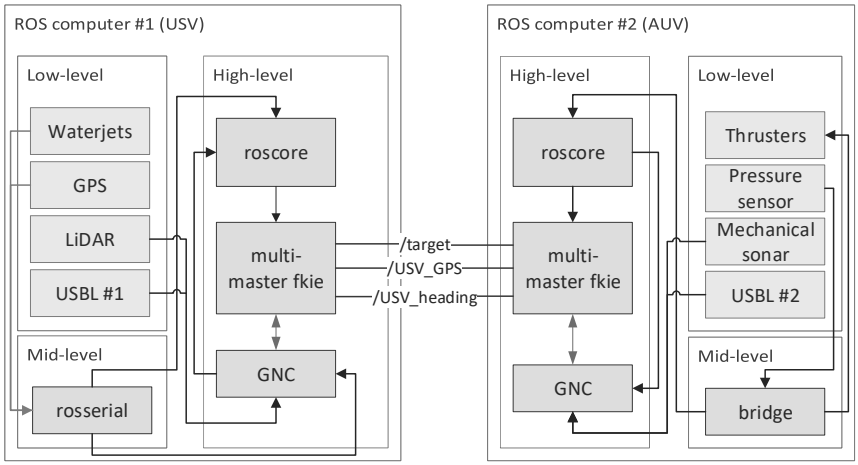


Figure 5.1 GNC architecture for the co-operative system using the multi-master-fkie approach. The schematic includes the connections between all sensors and actuators for each vehicle. [**Publication IV**]

for the system implementation in both USV and aColor AUV, involving the GNC algorithms implemented in the previous chapter, but here for the co-operative application. The decentralized approach helps the co-operative system run each vehicle separately, avoiding any possible disconnection problems between the platforms. Finally, the experimental results show the co-operative system implementation for a target detection scenario, where the USV works as a support platform for further inspection of the object.

5.2.1 Co-operative System Implementation

Based on **Publication IV**, the co-operative autonomous system successfully performs its task with the multi-vehicle system approach, where the applied modularity allows for the development, testing, and implementation of each of the GNC algorithms separately. Similar to the system's implementation in the previous chapter, the GNC architecture includes the target detection for each active ranging sensor, the LOS-based path following, and directional and attitude control using the modular approach in the co-operative system. Each of these tasks runs separate ROS nodes, providing the capabilities to test and implement each module separately. Figure 5.2 shows the modular GNC architecture with all ROS topics involved, indicating the necessary subscribers and publishers in the ROS network. In general, both platforms use similar

target detection and guidance control systems. However, the AUV does not include any obstacle avoidance in the control scenario because it tries to detect the first obstacle in the planar coverage area. The USV incorporates the obstacle avoidance capabilities based on the SBB approach, allowing for a continuous operation of the vehicle from the initial USV position to the target location.

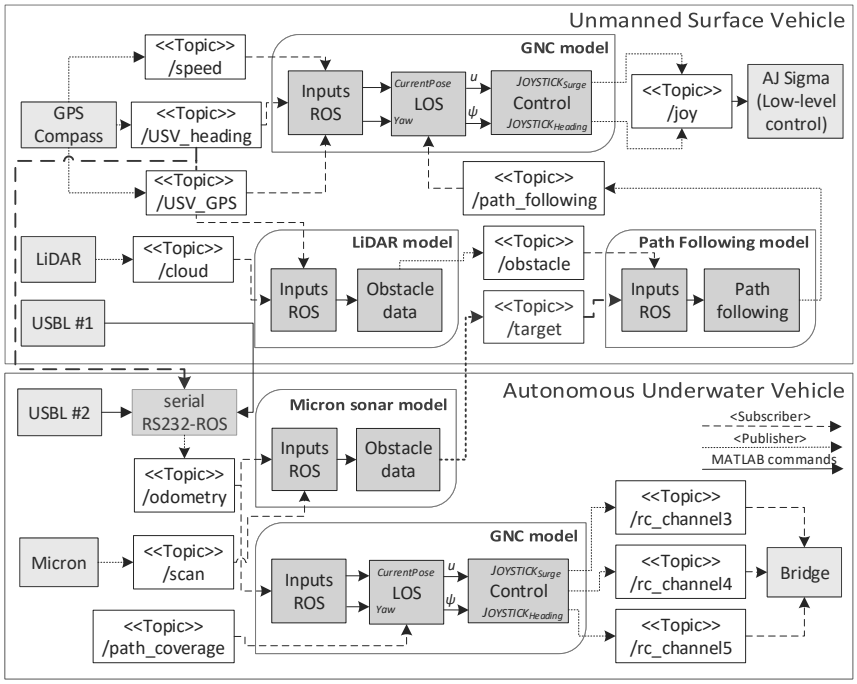


Figure 5.2 GNC architecture for the co-operative system, including all USV and aColor AUV modules involved. [**Publication IV**]

The co-operative system localization uses a GPS compass to gather the USV's pose and orientation in global coordinates and the USBL system to obtain the AUV's position in the USBL BODY reference frame. These positions relate to specific topics in the ROS system. The first generated ROS topic is the */odometry* from the AUV, which is based on the low-level serial messages generated and accepted by the SeaTrac USBL beacons [53]. The USBL system produces serial data strings in ASCII-Hex format, and the GNC algorithm decodes them to acquire the AUV localization data. This process uses the Serial package, translating the RS232 messages produced by the USBL system to a ROS topic [87]. Then, the GNC algorithm delivers PING messages from the main USBL #1 beacon placed at the USV to USBL #2 installed in the AUV. The message received from USBL #2 contains the necessary

AUV location in the BODY USBL coordinate system. After this, the GNC algorithm changes this reference frame to the NED coordinate system by utilizing a combination of translation and rotation matrices based on the */USV_heading* and */USV_GPS* variables from the GPS compass in the USV. The AUV follows a predefined path defined as the ROS topic */path_coverage*, which incorporates all waypoints to be followed by the underwater vehicle. The attitude control calculates the required heading command to reach each waypoint for the AUV's trajectory. Finally, the controller generates the required */rc_channel* commands and transmits them to the companion computer for the surge, heave, and yaw motions. This operation utilizes the BlueRov-ROS-playground ROS package [59].

As mentioned above, the AUV's purpose is to locate an underwater object in a predefined coverage area. If the target detection algorithm succeeds in its mission, the AUV will spot the detected target in the USBL reference frame, having [0,0] as the beacon position installed in the USV. Then, the USBL #2 transmits the target location to the USBL #1 placed in the USV, calculating its position in the NED coordinate system based on the USV's position and heading. The GNC algorithm calculates the target origin position in absolute coordinates, similar to the AUV's position, by rotating and translating the USBL reference frame. After the USV receives the target position included in the exchanged ROS topic */target*, the GNC algorithm declares the waypoints to perform the autonomous mission here starting from the initial USV position and ending at the target location. Then, the guidance system calculates the required course angle for the vehicle, similar to the single-vehicle implementation, and sends the joystick commands to the low-level control.

5.2.2 Experimental Results

Publication IV includes the control scenario for the co-operative system involving target detection, path planning, and control in both the USV and aColor AUV. These tasks depend greatly on the environmental drift forces produced by wind, wave, or currents, which involve many difficulties in open environment implementation. The main issue is that the control of the aColor AUV includes simple PID controllers and does not include those drift forces' compensations. Hence, this co-operative system is implemented with a modular approach, testing each vehicle separately to validate the GNC architecture. Figure 5.3 shows the AUV path-following implementation,

where the USV keeps stationary in the harbor at Pyhäjärvi Lake (Tampere, Finland).

Figure 5.3 USV and aColor AUV during the co-operative system field tests at Pyhäjärvi Lake (Tampere, Finland). [**Publication IV**]

The co-operative system includes underwater target detection as the first step in GNC implementation. The aColor AUV follows a predefined path based on a LOS-based, path-following algorithm, here using the attitude control with the surge, heave, and yaw motion controllers. The LOS-based guidance control determines the course angle to reach each waypoint of the predefined path. Figure 5.4 illustrates the aColor AUV trajectory utilizing the USBL system for navigation. Furthermore, Figure 5.5 presents the comparison between the course angle from the LOS-based guidance system and the field test data, demonstrating the correct performance of the simple attitude controller. Even though the PID parameters where obtained in a small water tank based on the Ziegler-Nichols method, the aColor AUV performed adequately in the open environment without needing to retune the PID controller. During path-following implementation, the mechanical imaging sonar placed in the aColor AUV tries to discover a target in a predefined coverage area by utilizing the target detection algorithm described in Section 4.1.1.

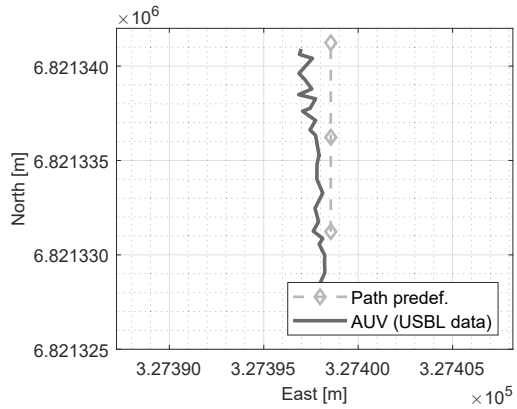


Figure 5.4 Co-operative system field tests: AUV trajectory for the path-following algorithm. [**Publication IV**]

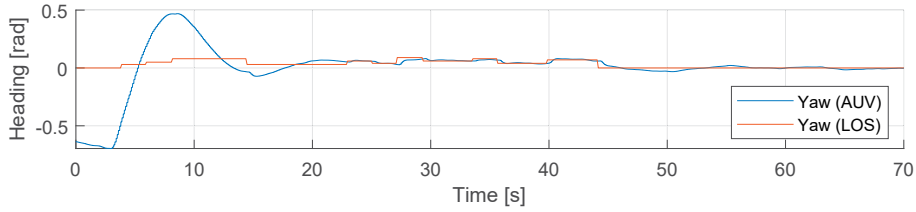


Figure 5.5 Co-operative system field tests: Comparison of the AUV course angle from the LOS-based guidance system with field test data. [**Publication IV**]

After detecting and locating the underwater target by the aColor AUV, the vehicle sends the target location to the USV's platform. As explained above, each vehicle contains its own ROS master. Hence, the AUV transmits the target position to the USV as a ROS topic */target* via the multi-master-fkie architecture. The GNC algorithm from the USV reads the target location and generates the necessary straight-line path to reach that position from the USV's location. Figure 5.6a illustrates the USV trajectory after defining the path based on the target location. Figure 5.6b illustrates the comparison between the yaw angle from the LOS-based guidance control and the field test data, and Figure 5.6c presents the corresponding LOS cross-track error $e(t)$. Additionally, Table 5.1 presents the non-dimensional indicators to evaluate the path-following performance of the USV platform. These plots and the table show the correct GNC architecture performance in the USV platform, despite the fact that the proposed control algorithms do not consider the environmental drift forces.

Table 5.1 Comparison of non-dimensional indicators for the USV trajectory.

Vehicle	RMSE	SD	MAE
USV	1.5482	0.7764	1.3397

5.3 Discussion

This chapter has described the GNC architectures for the co-operative system involving all capabilities of the autonomous offshore system. After designing and testing the GNC algorithms for an individual AOV, the co-operative system combined their competencies to include above- and below-water characterization. Relevant research

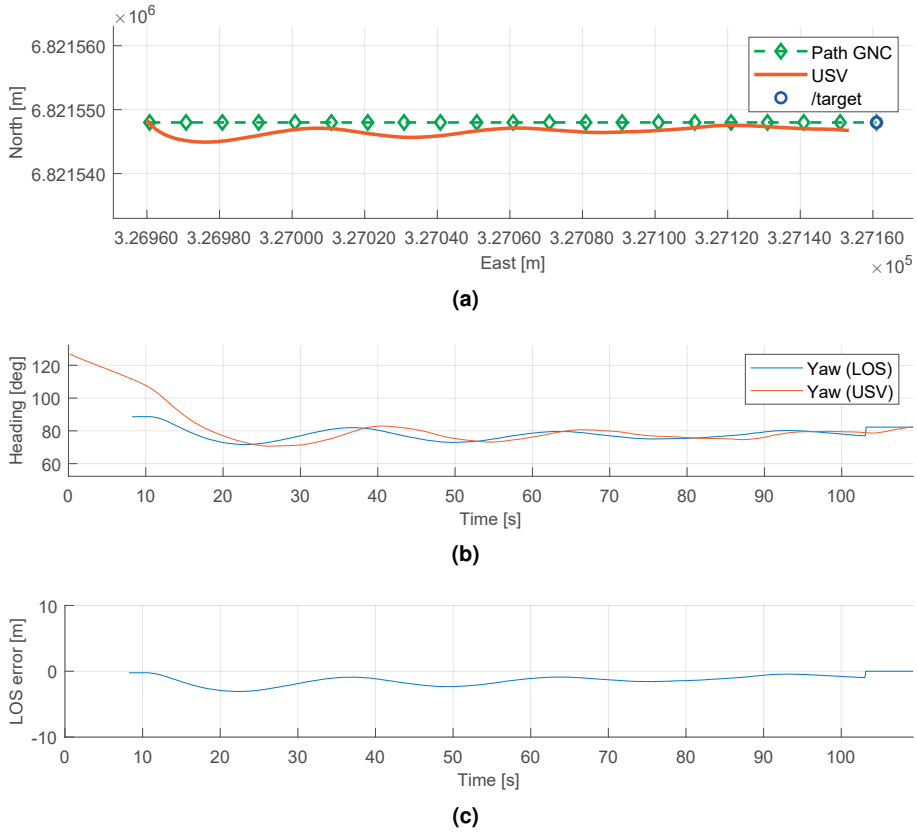


Figure 5.6 Co-operative system field tests: (a) USV trajectory, (b) Comparison of the USV yaw angle from the LOS-based guidance control with field test data, (c) LOS cross-track error $e(t)$ in the USV. [**Publication IV**]

works in the scientific community studied multi-vehicle systems, producing several advantages for perception systems compared with single-vehicle implementations. These works included homogeneous and heterogeneous systems formed by multiple USVs or AUVs or combining these vehicles in offshore applications. The current thesis combines the USV and aColor AUV, increasing the offshore system capabilities and using the same GNC architectures described in the previous chapter.

The experimental results presented in this chapter indicate the suitable performance of the co-operative system. The co-operative system included a LOS-based, path-following algorithm and a target detection algorithm in the aColor AUV. Then, the USV platform implemented path following with obstacle avoidance based on the target position obtained from the AUV. The presented results show the modular implementation and appropriate implementation of each GNC algorithm. As

mentioned in the previous chapter, the guidance control does not include the compensation for the drift forces from environmental variables. Thus, the path-following algorithms in both the AUV and USV platforms had some errors because of these environmental variables. These perturbances can be removed by improving the directional and attitude control in the AOVs, hence increasing the system accuracy. Moreover, the aColor AUV navigation included just the USBL system for positioning in absolute coordinates, not providing precise underwater localization of the vehicle. By enhancing the navigation and localization system, the path-following algorithm can improve its performance.

The co-operative system used a decentralized approach, having each AOV running an independent ROS master. These systems are more effective and decrease the communication network conditions compared with centralized systems. In this control scenario, the aColor AUV was directly connected to the USV via a tether to the network switch located at the USV, as shown in Figure 3.9. As explained above, each GNC algorithm implementation was performed separately. However, the multi-master-fkie architecture enabled the capability to communicate between the offshore vehicles, correctly operating the co-operative control scenario.

6 FINAL DISCUSSION

First, obtaining an accurate AOV mathematical model is essential for developing navigation algorithms, control methodology design, and simulation studies. The current thesis uses the MATLAB-Simulink tools for the design, modeling, and simulation of all studied AOVs, here using its SI and parameter estimation tools to estimate the hydrodynamic coefficients in the AOVs dynamic models. All these methods used field test data to estimate their coefficients in both transfer function and dynamic matrices. In general, the parameter estimation method was more accurate because it involved more dynamic coefficients in the AOVs mathematical models. However, SI provided a quick and simple solution to obtain a transfer function for the vehicle. In general, both methods give a suitable representation of the AOV. Nevertheless, the mathematical models can be improved by replacing the six DOFs dynamic model instead of the reduced orders of three DOFs for the USV and four DOFs for the AUVs.

The development of navigation algorithms and control methodology design is the next step after obtaining the AOV mathematical model. The primitive guidance and control system of an AOV includes both an attitude and path-following control system. The integral LOS guidance law has been selected for the guidance and control system in the current thesis without environmental forces estimation. Comparing the works from [10] and [7] with the results shown in Chapter 5, the USV implementation of the LOS-based path-following algorithm is correct. The non-dimensional indicators (RMSE, SD, and MAE) and cross-track error calculated during the field-testing shows an appropriate implementation for the co-operative system, reaching the final waypoint of the generated path. However, the environmental forces estimation and better tuning of the PID parameters can still improve the USV performance. The Girona500 AUV implementation in the sea environment describes a proper behavior of the position control with LOS guidance control for the yaw angle, providing better results in the non-dimensional indicators. The use of

diverse control methodology designs and algorithms in the AOVs shows the ability to implement different control systems, with the possibility to add improved capabilities to the current configurations.

This thesis uses the SBB approach to combine both obstacle avoidance capabilities and path-following by encompassing a static or slow-moving obstacle. The SBB approach allows for fast decision-making capabilities because of its simplicity and low data transfer. Comparing to the state-of-the-art in obstacle avoidance algorithms, this thesis improves [73] and [88] by using the SBB to create waypoints for the AOV trajectory. However, as mentioned in Chapter 4, there are some needed improvements to raise the system to the next level. COLREGs are mandatory if the USV needs to operate in populated area waterways. Thus, it would be necessary to research the inclusion of the SBB approach with COLREGs. Furthermore, the target detection algorithm in both above- and below-water environments provides a simple procedure for the co-operative system.

Finally, USVs usually co-operate with other autonomous vehicles, such as AUVs and UAVs, to accomplish more effective offshore missions. However, GNC methods can be relatively complex, so it is becoming crucial to fuse the data gathered from individual vehicles. As shown in Chapter 5, a decentralized modular and multi-layer GNC architecture with a multi-master approach allows for testing each offshore vehicle separately and the inclusion of new platforms, if necessary. All GNC algorithms had the same procedures, from the design and testing in a simulation environment to the actual implementation in a real-world scenario. Moreover, the standalone ROS node generated by MATLAB-Simulink provided a fast solution for C++ programming. The complete mission would enhance the results of the co-operative offshore system, but the testing of each vehicle separately correctly validates the designed GNC architecture.

7 CONCLUSIONS AND FUTURE WORK

This research was concerned with the design, modeling, and implementation of path following with obstacle avoidance algorithms as GNC architecture for a co-operative autonomous offshore system. The co-operative system was formed by a USV and AUV, allowing for above- and below-water characterization. Toward this end, the current thesis first focused on the mathematical model development for the involved AOVs, here based on nonlinear equations of motion using SI and parameter estimation methods. Second, the thesis provided a series of guidance and control methods for path following algorithms with obstacle avoidance. These methods formed the developed GNC architecture using a modular and multi-layer approach, which implemented each operation separately. After designing the GNC architecture for each platform, the current thesis included the co-operative system's implementation based on decentralized control techniques. The co-operative application aims to locate an object in an underwater cover area using the developed target detection algorithms while performing path following with obstacle avoidance algorithms. The experimental results show the simulation and field test scenarios, which present the capabilities and adequate performance for the designed GNC architecture. In Chapter 1, three research questions were presented. These questions have been thoroughly discussed and answered in the discussion sections during the previous chapters. Nonetheless, the conclusion comprises the answers to these research questions:

RQ1. What kind of implementation methods are needed for situational awareness and mission control in a system of multiple unmanned offshore vehicles?

First, the availability of an adequately accurate mathematical model in each vehicle is imperative for simulation study purposes, controller design, and development. The mathematical model has been developed using SI and parameter estimation methods for the USV and AUVs based on field test data. This thesis uses the SI and parameter estimation MATLAB-Simulink tools for simplifica-

tion because they provide sufficient accuracy and are close to the least-squares support vector machines method. Then, proper GNC systems with sensing, state estimation, and situational awareness capabilities can be designed for safe and efficient control of the co-operative system after developing an accurate mathematical model for each offshore vehicle. Thus, the target detection, path following, and guidance control algorithms can be designed in a simulation environment before implementation in a real-world scenario, providing all necessary capabilities for the autonomous operation. Situational awareness involves a target detection algorithm based on the SBB approach, and the mission control has a LOS-based, path-following algorithm as the chosen guidance and control system. These algorithms allow for similar implementation in both offshore vehicles because the only difference is the low-level actuators and sensors, here depending on the above- or below-water applications.

RQ2. What kind of architecture should be used for multi-sensor networks and integration in offshore vehicle applications?

The modular and multi-layer GNC architecture provides computationally cheap and easy implementation for the required autonomous capabilities. The modular approach in the GNC architecture allows for the continuous testing of each of the developed algorithms individually, with it being easier to detect an error or malfunction in the autonomous operation. Additionally, each of the modules includes a standalone ROS node generated by MATLAB-Simulink, which provides a fast solution for C++ programming by skipping numerous programming steps and fulfilling the required programming standards. The multi-layer architecture is formed by high-level, intermediate-level, and low-level controls, providing the necessary tools and packages to access sensor data, process it, and generate an appropriate response for the AOV actuators. The high-level control is in charge of the advanced logic operations and performs intricate computations. The low-level control involves the sensors and actuators for each AOV, forming the interface for basic vehicle operations. Finally, the intermediate-level links the low-level and high-level controls to perform data acquisition and basic logic operations.

RQ3. What kind of co-operative framework is needed for shared intelligence in multiple autonomous robotic systems?

After designing the GNC architecture with situational awareness and mission control for individual AOVs, the co-operative system combines them to implement above- and below-water characterization. This thesis comprises an autonomous multi-vehicle system working in co-operation formed by a USV and AUV. The multi-master architecture in a decentralized framework provides the necessary tools for shared intelligence between different offshore platforms, including flexible, profitable, and low communication requirements. Furthermore, using a ROS common framework enables a solution to include data acquisition and processing from sensors and produce the required commands to the vehicle actuators. The co-operative system uses a decentralized approach has each AOV running an independent ROS master. These systems are more effective and decrease the communication network conditions compared with centralized systems. In the co-operative system's implementation, each GNC algorithm implementation is performed separately by running a separate ROS node. Nonetheless, the multi-master-fkie architecture enabled the capability to communicate specific ROS topics between the offshore vehicles, correctly operating the co-operative control scenario.

7.1 Future Work

Future research will improve the USV and AUVs' mathematical models, here designing and modeling the six DOFs models for each vehicle. Furthermore, it will include more accurate parameter estimation methods to define the hydrodynamic coefficients with extensive field tests. Future work will also comprise the complete mathematical model of the aColor AUV after installing the necessary onboard sensors to gather all required navigation and localization data. Future research will also include guidance and control algorithms for comparison rather than LOS-based path following with other control systems that are not PID based. The control system will also comprise environmental variable compensation, which would considerably increase the system's capabilities. Regarding the obstacle avoidance capabilities, future work will include static, slow-motion, and high-speed target detection, improving the performance of the co-operative system, especially in the USV operation. The implementation will also consider multiple obstacle scenarios. This will be possible by calculating a projected SBB for each moving object in the offshore vehicle trajectory.

Other perception sensors rather than LiDAR or mechanical imaging sonar will also be studied and implemented in the target detection algorithms. Concerning the co-operative system, future work will involve additional platforms in the system, which could be additional USVs or AUVs or even including UAVs. An aerial vehicle would considerably increase the co-operative system's capabilities by including complete multi-domain awareness of the offshore environment.

REFERENCES

[1] Alamarin Jet Oy. *AJ 245*. 2020. URL: https://alamarinjet.com/products/jet/aj-245/.

[2] M. Alberri, S. Hegazy, M. Badra, M. Nasr, O. M. Shehata and E. I. Morgan. Generic ROS-based Architecture for Heterogeneous Multi-Autonomous Systems Development. *2018 IEEE International Conference on Vehicular Electronics and Safety (ICVES)*. IEEE. 2018, 1–6.

[3] P. Batista, C. Silvestre and P. Oliveira. A sensor-based controller for homing of underactuated AUVs. *IEEE transactions on robotics* 25.3 (2009), 701–716.

[4] J. V. Beck and K. J. Arnold. *Parameter estimation in engineering and science*. James Beck, 1977.

[5] Blue Robotics Inc. *T200 Thruster*. 2021. URL: https://bluerobotics.com/store/thrusters/t100-t200-thrusters/t200-thruster-r2-rp/.

[6] E. Børhaug, A. Pavlov, E. Panteley and K. Y. Pettersen. Straight line path following for formations of underactuated marine surface vessels. *IEEE transactions on control systems technology* 19.3 (2010), 493–506.

[7] E. Borhaug, A. Pavlov and K. Y. Pettersen. Integral LOS control for path following of underactuated marine surface vessels in the presence of constant ocean currents. *2008 47th IEEE Conference on Decision and Control*. IEEE. 2008, 4984–4991.

[8] P. Bouchier. Embedded ROS [ROS Topics]. *IEEE Robotics & Automation Magazine* 20.2 (June 2013), 17–19.

[9] M. Breivik and T. I. Fossen. Path following for marine surface vessels. *Oceans' 04 MTS/IEEE Techno-Ocean'04 (IEEE Cat. No. 04CH37600)*. Vol. 4. IEEE. 2004, 2282–2289.

[10] M. Breivik and T. I. Fossen. Guidance laws for autonomous underwater vehicles. *Underwater vehicles* 4 (2009), 51–76.

[11] P. Cardenas and E. A. de Barros. Estimation of AUV hydrodynamic coefficients using analytical and system identification approaches. *IEEE Journal of Oceanic Engineering* (2019).

[12] *CCPilot VC Technical manual*. Product revision: 2.4. CrossControl. Mar. 2019.

[13] P. Cieślak. Stonefish: An Advanced Open-Source Simulation Tool Designed for Marine Robotics, With a ROS Interface. *OCEANS 2019-Marseille*. IEEE. 2019, 1–6.

[14] N. R. Council et al. *Technology development for army unmanned ground vehicles*. National Academies Press, 2003.

[15] J. Curcio, J. Leonard, J. Vaganay, A. Patrikalakis, A. Bahr, D. Battle, H. Schmidt and M. Grund. Experiments in moving baseline navigation using autonomous surface craft. *Proceedings of OCEANS 2005 MTS/IEEE*. IEEE. 2005, 730–735.

[16] J. A. Curcio. Rules of the road for unmanned marine vehicles. *Springer Handbook of Ocean Engineering*. Springer, 2016, 517–526.

[17] G. DNV. *Class Guideline-Autonomous and remotely operated ships*. Tech. rep. DNVGL-CG-0264, September, 2018.

[18] J. Evans and M. Nahon. Dynamics modeling and performance evaluation of an autonomous underwater vehicle. *Ocean Engineering* 31.14-15 (2004), 1835–1858.

[19] M. F. Fallon, G. Papadopoulos, J. J. Leonard and N. M. Patrikalakis. Cooperative AUV navigation using a single maneuvering surface craft. *The International Journal of Robotics Research* 29.12 (2010), 1461–1474.

[20] E. G. Font, F. Bonin-Font, P.-L. Negre, M. Massot and G. Oliver. USBL integration and assessment in a multisensor navigation approach for AUVs. *IFAC-PapersOnLine* 50.1 (2017), 7905–7910.

[21] T. I. Fossen. Guidance and control of ocean vehicles. *University of Trondheim, Norway, Printed by John Wiley & Sons, Chichester, England, Doctors Thesis* (1999).

[22] T. I. Fossen and A. M. Lekkas. Direct and indirect adaptive integral line-of-sight path-following controllers for marine craft exposed to ocean currents. *International Journal of Adaptive Control and Signal Processing* 31.4 (2017), 445–463.

[23] T. I. Fossen and T. Perez. *Marine Systems Simulator (MSS)*. 2004. URL: `https://github.com/cybergalactic/MSS`.

[24] T. I. Fossen and S. I. Sagatun. Adaptive control of nonlinear systems: A case study of underwater robotic systems. *Journal of Robotic Systems* 8.3 (1991), 393–412.

[25] T. I. Fossen. Handbook of Marine Craft Hydrodynamics and Motion Control. John Wiley & Sons, Ltd, Apr. 2011.

[26] T. Fossen and A. Ross. Nonlinear modelling, identification and control of UUVs. *IEEE Control Engineering Series* 69 (2006), 13.

[27] E. Galceran, R. Campos, N. Palomeras, D. Ribas, M. Carreras and P. Ridao. Coverage path planning with real-time replanning and surface reconstruction for inspection of three-dimensional underwater structures using autonomous underwater vehicles. *Journal of Field Robotics* 32.7 (2015), 952–983.

[28] M. J. Grimble and M. A. Johnson. *Optimal control and stochastic estimation: theory and applications*. Vol. 1. John Wiley & Sons, 1988.

[29] R. Halterman and M. Bruch. Velodyne HDL-64E lidar for unmanned surface vehicle obstacle detection. *Unmanned Systems Technology XII*. Vol. 7692. SPIE Proceedings. SPIE, Apr. 2010, 76920D.

[30] J. Han, J. Xiong, Y. He, F. Gu and D. Li. Nonlinear Modeling for a Water-Jet Propulsion USV: An Experimental Study. *IEEE Transactions on Industrial Electronics* 64.4 (Apr. 2017), 3348–3358.

[31] J. Han, Y. Cho and J. Kim. Coastal SLAM with marine radar for USV operation in GPS-restricted situations. *IEEE Journal of Oceanic Engineering* 44.2 (2019), 300–309.

[32] J. Han, Y. Cho, J. Kim, J. Kim, N.-s. Son and S. Y. Kim. Autonomous collision detection and avoidance for ARAGON USV: Development and field tests. *Journal of Field Robotics* 37.6 (2020), 987–1002.

[33] A. J. Healey and D. Lienard. Multivariable sliding mode control for autonomous diving and steering of unmanned underwater vehicles. *IEEE journal of Oceanic Engineering* 18.3 (1993), 327–339.

[34] H. K. Heidarsson and G. S. Sukhatme. Obstacle detection and avoidance for an autonomous surface vehicle using a profiling sonar. *2011 IEEE International Conference on Robotics and Automation*. IEEE. 2011, 731–736.

[35] Y. Huang, L. Chen, P. Chen, R. R. Negenborn and P. Van Gelder. Ship collision avoidance methods: State-of-the-art. *Safety science* 121 (2020), 451–473.

[36] C. C. Insaurralde. *Intelligent Autonomy for Unmanned Marine Vehicles*. Springer, 2015.

[37] Iqua Robotics. *COLA2 Wiki (version 20.10)*. 2020. URL: `https://bitbucket.org/iquarobotics/cola2_wiki/src/master/README.md`.

[38] J. Kim, K. Kim, H. S. Choi, W. Seong and K.-Y. Lee. Estimation of hydrodynamic coefficients for an AUV using nonlinear observers. *IEEE journal of oceanic engineering* 27.4 (2002), 830–840.

[39] N. Koenig and A. Howard. Design and use paradigms for gazebo, an open-source multi-robot simulator. *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. Vol. 3. IEEE. 2004, 2149–2154.

[40] J. Larson, M. Bruch and J. Ebken. Autonomous navigation and obstacle avoidance for unmanned surface vehicles. *Unmanned Systems Technology VIII*. Vol. 6230. International Society for Optics and Photonics. 2006, 623007.

[41] J. J. Leonard, A. A. Bennett, C. M. Smith, H. Jacob and S. Feder. Autonomous underwater vehicle navigation. *MIT Marine Robotics Laboratory Technical Memorandum*. Citeseer. 1998.

[42] X. Liang, X. Qu, L. Wan and Q. Ma. Three-dimensional path following of an underactuated AUV based on fuzzy backstepping sliding mode control. *International Journal of Fuzzy Systems* 20.2 (2018), 640–649.

[43] Z. Liu, Y. Zhang, X. Yu and C. Yuan. Unmanned surface vehicles: An overview of developments and challenges. *Annual Reviews in Control* 41 (2016), 71–93.

[44] L. Ljung. System identification. *Wiley Encyclopedia of Electrical and Electronics Engineering* (1999), 1–19.

[45] M. M. M. Manhães, S. A. Scherer, M. Voss, L. R. Douat and T. Rauschenbach. UUV simulator: A gazebo-based package for underwater intervention and multi-robot simulation. *OCEANS 2016 MTS/IEEE Monterey*. IEEE. 2016, 1–8.

[46] *MATLAB Coder User's Guide*. Release 2020a. The MathWorks, Inc. Natick, MA, 2011.

[47] *Micron sonar - Product manual*. 0650-SOM-00003, Issue: 02. Tritech International Ltd.

[48] P. A. Miller, J. A. Farrell, Y. Zhao and V. Djapic. Autonomous underwater vehicle navigation. *IEEE Journal of Oceanic Engineering* 35.3 (2010), 663–678.

[49] S. Moe and K. Y. Pettersen. Set-based Line-of-Sight (LOS) path following with collision avoidance for underactuated unmanned surface vessel. *24th Mediterranean Conference on Control and Automation (MED)*. IEEE. 2016, 402–409.

[50] D. Moreno-Salinas, D. Chaos, J. M. de la Cruz and J. Aranda. Identification of a Surface Marine Vessel Using LS-SVM. *Journal of Applied Mathematics* 2013 (2013), 1–11.

[51] *MRS1000: Operating instructions*. 8020494/12FY/2019-04-02. SICK AG. Erwin-Sick-Str. 1, 79183, Waldkirch, Germany, Apr. 2019.

[52] D. Mu, G. Wang, Y. Fan, Y. Bai and Y. Zhao. Path following for podded propulsion unmanned surface vehicle: Theory, simulation and experiment. *IEEJ Transactions on Electrical and Electronic Engineering* 13.6 (June 2018), 911–923.

[53] J. A. Neasham, G. Goodfellow and R. Sharphouse. Development of the "Seatrac" miniature acoustic modem and USBL positioning units for subsea robotics and diver applications. *OCEANS 2015-Genova*. IEEE. 2015, 1–8.

[54] K. Nomoto, K. Taguchi, K. Honda and S. Hirano. On the Steering Qualities of Ships. *Journal of Zosen Kiokai* 1956.99 (1956), 75–82.

[55] S.-R. Oh, J. Sun, Z. Li, E. A. Celkis and D. Parsons. System identification of a model ship using a mechatronic system. *IEEE/ASME Transactions on Mechatronics* 15.2 (2009), 316–320.

[56] M. Ollikainen and M. Ollikainen. The Finnish coordinate reference systems. *Finnish Geodetic Institute and the National Land Survey of Finland* (2004). URL: https://www.maanmittauslaitos.fi/sites/default/files/Finnish_Coordinate_Systems.pdf.

[57] F. A. Papoulias. Bifurcation analysis of line of sight vehicle guidance using sliding modes. *Int. J. Bifurcation Chaos* 1 (1991), 849–865.

[58] L. Paull, S. Saeedi, M. Seto and H. Li. AUV navigation and localization: A review. *IEEE Journal of Oceanic Engineering* 39.1 (2013), 131–149.

[59] P. J. Pereira. *BlueRov-ROS-playground. Github repository*. 2020. URL: https://github.com/patrickelectric/bluerov_ros_playground.

[60] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler and A. Y. Ng. ROS: an open-source Robot Operating System. *ICRA workshop on open source software*. Vol. 3. 3.2. Kobe, Japan. 2009, 5.

[61] F. Rezazadegan, K. Shojaei, F. Sheikholeslam and A. Chatraei. A novel approach to 6-DOF adaptive trajectory tracking control of an AUV in the presence of parameter uncertainties. *Ocean Engineering* 107 (2015), 246–258.

[62] D. Ribas, N. Palomeras, P. Ridao, M. Carreras and A. Mallios. Girona 500 auv: From survey to intervention. *IEEE/ASME Transactions on mechatronics* 17.1 (2011), 46–53.

[63] D. Ribas, P. Ridao and J. Neira. *Underwater SLAM for structured environments using an imaging sonar*. Vol. 65. Springer, 2010.

[64] *ROS Toolbox User's Guide*. Release 2020a. The MathWorks, Inc. Natick, MA, 2019.

[65] J. Ross, J. Lindsay, E. Gregson, A. Moore, J. Patel and M. Seto. Collaboration of multi-domain marine robots towards above and below-water characterization of floating targets. *2019 IEEE International Symposium on Robotic and Sensors Environments (ROSE)*. IEEE. 2019, 1–7.

[66] R. Rout and B. Subudhi. NARMAX self-tuning controller for line-of-sight-based waypoint tracking for an autonomous underwater vehicle. *IEEE Transactions on Control Systems Technology* 25.4 (2016), 1529–1536.

[67] M. T. Sabet, P. Sarhadi and M. Zarini. Extended and Unscented Kalman filters for parameter estimation of an autonomous underwater vehicle. *Ocean Engineering* 91 (2014), 329–339.

[68] S. I. Sagatun and T. I. Fossen. Lagrangian formulation of underwater vehicles' dynamics. *Conference Proceedings 1991 IEEE International Conference on Systems, Man, and Cybernetics*. IEEE. 1991, 1029–1034.

[69] E. I. Sarda and M. R. Dhanak. Launch and recovery of an autonomous underwater vehicle from a station-keeping unmanned surface vehicle. *IEEE Journal of Oceanic Engineering* 44.2 (2018), 290–299.

[70] C. Shen, Y. Shi and B. Buckham. Integrated path planning and tracking control of an AUV: A unified receding horizon optimization approach. *IEEE/ASME Transactions on Mechatronics* 22.3 (2016), 1163–1173.

[71] J. Shin, D. J. Kwak and Y.-i. Lee. Adaptive path-following control for an unmanned surface vessel using an identified dynamic model. *IEEE/ASME transactions on mechatronics* 22.3 (2017), 1143–1153.

[72] E. Simetti, G. Casalino, S. Torelli, A. Sperinde and A. Turetta. Floating underwater manipulation: Developed control methodology and experimental validation within the TRIDENT project. *Journal of Field Robotics* 31.3 (2014), 364–385.

[73] E. Simetti, S. Torelli, G. Casalino and A. Turetta. Experimental results on obstacle avoidance for high speed unmanned surface vehicles. *2014 Oceans-St. John's*. IEEE. 2014, 1–6.

[74] R. Simoni, P. R. Rodríguez, P. Cieślak and D. Youakim. A novel approach to obstacle avoidance for an I-AUV. *2018 IEEE/OES Autonomous Underwater Vehicle Workshop (AUV)*. IEEE. 2018, 1–6.

[75] *Simulink Control Design User's Guide*. Release 2020a. The MathWorks, Inc. Natick, MA, 2004.

[76] *Simulink Design Optimization User's Guide*. Release 2020a. The MathWorks, Inc. Natick, MA, 2009.

[77] *System Identification Toolbox User's Guide*. Release 2020a. The MathWorks, Inc. Natick, MA, 1988.

[78] C. Tam, R. Bucknall and A. Greig. Review of collision avoidance and path planning methods for ships in close range encounters. *The Journal of Navigation* 62.3 (2009), 455.

[79] A. Tiderko, F. Hoeller and T. Röhling. The ROS multimaster extension for simplified deployment of multi-robot systems. *Robot Operating System (ROS)*. Springer, 2016, 629–650.

[80] A. Vasilijević, Đ. Nađ, F. Mandić, N. Mišković and Z. Vukić. Coordinated navigation of surface and underwater marine robotic vehicles for ocean sampling and environmental monitoring. *IEEE/ASME transactions on mechatronics* 22.3 (2017), 1174–1184.

[81] E. Vidal, M. Moll, N. Palomeras, J. D. Hernández, M. Carreras and L. E. Kavraki. Online multilayered motion planning with dynamic constraints for autonomous underwater vehicles. *2019 International Conference on Robotics and Automation (ICRA)*. IEEE. 2019, 8936–8942.

[82] N. Wang and X. Pan. Path Following of Autonomous Underactuated Ships: A Translation-Rotation Cascade Control Approach. *IEEE/ASME Transactions on Mechatronics* 24.6 (Dec. 2019), 2583–2593.

[83] Y. Wang, X. Yu, X. Liang and B. Li. A COLREGs-based obstacle avoidance approach for unmanned surface vehicles. *Ocean Engineering* 169 (Dec. 2018), 110–124.

[84] M. S. Wiig, K. Y. Pettersen and T. R. Krogstad. A 3D reactive collision avoidance algorithm for underactuated underwater vehicles. *Journal of Field Robotics* 37.6 (2020), 1094–1122.

[85] C. J. Willmott and K. Matsuura. Advantages of the mean absolute error (MAE) over the root mean square error (RMSE) in assessing average model performance. *Climate research* 30.1 (2005), 79–82.

[86] M. T. Wolf, C. Assad, Y. Kuwata, A. Howard, H. Aghazarian, D. Zhu, T. Lu, A. Trebi-Ollennu and T. Huntsberger. 360-degree visual detection and target tracking on an autonomous surface vehicle. *Journal of Field Robotics* 27.6 (2010), 819–833.

[87] W. Woodall and J. Harrison. *Serial, Cross-platform, Serial Port library written in C++*. 2020. URL: `http://wjwwood.io/serial/`.

[88] P. Wu, S. Xie, H. Liu, M. Li, H. Li, Y. Peng, X. Li and J. Luo. Autonomous obstacle avoidance of an unmanned surface vehicle based on cooperative manoeuvring. *Industrial Robot An International Journal* 44.1 (Jan. 2017), 64–74.

[89] J. Zhuang, L. Zhang, S. Zhao, J. Cao, B. Wang and H. Sun. Radar-based collision avoidance for unmanned surface vehicles. *China Ocean Engineering* 30.6 (2016), 867–883.

[90] J. G. Ziegler and N. B. Nichols. Optimum settings for automatic controllers. *trans. ASME* 64.11 (1942).

PUBLICATIONS

PUBLICATION I

Model-based Path Planning and Obstacle Avoidance Architecture for a Twin Jet Unmanned Surface Vessel

Jose Villa, Jussi Aaltonen, Kari T. Koskinen

2019 Third IEEE International Conference on Robotic Computing (IRC) 2019, 427–428.
DOI: 10.1109/IRC.2019.00083

Model-based path planning and obstacle avoidance architecture for a twin jet Unmanned Surface Vessel

Jose Villa
Mechatronics Research Group
Tampere University (TUNI)
Tampere, Finland
jose.villaescusol@tuni.fi

Jussi Aaltonen
Mechatronics Research Group
Tampere University (TUNI)
Tampere, Finland
jussi.aaltonen@tuni.fi

Kari T. Koskinen
Mechatronics Research Group
Tampere University (TUNI)
Tampere, Finland
kari.koskinen@tuni.fi

***Abstract*—The aim of this study is to implement a model-based control architecture for a twin jet Unmanned Surface Vessel (USV) using the line-of-sight (LOS) path algorithm for straight-line. The modeling and design of the motion control system are studied using MATLAB-Simulink and the real-time implementation is done by using Robot Operating System (ROS). To simplify the real-time implementation and thus, programming files, the necessary code generation is done by generating a Standalone ROS Node from Simulink software, making the process easier for researchers with non-advanced programming skills. Simulation and field test results are shown to validate the model-based control architecture.**

***Index Terms*—model-based, path following, simulation, implementation, USV, MATLAB-Simulink, ROS**

I. Introduction

In recent years, the research involving autonomous systems has increased in many different scenarios, including the unmanned surface vessels (USV) in lakes or sea areas. Research on USV has included trajectory tracking and path following control problems. Furthermore, several studies include different guidance laws for path following and collision avoidance [1], [2]. Those studies require an extensive code programming process which is time-consuming and involves many steps before implementing the desired algorithm successfully. There are some software tools which allows the generation of C and C++ code, such as MATLAB/Simulink. The modeling and design of the motion control system can be studied using MATLAB/Simulink, and the real-time implementation can be done by using Robot Operating System (ROS) [3].

In this work, a model-based control architecture for a USV is studied following the USV design used in a twin jet research vessel. Furthermore, a simple path following algorithm is described. In addition to that, all schematics for simulation and implementation are shown in MATLAB/Simulink to create the necessary model subsystems in order to generate and build a standalone ROS node for real-time operation of the USV.

II. USV Control architecture

A. Description of the USV platform

The Research vessel used in this study is an aluminum hull with thrust vectoring twin water-jet. It uses two marine diesel engine with 170 kW of rated power and an intelligent operation (IO) control system (AJ IO [4]). Furthermore, by using a twin jet configuration, the USV has an ultimate maneuvering accuracy as it can move in all directions without bow thrusters.

The Research Vessel used in this study contains a Linux CPU as a ROS Master computer, which is connected to the rest of instrumentation (including GPS Compass and IO system). The necessary control commands are sent by a computer running MATLAB/Simulink which is a ROS node in the system.

B. Path following algorithm

The aim of a path following algorithm is to reach every point of a predefined path independent of time. Line-of-sight (LOS) guidance is used in this study [5]. A LOS vector from the surface vessel to the next way-point or a point on the path between two way-points can be used for heading control. There are no temporal constraints such as the representation of obstacles and other positional constraints in this study. Furthermore, the switching mechanism is defined as a circle of acceptance for surface vessels [5].

C. Control system

The complete control system is based on two different PID controllers for the surge and heading control, and their parameters are obtained by using Rapid Control Prototyping (RCP) during the field tests. Surge control keeps the surface vessel in a predefined constant speed. LOS algorithm sends heading commands to the autopilot to accomplish the predefined path. Furthermore, the feedback loop includes a cascaded Low-Pass and Notch Filtering to reduce motions induced by waves [5]. By using this cascade filter, small waves have been suppressed getting constant outputs (surge and yaw) in the controller.

D. Block diagram (Simulation)

The block diagram for Simulation is based on four main subsystems (Fig. 1). Navigation subsystem contains all LOS path following controller equations. The control block contains both PID controller and cascade filter. The boat model has joystick parameters as input. The conversion from forces to joystick commands is included using tables provided by the jet manufacturer. The boat model subsystem contains all dynamic equations to get the surge and heading for the simulator.

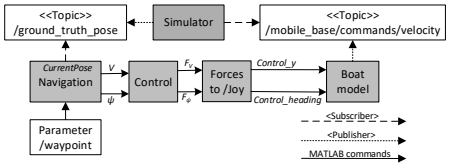


Fig. 1. ROS schematic for Navigation and Control (Simulation)

E. Block diagram (Implementation)

The implementation schematic can be seen in Fig. 2, showing the Simulink model with respective subsystems and both GPS Compass and IO system from the USV. The sensor data is obtained from the GPS Compass, which uses WGS84 (Latitude, Longitude) instead of a projected coordinate system, such as UTM (Universal Transverse Mercator) or ETRS-TM35FIN coordinate systems. Conversion from WGS84 to ETRS-TM35FIN system uses an already defined function [6].

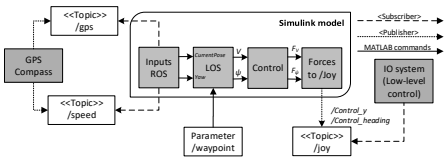


Fig. 2. Simulink schematic for Navigation and Control (Implementation)

The test model is based on the Robotic System Toolbox from MATLAB/Simulink [7]. In order to define the necessary ROS topics, a standalone ROS node is generated and built from the Simulink model [8], avoiding all significant programming issues and the coding time that every ROS package involves.

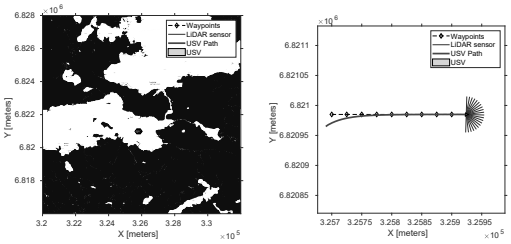
III. Simulation and field-test results

Using the Simulink model shown in subsection II-D, Fig. 3(b) shows the results of the path-following controller. Plots are simulated using ETRS-TM35FIN coordinate system. The simulation has been done adding an offset of (-10,-20) meters from the initial way-point as a starting point of the simulation.

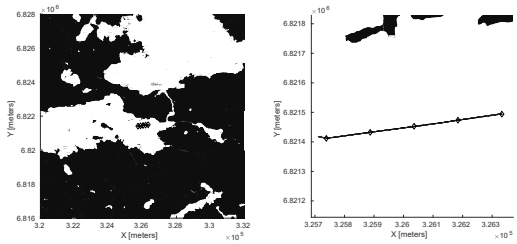
The field test results from the block diagram described in subsection II-E are shown in Fig. 3(d). The blue line shows the GPS data recorded during the test and the black dotted line is the predefined path. The error between this predefined and the USV path is less than 1 meter (measured by the tangential distance to the path). This difference is due to environmental elements such as wind or wave drift forces, which compensation is not considered in this study.

IV. Conclusions

In conclusion, a model-based control architecture for a twin jet Unmanned Surface Vessel is implemented for real-time implementation. Due to the real-time implementation, it is not possible to use the same software for simulation and implementation (MATLAB/Simulink) and a second software



(a) Map view (simulation) (b) Zoom view (simulation)



(c) Map view (field-test) (d) Zoom view (field-test)

Fig. 3. Simulink-ROS: Simulation and field-test results

is included (ROS). To avoid programming issues and the related time consuming, this paper describes the schematic and connections to generate and built a standalone ROS node. This ROS node includes a simple straight-line path following algorithm (LOS controller). Both simulation and field test results are shown to confirm the control architecture.

Acknowledgment

This paper is based on the aColor project funded by Technology Industries of Finland Centennial and Jane & Aatos Erkko Foundations. The authors gratefully acknowledge the contributions of Sauli Virta from Alamarin-Jet Oy.

References

[1] T. I. Fossen, M. Breivik, and R. Skjetne, "Line-of-sight path following of underactuated marine craft," *IFAC Proceedings Volumes*, vol. 36, no. 21, pp. 211–216, 2003.

[2] S. Moe and K. Y. Pettersen, "Set-based line-of-sight (los) path following with collision avoidance for underactuated unmanned surface vessel," in *Control and Automation (MED), 2016 24th Mediterranean Conference on*. IEEE, 2016, pp. 402–409.

[3] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng, "Ros: an open-source robot operating system," in *ICRA workshop on open source software*, vol. 3, no. 3.2. Kobe, Japan, 2009, p. 5.

[4] Alamarin-Jet Oy. (2018) Intelligent operation - AJ IO. [Online]. Available: https://www.alamarinjet.com/products-intelligent-operation

[5] T. I. Fossen, *Handbook of marine craft hydrodynamics and motion control*. John Wiley & Sons, 2011.

[6] O. Lammi. (2014) Coordinate system functions. Github Repository.

[7] The MathWorks, Inc. (2018) Design and test algorithms for robotics applications. [Online]. Available: https://se.mathworks.com/products/robotics.html

[8] ——. (2018) Generate a standalone ROS node from simulink. [Online]. Available: https://se.mathworks.com/help/robotics/examples/generate-a-standalone-ros-node-in-simulink.html

PUBLICATION II

Model-based Guidance, Navigation and Control Architecture for an Autonomous Underwater Vehicle

Jose Villa, Guillem Vallicrosa, Jussi Aaltonen, Pere Ridao, Kari T. Koskinen

Global Oceans 2020: Singapore – U.S. Gulf Coast 2020, 1–6.
DOI: 10.1109/IEEECONF38699.2020.9389247

Model-based Guidance, Navigation and Control architecture for an Autonomous Underwater Vehicle

1st Jose Villa
Mechatronics Research Group
Tampere University
Tampere, Finland
jose.villa@tuni.fi

2nd Guillem Vallicrosa
Computer Vision and Robotics Institute
Universitat de Girona
Girona, Spain
gvallicrosa@eia.udg.edu

3rd Jussi Aaltonen
Mechatronics Research Group
Tampere University
Tampere, Finland
jussi.aaltonen@tuni.fi

4th Pere Ridao
Computer Vision and Robotics Institute
Universitat de Girona
Girona, Spain
pere.ridao@udg.edu

5th Kari T. Koskinen
Mechatronics Research Group
Tampere University
Tampere, Finland
kari.koskinen@tuni.fi

***Abstract*—This article studies the design, modeling, and implementation of a model-based Guidance, Navigation, and Control (GNC) architecture for an Autonomous Underwater Vehicle (AUV). First, effective simulation modeling is developed using a theoretical six-degree-of-freedom (6DoF) dynamic model. Then, this study considers two GNC algorithms (simple and advanced). The simple GNC algorithm considers three different kinds of PID controllers (velocity, velocity-position, and position), and the advanced GNC algorithm enables path-following and data acquisition and processing from an underwater sensor. The path following is based on the position control using a unique PID controller and obtains its waypoints from a wall detection algorithm. This wall detection algorithm uses a mechanical imaging sonar as the main perception sensor. Finally, an implementation challenge in two control scenarios is addressed to validate the designed GNC architecture and to carry out model-verification of the position PID controller.**

***Index Terms*—model-verification, model-based, GNC, AUV, MATLAB-Simulink, ROS**

I. Introduction

Due to the amount of unknown and unexplored areas in oceans, seas, and lakes, and the extensive range of autonomous vehicle applications, underwater research is currently becoming more relevant in scientific research. Remote Operated Vehicles (ROVs) and Autonomous Underwater Vehicles (AUVs) are two of the most common classifications for underwater vehicles. There are numerous research topics for underwater vehicles from conventional sonar and video imaging surveys to autonomous intervention tasks, such as path planning [1], obstacle avoidance [2], or underwater manipulation [3]. Underwater Simultaneous Localization And Mapping (SLAM) has been a key topic in underwater research. In [4], different methods are described for map-based localization as well as a novel approach for SLAM.

EUMarineRobots project that has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 731103.

All these applications require accurate maneuvering of the underwater vehicle, where Guidance, Navigation, and Control (GNC) play an essential role. GNC deals with the design of systems that remotely or autonomously control vehicles that are operating underwater, on the surface, or in space. In [5], several case studies for underwater vehicles are presented, such as heading autopilot system or path-following control among others. In [6], they consider a hybrid control architecture for AUVs with a hybrid behavior-based scheme using reinforcement learning. In addition, [7] presents the design and implementation of a mission control system for an AUV.

Obstacle avoidance is one of the main capabilities for an autonomous vehicle, and it includes a control objective subject to non-intersection or non-collision position constraints. In [8], some obstacle avoidance approaches were presented in both simulation and field-test environments using a multibeam imaging sonar. However, in our study, the AUV is tested in a water tank, where it restricts the possibilities for obstacle avoidance tests. Hence, the obstacle avoidance has been replaced by wall detection, which is based on line extraction algorithms. The line extraction algorithms have been used by many researchers in numerous applications. The Split-and-merge algorithm is probably the most popular line extraction algorithm [9]. Additionally, split-and-merge has the best performance in real-time applications due to its superior speed [10]. Ransac (Random Sample Consensus) is an algorithm for the robust fitting of models in the presence of data outliers [11].

This paper includes both simple and advanced GNC algorithms for simulation and field-test for an AUV in a water tank. The validation for the model-based architecture is also studied, comparing the performance of the Girona500 AUV in both simulation and field-test environments. The simple algorithm includes three kinds of PID controllers (velocity, velocity-position, and position), and the advanced one combines path-following with a wall detection algorithm, acquiring and

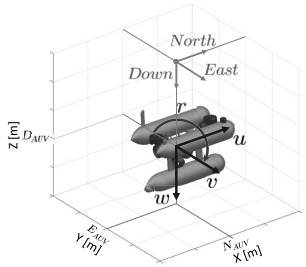
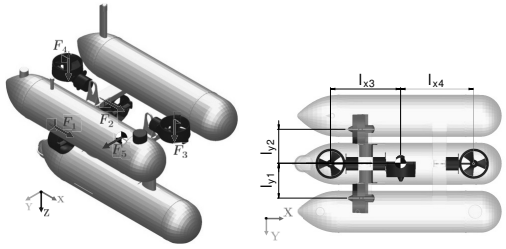


Fig. 1. Simplified model of the considered vehicle using the North-East-Down (NED) coordinate system. AUV motion is described by surge u (linear longitudinal (front/back) motion), sway v (linear transverse (side-to-side) motion), heave w (linear vertical (up/down) motion), and angular velocity r (turning rotation about its z-axis).



(a) (b)

Fig. 2. Five-thruster configuration in the AUV: (a) Thrust forces with their direction for each thruster, (b) Distances from each thruster to the center of mass of the AUV.

processing data from a mechanical imaging sonar.

II. AUV SIMULATION MODELING

A. Overview of Girona500 AUV

The AUV utilized in this study is the Girona500 [12], which provides high configurability for different scientific instrumentation, and allows the use of the Robot Operating System (ROS) as a framework [13]. The instrumentation and sensors used in this study are Doppler Velocity Log (DVL), depth sensor, and AHRS (Attitude and Heading Reference System) for the navigation of the Girona500, and a mechanical imaging sonar (Tritech Micron [14]) for data acquisition and processing from an underwater sensor for the wall detection.

Fig. 1 shows a simplified model of the AUV. This AUV uses a five-thruster configuration to provide thrust forces when moving in the surge, sway, heave motions, or performing turns. Also, the position and velocities of the AUV are illustrated in Fig. 1. The general motion of the AUV in 6DoF is modeled by using the North-East-Down (NED) local coordinate system. AUV position and velocities are considered with the following vectors

$$\eta = [N, E, D, \phi, \theta, \psi]^\top, \upsilon = [u, v, w, p, q, r]^\top. \quad (1)$$

where N, E, D denote the NED positions in Earth-fixed coordinates, ϕ, θ, ψ are the Euler angles, u, v, w are the Body-fixed linear velocities, and p, q, r are the Body-fixed angular velocities [5].

B. Vehicle dynamics

Before implementing the GNC algorithms, the design and modeling of the AUV have been studied using a theoretical 6DoF dynamic model [15]. This dynamic model is based on nonlinear equations of motion, and its parameters are obtained from the Girona500. For an underwater vehicle, the hydrodynamic forces and moments will be due to added mass and damping, while the hydrostatic forces and moments are due to weight and buoyancy. This suggests that

$$M\dot{\nu} + C(\nu)\nu + D(\nu)\nu + g(\eta) = \tau. \quad (2)$$

where M is the mass matrix ($M = M_{addedmass} + M_{rigidbody}$), C presents the centripetal and Coriolis terms, D presents the damping coefficients, and g presents the vertical forces, which consist of buoyancy and gravity in body frame.

The actuator forces and moments relate to the control forces and moments by

$$\tau = Tf. \quad (3)$$

where T is thrust configuration matrix, and f is the control forces and moments vector. The thrust configuration matrix T for the five-thruster configuration considered in this study is defined as

$$T = \begin{bmatrix} 1.0 & 1.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.0 & 1.0 \\ 0.0 & 0.0 & 1.0 & 1.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & -l_{x3} & +l_{x4} & 0.0 \\ -l_{y1} & +l_{y2} & 0.0 & 0.0 & 0.0 \end{bmatrix}. \quad (4)$$

where $l_n = [l_{xn}, l_{yn}, l_{zn}]$ is the distance from the thruster n to the centre of mass of the AUV. In our case, thrusters are located as is shown in Fig. 2(b), where thrusters 1 and 2 effects in surge and yawing, thrusters 3 and 4 in heave and rolling, and thruster 5 in sway. As thruster 5 is located at the center of mass, it does not produce any rotational motion.

C. Simulation environment

The dynamic model of the AUV is created in MATLAB-Simulink using Simscape [16], and it allows designing and analyzing the modeled control systems by using Simulink Control Design [17]. In addition, a PID Tuner tool for a linearized point of the dynamic model is used to obtain the desired PID controller parameters according to time-response and transient behavior. Hence, an appropriate GNC can be designed and analyzed in the same model using a variety of control systems. The block diagram of the AUV simulation modeling is shown in Fig. 3, where the blocks for position and velocity controller contains the respective PID controller for each motion. The 6DoF dynamic model is included in the simulation modeling using the Simscape library in MATLAB-Simulink. Fig. 4 shows the block diagram of the dynamic model of the AUV.

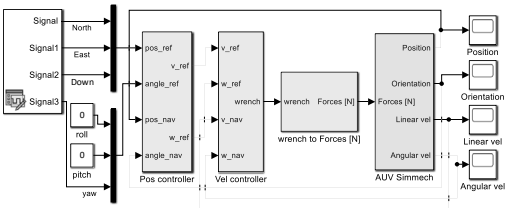


Fig. 3. Block diagram of the AUV simulation modeling for the cascade position and velocity controller. Input is generated by a signal generator for the four controlled positions (North, East, Depth, and Yaw).

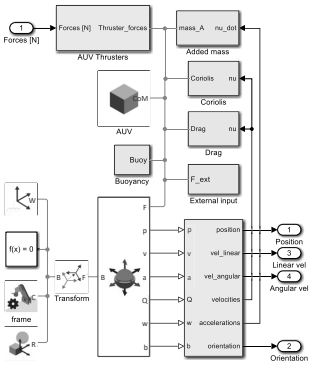


Fig. 4. Block diagram of the dynamic AUV simulation modeling. Green blocks are related to Eq. 2, and blue blocks are related to the control algorithms, including the sensors for position, velocity, and acceleration of the AUV.

III. Simple and advanced GNC systems

The AUV is controlled in all translational motions (surge, sway, and heave), and in yaw rotational motion, including a separate PID controller for each movement. The control and navigation algorithms for these movements involve two different parts: (i) simple control algorithms (velocity, velocity-position, and position controllers), and (ii) advanced control algorithms involving data acquisition and processing from an underwater sensor.

A. Simple GNC algorithms

The simple GNC algorithms include a velocity control using a unique PID controller, position control using cascade controller (position and velocity PID controllers), and position control using a unique PID controller. PID parameters for all control algorithms are selected from the dynamic model in MATLAB/Simulink for certain time response and transient behavior specifications. The AUV's pose and velocity are obtained from the default COLA2 navigation module [18], which uses the NED local coordinate system. This module collects necessary information from the AUV navigation sensors, and merge these sensors to obtain accurate position and estimated velocity.

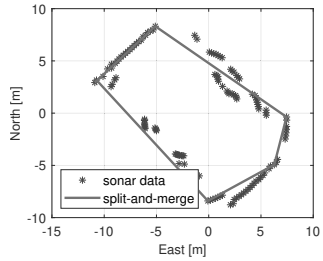


Fig. 5. Wall detection using mechanical imaging sonar and split-and-merge line extraction algorithm. The error and misalignment in the parallel walls are produced by the echos of the imaging sonar in the water tank.

For the unique PID controllers, the controller receives the current vehicle's pose (or velocity) from the COLA2 navigation module, and the desired pose (or velocity) from the GNC algorithm. Then, the controller computes the necessary force and torque in each motion to achieve the desired pose (or velocity).

For the position control using cascade controller (position and velocity PID controllers), first, the position controller $C_{pos}(s)$ receives the current vehicle's pose η from the COLA2 navigation module, and the desired pose request η_{ref} from the path definition message. The position controller's output is the necessary velocities to achieve the requested position. Then, the velocity controller $C_{vel}(s)$ combines a PID with an open-loop model-based controller. The desired velocity υ_{ref} is obtained from the position controller, while the current vehicle's velocity υ is given by the COLA2 navigation module. The velocity controller computes the necessary force and torque in each motion τ to achieve the desired velocity.

Once that the necessary force and torque in each motion τ has been calculated, to obtain the setpoint that each thruster has to generate f, the received forces and torques are combined using the pseudoinverse T^{+} of the thrust configuration matrix T [5]. This relation is defined as

$$f = T^{+}\tau. \tag{5}$$

B. Advanced GNC algorithms

The advanced GNC algorithm includes wall detection using the data from the mechanical imaging sonar, which is installed on top of the AUV. Hence, this advanced algorithm involves also the data acquisition and processing from an underwater sensor (imaging sonar), whose implementation is tested in the wall detection algorithm. This algorithm is used to detect the waypoints with a constant distance from the wall for the path following algorithm. Furthermore, the wall detection uses the sonar point cloud and the split-and-merge algorithm. Fig. 5 shows the results of the split-and-merge algorithm in the water tank.

Once that all the water tank walls have been identified, one of them is selected to perform the advanced GNC algorithm. From the line expression of the identified wall, two waypoints

Algorithm 1 Split-and-merge (adapted from [10])

1: Initial: set s_1 consists of N points (obtained from imaging sonar). Put s_1 in a list L.
2: Fit a line to the next set s_i in L. This line between $P_1 = (x_1, y_1)$ and $P_2 = (x_2, y_2)$ is defined as

$$ax + by + c = 0. \tag{6}$$

where $a = y_1 - y_2$, $b = x_2 - x_1$, and $c = x_1 y_2 - x_2 y_1$
3: Detect point P with maximum distance d_p to the line. For all points (x, y) in the set, d_p is defined as

$$d_p = max\left(\frac{|-ax - by - c|}{\sqrt{a^2 + b^2}}\right). \tag{7}$$

4: If d_p is less than a threshold, continue (go to step 2).
5: Otherwise, split s_i at P into s_{i1} and s_{i2}, replace s_i in L by s_{i1} and s_{i2}, continue (go to 2).
6: When all sets (segments) in L have been checked, merge collinear segments according to the given thresholds.

are selected from a constant distance from the wall and separated by 2 meters. The path-following for these two waypoints is based on the position control using a unique PID controller, combined with a switching mechanism. This switching mechanism is defined as a sphere of acceptance [19], which selects the next waypoint as a lookahead point if the position of the AUV lies within a sphere with radius R around $(N_{k+1}, E_{k+1}, D_{k+1})$. The sphere of acceptance is defined as

$$[N_{k+1} - N(t)]^2 + [E_{k+1} - E(t)]^2 + [D_{k+1} - D(t)]^2 \leq R_{k+1}^2. \tag{8}$$

where, if the time AUV position $(N(t), E(t), D(t))$ satisfies Eq. 8, the next waypoint $(N_{k+1}, E_{k+1}, D_{k+1})$ needs to be selected. Radius R is equal to two AUV lengths ($R = 2L_{AUV}$).

C. Modular system for the path planning algorithm

This advanced model uses two different modules for implementation: wall recognition with path definition, and control. Each of these two modules runs a separate ROS node in the system. This approach has been previously studied in [20], implementing a path-following algorithm in an Unmanned Surface Vehicle (USV) with a straight-line.

Fig. 6 illustrates this modular architecture with all ROS topics involved. Also, this schematic defines the necessary subscribers and publishers of the system. The imaging sonar module processes the data acquired from the mechanical imaging sonar and the COLA2 navigation module. This allows the GNC algorithm to detect the position of the walls around the AUV and create the necessary waypoints for the path-following algorithm. This module also checks the position of the AUV to select the next waypoint to reach in the trajectory. The control module includes the GNC algorithm, which generates the required thruster setpoints (based on the control forces and moments). The Girona500 navigation system receives these thruster setpoints reaching the desired AUV position.

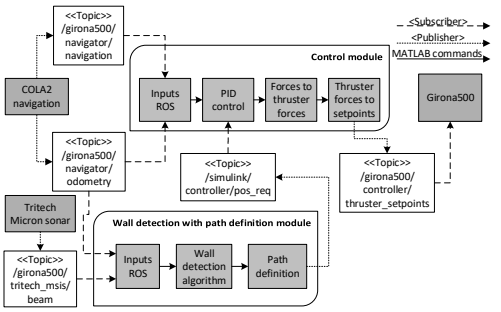


Fig. 6. Schematic of the modular GNC system. The two models are the modules included in the AUV platform, which are run in separate ROS nodes.

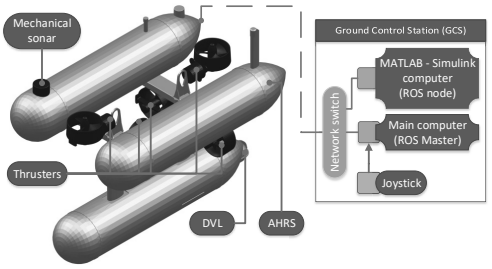


Fig. 7. System overview of the AUV: High-level (ROS computers), and low-level control (joystick, thrusters, DVL, AHRS, and mechanical imaging sonar).

IV. Experimental validation

A. System implementation

The AUV incorporates multiple mechatronic systems to obtain situational awareness, perform the path-following task, and control the motions (surge, sway, heave, and yawing) of the vehicle. All these mechatronic systems used in this study are shown in Fig. 7. The system can be described as high-level and low-level control. High-level control contains the ROS computers, and the low-level control involves the sensors and actuators of the AUV including mechanical imaging sonar, DVL, AHRS, depth sensor, thrusters, and joystick controller.

The AUV used in this study is mainly controlled by a Linux computer that runs MATLAB/Simulink (ROS node), which is connected to the rest of the instrumentation by a network switch via Ethernet. This MATLAB/Simulink Linux computer includes the different GNC algorithms using a Standalone ROS-node that permits rapid prototyping while testing [21]. This Standalone ROS-node provides a solution for the time-consuming process of C++ programming, and it is used for rapid-prototyping of the PID parameters of the motion controllers in the GNC algorithms. The ROS master runs the COLA2 navigation system to obtain the necessary data from the AUV to perform the autonomous tasks and to send the thruster setpoints commands from the GNC algorithm.

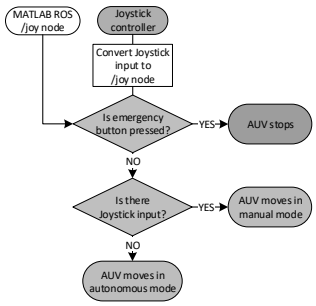
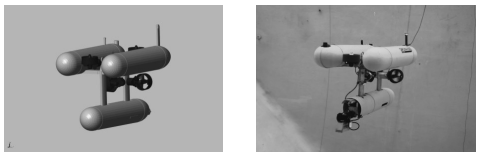


Fig. 8. Stateflow diagram for priority control level. A Joystick Controller is used as a higher priority level. If the joystick controller is not used during the test, the AUV moves according to the GNC algorithm.



(a) (b)

Fig. 9. Control Scenario I: Girona500 in (a) the simulation and in (b) the water tank.

The priority control level of this study is shown in Fig. 8, where the joystick controller is defined as the highest priority because of safety, being able to stop the AUV movement at any moment.

B. Experimental results: Control Scenario I

The first test is both simulated and implemented, and it is related to the simple GNC algorithm described in Subsection III-A. Fig. 10 shows the model-validation results for the unique PID controller, one of the simple GNC algorithms developed in this study. The PID controller uses the same PID control parameters for both simulation and field-test. As can be seen in these plots, the settling time is similar in both simulation and field-test results. However, the overshoot in the field-test scenario is higher than the simulation case. This difference proves that the simulation parameters are not completely accurate, and further study will be needed in this situation. It will require a more exact simulation modeling with a better estimation of the dynamic model parameters.

C. Experimental results: Control Scenario II

The implementation for the Control Scenario II is related to the advanced GNC algorithm described in Subsection III-B. Fig. 11 outlines the AUV trajectory from this Control Scenario in 2D and 3D plots, where the AUV follows the predefined path of two waypoints obtained from the wall detection algorithm. In addition, Fig. 12 shows the input control values (North, East, Down, and Yaw) obtained from

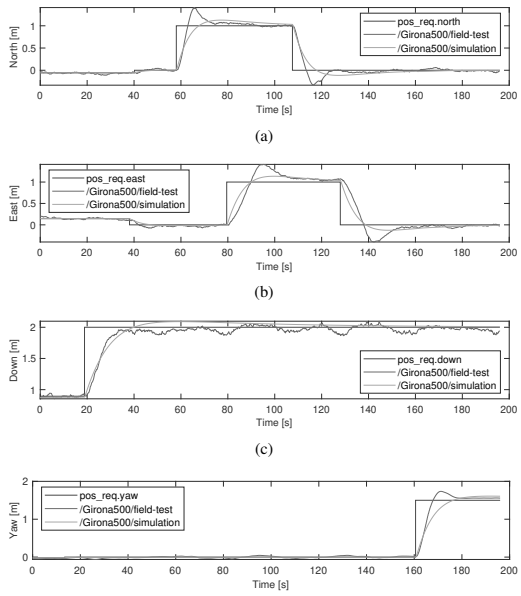


Fig. 10. Control Scenario I: Model-validation results for the unique PID controller. The plots show the comparison between the simulation and field-test controlled position variables for (a) North, (b) East, (c) Down, and (d) Yaw.

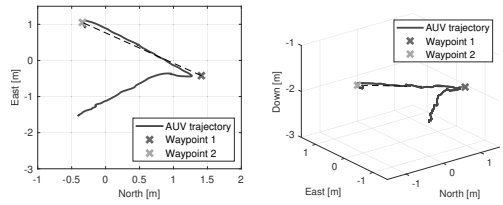


Fig. 11. Control Scenario II: Two waypoints were selected at a specified distance from the water tank walls to test the path-following in the advanced control algorithm: (a) AUV trajectory in 2D, (b) AUV trajectory in 3D.

the GNC algorithm. This Control Scenario has the Yaw DoF with higher priority than the other DoFs. This Yaw angle is equal to the slope of the two predefined waypoints, allowing a continuous path operation. Then, once this angle has been reached, the AUV moves to the selected waypoints.

V. Conclusion and Future Work

This article was concerned with the design, modeling, and implementation of a model-based GNC architecture for an AUV. This GNC architecture was verified using AUV simulation modeling, which was based on a theoretical 6DoF dynamic model with the parameters from the Girona500 AUV.

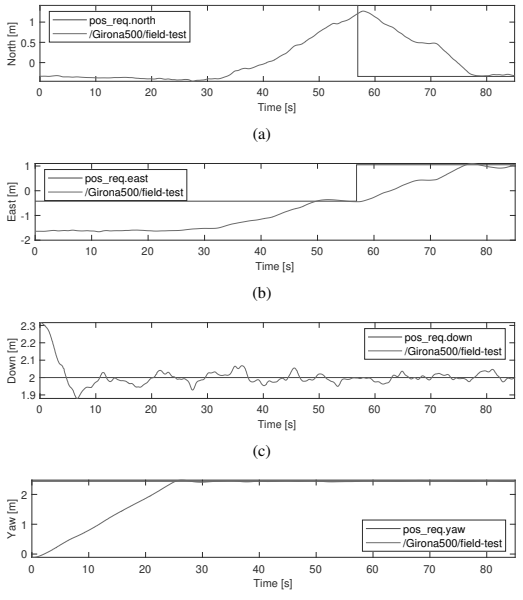


Fig. 12. Control Scenario II: Comparison between values from the GNC algorithm with the field-test data from the AUV in (a) North, (b) East, (c) Down, and (d) Yaw.

Once the simulation modeling was validated, simple and advanced GNC algorithms were developed and implemented in the AUV. The simple GNC algorithm contained three kinds of PID controllers (velocity, velocity-position, and position), while the advanced GNC algorithm was enabled with path-following. This path-following used the waypoints from a wall detection algorithm using a mechanical imaging sonar. Finally, the experimental results validated the designed GNC architecture, with control scenarios for each simple and advanced algorithms.

Future work will include the comparison of this architecture with other architectures with different controllers that are not PID-based, and the complete obstacle avoidance capabilities will be tested in an open environment. Furthermore, an improvement of the current AUV simulation modeling using system identification techniques will be studied, as well as design optimization methods for parameter identification.

ACKNOWLEDGMENT

This paper is based on the project MOCAV-G500 from EUMarineRobots that has received funding from the European Union's Horizon 2020 (grant agreement No 731103), with the collaboration of the aColor project, which is funded by Technology Industries of Finland Centennial and Jane & Aatos Erkko Foundations.

REFERENCES

[1] E. Galceran, R. Campos, N. Palomeras, D. Ribas, M. Carreras, and P. Ridao, "Coverage path planning with real-time replanning and surface reconstruction for inspection of three-dimensional underwater structures using autonomous underwater vehicles," *Journal of Field Robotics*, vol. 32, no. 7, pp. 952–983, 2015.

[2] R. Simoni, P. R. Rodríguez, P. Cieślak, and D. Youakim, "A novel approach to obstacle avoidance for an i-auv," in *2018 IEEE/OES Autonomous Underwater Vehicle Workshop (AUV)*. IEEE, 2018, pp. 1–6.

[3] E. Simetti, G. Casalino, S. Torelli, A. Sperinde, and A. Turetta, "Floating underwater manipulation: Developed control methodology and experimental validation within the trident project," *Journal of Field Robotics*, vol. 31, no. 3, pp. 364–385, 2014.

[4] D. Ribas, P. Ridao, and J. Neira, *Underwater SLAM for structured environments using an imaging sonar*. Springer, 2010, vol. 65.

[5] T. I. Fossen, *Handbook of marine craft hydrodynamics and motion control*. John Wiley & Sons, 2011.

[6] M. Carreras, J. Yuh, J. Batlle, and P. Ridao, "A behavior-based scheme using reinforcement learning for autonomous underwater vehicles," *IEEE Journal of Oceanic Engineering*, vol. 30, no. 2, pp. 416–427, 2005.

[7] M. Carreras, N. Palomeras, P. Ridao, and D. Ribas, "Design of a mission control system for an auv," *International Journal of Control*, vol. 80, no. 7, pp. 993–1007, 2007.

[8] B. Braginsky and H. Guterman, "Obstacle avoidance approaches for autonomous underwater vehicle: Simulation and experimental results," *IEEE Journal of oceanic engineering*, vol. 41, no. 4, pp. 882–892, 2016.

[9] T. Pavlidis and S. L. Horowitz, "Segmentation of plane curves," *IEEE transactions on Computers*, vol. 100, no. 8, pp. 860–870, 1974.

[10] V. Nguyen, S. Gächter, A. Martinelli, N. Tomatis, and R. Siegwart, "A comparison of line extraction algorithms using 2d range data for indoor mobile robotics," *Autonomous Robots*, vol. 23, no. 2, pp. 97–111, 2007.

[11] M. A. Fischler and R. C. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Commun. ACM*, vol. 24, no. 6, p. 381–395, Jun. 1981.

[12] D. Ribas, N. Palomeras, P. Ridao, M. Carreras, and A. Mallios, "Girona 500 auv: From survey to intervention," *IEEE/ASME Transactions on mechatronics*, vol. 17, no. 1, pp. 46–53, 2011.

[13] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng, "Ros: an open-source robot operating system," in *ICRA workshop on open source software*, vol. 3, no. 3.2. Kobe, Japan, 2009, p. 5.

[14] *Micron sonar - Product manual*, Tritech International Ltd, 0650-SOM-00003, Issue: 02.

[15] T. Fossen and A. Ross, "Nonlinear modelling, identification and control of uuvs," *IEE Control Engineering Series*, vol. 69, p. 13, 2006.

[16] *Simscape User's Guide*, The MathWorks, Inc., Natick, MA, 2007, release 2019a.

[17] *Simulink Control Design User's Guide*, The MathWorks, Inc., Natick, MA, 2004, release 2019a.

[18] Iqua Robotics. (2019) Cola2 wiki. [Online]. Available: https://bitbucket.org/iquarobotics/cola2_wiki/src/master/README.md

[19] A. J. Healey and D. Lienard, "Multivariable sliding mode control for autonomous diving and steering of unmanned underwater vehicles," *IEEE journal of Oceanic Engineering*, vol. 18, no. 3, pp. 327–339, 1993.

[20] J. Villa, J. Aaltonen, and K. T. Koskinen, "Model-based path planning and obstacle avoidance architecture for a twin jet unmanned surface vessel," in *2019 Third IEEE International Conference on Robotic Computing (IRC)*. IEEE, 2019, pp. 427–428.

[21] *ROS Toolbox User's Guide*, The MathWorks, Inc., Natick, MA, 2019, release 2019a.

PUBLICATION III

Path-Following with LiDAR-based Obstacle Avoidance of an Unmanned Surface Vehicle in Harbor Conditions

Jose Villa, Jussi Aaltonen, Kari T. Koskinen

IEEE/ASME Transactions on Mechatronics 25, no. 4 (2020): 1812-1820.
DOI: 10.1109/TMECH.2020.2997970

Path-Following with LiDAR-based Obstacle Avoidance of an Unmanned Surface Vehicle in Harbor Conditions

Jose Villa, Jussi Aaltonen, and Kari T. Koskinen

***Abstract*—This article studies the design, modeling, and implementation challenges of a path-following with obstacle avoidance algorithms as Guidance, Navigation, and Control (GNC) architecture of an Unmanned Surface Vehicle (USV) in harbor conditions. First, an effective mathematical model is developed based on System Identification (SI), validating the USV model with field-test data. Then, a guidance system is addressed based on a Line-Of-Sight (LOS) algorithm, which uses a LiDAR as the main perception sensor for the obstacle avoidance algorithm. The GNC architecture uses a modular approach, including obstacle detection, path-following, and control in the USV platform. Finally, an implementation challenge in two control scenarios, simulation and field-test, is addressed to validate the designed GNC architecture.**

***Index Terms*—Path-following, obstacle avoidance, system identification, model-validation, USV.**

I. Introduction

DIFFERENT aspects in society, such as occupational safety and security, as well as longer operation times, have led to a demand for research and development of innovative autonomous systems. These autonomous systems include Unmanned Surface Vehicles (USVs) and Autonomous Underwater Vehicles (AUVs) as main offshore vehicles. USVs can be studied for numerous potential applications in an advantageous way, such as scientific research, environmental missions, or ocean resource exploration.

To accomplish all these offshore applications, the availability of an adequately accurate USV model is imperative for simulation study purposes, controller design, and development. USV models are commonly reduced-order for horizontal plane control (surge, sway, and yaw motions). These models have been used in numerous studies, such as the 3 Degrees-Of-Freedom (DOFs) horizontal plane models for maneuvering based on the rigid-body-kinetics [1], or the model representation of Nomoto [2] for heading autopilot among others. Furthermore, System Identification (SI) can be included to obtain an accurate model for simulation studies using field-test data [3]. SI using the Nomoto model, SI based on Particle-Swarm-Optimization (PSO), and SI for the 3-DOFs ship maneuvering model are presented for USV approaches in [4], [5], and [6], respectively. Moreover, [7] proposes a nonlinear modeling scheme for a waterjet propulsion USV system. However, both the Nomoto model and [7] use the rudder angle as model input.

For controller design and development, safe and efficient control of USVs depends heavily on proper Guidance, Navigation, and Control (GNC) systems with sensing, state estimation, and situational awareness capabilities. A path-following is adopted as a guidance system in this work, as it is easier to implement than trajectory tracking, and is closer to practical engineering. For path-following in offshore operations, most of the studies have been done in a free obstacle path scheme using Line-Of-Sight (LOS) algorithm [8] or a guidance-based algorithm [9]. Recent progress on path-following is focused more on dealing with external disturbances to improve the control performance [10].

To perform missions in real-world environments, USVs are required to have the ability to detect obstacles, recognize and track targets, and map environments. To obtain situational awareness of the USV, passive (e.g., stereo cameras) and active (LiDAR or radar) perception methods have been used in numerous studies. However, the majority of the obstacle detection techniques rely on depth measurements, in which LiDAR sensors are the most robust method of obtaining depth data. In [11], a 3-D scanning LiDAR performance was focused on the marine environment for a USV. However, 2-D or 3-D representation of LiDAR can suffer the clutter phenomena of a marine environment [12]. In this paper, SICK MRS1000 LiDAR [13] solves this effect, as it has 4 spread-out scan planes and a multi-echo analysis that avoids the noise produced by fog, rain, or dust. Also, the working range of this device is from 0.2 to 64 meters with a 275° aperture angle, being completely suitable for small USVs in harbor operations.

An approach to combine both path-following and obstacle avoidance capabilities can be the use of safety boundary boxes around a static or moving obstacle. In [14], the use of safety boundary boxes was studied for collision avoidance, where a corresponding collision boundary box is associated with each obstacle. The goal of this study is to find the optimal path while avoiding any collision boundary boxes. In addition, [15] includes a multi-layer obstacle avoidance based on a single LiDAR and presents an efficient solution to USV path planning in the case of sensor errors and collision risks, defining a safety box for obstacle recognition.

Besides to the safe and efficient control, USVs operating in populated waterways may require compliance with existing rules. These rules can be the collision regulations defined by the convention on the international regulations for preventing

The authors are with the Mechatronics Research Group (MRG), Tampere University (TAU), 33720, Tampere, Finland (e-mail: jose.villa@tuni.fi; jussi.aaltonen@tuni.fi, kari.koskinen@tuni.fi).

collisions at sea (COLREGs) [16]. Regarding COLREGs in USVs, [17] reports preliminary research results of a novel automatic obstacle avoidance approach. Furthermore, [18] studies an under-actuated USV for collision avoidance, ensuring path-following while abiding by the COLREGs. However, neither of these studies have been implemented in field-tests.

Once the GNC algorithm has been tested in a simulation environment, it can be implemented in a field-test. This implementation uses Robot Operating System (ROS), which is a flexible framework for writing robot software [19]. ROS provides the necessary tools to access sensors' data, process it, and generate a response for the vehicle actuators. To link all USV sensors and actuators in the same ROS network, rosserial provides a tool for connecting embedded computers with the rest of the system [20]. As an example of this system connectivity, [21] designs and implements a low-cost, high performance, and generic multi-layer ROS-based architecture for autonomous systems. In addition, there are several examples of state-of-the-art autonomous vehicles using ROS, such as self-driving taxis [22].

In this study, the USV model is obtained with a different approach than using the rudder angle as model input. The same model combines the waterjet and USV dynamics by using the joystick commands as model input. For obstacle avoidance, a safety boundary box is selected, providing fast decision-making capabilities due to its simplicity, low data transfer, and modular approach. In addition, a LOS algorithm is implemented without compensation of environmental elements, to focus further on the obstacle avoidance. This obstacle avoidance for a path-following algorithm uses a modular ROS architecture to provide a simple, computationally cheap, and easy implementation. Hence, obstacle avoidance capabilities in field-tests are the main focus of this research, to allow the application of COLREGs in the future.

In this work, a model-based GNC architecture for a USV is proposed for path-following with obstacle avoidance using a LiDAR as a perception method. In Section II, the USV modeling and simulation are presented using SI as the tool to define the maneuvering model. Then, in Section III, the control methodology design is included using the LOS-based guidance system for control. Obstacle avoidance capabilities are involved in defining a safety boundary box around the detected object and using its corners for the new waypoints of the path-following. Finally, in Section IV, the implementation of a GNC architecture is described as modular and multilayer, allowing the fast check for the optimum operability of the vehicle capabilities. Control scenarios in both simulation and field-test are shown to validate the proposed GNC architecture.

II. USV MODELING AND SIMULATION

A. Overview of under-actuated USV

The USV used in this study is an aluminum hull with thrust vectoring waterjet propulsion. The USV has optimal maneuverability using the twin waterjet configuration that facilitates the movement in all directions without bow and stern thrusters (see Fig. 1). The motion of the USV is simplified from six to three DOFs. These 3-DOFs are surge, sway, and yaw motions, while ignoring roll, pitch, and heave.

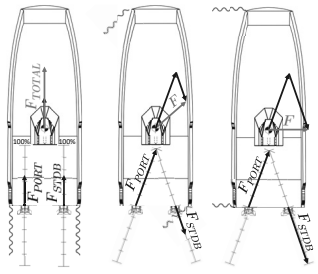


Fig. 1. Twin waterjet USV maneuvering.

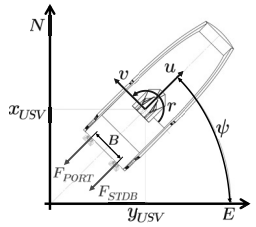


Fig. 2. Simplified model of the considered vehicle.

Fig. 2 shows a simplified model of the twin waterjet propelled USV. The waterjet propelled USV uses the Port and Starboard (STDB) waterjets to provide thrust forces when moving either forward, backward, sideways, or performing turns. Also, Fig. 2 illustrates the position and orientation of the USV. Due to the use of GPS-Compass as the navigation system, a coordinate transformation is applied to obtain the absolute position of the USV in the planar coordinate system. This transformation is between World Geodetic System 84 (WGS84), which provides longitude and latitude [l, μ] of the USV, and ETRS-TM35FIN [23], which shows the North-East-Down (NED) (x_{USV}, y_{USV}) position. The USV heading ψ is described using the attitude (Euler angles). Furthermore, the body-fixed reference frame is used for relative positioning, with linear [u, v] and angular [r] velocities.

B. Vehicle modeling

The development of an effective maneuvering model will facilitate the GNC algorithms design and simulation. The guidance system is defined as a path-following control, where the USV moves forward with reference speed u at the same time as minimizes the cross-track error e to the predefined path. Several heading controllers for marine crafts are based on the model representation of Nomoto [2]. The Nomoto autopilot model can be derived from the linearized maneuvering model of the USV, and Nomoto's second-order model refers to

$$\frac{r}{\delta}(s) = \frac{K_p(1+T_z s)}{(1+T_{p1}s)(1+T_{p2}s)}. \tag{1}$$

where r is the angular velocity, and δ is the rudder angle.

The model representation of Nomoto has the main advantage of its simplicity. Its parameters can be defined directly

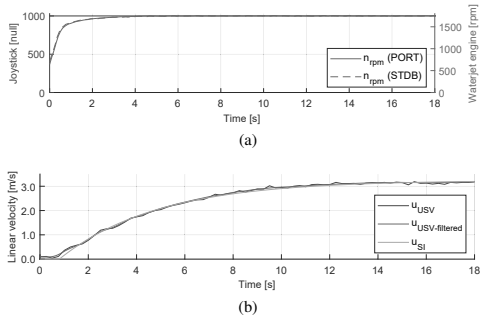


Fig. 3. USV in forward motion (surge): (a) Waterjet engine response for a constant Joy_{surge} input, (b) Model-validation using SI for surge velocity.

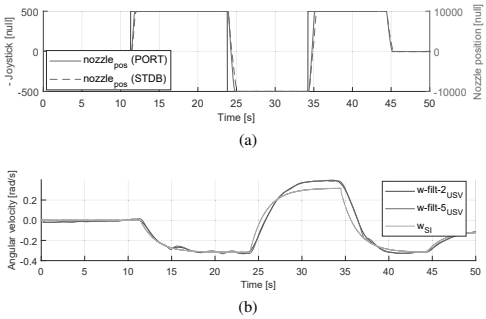


Fig. 4. USV in yaw motion: (a) $nozzle_{pos}$ response for a zigzag Joy_{yaw} input, (b) Model-validation using SI for angular velocity.

from the field-testing data, and thus, the hydrodynamic derivatives do not need to be computed explicitly.

C. Model-validation using System Identification

Propulsive thrust in the water jet propulsion unit is created by the reaction force, which is caused by the kinetic energy of the mass flow generated by the propulsion unit impeller. The nozzle position $nozzle_{pos}$ changes the direction of the jet flow coming out of the nozzle, which creates the force needed for turning. Hence, the total thrust force (F_{Total} in Fig. 1) is a combination of the engine rpm of the waterjet n_{rpm} and $nozzle_{pos}$. n_{rpm} is directly obtained from the waterjet engine, and $nozzle_{pos}$ is a variable from -10.000 to 10.000, with 0 as the neutral position (forward motion). Fig. 3a and 4a show the settling time effect for joystick input in both n_{rpm} and $nozzle_{pos}$, which are related to the waterjet dynamics. A combination of a 1D lookup table with second-order transfer functions is proposed to simplify the USV modeling, including the effect of both waterjet and USV dynamics. The parameters for the 1D Lookup table are obtained from field-testing data for the forward and right-turn motions and are presented in Table I. These parameters are used to map the input to output values, approximating a mathematical function.

The mathematical model of the USV includes a constant surge velocity with a variable heading of the USV. As it is presented in [4], the behavior obtained with MATLAB SI tool [24] is close to the Least Squares Support Vector Machines (LS-SVM) approach. Thus, the MATLAB SI tool has been chosen for simplicity to obtain both surge and yaw models.

TABLE I
1D LOOKUP TABLE PARAMETERS

Joy_{surge}	400	500	600	700	800	900	1000
n_{rpm}	690	920	1110	1300	1480	1650	1820
Joy_{yaw}	0	50	150	200	250	300	400
$nozzle_{pos}$	0	1175	3500	4665	5830	7000	9325

TABLE II
SURGE AND YAW MOTION PARAMETERS

Motion	T_z	T_{p1}	T_{p2}	K_p	T_d
Surge	0.17563	4.08900	0.17299	2.930×10^{-3}	0.8
Yaw	0.09835	1.81108	0.00144	-3.177×10^{-5}	0.0

After studying several SI models, the surge motion is defined in (2) as a transfer function with one zero (T_z), two poles (T_{p1}, T_{p2}), a process gain (K_p), and an Input/Output delay (T_d). The engine rpm of the waterjet n_{rpm} is declared as input, while surge velocity u is the output of the transfer function. Fig. 3b shows both SI and field-tests (raw and filtered USV linear velocity) step response for a constant joystick Joy_{surge} input value.

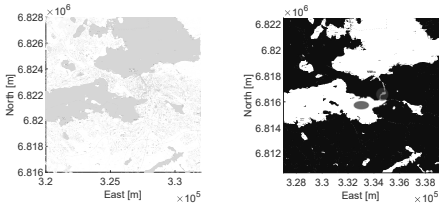
$$\frac{u}{n_{rpm}}(s) = \exp(-T_d s)\frac{K_p(1+T_z s)}{(1+T_{p1}s)(1+T_{p2}s)}. \qquad (2)$$

The yaw motion is defined using the same Nomoto's second-order model declared in (1). However, the model input is the nozzle position of the waterjet $nozzle_{pos}$ instead of the original rudder angle δ used in the Nomoto's model. The nozzle position generates different angular velocity for a constant engine rpm. The output is the angular velocity r for yaw motion, obtaining the yaw angle ψ from its integration. Fig. 4b shows both SI and field-tests (two filtered plots from the yaw angular velocity) step response for a constant Joy_{surge} input value, and a zigzag in Joy_{yaw}. The variance between the left and right turns is produced by the misalignment of the center of mass and center of buoyancy in the USV (inclination of 1.5° to the right).

Table II includes the identified model parameters for the transfer functions of surge and yaw motions included in the mathematical model. The variables u and r are obtained according to their 1D Lookup tables and transfer functions (from (2) and (1) respectively). Thus, there is a different set of dynamics comparing to a USV with a propeller and rudder, a different set of parameters, but the approach still works.

D. Simulation environment

A simulation experiment is essential to verify whether the GNC algorithm is valid in the USV workspace. The map of the control scenario must be processed, requiring the definition of two separate matrices. The first matrix includes RGB data per pixel, and the second one contains the planar coordinate



(a) (b) BW map

Fig. 5. Map processing: (a) RGB map with ETRS-TM35FIN planar coordinate system, (b) BW map after map processing where red area refers to Control Scenario I and blue area to Control Scenario II in Section IV.

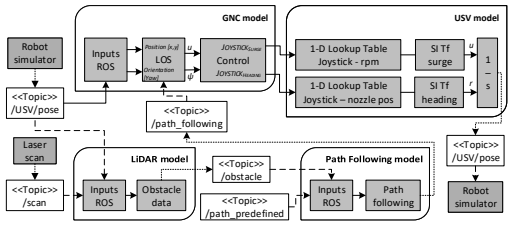


Fig. 6. Block-diagram representation of the simulated USV model.

system, which is added by reading the world file. Then, the RBG image, shown in Fig. 5a, is converted to Black and White (BW), and any noisy points are removed using standard image processing software (see Fig. 5b). Finally, the BW image is converted into a binary occupancy grid in MATLAB software, which is also the simulation environment for this study. A binary occupancy grid uses true values to represent the occupied workspace (black), such as obstacles, and false values for the free workspace (white). This map is used as a simulation scenario for the GNC system, allowing the inclusion of external objects in a specific position, and the location of the USV at a certain point on the map. In addition, this map provides the possibility for future path planning research, in which a binary occupancy grid is essential.

The GNC controller for a simulated USV is created and computes the control commands to follow the desired path, which is a set of waypoints defined explicitly. The simulated vehicle has kinematic equations for the motion of an underactuated USV, obtained from the SI process mentioned in this Section. Furthermore, situational awareness capabilities are included using a laser scan object. The laser scan message is processed to extract scan ranges and angles, and compute these variables to obtain the position of the detected obstacle. This is a similar approach to the one obtained from the USV LiDAR. The controller receives the vehicle pose and laser scan data from the simulated vehicle and sends joystick commands to drive the vehicle on the given path. The schematic of the simulated USV model is shown in Fig. 6. The main goal of the mathematical model of the USV is to test the obstacle avoidance capabilities. Hence, the hypothesis of independence between surge and yaw dynamics is acceptable even though it does not include drift or environmental disturbances.

III. GNC SYSTEM WITH LINE-OF-SIGHT BASED MODEL

A. Line-of-sight guidance system

A path-following algorithm aims to reach every waypoint of a predefined path independent of time. A commonly used method for path-following is the named LOS guidance, which is adopted as a reference trajectory for the USV in this study. A LOS vector from the surface vehicle to the next waypoint or a point on the path between two waypoints can be used for heading control, similar to [1].

For lookahead-based steering, the course angle is separated into two parts, defined as

$$\chi_d(e) = \chi_p + \chi_r(e). \tag{3}$$

where $\chi_p = \alpha_k$ is the path-tangential angle defined in (4), while χ_r is a velocity-path relative angle, which ensures that the velocity has the direction towards a point on the path that is in a lookahead distance $\Delta(t) > 0$ along of the direct projection [25].

$$\alpha_k = \text{atan2}(y_{k+1} - y_k, x_{k+1} - x_k). \tag{4}$$

The steering law can be interpreted as a saturating control law

$$\chi_r(e) := \arctan(-K_p e - K_i \int_0^t e(\tau) d\tau). \tag{5}$$

where the proportional gain is $K_p = 1/\Delta(t) > 0$, and $K_i > 0$ represents the integral gain. This lookahead-based steering law is equivalent to saturated proportional control law and $e(t)$ is the cross-track error given by

$$e(t) = -[x_{USV}(t) - x_k]\sin(\alpha_k) + [y_{USV}(t) - y_k]\cos(\alpha_k). \tag{6}$$

The lookahead-based steering can be implemented related to the heading controller applying the transformation shown in (7). The variable sideslip (drift) angle β [1] has been avoided to simplify the steering law.

$$\psi_d = \chi_d - \beta = \chi_p + \chi_r - \beta. \tag{7}$$

The switching mechanism is defined as a circle of acceptance for surface vehicles [1], which selects the next waypoint as a lookahead point if the position of the USV lies within a circle with radius R around (x_{k+1}, y_{k+1}). This circle of acceptance is defined as

$$[x_{USV}(t) - x_{k+1}]^2 + [y_{USV}(t) - y_{k+1}]^2 \leq R_{k+1}^2. \tag{8}$$

where, if the time surface vehicle position $(x_{USV}(t), y_{USV}(t))$ satisfies (8), the next waypoint (x_{k+1}, y_{k+1}) needs to be selected. Radius R is equal to two USV lengths ($R = 2L$). LOS guidance system and circle of acceptance are shown in Fig. 7.

B. Obstacle avoidance using LiDAR

Obstacle avoidance capabilities are an essential role of the GNC algorithm, as it provides the safety feature for the autonomous vehicle operation. In this study, LiDAR has been used as the main perception sensor. This sensor provides obstacle information in Cartesian coordinates for each beam with a 275° aperture angle. This position gives the possibility

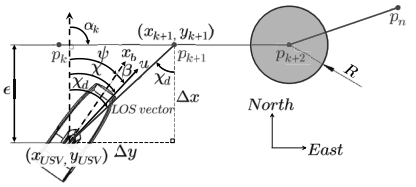


Fig. 7. LOS guidance system and circle of acceptance.

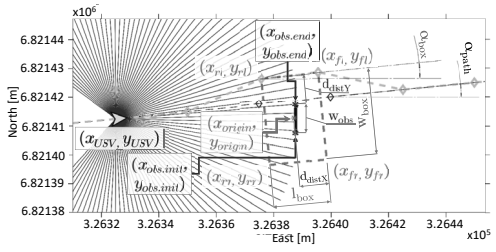


Fig. 8. Safety boundary box for obstacle recognition using LiDAR: Safety boundary box (dotted red line) is generated to create the new path for the USV (dotted green line).

to locate the obstacle in the relative map on the vehicle, or absolute coordinates. Also, as the LiDAR used in this study has four spread-out scan planes, if there is an object on a slope like the shore of an island, the closer detected point is always used for the obstacle avoidance.

Once that the obstacle has been located relative to the USV, a safety boundary box is defined around this object, providing additional information for the GNC algorithm. The implementation of this boundary box is shown in Fig. 8. This boundary box is defined in (9) as a rectangle-shaped box with width w_{box} and length l_{box} parameters. Both dimensions are determined with the obstacle's width w_{obs} and length l_{obs}, and by a predefined fixed parameter for safety distance d_{safe_X} and d_{safe_Y} in X and Y axis, respectively. The boundary box origin (x_{origin}, y_{origin}) is defined according to the absolute USV position and the LiDAR data defined in (10).

$$\begin{cases} w_{box} = d_{safe_Y} + \frac{w_{obs}}{2} \\ l_{box} = d_{safe_X} + \frac{l_{obs}}{2} \end{cases}. \tag{9}$$

$$\begin{bmatrix} x_{origin} \\ y_{origin} \end{bmatrix} = \begin{bmatrix} x_{USV} \\ y_{USV} \end{bmatrix} + \begin{bmatrix} \frac{x_{obs,init} + x_{obs,end}}{2} \\ \frac{y_{obs,init} + y_{obs,end}}{2} \end{bmatrix}. \tag{10}$$

As the obstacle avoidance is designed for harbor operations, the shape of this boundary box is selected according to the distance from the left to the right side walls of the harbor route. Also, the shape can be related to the vehicle position according to the GPS (once it enters a predefined harbor area). The safety boundary box has the same angle α_{box} as the slope of the predefined path α_{path}, allowing a continuous path operation. The corners of the boundary box are used as new waypoints of the path-following in the GNC algorithm, continuing with the LOS control approach. The obstacle avoidance algorithm also selects the new waypoint depending on which side is a closer trajectory to the vehicle. The first waypoint can be rear left (x_{rl}, y_{rl}) or rear right (x_{rr}, y_{rr}), while the next one is the correspondent side of the front corner. Equations (11) and (12) define all boundary box corners, where a and b refers to rear/front or left/right respectively, and i and j to the sign for the second element. $R(\alpha_{path})$ is the rotation matrix in the XY-plane counterclockwise through α_{path}.

$$\begin{bmatrix} x_{ab} \\ y_{ab} \end{bmatrix} = \begin{bmatrix} x_{origin} \\ y_{origin} \end{bmatrix} + R(\alpha_{path}) \begin{bmatrix} i\frac{w_{box}}{2} \\ j\frac{l_{box}}{2} \end{bmatrix}. \tag{11}$$

$$\begin{bmatrix} x_{ab} \\ y_{ab} \end{bmatrix} = \begin{cases} i = 1 & \text{if } a = f \\ i = -1 & \text{if } a = r \\ j = 1 & \text{if } b = l \\ i = -1 & \text{if } b = r \end{cases}. \tag{12}$$

The new waypoints are selected for the LOS algorithm to perform a smooth USV trajectory, including additional waypoints to the predefined path. If there is an obstacle outside the USV predefined path, the boundary box is created, but it does not affect the next waypoint. In addition, the algorithm continuously updates the safety boundary box position, being suitable for the avoidance of stationary and slow-motion objects. Once that the obstacle has been avoided, the USV follows the predefined path until it recognizes another object in its trajectory or it reaches the ending waypoint.

C. Surge and Yaw controllers

A LOS path-following controller can be designed for a USV by representing the desired path by waypoints (x_n, y_n). This controller sends heading commands to the yaw controller to match the predefined path. The main control system is based on two separate PID controllers for surge and yaw control, and their parameters are obtained by using Rapid Control Prototyping (RCP) during the field-tests. The surge controller keeps the USV at a predefined constant speed.

The feedback loop incorporates low-pass and notch filters to reduce motions induced by waves [1]. A first-order low-pass filter with a time constant T_f can be designed according to

$$h_{lp}(s) = \frac{1}{1 + T_f s}. \tag{13}$$

This filter is used to suppress forces over the frequency $1/T_f$. Although this criterion is hard to specify for USVs, it has been defined as T_f = 0.1 s time constant after simulation and implementation results.

The bandwidth of the controller ω_b can be close to or within the range $\omega_{min} < \omega_e < \omega_{max}$ of the wave spectrum for small USVs. Adding a low-pass filter in cascade with a notch filter handles this problem. However, the estimation of the notch frequency ω_n might not be accurate. Therefore, a filter structure formed by three cascaded notch filters with fixed center frequencies has been added in the feedback loop

$$h_n(s) = \prod_{i=1}^{3} \frac{s^2 + 2\xi\omega_i s + \omega_i^2}{(s + \omega_i)^2}. \tag{14}$$

The center frequencies of the notch filters have been assumed as ω_1 = 0.1 rad/s, ω_2 = 0.2 rad/s, and ω_3 = 0.4 rad/s

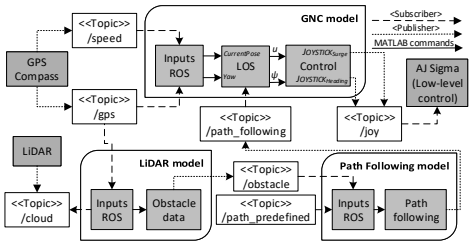


Fig. 9. Schematic of the modular GNC system: The three models are the modules included in the USV platform, which are run in separate ROS nodes.

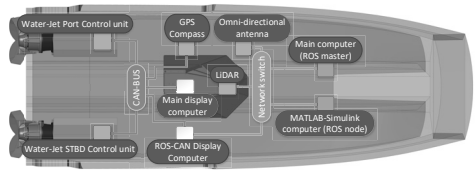


Fig. 10. System overview of the USV platform with high-level (blue boxes), intermediate-level (white boxes), and low-level control (purple boxes).

because of the lack of big waves in the field-test conducted in harbor conditions. Hence, small wave effects have been suppressed by getting steady outputs in both surge and yaw controllers.

D. Modular system for the path planning algorithm

The path-following algorithm uses a modular approach to include obstacle detection, path-following, and control in the USV platform. Each of these modules runs a separate ROS node in the system, allowing the use of remotely operated applications. Hence, the obstacles can be positioned while the USV is operating. This approach has been previously studied in [26], implementing a path-following algorithm with a straight line. However, the algorithm in [26] did not include any collision avoidance capabilities.

All these modules use ROS messages to communicate between them, being simple to check if each module is operating correctly. Fig. 9 illustrates this modular architecture with all topics involved, defining the subscribers and publishers of each topic with the dashed or continuous line respectively. The obstacle detection module (LiDAR model) processes the data acquired from the LiDAR and the GPS-Compass and allows the GNC algorithm to obtain the position of the obstacle in absolute coordinates. Once the obstacle has been detected, its origin position, length, and width are sent to the path-following model, allowing low data transfer. This model checks if the obstacle is within the predefined path. Therefore, the path-following topic is generated depending on the interference of the obstacle in the path. This path-following topic includes the waypoints for the GNC algorithm in the control module. The GNC guidance algorithm generates the required USV heading command, sending this parameter to the controller. The controller generates the required Joystick parameters for surge Joy_{surge} and yaw Joy_{yaw} to reach the LOS values. These joystick parameters are sent to the low-level control for the USV operation, using the same commands as a manual 3-axis joystick (surge, sway, and yaw motion).

IV. Experimental validation

A. System implementation

The USV incorporates multiple mechatronics systems (see Fig. 10) to sense the surrounding environment, plan a path to a destination, and control steering and speed of the vehicle. Hence, the system can be described as high-level control (computers with ROS), which performs complex computations, and low-level control (sensors and waterjet control units), which is used as an interface for basic vehicle operations. In addition, intermediate-level control (display computers) is the main link between low-level data acquisition and high-level logic operations. A LiDAR provides collision avoidance capabilities for the USV, along with the 3D map construction of the environment. All this instrumentation installed in the USV employs ROS as a framework. Thus, this framework provides necessary tools and packages to access sensor data, process it, and generate an appropriate response for different actuators.

The USV used in this study contains a Linux computer (ROS Master), which is connected to the rest of the instrumentation by a network switch via Ethernet. This Linux computer has ROS installed to send and receive the necessary commands for the USV operation. The MATLAB-Simulink computer (ROS node) is only used for testing purposes, and it includes a stand-alone ROS-node that permits a rapid prototyping procedure while testing [27]. This stand-alone ROS-node provides a solution for the labor process of C++ programming, skipping several steps to implement successfully the desired algorithm.

Two CCPilot VC display computers are used for the intermediate-level control [28]. These display computers are freely programmable, contain two CAN interfaces, Ethernet port, and they have IP66 class, being suitable for offshore environments. The ROS-CAN display computer is used to translate between CAN bus and ROS messages received from low-level and high-level systems. It uses rosserial to be connected with the rest of the system, which allows utilizing ROS with embedded systems. This display computer receives the CAN bus message from the GPS-Compass, and it can create the necessary ROS messages for the control unit (GPS position, heading, and speed of the USV) without requiring any additional converter (e.g., USB to CAN adaptor). Also, this display computer communicates via CAN bus with the main display computer, and it sends the joystick commands obtained from the high-level ROS computers, connecting the two waterjet control units with the rest of the system. The main display computer is in charge of sending joystick commands to the waterjet control units based upon priority levels. Fig. 11 shows the USV priority control level, where the ROS $/joy$ node acts as a virtual joystick with control commands Joy_{surge} and Joy_{yaw}. The steering wheel and 3-axis joystick, both forming the manual control of the USV, provides the safety feature in the autonomous algorithm.

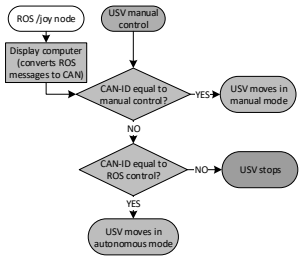


Fig. 11. Stateflow diagram for priority control level using CAN-ID: The manual joystick operation is defined as the highest priority because of safety.

B. Experimental results: Control Scenario I

The first test simulates the USV in Tammerkoski, which is a river in Tampere, Finland. This Control Scenario focuses on the obstacle avoidance implementation, not including environmental disturbances in the system. A series of GPS waypoints form the predefined path, which has a stationary obstacle in the middle of its path. Fig. 12 shows the experimental results for this Control Scenario, where the USV follows a predefined path avoiding a stationary obstacle. First, the blocking obstacles are detected by the LiDAR model, defining their origin, width, and length for the Path-following model. Then, the GNC algorithm creates a new path based on the safety boundary box, and the USV completes its trajectory.

This scenario uses the mathematical model developed in Section II, and the GNC architecture introduced in Section III. Since the harbor has a narrow channel, the safety boundary box is declared with small w_{box} and l_{box} dimensions. This test validates the GNC algorithm, generating precisely the safety boundary box, and reaching the last waypoint of the USV path.

C. Experimental results: Control Scenario II

The second test is implemented in the USV at the Pyhäjärvi lake in Tampere, Finland (see Fig. 13). The implementation consists of a path-following with collision avoidance capabilities of a static obstacle (buoy). This control scenario shows the GNC architecture capabilities for harbor conditions, but it is demonstrated in a clear obstacle area in the middle of the lake due to safety conditions.

As shown in Fig. 14, it is confirmed that the path-following with obstacle avoidance experiments were satisfactorily performed with the proposed GNC approach, whereby the black dashed, green dashed, and blue solid lines are the predefined, GNC, and actual paths respectively. The path is predefined with a series of GPS waypoints, and the USV initial position is defined as random. This randomness demonstrates the capability of the USV to reach the path from a distant initial position. Once that the blocking obstacle is detected, the GNC algorithm creates a new path based on the safety boundary box. Corresponding LOS cross-track error $e(t)$ is shown in Fig. 15, which demonstrates the correct performance of the designed GNC algorithm. In addition, Fig. 16 shows the input control values (surge and yaw parameters). The scenario has a constant surge velocity with a variable heading of the USV, obtained from the LOS guidance system.

V. Conclusion and Future Work

This article was concerned with the path-following with obstacle avoidance using a LiDAR of a USV in harbor conditions. The simulated USV model was presented to verify the designed GNC architecture. This model was based on SI methods using field-test data for surge and yaw motions of the USV. Once the USV model was validated, the GNC system with LOS based model was developed with collision avoidance capabilities. This GNC system uses a modular approach to include obstacle detection, path-following, and control in the USV platform. After designing the GNC architecture, a system implementation of the modular approach was included in the USV with three control levels (high, intermediate, and low). The experimental results show two control scenarios in both simulation and field-test, presenting the capabilities and the adequate performance of the designed GNC architecture.

Future work will include the implementation of the GNC algorithm with high-speed moving obstacles and with multiple stationary/moving obstacles, calculating a projected safety boundary box for each moving obstacle. In addition, future work will study the use of other perception sensors rather than LiDAR for long-range obstacle avoidance, such as radar.

Acknowledgment

This paper is based on the aColor project funded by Technology Industries of Finland Centennial and Jane & Aatos Erkko Foundations. The authors gratefully acknowledge the contributions of Sauli Virta from Alamarin-Jet Oy.

References

[1] T. I. Fossen, *Handbook of Marine Craft Hydrodynamics and Motion Control*. John Wiley & Sons, Ltd, Apr 2011.

[2] K. Nomoto, K. Taguchi, K. Honda, and S. Hirano, "On the steering qualities of ships," *Journal of Zosen Kiokai*, vol. 1956, no. 99, pp. 75–82, 1956.

[3] L. Ljung, "System identification," *Wiley Encyclopedia of Electrical and Electronics Engineering*, pp. 1–19, 1999.

[4] D. Moreno-Salinas, D. Chaos, J. M. de la Cruz, and J. Aranda, "Identification of a surface marine vessel using ls-svm," *Journal of Applied Mathematics*, vol. 2013, pp. 1–11, 2013.

[5] J. Shin, D. J. Kwak, and Y.-i. Lee, "Adaptive path-following control for an unmanned surface vessel using an identified dynamic model," *IEEE/ASME transactions on mechatronics*, vol. 22, no. 3, pp. 1143–1153, 2017.

[6] S.-R. Oh, J. Sun, Z. Li, E. A. Celkis, and D. Parsons, "System identification of a model ship using a mechatronic system," *IEEE/ASME Transactions on Mechatronics*, vol. 15, no. 2, pp. 316–320, 2009.

[7] J. Han, J. Xiong, Y. He, F. Gu, and D. Li, "Nonlinear modeling for a water-jet propulsion usv: An experimental study," *IEEE Transactions on Industrial Electronics*, vol. 64, no. 4, pp. 3348–3358, Apr 2017.

[8] D. Mu, G. Wang, Y. Fan, Y. Bai, and Y. Zhao, "Path following for podded propulsion unmanned surface vehicle: Theory, simulation and experiment," *IEEJ Transactions on Electrical and Electronic Engineering*, vol. 13, no. 6, pp. 911–923, Jun 2018.

[9] M. Breivik and T. I. Fossen, "Path following for marine surface vessels," in *Oceans' 04 MTS/IEEE Techno-Ocean'04 (IEEE Cat. No. 04CH37600)*, vol. 4. IEEE, 2004, pp. 2282–2289.

[10] N. Wang and X. Pan, "Path following of autonomous underactuated ships: A translation-rotation cascade control approach," *IEEE/ASME Transactions on Mechatronics*, vol. 24, no. 6, pp. 2583–2593, Dec 2019.

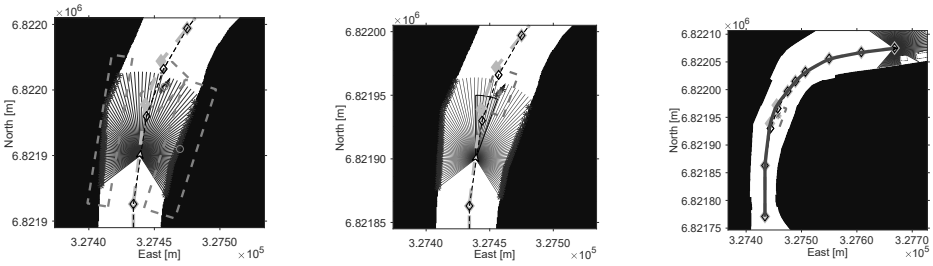


Fig. 12. Control Scenario I: (a) LiDAR data acquisition and processing for object recognition: all safety boundary box for the LiDAR detected obstacles (dotted purple lines) are created to select the one/s to be avoided. (b) LiDAR data processing for blocking obstacle: the safety boundary box of the blocking obstacle (dotted red line) is created with the slope of the path at the obstacle position. (c) Simulated USV trajectory.

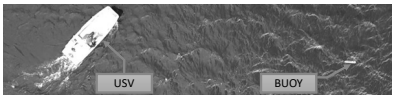


Fig. 13. Control Scenario II: USV during field-test in Tampere, Finland.

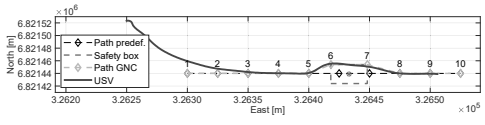


Fig. 14. Control Scenario II: USV path-following with obstacle avoidance of a stationary object. The path generated from the GNC algorithm is marked with a green line, and it uses the left-side corners of the safety boundary box (dotted red line). Waypoints are marked with their respective order number.

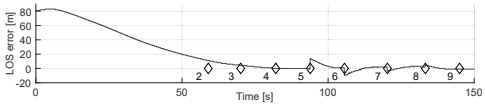


Fig. 15. Control Scenario II: Line-Of-Sight cross-track error $e(t)$ for the lookahead-based steering law defined in (6).

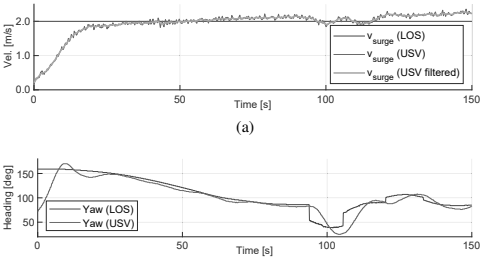


Fig. 16. Control Scenario II: (a) Comparison of constant surge velocity of 2 m/s with field-test data from the USV, (b) Comparison of heading angle from the LOS guidance system with field-test data from the USV.

[11] R. Halterman and M. Bruch, "Velodyne hdl-64e lidar for unmanned surface vehicle obstacle detection," in *Unmanned Systems Technology XII*, ser. SPIE Proceedings, vol. 7692. SPIE, Apr 2010, p. 76920D.

[12] O. Gal and E. Zeitouni, "Tracking objects using phd filter for usv autonomous capabilities," in *Robotic sailing 2012*. Springer, 2013, pp. 3–12.

[13] *MRS1000: Operating instructions*, SICK AG, Erwin-Sick-Str. 1, 79183, Waldkirch, Germany, Apr 2019, 8020494/12FY/2019-04-02.

[14] E. Simetti, S. Torelli, G. Casalino, and A. Turetta, "Experimental results on obstacle avoidance for high speed unmanned surface vehicles," in *2014 Oceans-St. John's*. IEEE, 2014, pp. 1–6.

[15] P. Wu, S. Xie, H. Liu, M. Li, H. Li, Y. Peng, X. Li, and J. Luo, "Autonomous obstacle avoidance of an unmanned surface vehicle based on cooperative manoeuvring," *Industrial Robot An International Journal*, vol. 44, no. 1, pp. 64–74, Jan 2017.

[16] J. A. Curcio, *Rules of the Road for Unmanned Marine Vehicles*. Springer International Publishing, 2016, ch. chapter 23, pp. 517–526.

[17] Y. Wang, X. Yu, X. Liang, and B. Li, "A colregs-based obstacle avoidance approach for unmanned surface vehicles," *Ocean Engineering*, vol. 169, pp. 110–124, Dec 2018.

[18] S. Moe and K. Y. Pettersen, "Set-based line-of-sight (los) path following with collision avoidance for underactuated unmanned surface vessel," in *2016 24th Mediterranean Conference on Control and Automation (MED)*. IEEE, Jun 2016, pp. 402–409.

[19] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng, "Ros: an open-source robot operating system," in *ICRA workshop on open source software*, vol. 3, no. 3.2. Kobe, Japan, 2009, p. 5.

[20] P. Bouchier, "Embedded ros [ros topics]," *IEEE Robotics & Automation Magazine*, vol. 20, no. 2, pp. 17–19, Jun 2013.

[21] M. Alberri, S. Hegazy, M. Badra, M. Nasr, O. M. Shehata, and E. I. Morgan, "Generic ros-based architecture for heterogeneous multi-autonomous systems development," in *2018 IEEE International Conference on Vehicular Electronics and Safety (ICVES)*. IEEE, 2018.

[22] M. Aeberhard, T. Kühbeck, B. Seidl, M. Friedl, J. Thomas, and O. Scheickl, "Automated driving with ros at bmw," *ROSCon 2015 Hamburg, Germany*, 2015.

[23] M. Ollikainen and M. Ollikainen, "The finnish coordinate reference systems," *Finnish Geodetic Institute and the National Land Survey.*, 2004.

[24] *System Identification Toolbox User's Guide*, The MathWorks, Inc., Natick, MA, 1988, release 2019a.

[25] F. A. Papoulias, "Bifurcation analysis of line of sight vehicle guidance using sliding modes," *Int. J. Bifurcation Chaos*, vol. 1, pp. 849–865, 1991.

[26] J. Villa, J. Aaltonen, and K. T. Koskinen, "Model-based path planning and obstacle avoidance architecture for a twin jet unmanned surface vessel," in *2019 Third IEEE International Conference on Robotic Computing (IRC)*. IEEE, Feb 2019, pp. 427–428.

[27] *ROS Toolbox User's Guide*, The MathWorks, Inc., Natick, MA, 2019, release 2019a.

[28] *CCPilot VC Technical manual*, CrossControl, Mar 2019, product revision: 2.4.

PUBLICATION IV

A Co-Operative Autonomous Offshore System for Target Detection Using Multi-Sensor Technology

Jose Villa, Jussi Aaltonen, Sauli Virta and Kari T. Koskinen

Remote Sensing 12.24 (2020), 4106
DOI: 10.3390/rs12244106

remote sensing

MDPI

Article

A Co-Operative Autonomous Offshore System for Target Detection Using Multi-Sensor Technology

Jose Villa [1,*], Jussi Aaltonen [1], Sauli Virta [2] and Kari T. Koskinen [1]

[1] Mechatronics Research Group (MRG), Tampere University (TAU), 33720 Tampere, Finland; jussi.aaltonen@tuni.fi (J.A.); kari.koskinen@tuni.fi (K.T.K.)

[2] Alamarin-Jet Oy, 62300 Härmä, Finland; sauli.virta@alamarinjet.com

* Correspondence: jose.villa@tuni.fi; Tel.: +358-50-448-1926

Received: 16 November 2020; Accepted: 11 December 2020; Published: 16 December 2020

Abstract: This article studies the design, modeling, and implementation challenges for a target detection algorithm using multi-sensor technology of a co-operative autonomous offshore system, formed by an unmanned surface vehicle (USV) and an autonomous underwater vehicle (AUV). First, the study develops an accurate mathematical model of the USV to be included as a simulation environment for testing the guidance, navigation, and control (GNC) algorithm. Then, a guidance system is addressed based on an underwater coverage path for the AUV, which uses a mechanical imaging sonar as the primary AUV perception sensor and ultra-short baseline (USBL) as a positioning system. Once the target is detected, the AUV sends its location to the USV, which creates a straight-line for path following with obstacle avoidance capabilities, using a LiDAR as the main USV perception sensor. This communication in the co-operative autonomous offshore system includes a decentralized Robot Operating System (ROS) framework with a master node at each vehicle. Additionally, each vehicle uses a modular approach for the GNC architecture, including target detection, path-following, and guidance control modules. Finally, implementation challenges in a field test scenario involving both AUV and USV are addressed to validate the target detection algorithm.

Keywords: target detection; co-operative; autonomous; multi-robot; USV; AUV

1. Introduction

In recent years, the use of autonomous offshore vehicles, which includes autonomous underwater vehicles (AUVs) and unmanned surface vehicles (USVs), for marine interventions has attracted increasing interest from research scientists, maritime industries, and the military. These interventions include several activities such as offshore surveillance, offshore target detection, seabed explorations, or search and rescue (SAR) missions. Additionally, the use of multi-robot platforms can improve the performance in these activities, as they can include above and below-water characterization. Regarding a multi-robot platform, Vasilijević et al. [1] presented the co-operative robotic system consisting of an AUV and a USV for ocean sampling and environmental monitoring. In [2], the study used a heterogeneous collaborative system of above, surface, and underwater robots to obtain a multi-domain awareness on a floating target. The heterogeneous system consists of a USV, an AUV, and an unmanned aerial vehicle (UAV). Additionally, Gu et al. [3] presented a homogeneous study, where a guidance and control law design method for coordinated path following of networked under-actuated robotic USVs under directed communication links. In [4], the control scenario simulated a homogeneous AUV fleet to study formation tracking control and collision-obstacle avoidance.

To accomplish the target detection in the offshore environment, the availability of accurate USV and AUV mathematical models is crucial for simulation study purposes, controller design, and development. The theoretical six-degrees-of-freedom (DOFs) dynamic model [5], based on nonlinear equations of

motion, can be used for the design and modeling of the AUV. Equally, the USV can use the same dynamic model of the AUV but with reduced order for the three DOFs horizontal plane control (surge, sway, and yaw motions). Several tools can help to obtain the coefficients of the dynamic model equations and the necessary transfer functions of each vehicle. These tools can include the parameter estimation from MATLAB-Simulink [6], and the system identification (SI) [7,8], introduced to develop the mathematical model using field test data. In [9], SI of the maneuvering data determined the hydrodynamic coefficients of a USV. Also, the mathematical model of the USV includes the propulsion and power system. Commonly, the rudder and propeller, or waterjet propulsion systems provide the heading and the speed control of most existing USVs. In [10], a twin waterjet propelled USV was modeled based on SI, but it neglects the calculation for the dynamics of the propulsion system.

Target detection in offshore environments is a fundamental activity that combines different perception sensors. Numerous studies use passive (stereo cameras) or active (LiDAR or radar) perception methods to obtain situational awareness of a USV. Nonetheless, most of the obstacle detection methods rely on depth measurements, in which LiDAR sensors are the most reliable method of obtaining depth data. Correspondingly, sonar devices are still the most convenient option for collecting data on underwater environments. Mechanical imaging sonar, multibeam, profiler, or sidescan are some of the main sonar imaging and ranging devices. For the target detection with sonar devices, how detectable is a target is mainly dependent on the physical characteristics of the target and acoustic signal. Some studies use sonar devices for target detection capabilities, as in [11], where a profiler sonar was adopted for obstacle detection. According to [12], a method for underwater obstacle detection (standard buoy) was developed using forward-looking sonar and a probabilistic local occupancy grid.

Correct localization and navigation are crucial to ensure the accuracy of the gathered data for all these applications. Above the water surface, most of the autonomous systems rely on radio or global positioning and spread-spectrum communications, as a GPS-compass installed in the USV platform. However, those signals propagate only in short distances in an underwater scenario, where acoustic-based systems perform better. Regarding underwater navigation, the three fundamental methods are dead-reckoning (DR) and inertial navigation systems (INS), acoustic navigation, and geophysical navigation techniques [13]. These navigation methods require specific survey and navigation sensors installed in the AUV. The Girona 500 [14] is an example of AUV that performs the traditional dead-reckoning navigation utilizing a doppler velocity log (DVL) and a solid-state attitude and heading reference system (AHRS). Also, the absolute position can be obtained through a GPS when the vehicle is on the surface and using an ultra-short baseline (USBL) while underwater. The high-accuracy USBL system allows the localization of the AUV and the communication between the vehicle and the surface unit. In [15], the study provided a navigation algorithm for an underwater vehicle with a Kalman filter to estimate the error state via measurement residuals from aiding sensors. These aiding sensors incorporate an attitude sensor, a DVL, a long-baseline (LBL) system, and a pressure sensor. In acoustic navigation techniques, acoustic transponders and modems perform localization by measuring the time-of-flight of signals from acoustic beacons or modems. USBL navigation allows an AUV to localize itself relative to a USV, and it provides an efficient and reliable acoustic communication network [16]. In [17], the study presented the design and implementation of an USBL-aided navigation approach for an AUV in a two-parallel extended Kalman filter (EKF). It also includes the measurements provided by a DVL, a Visual Odometer, an inertial measurement unit (IMU), a pressure sensor, and a GPS.

Safe and adequate control of the offshore vehicles depends notably on proper guidance, navigation, and control (GNC) systems. This study adopts a path-following as the guidance system for both offshore platforms. The path-following approach is closer to practical engineering, and it is easier to implement than trajectory tracking. A generally used method for path-following in autonomous vehicles is the named line-of-sight (LOS) guidance. LOS guidance is classified as a three-point guidance scheme, involving a commonly stationary reference point along with the interceptor and the target [5]. In [18], the study developed a guidance-based algorithm for path-following using the LOS algorithm

in offshore operations. Additionally, in [10], a path-following with obstacle avoidance based on the safety boundary box approach was implemented in a USV with a LOS-based guidance system.

Due to the co-operative offshore system in this study, it becomes necessary to fuse information obtained from the individual vehicles. Robot Operating System (ROS) has been an effective tool when working with multi-robot systems. This tool is a flexible framework for writing robot software and provides the tools to acquire sensors' data, process it, and generate the necessary response for the vehicle actuators [19]. Multi-robot systems can either be centralized with a ROS master node at the ground control station (GCS) or decentralized with each autonomous vehicle (AV) running an independent ROS master. In the case of the decentralized control techniques, they are more flexible, profitable, and generally reduce the communication network requirements compared with centralized control [20]. However, they are also more challenging due to obstacles, uncertainties, and communication constraints, such as noises, delays, dropouts, or failures. In this case, the multi-master approach provides a solution where each vehicle keeps its own ROS master and also exchange the necessary information with other components of the multi-robot system. In [21], they proposed a package that efficiently developed multi-master architectures.

In the presented manuscript, the mathematical model of the USV consists of the simplified three DOFs dynamic model [5], where their parameters are obtained from field test data using the parameter estimation tool. Additionally, the waterjet model has been included in the mathematical model of the USV using data from the manufacturer and transfer functions based on SI. The AUV platform considered in this study does not incorporate a DVL, neglecting the velocity feedback of the vehicle. However, the installed USBL provides an absolute position and a communication link between the USV and the AUV. Thus, the AUV platform includes a basic setup for underwater localization, but it is not able to precisely locate the vehicle underwater. The path-following algorithm uses the LOS approach for heading control to simplify the guidance control of the AUV, keeping a constant depth and constant surge speed. The target detection algorithm uses a modular ROS architecture to provide a computationally cheap and simple implementation in both offshore platforms. Furthermore, the offshore system includes two different perception sensors based on the same target detection algorithm. Finally, a multi-master architecture is in charge of the interaction between the AUV and USV, providing an easy plug-and-play solution for the multi-robot system.

In this work, a model-based GNC architecture for a co-operative autonomous offshore system is proposed for target detection using multi-sensor technology. In Section 2, the USV modeling and simulation are presented using the parameter estimation tool to define the waterjet and USV maneuvering model. Furthermore, this section includes an overview of the USV and AUV platforms. Then, in Section 3, the GNC system for the co-operative tasks is included using the LOS-based guidance system for control. The target detection algorithm is developed using a mechanical imaging sonar at the AUV and a LiDAR at the USV as the primary perception sensor for underwater and surface inspection, respectively. Finally, in Section 4, the implementation of a GNC architecture is described as modular and multilayer for the multi-robot system. A control scenario in a field test is shown in this section to validate the proposed target detection algorithm.

2. Modeling and Simulation for the Offshore Vehicles

The co-operative autonomous offshore system consists of two different vehicles: a USV and an AUV. This section gives an overview of both subsystems, and it describes the simulation model of the USV, which provides the capability to develop the GNC algorithms.

2.1. Overview of Under-Actuated USV

This article uses an under-actuated USV as the primary vehicle in the co-operative autonomous offshore system. The USV is an aluminum hull with a thrust vectoring waterjet propulsion system, which provides optimal maneuverability using a twin waterjet configuration. Figure 1 shows a simplified model of the vehicle, where the port and starboard (STDB) waterjets produce the necessary thrust forces

to move forward, backward, sideways or performing turns. Additionally, Figure 1 includes the position and orientation of the USV in the North-East-Down (NED) coordinate system. The NED coordinate system is related to planar Cartesian coordinates, so a coordinate transformation is performed from the GPS-compass output to get the USV's absolute position. This transformation is between longitude and latitude (l, μ) from the world geodetic system 84 (WGS84) coordinate system and ETRS-TM35FIN [22], which displays the NED position (x_{USV}, y_{USV}). The Euler angles provide the USV heading or yaw angle ψ. The motion of the USV has three DOFs, which are surge, sway, and yaw (linear (u, v), and angular r velocities) while ignoring roll, pitch, and heave motions.

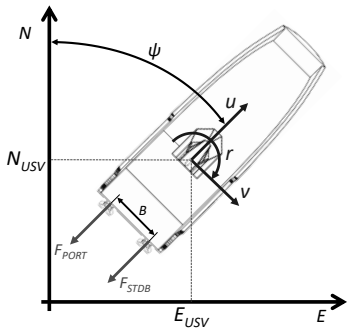


Figure 1. Simplified model of the unmanned surface vehicle (USV) using the North-East-Down (NED) coordinate system. USV motion is described by surge u (linear longitudinal motion), sway v (linear transverse), and yaw motion r (turning rotation about its z-axis).

2.2. USV Modeling

The development of an adequate maneuvering model will simplify the GNC algorithms design and simulation. The three DOFs horizontal plane model for maneuvering of a USV consists of the rigid-body kinetics [5]

$$\boldsymbol{M}\dot{\boldsymbol{\nu}} + \boldsymbol{C}(\boldsymbol{\nu})\boldsymbol{\nu} + \boldsymbol{D}(\boldsymbol{\nu})\boldsymbol{\nu} = \boldsymbol{\tau} + \boldsymbol{\tau}_{\mathrm{wind}} + \boldsymbol{\tau}_{\mathrm{wave}}, \tag{1}$$

where $\boldsymbol{\nu} = [u, v, r]^T$ is the velocity vector composed of surge, sway and yaw. $\boldsymbol{\tau} = [\tau_{\mathrm{u}}, 0, \tau_{\mathrm{r}}]$ is the vector forces and moments generated by twin waterjet configuration, while $\boldsymbol{\tau}_{\mathrm{wind}}$ and $\boldsymbol{\tau}_{\mathrm{wave}}$ are the environmental forces. $\boldsymbol{M}$, $\boldsymbol{C}(\boldsymbol{\nu})$, and $\boldsymbol{D}(\boldsymbol{\nu})$ are the mass, Coriolis and damping matrices, respectively, where $\boldsymbol{M}$ and $\boldsymbol{C}(\boldsymbol{\nu})$ combine added and rigid-body terms. The mass matrix $\boldsymbol{M}$ is defined by

$$\boldsymbol{M} = \boldsymbol{M}_{\mathrm{RB}} + \boldsymbol{M}_{\mathrm{A}} = \begin{bmatrix} m - X_{\dot{u}} & 0 & 0 \\ 0 & m - Y_{\dot{v}} & m x_{\mathrm{g}} - Y_{\dot{r}} \\ 0 & m x_{\mathrm{g}} - Y_{\dot{r}} & I_{\mathrm{z}} - N_{\dot{r}} \end{bmatrix}, \tag{2}$$

where m is the mass of the vehicle, I_z is the moment of inertia about z_{b} axis, $\boldsymbol{r}_{\mathrm{g}}^{b} = [x_{\mathrm{g}}, y_{\mathrm{g}}, z_{\mathrm{g}}]^{\intercal}$ is the vector from origin o_{b} to centre of gravity CG, and $X_{\dot{u}}$, $Y_{\dot{v}}$, $Y_{\dot{r}}$, and $N_{\dot{r}}$ represent hydrodynamic added mass. The moment of inertia I_{z} at the pivot point has been estimated based on the calculation of the moments of inertia in the rear $I_{\mathrm{z,rear}}$ and front $I_{\mathrm{z,front}}$ of the USV. These moments of inertia are defined by

$$I_{\mathrm{z,rear}} = m_{\mathrm{pt}}\, l_{\mathrm{pt}}^2 + \left(\frac{1}{3}\, m_{\mathrm{hull}}\, c_{\mathrm{g}}\right) l_{\mathrm{pivot}}^2, \tag{3}$$

$$I_{\mathrm{z,front}} = \frac{1}{3}\, m_{\mathrm{hull}} \left(1 - c_{\mathrm{g}}\right) \kappa \left(l_{\mathrm{USV}} - l_{\mathrm{pivot}}\right)^2, \tag{4}$$

where m_{pt} is the estimated powertrain mass (engines, waterjets, fuel, etc.), l_{pt} is the estimated location of the powertrain mass, m_{hull} is the hull weight without powertrain mass, c_{g} is the relative center of

mass point having one as the front of the USV, l_{pivot} is the pivot point location, κ is a scaling factor as the mass is not evenly distributed from the pivot point to the front of the USV, and l_{USV} is the length of the USV. The total moment of inertia I_z is defined by

$$I_z = \left(I_{z,\text{rear}} + I_{z,\text{front}}\right) I_{\text{cor}}, \tag{5}$$

where I_{cor} is the tuning factor for the moment of inertia.

The Coriolis-centripetal matrix $\boldsymbol{C}(\boldsymbol{\nu})$ can always be parameterized such that $\boldsymbol{C}(\boldsymbol{\nu}) = \boldsymbol{C}^{\intercal}(\boldsymbol{\nu})$ [23]. However, linearization of the Coriolis and centripetal forces $\boldsymbol{C}_{\text{RB}}(\boldsymbol{\nu})$ and $\boldsymbol{C}_{\text{A}}(\boldsymbol{\nu})$ about zero angular velocity ($p = q = r = 0$) implies that the Coriolis and centripetal terms can be removed from the above expressions, that is $\boldsymbol{C}_{\text{RB}}(\boldsymbol{\nu}) = \boldsymbol{C}_{\text{A}}(\boldsymbol{\nu}) = 0$ [24]. Additionally, the mathematical model is simplified to take into account only surge and yaw motions, so Coriolis and centripetal terms have been removed at the three DOFs dynamic model in this study.

The different damping terms contribute to linear and quadratic damping [5]. Nonetheless, it is generally difficult to distinguish these effects. The total hydrodynamic damping matrix $\boldsymbol{D}(\boldsymbol{\nu}_r)$ is the sum of the linear part $\boldsymbol{D}_{\text{lin}}$ and the nonlinear part $\boldsymbol{D}_{\text{nlin}}(\boldsymbol{\nu}_r)$ such that

$$\boldsymbol{D}(\boldsymbol{\nu}_r) = \boldsymbol{D}_{\text{lin}} + \boldsymbol{D}_{\text{nlin}}(\boldsymbol{\nu}_r), \tag{6}$$

where $\boldsymbol{D}_{\text{lin}}$ is the linear damping matrix produced by potential damping and possible skin friction, and $\boldsymbol{D}_{\text{nlin}}(\boldsymbol{\nu}_r)$ is the nonlinear damping matrix as a result of the quadratic damping and higher-order terms, defined by

$$\boldsymbol{D}_{\text{lin}} = \begin{bmatrix} -X_u & 0 & 0 \\ 0 & -Y_v & -Y_r \\ 0 & -Y_r & -N_r \end{bmatrix}, \tag{7}$$

$$\boldsymbol{D}_{\text{nlin}}(\boldsymbol{\nu}_r) = \begin{bmatrix} -X_{|u|u} & 0 & 0 \\ 0 & -Y_{|v|v} & 0 \\ 0 & 0 & -N_{|r|r} \end{bmatrix} |\boldsymbol{\nu}_r|. \tag{8}$$

The USV used in this study includes the AJ245 waterjet units [25]. The nozzle position P_{nozzle} varies the direction of the jet flow, which generates the force needed for turning. Thus, the total thrust force F_{total} combines the engine rpm of the waterjet n_{rpm} and P_{nozzle}. The variable n_{rpm} is directly gathered from the waterjet engine, and P_{nozzle} is a variable from $-10{,}000$ to 10,000, with 0 as the neutral position and equal to forward motion. Table 1 shows the data obtained from the manufacturer Alamarin-Jet Oy for these waterjet units at a specific operating point. This operating point is selected at 1800 rpm, nozzle in the neutral position, and bucket in the full up position.

Table 1. Data obtained from manufacturer for an operating point of a single AJ245 waterjet unit.

Surge Speed [kt]	Thrust Force [kN]
2	2
4	1.85
6	1.7

The thrust forces and torques for the mathematical model of the USV are defined according to the manufacturer's data and an affinity law. Thus, a two-dimensional (2D) lookup table can include the relation between the shaft rotational speed of the waterjet engine N with the thrust force per waterjet F. The affinity law used to obtain the thrust force at the waterjet units is defined by

$$\frac{F_1}{F_2} = \left(\frac{N_1}{N_2}\right)^2. \tag{9}$$

Figure 2 shows the results for the affinity law with the manufacturer's data for a waterjet engine from 600 to 2400 rpm, which match the operational engine speeds of this study.

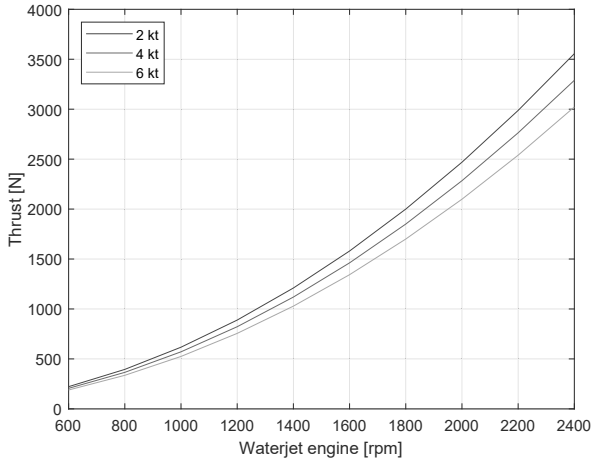


Figure 2. Thrust force F generated by the waterjet propulsion system depending on the shaft rotational speed N.

In the mathematical model, a 2D lookup table provides the engine rpm and the surge speed of the USV as inputs, and the total thrust generated by the waterjet unit as output. Also, a one-dimensional (1D) lookup table $f(Joy_u)$ obtains the engine rpm depending on the joystick input for surge motion, and a second-order transfer function adds the waterjet dynamics of the engine rpm into the mathematical model. This transfer function is obtained using the SI tool from MATLAB and the field test data of the USV. Thus, the engine rpm is calculated based on the combination of the 1D lookup table and the engine rpm transfer function, defined by

$$n_{\text{rpm}}(s) = \frac{0.317s^2 + 2.793s + 1.828}{s^2 + 3.499s + 1.828} f(Joy_u). \tag{10}$$

In the case of the heading motion of the USV, the total efficiency η_{nozzle} for the thrust force depends on the nozzle position (which refers to the angle of the waterjet thrust force α_{nozzle}). According to the waterjet manufacturer, if the nozzle position is deviated to a maximum nozzle angle $\eta_{\text{nozzle}} = \pm 25^\circ$ (related to $P_{\text{nozzle}} = \pm 10{,}000$) , efficiency drops exponentially to 30–40% of the maximum (center). The exponential function is obtained using the general exponential model.

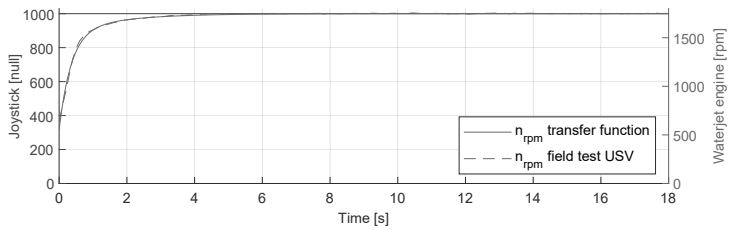
$$\eta_{\text{nozzle}}(P_{\text{nozzle}}) = a \exp(b\, P_{\text{nozzle}}), \tag{11}$$

where $a = 1$ and $b = -9.163 \times 10^{-5}$.

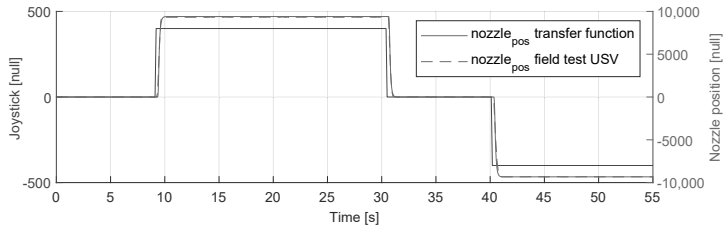
Similarly to the dynamics of the waterjet calculation for the engine rpm, the nozzle position includes a 1D lookup table $f(Joy_r)$ and a first-order transfer function. This transfer function is obtained also from the SI tool from MATLAB based on field test data. The nozzle position of each waterjet is defined by

$$P_{\text{nozzle}}(s) = \frac{-\exp(-0.25s)}{0.1s + 1} f(Joy_r). \tag{12}$$

Regarding the behavior of the second-order transfer functions for both engine rpm and nozzle position, Figure 3 shows the comparison between the SI tool transfer function and field test data for both n_{rpm} and P_{nozzle} variables.



(**a**)



(**b**)

Figure 3. Comparison between the USV field test data and the system identification (SI) transfer functions: (**a**) Engine rpm n_{rpm}. (**b**) Nozzle position P_{nozzle}.

Additionally, the parameters for the 1D Lookup table are obtained from field test data and are presented in Table 2.

Table 2. 1D Lookup Table parameters.

Joy_u	400	500	600	700	800	900	1000
n_{rpm}	690	920	1110	1300	1480	1650	1820
Joy_r	0	50	150	200	250	300	400
P_{nozzle}	0	1175	3500	4665	5830	7000	9325

Finally, the vector $\boldsymbol{\tau} = [\tau_{\text{u}}, 0, \tau_{\text{r}}]$, which represents the forces and moments generated by the two waterjets, is defined by

$$\begin{cases} \tau_{\text{u}} = (F_{\text{PORT}} + F_{\text{STDB}})\eta_{\text{nozzle}} \\ \tau_{\text{r}} = l_{\text{pivot}} \sin(\alpha_{\text{nozzle}})(F_{\text{PORT}} + F_{\text{STDB}})\eta_{\text{nozzle}} \end{cases}. \tag{13}$$

Figure 4 shows the schematic with all the necessary functions for the USV dynamic model, from the joystick controller input to the vehicle's position output. The waterjet model includes the 1D lookup table to translate between joystick commands to rpm, the second-order transfer function, and the 2D lookup table related to the thrust force of each waterjet unit. Furthermore, it also includes the 1D lookup table to translate between joystick commands to the nozzle position, the first-order transfer function, the thrust force efficiency depending on the nozzle position, and the calculation of the total torque. Both thrust force τ_{u} and torque τ_{r} are the inputs in the mathematical model of the USV based on the three DOFs dynamic model. The position and orientation of the USV are performed by integrating the velocity vector $\boldsymbol{\nu}$.

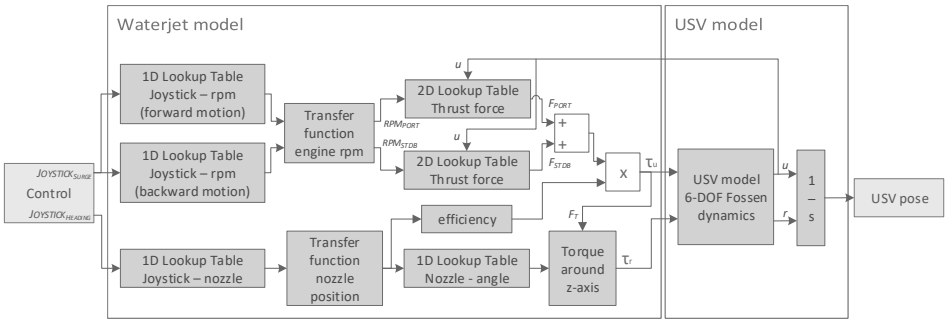


Figure 4. Schematic of the mathematical model of the USV including both waterjet propulsion system and USV dynamic models.

2.3. USV Model-Validation Using Parameter Estimation

The matrices $\boldsymbol{M}$ and $\boldsymbol{D}(\boldsymbol{\nu})$ of the three DOFs Dynamic model are estimated with the parameter estimation tool from MATLAB-Simulink. The matrices are defined in the Simulink model by creating the matrices from input values. Then, the MATLAB-Simulink tool can estimate the individual coefficients of the dynamic matrices.

There are two different parameter estimation runs related to surge and yaw motion. Table 3 shows the constant values shared in both experiments, while Table 4 shows the coefficients obtained from the parameter estimation tool with their results. Only surge and yaw motion coefficients, X_u, $X_{\dot{u}}$, $X_{|u|u}$ and N_r, $N_{\dot{r}}$, $N_{|r|r}$ respectively, have been considered and estimated in this study, as the mathematical model focuses in these two USV motions.

Table 3. Principal characteristics of the under-actuated USV.

Parameter	Value
m	3500 [kg]
m_{pt}	1100 [kg]
m_{hull}	2400 [kg]
l_{USV}	8 [m]
l_{pivot}	2.40 [m]
l_{pt}	2.16 [m]
κ	0.70
c_{g}	0.30
I_{cor}	0.6
I_z from (5)	11,284.61 $\left[\text{kg m}^2\right]$
x_{g}	0.0425 [m]

Table 4. Parameter estimation results for the surge and yaw motion coefficients.

Parameter	Value
X_u	−10.586
$X_{\dot{u}}$	−3277
$X_{\|u\|u}$	315.45
N_r	3907.9
$N_{\dot{r}}$	−36.555
$N_{\|r\|r}$	3459.6

Figure 5 shows the comparison between the field tests, which include raw and filtered USV linear and angular velocity, the three DOFs dynamic model with the coefficients obtained from the parameter estimation, and the SI results from [10], for the joystick controller input shown in Figure 3. As shown

in both linear and angular velocities results, the parameter estimation results improve the previous SI approach, giving an accurate output of the USV maneuvering compared to the field test results.

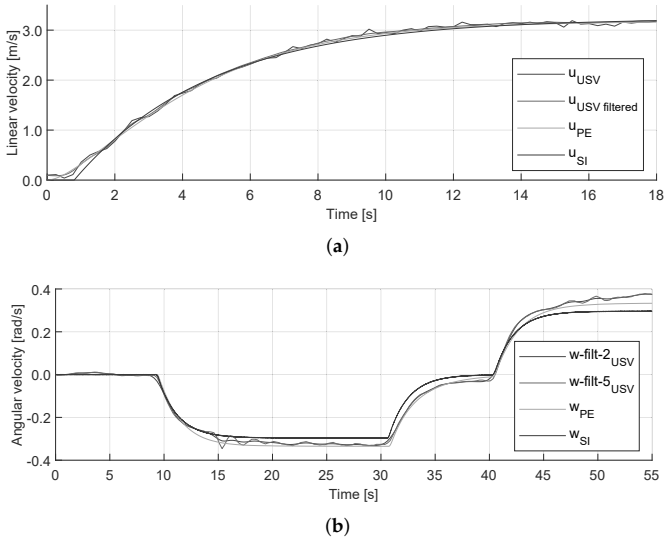


Figure 5. Comparison plot between SI tool, parameter estimation (PE) app, and field test data: (**a**) Surge motion. (**b**) Heading motion.

2.4. Overview of the AUV

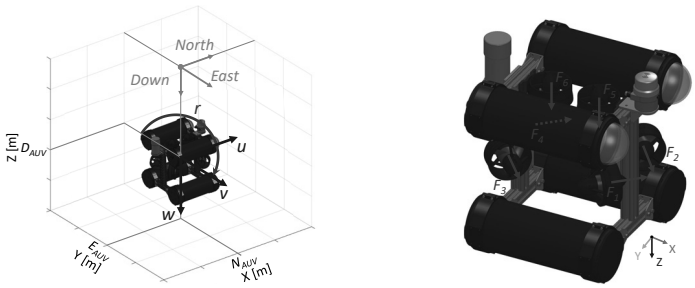
This article uses a high configurable AUV platform for different scientific instrumentation. This vehicle contains basic instrumentation and sensors for localization and target detection, including a USBL and a depth sensor for underwater localization and navigation, an AHRS from the flight control for the navigation of the AUV, and a mechanical imaging sonar (Tritech Micron [26]) as main underwater perception sensor.

Figure 6a shows a simplified model of the AUV. This AUV uses a six-thruster configuration to provide thrust forces when moving in the surge, sway, heave motions, or performing turns. Also, the position and velocities of the AUV are illustrated in Figure 6a. The general AUV motion in six DOFs is modeled by using the NED local coordinate system. AUV position and velocities are considered with the following vectors

$$\boldsymbol{\eta} = [N, E, D, \phi, \theta, \psi]^\top, \boldsymbol{\nu} = [u, v, w, p, q, r]^\top, \tag{14}$$

where N, E, D denote the NED positions in Earth-fixed coordinates, ϕ, θ, ψ are the Euler angles, u, v, w are the body-fixed linear velocities, and p, q, r are the body-fixed angular velocities [5].

The design and modeling of the AUV should be studied using a theoretical six DOFs dynamic model [27]. However, due to the lack of instrumentation, it is not possible to obtain accurate navigation data. Thus, the AUV is not fully simulated, and just simple control commands are established for navigation. Once that navigation data is available, it is possible to use the same approach as the USV mathematical model to obtain the six DOFs dynamic model, using the parameter estimation or SI tools based on field test data. Regarding the control of the AUV, thrusters are located as it is shown in Figure 6b, where thrusters T_1, T_2, T_3, and T_4 effects in surge, sway, and yawing, and thrusters T_5 and T_6 effects in heave and rolling motions.



(**a**) (**b**)

Figure 6. Six-thruster configuration in the AUV: (**a**) Simplified model of the considered vehicle using the NED coordinate system. (**b**) Thrust forces with their direction for each thruster.

3. Gnc System for the Co-Operative Tasks

This study has the target detection and the guidance algorithms as main modules of the GNC architecture of the offshore multi-vehicle system. This section describes both of these algorithms for each platform and the description of the multi-vehicle guidance system.

3.1. Target Detection System

The mechanical imaging sonar installed at the AUV and the LiDAR at the USV are the primary perception sensors in the co-operative autonomous offshore system. The target detection algorithm includes the application in both perception sensors, depending on the position of the objects (underwater or over the water surface).

For the mechanical imaging sonar, the employed algorithm consists of analyzing the acoustic intensity at every bin to determine the presence of an underwater vehicle. The Tritech Micron sonar [26] has an operating frequency chirp centered on 700 kHz, a beamwidth of 35° vertical and 3° horizontal, a range from 0.3 to 75 m, a range resolution of approximately 7.5 mm, and a configurable mechanical resolution of 0.45°, 0.9°, 1.8°, and 3.6°. In this study, the maximum range used to detect an obstacle is 10 m, a forward field-of-view (FoV) of 90°, and a mechanical resolution of 1.8°. If the target is known a priori to be narrow, the imaging sonar can be configured with a lower resolution to detect the object.

Regarding the data obtained from the mechanical imaging sonar, it contains the heading of the beam θ_{scan}, the location of the specific point in Cartesian coordinates P_{scan}, and the intensity at every bin I_{scan}. The dynamic range of the mechanical imaging sonar is 80 dB. Then, the dynamic range controls allow to adjust the position of a sampling window within the defined dynamic band range of the received signal, and it translates the intensity at every bin to an integer value ranging between 0 and 255.

After data acquisition from the mechanical imaging sonar, Algorithm 1 shows the post-processing steps for target detection. This algorithm includes the position of the highest intensity value for each bin in polar coordinates, filtering the data in the range of [0,1.5] meters to avoid possible noise from the AUV structure.

Algorithm 1 provides the post-processing of a single bin of a specific angle. An additional function forms an array of number of scans n_{scans}, obtained from $\theta_{\text{scan,min}}$, $\theta_{\text{scan,max}}$, and $\theta_{\text{scan,increment}}$ parameters of the mechanical imaging sonar to create the complete array of scans from the sonar. After gathering the scan array, the position of the targets needs to be calculated. The data from the perception sensors is obtained in the body-fixed reference frame (BODY), and it requires a translation into an absolute coordinate system. This translation is defined by

$$\begin{bmatrix} x_{\text{obs}} \\ y_{\text{obs}} \end{bmatrix} = \boldsymbol{R}_{\boldsymbol{z}}(\psi_{\text{AV}}) \begin{bmatrix} x_{\text{scan}} \\ y_{\text{scan}} \end{bmatrix}, \tag{15}$$

where $\boldsymbol{R_z}(\psi_{\mathrm{AV}})$ is the rotation matrix around the z-axis using the heading angle ψ_{AV} of the selected AV. This rotation matrix translates between the BODY and the East-North-Up (ENU) coordinate system. The rotation matrix $\boldsymbol{R_z}(\psi_{\mathrm{AV}})$ in 2D is defined by

$$\boldsymbol{R_z}(\psi_{\mathrm{AV}}) = \begin{bmatrix} \cos(\psi_{\mathrm{AV}}) & \sin(\psi_{\mathrm{AV}}) \\ -\sin(\psi_{\mathrm{AV}}) & \cos(\psi_{\mathrm{AV}}) \end{bmatrix}. \tag{16}$$

Algorithm 1: Post-processing of the mechanical imaging sonar data for target detection.

Input: Intensities I_{scan}, positions P_{scan} in Cartesian coordinates $[X,Y]$, and current heading θ_{scan} value obtained from the mechanical imaging sonar.
Output: Position *micron* of the highest intensity value in polar coordinates.

initialization;
`/* Remove data in the range from 0 to 1.5 m to avoid possible noise from the AUV structure.` n_{scan} `equal to number of scans. */`
for $i = 1$ **to** n_{scan} **do**
  calculate distance d_{scan} from P_{scan};
  if $d_{scan}(i) < 1.5$ **then**
    remove intensity $I_{\mathrm{scan}}(i)$;
  end
end
find maximum intensity $I_{\mathrm{scan,max}}$ from the I_{scan} data;
calculate value ρ_{scan} related to distance in polar coordinates;
`/* Return values for intensities greater than integer value of 80. Output in polar coordinates. */`
if $I_{scan,max} > 80$ **then**
  return $micron = [\theta_{scan}, \rho_{scan}]$;
else
  return $micron = [\theta_{scan}, NaN]$;
end

After locating the obstacle by the mechanical imaging sonar in the ENU coordinate system, the target's origin position $(N_{\mathrm{o}}, E_{\mathrm{o}})$ is defined by

$$\begin{bmatrix} N_{\mathrm{o}} \\ E_{\mathrm{o}} \end{bmatrix} = \begin{bmatrix} N_{\mathrm{AV}} \\ E_{\mathrm{AV}} \end{bmatrix} + \boldsymbol{R_x}(\gamma) \begin{bmatrix} \frac{x_{\mathrm{obs,init}} + x_{\mathrm{obs,end}}}{2} \\ \frac{y_{\mathrm{obs,init}} + y_{\mathrm{obs,end}}}{2} \end{bmatrix}, \tag{17}$$

where $\boldsymbol{R_x}(\gamma)$ is the rotation matrix around x-axis with $\gamma = pi$ [rad]. This matrix is used to translate between ENU to NED coordinate system used for the offshore navigation. The $\boldsymbol{R_x}(\gamma)$ rotation matrix in 2D is defined by

$$\boldsymbol{R_x}(\gamma) = \begin{bmatrix} 1 & 0 \\ 0 & \cos(\gamma) \end{bmatrix}. \tag{18}$$

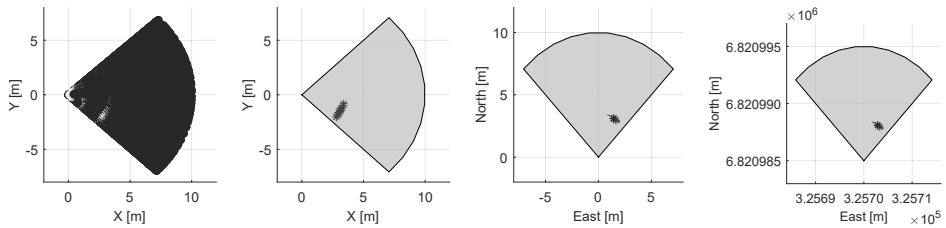
Algorithm 2 includes the detected target localization for the perception sensor data array. This algorithm distinguishes between different targets depending on the consecutive elements in the data array, and the origin position of the targets is sent to the GNC algorithm to proceed with the autonomous navigation of the offshore system.

Algorithm 2: Localization of the detected targets.

Input: *scan* data array in Cartesian coordinates and *RobotPose* (position and orientation).
Output: Obstacle origin [N_o, E_o] calculated in absolute NED coordinates.

```
initialization;
translate scan data from BODY to ENU according to (15);
define consecutive non-NaN elements of the scan data array as same obstacle data;
if obstacle data is non-empty then
    create vector to distinguish between different obstacles;
    define obstacle.x and obstacle.y for the different obstacles detected by the scan;
    define number of obstacles n_obs as equal to number of columns in obstacle.x;
    if n_obs > 0 then
        for i = 1 to n_obs do
            calculate the obstacle origin [N_o(i), E_o(i)] in NED according to (17);
        end
        closely spaced obstacles are defined as same obstacle origin [N_o, E_o];
    end
end
```

Figure 7 shows the steps from the scan data obtained from the mechanical imaging sonar in the BODY reference frame to the final origin position of the detected targets. Figure 7a shows the raw data from the mechanical imaging sonar. Then, Figure 7b shows the post-processing described in Algorithm 1. Finally, Figure 7c,d represents the origin position of the targets in NED coordinate system, with relative to origin [0,0] and absolute coordinates respectively.



(**a**) (**b**) (**c**) (**d**)

Figure 7. Post-processing of the mechanical imaging sonar data in the target detection algorithm: (**a**) Scan data acquired from sonar. (**b**) Post-processing based on Algorithm 1. (**c**) Relative position in NED with origin as [0,0] and calculation of target's origin. (**d**) Absolute position in NED of the targets.

Regarding the USV platform, the SICK MRS1000 LiDAR [28] is the primary perception sensor. This LiDAR has four spread-out scan planes and a multi-echo analysis to be used in harsh environment applications, as it can avoid the noise produced by fog, rain, or dust. Also, this device has a 275° aperture angle, and a working range from 0.2 to 64 m. Thus, in case that the target is above the water surface, it can be detected by the LiDAR sensor.

The algorithm for target detection is similar to the described for the mechanical imaging sonar. The only difference is that the LiDAR contains four spread-out scan planes, acquiring three-dimensional (3D) scan data (see Figure 8a). The target detection algorithm is simplified by translating the received data to 2D by avoiding the z-axis from the sensor data (see Figure 8b). Figure 8c shows the maximum detection range and aperture angle with the scan data in the BODY reference frame. Finally, Figure 8d shows the origin's position of the targets in the NED coordinate system after applying Algorithm 2.

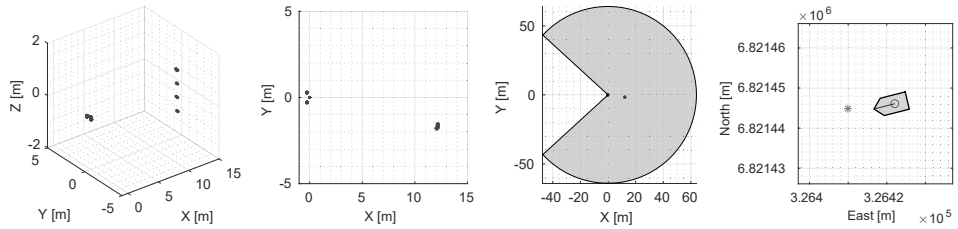


Figure 8. Post-processing of the LiDAR in the target detection algorithm: (**a**) LiDAR scan data in 3D. (**b**) LiDAR scan data in 2D. (**c**) Detection area of the USV in BODY including scan data. (**d**) Absolute position in NED of the targets.

The same procedure detects obstacles from the LiDAR for the path-following with the obstacle avoidance algorithm. After obtaining the origin position $[N_o, E_o]$ from Algorithm 2, the obstacle avoidance algorithm can define a safety boundary box around the obstacle [10].

3.2. Guidance System for Multi-Vehicle System

The multi-vehicle system aims firstly to detect a target using the AUV in a specific offshore area, and after that, sends the location to the USV to do further exploration of the target. Thus, a path-following algorithm is essential for both AUV and USV subsystems. This algorithm intends to reach every waypoint of a specific path independent of time. A commonly used method for path-following is the named LOS guidance, which is chosen as a reference trajectory in this study.

3.2.1. Auv Guidance System

The heading control can use a LOS vector from the AUV position to the next waypoint, similar to [5]. The LOS path-following controller used in this study is the same as the one defined in [10]. However, the AUV movement includes a heave motion, which is avoided by keeping a constant depth for the path-following algorithm. This controller computes the course angle ψ_d based on the path-tangential angle χ_p and the velocity-path relative angle χ_r. The lookahead-based steering can be implemented related to the heading controller applying the transformation defined as

$$\psi_d = \chi_p + \chi_r - \beta, \tag{19}$$

where the variable sideslip (drift) angle β [5] has been omitted in this study to simplify the steering law. The velocity-path relative angle χ_r establishes that the velocity has the direction facing a path location that is in a lookahead distance $\Delta(t) > 0$ along of the direct projection [29]. The path-tangential angle χ_p and the velocity-path relative angle χ_r are defined as

$$\chi_p = \text{atan2}(E_{k+1} - E_k, N_{k+1} - N_k), \tag{20}$$

$$\chi_r(e) = \arctan(-K_P e - K_I \int_0^t e(\tau) d\tau), \tag{21}$$

where (N_k, E_k) and (N_{k+1}, E_{k+1}) are the positions of the passed and next waypoint, respectively, the proportional gain is $K_P = 1/\Delta(t) > 0$, and $K_I > 0$ represents the integral gain. The cross-track error $e(t)$ is given by

$$e(t) = -[N_{AUV}(t) - N_k] \sin(\chi_p) + [E_{AUV}(t) - E_k] \cos(\chi_p). \tag{22}$$

The switching mechanism is declared as a sphere of acceptance for AUVs [30]. This mechanism selects the next waypoint as a lookahead point if the AUV position lies within a sphere with a radius R around the position $(N_{k+1}, E_{k+1}, D_{k+1})$. The sphere of acceptance is defined as

$$[N_{k+1} - N(t)]^2 + [E_{k+1} - E(t)]^2 + [D_{k+1} - D(t)]^2 \leq R_{k+1}^2, \tag{23}$$

where, if the time AUV position $(N(t), E(t), D(t))$ satisfies Equation (23), the next waypoint $(N_{k+1}, E_{k+1}, D_{k+1})$ needs to be selected. Radius R is equal to three AUV lengths L_{AUV} ($R = 3L_{\text{AUV}}$), as the position is only obtained from the USBL system.

After obtaining the course angle from the LOS path-following algorithm, this algorithm sends the heading commands to the yaw controller to match the aimed path. The main control system of the AUV is formed by three separate PID controllers for surge, heave, and yaw motions. Apart from the heading controller, the heave controller keeps the AUV at a constant depth. Their PID parameters for heading controller are obtained by using rapid control prototyping based on the Ziegler-Nichols PID tuning [31] during field tests. Both amplitude K_{zn} and period T_{zn} are calculated for the AUV at the water tank, and then, the PID parameters are defined based on Table 5. Furthermore, a simple proportional controller has been selected in the heave controller. The surge motion is implemented as a constant PWM value to the thrusters.

Table 5. PID parameters for AUV.

Controller	T_{zn} [s]	K_{zn}	K_{P}	K_{I}	K_{D}
Yaw	2.10	5.80	0.580	0.276	0.812
Heave	-	-	300	0.0	0.0
LOS	-	-	0.333	0.0	0.0

3.2.2. USV Guidance System

Same as the AUV guidance system, USV heading control uses a LOS vector from the USV position to the next waypoint. The LOS path-following controller used in this study is the same as the one defined in [10], including the obstacle avoidance capabilities with the safety boundary box approach. The LOS path-following controller of the USV uses the same path-tangential angle χ_{p} defined in Equation (20), the velocity-path relative angle defined in Equation (21), and the total lookahead-based steering from Equation (19). The switching mechanism is selected as a circle of acceptance for surface vehicles [5]. It selects the next waypoint as a lookahead point if the position of the USV lies within a circle with radius R around (N_{k+1}, E_{k+1}). This circle of acceptance is defined as

$$[N_{\text{USV}}(t) - N_{k+1}]^2 + [E_{\text{USV}}(t) - E_{k+1}]^2 \leq R_{k+1}^2, \tag{24}$$

where, if the time surface vehicle position $(N_{\text{USV}}(t), E_{\text{USV}}(t))$ satisfies (24), the next waypoint (N_{k+1}, E_{k+1}) needs to be selected. Radius R is equal to two USV lengths L_{USV} ($R = 2L_{\text{USV}}$).

3.2.3. Multi-Vehicle Guidance System

At the beginning of the control scenario, the USV keeps its position in dynamic positioning (DP) mode while the AUV is trying to search for targets in the coverage area. A DP vessel is a vessel that maintains its position exclusively using active thrusters [24]. This study considers the use of conventional controllers with cascade with low-pass and notch filters to simplify the implementation. The control problem is solved by using PID-controllers for surge, sway, and yaw motions.

The AUV in this study aims to detect a target in a specific offshore area. The coverage area is defined as a set of waypoints to cover a far-reaching range inside. However, this coverage area has been substituted by a straight-path to simplify the control scenario. After detecting the object by the target detection system, it sends a stop command to the AUV, and the vehicle stays in its position until

it received further instructions from the USV. As the AUV does not contain enough instrumentation to have a precise localization of the subsystem, the AUV in this study stops its thrusters instead of having a DP control of its final position. Additionally, if the target detection algorithm does not recognize any target in the coverage area, the AUV stops after reaching the last waypoint of the predefined path.

After receiving the target position by the USV, the path-following algorithm creates the waypoints with a straight-line trajectory. The first waypoint matches the current position of the USV at the time that the target position is received, and the last waypoint is the target position itself. With a constant distance between waypoints of 10 m, the number of waypoints is related to the length of the straight-line path. These waypoints are sent to the LOS path-following algorithm to calculate the course angle of the USV. Furthermore, an additional switching mechanism is included using the same principle as the circle of acceptance defined in (24) to stop the LOS path-following controller once the USV has reached the last waypoint of the predefined path. Then, the guidance system does not send any heading or surge commands to the controllers, and there is no output from the target detection algorithm. In this case, the USV changes to DP internal algorithm keeping its position constant.

Figure 9 shows the priority control level for the multi-vehicle guidance system. First, the AUV starts the path-following of the coverage area based on predefined waypoints. The vehicle continues to the next waypoint until the mechanical imaging sonar detects a target. Then, the AUV stops its operation, and the target position is transmitted to the USV. The USV keeps its position in DP mode and, when the target position is received, it starts the path-following with obstacle avoidance operation with the target position as the final waypoint of the USV trajectory. After reaching the last waypoint, the USV stops and uses the DP mode to keep its position, allowing the GCS to have further inspection of the detected target. Additionally, the steering wheel and 3-axis joystick, both forming the manual control of the USV, provides the safety feature in the autonomous algorithm.

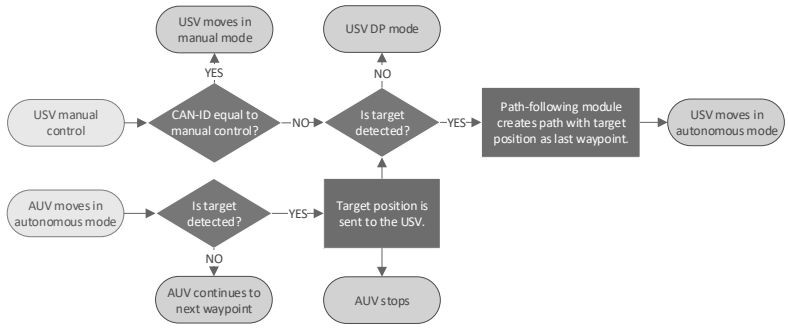


Figure 9. Stateflow diagram for priority control level in the multi-vehicle guidance system. The target detection algorithm at the AUV enables the autonomous operation of the USV.

4. Experimental Validation

4.1. System Implementation

For this particular study, the USV and AUV platforms incorporate multiple mechatronic systems to implement the target detection algorithm. Both vehicles include high-level control (computers with ROS), which performs complex computations and processes the data obtained from localization and perception sensors, and low-level control (sensors and actuators units), that runs as the basic interface for vehicle operations. Also, an intermediate-level (or mid-level) control is included, which is the main link between low-level data acquisition and high-level logic operations.

Figure 10 shows the mechatronic systems used in the USV, including also the connection to the AUV and external MATLAB-Simulink computer through the main network switch. These devices are the link to the co-operative autonomous offshore system. In general, the USV platform is equipped with a payload for navigation (high precision GPS-Compass), LiDAR as the main perception sensor, SeaTrac

acoustic system for USBL localization, and communication with the AUV, and WiFi for communication with the GCS. The USV system implementation is the same as the one studied in [10]. For the high-level control, the ROS master includes the necessary stand-alone ROS-nodes for the path-following with obstacle avoidance. The display computers act as intermediate-level control for translation between CAN bus and ROS messages. Also, they are in charge of sending joystick commands to the waterjet control units based upon priority levels.

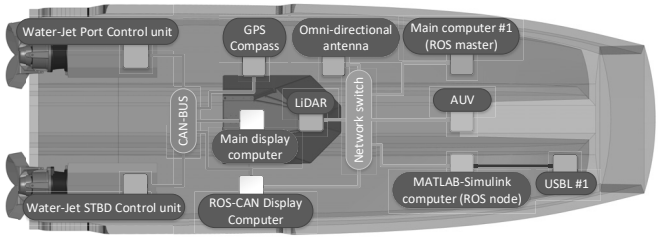


Figure 10. System overview of the USV platform with high-level (blue boxes),intermediate-level (white boxes), and low-level control (purple boxes), including the connection to the AUV platform (adapted from [10]).

Figure 11 shows the mechatronic systems used in the AUV platform. The AUV is connected to the USV via a neutrally buoyant tether to have a direct connection between the vehicles. Similarly to the USV platform, the AUV contains high-level control with the ROS computer and an intermediate-level control as a bridge between the main ROS computer and the companion computer, which communicates using the MAVLink protocol. The low-level control includes actuators and sensors, formed by six thrusters and their respective electronic speed controllers (ESCs), a pressure sensor for depth measurements, a mechanical imaging sonar as the perception sensor, and the USBL SeaTrac acoustic system for positioning and communication. Finally, the AUV includes a companion computer with the flight controller and the ROS computer (Linux computer) connected to a network switch. The ROS computer performs the complex computations for autonomous operation and target detection.

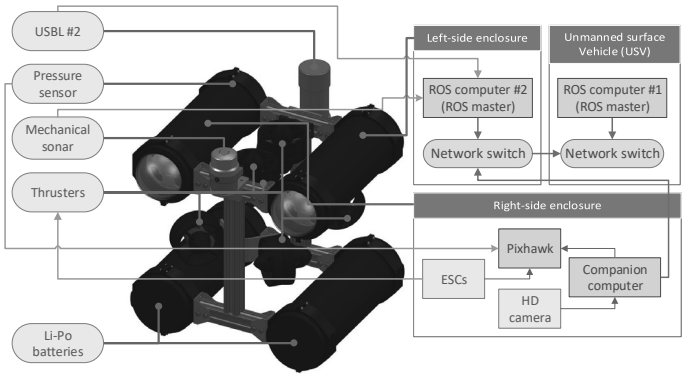


Figure 11. System overview of the AUV: High-level (Robot Operating System (ROS) computers), intermediate-level (companion computer and Pixhawk flight controller), and low-level control (thrusters, ultra-short baseline (USBL), pressure sensor, and mechanical imaging sonar).

The approach used in this study for the multi-robot architecture is multimaster-fkie, which provides simplicity and ROS compatibility [21]. This package is a fully compatible multi-master implementation for topic and services transactions. Nevertheless, this implementation can cause some drawbacks due to the continuous master state scanning and the delay between changes in advertising, as well as

information exchange. As this study requires a total of three ROS topics, this package is useful as an easy plug-and-play solution.

Figure 12 illustrates the communication between the USV and AUV platforms, including the nodes for the multimaster-fkie architecture. The exchanged topics are */target*, which is the position of the detected target, */usv_gps* obtained from the USV GPS-compass and used to get the absolute Cartesian coordinates of the AUV position, and */usv_heading* which rotates the USBL coordinate system according to the heading of the USV. The diagram also includes the links between the high-level, mid-level, and low-level control in both platforms.

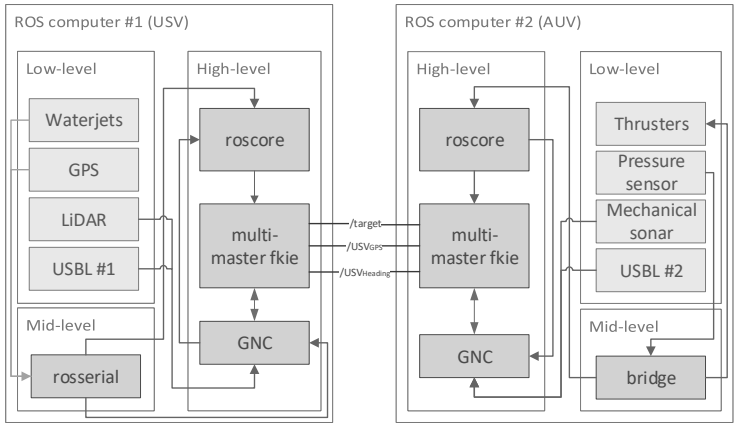


Figure 12. Communication of the autonomous offshore system based on the multimaster-fkie architecture. Each vehicle shows the internal connection between the sensors and actuators with the rest of the system.

4.2. Modular System for Multi-Sensor Technology

The target detection algorithm uses a modular approach to include target detection from each perception sensor, path-following, and guidance control from both USV and AUV platforms. Each of these modules runs a separate ROS node in the autonomous offshore system. This approach has been previously studied and successfully implemented in [10,32]. However, the algorithms of the mentioned studies did not include co-operative capabilities between multiple autonomous vehicles.

Figure 13 illustrates the modular architecture with all topics involved, defining the subscribers and publishers of each topic. The only difference between the two vehicles is the path-following model at the USV for obstacle avoidance, which is in charge of modifying the USV trajectory using the safety boundary box approach.

The GPS-Compass obtains the absolute position of the USV in global coordinates, while the USBL collects the position of the AUV in the BODY reference frame of the USBL. The ROS topic */odometry* in the AUV is based on the low-level serial messages accepted and generated by the SeaTrac USBL beacons [16]. These serial messages are ASCII-Hex characters of the message string, which are decoded into an array of bytes representing their values. The ROS topic is generated using the Serial package [33], which translates the RS232 messages to a ROS topic array. After that, PING messages are sent from the main USBL #1 beacon located at the USV, and the response from the AUV (USBL #2) produces the necessary serial messages containing the AUV position in the BODY USBL coordinate system. Finally, the change from this reference frame to the NED coordinate system is defined by the combination of a translation and a rotation matrix. These matrices use the initial heading of the USBL and the */heading* and */gps* variables from the GPS-Compass.

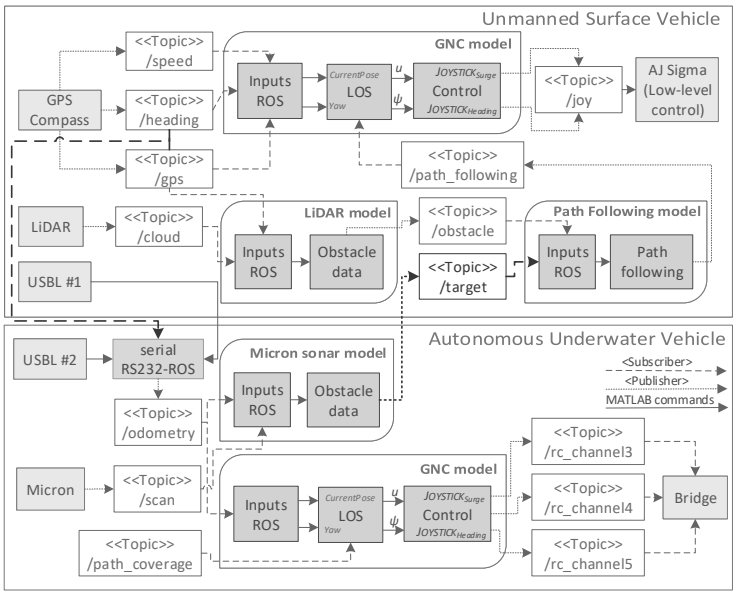


Figure 13. Schematic of the modular multi-vehicle guidance system with target detection. All different modules from USV and autonomous underwater vehicle (AUV) were included. ROS topics */gps*, */heading* and */target* (purple connectors) are the exchange topics in the control scenario.

The predefined path for the AUV is defined as the ROS topic */path_coverage*, which includes the waypoints for the GNC algorithm in the control module. The GNC guidance algorithm generates the required AUV heading command, sending this parameter to the AUV controller. The controller generates the required inputs */rc_channel*3, */rc_channel*4, */rc_channel*5, and sends them to the companion computer for surge, heave, and yaw motions, respectively, based on the BlueRov-ROS-playground ROS package [34].

Regarding the USV, the exchanged ROS topic */target* contains the target's origin position. Thus, once this topic is received in the path-following model, it defines the necessary waypoints to perform the autonomous mission. These waypoints are sent to the GNC model, where the LOS-algorithm calculates the required course angle for the controller. Finally, the controller generates the required joystick commands for surge $/Joy_u$ and yaw $/Joy_r$ to reach the LOS values. These joystick commands are sent to the low-level control (display computers) to perform the autonomous USV operation, using the same outputs as a manual three-axis joystick.

4.3. Experimental Results

The control scenario for this study includes target detection, path-planning, and guidance control in both offshore vehicles. However, even though the modular ROS architecture provides a computationally cheap and easy implementation in both offshore platforms, the operation of both platforms in an offshore scenario depends highly on environmental elements such as wind or wave drift forces. As the guidance control bases its operation on simple PID controllers without the compensation of these environmental elements, it makes it highly challenging to gather useful field test data from the offshore system. Thus, the experimental results of this study are shown in a modular way, testing each of the subsystems separately to validate the target detection algorithm using multi-sensor technology. Figure 14a illustrates the location for the AUV and USV field tests at the Pyhäjärvi lake in Tampere, Finland. The water-flow direction from a hydro-power plant is also defined to show the environmental drift forces. Figure 14b shows the implementation for the AUV path-following, where the USV stays

stationary at the harbor. Regarding the USV field test, it is demonstrated in a clear obstacle area at the lake.

(**a**) (**b**)

Figure 14. Control scenario: (**a**) Location of the AUV (red arrow) and USV (green arrow) field tests at the Pyhäjärvi lake in Tampere, Finland, being affected by the water-flow (blue arrow) from a hydro-power plant. (**b**) AUV and USV platforms at the harbor during the AUV field tests.

The first step in the target detection algorithm is the AUV path-following. This module is tested at the harbor with a set of three waypoints defined in the NED coordinate system. The surge motion has a constant PWM value to the thrusters, and the yaw and heave motions are implemented using separate PID controllers. The LOS-based guidance system calculates the necessary course angle to reach every waypoint of the predefined path. Figure 15 shows the AUV trajectory using the USBL data for navigation, where the AUV initial position and orientation are defined as random. The AUV moves slightly to the left side of the path-following due to the environmental drift forces. As it is shown in Figure 14a, the field tests have been done in an estuary area of a narrow and shallow lake, where the flow from a hydro-power plant affects considerably. These flow conditions vary depending on the river discharge rate. During the time of testing, the river discharge was 38 m^3/s to the south direction, and the wind speed was equal to 6 m/s with southwest wind direction.

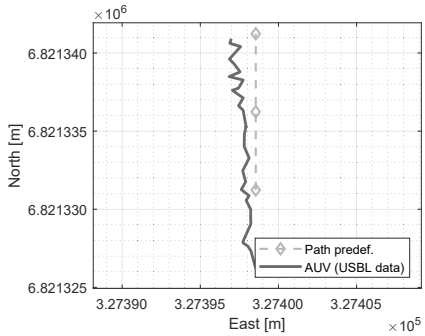


Figure 15. AUV Control scenario: AUV trajectory for the path-following algorithm.

Figure 16a shows the comparison between the input control values for the yaw angle and the field test data, and Figure 16b displays the same comparison for heave motion. In this case, the multi-vehicle system contributes to the GPS-Compass data at the USV, providing the ROS topics */gps* and */heading* to the USBL acoustic system for positioning.

During the implementation of the GNC model, the target detection algorithm processes the mechanical imaging sonar data to detect and locate any possible obstacle around the AUV. Figure 7

illustrates the adequate performance of this module, where a static obstacle (buoy) is detected and located in absolute NED coordinates.

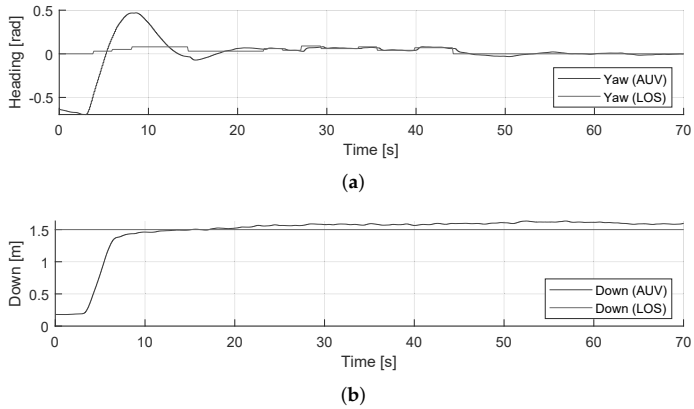


Figure 16. AUV Control scenario: (**a**) Comparison of heading angle from the LOS guidance system with field-test data. (**b**) Comparison of the constant depth of 1.5 m with field-test data.

Once the AUV detects and locates the target, it sends the target's position to the USV platform via multimaster-fkie architecture. The last control scenario in the experimental results demonstrates the co-operative autonomous offshore system with the path-following with obstacle avoidance capabilities of the USV. The USV main computer receives the */target* ROS topic from the AUV main computer. Then, the GNC model provides the necessary surge and yaw motions to reach the target's position based on the LiDAR and path-following models. Figure 17 shows the USV trajectory once the path has been defined according to the ROS topic */target*. Additionally, Figure 18 shows the comparison between the LOS guidance system and the field test data for yaw motion, and Figure 19 shows the corresponding LOS cross-track error $e(t)$, which demonstrates the correct performance of the guidance control, even though environmental variables are not considered in this study. During the USV field tests, the river discharge was 30 m^3/s to the south direction, and the wind speed was equal to 3.7 m/s with south-southwest wind direction.

The experimental results of this study indicate the correct performance of the target detection algorithm using multi-sensor technology. These results are implemented in a modular way, and they show the appropriate implementation of each model, including target detection, path-following, and guidance control. The path-following algorithms in the AUV and USV platforms include some error due to the environmental variables, such as wind and wave drift forces. These variables need to be considered to increase the accuracy of the system, and they can be removed by improving the GNC controllers. Furthermore, the AUV navigation includes only the USBL beacons for positioning, which is not able to locate precisely the vehicle underwater. By improving the navigation system, the path-following algorithm will enhance its performance.

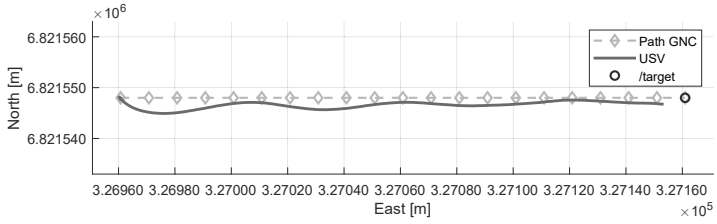


Figure 17. USV Control scenario: USV trajectory for the path-following algorithm, where the last waypoint is equal to the ROS topic */target*.

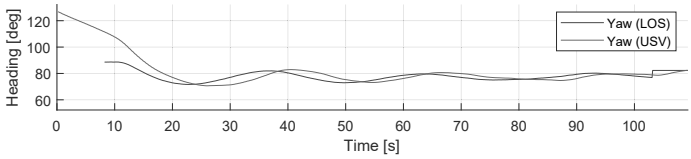


Figure 18. USV Control scenario: Comparison of heading angle from the LOS guidance system with field-test data. After reaching the $/target$ position, the yaw angle is equal to the constant velocity-path relative angle χ_r for DP mode.

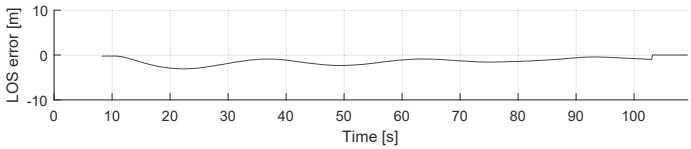


Figure 19. USV Control scenario: LOS cross-track error $e(t)$ for the lookahead-based steering law defined in (22). This error is produced by the environmental variables, as the drift angle β is not included in the LOS-based guidance control.

5. Conclusions and Future Work

This article was concerned with the target detection using multi-sensor technology in a co-operative autonomous offshore system. The offshore system had a USV and an AUV, and the fundamental purpose of the algorithm was to detect an underwater target in a preplanned coverage area. The mathematical model of the USV, including also the waterjet propulsion system model, was presented to verify the designed GNC architecture. This model included parameter estimation methods to obtain the dynamic coefficients using field test data for both surge and yaw motions. This study developed a basic target detection algorithm for any offshore perception sensors, showing the results for a mechanical imaging sonar at the AUV and a LiDAR at the USV. The guidance system included the LOS model for path-following on both platforms. After designing the GNC architecture, both vehicles incorporated a system implementation of the modular approach with high, intermediate, and low-level controls. The experimental results showed a field test control scenario that presents the capabilities and adequate performance of the target detection algorithm.

Future work will include an accurate mathematical model of the AUV for simulation, which requires the complete navigation data (position, velocity, and acceleration feedback) from the vehicle. Additionally, the coverage path planning can replace the straight-line trajectory of this study, having more coverage area and increasing the capabilities of the system. The AUV scenario will include the capabilities of making decisions in the presence of several obstacles, and further navigational sensors will be installed for more precise localization of the AUV (e.g., DVL). Finally, future work will also include additional platforms into the system, as it could be other USV or AUV, or even a UAV, which would increase the capabilities of the system working in the air.

Author Contributions: J.V. conceptualized and designed the methodology, developed the software and validation of the model, performed the experiments, analyzed the data, and wrote the paper; S.V. performed the experiments; J.A. and K.T.K. supervised the study and made the writing—review and editing of this paper. All authors have read and agreed to the published version of the manuscript.

Funding: This research is based on the Autonomous and Collaborative Offshore Robotics (aColor) project, funded by the Technology Industries of Finland Centennial and Jane & Aatos Erkko Foundations.

Acknowledgments: The authors would like to thank the contributions from Alamarin-Jet Oy for facilitating their research surface vehicle as platform in this study.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

USV	unmanned surface vessel
AUV	autonomous underwater vehicle
SAR	search and rescue
UAV	unmanned aerial vehicle
DOF	degree of freedom
SI	system identification
DR	dead reckoning
INS	inertial navigation systems
DVL	doppler velocity log
AHRS	attitude and heading reference system
USBL	ultra-short baseline
LBL	long baseline
EKF	extended Kalman filter
IMU	inertial measurement unit
GNC	guidance, navigation, and control
LOS	line of sight
ROS	robot operating system
GCS	ground control station
AV	autonomous vehicle
STDB	starboard
NED	North-East-Down
FoV	field of view
ENU	East-North-Up
PID	proportional-integral-derivative
DP	dynamic positioning
ESC	electronic speed controller
1D	one-dimensional
2D	two-dimensional
3D	three-dimensional

References

1. Vasilijević, A.; Nađ, Đ.; Mandić, F.; Mišković, N.; Vukić, Z. Coordinated navigation of surface and underwater marine robotic vehicles for ocean sampling and environmental monitoring. *IEEE/ASME Trans. Mechatron.* **2017**, *22*, 1174–1184. [CrossRef]
2. Ross, J.; Lindsay, J.; Gregson, E.; Moore, A.; Patel, J.; Seto, M. Collaboration of multi-domain marine robots towards above and below-water characterization of floating targets. In Proceedings of the 2019 IEEE International Symposium on Robotic and Sensors Environments (ROSE), Ottawa, ON, Canada, 17–18 June 2019; pp. 1–7.
3. Gu, N.; Peng, Z.; Wang, D.; Shi, Y.; Wang, T. Antidisturbance Coordinated Path Following Control of Robotic Autonomous Surface Vehicles: Theory and Experiment. *IEEE/ASME Trans. Mechatron.* **2019**, *24*, 2386–2396.
4. Pham, H.A.; Soriano, T.; Ngo, V.H.; Gies, V. Distributed Adaptive Neural Network Control Applied to a Formation Tracking of a Group of Low-Cost Underwater Drones in Hazardous Environments. *Appl. Sci.* **2020**, *10*, 1732. [CrossRef]
5. Fossen, T.I. *Handbook of Marine Craft Hydrodynamics and Motion Control*; John Wiley & Sons: New York, NY, USA, 2011.
6. The MathWorks, Inc. *Simulink Design Optimization User's Guide*; Release 2020a; The MathWorks, Inc.: Natick, MA, USA, 2009.
7. Ljung, L. System identification. *Wiley Encycl. Electr. Electron. Eng.* **1999**, 1–19. [CrossRef]
8. The MathWorks, Inc. *System Identification Toolbox User's Guide*; Release 2020a; The MathWorks, Inc.: Natick, MA, USA, 1988.

9. Klinger, W.B.; Bertaska, I.R.; von Ellenrieder, K.D.; Dhanak, M.R. Control of an unmanned surface vehicle with uncertain displacement and drag. *IEEE J. Ocean. Eng.* **2016**, *42*, 458–476. [CrossRef]
10. Villa, J.; Aaltonen, J.M.; Koskinen, K.T. Path-Following with LiDAR-based Obstacle Avoidance of an Unmanned Surface Vehicle in Harbor Conditions. *IEEE/ASME Trans. Mechatron.* **2020**, 1812–1820. [CrossRef]
11. Villar, S.A.; Solari, F.J.; Menna, B.V.; Acosta, G.G. Obstacle detection system design for an autonomous surface vehicle using a mechanical scanning sonar. In Proceedings of the 2017 XVII Workshop on Information Processing and Control (RPIC), Mar del Plata, Argentina, 20–22 September 2017; pp. 1–6.
12. Ganesan, V.; Chitre, M.; Brekke, E. Robust underwater obstacle detection and collision avoidance. *Auton. Robot.* **2016**, *40*, 1165–1185. [CrossRef]
13. Leonard, J.J.; Bennett, A.A.; Smith, C.M.; Jacob, H.; Feder, S. *Autonomous Underwater Vehicle Navigation*; MIT Marine Robotics Laboratory Technical Memorandum: Boston, MA, USA, 1998.
14. Ribas, D.; Palomeras, N.; Ridao, P.; Carreras, M.; Mallios, A. Girona 500 auv: From survey to intervention. *IEEE/ASME Trans. Mechatron.* **2011**, *17*, 46–53. [CrossRef]
15. Miller, P.A.; Farrell, J.A.; Zhao, Y.; Djapic, V. Autonomous underwater vehicle navigation. *IEEE J. Ocean. Eng.* **2010**, *35*, 663–678. [CrossRef]
16. Neasham, J.A.; Goodfellow, G.; Sharphouse, R. Development of the "Seatrac" miniature acoustic modem and USBL positioning units for subsea robotics and diver applications. In Proceedings of the OCEANS 2015-Genova, Genoa, Italy, 18–21 May 2015; pp. 1–8.
17. Font, E.G.; Bonin-Font, F.; Negre, P.L.; Massot, M.; Oliver, G. USBL integration and assessment in a multisensor navigation approach for AUVs. *IFAC-PapersOnLine* **2017**, *50*, 7905–7910. [CrossRef]
18. Breivik, M.; Fossen, T.I. Path following for marine surface vessels. In Proceedings of the Oceans' 04 MTS/IEEE Techno-Ocean'04 (IEEE Cat. No. 04CH37600), Kobe, Japan, 9–12 November 2004; Volume 4, pp. 2282–2289.
19. Quigley, M.; Conley, K.; Gerkey, B.; Faust, J.; Foote, T.; Leibs, J.; Wheeler, R.; Ng, A.Y. ROS: An open-source Robot Operating System. In Proceedings of the ICRA Workshop on Open Source Software, Kobe, Japan, 12–17 May 2009; Volume 3, p. 5.
20. Børhaug, E.; Pavlov, A.; Panteley, E.; Pettersen, K.Y. Straight line path following for formations of underactuated marine surface vessels. *IEEE Trans. Control Syst. Technol.* **2010**, *19*, 493–506. [CrossRef]
21. Tiderko, A.; Hoeller, F.; Röhling, T. The ROS multimaster extension for simplified deployment of multi-robot systems. In *Robot Operating System (ROS)*; Springer: Berlin/Heidelberg, Germany, 2016; pp. 629–650.
22. Ollikainen, M.; Ollikainen, M. *The Finnish Coordinate Reference Systems*; Finnish Geodetic Institute and the National Land Survey of Finland: Masala, Finland, 2004. Available online: https://www.maanmittauslaitos.fi/sites/maanmittauslaitos.fi/files/old/Finnish_Coordinate_Systems.pdf (accessed on 11 December 2020).
23. Sagatun, S.I.; Fossen, T.I. Lagrangian formulation of underwater vehicles' dynamics. In Proceedings of the Conference Proceedings 1991 IEEE International Conference on Systems, Man, and Cybernetics, Charlottesville, VA, USA, 13–16 October 1991; pp. 1029–1034.
24. Fossen, T.I. *Guidance and Control of Ocean Vehicles*; John Wiley & Sons: New York, NY, USA, 1994.
25. g Alamarin Jet Oy. AJ 245. Available online: https://alamarinjet.com/products/jet/aj-245/ (accessed on 11 December 2020).
26. Tritech International Ltd. *Micron Sonar—Product Manual*; 0650-SOM-00003, Issue: 02; Westhill: Aberdeenshire, UK, 2020.
27. Fossen, T.; Ross, A. Nonlinear modelling, identification and control of UUVs. *IEE Control Eng. Ser.* **2006**, *69*, 13.
28. Sick AG. *MRS1000: Operating Instructions*; 8020494/12FY/2019-04-02; Sick AG: Waldkirch, Germany, April 2019.
29. Papoulias, F.A. Bifurcation analysis of line of sight vehicle guidance using sliding modes. *Int. J. Bifurc. Chaos* **1991**, *1*, 849–865. [CrossRef]
30. Healey, A.J.; Lienard, D. Multivariable sliding mode control for autonomous diving and steering of unmanned underwater vehicles. *IEEE J. Ocean. Eng.* **1993**, *18*, 327–339. [CrossRef]
31. Ziegler, J.G.; Nichols, N.B. Optimum settings for automatic controllers. *Trans. ASME* **1942**, *64*, 759–768. [CrossRef]
32. Villa, J.; Aaltonen, J.; Koskinen, K.T. Model-based path planning and obstacle avoidance architecture for a twin jet Unmanned Surface Vessel. In Proceedings of the 2019 Third IEEE International Conference on Robotic Computing (IRC), Naples, Italy, 25–27 February 2019; pp. 427–428.

33. Woodall, W.; Harrison, J. Serial, Cross-Platform, Serial Port Library Written in C++. Available online: http://wjwwood.io/serial/ (accessed on 11 December 2020).
34. Pereira, P.J. BlueRov-ROS-Playground. Github Repository. Available online: https://github.com/patrickelectric/bluerov_ros_playground (accessed on 11 December 2020).

PUBLICATION V

Model-Validation and Implementation of a Path-following Algorithm in an Autonomous Underwater Vehicle

Jose Villa, Guillem Vallicrosa, Jussi Aaltonen, Pere Ridao, Kari T. Koskinen

Unpublished manuscript (2021)