Motion Control of Single and Multiple Autonomous Marine Vehicles

Jorge Miguel dos Santos Ribeiro

A thesis submitted to the Department of Electrical and Computer Engineering in partial fulfilment of the requirements for the degree of

Master of Science in Electrical and Computer Engineering

President: Doutor Carlos Filipe Gomes Bispo
Advisor: Doutor António Pedro Rodrigues de Aguiar
Co-Advisor: Doutor António Manuel dos Santos Pascoal
Examiner: Doutor Miguel Afonso Dias de Ayala Botto

October 2011
Abstract

Worldwide, in the field of ocean exploration, there has been a surge of interest in the development of autonomous marine robotic vehicles equipped with advanced systems to steer them accurately and reliably in the harsh marine environment and allow them to collect data at the surface and underwater.

Motivated by this fact, the first part of this dissertation proposes motion control algorithms (heading, speed, positioning, way-point and path-following controllers) for single marine vehicles. For the path-following controller, the solution proposed is based on an outer loop control structure designed at the kinematic level responsible for computing the heading commands to make the vehicle move along a desired path, and an inner loop that regulates the actuators so that a given heading reference signal is tracked. Simulation and experimental results with the Medusa vehicles illustrate the proposed control systems.

The second part of the thesis is dedicated to the cooperative control of multiple marine vehicles. The cooperation of multiple vehicles offers several advantages and leads to safer, faster, and far more efficient ways of exploring the ocean frontier, especially in hazardous conditions.

The third part of the thesis focuses on practical issues such as the implementation of the proposed algorithms in the vehicles, presents the software architecture adopted, and describes some graphical tools to supervise and manage the sea tests.

Keywords: Autonomous Marine Robotic Vehicles, Motion Control, Path following, Cooperative Control, Experimental tests.
Resumo

Em todo o mundo, no domínio da exploração do oceano, tem-se registado um crescente interesse no desenvolvimento de veículos marinhos autónomos equipados com sistemas avançados com o intuito de dirigi-los com precisão e confiança no inóspito ambiente marinho e permitir a recolha de dados da superfície e debaixo de água.

A primeira parte desta dissertação propõe-se algoritmos de controlo de movimento (controladores de orientação, velocidade, posição e caminho) para veículos marinhos individuais. No controlador de caminho, a solução proposta é baseada numa estrutura de malha de controlo externa, projetada ao nível da cinemática responsável por calcular as referências de orientação para os veículos de modo a que estes se movam ao longo de um caminho desejado, e uma malha interna que regula os atuadores de modo a que seja seguido o sinal de referência. São apresentados resultados de simulação e experimentais com os veículos "Medusa" que ilustram os sistemas de controlo propostos.

A segunda parte da tese é dedicada ao controlo cooperativo de múltiplos veículos marinhos. Esta cooperação oferece várias vantagens e leva a formas mais seguras, rápidas e muito mais eficientes de explorar o oceano, especialmente em condições perigosas.

A terceira parte da tese concentra-se em questões práticas, tais como a implementação dos algoritmos propostos nos veículos, apresenta a arquitetura de software e descreve algumas ferramentas gráficas de supervisão para auxiliar nos testes de mar.

Palavras-chave: Veículos marinhos autónomos, Controlo Autónomo de um Veículo, Seguidor de Caminho, Controlo Cooperativo, Testes Experimentais.
Acknowledgments

For the realization of this thesis everyone was important, especially those who stood by my side during the last semester, however, and not wanting to belittle anyone, there are some names that are essential to highlight. So, starting by thanking Professor António Pedro Aguiar that from his extensive knowledge in the area, and his huge commitment in teaching it, provided this work.

Not less important was the influence of Professor António Manuel Pascoal who, with his great experience guided this study/work in the right direction.

I also would like to thank with special attention to my project and team colleagues, João Botelho, Luís Sebastião and Manuel Rufino for their patience in giving suggestions and motivating me in some difficult situations.

To all my course colleagues who helped me in these last years, both in academic and emotional way, thanks. Taking in particular consideration Catarina Pais, André Silva, Henrique Silva and Pedro Santos.

I can not forget the important force given by my parents and brother, Teodoro Ribeiro, Maria Marques and André Ribeiro that allowed the completion of a college degree and supported me in all phases of the course, stressing the help of my brother in the end of the dissertation.
Agradecimentos

Para a realização da presente dissertação todas as pessoas foram importantes, principalmente as que estiveram do meu lado no decorrer do último semestre; contudo, e não querendo menosprezar ninguém, há alguns nomes que são essenciais de realçar. Assim, começo por agradecer ao professor António Pedro Aguiar que, a partir do seu abrangente conhecimento na área, e um enorme empenho em transmiti-lo, proporcionou a realização deste trabalho.

Não menos importante foi a influência do professor António Manuel Pascoal que com a sua grande experiência guiou este estudo/trabalho na direcção para o caminho certo.

Gostava também de agradecer com especial atenção aos meus colegas de projecto e equipa, João Botelho, Luís Sebastião e Manuel Rufino pela sua grande paciência, dando sugestões e motivando-me em algumas das situações mais difíceis.

A todos os meus colegas de curso que me ajudaram nestes últimos anos tanto ao nível académico como emocional, um obrigado. Neste contexto e por terem tido um papel especial gostaria de destacar a Catarina Pais, o André Silva, o Henrique Silva e o Pedro Santos.

Não me posso esquecer da importante força dada pelos meus pais e irmão; Teodoro Ribeiro, Maria Marques e André Ribeiro que permitiram a conclusão de um curso superior e me apoiaram em todas as fases do percurso; salientando a ajuda do meu irmão na conclusão da dissertação.
List of Figures

1.1 Coordinated Operation of the DELFIM ASC and the INFANTE AUV for scientific missions in the Azores (Source: [13]). ........................................ 2
1.2 Marine Habitat Mapping using multiple vehicles in Azores (Source: [5]). . . 3
1.3 Diver localization and guidance (Left Source: Artist - Dinsa Cunha Silva, 2008; Right Source: [1]). ............................................................... 4

2.1 Body-fixed and Inertial reference frames. ............................................ 10
2.2 MEDUSA dimensions. ................................................................. 15
2.3 The MEDUSA vehicle in different views. ........................................... 16
2.4 Three MEDUSAs in Sesimbra after a full day of trials. ......................... 17
2.5 Thruster Model Diagram. ............................................................. 17
2.6 Thruster characterization real data versus simulated data with different zoom levels - Taguspark test tank. ............................................... 18
2.7 Surge velocity model analysis: simulated and experimental results. ......... 19
2.8 Yaw velocity model analysis: simulated and experimental results. ........ 20

3.1 Heading Controller simulation and experimental tests. ......................... 22
3.2 Implemented filter for position and velocity estimation. ....................... 24
3.3 Observer output performance. ....................................................... 24
3.4 Inertial and Body velocity vectors. ................................................ 25
3.5 Speed Controller simulation test. .................................................... 26
3.6 Smith and Speed controller Diagram. ............................................. 27
3.7 Comparing the Smith Predictor performance. .................................... 27
3.8 Importance of $k_u$ parameter ($k_u = 0.5$). ................................... 28
3.9 Waypoint Diagram. ................................................................. 29
<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.10</td>
<td>Performance of the way-point Controller.</td>
<td>29</td>
</tr>
<tr>
<td>3.11</td>
<td>Waypoint Controller - Simulation test with high current.</td>
<td>30</td>
</tr>
<tr>
<td>3.12</td>
<td>Path-following nomenclature for a straight line.</td>
<td>31</td>
</tr>
<tr>
<td>3.13</td>
<td>Path-following for a straight line along x axis.</td>
<td>32</td>
</tr>
<tr>
<td>3.14</td>
<td>Path-following for a lawn mower maneuver.</td>
<td>32</td>
</tr>
<tr>
<td>3.15</td>
<td>Path-following for a lawn mower maneuver in ”Parque das Nações” dock.</td>
<td>33</td>
</tr>
<tr>
<td>4.1</td>
<td>Vehicle general control architecture.</td>
<td>36</td>
</tr>
<tr>
<td>4.2</td>
<td>Line Mission with a formation in line.</td>
<td>39</td>
</tr>
<tr>
<td>4.3</td>
<td>Lawn-mower Mission with a formation in triangle.</td>
<td>39</td>
</tr>
<tr>
<td>4.4</td>
<td>Lawn-mower Mission with a formation in line.</td>
<td>40</td>
</tr>
<tr>
<td>5.1</td>
<td>Communications between all the processes involved.</td>
<td>42</td>
</tr>
<tr>
<td>5.2</td>
<td>Hardware in the loop Configuration.</td>
<td>44</td>
</tr>
<tr>
<td>5.3</td>
<td>Browser Console for Sesimbra Trial doing CPF.</td>
<td>47</td>
</tr>
<tr>
<td>A.1</td>
<td>Waypoint Controller - Simulation test with high current.</td>
<td>51</td>
</tr>
<tr>
<td>A.2</td>
<td>CPF Controller - Simulation test with high current.</td>
<td>52</td>
</tr>
<tr>
<td>B.1</td>
<td>Vehicles performing a Lawn Mower maneuver for bathymetric data acquisition</td>
<td>54</td>
</tr>
<tr>
<td>B.2</td>
<td>Vehicles performing a circle for acoustic data acquisition</td>
<td>55</td>
</tr>
<tr>
<td>B.3</td>
<td>Vehicles doing intersected lawn mowers for video recording</td>
<td>56</td>
</tr>
</tbody>
</table>
# List of Tables

<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>Notation used for marine vehicles.</td>
<td>11</td>
</tr>
<tr>
<td>2.2</td>
<td>Physical Properties.</td>
<td>18</td>
</tr>
<tr>
<td>2.3</td>
<td>Normalized model parameters.</td>
<td>19</td>
</tr>
<tr>
<td>4.1</td>
<td>Communication topology for three vehicles where only 2 and 3 receives data</td>
<td>38</td>
</tr>
<tr>
<td></td>
<td>from 1.</td>
<td></td>
</tr>
<tr>
<td>4.2</td>
<td>Communication topology for three vehicles where everyone knows each other</td>
<td>38</td>
</tr>
<tr>
<td></td>
<td>position.</td>
<td></td>
</tr>
</tbody>
</table>
## List of Abbreviations

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AHRS</td>
<td>Attitude and heading reference system</td>
</tr>
<tr>
<td>AMV</td>
<td>Autonomous Marine Vehicle</td>
</tr>
<tr>
<td>AUV</td>
<td>Autonomous Underwater Vehicle</td>
</tr>
<tr>
<td>ASC</td>
<td>Autonomous Surface Craft</td>
</tr>
<tr>
<td>CPF</td>
<td>Cooperative Path Following</td>
</tr>
<tr>
<td>DOF</td>
<td>Degree of Freedom</td>
</tr>
<tr>
<td>GPS</td>
<td>Global Positioning System</td>
</tr>
<tr>
<td>IMU</td>
<td>Inertial Measurement Unit</td>
</tr>
<tr>
<td>IP</td>
<td>Internet Protocol</td>
</tr>
<tr>
<td>PD</td>
<td>Proportional-Derivative</td>
</tr>
<tr>
<td>PI</td>
<td>Proportional-Integral</td>
</tr>
<tr>
<td>RPM</td>
<td>rotations per minute</td>
</tr>
<tr>
<td>UDP</td>
<td>User Datagram Protocol</td>
</tr>
</tbody>
</table>
Contents

Abstract iii

Resumo v

Acknowledgments vii

Agradecimentos ix

List of Figures xii

List of Tables xiii

List of Abbreviations xv

1 Introduction 1

1.1 Background . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 1

1.2 Research Objectives . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 4

1.3 Main Contributions . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 5

1.3.1 Vehicle Modelling . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 6

1.3.2 Control of a single vehicle . . . . . . . . . . . . . . . . . . . . . . . . 6

1.3.3 Cooperative control of multiple vehicles . . . . . . . . . . . . . . . . 6

1.3.4 Implementation . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 6

1.4 Thesis Outline . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 7

2 Autonomous Underwater Vehicle Model 9

2.1 Coordinate Frames . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 9

2.2 Vehicle Model . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 11

2.2.1 Kinematic Equations . . . . . . . . . . . . . . . . . . . . . . . . . . . . 11
# CONTENTS

2.2.2 Dynamic Equations ................................................. 12
2.2.3 Simplified Equations of Motion ................................. 13
2.3 Vehicle Characterization - The Medusa .......................... 14
  2.3.1 SEABOTIX Thruster Model ...................................... 17
  2.3.2 Vehicle Model parameters ...................................... 18

3 Single Vehicle Motion Control ....................................... 21
  3.1 Heading Controller .................................................. 21
  3.2 Speed Controller .................................................... 22
    3.2.1 Smith Predictor .................................................. 26
  3.3 Way-point and Hold Position Controller ......................... 27
  3.4 Path-Following Controller ......................................... 30

4 Multiple Vehicle Motion Control ................................... 35
  4.1 General architecture for multiple vehicle cooperation ........... 35
  4.2 Cooperative Path-Following ......................................... 36
    4.2.1 Results .......................................................... 39

5 Implementation .......................................................... 41
  5.1 Software Architecture ............................................... 41
  5.2 Hardware in the loop Simulation .................................... 43
  5.3 Browser Console ..................................................... 44

6 Conclusions and Further Research .................................. 49
  6.0.1 Future work ....................................................... 50

A Controllers with currents ............................................. 51

B Browser Console ......................................................... 53

References ................................................................. 57
Chapter 1

Introduction

This thesis is motivated by an operational mission scenario where multiple marine autonomous robotic vehicles are required to maintain a desired formation and travel along a desired path with a specified speed profile.

This chapter presents the dissertation background and places it in the context of the European Project Co3-AUVs\(^1\) where IST has a fundamental role. The main research goals as well as the main contributions presented and the outline of the dissertation are described.

1.1 Background

The ocean covers approximately 71% of the Earth’s surface and harbour extensive biological diversity. With 200,000 marine species currently known, scientists estimate that the number of species yet to be found could probably be tenfold \(^{12}\). The oceans are an enormous source of food, minerals, and energy waiting to be harvested. They also play a key role in regulating the climate of the planet. For these reasons, there is considerable interest in the exploration and exploitation of the oceans to better understand its dynamics and to promote the rational and sustainable exploitation of marine resources. This calls for the development of advanced systems and methods to sample the ocean at an unprecedented scale, so as to get a synoptic view of the physical, chemical, geological, and biological phenomena that occur in the water column and at the interfaces between the ocean and the atmosphere as well as the seabed \(^5\).

In practice, this mandates the use of increasingly sophisticated autonomous marine

\(^{1}\)Cooperative Cognitive Control for Autonomous Underwater Vehicles
robots capable of roaming the ocean and accessing sites that have hitherto been out of reach of humans or remotely operated vehicles. Even for sites that can be accessed with simple robots, the use of autonomous systems has the potential to substantially decreased the costs and increase the efficiency of scientific and commercial operations at sea. This change of paradigm is steadily being brought about due to the availability of advanced technology that includes embedded computer systems, new communication systems, sensors, and actuators.

Figure 1.1: Coordinated Operation of the DELFIM ASC and the INFANTE AUV for scientific missions in the Azores (Source:[13]).

Central to the development and operation of advanced systems for ocean exploration and exploitation is the availability of reliable methods for robot motion control that can yield robust performance in the face of stringent communication limitations, external disturbances, and reduced actuator control authority. Suffice it to remember that these robots are required to operate in a very hostile environment, with limited access to GPS signals, strong attenuation of electromagnetic waves, and complex underwater propagation scenarios. There is also added difficulty in controlling under-actuated robots (that is, robots with a smaller number of control inputs than the number of degrees of freedom) in the presence of unpredictable currents.

In order to overcome the difficulties that arise during the operation of autonomous marine robots, there has been over the past decade a flurry of activity aimed at the development of new methods for robust single vehicle control. Identical efforts have been witnessed in a number of areas where under-actuated systems are commonly used. Together, they address challenging problems involved in the control of hovercraft, spacecraft, helicopters, missiles, surface vessels, and underwater vehicles [2].
1.1. BACKGROUND

ASIMOV project [1998-2001] is an example where the main goal was to perform a mission at sea (near the Azores islands) at depths of 100 meters and find patterns in community diversity at the vents in the area and drive an AUV close to seabed and detect bubble emissions from discharging vents [13].

Recently, the attention has been focused on the design and operation of groups of autonomous robots that must cooperate to achieve a given task. The key concept is to distribute among a network of vehicles different sensors or capabilities in order to reduce the time involved in carrying out a mission and increase its efficiency and reliability and, in some cases, to allow for the execution of tasks that are out of reach of a single vehicle (Figure 1.1).

As example Figure 1.2 in EU GREX project [2006-2009], where an autonomous surface craft(ASC) and an autonomous underwater vehicle(AUV) carry only some required sensors (dependent on the environment), doing a synchronised maneuver, while maintaining vertical alignment.

It is against this backdrop of ideas that this thesis addresses a number of challenging issues that have been defined in the scope of the EC Co3-AUVs project [2009-2012], which aims at the development, implementation, and testing of cognitive systems for coordination.

---

2 Coordination and Control of cooperating heterogeneous unmanned systems in uncertain environments.
and cooperative control of multiple Autonomous Marine Vehicles (AMVs), in a scenario referred to as Cooperative Control and Navigation of Multiple Marine Robots for Assisted Human Diving Operations. In this scenario, a number of autonomous surface craft are required to maintain a desired formation and guide an underwater target (a human or an underwater vehicle) along a desired path using acoustic communications. The diver should follow the vehicles commands and at the same time explore and examine seabed (check Figure 1.3). Another practical scenario requires that the vehicle fleet follow a diver to track and log his position as the diver explores the underwater world at leisure.

Figure 1.3: Diver localization and guidance (Left Source: Artist - Dinsa Cunha Silva, 2008; Right Source: [1]).

1.2 Research Objectives

The main objective of this thesis is to design and test at sea a number of systems required for multiple vehicle cooperation in the presence of external disturbances. As such, the thesis contribute to bridge the gap between theory and practice. To this effect, the dissertation affords the reader a concise presentation of the four main issues involved in the design and implementation of cooperative control systems.

Namely,

• **Vehicle Modelling** - The objective is to obtain a realistic mathematical model for each of the vehicles involved, by resorting to first physics principles and simple identification methods. The models developed allow for extensive simulations as well
1.3. MAIN CONTRIBUTIONS

as controller parameter tuning in a hardware-in-the-loop simulation environment, prior to deployment at sea.

- **Single Vehicle Motion Control** - This task addresses the problem of designing, for each of the vehicles involved, inner loops for yaw and speed control taking into account the vehicle dynamics, actuator constraints, and sensor measurement noise. The performance of the resulting control laws must necessarily be assessed in simulation, prior to implementation in the real vehicles that are property of IST.

- **Multiple Vehicle Cooperative Motion Control** - In this topic, the emphasis is placed on the development of control structures effectively enabling a number of vehicles to jointly maneuver along desired paths while maintaining a specified geometric pattern (Cooperative Path Following). In the set-up adopted, cooperation among the vehicles is achieved by exchanging data on the "normalized length" of the path traversed by each vehicle and adjusting their speeds accordingly so as to reach formation. Path following of each vehicle is done at a single vehicle level, by generating appropriate yaw references. The resulting control architecture displays an inner-outer control loop structure, whereby speed and yaw commands are to be tracked by the inner loops developed for single vehicle control.

- **Implementation and tests at sea** - This task involves the design and implementation of a software architecture for vehicle control, the sensor and actuator interfaces, a communications layer for inter-vehicle communications, a console for mission programming and execution, software safety features, and the performance evaluation by carrying several tests at sea.

1.3 Main Contributions

This work presents motion control systems for multiple marine vehicles, starting first with the goal of obtaining a realistic model, then the description of single motion controllers and an algorithm for cooperative path-following.

All the proposed algorithms were implemented in marine robotic vehicles *MEDUSA* and tested at sea.
CHAPTER 1. INTRODUCTION

1.3.1 Vehicle Modelling

Vehicle modelling consists in deriving a set of equations that describe the motion of the vehicle (in this case a surface vehicle) as realistic as possible. In fact it should be mentioned that the model also depends on the envisioned application. For control design the model should be simple and capture the main behaviour. In contrary, for test the controllers in simulation, the model should not only represent the vehicle dynamics, kinematics but also include external perturbations (e.g. currents).

In this thesis the model was based on an existing model of another vehicle, called SIRENE [3], and then the parameters were calibrated by resorting to real data from the tests of the vehicles at sea.

1.3.2 Control of a single vehicle

Developing a guided motion control can be specially challenging in the field of marine robotics because the dynamics are often complex and cannot be simply ignored or drastically simplified. In addiction it is common for marine vehicles to be underactuated, that is, to have fewer actuators than degrees-of-freedom. Motion control for this class of vehicles is especially challenging because most of these systems exhibit nonholonomic constraints.

In this thesis, we propose several control systems which were designed taking into account vehicle’s manoeuvring and agility constrains. The motion control algorithms proposed include the heading controller, the speed controller, and the path-following controller. Their performance was extensively tested in a real vehicle called MEDUSA.

1.3.3 Cooperative control of multiple vehicles

In many ocean missions the vehicles are intended to cooperate to achieve some task. The proposed coordination strategy consists in executing a path-following algorithm for each of the vehicles, together with a synchronization algorithm that changes the nominal speeds of the vehicles so as to achieve the desired temporal synchronism and takes into account explicitly the topology of the inter-vehicle communication network.

1.3.4 Implementation

In this thesis we describe the software architecture adopted to implement the controllers developed in the real vehicles. The main idea for the middle-ware software was to make it
easily changed and configured to support new controllers, sensors or hardware changes. To this effect, was decided that each vehicle will run Matlab in real time.

1.4 Thesis Outline

This thesis is organized as follows:

Chapter 2 introduces the MEDUSA class of autonomous marine robots used in the work and describes the dynamic model adopted for simulation and control systems design.

Chapter 3 describes the algorithms used for single motion vehicle control.

Chapter 4 is an exposition of the algorithm adopted for multiple vehicle cooperative motion control.

Chapter 5 describes the hardware and software architectures used for vehicle and mission control at sea.

Chapter 6 summarizes the results obtained and discusses topics that warrant further research and development effort.
Chapter 2

Autonomous Underwater Vehicle Model

With the objective of designing the motion control algorithms and testing them before going to water, in this chapter we derive the equations that rule the motion of an autonomous marine vehicle with special focus on the MEDUSA vehicle. To this end, we start to define the coordinate frames (Section 2.1), the general vehicle motion equations (Section 2.2.2), the characterization of the vehicle MEDUSA (Section 2.3), and finally its mathematical model (Section 2.2).

2.1 Coordinate Frames

To derive the equations of motion for a marine vehicle it is standard practice to define two coordinate frames: Earth-fixed inertial frame \( \{U\} \) composed by the orthonormal axes \( \{x_U, y_U, z_U\} \) and the body-fixed frame \( \{B\} \) composed by the axes \( \{x_B, y_B, z_B\} \), as indicated in Figure 2.1.

The body axes are defined as follows:

- \( x_B \) is the longitudinal axis (directed from the stern to fore);
- \( y_B \) is the transversal axis (directed from port to starboard);
- \( z_B \) is the normal axis (directed from top to bottom).

To simplify the model equations the origin of the body-fixed frame is normally chosen to coincide with the center of mass of the vehicle. The motion control of \( \{B\} \) (that corresponds
to the motion of the vehicle) is described relative to the inertial frame \{U\}.

In general, six independent coordinates are necessary to determine the evolution of the position and orientation (six degrees of freedom), three position coordinates \((x, y, z)\) and using the Euler angles three orientation angles \((\phi, \theta, \psi)\). The six motion components are defined as surge, sway, heave, roll, pitch, and yaw, and adopting the SNAME notation it can be written as in the Table 2.1 or in a vectorial form:

- \(\eta_1 = [x, y, z]^T\) - position of the origin of \{B\} expressed in \{U\}, North\((x)\), East\((y)\), Down\((z)\).
- \(\eta_2 = [\phi, \theta, \psi]^T\) - orientation of \{B\} with respect to \{U\}, Roll\(\phi\), Pitch\(\theta\), Yaw\(\psi\).
- \(\nu_1 = [u, v, w]^T\) - linear velocity of the origin of \{B\} relative to \{U\}, expressed in \{B\}.
- \(\nu_2 = [p, q, r]^T\) - angular velocity of \{B\} relative to \{U\}, expressed in \{B\}.
- \(\tau_1 = [X, Y, Z]^T\) - actuating forces expressed in \{B\}.
- \(\tau_2 = [K, M, N]^T\) - actuating moments expressed in \{B\}.

In compact form yields

\[
\eta = [\eta_1^T, \eta_2^T]^T
\]

\[
\nu = [\nu_1^T, \nu_2^T]^T
\]

\[
\tau_{RB} = [\tau_1^T, \tau_2^T]^T
\]
2.2 Vehicle Model

2.2.1 Kinematic Equations

The kinematics treats only the geometrical aspects of motion, and relates the velocities with position. Using the coordinate frames notation in Section 2.1, the kinematic equations can be expressed as

\[ \dot{\eta} = J(\eta) \nu \]  

(2.4)

with

\[ J(\eta) = \begin{bmatrix} \mathbb{1} & 0_{3 \times 3} \\ 0_{3 \times 3} & T_{\Theta}(\Theta) \end{bmatrix} \]  

(2.5)

where

\[ \mathbb{U}_B R(\Theta) = \begin{bmatrix} c\psi c\theta & c\psi s\theta s\phi - s\psi c\phi & c\psi s\theta c\phi + s\psi s\phi \\ s\psi c\theta & s\psi s\theta s\phi + c\psi c\phi & s\psi s\theta c\phi - c\psi s\phi \\ -s\theta & c\theta s\phi & c\theta c\phi \end{bmatrix}, s\cdot = \sin(\cdot), c\cdot = \cos(\cdot) \]  

(2.6)

is the rotation matrix from \{B\} to \{U\}, [S], defined by means of three successive rotations (zyx-convention):

\[ \mathbb{U}_B R(\Theta) := R_{z,\psi} R_{y,\theta} R_{x,\phi} \]  

(2.7)

and

\[ T_{\Theta}(\Theta) = \begin{bmatrix} 1 & s\phi t\theta & c\phi t\theta \\ 0 & c\phi & -s\phi \\ 0 & s\phi/c\theta & c\phi/c\theta \end{bmatrix}, \theta \neq \pm 90^\circ \]  

(2.8)
is the Euler attitude transformation matrix that relates the body-fixed angular velocities $(p, q, r)$ with the roll($\dot{\phi}$), pitch($\dot{\theta}$) and yaw($\dot{\psi}$) rates.

Notice that $T_{\Theta}(\Theta)$ is not defined for the pitch angle $\theta \neq \pm 90^\circ$, resulting from using Euler angles to describe the vehicle’s motion. To avoid this singularity, another possibility is to use a quaternion representation, [8]. However, due to physical restrictions the vehicle will operate far from this singularity ($\theta \approx 0$ and $\phi \approx 0$) which implies that we can use the Euler representation.

### 2.2.2 Dynamic Equations

Since the hydrodynamic forces and moments have simpler expressions if written in the body frame because they are generated by the relative motion between the body and the fluid, it is convenient to formulate Newton’s law in $\{B\}$ frame. In that case, the rigid-body equation can be expressed as

$$M_{RB}\ddot{\nu} + C_{RB}(\nu)\nu = \tau_{RB}, \quad (2.9)$$

where $M_{RB}$ is the rigid body inertia matrix, $C_{RB}$ represents the Coriolis and centrifugal terms and $\tau_{RB}$ is a generalized vector of external forces and moments and can be decomposed as

$$\tau_{RB} = \tau + \tau_A + \tau_D + \tau_R + \tau_{dist}, \quad (2.10)$$

- $\tau$ - Vector of forces and torques due to thrusters/surfaces, [5], which usually can be viewed as the control input,

- $\tau_A$ - The force and moment vector due to the hydrodynamic added mass,

$$\tau_A = -M_A\dot{\nu} - C_A(\nu)\nu, \quad (2.11)$$

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

$$\tau_D = -D(\nu)\nu, \quad (2.12)$$

where $D(\nu)$ denotes the hydrodynamic damping matrix (positive definite).
2.2. VEHICLE MODEL

- \( \tau_R \) - Restoring forces and torques due to gravity and fluid density,

\[
\tau_R = -g(\eta) \tag{2.13}
\]

- \( \tau_{\text{dist}} \) - External disturbances: waves, wind, etc.

Replacing (2.10) on (2.9), taking into account (2.11), (2.12), the dynamic equations can be written as

\[
\begin{align*}
M_{\text{RB}} \ddot{\nu} + C_{\text{RB}}(\nu) \nu + M_A \ddot{\nu} + C_A(\nu) \nu + D(\nu) \nu + g(\eta) &= \tau + \tau_{\text{dist}}, \\
M \dot{\nu} + C(\nu) \nu + D(\nu) \nu + g(\eta) &= \tau + \tau_{\text{dist}},
\end{align*}
\tag{2.14}
\]

where \( M = M_{\text{RB}} + M_A, C(\nu) = C_{\text{RB}}(\nu) + C_A(\nu) \).

2.2.3 Simplified Equations of Motion

The vehicle in question will operate only in the horizontal plane (See Section 2.3). In that case, we only have three degrees of freedom \([x,y,\psi]\) and the kinematics equations take the simple form

\[
\begin{align*}
\dot{x} &= u \cos \psi - v \sin \psi, \\
\dot{y} &= u \sin \psi + v \cos \psi, \\
\dot{\psi} &= r.
\end{align*}
\tag{2.15}
\]

Denoting \( F_{sb} \) and \( F_{ps} \) as the starboard and port-side thruster forces respectively, and \( l \) the length of the arm with respect to the center of mass, we obtain

\[
\begin{align*}
\tau_u &= F_{sb} + F_{ps}, \\
\tau_r &= l(F_{ps} - F_{sb}),
\end{align*}
\]

where \( \tau_u \) is the external force in surge (common mode), and \( \tau_r \) the external torque about the \( z \)-axis (differential mode between thrusters).

Neglecting roll, pitch and heave motion, the equations for \((u,v,r)\) without disturbances are

\[
\begin{align*}
m_u \ddot{u} - m_v vr + d_u u &= \tau_u, \\
m_v \ddot{v} + m_u ur + d_v v &= 0, \\
m_r \ddot{r} - m uvw + d_r r &= \tau_r,
\end{align*}
\tag{2.16}
\]
where
\[ m_u = m - X_u \quad d_u = -X_u - X_{|u|u}|u| \]
\[ m_v = m - Y_v \quad d_v = -Y_v - Y_{|v|v}|v| \]
\[ m_r = I_z - N_r \quad d_r = -N_r - N_{|r|r}|r| \]
\[ m_{uv} = m_u - m_v \]

In (2.17), the symbols \( m_u, m_v, m_r \) and \( m_{uv} \) are mass and hydrodynamic added mass, and the symbols \( d_u, d_v, d_r \) represent hydrodynamic damping effects.

All the previous equations were expressed without considering the influence of ocean currents. When we introduce a constant irrotational ocean current, \( v_c \), forming an angle \( \psi_c \) with respect to the fixed frame, the kinematic equations (2.15) hold with \( u = u_r + u_c \) and \( v = v_r + v_c \), where \( u_r \) and \( v_r \) are the components of the AUV velocity with respect to the current and \( u_c \) and \( v_c \) are the components of the ocean current velocity in the body frame.

In this case, the previous dynamic equations (2.16) are modified to

\[ m_u \ddot{u} - m_v (v_r + v_c)r + d_u u_r = \tau_u, \]
\[ m_v \ddot{v} + m_u (u_r + u_c)r + d_v v_r = 0, \]
\[ m_r \ddot{r} - m_{uv} (u_r + u_c)(v_r + v_c) + d_r r = \tau_r, \]

\[ m_u = m - X_u \quad d_u = -X_u - X_{|u|u}|u_r| \]
\[ m_v = m - Y_v \quad d_v = -Y_v - Y_{|v|v}|v_r| \]
\[ m_r = I_z - N_r \quad d_r = -N_r - N_{|r|r}|r| \]
\[ m_{uv} = m_u - m_v \]

2.3 Vehicle Characterization - The Medusa

The vehicles, named MEDUSA, are autonomous semi-submerged robotic vehicles developed at the Laboratory of Robotics and Systems in Engineering and Science (LARSyS)/ISR of the Instituto Superior Técnico of Lisbon, Portugal.

Each MEDUSA-class vehicle weights approximately 30Kg and consists of two acrylic housing tubes of size 0.15m by 1.035m (diameter x length) with aluminum end caps, attached to a central aluminum frame. The distance between the two bodies is 0.15m. The lower underwater tube contains two packs of 7-cell lithium polymer batteries together with
2.3. VEHICLE CHARACTERIZATION - THE MEDUSA

Figure 2.2: MEDUSA dimensions.

the thruster electronics, an underwater camera, an acoustic modem (Tritech UK) and other
minor sensors. The upper body, partially out of the water (see Figure 2.2 for the vehicle
dimensions with more detail), contains the main computer unit (Epic computer NANO PV
D5251, with an Intel Atom D525 dual-core processor, low power, 1.8GHz with 2GB RAM)
together with navigation sensors (an attitude sensor - VECTORNAV VN-100) and a Global
Positioning System (GPS - Ashtech MB100).

Attached to the main frame there are two stern thrusters (SEABOTIX HPDC1507
Brushless thrusters), which control directly the surge and yaw motions. At the nominal
speed of 1.0 m/s, the vehicles have an autonomy of 12 hours. The maximum speed of
the vehicles is 1.5 m/s. Inner vehicle and vehicle/underwater target communications are
enabled via Wi-Fi and an acoustic modem network, respectively.

Roll and pitch does not need to be actuated because of the buoyancy, which guarantee
that the vehicle has large enough metacentric height. Since this vehicle is intended to be a
surface craft the heave motion is not being controlled as well. More information about the
mechanical details and electronic hardware required for the mission can be found in [1].

The final vehicle can be seen in the following figures (2.3), as well as the three (red,
black, and yellow) vehicles (Figure 2.4) in Sesimbra.
Figure 2.3: The MEDUSA vehicle in different views.
2.3.1 SEABOTIX Thruster Model

The SEABOTIX thrusters are powerful brushless DC thrusters with individual oil compensator (compensate the outside depth pressure), rated for 400 meters depth and they also have an electronic board that contains a RPM controller. With the objective of getting a realistic model of the thrusters, several experimental tests were conducted in the ISR/IST test tank (IST Taguspark campus). From the test results we concluded that a

\[ K_0 = 7.2115, \quad Delay = 0.346s, \]

a saturation and a rounding block, since the thrusters are commanded by integer numbers between -100 and 100. The conversion between percentage to RPM (gain), and the
CHAPTER 2. AUTONOMOUS UNDERWATER VEHICLE MODEL

Conversion from RPM to force was taken from [14].

Figure 2.6: Thruster characterization real data versus simulated data with different zoom levels - Taguspark test tank.

From Figure 2.6 it is easy to see that the obtained model is very close to the reality.

2.3.2 Vehicle Model parameters

At this point we have a realistic thruster model but we still need to find the MEDUSA physical parameters for the motion equations (Section 2.2.3). To this end, the first step was to apply a normalization using the Prime-System, [7], of the non-dimensional parameters obtained for the SIRENE vehicle. The normalization of these values for MEDUSA, were achieved by using a normalization factor \( L = \left( \frac{M_M}{M_S} \right)^{\frac{1}{3}} \), where \( M_M \) is MEDUSA’s mass and \( M_S \) SIRENE’s mass.

<table>
<thead>
<tr>
<th>Unit</th>
<th>SIRENE</th>
<th>MEDUSA</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mass</td>
<td>4000kg</td>
<td>30kg</td>
</tr>
<tr>
<td>( I_z )</td>
<td>2660kg( m^2 )</td>
<td>4.14kg( m^2 )</td>
</tr>
</tbody>
</table>

Table 2.2: Physical Properties.

\(^2\)Autonomous Underwater Shuttle designed for the transportation of Benthic Laboratories, more information in [3].
2.3. VEHICLE CHARACTERIZATION - THE MEDUSA

The second step was to tune the values obtained using real data (See Figure 2.7 and Figure 2.8). At the end of this process we have arrived to the following values:

<table>
<thead>
<tr>
<th>X</th>
<th>Y</th>
<th>N</th>
</tr>
</thead>
<tbody>
<tr>
<td>$X_u$</td>
<td>$25\text{kg}$</td>
<td>$-2.325\text{kg}$</td>
</tr>
<tr>
<td>$X_v$</td>
<td>$-0.2\text{kg/s}$</td>
<td>$-55.117\text{kg/s}$</td>
</tr>
<tr>
<td>$X_{[u]}_u$</td>
<td>$-19.5\text{kg/m}$</td>
<td>$-147.900\text{kg/m}$</td>
</tr>
</tbody>
</table>

Table 2.3: Normalized model parameters.

![Model performance (Speed)](image)

Figure 2.7: Surge velocity model analysis: simulated and experimental results.

Figures 2.7 and 2.8 show experimental results. Notice that the real commands used for the real thrusters were used as inputs for the *Simulink* model. At this stage, in the model the external perturbations were not taken into account such as currents, waves, sensor noise. These perturbations can be seen in Figure 2.7, were the vehicle followed a straight line along the current (0-200 seconds) and another one towards the current (200-450 seconds) with the same common mode. Overall, from the figures it can be seen that the tuned model give similar results as obtained using the real vehicle.
Figure 2.8: Yaw velocity model analysis: simulated and experimental results.
Chapter 3

Single Vehicle Motion Control

This chapter presents the motion control algorithms for single marine vehicles. The proposed algorithms include the heading controller, the speed controller, the way-point and hold positioning controller and path-following controller. For all of them, simulation results and experimental tests with the MEDUSA vehicles are described.

3.1 Heading Controller

The objective of the heading controller is to steer automatically the vehicle to a given desired direction (yaw). To this effect, the control algorithm accepts as inputs the reference signal $\psi_d$, the yaw $\psi$ and yaw rate $\dot{\psi}$ that are accessible by the vehicle’s AHRS. Let

$$\tilde{\psi} = \psi - \psi_d \quad (3.1)$$

be the heading error. A simple controller is the proportional-derivative feedback law:

$$u_d = -k_p\tilde{\psi} - k_d\dot{\psi} \quad (3.2)$$

where $u_d$ denotes the differential mode between the thrusters (see Section 2.3.1).

In (3.1), since $\psi \in [0, 2\pi]$ the error $\tilde{\psi}$ has to be redefined according to

$$\tilde{\psi} > \pi \quad (\text{rightside})$$
$$\tilde{\psi} = \tilde{\psi} - 2\pi$$

$$\tilde{\psi} < -\pi \quad (\text{leftside})$$
$$\tilde{\psi} = \tilde{\psi} + 2\pi \quad (3.3)$$
Figure 3.1 illustrates the performance of the heading controller in simulation as well as with real data using the red MEDUSA vehicle. This experimental test was done without any common mode, that is, the vehicle was turning around its center of rotation.

Figure 3.1: Heading Controller simulation and experimental tests.

### 3.2 Speed Controller

The speed controller is responsible for keeping the vehicle at a certain desired speed and be robust as much as possible to external perturbations like ocean waves and currents.

Since it is not possible to access directly the vehicle’s forward velocity because there is no sensor to measure it, the strategy is to include an estimator that receives the position given by the GPS. For that purpose and also to obtain a position without outliers and with less noise it was designed a filter where the input is one axis position and the outputs are the estimated position and velocity, for that axis.

It is pretended to find an observer for the following system:

\[
    \begin{bmatrix}
        \dot{x} \\
        \dot{v}
    \end{bmatrix} = \begin{bmatrix}
        0 & 1 \\
        0 & 0
    \end{bmatrix}
    \begin{bmatrix}
        x \\
        v
    \end{bmatrix}
    + A
\]

\[
y = \begin{bmatrix}
    1 & 0
\end{bmatrix}
    \begin{bmatrix}
        x \\
        v
    \end{bmatrix}
\]  

(3.4)
3.2. SPEED CONTROLLER

The observability matrix is

\[
O = \begin{bmatrix} C \\ CA \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \tag{3.5}
\]

Since \( O \) is full rank we can conclude that the system is observable and therefore it is possible to design an observer so that the estimation error \( \hat{x} = x(t) - \hat{x}(t) \) will converge to zero.

Let the observer be given by

\[
\dot{\hat{x}}(t) = A\hat{x}(t) + L \left( y(t) - C\hat{x}(t) \right), \quad \text{where} \quad L = [L_1 L_2]^T \tag{3.6}
\]

Then,

\[
\dot{\hat{x}} = (A - LC) \hat{x} \tag{3.7}
\]

The gains \( L_1 \) and \( L_2 \) are found such that

\[
det(sI - A + LC) = \begin{vmatrix} s + L_1 & -1 \\ L_2 & s \end{vmatrix} = s^2 + L_1 s + L_2, \quad K_1 = 2 \xi \omega_n, \quad K_2 = \omega_n^2 \tag{3.8}
\]

Selecting \( \xi = 0.707 \), it is necessary to find the natural frequency \( \omega_n \) for the observer. This frequency cannot be too low because the observer would be slow to converge (the estimation error will take too much time to reach zero) and cannot be very high because the estimator would follow the measurement error. In the case of the \( MEDUSA \) vehicles \( \omega_n \) was set to \( \omega_n = 0.5 \text{rad/s} \).

The implemented observer diagram is shown in Figure 3.2, with a technique to initialize the integrators for a period of time, making sure that the initial condition of the observer \( \hat{x}_0 \) is close to the real position of the vehicle.

Figure 3.2 shows the evolution in time of relevant signals of the filter for an experiment where the vehicle was moving in straight line with a speed around 0.2 m/s.
Figure 3.2: Implemented filter for position and velocity estimation.

Figure 3.3: Observer output performance.

In Figure 3.3a the trajectory of the vehicle’s position provided by the GPS in one axis, the real and its estimated position produced by the filter are displaced. As it can be seen the difference between them is minimum because in this test we were using RTK\(^1\) GPS. In Figure 3.3b it is plotted the velocity using a pure differentiation of the position (in RTK mode is less noisy than usual), the velocity given by the GPS and the velocity filter output which has a better attenuation of the noise compared with the other two.

\(^1\)Real Time Kinematic (RTK) satellite navigation is a technique used in land survey and in hydrographic survey based on the use of carrier phase measurements of the GPS signal where a single reference station provides the real-time corrections, providing up to centimetre-level accuracy.
3.2. SPEED CONTROLLER

Having the estimates \( \hat{V}_x \) and \( \hat{V}_y \) we can now easily derive the vehicle’s velocity expressed in body frame by rotating them, that is, \( u \) and \( v \) (see Figure 3.4):

\[
\begin{bmatrix}
\hat{u} \\
\hat{v}
\end{bmatrix} = \begin{bmatrix}
sin(\psi) & \cos(\psi) \\
\cos(\psi) & -\sin(\psi)
\end{bmatrix}\begin{bmatrix}
\hat{V}_x \\
\hat{V}_y
\end{bmatrix}
\] (3.9)

Note that the vehicle movement due to currents and other perturbations will be measured in the inertial frame and will take part in \( \hat{u} \) and \( \hat{v} \). At this point we have the ingredients to design the speed controller.

Let

\[
\tilde{u} = \hat{u} - u_d
\] (3.10)

be the error between the measurement speed \( u \) and the desired speed \( u_d \).

And

\[
\xi = \int_0^t \tilde{u}(\tau) \, d\tau
\] (3.11)

is the integration error that will be used to take into account the fact that the vehicle’s thrusters are commanded in RPMs and without the integration part the controller cannot learn the relation between RPMs and nominal speed (this relation is not constant along time since there are currents and other external perturbations, neither is linear).

Figure 3.4: Inertial and Body velocity vectors.

With (3.10) and (3.11) we define a PI (proportional-integral) controller for the common mode:

\[
u_c = -k_{pu}\tilde{u} - k_{iu}\xi, \quad \text{where } k_{pu}, k_{iu} > 0
\] (3.12)

Figure 3.5 shows simulations results using the PI controller. It can be seen that the vehicle needs about 50 seconds to reach the reference speed, which is too slow. This is due
to the fact the thruster model contains a time-delay. To compensate the time-delay and obtain a better performance we decided to implement a Smith Predictor, [9], [10].

3.2.1 Smith Predictor

The main goal of the Smith Predictor algorithm is to compensate the time-delay in the process by including an inner-loop that tries to predict the future vehicle’s state based on the inputs. To implement the Smith predictor we used a simple version of the model described in Section 2.2 for surge velocity, described by the system (in discrete time)

\[ H(z) = \frac{0.0006641z}{z^2 - 0.6368z - 0.2936} \]  

(3.13)

Then, it is delayed by a few samples and given as compensation of the error \( \hat{u} \), reducing the control oscillations [9]. Figure 3.6 shows the implementation details of the speed controller using a Smith predictor. The simulation results for the PI controller with and without compensation of the delay is presented in Figure 3.7.

Comparing the two it can be understood that with the Smith Predictor it is possible to increase the gains, without having the problem of oscillations or too much overshoot.
3.3. WAY-POINT AND HOLD POSITION CONTROLLER

Controlling a marine robotic vehicle in a real scenario is not an easy task because the human operators need time to plan the missions and in the meanwhile the vehicle can drift due to waves and ocean currents. Motivated by this fact, a point stabilization controller was developed to steer the vehicle to a given point \((x_d, y_d)\) and keep it in a neighbourhood around the desired point.

Inspired by the point stabilization algorithm described in [4] we implemented the following strategy:

1) while the vehicle is in a neighbourhood of radius \(\epsilon_d\) of the desired point set to zero the
speed reference \((u_d = 0)\) and maintain a given orientation using the heading controller. This implies that the vehicle is not wasting unnecessary batteries with the thrusters.

2) If the vehicle is outside the neighborhood use the heading and speed controllers to bring it back.

The switching between these two modes of operation is done applying a convenient hysteresis to avoid chattering.

In mode 2) the references for the speed and heading controller are given by

\[
\begin{align*}
    u_d &= k_u \sin^{-1}\left(\frac{d}{|d|+k_s}\right) \frac{2}{\pi}, \\
    \psi_d &= \text{atan2}(y_d - y, x_d - x), \\
    d &= \sqrt{(x - x_d)^2 + (y - y_d)^2} - d, \\
\end{align*}
\]

(3.14)

\(k_u > 0\) defines the upper limit on the speed reference \(u_d\) and \(k_s > 0\) is the parameter for tuning the vehicle speed with respect to distance error as it is shown in Figure 3.8.

In (3.14), the function \(P = \text{atan2}(Y,X)\) returns a variable \(P\) containing the four-quadrant inverse tangent (arctangent) of the real parts of \(Y\) and \(X\). The elements of \(P\) are in the interval \([-\pi/2, \pi/2]\).

![Impact of \(K_s\) on speed reference](image)

Figure 3.8: Importance of \(k_s\) parameter \((k_u = 0.5)\).

The way-point and hold position controller was implemented in Simulink as it is shown in Figure 3.9.
Figure 3.9: Waypoint Diagram.

Figure 3.10 shows simulation and experimental results for one scenario where the vehicle starts at a distance of 65 meters and the radius $\epsilon_d$ was set to $\epsilon_d = 1 m$. From the Figure 3.10 it can be concluded that the vehicle converges to the desired position even with the presence of ocean currents or other perturbations.

Figure 3.11 shows a simulated test with an ocean current of 0.3 m/s.

(a) Real Test
(b) Simulation Test

Figure 3.10: Performance of the way-point Controller.
3.4 Path-Following Controller

In ocean missions of interest, marine vehicles are required to converge and follow spatial paths accurately. Once in the path, the vehicle should follow it with a desired speed. In this case, no temporal constrains were imposed, contrasting with trajectory tracking controllers where the reference for the vehicle motion is time parametrized. This may lead to situations where the required speed with respect to the water may be too small (leading to an unstable control) or too high (exceeding the capability of the vehicle) [11].

In this work, the path following controller implemented uses the Heading and Speed controllers described in Section 3.1 and Section 3.2 respectively. The Path-following algorithm works in an outer-loop control design and it is responsible to issue the convenient heading and speed commands based on the actual position and desired mission. For simplicity, we consider that the mission is only composed by straight lines and arcs. Let $e$, be the cross track error (the distance to the nearest point in the path) and $\beta$ the angle between the tangent to the point in the path and the vertical axis (see Figure 3.12).

The desired heading and speed commands are given by

$$
\psi_d = \beta + \sin^{-1}(\text{sat}(\psi_c))
$$

$$
u_d = v_L$$

(3.15)
3.4. PATH-FOLLOWING CONTROLLER

where,

\[
\text{sat}(\psi_c) = \begin{cases} 
\psi_c & \text{if } |\psi_c| < 1 \\
1 & \text{if } |\psi_c| \geq 1 \\
-1 & \text{if } |\psi_c| \leq -1 
\end{cases}
\]  
(3.16)

and

\[
\psi_c = -\int_0^\tau \left( \frac{k_1}{v_L} e(\tau) + \frac{k_2}{v_L} \dot{e}(\tau) \right) d\tau 
\]  
(3.17)

In \text{3.17} the gains \( k_1 > 0 \) and \( k_2 > 0 \) are chosen as function of the desired natural frequency and damping factor for a second order system and \( v_L \) is the mission nominal velocity.

In [11], it is proved that the cross track error converges to zero assuming that the heading controller is fast enough. Figures 3.13 and 3.14 illustrate in simulation the path-following controller for a straight line and a lawn mower path (a maneuver commonly used to exhaustively survey area) composed by straight lines and arcs. In Figure 3.15 the results of a similar lawn mower maneuver are displayed for a real scenario where one MEDUSA vehicle was collecting bathymetric data in \textit{Parque das Nações} dock, Lisbon, Portugal. From the figures it can be seen that the algorithm works well both in simulation as in reality with some limitations in the arc radius due to vehicle physical limitations (turning with higher speed references).
CHAPTER 3. SINGLE VEHICLE MOTION CONTROL

Figure 3.13: Path-following for a straight line along x axis.

Figure 3.14: Path-following for a lawn mower maneuver.
Figure 3.15: Path-following for a lawn mower maneuver in "Parque das Nações" dock.
Chapter 4

Multiple Vehicle Motion Control

This chapter starts to describe a general architecture for multiple vehicle cooperative motion control. Then, we present the cooperative path-following maneuver and illustrate it through computer simulations and experimental data obtained in a series of tests carried out in Sesimbra October 2011.

4.1 General architecture for multiple vehicle cooperation

Figure 4.1 illustrates the proposed architecture for multiple vehicle cooperation that is composed by the following blocks:

i. **Path-following controller** (described in Section 3.4) a dynamical system whose inputs are a path $P_{di}$, a desired speed profile that is common to all vehicles, and the vehicle’s output $y_i$. Its outputs are the vehicles input $v_L$, computed so as to make it follow the path at the assigned speed, an heading reference $\psi_d$, and a generalized path-variable $\gamma_i$.

ii. **Coordination Controller** (described in this Section) is the dynamical system that enforces coordination with other vehicle, receiving as inputs the generalized path-variable $\gamma_i$, and the generalized coordination states $\gamma_j$ of the $n$ agents it communicates with. It passes on to the vehicle’s inner-loops the associated speed profile $u_i$, coupled with the correction speed signal $v_{corr_i}$, which is used to synchronize vehicle $i$ with its neighbours.

iii. **The path generator** is, in this case, responsible for generating the desired path $P_{di}(\gamma_i)$:
The parameter $\gamma$ of each vehicle is seen as a coordination state, such that coordination exists between vehicles $i$ and $j$ if and only if $\gamma_i = \gamma_j$. The controller designed for coordination yields a correction term, $v_{corr} = [v_{corr_1}, \ldots, v_{corr_n}]^T$, that is added to the desired speed of each vehicle, with the goal of driving the coordination error to zero. Thus, the desired velocity that enters the inner-loop is

$$v_d = v_L + v_{corr} \quad (4.1)$$
A decentralized control strategy is designed where the coordination speed \( v_{corr} \) of vehicle \( i \) is determined based on the available measurements of the \( \gamma_i \) parameters, that is, based on the coordination states of the vehicles that can communicate with vehicle \( i \). Let \( \gamma = [\gamma_1, \cdots, \gamma_n]^T \) be the vector containing the coordination states of each vehicle including the target to follow and \( N_i \) denote the set of vehicles that vehicle \( i \) can communicate with.

Defining the coordination error as
\[
\tilde{\gamma}_i = \gamma_i - \frac{1}{|N_i|} \sum_{j \in N_i} \gamma_j
\] (4.2)
that is basically the error between the vehicle gamma with the mean of the gammas from the other vehicles, a simple controller can be designed to lead \( \tilde{\gamma}_i \) to zero as
\[
v_{corr_i} = -k_{sync} \sin^{-1}\left(\frac{\tilde{\gamma}_i}{|\tilde{\gamma}_i| + k_s}\right) \frac{2}{\pi}
\] (4.3)
In (4.3) \( k_{sync} \) defines the upper limit in the speed correction (cannot be higher than the nominal path speed, because the vehicle could run backwards) and \( k_s \) is the parameter to tune the reactivity to variations in gamma error.

The above mentioned algorithm can be defined in a vector form, by introducing the adjacency matrix where elements are defined as
- \( a_{ij} = 0 \) if there is no communication between vehicle \( i \) and \( j \);
- \( a_{ij} = 1 \) if vehicle \( i \) communicates with vehicle \( j \).

Introducing also the degree matrix \( D \) that is a square diagonal matrix with the number of vehicles that each vehicle communicates with. The error between gammas \( \tilde{\gamma} \) can be written as
\[
\begin{bmatrix}
\tilde{\gamma}_1 \\
\vdots \\
\tilde{\gamma}_n
\end{bmatrix} =
\begin{bmatrix}
\gamma_1 \\
\vdots \\
\gamma_n
\end{bmatrix} - D^+ \times A \times
\begin{bmatrix}
\gamma_1 \\
\vdots \\
\gamma_n
\end{bmatrix}
\] (4.4)
where \( D^+ \) denotes the ”pseudoinverse” of \( D \). The calculus of speed corrections will stay the same using this \( \tilde{\gamma}_i \) and \( i \) the vehicle that is calculating it.

Using this framework it is possible to have different network topologies.
A = \begin{bmatrix}
0 & 0 & 0 \\
1 & 0 & 0 \\
1 & 0 & 0 \\
\end{bmatrix} \quad D = \begin{bmatrix}
0 & 0 & 0 \\
0 & 1 & 0 \\
0 & 0 & 1 \\
\end{bmatrix}

Table 4.1: Communication topology for three vehicles where only 2 and 3 receives data from 1.

This will be similar to a leader/follower topology and it is easily seen that the performance will decrease, imagine that 1 runs at a speed that 2 and 3 can not achieve, 1 will not reduce its own velocity in order to preserve the formation (worst case scenario).

This improvement will also work if vehicles that communicate all with each other;

\[
A = \begin{bmatrix}
0 & 1 & 1 \\
1 & 0 & 1 \\
1 & 1 & 0 \\
\end{bmatrix} \quad D = \begin{bmatrix}
2 & 0 & 0 \\
0 & 2 & 0 \\
0 & 0 & 2 \\
\end{bmatrix}
\]

Table 4.2: Communication topology for three vehicles where everyone knows each other position.
4.2. COOPERATIVE PATH-FOLLOWING

4.2.1 Results

The algorithm for cooperative path following was implemented and tested with great success with three vehicles. Before testing this algorithm on the field, simulations were performed with the model specified in Section 2.2. Simulation results for different missions and formations are shown in Figures 4.2a, 4.3a, 4.4a. Figures 4.2b, 4.3b, 4.4b show the same missions in a real scenario at Sesimbra sea.

![Simulation](image1)

![Real](image2)

Figure 4.2: Line Mission with a formation in line.

![Simulation](image3)

![Real](image4)

Figure 4.3: Lawn-mower Mission with a formation in triangle.

The figures demonstrate the good behavior of the algorithm for different formation and even in the presence of currents, where it can be noticed for the straight lines in Figure 4.4.
that the path-following controller learns how to counteract the current effect by pointing the vehicle towards its direction. Notice also that the mission path in Figures 4.4 and 4.3 where specified in such a way that they present discontinuities in the first derivative (velocity) in the transitions line/arc. Nevertheless, the convergence of the algorithm does not deteriorate significantly.
Chapter 5

Implementation

This chapter describes the software architecture adopted to implement the controllers developed in the real vehicles. The main idea for the middleware software was to make it easily changed and configured to support new controllers, sensors or hardware changes. To this effect, was decided that each vehicle will run Matlab in real time.

5.1 Software Architecture

For control, navigation, and mission control systems implementation, each vehicle runs a standard Linux operative system on a 2GHz Intel dual-core, 2 GB RAM platform. The architecture adopted allows for seamless implementation and hardware-in-the-loop testing of solutions developed in the Matlab environment. For this effect independent processes were developed for each sensor and actuator, to allow a communication interface with the hardware. A general block diagram of the inter-process communication architecture is shown in Figure 5.1. The middleware software infrastructure responsible for coordinating and communicating with the processes in the open source MOOS\(^1\). The core of MOOS is a robust network based communication architecture that simplifies substantially the effort of building an application and where all communications are made through a Database called MOOSDB. With this set-up, each process must only analyse the sensor data and post the important variables on this MOOSDB. In MOOS software package there are a huge variety of additional tools/applications to cover some basic tasks like visually debug a set of communicating processes, logging all the necessary data and missioning control.

\(^{1}\)Mission Oriented Operating Suite developed by Oceanai MIT - http://www.moos-ivp.org
Write the last values every 500ms

Write the last values in Simulink every 200ms

MOOSDB

THRUSTERS
GPS
IMU
MODEM
BATTERIES

MATLAB Server

Safety Feature

Display

HTTP Server

External PC

Browser

5 Hz

5 Hz

10 Hz

0.2 Hz

1 Hz

5 Hz

2 Hz

5 Hz

By Query

Matlab Commands (RPMs)
Safety Feature
Sensor Data
Hardware Communication

Figure 5.1: Communications between all the processes involved.

In Figure 5.1 the blocks marked as ellipses are the main processes implemented on the vehicle. The top part of the diagram shows the actuator and sensor processes with their sampling frequencies. Bellow MOOSDB are the controller and the visualization parts. The role of some blocks is briefly described below.

- The **THRUSTERS** module handles the bidirectional communication between the thrusters and the Database. It constantly polls the status of both thrusters and posts the result on the database. Likewise it sends to the thruster the speed commands as soon as they become available in the database (posted by the **MATLAB Server**).

- The **IMU**, **GPS**, **MODEM** and **BATTERIES** modules, parse the data coming from the sensors and post it to the database.

- The **Communications** block is responsible for acquiring selected variables and broadcasting them via UDP to all the other vehicles. It also does the inverse role of
5.2. HARDWARE IN THE LOOP SIMULATION

receiving the variables from the remaining vehicles and posting them in the Database to be accessed by the other processes (e.g. the MATLAB Server).

- The MATLAB Server block plays a central role in the architecture because it broadcasts important data and sets the clock of Matlab/Simulink (turning it into a real clock instead of the normal virtual clock).

- The Safety Feature process communicates periodically every 2 seconds through a Wi-Fi link with a computer outside the vehicle and expects a reply from that computer. When that reply fails to arrive the vehicle becomes aware that the link is broken and stops the thrusters after a timeout. This safety feature is intended to stop the vehicle whenever the communication is lost thus it is impossible to send a command to stop the vehicle. This way it can be assured that the vehicle will not go inadvertently and without control against nearby obstacles, neither run ashore.

- The Display and Http Server are responsible for exporting all the variables contained in the Database, periodically or by querying. As an example of the capabilities offered by this server, section 5.3 shows the browser console implemented.

- The MATLAB/SIMULINK block is a software package developed by MathWorks for modelling, simulating and analysing dynamical systems. MATLAB/SIMULINK can be used to implement all the algorithms for cooperative motion control described before.

This structure is very helpful in development stages since it is possible to replace/add one block (e.g. Sensor Block) by a simulated one, or increasingly remove blocks from Matlab and add new blocks without having to change the others. For instance, if the Heading controller is fully tested and working, it may be easily attached as new block as a standalone process.

5.2 Hardware in the loop Simulation

Hardware in the loop simulation is a process that combines in the loop real time simulation, with real components (sensors and actuators) and simulated vehicle dynamics. With this technique it is possible to test every block on a friendly and controlled environment
(e.g. Laboratory) in a fast and safe way. To achieve the above mentioned goals it is desirable to keep as many original processes running in real conditions as possible, without performing any change. On that ground, GPS and IMU are simulated by a laptop connected through serial port to the vehicle computer exactly at the same ports as GPS and IMU are. The vehicle runs the same applications except for the thrusters module which is slightly modified: instead of sending the speed commands to the hardware only, it also sends the RPMs through TCP-IP connection to an external laptop (shown in Figure 5.2). That laptop runs the vehicle model and simulates the real hardware messages, while in the vehicle it is running in the same way as in a real mission.

![Hardware in the loop Configuration.](image)

Figure 5.2: Hardware in the loop Configuration.

### 5.3 Browser Console

One important aspect in a real mission at sea is to know the location of the vehicles at all time and obtain their status from time to time. Typically examples include the status of the batteries and, if there is a failure in the thrusters (like a plastic bag rolled around the propellers or a temperature problem). The main requirements for the proposed browser console were the following:

- Possibility of checking every little detail in the vehicle;

- plot periodically all the vehicles position and orientation;
5.3. **BROWSER CONSOLE**

- stop the vehicles immediately;
- give way point references;
- change scenario;
- receive warning messages when something goes wrong (e.g. batteries running out; leak detection);
- as light as possible, not to over charge Wi-Fi link;
- easy to access with any computer;
- user friendly.

The solution chosen was to have a console implemented in HTML so that it is possible to run it in any pc with a browser installed.

Figure 5.3 shows the console running in a browser (Google Chrome) on a laptop, when the vehicles were doing a cooperative path following mission with a triangle formation in Sesimbra. The red circles with a number inside (that do not exist in the real console) mean the following:

1 - drop down menu with all the available scenarios and different levels of zoom (the available options for now are: Sesimbra; Parque das Nacoes - doca dos Olivais; Oeiras Swimming Pool)

2 - drop down menu where it is possible to ask for the raw messages (Thrusters, GPS, IMU, etc), and ask for a brief description of the vehicle status.

3 - variables drop down menu gives the possibility of select/unselect the variables that we would like to see in 15 (available options: GPS, IMU, Thrusters, Battery, Altimeter), some of the variables are always selected because of the display requirements. Selection of a huge number of variables for a long time should be avoided due to the increase in data transmission causing the over charge of the Wi-Fi link.

4 - battery indicator with color that varies between green(100%) and red(0%)

5 - clear the path made by the vehicles (24)

6 - clear only the way point circles
7 - draw a circle with the cursor pointer, for debugging purposes

8 - zooms an area in the map

9 - original map without any zoom

10 - sends the vehicle to the point clicked by the user and draws a yellow circle in that point

11, 12 - X and Y coordinates of a point. The user can press the letter "j" or "k" on the keyboard and the vehicle goes to that desired position. With this is possible to configure these keyboard keys to send the vehicles to the initial positions of a mission (pressing the same key for each vehicle).

13 - mouse X and Y coordinates in meters

14 - vehicle IP for polling the information (in this case for vehicle 3 because it is selected (17) )

15 - all the user requested variables, with its respective values

16 - X and Y coordinates for holding position like 11 and 12, button stop to stop holding that reference position

17 - vehicle selection (1 - Medusa (red); 2 - Euryale (black); 3 - Stheno (yellow))

18 - GPS mode, that changes its color depending on the mode (Autonomous GPS - Red; RTK Float - Yellow; RTK fixed - Green)

19, 20 - Lower and Upper leak display (Red - Leak in the tube; Green - No leak)

21 - stops the thrusters immediately

22 - map scenario

23 - vehicles icon (Red - Vehicle 1; Yellow - Vehicle 2; Blue - Vehicle 3) in the real position with its own orientation

24 - vehicle described path with its corresponding color

With this tool it is possible to have a person without having in deep knowledge about marine vehicles and control systems the possibility to operate 3 vehicles easily. Since it is written in HTML it is possible to add new features and correct errors in a mission trial.
Figure 5.3: Browser Console for Sesimbra Trial doing CPF.
Chapter 6

Conclusions and Further Research

This thesis addressed the problem of developing the software architecture and system motion control for single and multiple robotic marine vehicles. To this end, a dynamic model that governs the motion of the vehicle Medusa was proposed. At the first stage the values of the proposed parametric model were obtained by resorting to the model of Sirene vehicle and using a convenient normalization. In a second stage, using real data acquired through sea trials, some of the parameters were re-tuned obtaining in this way a very realistic model.

Next we presented a series of controllers for guidance/navigation of a single vehicle to perform simple tasks like follow a specified orientation, with a desired speed or go to a point and stay there even in the presence of currents. These controllers were fully tested in simulation and in open sea with good results.

For the path-following and cooperation of multiple vehicles it was designed and implemented an inner-outer loop structure to be possible to separately test each block individually and obtain a simple way to tune the gains. In the thesis we show the experimental results for three vehicles performing missions in Sesimbra’s sea with different formation patterns.

To implement the motion controllers proposed, a software architecture was designed taking into account that should be easy to add new sensors and actuators with total independence on the vehicle in question. The architecture can be exported to a completely different vehicle without significant changes (only the processes responsible by interfacing with the hardware) turning it into a flexible and robust architecture. Some graphical tools to help manage the real sea tests were also developed.
6.0.1 Future work

The problems addressed while developing this work cover a vast number of fields. Some of the results obtained are preliminary and point out to possible avenues for future research.

- **Collision avoidance** - To be possible to control multiple vehicles safely in ocean, it is extremely important to have a mechanism to avoid collisions between vehicles and with obstacles in the vehicles path. Recent results in this direction can be found in [6].

- **Dynamic formations** - One future objective is to allow the vehicles to be capable of tracking a diver, following a path and maintain a formation pattern that is dependent of diver’s depth. In this case the vehicles need to adapt the formation to have better readings. This thesis presented a cooperative path-following using a constant formation along the path. The next step is to extend it to dynamic formations.
Appendix A

Controllers with currents

<table>
<thead>
<tr>
<th>X (m)</th>
<th>Y (m)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>50</td>
</tr>
<tr>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>200</td>
<td>200</td>
</tr>
<tr>
<td>300</td>
<td>300</td>
</tr>
<tr>
<td>400</td>
<td>400</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Starboard RPM</th>
<th>Portside RPM</th>
</tr>
</thead>
<tbody>
<tr>
<td>50</td>
<td>50</td>
</tr>
<tr>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>150</td>
<td>150</td>
</tr>
<tr>
<td>200</td>
<td>200</td>
</tr>
<tr>
<td>250</td>
<td>250</td>
</tr>
<tr>
<td>300</td>
<td>300</td>
</tr>
<tr>
<td>350</td>
<td>350</td>
</tr>
<tr>
<td>400</td>
<td>400</td>
</tr>
</tbody>
</table>

Figure A.1: Waypoint Controller - Simulation test with high current
Figure A.2: CPF Controller - Simulation test with high current
Appendix B

Browser Console
Figure B.1: Vehicles performing a Lawn Mower maneuver for bathymetric data acquisition
Figure B.2: Vehicles performing a circle for acoustic data acquisition
Figure B.3: Vehicles doing intersected lawn mowers for video recording
References


