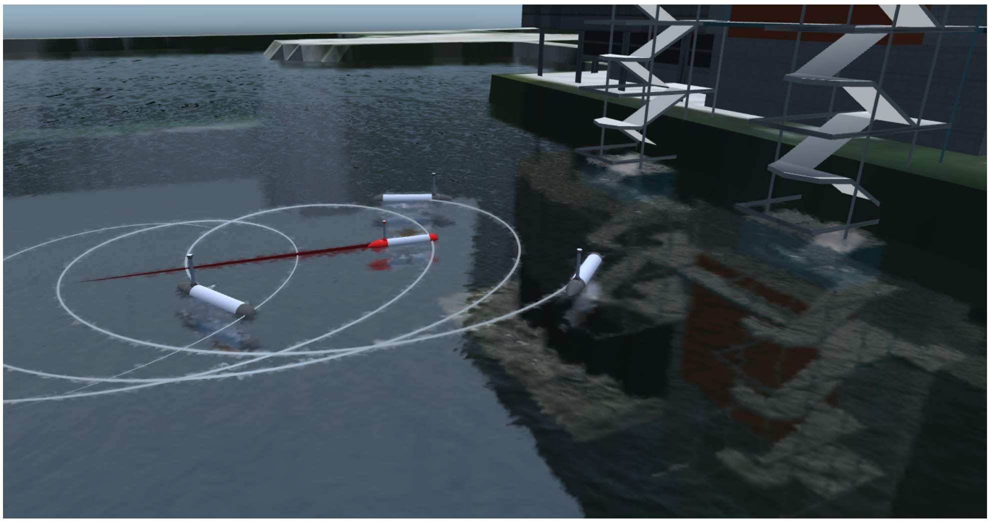


Sensor-Based Cooperative Control of Multiple Autonomous Marine Vehicles

Diogo Rebelo Teixeira

Thesis to obtain the Master of Science Degree in

Electrical and Computer Engineering

Supervisor: Prof. António Manuel do Santos Pascoal

Examination Committee

Chairperson: Professor João Fernando Cardoso Silva Sequeira
Supervisor: Professor António Manuel do Santos Pascoal
Member of the Committee: Professor Fernando Lobo Pereira

November 2019

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

I dedicate this thesis to my grandfather Adriano

Acknowledgments

This thesis represents the end of a five-year adventure. I must admit that these years were terrific, and I will always remember with some logging.

I would first like to thank my mom and dad for their love and support. Everything I did was only possible thanks to your effort! You made it possible for me to devote myself to the course entirely. Thank you for being there whenever I needed it and for the unconditional support you gave me. I hope I will someday to return all that you did for me. I feel fortunate to have you in my life!

Second, I would like to thank my supervisor, Professor António Pascoal, for his guidance and patience. I would like to thank him for the support he gave me even outside of the university and at late hours. His experience and knowledge were essential to my learning process and the work that led to this dissertation.

I would like to thank my close family for their support and affection. I would like to thank my godmother for the time and resilience she showed in teaching me so many things over the years. Even though I am stubborn, she always tried hard to teach me the things that I now recognize the importance of. I would like to thank her dedication to be the second reader of this work. To my brothers, André and Martim, a huge thanks for all the unforgettable moments, and for allowing me to be part of their childhood. To my dear Cláudia, a special thanks for your love and for being by my side all the time. Even when I felt difficulties, you were always there to help me through.

I would like to thank my friends from ”Casa Rosinha” who made this journey with me. To Fernando and Rafael, I want to thank their hospitality for having so often arranged a place for me to sleep. I would like to thank Cristian for his friendship, for being by my side during the long study hours and for being my training partner. To João, I would like to thank for the help in all those projects that I truly do not know I would ever complete without you. To Basile and Pedro, the calmest people in the whole group, I would like to thank the help they gave me. Your help was very important through this path at IST. To all of you, I hope to keep you as friends for life!

Abstract

About 70% of planet Earth is covered by water, which is largely concentrated in the oceans. From an early age, the human being showed interest in exploring this environment. However, the ocean is an adverse and hostile environment that places at risk the lives of those who dare to explore it.

In recent years, technological progress has led to major developments in the area of autonomous marine vehicles, which are the most viable option in the study and exploration of the oceans and its living or non-living resources. Given the inhospitable nature of the marine environment, the operation of these vehicles requires the development of control and navigation systems that allow them to operate in a robust and reliable manner. In particular, there is growing interest in the development of systems that enable multiple vehicles to cooperate, thereby increasing the efficiency, effectiveness, and robustness of technological solutions for the study of the oceans.

This thesis focuses the control of multiple autonomous marine vehicles acting in cooperation, using so-called cooperative control systems. With this objective in mind, we study individual vehicle control algorithms that serve as a basis for the later study of cooperative control techniques in which vehicles exchange relevant information via acoustic communication networks.

In the first part of the thesis we analyze the mathematical model of marine robotic vehicles of the MEDUSA class. These vehicles serve as basis for the work carried out in the controls area. Afterwards, the control systems of a single vehicle will be analyzed. In the 2D plane, the objective is to equip this vehicle with systems that will allow it to follow a path defined by the concatenation of a number of spatial curves, together with a speed profile along these curves (path following). For this purpose, a control structure with so-called inner-outer loops shall be used: (i) the outer loop, referred to as guidance, generates a reference for the speed and orientation of the vehicle, and (ii) the internal loop is responsible for recruiting the activity of the vehicle's actuators (propellers) so that the actual speed and orientation of the vehicle follow the references with good accuracy.

The second part of this thesis is focused on the problem of cooperative control of multiple vehicles with emphasis on the problem of cooperative path following. In this context, the objective is to make a set of vehicles to follow defined paths at a common normalized speed, and at the same time adopt a specified geometric formation, time-variant or invariant, defined according to the type of mission to be performed. Special attention is given to the fact that vehicles exchange relevant information through an acoustic communication network, that is subjected to formidable constraints due to the harsh conditions imposed by the water medium. To this end, we will study cooperative control algorithms that call for a symbiosis between theory of dynamical systems and graph theory, where the latter In this context, the central problem that we solve is that of making a group of vehicles (called trackers) follow a closed path, the center of which undergoes motions aimed at pursuing a given target.

The effectiveness of the cooperative control algorithms will be evaluated in simulation in the latter part of the work.

Keywords: Autonomous marine vehicle motion control, Cooperative control, Path following

Resumo

Cerca de 70% do planeta Terra encontra-se coberto por água, sendo que a maior porção se encontra nos oceanos. Desde cedo o ser humano mostrou interesse em explorar este ambiente. No entanto, o oceano é um ambiente adverso e hostil que coloca em risco a vida daqueles que ousam explorá-lo.

Nos últimos anos, o progresso tecnológico permitiu grandes desenvolvimentos na área dos veículos marinhos autónomos, sendo atualmente a opção mais viável para o estudo e exploração dos oceanos e dos seus recursos vivos ou não vivos. Dada a natureza inóspita do ambiente marinho, o funcionamento destes veículos requer o desenvolvimento de sistemas de controlo e navegação que lhes permitam operar de maneira robusta e confiável. Em particular, existe um interesse crescente no desenvolvimento de sistemas que permitam a cooperação de múltiplos veículos, aumentando assim a eficiência, eficácia e robustez das soluções tecnológicas para o estudo dos oceanos.

Esta tese foca o controlo de múltiplos veículos marinhos autónomos atuando em cooperação, utilizando sistemas de controlo denominados cooperativos. Com este objetivo em mente, estudamos algoritmos de controlo individual de veículos que servem como base para o estudo posterior de técnicas de controlo cooperativo em que os veículos trocam informações relevantes utilizando uma reda de comunicação acústica. Na primeira parte da tese, analisaremos o modelo matemático dos veículos marinhos robóticos da classe MEDUSA. Estes veículos servem de base para o trabalho realizado na área de controlo. De seguida serão estudados os algoritmos de controlo de movimento de um só veículo. No plano 2D, o objectivo é equipar os veículos com mecanismos que lhes permitam o seguimento de caminhos, definidos através de curvas, em conjunto com um perfil de velocidade a elas associado. Para este propósito, uma estrutura de controlo denominada loop interno-externo deve ser utilizado: (i) a malha externa, chamada de guia, gera uma referência para a velocidade e orientação do veículo, e (ii) a malha interna é responsável pelo recrutamento da atividade nos atuadores do veículo (hélices) para que a velocidade e orientação reais do veículo sigam as referências com boa precisão.

A segunda parte focada nesta tese é o controlo cooperativo de múltiplos veículos com ênfase sobre o problema do seguimento cooperativo de caminhos. Neste contexto, o objectivo é permitir que um conjunto de veículos siga caminhos definidos a uma velocidade padrão ao mesmo tempo que adota uma formação (variante ou invariante no tempo), definida de acordo com o tipo de missão a ser executada. Neste contexto, é dada especial atenção aos problemas decorrentes do facto de os veículos trocarem informação através de uma rede de comunicação acústica. Para este propósito, serão estudados agoritmos de controlo cooperativo que exigem uma simbiose entre a teoria dos sistemas dinâmicos e a teoria dos grafos, onde esta última é usada para descrever as características de comunicação acústicas entre veículos.

Na última parte do trabalho, a eficácia dos algoritmos de controle cooperativo será avaliada em simulação.

Palavras-chave: Controlo do movimento de veículos marinhos autónomos, Controlo cooperativo, Seguimento de caminhos

Contents

List of Figures

List of Tables

Glossary

AMV Autonomous Marine Vehicle. 1, 6, 13

ASC Autonomous Surface Craft. 2, 3

ASV Autonomous Surface Vehicle. 51

AUV Autonomous Underwater Vehicle. xi, 1, 2, 3, 4, 7, 51, 52

CMPF Cooperative Moving-Path Following. xii, 6, 7, 55, 57, 58, 59, 62, 64, 65, 66, 74, 75

CPF Cooperative Path Following. xi, 4, 5, 6, 7, 29, 57

DGPS Differential Global Positioning System. 23, 24

DOF Degrees of Freedom. 9, 10, 12, 77

DSOR Dynamical Systems and Ocean Robotics Laboratory. 1, 2

DVL Doppler velocity log. 23, 24

ISR Institute for Systems and Robotics. 1, 2, 3, 13, 14

LOS Line of Sight. xi, 4, 19, 20, 21, 23, 25, 28

LQR Linear Quadratic Regulator. 17, 77

MPF Moving-Path Following. xii, 5, 7, 52, 53, 54, 56

PF Path Following. xi, 4, 7, 15, 18, 19, 20, 23, 29, 33, 51, 52, 53, 55, 56, 57, 74, 75

ROS Robot Operating System. xii, 73, 74

ROV Remotely Operated Vehicle. 1

SNAME Society of Naval Architects and Marine Engineers. xiii, 10

WiMUST Widely Scalable Mobile Underwater Sonar Technology. xi, 3, 6, 51

Nomenclature

Greek symbols

$\boldsymbol{\eta}$	Position and orientation vector
γ	Synchronization parameters vector
ς	Coordination error vector
χ_p	Path course angle
ω_0	Desired angular velocity
ω_{P}	Path angular velocity
ω_n	Natural frequency
ψ	Heading angle
ψ_d	Heading reference
τ_r	Thrusters moment
τ_u	Thrusters force
$\tilde{\psi}$	Heading error
ζ	Damping factor

Roman symbols

$\boldsymbol{c}$	Target position vector
$\boldsymbol{p}$	Vector with x and y coordinates
$\boldsymbol{p}^*$	Relative position vector reference
$\boldsymbol{v}$	Surge and sway vector
$\boldsymbol{v}_c$	Velocity correction vectors
$\boldsymbol{v}_d$	Desired velocity vector
$\mathbf{v}_{\mathrm{corr}}$	Reference frames correction velocity vector
$\mathbf{v}_{\mathrm{c}}$	Ocean current velocity vector measured in $\{U\}$
$\mathbf{v}_{\mathrm{P}}$	Target linear velocity vector
$\mathbf{v}_{\mathrm{T}}$	Total velocity vector measured in $\{U\}$

$\mathcal{E}$	Edge set
$\mathcal{F}$	Complementary filter
$\mathcal{G}$	Graph
$\mathcal{M}_{pv}$	Process model
$\mathcal{N}$	Neighborhood
$\mathcal{P}$	Path
$\mathcal{VP}$	Virtual path
$\mathcal{V}$	Node set
A	Adjacency matrix
D	Degree matrix
$\mathrm{R}(\cdot)$	Rotation matrix
$\mathrm{v}_\mathrm{c}^{\perp}$	Current component perpendicular to the total velocity vector
$\mathrm{v}_\mathrm{c}^{\parallel}$	Current component parallel to the total velocity vector
$\{B\}$	Body reference frame
$\{P\}$	Path reference frame
$\{U\}$	Inertial reference frame
$\{VP\}$	Virtual path reference frame
d	Look-ahead distance
e	Cross-track error
K	Reference frames correction term bound
k_1, k_2	P. Maurya's guidance law gains
k_3, k_4	Complementary filter gains
k_c	Synchronization threshold for event triggered communications
K_e	Velocity correction term bound
L_D	Normalized Laplacian matrix
R	Circular path radius
S	Distance travelled by the vehicle's projection on the path
v_r	Formation reference velocity

CHAPTER 1

Introduction

THIS thesis addresses the general problem of designing control systems that will afford groups of Autonomous Marine Vehicles (AMVs) the capability to follow desired spatial paths with a normalized speed profile that enable the use of multiple AMVs to achieve a common goal. The goal is that vehicles follow a set of desired paths, with a speed profile that may be path-dependent and with a specific formation, that may or may not vary in time, thus achieving temporal and spatial coordination.

The thesis is motivated by real missions performed that are at the core of some of the projects undertaken at Dynamical Systems and Ocean Robotics Laboratory (DSOR) of the Institute for Systems and Robotics (ISR). The present chapter offers a brief introduction and motivates the problem that is the main focus of the thesis. In addition, the chapter summarizes the main contributions of the work and gives the thesis outline.

1.1 Background

More than two-thirds of the Earth are covered by water, without which, life on this planet would be impossible. According to scientists, around 91% of the ocean species are yet to be identified, and about 95% of the ocean area remains unexplored [25].

The oceans have always been a source of food and minerals. And, more recently, a valuable source of sustainable energy. The oceans provide also a wide range of goods and services, highly relevant to our health and the world economy. They are also a key factor in the regulation of the climate. It is therefore in the best interest of humankind that we be able to explore and take advantage of the valuable resources that the oceans harbor. However, exploring the ocean is not an easy task, and numerous lives have been lost in this endeavor. The Mariana Trench, the deepest part of the ocean, is approximately 11km deep [24]. Only Jacques Piccard and Navy Lt. Don Walsh have been able to make a descent to such depths using a U.S Navy submersible called *Trieste*, [6]. Over the last years, some unmanned submarines were able to reach this depth, revealing the potential that these vehicles have. AMVs are nowadays the best option to explore the oceans and to achieve the best use of ocean resources. AMVs can operate independently from support ships for periods of minutes to days, and some are able to explore very complex terrain. For example, the fully AUV *Sentry*, represented in figure 1.1, has been used as a stand-alone vehicle and in tandem with Alvin or a Remotely Operated Vehicle (ROV) to increase the efficiency of deep-submergence research.[18]

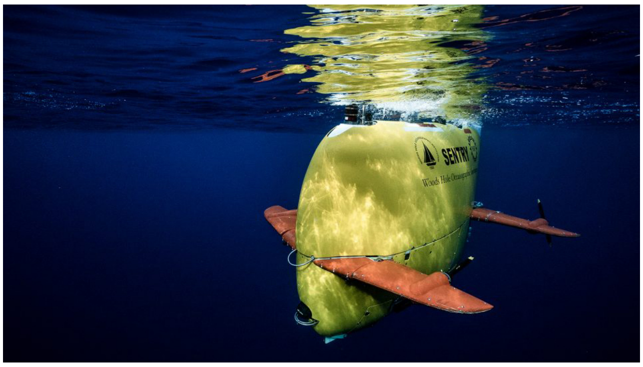


Figure 1.1: AUV Sentry [19]

Enormous efforts have been conducted over the last decades to develop more sophisticated and robust control systems for single vehicle control. More recently, more challenging problems in the area of cooperative motion control have been receiving worldwide attention. An example of a project in this area is GREX [2006-2009] entitled *Coordination and control of cooperating heterogeneous unmanned systems in uncertain environments* [9]. Both theoretical and practical issues were addressed in the scope of the project. One of the main goals was to create a conceptual framework and middleware to coordinate a group of diverse, heterogeneous physical objects (such as robotic vehicles) working in cooperation to achieve a defined practical goal in an optimized manner. In the course of this project, two vehicles, one AUV, and one Autonomous Surface Craft (ASC) were able to perform a synchronized maneuver and simultaneously maintain the vertical alignment.

CO^3-AUVs project [2009-2012], entitled *Cooperative Cognitive Control for AUVs* [35], was dedicated to Cooperative World Modeling, Cooperative Mission Execution, and Cooperative Skills. Some of the research issues addressed were the development of functionalities to monitor critical underwater infrastructures seamlessly and detect anomalous situations as well as the study of advanced AUVs capable of interacting with humans to perform functions as companion/support platforms during scientific and commercial dives.

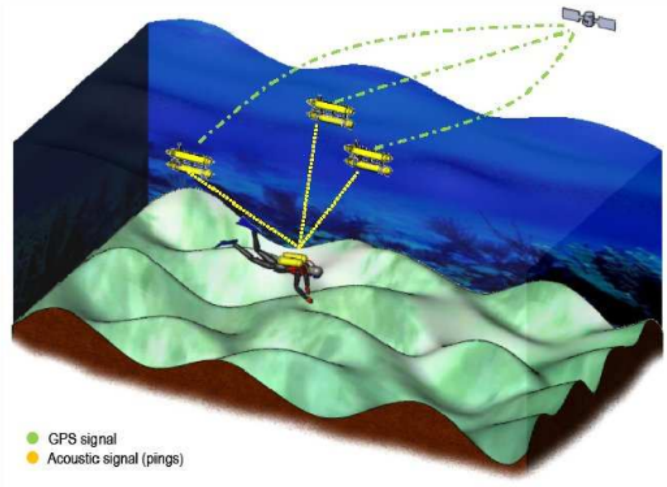


Figure 1.2: Interaction between humans and AUVs - CO^3 - AUVs project [1]

MORPH [2012-2016], entitled *Marine Robotic System of Self-Organizing, Logically Linked Physical Nodes* [10], was a R&D effort in which DSOR group of ISR participated. The work addressed the problem of cooperative control. Groups of AUVs were required to operate in areas with low visibility and unknown obstacles, where a single vehicle would have very limited capabilities. During the project a group of vehicles, composed of ASCs and AUVs, operated on a vertical cliff in cooperation, working as a large virtual vehicle. Figure 1.3 captures the maneuver scheme.

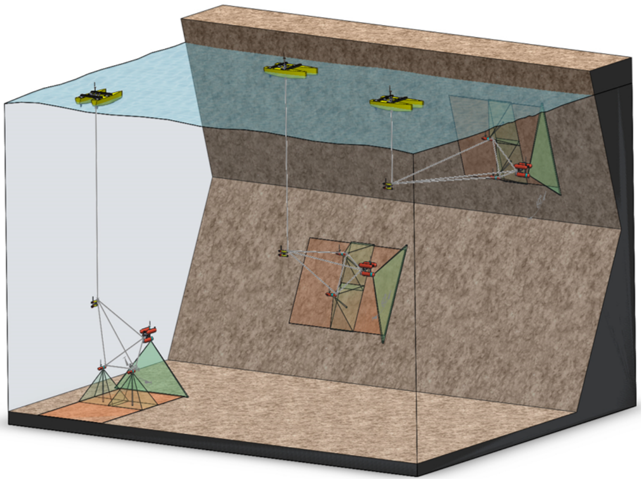


Figure 1.3: Team of agents operating as a virtual super marine vehicle at MORPH project [14]

More recently, the WiMUST [2015-2018], entitled *Widely Scalable Mobile Underwater Sonar Technology* [8], in which ISR participated, aimed at the development of advanced cooperative navigation and control systems for groups of autonomous vehicles, as a means to fully automate geotechnical surveys at sea.

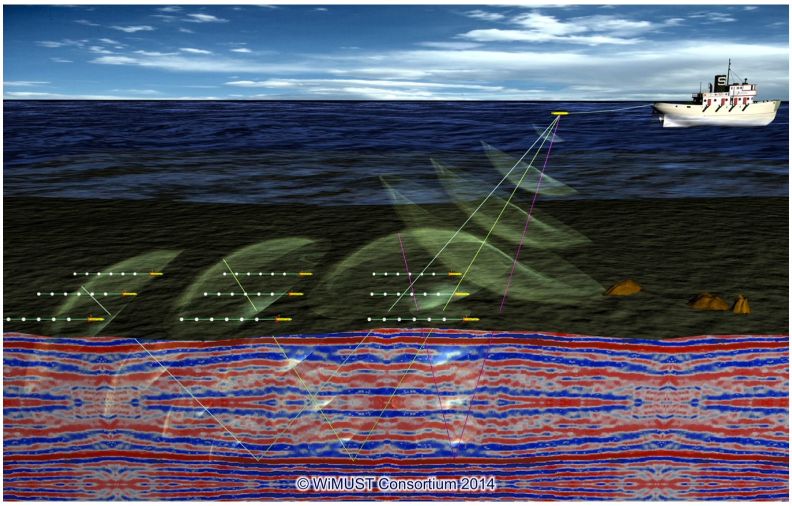


Figure 1.4: Marine vehicles following paths in a synchronized maneuver: the WiMUST project [38]

It is against the above background of ideas, namely in what regards the objectives set forth in the scope of the WiMUST project, that in this thesis we address the problem of cooperative motion control with a view to enable groups of marine vehicles perform increasingly demanding scientific and commercial missions.

1.2 State of the Art

The problem of formation control and motion coordination of multiple agent-systems has been the subject of tremendous research and development efforts in recent years. This area of study has arisen from other areas previously studied in the literature such as PF.

The PF problem can be briefly described as that of affording a vehicle the capability to follow a specified spatial path, without explicit temporal constraints, that is, without forcing the vehicle to be at a specific point in the path at a given time instant. However, it is possible to specify a desired speed profile for the vehicle as a function of where the vehicle is on the path. For PF, several control laws have been proposed and shown potential. A by now classical and pervasive PF control law at a kinematic level is LOS, summarized in Skjetne et al. [12]. LOS is quite simple in its essence and in layman's terms, it amounts to always pointing the vehicle to a point on the path ahead ahead of the closes point on the same path, the along-path distance between these two points (visibility distance) playing the role of a design or turning knob. Under some conditions, the algorithm guarantees that the cross-track error converges to zero, thus making the vehicle converge to and follow the path at a specified along-path speed. An important aspect of LOS is that the algorithm allows for the specification of the desired speed along the path, while keeping its simplicity. As a consequence, one may easily schedule the vehicle's speed as a function of the path itself. This is extremely important in a number of practical applications.

Interesting work on PF control laws was also described in Petterson et al. [11], where the authors presented a non-linear adaptive PF control law that could compensate the forces generated by the vehicle's sideslip. The proposed guidance law was intended for maneuvering AUVs or unmanned aerial vehicles inm the horizontal plane.

Another approach aimed at simplifying the presentation of PF control laws with due account for the vehicle dynamics was introduced in Maurya et al. [22]. Here the authors proposed an inner-outer loop structure that exhibits three main advantages: i) it allows for the decoupling of the vehicle's inner and outer control loops as a mean to separate kinematic and dynamic issues, ii) it does not require in depth knowledge of the vehicle dynamics in order to design the outer control loop and, iii) the PF structure can be applied to a wide range of vehicles, including aerial.

Once good path following algorithms were available, it became possible to study the problem of CPF involving multiples heterogeneous vehicles. This is a problem of considerable importance, in view of the current trend to use networked systems to explore ocean resources. The CPF, i.e, the capability of steering a group of vehicles along desired spatial paths at the same normalized speed, at the same time, holding a desired inter vehicle formation was studied in R. Ghabcheloo et al. [13], I Kaminer et al. [20], F. Vanni [37]. At the coordination level, in [37] ,the authors presented a distributed control algorithm that explicitly addressed the properties of the underlying inter-vehicle communications network that could present significant time delays.

To solve the CPF problem it is mandatory to address explicitly the constraints imposed by the topology of the inter-vehicle communications network. To tackle this issue, it is appropriate to use tools from graph theory to capture and describe the communication links. In this set-up, each node in the graph is a vehicle, while a link means that there is a communication path between two vehicles. Putting together distributed control and graph theory is the key to the study of networked systems with a view to ensuring CPF in the presence of different, possibly time-varying communication topologies. The application of these concepts to CPF was first described in [13]. A good introduction to algebraic theory, which is at the core of the work in [13] can be found in [2].

Recently, an interesting problem stated to be studied that is an extension of PF. In the new set-up, a group of vehicles are requested to converge to and follow a desired path while maintaining a desired formation along the path, the latter undergoing translational motions in response to the motion of a target to be tracked. See for example [3] for an introduction to this problem, therein referred to as the problem of circular formation control for cooperative target tracking. The solution proposed in [3] assumed that the target's velocity was known. The author proposed a control design to stabilize the multi-vehicle system to a circular motion whose center would converge to the the moving target. It was also proposed a new framework based on affine transformations to extend the results for more complex time-varying formations. By introducing a term that is, potential function in the control law, the authors derived a controller capable of maintaining the vehicles uniformly distributed along the circular formation. In later work, [4] the same authors proposed a similar strategy that relies only on the relative position of the vehicles with respect to the target, thus, requiring no information about the absolute position of the vehicles as well as not requiring knowledge of the target's speed and acceleration. The strategy lifted a set of constraints that were imposed in the earlier strategy, thus allowing for the use of this strategy on a wider range of applications. However, it introduced a region of convergence for which the center of the circular motion would converge, not guaranteeing the convergence of the position error to zero. The robustness of the proposed strategy was analyzed as well as its response to external disturbances.

It must be stressed that in [4] the formation control is done based on a exosystem. Relative references are provided for the vehicles to track. The relative references converge to the uniform formation, and later, the vehicles track the synchronized relative positions. The fact that the vehicles are tracking references makes the system less robust. Therefore this thesis introduces a method that has similarities with the so called Aguiar et al. [29] (MPF) methodology, but: i) it is very easy to implement using inner-outer structure and, ii) allows cooperation between the trackers and moving target.

The work described in the present thesis builds on the results described above. At the same time, the results of the work exposed in [33] and [34] on basic vehicle modeling and path following are used as stepping stones towards the final goal of deriving efficient CPF algorithms, including an algorithm for cooperative target tracking.

1.3 Motivation

Marine acoustic surveys are among the research methods employed to map the seafloor and the sub-seafloor. Such studies are undertaken by the oil and gas industry, the military, the fishing industry, and researchers for scientific purposes. The surveys produce detailed images of the ocean floor geology that enable the location of oil and gas deposits, fish shoals and submerged infrastructures.

Traditionally, and even in these days, acoustic surveys are carried out using large vessels that tow a high-power acoustic-source and a set of streamers, arrays of hydrophones attached to long cables. The waves emitted by the acoustic sources bounce off the ocean floor and the areas under it, deep inside sediments and other geological formations. The reflected waves are then captured by the hydrophones and the resulting signals are processed to unravel underwater geological structures, sunken man-made artifacts, and even objects of interest in the water column. *Blue Whiting Acoustic Survey* [27] is an example of an acoustic survey that is conducted with the support of large vessels. The surveys are carried out annually by the Institute of Marine Research of Norway. In the 2017 survey, four large vessels were used, one fishing vessels (*FV*) and three research vessels (*RV*), which allowed to cover a total area of more than $60\mathrm{M}^2$. *Celtic Sea Herring Acoustic Survey* is another survey carried annually, in October, in an effort to identify the whereabouts of winter spawning fish before the annual inshore spawning migration. [26]

The use of large vessels allows for fast production rates, as a single array of hydrophones may have lengths of several kilometers. However, maneuvering such systems require an enormous amount of resources, ranging from the crew needed to operate the vessels to the scientific equipment and the amount of fuel consumed during the prolonged missions, that comes, in most cases, from nonrenewable sources.

With technological developments, alternatives to the traditional methods have been studied. The EU-funded WiMUST project aimed at improving the existent methods by using a group of AMVs, each towing a shorter streamer with hydrophones. The use of autonomous vehicles allows for the decoupling of the acoustic source (transducer) from the receivers. This, in turn, leads to situations where the streamers can be moved independently of the support ship (carrying acoustic sources), thus making the whole ensemble more flexible and versatile. In the course of a mission, the set of autonomous agents may adjust their formation according to the mission being performed. During the project the researchers explored both 2D and 3D operation scenarios. In 2D, the group of agents operate at the surface and exchange information using wireless radio links. In 3D, the objective is to submerge the autonomous agents so that the distance between the acoustic sensing array and the ocean floor is reduced. When underwater, the vehicles must communicate using an acoustic network, which offers much lower data rates [17]. At this point, it should be stressed that from the geologists' standpoint, what is relevant is for cells on the seafloor to be ”hit” by the waves emitted by the acoustic sources from as many angles as possible. This can be achieved, for example, by making the acoustic sources rotate around the group of streamres. This objective calls for more advanced cooperative motion algorithms that were not addressed in the scope of WiMUST.

Motivated by the above comments and recent technological developments, in this thesis we aim at improving and providing different methods for the concerted operation of multiple autonomous vehicles. During the WiMUST project, each autonomous vehicle was performing the equivalent of CPF over a predefined path, with a reference speed chosen so as to track the two acoustic sources at the surface. One of the results of the project was that the efficiency of the maneuver was directly related to the number of available robots, as well as the quality of their coordination over the maneuver. Throughout this thesis, we will develop a control system that allows the use of a two or more vehicles towing the speakers (Acoustic Sources), while encircling the underwater vehicles towing the streamers. We formulate the problem as a CMPF task, where the agents are required to converge to a geometric path specified with respect to a moving frame, while converging to a specified formation. To ”acoustically excite” the seafloor cells from different angles we derive a method that uniformly distributes the vehicles over the path. To further improve the results we consider a circular geometric path. The target, on which we fix the moving frame, must cooperate with the emitting device so as to synchronize their positions over the path. The use of such control strategies assumes that the vehicles exchange relevant information using an acoustic communication network. Several simulation results are presented, proving the efficiency of the adopted methodology.

1.4 Objectives

The projects mentioned above show some of the possible applications for the cooperation of multiple AMVs. The main objectives of this thesis are to develop, simulate, and test a set of control systems essential for multiple vehicle cooperation.

In this thesis, vehicle modeling, single and multiple vehicle motion control will be approached. The challenges expected throughout the development of the thesis are:

Vehicle modeling - To develop control systems, it is necessary to obtain a reliable model for the used vehicle. Thus, to optimize the control of the vehicle, several steps were followed:

- Obtaining a model by resorting to physics and mathematical principles;
- Simulating the vehicle's behavior.

Single Vehicle Motion Control - Algorithms for vehicle motion control are developed. Several parameters will be tuned to achieve the desired behavior.

Cooperative Motion Control of Multiple Vehicles - By using an acoustic network, vehicles are able to communicate among them. The goal is to develop systems that given each vehicle's status, allow the vehicles to adjust their velocities to achieve coordination, and maintain a desired formation over the path.

1.5 Main Contributions

This thesis presents a series of intermediate results that culminate with a method allowing a group of vehicles to perform CPF with respect to a path defined in a moving reference frame. The motion of the latter frame is in turn governed by the objective of driving its origin to be on top of the position of an underwater target. We consider the situation where the motion of the target is not known in advance. We start with the analysis of the MEDUSA vehicle model, after which, we analyse the control system for a single vehicle and later, the control system for multiple vehicles. Simulations (in *MATLAB*), considering the MEDUSA class vehicle model, are provided to validate the efficacy of the proposed control systems. To further demonstrate the efficacy of the control systems, external disturbances, such as the ones that occur in the course of realistic missions, are considered. The main contributions of this work are listed bellow:

1. The CPF problem is formulated and solved. We present a control system that ”decouples” the coordination problem from the PF problem. The proposed strategy takes into account the communication network between the vehicles and guarantees the convergence of the vehicles to the desired formation. The adopted strategy has the particularity that the formation can be easily adjusted over the maneuver.
2. The CMPF problem is introduced, and a MPF control law is proposed. By using an external coordination controller the proposed strategy ensures that the vehicles converge to a desired formation which, in the context of the present work, corresponds to a constant and equal distance between two consecutive vehicles, i.e., to an equally spaced distribution over the path. Other formations can be achieved by simply adjusting a path parameter. The robustness of the solution against ocean currents is analyzed and a control law with current compensation is also proposed.
3. A control algorithm is proposed to adjust the moving frame linear velocity, as means to deal with temporary vehicle failures. To this end, the cross-track error of the AUVs with respect to the path is used in a logic decision scheme that controls the moving frame linear velocity. A cooperative scheme is also used to synchronize the motion of the reference frame and that of the target.
4. We present a logic based communication scheme that can be used for scenarios where the communications between vehicles have a high cost, such as underwater scenarios. The adopted strategy relies on the use of synchronized estimators. Discrete communications are performed at time instants determined by a logic function.

1.6 Thesis Outline

The structure of this thesis is as follows:

Chapter 2 introduces the MEDUSA class of autonomous vehicles used in this work and presents a model that will be later used for simulations and control design purposes.

Chapter 3 describes the control algorithms used to control the motion of a single vehicle and to guarantee that it follows a pre-defined path. Ocean currents will be introduced, and a complementary filter is analyzed. A control law that compensates the ocean currents is proposed.

Chapter 4 introduces a motivation scenario with multiple vehicles. We use the control algorithms developed in chapter 3 to individually control the motion of each vehicle. On an outer loop, a coordination controller is introduced with the aim of guaranteeing the desired inter-vehicle formation. The properties of the communication network used for the exchange of data among the vehicles are considered and taken into account by the coordination controller.

Chapter 5 presents a control design that enables a set of vehicles to perform trajectory tracking while encircling a moving target. An alternative and simpler approach, relying on trigonometry concepts, is proposed. The proposed strategy does not require any communication between vehicles. The simulation results reveal an efficacy equal to the one obtained using the original control architecture.

Chapter 6 presents a cooperative moving-path following problem and a control law is proposed. A coordination controller, that uses information from the vehicles' state guarantees convergence to the desired formation.

Chapter 7 presents a program, built in *Unity* that allow for a clean visual perception of the vehicle's behavior during a simulation of a mission in the water.

Chapter 8 summarizes the results obtained and discusses topics that warrant further research

CHAPTER 2

Autonomous Marine Vehicle Model

We start by obtaining a mathematical model for a generic marine vehicle moving in three-dimensional space, thus requiring a six Degrees of Freedom (DOF) description. For the purposes of this thesis, we will simplify the model to a three degree of freedom one, since we will focus on planar motions. Some further simplifications will be made for the surface version of MEDUSA class vehicles, the vehicles that will be used throughout this thesis. This chapter is based on the class notes of A. Pascoal. [31].

2.1 Reference Frame

Before obtaining the vehicle model, it is common practice to define two reference frames and the notation used. We start by analyzing the motion of a marine vehicle in six DOF and define two reference frames. The *Inertial Reference Frame*, $\{U\}$ is composed by the axes $\{x_U, y_U, x_U\}$, and the *Body Reference Frame*, $\{B\}$ is composed by the axes $\{x_B, y_B, z_B\}$. The two mentioned reference frames are represented in figure (2.1).

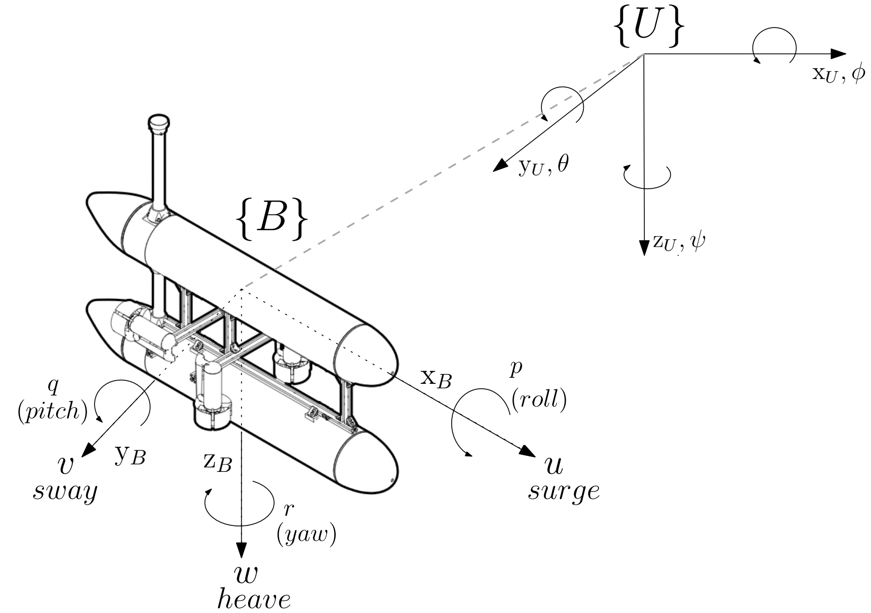


Figure 2.1: Adopted reference frames and notation

Regarding the $\{B\}$ referential, the axis $\{x_B, y_B, z_B\}$ are defined as:

- x_B - Longitudinal axis - directed from the stern to fore;
- y_B - Transversal axis - directed from portside to starboard;
- z_B - Normal axis - directed from top to bottom.

The first referential, $\{U\}$, has the property that it can be fixed to any place on earth. For simplicity, the second referential $\{B\}$ is fixed to the vehicle's center of mass, which means that the axis x_B, y_B, and z_B coincide with the principal axes of inertia.

For a general case, a marine vehicle can move in 6 DOF. Thus, six independent coordinates are necessary to describe the vehicle's position and orientation. For the vehicle's position, we use the coordinate vector $[x, y, z]^T$ and, for the vehicle's orientation the coordinate vector used is $[\phi, \theta, \psi]^T$, consisting of the angles of roll, pitch, and yaw that locally parmametrize the rotation matrix from $\{B\}$ to $\{U\}$.

We will use the nomenclature defined by the SNAME for treating motion of a submerged body through a fluid, and represent the vehicle's position and orientation, torque, speed, and forces according to the following definitions:

- $v_1 = [u, v, w]^T$: Linear velocity of the origin of $\{B\}$ with respect to $\{U\}$ expressed in $\{B\}$;
- $v_2 = [p, q, r]^T$: Rotational velocity of $\{B\}$ with respect to $\{U\}$ expressed in $\{B\}$;
- $\eta_1 = [x, y, z]^T$: Position of the origin of $\{B\}$ measured in $\{U\}$;
- $\eta_2 = [\phi, \theta, \psi]^T$: Orientation of $\{B\}$ with respect to $\{U\}$;
- $\mathbf{N}_{RB} = [K, M, N]^T$: External torques measured in $\{B\}$;
- $\mathbf{F}_{RB} = [X, Y, Z]^T$: External forces measured in $\{B\}$.

DOF	Forces And Moments	Linear And Angular Velocities	Position And Euler Angles
1 - Motion in the x-direction (surge)	X	u	x
2 - Motion in the y-direction (sway)	Y	v	y
3 - Motion in the z-direction (heave)	Z	w	z
4 - Rotation about the x-axis (roll, heel)	K	p	ϕ
5 - Rotation about the y-axis (pitch, trim)	M	q	θ
6 - Rotation about the z-axis (yaw)	N	r	ψ

Table 2.1: SNAME notation for marine vehicles (1950)

2.2 Kinematics

Kinematics is a branch of classical mechanics concerned with the motion of objects without considering the forces responsible for that motion. Recalling the notation presented in section 2.1, we can represent the kinematic equations as

$$\begin{bmatrix} \dot{\boldsymbol{\eta}}_1 \\ \dot{\boldsymbol{\eta}}_2 \end{bmatrix} = \begin{bmatrix} {}^{U}_{B}\mathrm{R}(\boldsymbol{\eta}_2) & 0_{3\times3} \\ 0_{3\times3} & Q(\boldsymbol{\eta}_2) \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \end{bmatrix}, \tag{2.1}$$

where ${}^{U}_{B}\mathrm{R}(\boldsymbol{\eta}_2)$ represents the rotation matrix from $\{B\}$ to $\{U\}$ and is obtained by performing a series of rotations along the principal axis at a specific order $z \to y \to x$, that is

$${}^{U}_{B}\mathrm{R}(\boldsymbol{\eta}_2) = \mathrm{R}_{z,\psi}\mathrm{R}_{y,\theta}\mathrm{R}_{x,\phi}. \tag{2.2}$$

It follows that the rotation matrix can be expressed as

$${}^{U}_{B}\mathrm{R}(\boldsymbol{\eta}_2) = \begin{bmatrix} \mathrm{c}\psi\mathrm{s}\theta & -\mathrm{s}\psi\mathrm{c}\phi + \mathrm{c}\psi\mathrm{s}\theta\mathrm{s}\phi & \mathrm{s}\psi\mathrm{s}\phi + \mathrm{c}\psi\mathrm{s}\theta\mathrm{c}\phi \\ \mathrm{s}\psi\mathrm{c}\theta & \mathrm{c}\psi\mathrm{c}\phi + \mathrm{s}\psi\mathrm{s}\theta\mathrm{s}\phi & -\mathrm{c}\psi\mathrm{s}\phi + \mathrm{s}\psi\mathrm{s}\theta\mathrm{s}\phi \\ -\mathrm{s}\theta & \mathrm{c}\theta\mathrm{s}\phi & \mathrm{c}\theta\mathrm{c}\phi \end{bmatrix}, \tag{2.3}$$

where s and c represent the trigonometric functions $\sin(\cdot)$ and $\cos(\cdot)$, respectively. The other mathematical entity presented in equation (2.1) is $Q(\boldsymbol{\eta}_2)$, that represents the angular velocity transformation, relating $\dot{\boldsymbol{\eta}}_2$ = $[\dot{\phi}, \dot{\theta}, \dot{\psi}]^T$ with v_2 = $[p, q, r]^T$, defined as

$$Q(\boldsymbol{\eta}_2) = \begin{bmatrix} 1 & \mathrm{s}\phi\mathrm{t}\theta & \mathrm{c}\phi\mathrm{t}\theta \\ 0 & \mathrm{c}\phi & -\mathrm{s}\phi \\ 0 & \mathrm{s}\phi/\mathrm{c}\theta & \mathrm{c}\phi/\mathrm{c}\theta \end{bmatrix}, \tag{2.4}$$

where t represents the trigonometric function $\tan(\cdot)$. The matrix $Q(\boldsymbol{\eta}_2)$ is singular for $\theta = 90^\mathrm{o}$, i.e., it is not defined for this value of θ. However, MEDUSA vehicles will operate far from this condition, which means that we can use the above Euler representation to represent the attitude of the vehicle.

2.3 Dynamics

In section 2.2, we dealt with the kinematic equations of the MEDUSA vehicles. This section studies the impact that the forces and moments that act upon the vehicle have in its motion. The method that will be used is Newton-Euler's formation using the body frame $\{B\}$. This method combines the translational and rotational dynamics of a rigid body. The rigid-body equation can be expressed as

$$M_{RV}\dot{\boldsymbol{v}} + C_{RB}(\boldsymbol{v})\boldsymbol{v} = \boldsymbol{\tau}_{RB}, \tag{2.5}$$

where M_{RV} is the rigid-body inertia matrix, C_{RB} represent the Coriolis, centripetal and gyroscopic terms, and $\boldsymbol{\tau}_{RB}$ represent the generalized vector of external forces and torques

$$\boldsymbol{\tau}_{RB} = \left[\sum F^T RB, \sum N^T RB\right]^T. \tag{2.6}$$

This generalized vector can be decomposed as

$$\boldsymbol{\tau}_{RB} = \boldsymbol{\tau} + \boldsymbol{\tau}_A + \boldsymbol{\tau}_D + \boldsymbol{\tau}_R + \boldsymbol{\tau}_{dist}, \tag{2.7}$$

where

- τ - Forces and torques due to thrusters that can be viewed as the generalized control input;
- τ_A - Forces and torques caused by the hydrodynamic added mass, studied by computing the kinetic energy imparted by the vehicle to the surrounding displaced fluid and defined as

$$\boldsymbol{\tau}_A = -M_A(\dot{\boldsymbol{v}}) - C_A(\boldsymbol{v})\boldsymbol{v}; \tag{2.8}$$

- τ_D - Hydrodynamic terms due to lift, drag, skin friction, etc. represented in the form,

$$\boldsymbol{\tau}_D = -D(\boldsymbol{v})\boldsymbol{v}; \tag{2.9}$$

- τ_R - Restoring forces and torques due to the interplay between gravity and buoyancy forces expressed as

$$\boldsymbol{\tau}_R = -g(\boldsymbol{\eta}); \tag{2.10}$$

- τ_{dist} - Terms representing external disturbances such as currents, wind, etc.

By inserting formulas (2.8),(2.9) and (2.10) in equation (2.7) and by applying the results in formula (2.5), the latter can be finally written in compact notation as

$$M_{RV}\dot{\boldsymbol{v}} + C_{RB}(\boldsymbol{v})\boldsymbol{v} + M_A(\dot{\boldsymbol{v}}) + C_A(\boldsymbol{v})\boldsymbol{v} + D(\boldsymbol{v})\boldsymbol{v} + g(\boldsymbol{\eta}) = \boldsymbol{\tau} + \boldsymbol{\tau}_{dist}. \tag{2.11}$$

2.4 Simplified Equations

In the previous sections we analyzed the vehicle's model for a generic case with six DOF. Regarding this thesis, the vehicle will operate in the xy plane. This assumption reduces the number of DOF to three, since $\phi = 0$, $\theta = 0$ and $z = 0$. The rotation matrix defined in (2.3) can be simplified for this specific case to yield

$$\mathrm{R}(\psi) = \begin{bmatrix} 0 & \cos(\psi) & -\sin(\psi) \\ 0 & \sin(\psi) & \cos(\psi) \\ 1 & 0 & 0 \end{bmatrix}. \tag{2.12}$$

By applying these new conditions to the kinematic model, we can rewrite it as

$$\begin{aligned} \dot{x} &= u \cdot \cos(\psi) - v \cdot \sin(\psi) \\ \dot{y} &= u \cdot \sin(\psi) + v \cdot \cos(\psi) \\ \dot{\psi} &= r. \end{aligned} \tag{2.13}$$

We can also apply these conditions to the vehicle's dynamics, neglecting roll, pitch, and heave motion. In this situation, the dynamic equations involving u, v and r are described by

$$m_u\dot{u} - m_v vr + d_u u = \tau_u, \tag{2.14a}$$

$$m_v\dot{v} + m_u ur + d_v v = 0, \tag{2.14b}$$

$$m_r\dot{r} - m_{uv}uv + d_r r = \tau_r, \tag{2.14c}$$

where τ_u represents the forward thruster force, τ_r represents the thruster torque about the z-axis, and

$$\begin{aligned} m_u &= m - X_{\dot{u}} \\ m_v &= m - Y_{\dot{v}} \\ m_r &= I_z - N_{\dot{r}} \\ m_{uv} &= m_u - m_v \end{aligned} \qquad \begin{aligned} d_u &= -X_u - X_{u|u|}|u| \\ d_v &= -Y_v - Y_{v|v|}|v| \\ d_r &= -N_r - N_{r|r|}|r|, \end{aligned} \tag{2.15}$$

where m_u, m_v, m_r and m_{uv} represent mass and hydrodynamic added mass and d_u, d_v and d_r are hydrodynamic damping effects.

2.5 MEDUSA Vehicle Characterization

As mentioned before, during this thesis, a particular class of vehicles will be used, the MEDUSA-class. The MEDUSA AMVs are autonomous submerged/semi-submerged robotic vehicles developed at the Laboratory of Robotics and Systems in Engineering and Science (LARSyS)/ISR of the Instituto Superior Técnico de Lisboa, Portugal.

Each MEDUSA class vehicle weighs approximately 30kg in the air and consists of two acrylic housing tubes of size 150mm by 1035mm (diameter x length) with Aluminum end caps, attached to a central Aluminum frame. The distance between bodies is 150mm [23].

The two tubes contain the components required for the vehicle to navigate. The lower tube, the one that goes underwater, contains the 7-cell lithium polymer batteries, the acoustic modem essential for communications and the echo-sounder. The upper tube, that navigates partially out of the water, contains the majority of the components such as the main computer (Epic computer NANO PV D5251, with an Intel Atom D525 dual-core processor, low power, 1.8GHz with 2GB RAM and 64GB solid-state disk drive), the navigation sensors, that provide readings on the angular positions of the vehicle (ϕ, θ, ψ), and the Global Positioning System, responsible for providing the vehicle's position in inertial frame $\{I\}$. More detailed information about the MEDUSA components can be found in picture 2.2.

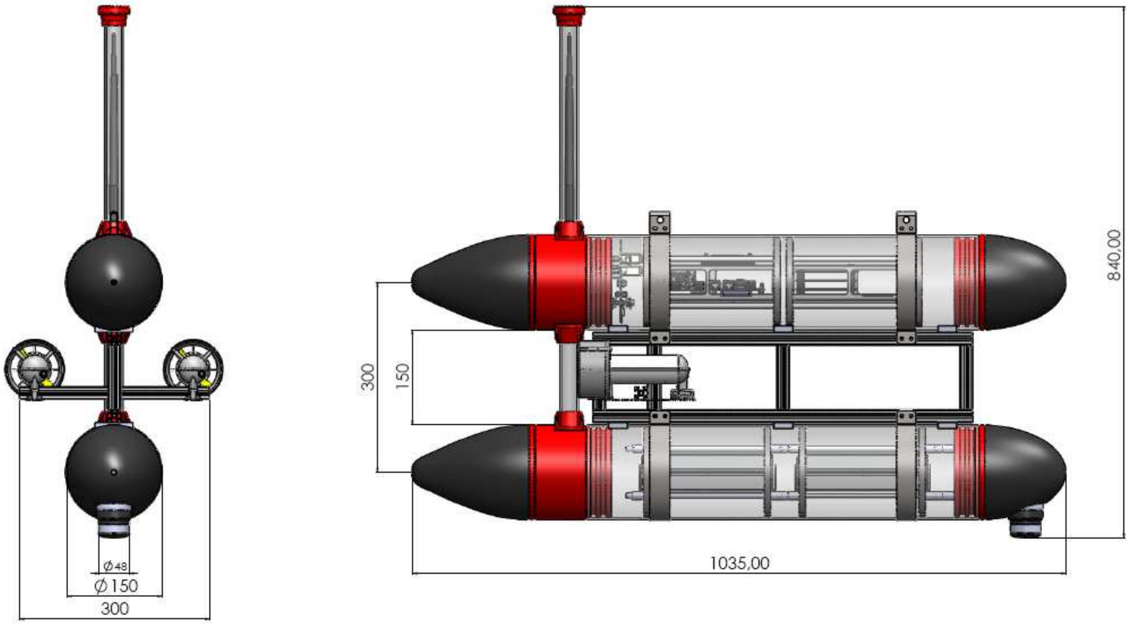


Figure 2.2: Basic organization of the MEDUSA vehicles

Since there is no actuator to control sway, we can resort to the model defined in (2.14) and replace τ_r by 0. The values for the model parameters were obtained by previous researchers at ISR and are listed in table 2.2.

$X_{\dot{u}}$	-20 kg	$Y_{\dot{v}}$	-30kg	$N_{\dot{r}}$	-8.69 kg m^2
X_u	-0.2 kg s^{-1}	Y_v	-50 kg s^{-1}	N_r	-4.14 kg m s^{-1}
$X_{\|u\|u}$	-25 kg s^{-1}	$Y_{\|v\|v}$	0.01 kg m^{-1}	$N_{\|r\|r}$	-6.23 kg m

Table 2.2: Medusa Model Parameters

The previous table presented the normalized model parameters. There are still two parameters missing, that correspond the mass and moment of inertia about the z-axis, given by $m = 17$kg and $I_z = 1$kg m^2, respectively.

CHAPTER 3

Single Vehicle Motion Control

In the previous chapter we summarized a mathematical model for the vehicle. We now address the problem of single vehicle motion control for path following. To this end, we adopt an inner-outer loop control structure. At the inner loop level, we design a heading controller that makes the vehicle's heading track a desired reference, using the differential mode in the thrusters. At the outer loop level, we design a controller (guidance law) that generates heading references for the inner loop taking into consideration the path and the vehicle's position with respect to the path. The speed is also an important aspect in PF as the vehicles are required to navigate along the path with a specific speed profile. For this purpose, we derive a control law that acts on thrusters' common mode to ensure that the actual vehicle speed track desired speed profiles. Throughout this chapter, we will analyze and simulate the algorithms necessary to control the motion of a single vehicle.

The architecture of the system necessary to control the motion of a single vehicle is represented in figure 3.1. In the figure, it is possible to see part of the information exchanged between the different controllers.

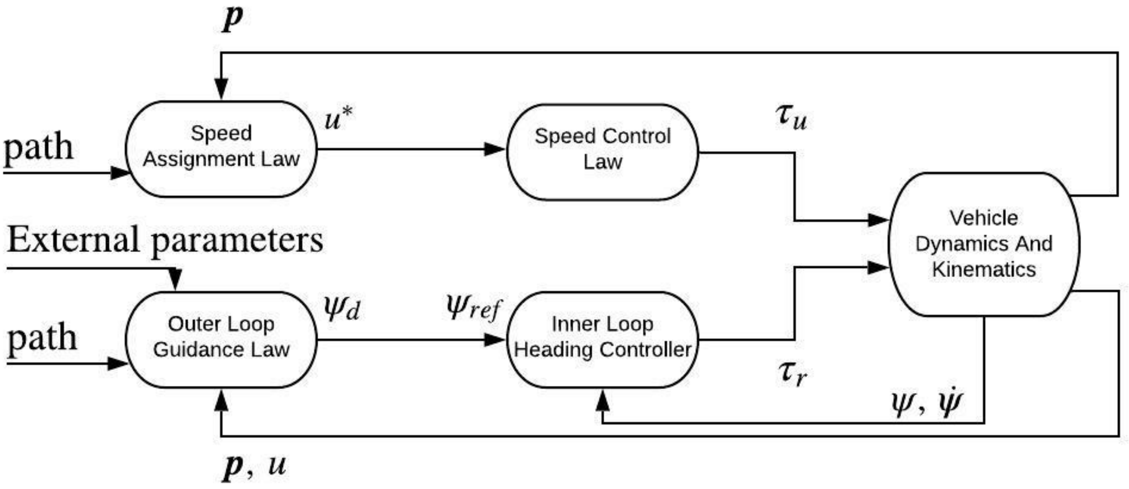


Figure 3.1: Single vehicle motion: PF system architecture

3.1 System Model

We start by recalling the dynamic model obtained in (2.14). At this stage, we assume that the vehicle is equipped with a local speed control law that keeps the forward speed u constant. This assumption allows us to ignore equation (2.14a), since $\dot{u} = 0$, and only take into consideration equations (2.14b) and (2.14c). Then, by resorting to the kinematic model obtained in (2.13) and, after some manipulations, we acquire the model

$$\begin{aligned} \dot{v} &= \frac{1}{m_v} \cdot (-m_u r - d_v v) \\ \dot{r} &= \frac{1}{m_r} \cdot (-d_r r + m_{uv} v + \tau_r) \\ \dot{\psi} &= r. \end{aligned} \tag{3.1}$$

If we consider the equations for the hydrodynamic added mass and the hydrodynamic damping effects, expressed in (2.15), we conclude that the model above is not linear, meaning that it is not possible to apply linear control design tools to obtain a heading controller. We therefore linearize the model about the vehicle's operating conditions, defined by $v \approx 0$ and $r \approx 0$. This assumption allows us to rewrite some parcels from (2.15) since the quadratic terms are neglectable,

$$\begin{aligned} d_r &= -N_r - N_{r|r|}|r| \approx -N_r \\ d_v &= -Y_v - Y_{v|v|}|v| \approx -Y_v. \end{aligned} \tag{3.2}$$

By inserting (3.2) into (3.1) yields the linear model

$$\begin{aligned} \dot{v} &= \frac{1}{m_v} \cdot (-m_u r + Y_v v) \\ \dot{r} &= \frac{1}{m_r} \cdot (N_r r + m_{uv} v + \tau_r) \\ \dot{\psi} &= r. \end{aligned} \tag{3.3}$$

Let $\boldsymbol{x} = [\psi, r, v]^T$ be the system's state, $y = \psi$ the system's output, and $u = \tau_r$ the system's input. With this notation the linear model obtained in (3.3) admits the representation

$$\begin{aligned} \dot{\boldsymbol{x}} &= \begin{bmatrix} 0 & 1 & 0 \\ 0 & \frac{N_r}{m_r} & \frac{m_{uv}}{m_r} \\ 0 & -\frac{m_u}{m_v} & \frac{Y_v}{m_v} \end{bmatrix} \boldsymbol{x} + \begin{bmatrix} 0 \\ \frac{1}{m_r} \\ 0 \end{bmatrix} u \\ y &= \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \boldsymbol{x}. \end{aligned} \tag{3.4}$$

3.2 Heading Controller

The goal for this section is to design an inner loop for the heading angle to track a reference ψ_{ref}. The outer loop will later be analyzed and developed. We start by defining the heading error as

$$\tilde{\psi} = \psi - \psi_{ref}, \tag{3.5}$$

where ψ is the yaw angle and $\tilde{\psi}$ the heading error. We adopt the state variable model presented in (3.4) to design the heading controller, by assuming that the pair (A, B) is controllable, and the pair (A, C) is observable. These properties allow the use of the Linear Quadratic Regulator (LQR) to design the controller for the system.

We start by projecting a regulator that stabilizes the system about the origin, i.e., guarantees convergence of all state variables to the origin. Since the goal is to design a heading controller that makes the vehicle's heading track a desired reference, that does not need to be necessarily 0, we will then modify the feedback scheme to obtain a servomechanism that allows tracking heading references. We start by defining a quadratic cost function denoted J, to be minimized, given by

$$J = \int_0^{+\infty} (\boldsymbol{x}^T Q \boldsymbol{x} + u^T R u) dt, \tag{3.6}$$

where Q is a real symmetric positive-definite matrix and R is a positive constant that penalize the state error and the input, respectively. It is crucial to balance these two parameters to obtain the desired response. Since we are dealing with variables representing physical quantities, we have variables with drastically different orders of magnitude. One possible way of overcoming this problem, using so-called Bryson's rule [5], is to to define the matrices Q and R so that we normalize all variables according to the maximum values of their absolute values. In the present case, defining the normalizing vector

$$\boldsymbol{\eta} = [\frac{1}{\psi_{max}}, \frac{1}{r_{max}}, \frac{1}{v_{max}}], \tag{3.7}$$

it follows that the normalized state vector x can be rewritten as $\eta \boldsymbol{x}$. Thus, $\boldsymbol{x}^T \boldsymbol{x} = \boldsymbol{x}^T Q \boldsymbol{x}$, where

$$Q = \boldsymbol{\eta}^T \boldsymbol{\eta}, \tag{3.8}$$

which is the matrix on (3.6). Similarly, R in (3.6) is given by

$$R = \frac{1}{\tau_{max}}. \tag{3.9}$$

If the pair (A, B) is controllable and the pair (A, C) is observable, the solution for the LQR problem exists and is unique. The solution can be written in state feedback form as

$$u = -K\boldsymbol{x}, \tag{3.10}$$

where K is a gain matrix, given by

$$K = R^{-1} B^T P, \tag{3.11}$$

where P is the only positive-definite solution of the algebraic Riccati equation [28]

$$A^T P + PA - PBR^{-1} B^T P + Q = 0. \tag{3.12}$$

Furthermore, this closed form solution yields asymptotically stable closed loop solution.

From the above regulator structure, a servomechanism is obtained by introducing the heading error and computing u_d as

$$u_d = -k_p \cdot \tilde{\psi} - k_d \cdot \dot{\psi}, \tag{3.13}$$

where $u_d = \tau_r$ represents the differential mode of the thrusters, k_p represents the proportional gain, and k_d represents the derivative gain.

After normalizing the variables, we can adjust the weights of each parameter by multiplying its entry in Q, R by a weight factor. A higher Q will make the system converge faster to the desired value, regardless of the control effort. A higher R will ensure that the control signal remains small, even if it takes more time for the system to reach the goal.

The above computations can be done using *MATLAB*. The system model can then be implemented in *Simulink* together with the designed control law. Figure 3.2 shows the result of a simulation performed with the heading controller.

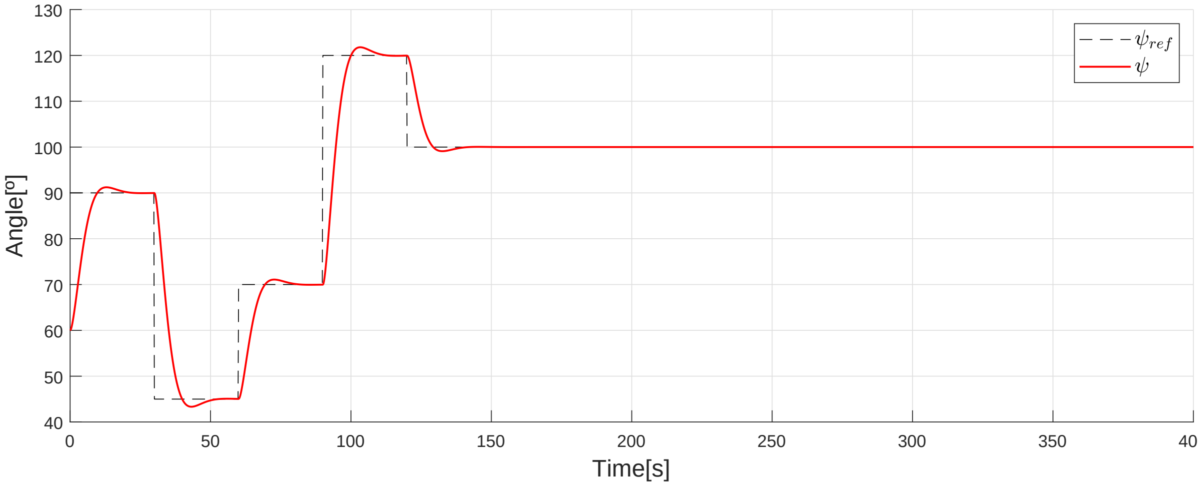


Figure 3.2: Heading Controller Simulation

By analyzing figure 3.2, we conclude that the designed controller is capable of steering the vehicle's heading towards the reference ψ_{ref} without significant oscillations. The simulation was conducted with the vehicle navigating with a surge speed, u, of approximately $1\,\mathrm{m\,s^{-1}}$ as it corresponds to the value for which the model's linearization was made.

3.3 Speed Control

Before starting the design of the speed control law it is important to note that the MEDUSA vehicle's model implemented in *Simulink* is not linear. The model presents a non-zero velocity in sway. Nevertheless, and considering its order of magnitude, the velocity v (sway) can be neglected. For this reason, we will derive a control law for u (surge).

In PF we must take into account that the vehicle is required to follow a path with a specific speed profile without explicit temporal restrictions. Since, by mission design, the speed of the vehicle may by path dependent, at each computation time, we determine the projection of the vehicle into the path and provide, as speed reference, the speed previously assigned at the projection point. By going back to

(2.14) and to (2.15), we can neglect the vehicle's accelerations over the x axis. Assuming that there is no velocity in sway (making $v = 0\,\mathrm{m\,s^{-1}}$) and after some manipulations, we obtain an equation that connects the desired speed u^* with the forward thrusters' force τ_u, given by

$$\tau_u = u^* d_u = u^*(-X_u - X_{u|u|}|u^*|). \tag{3.14}$$

The previous control law ensures that given a specific speed, we can act on the thrusters' common mode in order to achieve that speed. A simulation was performed where the speed reference was changed over time to validate the control law. The simulation was conducted without differential mode in the thrusters, meaning that the vehicle was moving along a straight line. The results of this simulation are presented in figure 3.3.

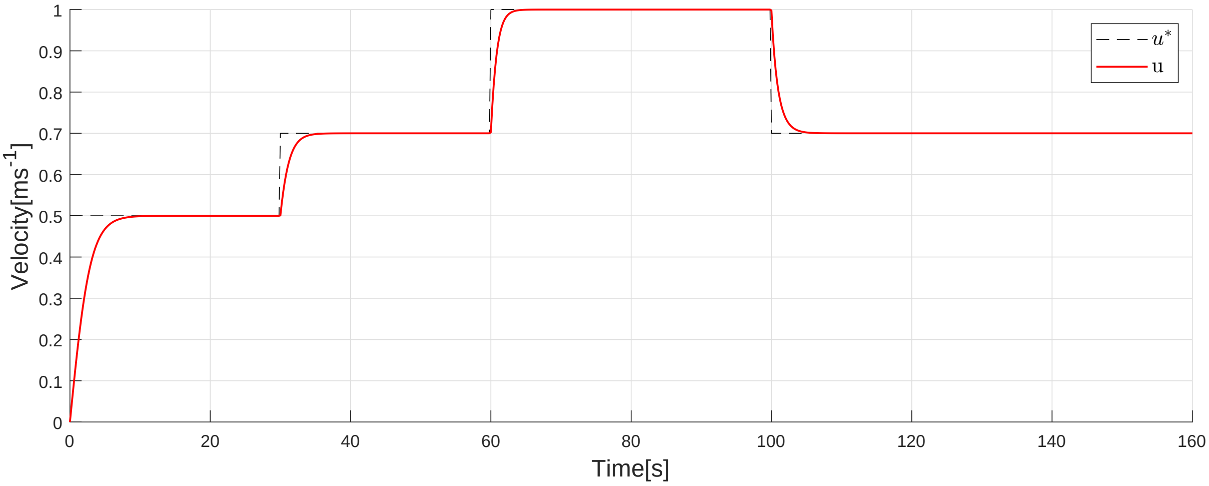


Figure 3.3: Speed Control Simulation

By examining figure 3.3, we conclude that the law from (3.14) is capable of making the vehicle move at a desired speed.

3.4 Guidance Laws for Straight Lines

In section 3.2, we analyzed and designed an inner loop capable of making the vehicle track a desired heading reference, at a desired speed. In this section, we will analyze the vehicle's outer loop. This control system, often times called guidance loop, will provide a reference angle based on the desired path and the vehicle's current position. Given an initial condition, the algorithm should be able to make the vehicle converge to the path, that is, make the cross-track error tend to zero.

3.4.1 Line of Sight - Guidance Law

The first method of PF that we study is LOS, a method that is pervasive in the literature [21]. The main idea of LOS is to mimic the actions of a cheetah when hunting prey. Instead of running towards the prey, the cheetah aims towards a point slightly in front of the prey, thus achieving smooth motions while reducing energy expenditure.

We consider a path defined by successive waypoints and that a straight line connects two consecutive waypoints. This line forms an angle χ_p with x-axis of the *Inertial Reference Frame*. The first step is to determine the vehicle's projection on the path, corresponding to the point on the path that is closest to the vehicle. Then, we provide a safety margin, also called look-ahead distance, yielding the point where we want to aim the vehicle. This point, characterized by its coordinates, is known as the LOS point. The concepts and nomenclature used by the LOS guidance law are captured in figure 3.4.

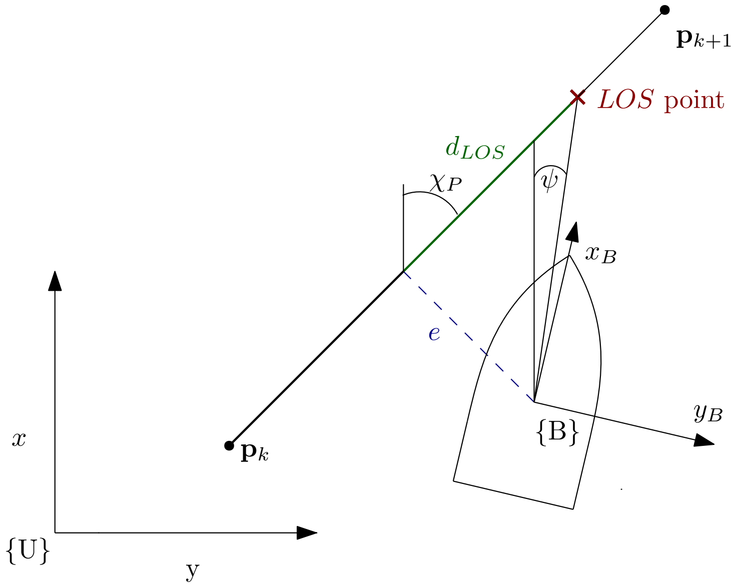


Figure 3.4: Line of sight guidance law

In figure 3.4, it is possible to see a path characterized by two waypoints connected by a straight line. The two consecutive waypoints are represented as $\mathbf{p}_k$ and $\mathbf{p}_{k+1}$. The cross-track error, e, is the distance between the vehicle's mass center and its projection on the path. The goal of PF is to minimize this distance, thus forcing the vehicle to converge to the path. The look-ahead distance appears in the figure, represented by d_{LOS}. The output of the control algorithm is an angle, ψ_d, that is given as reference to the heading controller designed above. Still to define are the equations for the angles involved in the LOS principle, given by

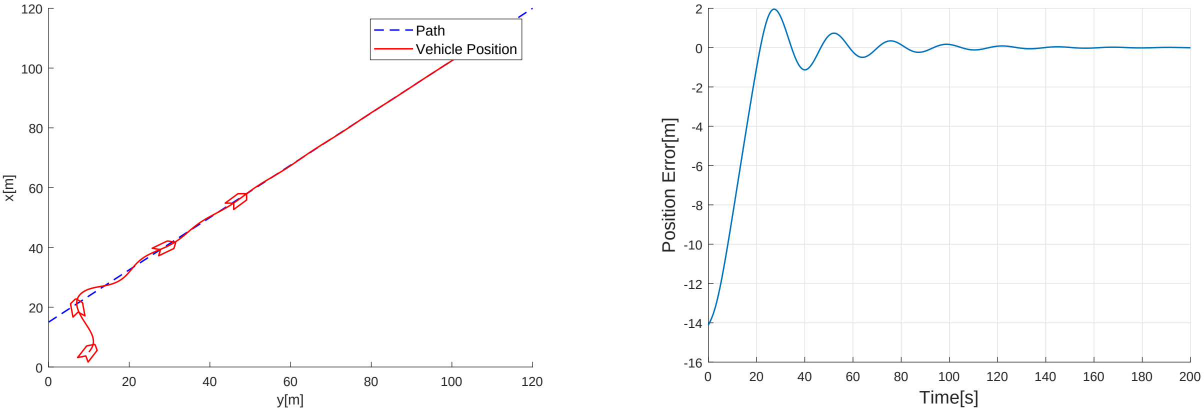
$$\chi_p = \text{atan2}(y_{k+1} - y_k, x_{k+1} - x_k), \tag{3.15a}$$

$$e = -\sin(\chi_p)(x - x_k) + cos(\chi_p)(y - y_k), \tag{3.15b}$$

$$\psi_d = \arctan(-\frac{e}{d_{LOS}}) + \chi_p. \tag{3.15c}$$

Considering that the desired angle is given by (3.15c) and that the heading angle is perfectly tracked, i.e., $\psi = \psi_d$, then e converges exponentially to the origin. The proof is given in [21], by deriving the equation for the cross-track error, (3.15b), and introducing a Lyapunov function candidate.

Several simulations were performed to assess the performance of the PF guidance control law. We started by providing a path aligned with the vehicle's initial orientation, and an initial cross-track error equal to zero. The results of this simulation show that the vehicle was able to follow the predetermined path. After this, we defined a path that was not aligned with the vehicle's initial state and considered an initial cross-track error different from zero. The aim was to see if the vehicle would converge to the path and then follow it. The results obtained for this simulation are represented in figure 3.5. We conclude that the vehicle was able to achieve the aim. The distance e reduced with time, showing that the vehicle was converging to the path.

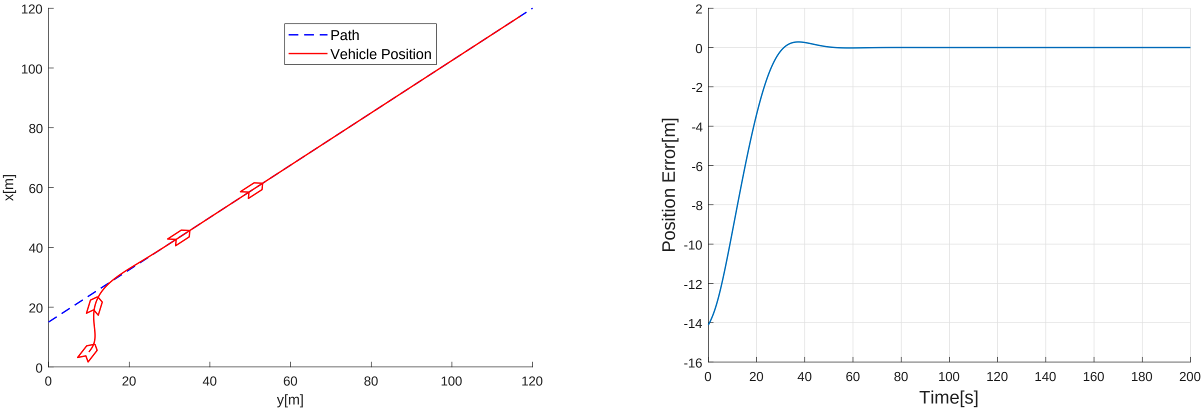


(a) Vehicle's trajectory

(b) Cross-track Error

Figure 3.5: Simulation results for a straight path using LOS guidance law with $d_{LOS} = 3\text{m}$

The simulation presented in figure 3.5 was conducted using a look-ahead distance equal to 3m. It is possible to see that the vehicle converged to the path as expected. However, it assumed an undesired oscillating behavior. A short look-ahead distance usually causes this behavior. Thus, we increased it to 8m. The results are represented in figure 3.6.



(a) Vehicle's trajectory

(b) Cross-track Error

Figure 3.6: Simulation results for a straight path using LOS guidance law with $d_{LOS} = 8\text{m}$

By comparing the results represented in figures 3.5 and 3.6, we conclude that the look-ahead distance can be used as a ”tune parameter”, capable of changing the vehicle's behavior as it approaches the path. By setting a lower value, the vehicle should converge faster to the path. However, it will have a more oscillating behavior, since we are heading the vehicle towards a point closer to its projection. If we use a higher value, the vehicle should oscillate less around the path, which is good. The negative side for choosing a higher value is that the vehicle will converge to the path more gradually, which means that it will navigate a greater distance while approaching the path.

3.4.2 P. Maurya - Guidance Law

An alternative to the guidance law presented in (3.15c) is the P. Maurya et al. guidance law [22]. As above, the objective is to steer the vehicle towards the path, forcing e to converge to the origin. By assuming that the side-slip angle is zero, we get a simplified case, and we can write

$$\dot{e} = u\sin(\psi) + v_{cy}, \tag{3.16}$$

which, in the absence of v_{cy} (representing the component of the ocean current in the inertial y-axis) can be written as

$$\dot{e} = uU, \tag{3.17}$$

where $U = \sin(\psi)$. By making the observation that the variable U should be free to manipulate and by setting $U = -k_1 e$ it is trivial to show that we can ensure asymptotic convergence of the cross-track error to the origin. However, the possible existence of a bias v_{cy} motivates the introduction of an additional integral term of the cross-track error to the virtual input U, that can be re-written as

$$U = -\frac{1}{u}\left(k_1 e + k_2 \int_0^t e\, d\tau\right). \tag{3.18}$$

By replacing the previous equation on the dynamics of e, (3.17), we get

$$\dot{e} + k_1 e + k_2 \int_0^t e\, d\tau = 0. \tag{3.19}$$

By introducing a new variable $\zeta = \int_0^t e d\tau$, equation (3.19) can be rewritten as

$$\ddot{\zeta} + k_1\dot{\zeta} + k_2\zeta = 0. \tag{3.20}$$

The equation above characterizes a second-order system and, the gains k_1 and k_2 should be chosen to obtain the desired natural frequency and damping factor. Thus, the relation between the gains k_1 and k_2 with the desired natural frequency and damping factor is given by

$$k_1 = 2\omega_n\zeta, \tag{3.21a}$$

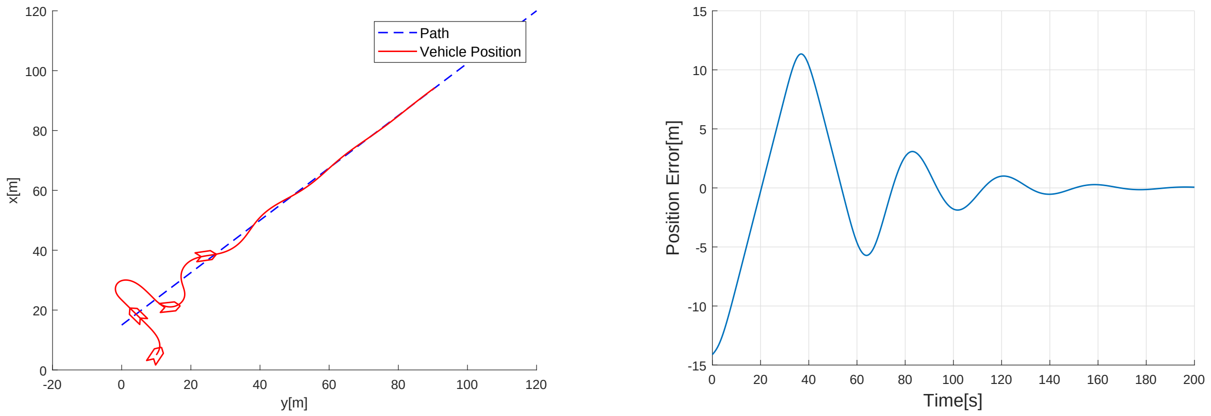
$$k_2 = {\omega_n}^2. \tag{3.21b}$$

Taking into consideration the error dynamics (3.16), that the variable U should be free to be manipulated, and the intended expression for U, it becomes clear that the desired angle must be of the form $\psi_d = \sin^{-1}\left(\text{sat}(U)\right)$. Thus, the virtual control law can then be written as

$$\psi_d = \sin^{-1}\left(\text{sat}\left(-\frac{k1}{u}e - \frac{k_2}{u}\int_0^t ed\tau\right)\right), \tag{3.22}$$

where $\text{sat}(\cdot)$ is the saturation function limited to the range [-1,1]. As mentioned in [22] it is possible to show, using Lyapunov-based analysis, that the non-linear control law presented in (3.22) yields convergence of the cross-track error to zero if the actual vehicle heading equals the desired heading reference ψ_d.

As in the previous algorithm, some simulations were performed in order to observe the vehicle's behavior and the control law performance.

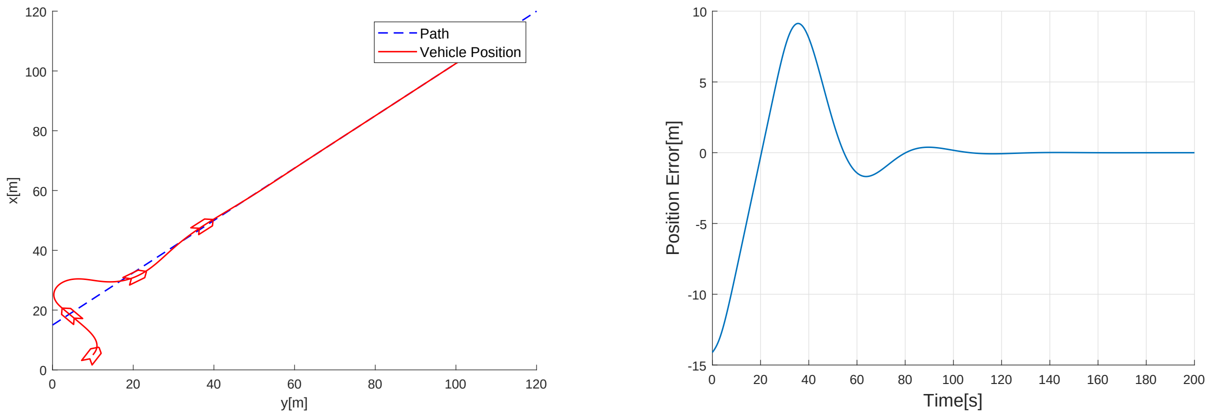


(a) Vehicle's trajectory

(b) Cross-track Error

Figure 3.7: Simulation results for a straight path using P. Maurya's guidance law with $\omega_n = 0.12$ and $\zeta = 0.7$

The results obtained, represented in figure 3.7, show that P. Maurya control law also ensures that the vehicle follows a straight line. However, there is a considerable oscillation as the vehicle converges to the path. This oscillation is higher than expected as the damping factor ζ was set to 0.7. However, we must take into consideration that in the model used in the simulation (MEDUSA vehicle model), both dynamics and kinematics are considered. Thus, the actual vehicle's heading does not correspond exactly to the heading reference, ψ_d. The convergence of the cross-track error to zero was verified, although with a remarkable oscillation. To reduce this oscillation, we may reduce the natural frequency of the system, ω_n. Another experiment was conducted using a lower value for this parameter. The change of the value of ω_n resulted in a less pronounced oscillation. The results obtained are represented in figure 3.8.



(a) Vehicle's trajectory

(b) Cross-track Error

Figure 3.8: Simulation results for a straight path using P. Maurya's guidance law with $\omega_n = 0.08$ and $\zeta = 0.7$

3.4.3 Current Estimation

There is no sensor capable of providing directly the inertial water velocity. This can only be done indirectly. For example, when the vehicle is at the surface, the water velocity can be obtained using a Doppler velocity log (DVL) and Differential Global Positioning System (DGPS), as explained below. Estimating the current (water velocity) is of crucial importance in some PF methods, for example LOS. We will base our work on the results obtained in Oliveira et al. [30], and that of G. Sanches [33]. The problem at hand is that of estimating the ocean current based on measurements $\mathbf{p}_m$ and $\mathbf{v}_m$ of the vehicle's inertial position and velocity with respect to the water, respectively. The first is given by a DGPS unit, while the second is provided by a DVL.

We start by going back to the kinematics simplified model, (2.13). We define a new vector for the vehicle's inertial position $\mathbf{p} = [x, y]^T$ and a vector for the linear velocity $\mathbf{v} = [u, v]^T$. The ocean current is assumed to be constant and is represented by $\mathbf{v}_\mathrm{c}$, which can be decomposed as $[\mathrm{v}_{c_x}, \mathrm{v}_{c_y}]^T$. The kinematic equations can now be written as

$$\dot{\mathbf{p}} = \mathrm{R}(\psi)\mathbf{v} + \mathbf{v}_\mathrm{c}. \tag{3.23}$$

The process model $\mathcal{M}_{pv}$ is given by

$$\mathcal{M}_{pv} = \begin{cases} \dot{\mathbf{p}} = \mathbf{v}_m + \mathbf{v}_\mathrm{c} \\ \dot{\mathbf{v}}_\mathrm{c} = 0 \\ \mathbf{p}_m = \mathbf{p} \end{cases} \tag{3.24}$$

We recall that we are studying the case of an ocean surface vehicle where the position measurement $\mathbf{p}_m$ is provided by a DGPS and $\mathbf{v}_m$ is provided by a Doppler Sonar.

Given the above process model, it is possible to develop a complementary filter with the realization

$$\mathcal{F} = \begin{cases} \dot{\hat{\mathbf{p}}} = k_3(\mathbf{p}_m - \hat{\mathbf{p}}) + \mathbf{v}_m + \hat{\mathbf{v}}_\mathrm{c} \\ \dot{\hat{\mathbf{v}}}_\mathrm{c} = k_4(\mathbf{p}_m - \hat{\mathbf{p}}) \end{cases}, \tag{3.25}$$

where k_3 and k_4 are gains, and $\hat{\mathbf{p}}$ and $\hat{\mathbf{v}}_\mathrm{c}$ represent, respectively, the position estimates and current velocity estimates. In [30], the authors showed that for $k_3, k_4 > 0$ the above filter is asymptotically stable.

To confirm the efficacy of the estimator and that the values of $\hat{\mathbf{v}}_\mathrm{c}$ converge to the real values of the ocean currents, we implemented the complementary filter and performed a simulation with non-zero ocean currents. The results are shown in figure 3.9. The estimators are initialized at the origin, i.e., the initial values of the estimates are set to zero. Thus, during the initial instants, there exists a significant error between the estimates and the ”actual” values of the current. The values of k_3 and k_4 can be tuned to change the filter behavior. The simulation was conducted with $k_3 = 2$ and $k_4 = 0.2$. It is possible to see an initial oscillation as the estimates converge to the real values. As time goes to infinity, the values of the estimates stabilize over at the actual values. Thus, we conclude the design of the ocean current estimator.

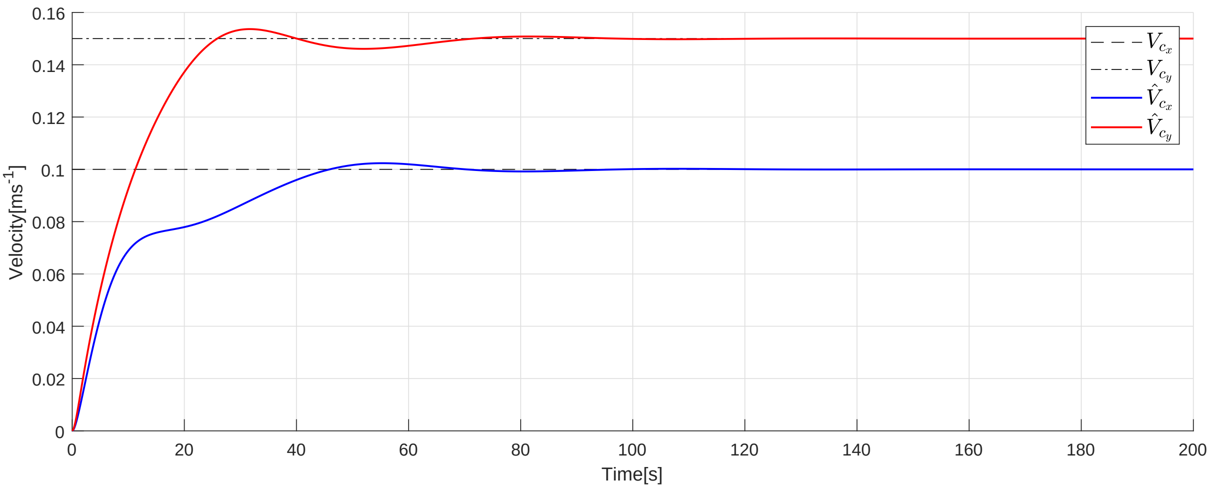
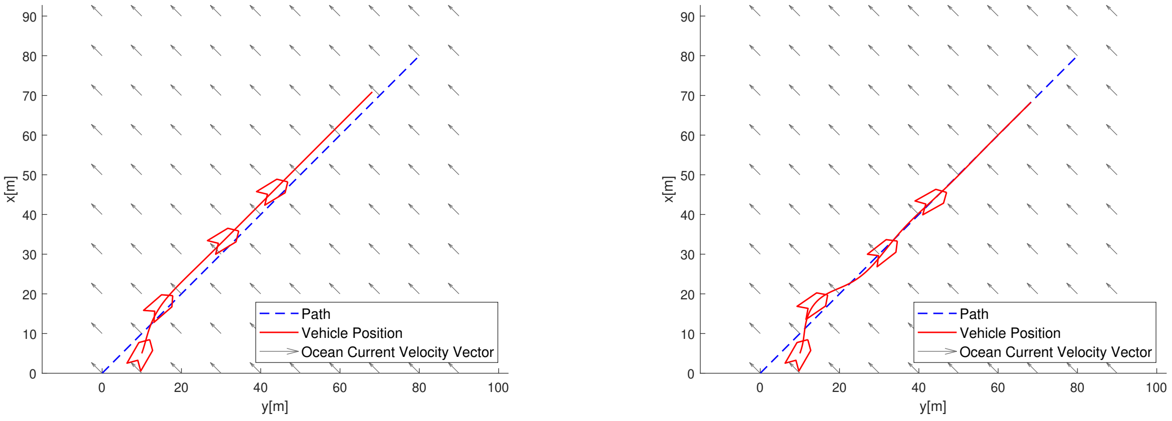


Figure 3.9: Current estimation using a complementary filter

3.4.4 Current Compensation

At this point it is important to study the vehicle's behavior and its approach to the path when there exist ocean currents which, in the case of LOS path following, requires proper compensation. For straight lines, both control laws, (3.15) and (3.22) were able to make the cross-track error converge to the origin when no disturbances were considered. The ocean current component that is parallel to the path does not affect the value for which the cross-track error converges, but only the total speed value. Thus, to evaluate the impact of the current on the capability of the guidance laws to make e converge to the origin, we introduce a current perpendicular to the path. The results obtained are shown in figure 3.10.



(a) LOS guidance law

(b) P. Maurya's guidance law

Figure 3.10: Vehicle's trajectory along a straight path in the presence of ocean currents

By looking at figure 3.10, we conclude that, in the presence of ocean currents, the use of LOS guidance law does not guarantee convergence of the vehicle to a straight line. Consequently, for any other path type, convergence is also not guaranteed. The LOS guidance law computes the heading reference using trigonometry concepts. However, when the vehicle's total velocity is different from the vehicle's relative velocity in relation to the water, there exists a misalignment between the velocity vectors, that is not considered by the principle, causing the cross-track error to converge to a non-zero value. The P. Maurya's guidance law computes the heading reference using a proportional and integral term. The convergence of the vehicle to a straight line, in the presence of ocean currents, is guaranteed by action of the integral term in (3.22). Thus, for a straight line, this principle does not require any compensation to guarantee convergence of the vehicle to the path. For a non-zero curvature path, in the presence of ocean currents the convergence of e to the origin is not guaranteed, as the orientation of the total velocity vector that ensures the convergence to the path varies along the course.

Considering the misalignment between the total velocity vector and the relative velocity vector we may design a guidance law to ensure convergence of the vehicle to the path. A representation of the vectors misaligment can be seen in figure 3.11.

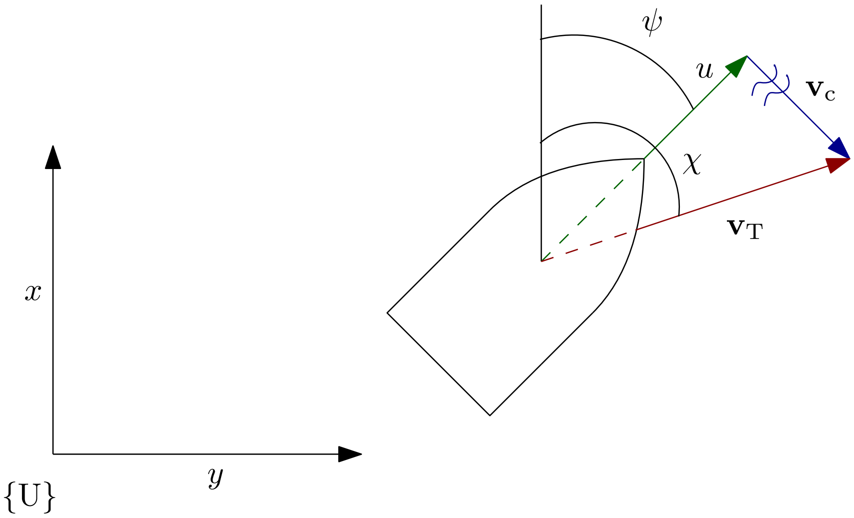


Figure 3.11: Velocity vectors considering ocean currents

As shown in figure 3.11, the total velocity vector $\mathbf{v}_\mathrm{T}$ is misaligned with the vehicle's relative velocity $\mathbf{v}_r$. Due to the vehicle's nature and the conditions used during this work we can once again disregard the sideslip, meaning that the vehicle's relative velocity vector $\mathbf{v}_r$ is equal to u. We define $\mathbf{v}_\mathrm{T}$ as

$$\mathbf{v}_\mathrm{T} = \mathrm{R}(\psi)\begin{bmatrix} u \\ 0 \end{bmatrix} + \mathbf{v}_\mathrm{c}, \tag{3.26}$$

where $\mathbf{v}_\mathrm{c}$ denotes the ocean current velocity. The previous equation can be decomposed in the x and y coordinates to obtain

$$\begin{bmatrix} \mathrm{v}_{\mathrm{T}_x} \\ \mathrm{v}_{\mathrm{T}_y} \end{bmatrix} = \begin{bmatrix} \cos(\psi) \\ \sin(\psi) \end{bmatrix} u + \begin{bmatrix} \mathrm{v}_{\mathrm{c}_x} \\ \mathrm{v}_{\mathrm{c}_y} \end{bmatrix}. \tag{3.27}$$

A proper way to approach the problem is to perform the mathematical computations in a reference frame attached to vector $\mathbf{v}_\mathrm{T}$. To do this, we start by performing a coordinate transformation from the inertial to the desired frame. The angle between the inertial reference frame and a reference frame attached to the total velocity vector corresponds to the desired course angle χ_d, we perform the transformation by multiplying both sides of the equation by $\mathrm{R}^T(\chi_d)$, and after some algebra we obtain

$$\begin{bmatrix} \frac{\mathrm{v}_{\mathrm{T}_x}}{\cos(\chi)} \\ 0 \end{bmatrix} = \begin{bmatrix} \cos(\chi_d - \psi) \\ \sin(\chi_d - \psi) \end{bmatrix} u + \begin{bmatrix} \mathrm{v}_\mathrm{c}^{\perp} \\ \mathrm{v}_\mathrm{c}^{\parallel} \end{bmatrix}. \tag{3.28}$$

The ocean current velocity is now decomposed into two components $\mathrm{v}_\mathrm{c}^{\parallel}$, representing the parallel component to the path and, $\mathrm{v}_\mathrm{c}^{\perp}$, representing the perpendicular component to the path. Using the second equation from (3.28) we compute the desired heading angle as

$$\psi_d = -\sin^{-1}\left(-\frac{\mathrm{v}_c^{\perp}}{u}\right) + \chi_d. \tag{3.29}$$

In a practical situation, the value of $\mathrm{v}_c^{\perp}$ is not known in advance but can be estimated using the ocean current estimate $\hat{\mathbf{v}}_c$. The ocean current component perpendicular to the path is determined as $\mathrm{v}_c^{\perp} = -\sin(\chi)\hat{\mathrm{v}}_{c_y} + \cos(\chi)\hat{\mathrm{v}}_{c_x}$. Due to the domain of the function $\sin^{-1}(\cdot)$ the perpendicular component of the ocean current needs to be smaller than the surge speed.

Using the values provided by the complementary filter in the control law from (3.29) and considering a non-null ocean current the vehicle's convergence to the path is guaranteed as represented in figure 3.12. In the figure it is possible to notice how the vehicle's orientation is automatically computed so that its total inertial velocity is aligned with the path.

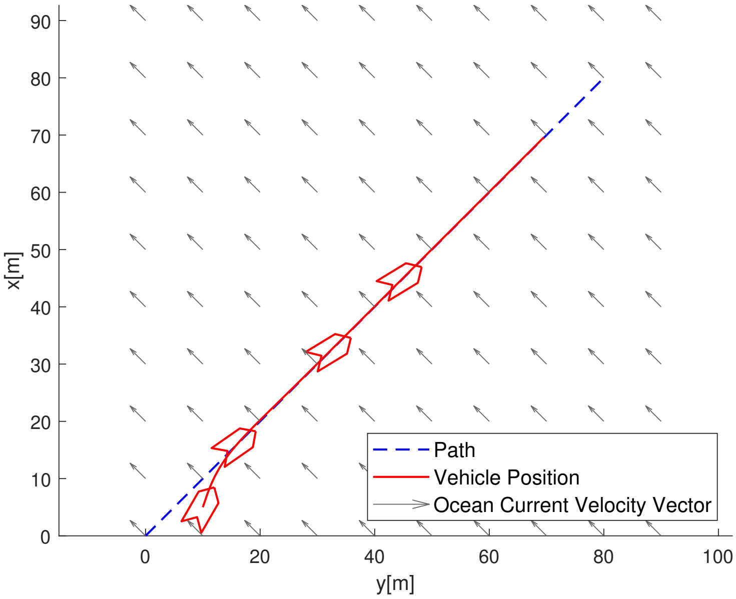
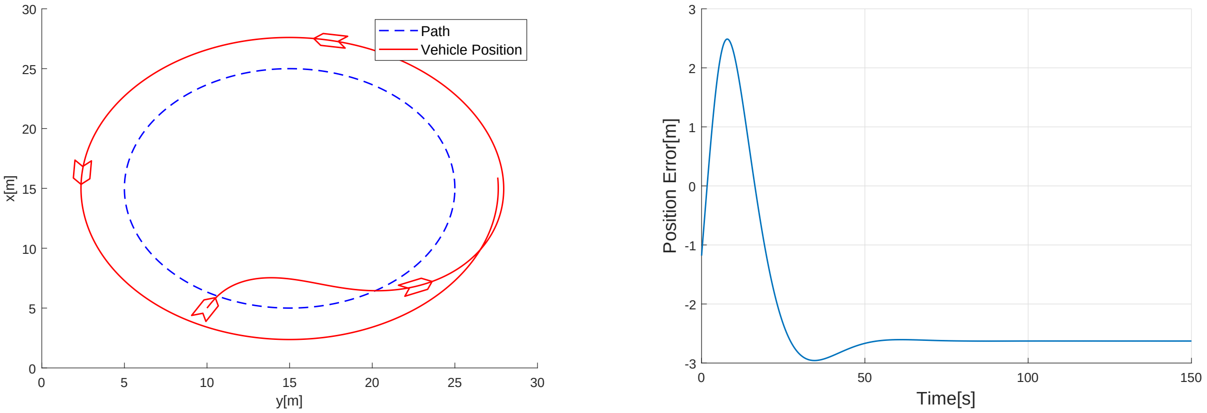


Figure 3.12: Vehicle's trajectory using a guidance law with ocean current compensation

3.5 Guidance Laws for Constant Curvature Paths

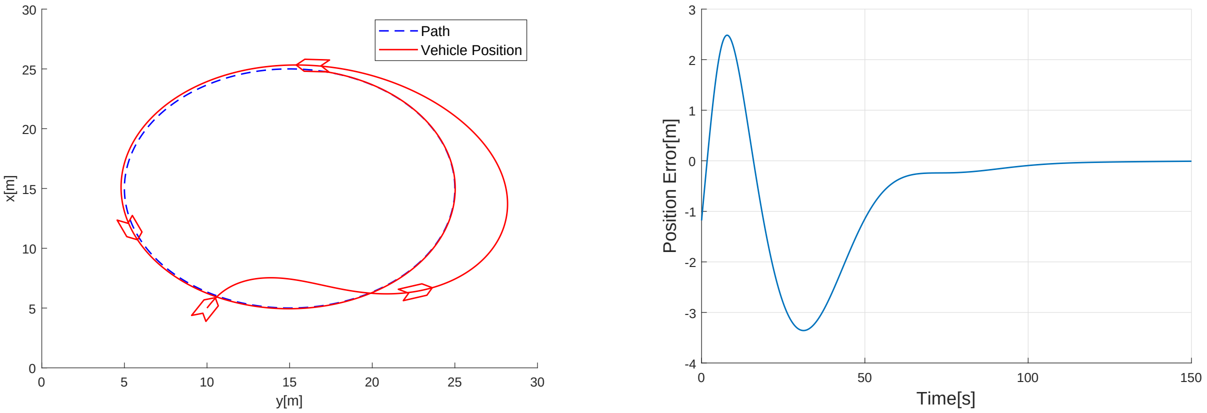
For practical purposes, paths are often defined as concatenations of segments of straight lines and circumferences. The guidance laws previously presented were designed for straight lines. Thus, it is necessary to observe how the vehicles behave when navigating along a curved path when using each of the guidance laws defined before. We assume constant-curvature curves, such as a circular path, and analyze the vehicle's approach to the path. For the case of straight lines, the path itself was used to generate heading references. For curved paths, studied over this section, we use the tangent line to the vehicle's projection on the path to generate the heading references. Given a curved path, the tangent line to any point in the path can be seen as a straight line. Considering the instantaneous tangent lien to the vehicle's projection we apply the guidance laws from (3.15) and (3.22) to generate the heading reference angle. The results can be seen in figures 3.13 and 3.14.



(a) Vehicle's trajectory

(b) Cross-track Error

Figure 3.13: Simulation results for a circular path using a LOS guidance law



(a) Vehicle's trajectory

(b) Cross-track Error

Figure 3.14: Simulation results for a circular path using P. Maurya's guidance law

In figure 3.13a, it is possible to see that the vehicle converges to a circular motion as expected. However, the cross-track error, represented in 3.13b, does not converge to the origin as the radius of the path and the vehicle's circular trajectory are different. For constant-curvature paths, the LOS guidance law introduces a bias in the cross-track error that depends on the vehicle's speed and the curvature radius. The introduction of an integral term on the cross-track error would solve this problem. In figure 3.14b it is possible to see the convergence of the cross-track error to the origin over the simulation. In the simulation considered, the P. Maurya's guidance law guaranteed the convergence of the cross-track error to the origin. However, for curved paths with higher curvatures (lower circumference radii), the convergence of the cross-track error to the origin is not guaranteed.

CHAPTER 4

Multiple Vehicle Motion Control

IN chapter 3, the motion control of a single vehicle was studied, and the PF algorithms were analyzed. Throughout this chapter, we will examine both the concepts and architecture of multiple vehicle cooperative motion control. The goal is to design a controller that drives a number of vehicles to a desired spatial formation. For this purpose, we introduce, in a outer loop, a coordination controller that adjusts the vehicle's speeds so as to guarantee that its positions converge to the desired formation. To control the motion of each vehicle, the control systems previously analyzed will be used. The previous results led to the choice of P. Maurya's guidance law to steer the vehicles along their paths. The control architecture is represented in figure 4.1. The nomenclature represented in the figure will be explained in the chapter.

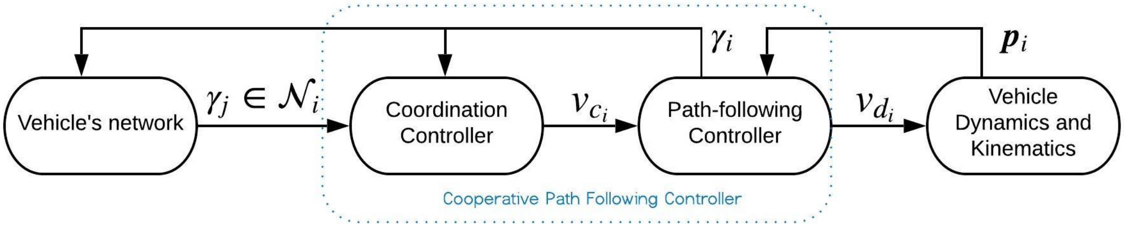

Vehicle's network
$\gamma_j \in \mathcal{N}_i$
Coordination Controller
v_{c_i}
Path-following Controller
v_{d_i}
Vehicle Dynamics and Kinematics
γ_i
$\boldsymbol{p}_i$
Cooperative Path Following Controller

Figure 4.1: Multiple vehicle motion: CPF system architecture

4.1 Concepts of Multiple Vehicle Motion Control

To tackle the problem of multiple vehicle motion control, we divide it into two parts. In the first part, a PF controller runs independently on each vehicle. In the second part, we adjust the motion of each vehicle to achieve synchronization between them. The exposition that follows draws form the work by R. Ghabcheloo et al. [13].

The key to achieve synchronization is to correctly parametrize each path, so that if the synchronization parameters of the vehicles achieve consensus, then the vehicles are in the desired formation. Considering this, we start with the analysis of a simple case considering a formation with two vehicles required to maneuver side-by-side along two parallel lines.

In this scenario, the synchronization parameter γ used is the distance traveled in the path by the projection of each vehicle, i.e., $\gamma_i = S_i$, where γ is the synchronization parameter, S is the distance traveled by the vehicle's projection and the subscript i denotes an arbitrary vehicle in the formation. This notation is used in the sequel. Figure 4.2 is a graphic representation of this scenario.

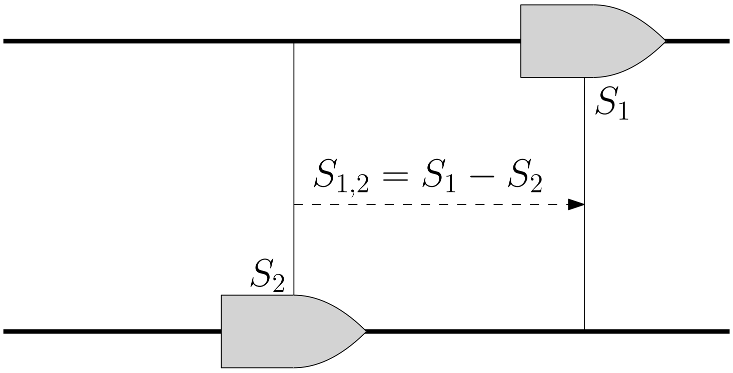


Figure 4.2: Straight line parametrization

The coordination error $\gamma_{1,2}$ is defined as the difference between the synchronization parameters, that is

$$\gamma_{1,2} = \gamma_1 - \gamma_2. \tag{4.1}$$

We can represent the vehicle's speed using this parameter. Notice for example for $i = 1$ that

$$\frac{\partial \gamma_1}{\partial t} = v_r, \tag{4.2}$$

where v_r represents the desired speed for the formation. Once we achieve coordination between vehicles, meaning $\gamma_{1,2} = 0$, we can make the speed of vehicle 2 equal to v_r, so that vehicle 2 ”keeps up” with vehicle 1.

So far, we analyzed straight paths aligned with each other. However, we may find ourselves in a situation where the paths are not aligned but have an angle α between them. Since we know the angle, we can relate the along-path lengths of each vehicle, as shown in figure 4.3. For clarity of exposition, only two vehicles are considered and represented in the following figure. The objective is for the two vehicles to maneuver along the two paths γ_1 and γ_2 forming the angle α between them so that the line joining them makes the same angle with the horizontal.

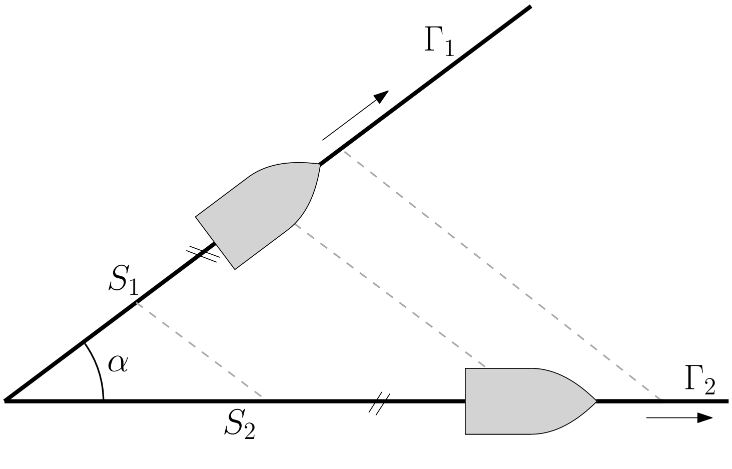


Figure 4.3: Straight lines with parametrization misalginment

By resorting to basic trigonometry knowledge, we can write the relation between S_1 and S_2 as

$$S_2 = 2S_1 \cos(\alpha). \tag{4.3}$$

By defining γ_1 and γ_2 as $\gamma_1 = S_1$ and $\gamma_2 = \frac{1}{2\cos(\alpha)}S_2$, we can describe the coordination error as $\gamma_{1,2} = \gamma_1 - \gamma_2$. The objective is to drive $\gamma_{1,2}$ to zero. Once synchronization is achieved,

$$\frac{\partial \gamma_1}{\partial t} = \frac{\partial \gamma_2}{\partial t} = v_r. \tag{4.4}$$

Taking into account equation (4.4) we can now determine the vehicle's speed in the inertial coordinates as

$$\frac{\partial \gamma_i}{\partial t} = \frac{\partial \gamma_i}{\partial S_i}\frac{\partial S_i}{\partial t} = v_r. \tag{4.5}$$

When applied to this case, we obtain the inertial speeds $\frac{\partial S_1}{\partial t} = v_r$ and $\frac{\partial S_2}{\partial t} = 2\cos(\alpha)v_r$. From the previous results we conclude that for non-aligned straight lines, the normalized speed (in γ coordinates) is the same for all vehicles. However, it must be stressed that the inertial speeds are not the same for all vehicles.

Until now we have been parametrizing straight lines. However, the path can be composed of straight lines and segments of circumferences, creating the need to parameterize curved segments as well. Contrary to what was done for straight lines, the along-path length cannot be used as a parameter. An example is a scenario containing two circular paths with different radii, where the vehicles are required to maintain the radial alignment. Since the two circumferences have different radii, the perimeters will also differ. This means that when the vehicle are synchronized they will travel different distances, thus the need to define a different synchronization parameter. In figure 4.4, we can see a representation of this scenario.

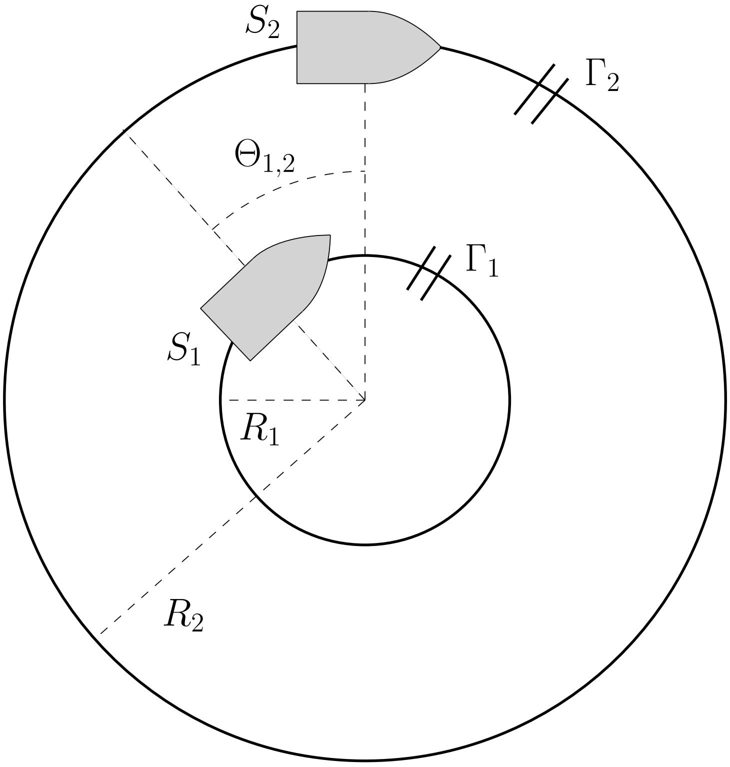


Figure 4.4: Circular path parametrization

The goal is to drive $\theta_{1,2}$ to zero, where $\theta_{1,2}$ is the angular ”distance” between the vehicles. We can do an equivalent process by normalizing the length of the circumferences to 2π. To do so, we divide the along path length by the circumference radius, R to obtain

$$\gamma_i = \frac{S_i}{R_i}, \tag{4.6}$$

where S denotes the arc length given by

$$S = R \cdot \theta, \tag{4.7}$$

where θ is the central angle of the arc expressed in radians.

Thus, we get a good measure of the coordination error by defining $\gamma_{1,2} = \gamma_1 - \gamma_2$. By making $\partial\gamma_1/\partial t = \partial\gamma_2/\partial t = v_r$ we obtain the same speed profile in all γ_i coordinates. In fact, recalling the derivative of composition functions yields

$$\frac{\partial\gamma_i}{\partial t} = \frac{\partial\gamma_i}{\partial S_i}\frac{\partial S_i}{\partial t} = v_r. \tag{4.8}$$

Thus, we conclude that the inertial speeds scale naturally and proportionally as

$$\frac{\partial S_i}{\partial t} = R_i v_r, \tag{4.9}$$

as desired.

4.2 Two Vehicle Coordination Controller

Considering the different parametrizations presented above, the design of a controller that ensures the vehicle's synchronization remains. We use a proportional controller applying an input that is directly proportional to the difference between the synchronization parameters of each vehicle. The fleet shall navigate synchronously along the path with a determined velocity, v_r. To this end, the synchronization law given by

$$\begin{aligned} v_1 &= v_r - K_c \cdot (\gamma_1 - \gamma_2) \\ v_2 &= v_r + K_c \cdot (\gamma_1 - \gamma_2), \end{aligned} \tag{4.10}$$

where $K_c > 0$ is a proportional gain is used. The proof that the use of this controller ensures the convergence of the vehicles to the desired formation can be found by writing the differential equation for the coordination error

$$\frac{\partial\gamma_{1,2}(t)}{\partial t} = \frac{\partial\big(\gamma_1(t) - \gamma_2(t)\big)}{\partial t}. \tag{4.11}$$

The variation of the synchronization parameter is equal to the vehicles' speeds. Thus,

$$\begin{aligned} \frac{\partial\gamma_1}{\partial t} &= v_1 \\ \frac{\partial\gamma_2}{\partial t} &= v_2, \end{aligned} \tag{4.12}$$

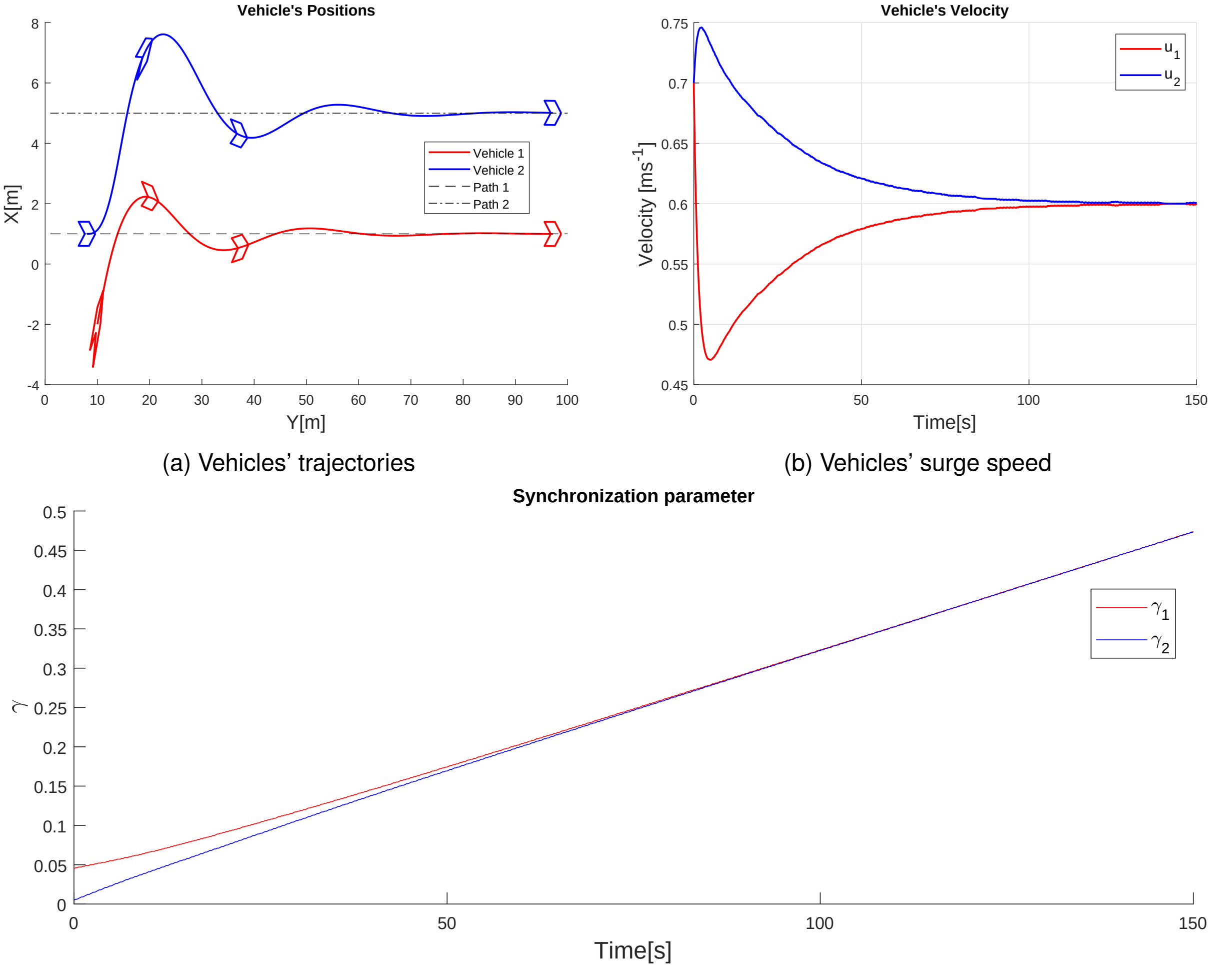
where v_1 and v_2 is given by (4.10). By combining equations (4.10), (4.11) and (4.12), and after some algebra we obtain

$$\frac{\partial e(t)}{\partial t} = -2K_c\big(\gamma_1(t) - \gamma_2(t)\big) = -2K_c e(t). \tag{4.13}$$

Analyzing equation (4.13) we conclude that with a positive gain, $K_c > 0$, the control law ensures that the error converges to the origin exponentially.

The vehicles cannot have negative velocities. For this reason, a saturation term is added to ensure that the speeds will reach a range of possible physical values. This change will not imply significant differences in the overall synchronization controller. The difference is that the correction term $K_c(\gamma_1 - \gamma_2)$ is now bounded, meaning that the rhythm at which the vehicles will synchronize is also bounded.

After studying the concepts used to synchronize vehicles, simulations were conducted illustrating three scenarios and two path types. An important remark to be made is that throughout the simulations, the PF controller was considered, composed by the P. Maurya's guidance law together with the heading controller. In the first simulation, we considered two parallel paths aligned in the x coordinates. As mentioned above, for this scenario, we used the along-path length as the synchronization parameter. However, this parameter can reach large values, so we normalize it. Thus, the coordination parameters are now bounded to the interval $[0, 1]$. This change does not affect the results and is only made to allow a better analysis of the results, represented in figure 4.5.



(a) Vehicles' trajectories

(b) Vehicles' surge speed

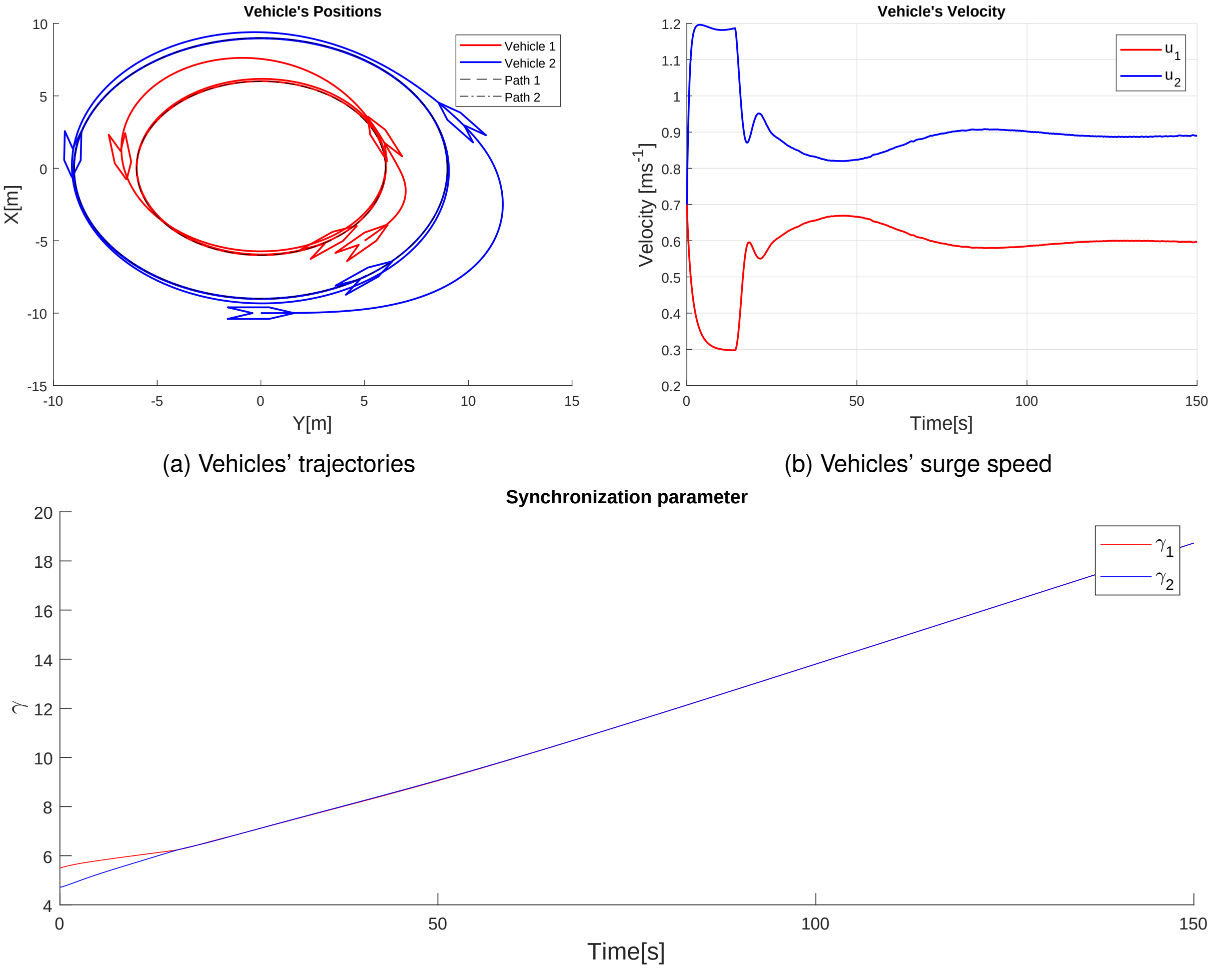
(c) Synchronization Parameters

Figure 4.5: Simulation results for parallel straight paths using a synchronization controller

Figure 4.5a shows the vehicle's trajectory. It is possible to see the vehicles misalignment at the beginning and the coordination achieved at the end of the simulation, where the vehicles are navigating side-by-side with the same y coordinate. To achieve synchronization, the vehicles' velocities are adjusted over the maneuver, as seen in figure 4.5b.

The vehicle represented in line has its velocity increased, simultaneously the vehicle represented by a red line has its velocity decreased. The correction term gets smaller as the synchronization is achieved. The vehicles' velocities converge to the steady-state normalized velocity that, in this case, is the same for both vehicles. In figure 4.5c we can see the synchronization parameters converging. The rate of convergence can be changed by setting different values for the saturation function. In this simulation, the saturation function was set to ± 0.3.

In the second simulation, the paths were concentric circumferences. The radii of the circumferences were set to $6\mathrm{m}$ and $9\mathrm{m}$. The results are presented in 4.6.



(a) Vehicles' trajectories

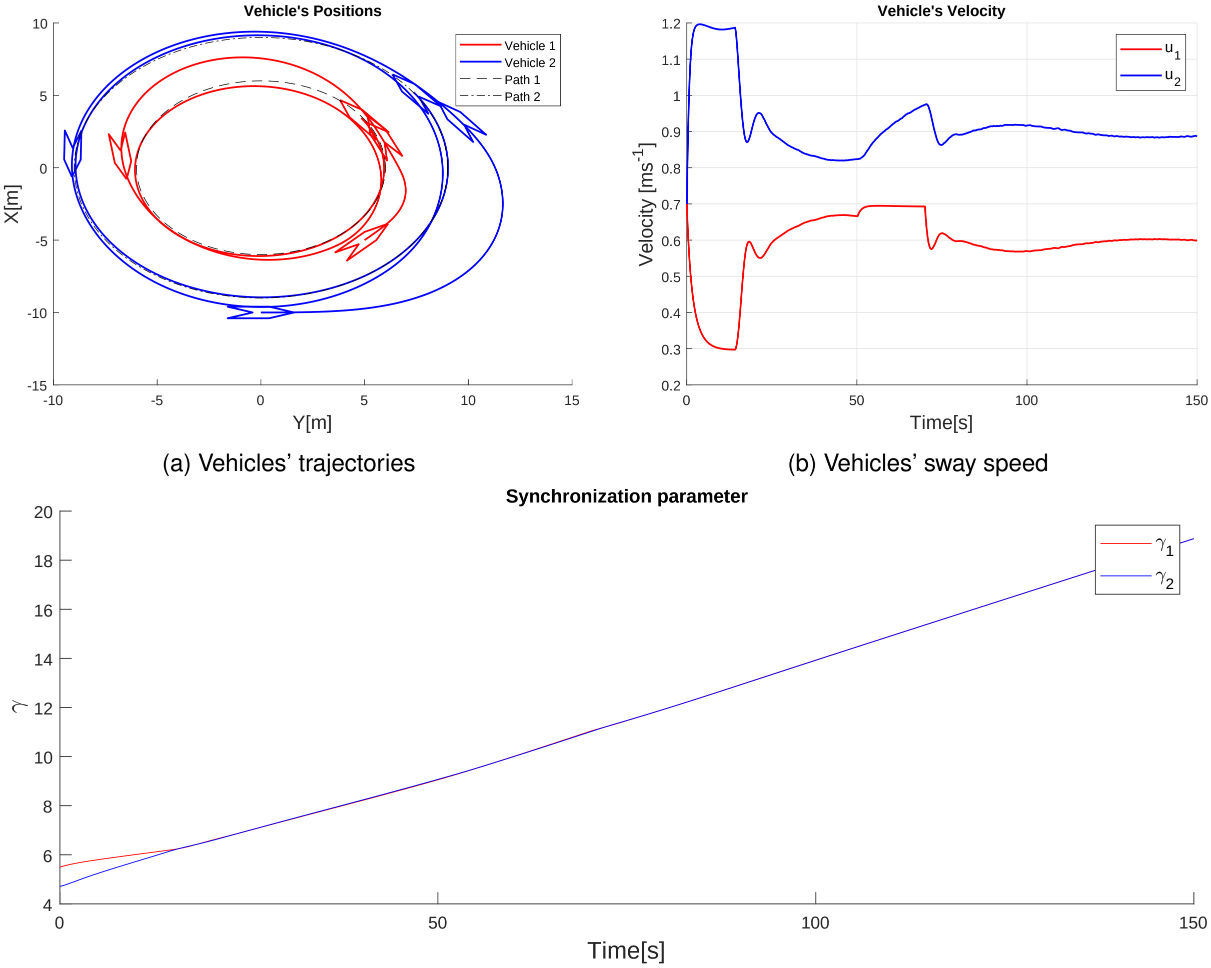
(b) Vehicles' surge speed

(c) Synchronization Parameters

Figure 4.6: Simulation results for concentric circular paths using a synchronization controller

In figure 4.6a, we can observe the convergence of the vehicles to the paths, as expected. After some time has elapsed in the simulation, the vehicles are aligned radially. As in the previous scenario, the velocities, represented in 4.6b, were adjusted automatically to achieve synchronization. As expected, the speeds converge to their steady-state values. However, these values are different for the two vehicles and are proportional to the path radii. Thus, the circumference radius must not be too different as the outer vehicle may not be capable of reaching the steady-state velocity necessary to be synchronous. The synchronization parameters represented in 4.6c converge and, without disturbances, remain throughout the rest of the simulation.

To observe the performance of the controller in the presence of disturbances, we conducted a simulation with an identical scenario to 4.6. One of the vehicles suffered a malfunction in the time interval $[50, 70]\mathrm{s}$ where its speed was artificially set to $0.7\,\mathrm{m\,s^{-1}}$, a value different from the steady-state value. The results are plotted in figure 4.7.



(a) Vehicles' trajectories

(b) Vehicles' sway speed

(c) Synchronization Parameters

Figure 4.7: Simulation results for concentric circular paths using a synchronization controller in the presence of vehicle malfunctioning

The main difference between the results from 4.6 and 4.7 lies on the vehicle's speeds. In 4.7b, it is possible to notice the malfunction in vehicle 1. Its speed is constant during the malfunction time interval. The synchronization was achieved before the malfunction and, to compensate it, vehicle 2 has its velocity increased. Once the malfunction ceases the speeds converge to their steady state-values. Since the difference between the speed during the malfunction and the steady-state speed is not remarkable, the slope of the synchronization parameter evolution, represented in 4.7c, does not vary much.

If the formation is not characterized by having the vehicles navigating side by side the path parametrization is adjusted and an offset is added to ensure that when the synchronization parameters difference is null, the vehicles are in the desired formation. If the path parameters are not adjusted, this adjustment can be made in the controller, by subtracting or adding a constant to the difference of the parameters.

4.3 Communication Theory and Inner-Vehicle Communication Representation

Throughout this section, we will extend the communication scheme for scenarios with N vehicles involved. We use a graph to represent the inter-vehicle communication network, where the vehicles are represented as nodes and the communication links are represented as edges. Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ represent an undirected graph with the node set $\mathcal{V} = \{1, 2, ..., N\}$ and with the edge set $\mathcal{E} = \{\{(v_i, v_j)\} \in \mathcal{V} \times \mathcal{V}\}$ where the edge (v_i, v_j) exists only if vehicle i communicates with vehicle j . See [2] for detailed information about graph theory.

We define a square matrix, $\mathrm{A} = (a_{ij} \in \mathbb{R}^{N \times N})$, whose elements, indicating whether two vehicles, represented by a pair of edges, are adjacent or not, are defined as

$$a_{ij} = \begin{cases} 1 & \text{if } (v_i, v_j) \in \mathcal{E} \\ 0 & otherwise \end{cases}. \tag{4.14}$$

By resorting to graph theory, it is possible to define the degree matrix D for the graph $\mathcal{G}$ as

$$d_{ij} = \begin{cases} \deg(v_i) & \text{if } i = j \\ 0 & otherwise \end{cases}, \tag{4.15}$$

where $\deg(v_i)$ denotes the number of vehicles communicating with vehicle i, represented in the graph by the number of occurrences of edges terminating at node v_i.

From 4.14 and 4.15 we can define the Laplacian matrix for the graph $\mathcal{G}$ as

$$\mathrm{L} = \mathrm{D} - \mathrm{A}, \tag{4.16}$$

where its elements are defined as

$$\mathrm{L}_{ij} = \begin{cases} \deg(v_i) & \text{if } i = j \\ -1 & \text{if } i \neq j \text{ and } v_i \text{ is adjacent to } v_j \\ 0 & \text{otherwise.} \end{cases}. \tag{4.17}$$

It is also possible to define the normalized Laplacian Matrix [7], as

$$\mathrm{L_D} = \mathrm{D}^{-1}(\mathrm{D} - \mathrm{A}). \tag{4.18}$$

The matrix $\mathrm{L_D}$ will be used later to design a general synchronization controller.

4.4 Coordination Controller Considering Continuous Communications

Considering the graph representation presented above, it remains to define a coordination controller that, in the presence of continuous communications, ensures the convergence of the vehicles to the desired formation. We adopt a decentralized control scheme, that for each vehicle i uses the coordination parameter of the vehicle, γ_i together with the coordination parameters of the vehicles in the neighborhood of the vehicle, γ_j, $\forall j \in \mathcal{N}_i$ to adjust the speed of vehicle i. We use the Laplacian normalized matrix $\mathrm{L_D}$ from (4.18) to define the coordination error as

$$\varsigma = \mathrm{L_D}\gamma \tag{4.19}$$

where ς is the coordination error vector and $\gamma = [\gamma_1, \cdots, \gamma_N]^T$ is the vector of coordination parameters. Each entry of vector ς is defined as

$$\varsigma_i = \gamma_i - \frac{1}{|\mathcal{N}_i|} \sum_{j \in \mathcal{N}_i} \gamma_j \tag{4.20}$$

As done in the scenario where two vehicles are involved, the key idea is to add a correction term to the reference speed for each vehicle. Thus, the speed reference provided to the speed law in (3.14) is equal to the sum of the speed reference for the vehicle with the speed correction term, that is,

$$v_d = v_r + v_c. \tag{4.21}$$

where v_d is the desired speed.

The correction term should be bounded to prevent the desired speed from reaching negative values as well as high values. Inspired by the work in F. Vanni [37] we introduce a coordination law defined by

$$v_c = -k_e \tanh\left(\mathrm{L_D}\gamma\right), \tag{4.22}$$

where $k_e > 0$ is the bound for the correction term. The speed correction term for each vehicle is given by

$$v_{c_i} = -k_e \tanh\left(\gamma_i - \frac{1}{|\mathcal{N}_i|} \sum_{j \in \mathcal{N}_i} \gamma_j\right). \tag{4.23}$$

The use of the hyperbolic tangent ensures the boundedness of the correction term. The proof that the speed v_d converges asymptotically to v_r is made in the mentioned work by assuming that the communications graph is connected. This, in turn, implies that the Laplacian $\mathrm{L_D}$ observes some conditions that are required in the proof of convergence. Full details are available in [13].

4.5 Coordination Controller Considering Discrete Communications

In the previous section, we designed a coordination controller that relies on the continuous exchange of data among the vehicles. When the vehicles are at the surface, the use of WiFi allows for almost continuous communications. However, in underwater missions, continuous communications are not possible. Thus the need to design a coordination controller that relies on discrete communications arises.

Given the path parametrizations presented in 4.1, the rate of the coordination parameter is given by

$$\dot{\gamma} = v_d(\gamma). \tag{4.24}$$

By applying the results from (4.21) in (4.24) we obtain a continuous time space model given by

$$\dot{\gamma} = v_r(\gamma) - k_e \tanh\left(\mathrm{L_D}\gamma\right). \tag{4.25}$$

Now, it remains to apply the finite difference equation to 4.25 to obtain an equation for the estimator of the coordination parameter, given by

$$\hat{\gamma}(k+1) = \hat{\gamma}(k) + \Big(v_r\big(\hat{\gamma}(k)\big) - k_e \tanh\big(\mathrm{L_D}\hat{\gamma}(k)\big)\Big)\Delta t \tag{4.26}$$

Each vehicle in the formation runs an estimator, defined in (4.26), to estimate the synchronization parameter of its neighborhood. Thus, the coordination error from (4.20) is rewritten as

$$\hat{\varsigma}_i = \gamma_i - \frac{1}{|\mathcal{N}_i|}\sum_{j\in\mathcal{N}_i}\hat{\gamma}_j. \tag{4.27}$$

In accordance, the correction term is thus defined as

$$v_c = -k_e \tanh\big(\gamma - \mathrm{D}^{-1}\mathrm{A}\hat{\gamma}\big). \tag{4.28}$$

The proof that this controller yields good performance can be found in [13].

CHAPTER 5

Cooperative Target Tracking

THIS chapter introduces a control structure, based on trajectory tracking, that allows a set of agents to perform an encircling maneuver around a moving target. The control structure is based on the work of L. Arranz [4] [3]. Convergence of the vehicles to the desired formation is achieved with the introduction of a potential function. In the context of this thesis, we study the situation where the path is circular. In the last publication by L. Arranz, a new framework was proposed that allows extending the results to more complex formations.

5.1 Problem Formulation

The adopted strategy does not require the knowledge of the target's absolute position, velocity and acceleration, but only the relative position of the vehicles in relation to the target, expressed in the body reference frame, $\{B\}$, of the respective agent.

We start by recalling the kinematic model, presented in (2.13), by assuming that the sway speed is zero. This assumption allows writing a simplified model for the vehicles, equal to the unicycle kinematics, given by

$$\begin{aligned} \dot{\mathbf{p}}_i &= \mathrm{R}(\psi_i)[u_i, 0]^T \\ \dot{\psi}_i &= r_i, \end{aligned} \tag{5.1}$$

where $\mathbf{p}_i \in \mathbb{R}^2$ is the position vector and ψ_i is the heading angle of vehicle i.

The control design uses a decentralized structure that allows each vehicle to make its independent decisions, contrarily to what happens in a centralized structure where a vehicle takes the decisions on behalf of all vehicles. The fact that the vehicles communicate through a specific network is taken into consideration by the control design, represented in figure 5.1. By examining figure 5.1, it is possible to see the control decentralization as each vehicle has its exosystem. The exosystem plays an important role in the control structure, as it is responsible for generating references for the relative positions, $\mathbf{p}_i^*$. For this purpose, the unicycle model is used to create a virtual vehicle inside the exosystem. A circular control law is then used to make the virtual vehicle converge to a circular motion around the origin. The synchronization is then tackled on two distinct steps: i) a potential function U is added to the circular control law, introducing a repulsion force between the virtual vehicles which makes the them converge to the uniform distribution over the circular trajectory around the origin, and ii) a controller guarantees that

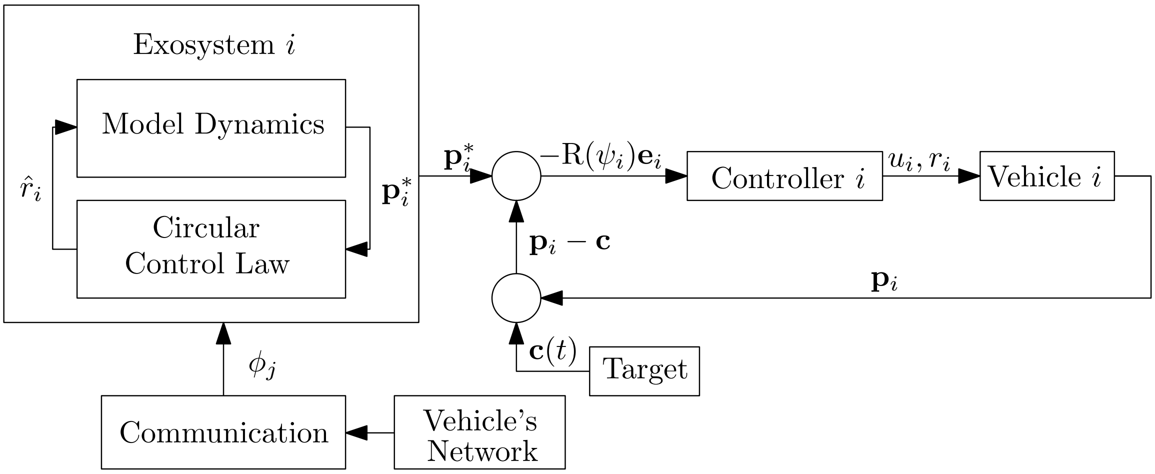


Figure 5.1: Control design using a decentralized exosystem

each vehicle is tracking the relative position generated by its exosystem. The communication between vehicles is performed at the exosystem level, as the potential function exchanges information about the virtual vehicles' heading angle. The communication network is considered as each exosystem has only access to the heading angle of a virtual vehicles if its real vehicles are linked by a communication channel.

The functioning of the control structure starts from the assumption that each vehicle is capable of measuring its relative position vector expressed in its reference frame, $\mathrm{R}(\psi_i)^T(\mathbf{p}_i - \mathbf{c})$. The error between the relative position vector $\mathbf{p}_i - \mathbf{c}$ and the reference $\mathbf{p}_i^*$, expressed in the body reference frame, can thus be defined as

$$\mathbf{e}_i = \mathrm{R}(\psi_i)^T\big((\mathbf{p}_i - \mathbf{c}) - \mathbf{p}_i^*\big). \tag{5.2}$$

The objective of the controller is to generate the control inputs, u_i and r_i, so that the error, given by (5.2), converges to zero. The unicycle dynamics are used to define the model dynamics of the exosystem, since it is a general model for an agent moving in a two-dimensional environment, considering that the vehicle has no lateral motion, that is sideslip. Considering the intended circular motion, the dynamics can be defined as

$$\begin{aligned} \dot{\mathbf{p}}_i^* &= \mathrm{R}(\phi_i)[|\omega_0|R,\ 0]^T \\ \dot{\phi}_i &= \hat{r}_i, \end{aligned} \tag{5.3}$$

where $\omega_0 \neq 0$ represents the desired angular velocity, R represents the radius intended for the circular motion, ϕ_i denotes the orientation of the velocity vector $\dot{\mathbf{p}}_i^*$, and $\hat{r}_i$ represents the control input.

While converging to the encircling maneuver, the vehicles should also converge to the desired formation. The formation corresponds to a uniform distribution, where the vehicles are equally spaced over the circumference. For this purpose following the exposition in [4], a potential function $U(\phi)$ is introduced in the circular control law that adds a repulsion force between vehicles. For the vehicles to stabilize at the uniform distribution, the distribution must be an equilibrium point of the potential function. Considering this potential function, the circular control law is defined as

$$\begin{aligned} \hat{r}_i &= \omega_0(1 + k\dot{\mathbf{p}}_i^{*T}\mathbf{p}_i^*) - \frac{\partial U}{\partial \phi_i} \\ \frac{\partial U}{\partial \phi_i} &= \frac{k_u}{|\mathcal{N}_i|}\sum_{j\in\mathcal{N}_i}\sum_{m=1}^{\lfloor\frac{\mathcal{N}_i}{2}\rfloor}\frac{\sin\left(m(\phi_i - \phi_j)\right)}{m}, \end{aligned} \tag{5.4}$$

where $\mathcal{N}_i$ denotes the neighborhood of vehicle i (the number of vehicles which vehicle i communicates with), $k, k_u > 0$ can be used as tune parameters, and $\lfloor \cdot \rfloor$ corresponds to the floor function, rounding its input to the largest integer that is smaller or equal to the input. The exosystem is now fully defined. It remains to define the controller that ensures convergence of $\mathbf{p}_i(t)$ to $\mathbf{p}_i^*(t) + \mathbf{c}(t)$. This convergence must be achieved without the use of the target's velocity and acceleration, $\dot{\mathbf{c}}(t)$ and $\ddot{\mathbf{c}}(t)$, respectively. The time derivative of the error defined in (5.2) is given by

$$\begin{aligned} \dot{\mathbf{e}}_i &= \mathrm{R}(\psi_i)^T \big(\dot{\mathbf{p}}_i - (\dot{\mathbf{p}}_i^* + \dot{\mathbf{c}})\big) - r_i \mathrm{R}_{\frac{\pi}{2}} \mathrm{R}(\psi_i)^T \big(\mathbf{p}_i - (\mathbf{p}_i^* + \mathbf{c})\big) \\ &= \mathrm{R}(\psi_i)^T \dot{\mathbf{p}}_i - \mathrm{R}(\psi_i)^T (\dot{\mathbf{p}}_i^* + \dot{\mathbf{c}}) - r_i \mathrm{R}_{\frac{\pi}{2}} \mathbf{e}_i \\ &= [u_i, 0]^T - \mathrm{R}(\psi_i)^T (\dot{\mathbf{p}}_i^* + \dot{\mathbf{c}}) - r_i \mathrm{R}_{\frac{\pi}{2}} \mathbf{e}_i, \end{aligned} \tag{5.5}$$

where $\mathrm{R}_{\frac{\pi}{2}}$ denotes the rotation matrix through an angle $\frac{\pi}{2}$. We now introduce the vector $\boldsymbol{\delta} = [-\delta, 0]^T$ where δ is an arbitrary small positive constant to obtain

$$\begin{aligned} \dot{\mathbf{e}}_i &= [u_i, 0]^T - \mathrm{R}(\psi_i)^T (\dot{\mathbf{p}}_i^* + \dot{\mathbf{c}}) - r_i \mathrm{R}_{\frac{\pi}{2}} (\mathbf{e}_i - \boldsymbol{\delta}) - r_i \mathrm{R}_{\frac{\pi}{2}} \boldsymbol{\delta} \\ &= [u_i, 0]^T - \mathrm{R}(\psi_i)^T (\dot{\mathbf{p}}_i^* + \dot{\mathbf{c}}) - r_i \mathrm{R}_{\frac{\pi}{2}} (\mathbf{e}_i - \boldsymbol{\delta}) - [0, \delta r_i]^T \\ &= \Delta [u_i, r_i]^T - \mathrm{R}(\psi_i)^T (\dot{\mathbf{p}}_i^* + \dot{\mathbf{c}}) - r_i \mathrm{R}_{\frac{\pi}{2}} (\mathbf{e}_i - \boldsymbol{\delta}), \end{aligned} \tag{5.6}$$

where $\Delta = \begin{bmatrix} 1 & 0 \\ 0 & \delta \end{bmatrix}$. The new error vector $(\mathbf{e}_i - \boldsymbol{\delta})$ is the distance between a point located at a distance δ from the mass center of agent i along the x-axis of the local reference frame and the desired position $(\mathbf{p}_i^* + \mathbf{c})$. The target velocity is assumed unknown. Thus, $\dot{\mathbf{c}}(t)$ can be assumed as an external disturbance. For this reason, the convergence of the error to the origin is not guaranteed. However, if $\dot{\mathbf{c}}(t)$ is bounded, i.e.,

$$\|\dot{\mathbf{c}}(t)\| \leq \gamma_c, \ \forall t, \tag{5.7}$$

for $\gamma_c \geq 0$, then it is possible to prove the stability of the tracking target control using ε-stability [4].

The control law is defined as

$$[u_i, r_i]^T = \Delta^{-1} \Big(\mathrm{R}(\psi_i)^T \dot{\mathbf{p}}_i^* - K(\mathbf{e}_i - \boldsymbol{\delta}) \Big), \tag{5.8}$$

where K is a positive-definite symmetric matrix, and u_i, r_i denote the linear velocity in the surge and angular velocity, respectively. The vehicles converge to a circular motion of radius R whose rotation direction is determined by the sign of ω_0 and whose center must converge to a circular region centered at the target position $\mathbf{c}(t)$ with radius

$$\varepsilon_c = \gamma_c / \lambda_{\min}(K), \tag{5.9}$$

where $\lambda_{\min}(K)$ denotes the minimum eigenvalue of matrix K. This result can be further expanded to show that the distance between each vehicle and the moving target will converge to the set

$$\mathcal{R}_c = [R_0 - \delta - \varepsilon_c, R_0 + \delta + \varepsilon_c]. \tag{5.10}$$

A proof of this result can be found in [4].

It is important to note that the angular velocity, r_i, depends on the term $1/\delta$, which comes from Δ^{-1} in equation (5.8). This relationship implies that decreasing the cross-track error (decreasing δ) implies an increase in the bound of the control input r_i. On the other hand, the control parameter K also impacts the position error as well as the control inputs bound. There is a proportionally inverse relation between the eigenvalues of K and the position error. For a higher value of the minimum eigenvalue of K the radius of the ε-stability decreases, and the center of the circular motion for which the vehicles converge is closer to the target position. However, for higher values of K the bounds for the control inputs increase. Thus, there exists a trade-off between the convergence region of the circular motion center and the bound for $[u_i, r_i]$.

5.2 Simulation Results

For a better understanding of this trade-off, and to help tune the values of K and δ, several simulations were conducted. The results obtained are presented and analyzed next.

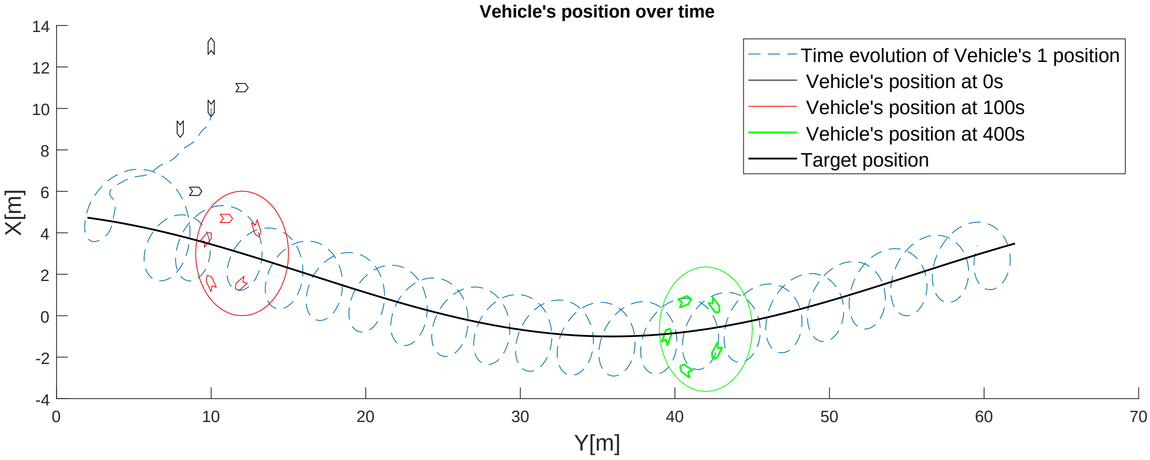


Figure 5.2: Circular motion around a moving path - $\delta = 0.2$m, $K = 2I_2$

Before analyzing the results, it is essential to note that the vehicle model, (5.1) is a simplification of the ”real” model. The sideslip being disregarded does not affect the results significantly. However, the delay existent in the vehicle, as the velocities are not achieved instantaneously, is not considered. In the same line of thought, the change of orientation is also not immediate. This phenomenon does not affect the primary behavior of the vehicle, but we may see some small oscillations that the use of this controller causes in the vehicles. We can now analyze the results from figure 5.2. Represented in green and red are the positions of the vehicles and the desired encircling trajectory at different time instants. Represented by a dashed blue line is the trajectory of one of the vehicles. The difference between the radius of the desired trajectory and the trajectory performed by the vehicles is evident, together with the displacement of the trajectories center. This finding is justified by the eigenvalues of K that are not high enough. To evaluate the performance of the vehicles for higher eigenvalues of K, a simulation is carried out in the same conditions, but with an increase in the value of K to $8I_2$. The results are represented in figure 5.3.

Comparing the results from 5.2 and 5.3 it is possible to see a similar shape in the dashed blue line representing the trajectory of one of the vehicles. The main differences lie in the radius of the trajectory performed by the vehicles, that is much closer to the desired one for $K = 8I_2$, and in the distance between the centers of the desired and the real trajectory, which is considerably smaller for the same situation. The vehicles' trajectories also assume a narrower spring effect, since the vehicles' velocities increase with the increase in K. However, as mentioned, the increase in K implies a higher bound for the input parameters, as seen in figure 5.4.

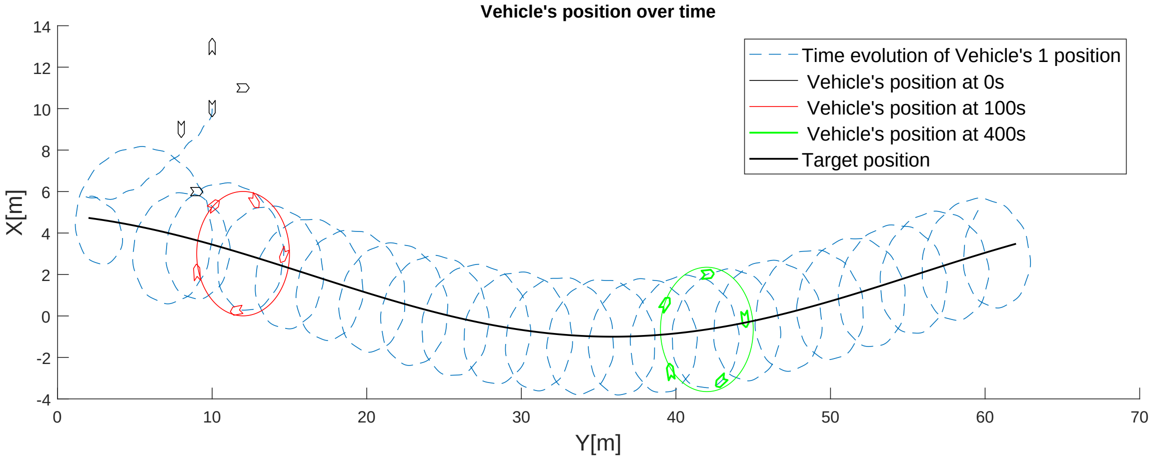
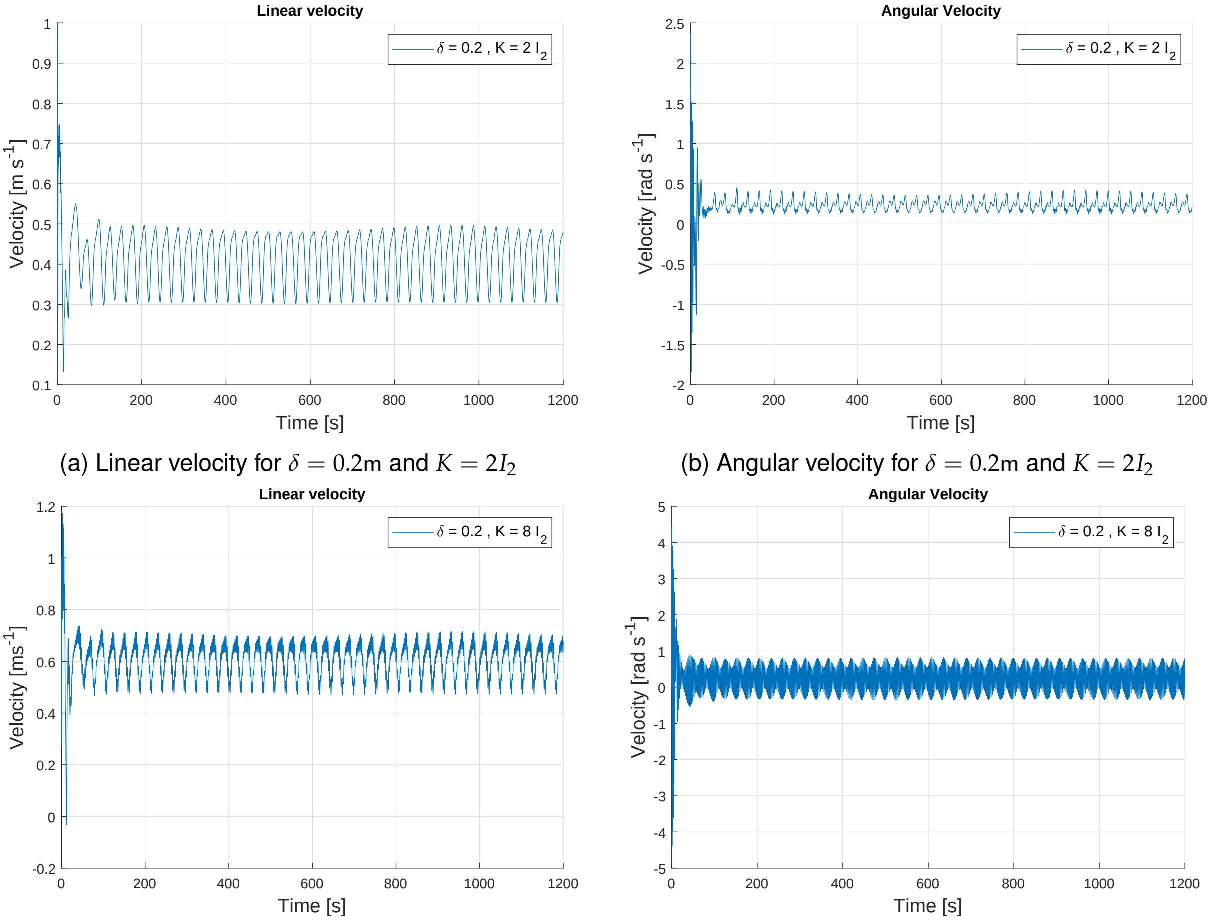


Figure 5.3: Circular motion around a moving path - $\delta = 0.2$m, $K = 8I_2$



(a) Linear velocity for $\delta = 0.2$m and $K = 2I_2$

(b) Angular velocity for $\delta = 0.2$m and $K = 2I_2$

(c) Linear velocity for $\delta = 0.2$m and $K = 8I_2$

(d) Angular velocity for $\delta = 0.2$m and $K = 8I_2$

Figure 5.4: Sway velocity, u, and yaw rate, $\dot{\psi}$, for different values of K

The increase in the value of K causes an increase in the mean linear velocity of the vehicles, as seen in 5.4a and 5.4c. These results are expected since the increase in the value of K has a direct relation with its eigenvalues. It is also possible to see a small secondary oscillating behavior in the linear velocity caused by the increase in the value of K. This oscillation is caused by higher variations of the control inputs which the vehicle is not capable of accomplishing. In the same line of thought the angular velocity, represented in 5.4b and 5.4d, has a higher oscillation frequency related to the higher value of K. Indefinitely increasing the values of K causes the vehicles to operate outside the region where the obtained model and the simplifications made are valid, causing the vehicles to have unpredictable behavior and not performing the encircling maneuver.

5.3 Robustness to External Disturbances

The control design, presented above, stabilizes the vehicles on a circular trajectory whose center converges to a region around the moving target. The fact that the target velocity is not required implies that the position error will not converge to the origin. However, if $\dot{\mathbf{c}}$ is bounded, then, the error will converge to a region around the origin. The previous results are valid when there are no external disturbances. We now introduce disturbances and analyze the control design results and the vehicles behavior. With the introduction of disturbances, the model from (5.1) gives rise to

$$\begin{aligned} \dot{\mathbf{p}}_i &= \mathrm{R}(\psi_i)[u_i, 0]^T + \mathbf{w}_i \\ \dot{\theta}_i &= r_i \end{aligned}, \tag{5.11}$$

where $\mathbf{w}_i$ is a vector representing the disturbances, for which there exists γ_w so that $\mathbf{w}_i$ is bounded, i.e.,

$$\|\mathbf{w}_i\| \leq \gamma_w, \ \forall_i. \tag{5.12}$$

The disturbance vector $\mathbf{w}_i$ can be used to represent physical disturbances that exist in the scenario where the vehicles are used, such as the ocean currents.

We start by rewriting the error dynamics from (5.5). The introduction of the disturbance vector $\mathbf{w}_i$ in (5.11) causes the term $\dot{\mathbf{p}}_i$ from (5.5) to become $\dot{\mathbf{p}}_i + \mathbf{w}_i$. Thus, the error dynamics are defined as

$$\dot{\mathbf{e}}_i = [u_i, 0]^T - \mathrm{R}(\psi_i)^T(\dot{\mathbf{p}}_i^* + \dot{\mathbf{c}} - \mathbf{w}_i) - r_i \mathrm{R}_{\frac{\pi}{2}} \mathbf{e}_i. \tag{5.13}$$

As done in (5.6) we introduce the vector $\boldsymbol{\delta} = [\delta, 0]^T$. The error dynamics from (5.13) are rewritten as

$$\dot{\mathbf{e}}_i = \Delta[u_i, r_i]^T - \mathrm{R}(\psi_i)^T(\dot{\mathbf{p}}_i^* + \dot{\mathbf{c}} - \mathbf{w}_i) - r_i \mathrm{R}_{\frac{\pi}{2}}(\mathbf{e}_i - \boldsymbol{\delta}) \tag{5.14}$$

It is evident that if both the target velocity and the disturbance's vector are known, the choice of $[u_i, r_i]^T = \Delta^{-1}\Big(\mathrm{R}(\psi_i)^T(\dot{\mathbf{p}}_i^* + \dot{\mathbf{c}} - \mathbf{w}_i) - K(\mathbf{e}_i - \boldsymbol{\delta})\Big)$ guarantees the convergence of the position error, $\mathbf{e}_i - \boldsymbol{\delta}$, to the origin. Nevertheless, to maintain the initial conditions where only the relative position of the vehicles in relation to the target (expressed in the body reference frame) is known, we consider the target velocity unknown for the vehicles. In the same line of thought, and to evaluate the robustness of the control law from (5.8) the disturbance's vector is also considered unknown. This scenario occurs in reality when the disturbances cannot be measured or modeled.

In [4] the author proved, using the notion of ε-stability, that the use of the control law expressed in (5.8) guarantees convergence of the relative distance between each agent and the target to the set

$$\mathcal{R}_w = [R_0 - \delta - \varepsilon_w, R_0 + \delta + \varepsilon_w] \tag{5.15}$$

where ε_w is the radius of the circular trajectory for which the vehicles will converge, given by

$$\varepsilon_w = \frac{\gamma_c + \gamma_w}{\lambda_{\min}(K)} \tag{5.16}$$

Once again, by looking at (5.15) and to the expression for ε_w, (5.16), it is possible to see that the choice of matrix K has a significant impact in the region of convergence of the relative distance between each agent and the target. Thus, the choice of a large value for K would decrease the width of the region, implying that the vehicles would encircle the target more precisely. However as seen above, the value of K also impacts the bound of the control inputs as higher values of K increase the control inputs' bound. Thus, there exist a trade-off between setting a lower bound for the control inputs and increasing the accuracy at which the agents encircle the target.

For a better understanding of the impact of the disturbances in the controller's performance, another simulation was conducted. To allow comparison between simulations, the parameters used remain the ones used in 5.3, $\delta = 0.2$m and $K = 8I_2$. The results obtained are represented in figure 5.5.

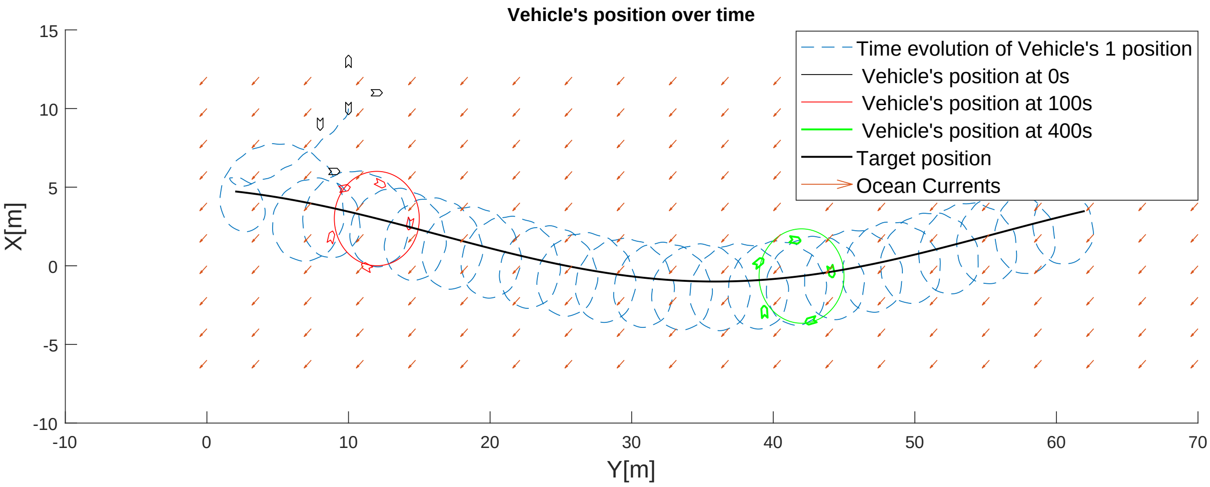


Figure 5.5: Circular motion around a moving path - $\delta = 0.2$m, $K = 8I_2$ in the presence of disturbances $\mathbf{w} = [-0.1, -0.1]\,\mathrm{m\,s^{-1}}$

In the results from 5.5, an ocean current was introduced. The ocean current's velocity vector was considered unknown for the formation. As expected, since no compensation is made, the vehicles are displaced according to a vector proportional to the current velocity vector. The circular motion performed by the vehicles is similar to the one obtained in 5.3. Increasing the values of K will result in a more precise circular motion, but at the same time, since the control parameters are varying in time with a higher frequency, the vehicles will have a more oscillating behavior. The vehicles will oscillate with a higher frequency along their trajectory, which is not desired. An increase in δ reduces the oscillatory behavior caused by the increase in K. However, we recall the negative impact of increasing K and δ. Once again, it is necessary to choose the values of these parameters to get an intermediate behavior and satisfactory bounds for the control inputs.

5.4 Alternative to the Exosystem

The previous control design uses the unicycle model to represent the model for the virtual vehicle created inside the exosystem. The goal for the exosystem is to generate a reference for the relative position, by stabilizing the motion of the virtual vehicle on a circular trajectory around the origin. The reference for the relative position should also converge to a uniform distribution to ensure that the real vehicle will converge to the formation. For a circular path, the case in study, the reference positions can be generated without the need for such complex design. We resort to trigonometry to generate the reference positions, that are initialized over the circular trajectory, without the need to converge to the circular trajectory.

When we consider the motion of a point located over the circumference with radius R, centered at the origin, its positions can be computed in complex notation as

$$z = x + jy = R\Big(\cos\big(\theta(t)\big) + j\sin\big(\theta(t)\big)\Big), \tag{5.17}$$

where j denotes the imaginary number, and $\theta(t)$ is a time varying function used as argument to the trigonometric functions. The angle θ at a certain time instant t, can be defined as

$$\theta = 2\pi\frac{t}{T} = \omega t, \tag{5.18}$$

where ω is the angular velocity intended for the formation.

By combining the previous equations, (5.17) and (5.18), we can define the reference for the relative position vector as,

$$\mathbf{p}^* = \Big[R\cos\big(\omega t\big), R\sin\big(\omega t\big)\Big]^T. \tag{5.19}$$

Yet to be defined is a mechanism that replaces the potential function and equally spaces the vehicles along the circumference. The positions generated for two consecutive vehicles must have a angular spacing of $\frac{2\pi}{N}$, where N is the number of vehicles involved. To achieve this spacing, we resort to the phase shifting of the trigonometric function, i.e., we introduce an offset in the trigonometric function, that depends on the vehicle number. The relative position vector from 5.19, can be re-defined to ensure the intended formation.

$$\mathbf{p}_i^* = \Big[R\cos\big(\omega t + 2(i-1)\frac{2\pi}{N}\big); R\sin\big(\omega t + 2(i-1)\frac{2\pi}{N}\big)\Big]^T, \tag{5.20}$$

where $2(i-1)\frac{2\pi}{N}$ denotes the phase shifting. The previous equation replaces the exosystem in the control design. The vehicles will converge to the desired formation and perform the desired maneuver along the moving path. To validate the strategy, we performed a simulation using the same parameters as in 5.3, $\delta = 0.2$m and $K = 8I_2$. The results are represented in figure 5.6.

The results from 5.6 are similar to the ones obtained in 5.3, as the steady-state reference positions are the same, apart from a temporal shifting. The difference relies on the approach to the circular path, that is done differently, as it is possible to see when we compare trajectory of one of the vehicles, represented by a dashed blue line in both figures. The exosystem references for the relative positions are initialized at the origin, and later converge to the circular motion, which makes the vehicles approach the target at the beginning, and after some time, converge to the path centered on the moving target. As for the exosystem alternative, the references for the relative positions are initialized over the circle centered at the origin, which makes the vehicle approach the circular path around the target instead of the target itself. The differences are not meaningful, as the different behaviors quickly fade away.

Thus, the introduction of this alternative provides a more straightforward method to obtain an equally spaced formation along a moving circular path. This method does not require the knowledge of the vehicle's model and generates *a priory* relative positions, that are shifted in space according to the

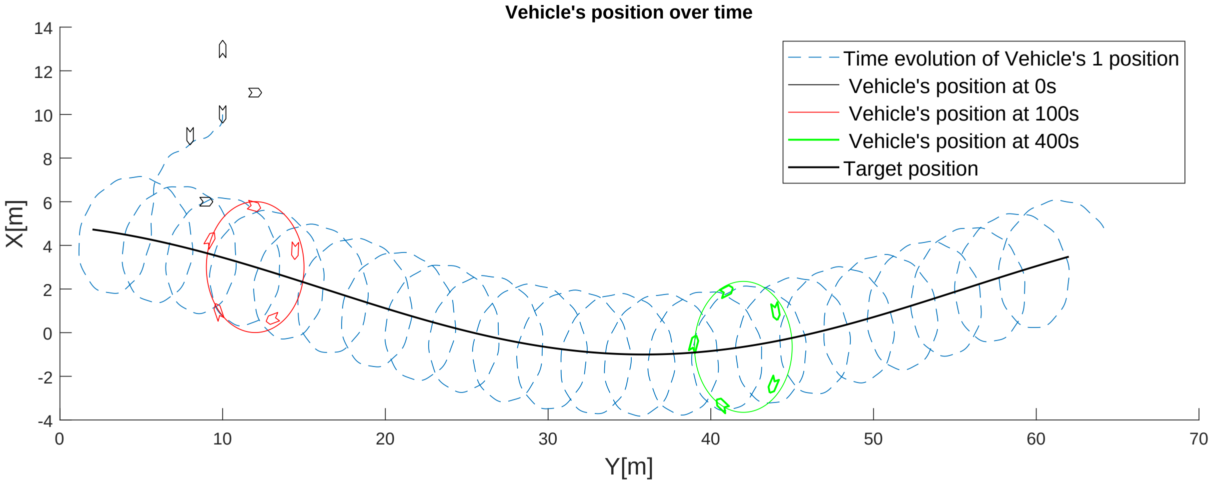


Figure 5.6: Circular motion around a moving target using an exosystem alternative

target position and used as reference positions for the vehicles.

Both control designs are performing trajectory tracking, where the time is involved as a constraint, i.e., a position is provided as a reference at each time instant. The reference position moves in time and assumes that the vehicles can track the relative positions. In the presence of a malfunction that prevents a vehicle from tracking its reference, , there exists no compensation from the controller, and the future references for the relative position will be generated regardless of the malfunction. The simulation results are good and promising. In real-life applications, some problems may arise as the disturbances may affect the vehicles independently, and the controller cannot adjust its functioning.

5.5 Real Vehicle State Feedback

The original design proposed in [4] synchronizes the vehicles around the path by providing synchronized reference positions for the vehicles to track. In the same line of thought, the reference positions, are synchronized without independently from the vehicle's state, i.e., the exosystem runs independently from its vehicle. Inside the potential function, the heading angle of virtual vehicles corresponding to adjacent vehicles is used. For this reasons, in the presence of a malfunction that prevents the vehicle from tracking its reference the exosystem will continue to run unaffected, and provide relative positions that does not take into consideration the malfunction. For this reason, and considering the context of this work, we conducted an experiment where we used real state vehicle's feedback. Thus, in the original control architecture, 5.1, we used the vehicle's heading in the potential function. This change is not supported by a rigorous mathematical analysis. The experience was conducted only to evaluate the possibility of using feedback from the vehicle's state to achieve the coordination between the vehicles. The control architecture is represented in figure 5.7.

We performed a simulation maintaining the conditions from 5.3, $\delta = 0.2\text{m}$ and $K = 8I_2$. The results obtained are represented in figure 5.8.

From the analysis of figure 5.8, it is possible to see that the vehicles successfully performed the maneuver. We find that at time 100s, the vehicles were not equally spaced over the desired path as it happened in 5.3. Thus, we conclude that the vehicles required more time to converge to the desired formation. Another aspect is related to the trajectory of one of the vehicles, represented by the blue dashed line. During the approach to the target, the vehicle assumed a strange behavior and less gradual.

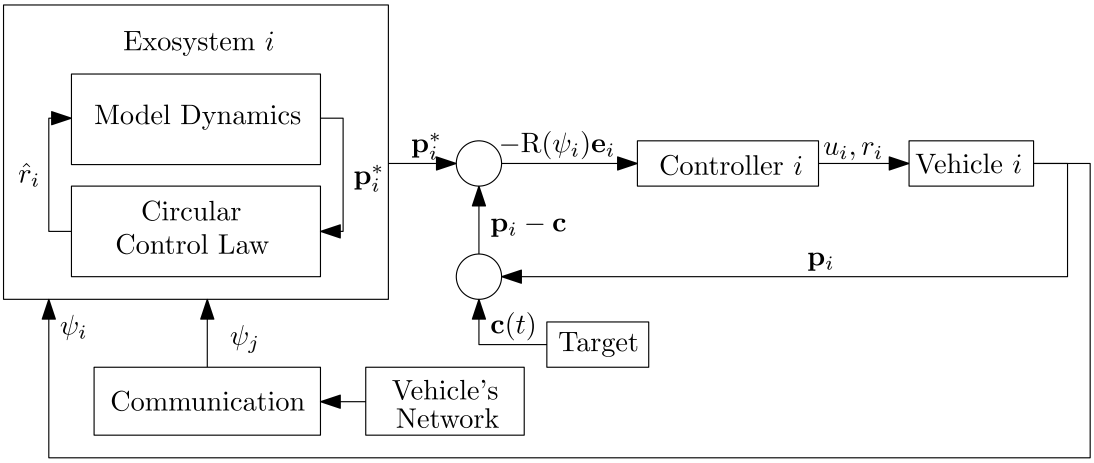


Figure 5.7: Control design using a decentralized exosystem with vehicles' heading feedback to exosystem

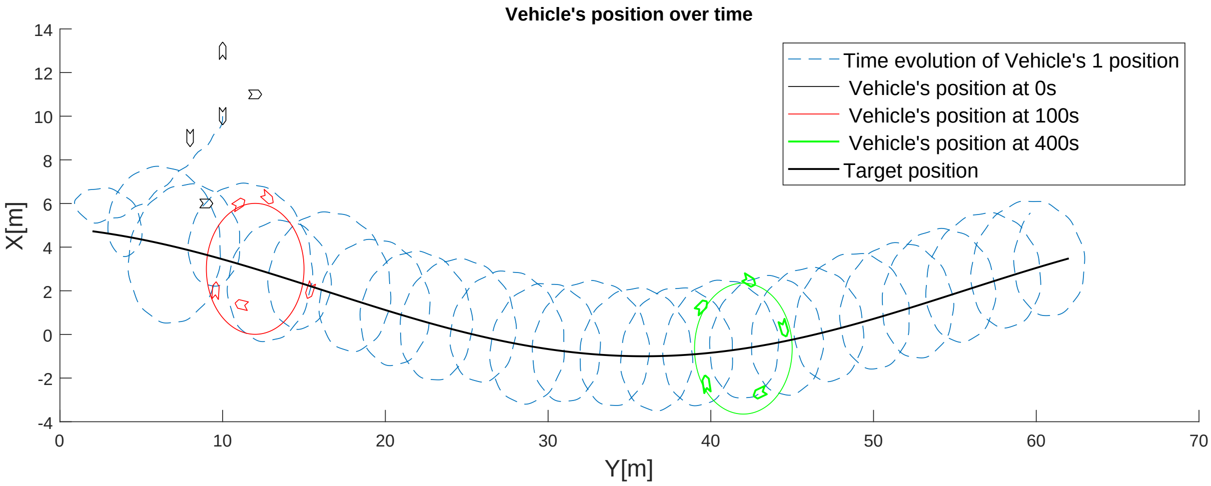


Figure 5.8: Circular motion around a moving target using vehicles' heading feedback into the exosystem

These findings are expected and are caused by the use of the real vehicles heading inside the potential function. In the original design, the references for the relative positions are initialized at the origin and later converge to the uniform distribution along a circumference centered at the origin. The uniform distribution is achieved by the potential function term. In the new design, the potential function is using the real vehicles' heading, which during the initial period is different from the virtual vehicle's heading. Thus, the potential function introduces in the virtual vehicles a repulsion force that is computed using an outside parameter, causing the virtual vehicles to converge to a different formation momentarily. Once the real vehicles track their references, the headings of the real and virtual vehicles becomes equal. At this point, the repulsion force ensures the convergence of the virtual vehicles to the desired formation, which indirectly causes the real vehicles to converge to the uniform distribution along the path.

It is important to say that this modification in the control design is not legitimized by a solid theoretical analysis. The simulation results are promising as the vehicles converged to the encircling maneuver. However, it is not possible to guarantee that the convergence is always verified and that in mission the results are the same. As a means to overcome these limitations, in the next chapter we introduce a novel algorithm for circular target tracking.

CHAPTER 6

Cooperative Moving Path Following

THROUGHOUT this chapter, we will design an innovative control law for target encircling purposes, whereby a group of vehicles is required to converge to a closed path and follow it at a desired speed, adopting a desired geometric configuration with respect to the path, while the center of the latter undergoes motions to pursue a target. We start by addressing the situation where the center of the path moves with the same linear speed as the target. The technique used is quite intuitive, and amounts to solving the PF control problem in a local reference frame attached to the path, after which the guidance law derived is expressed in the inertial reference frame. To synchronize the vehicles and guarantee the convergence to the equally spaced distribution over the path we use the controller designed in 4.4.

6.1 Motivating Mission

This thesis aims to extend the work done in the scope of the WiMUST project with the MEDUSA class vehicles. The project aimed at the development of control systems that allowed the used of autonomous vehicles in geophysical acoustic surveys at sea. The use of these vehicles will drastically improve the efficiency of the traditional methodologies. The key innovative concept was to have two Autonomous Surface Vehicles (ASVs) equipped with acoustic sources capable of insonifying the seafloor and the volume underneath it (the penetration under the seafloor is frequency dependent and can reach hundredths of meters). At the same time, a group of AUV tow streamers with hydrophones that collect the waves that bounce off the seafloor and the sub-seafloor. Interpretation by a geophysicist of the data obtained allow for channel inversion and therefore an interpretation or artifacts under the seafloor. Coordination among the surface and underwater vehicles is key to the success of the concept, because the relative geometry of the assets impacts directly on the quality of the survey. Thus the need to develop methods for cooperative motion control to operate the vehicles in a concerted manner, in the presence of temporary vehicle malfunctions and intermittent communication problems imposed by the water medium.

Geophysical experts are of the opinion that the quality of the survey can be improved if the ”virtual cells” at the bottom of the ocean can be ”illuminated” by acoustic waves coming from a large number of directions. To this end, there is the suggestion to maneuver the surface vehicles with the acoustic sources so as to continuously circle the group of underwater vehicles, thus emitting waves that truly hit the seafloor from different angles.

To achieve the above goals it is necessary to ensure that the vehicles move in formation, respecting desired geometrical constraints, but allowing at the same time the formation to be temporarily compliant to communication failures, external disturbances caused by currents and waves, and temporarily vehicle malfunctions. Some of the constraints can be simply understood by the fact that, in striking contrast to communications via WiFi at the surface, the underwater formation can only communicate among its members and with the surface vehicles using acoustic communications. The acoustic channel is low-bandwidth and prone to losses, and as a consequence the information to be sent across must be chosen in a very parsimonious manner. It is in this context that in this chapter we aim to contribute to the development of a solution to the above mentioned problems by exploring concepts from path following, cooperative path following, and logic-based communications. In particular, we advance a new conceptually simple solution to the so called moving path following problem first described in [3]. By exploring concepts and techniques from the above areas, a truly cooperative, compliant system for circular tracking of underwater targets (AUVs) is obtained. Simulations illustrate the efficacy of the methodology developed.

In what follows, for simplicity of exposition, we consider the case of a single AUV (target). The technique can be extended to a group of vehicles moving in a desired formation.

6.2 Target Translating

Consider a target moving arbitrarily over time. At the target's center, we fix a frame named $\{P\}$. Defined in $\{P\}$ is the path intended for the circling vehicles, denoted $\mathcal{P}$. It is essential to note that the path $\mathcal{P}$ is rigidly attached to $\{P\}$, which means that $\mathcal{P}$ is invariant in $\{P\}$. The set-up for the MPF problem, whereby a group of vehicles is supposed to converge to and follow the moving path at assigned speeds, is shown in figure 6.1, for the case of a non-rotating path. The position of the origin of $\{P\}$, which is coincident with the target's center, is represented by ${}^{U}\mathbf{p}_{\mathrm{o_p}}$. The linear velocity of $\{P\}$ in $\{U\}$ is represented by $\mathbf{v}_P$.

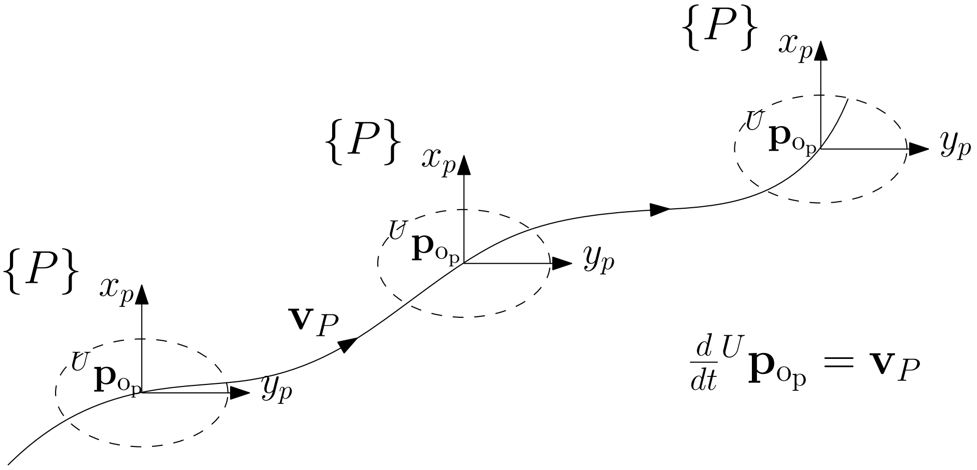

$\{P\}$
x_p
y_p
${}^{U}\mathbf{p}_{\mathrm{o_p}}$
$\mathbf{v}_P$
$\frac{d}{dt}{}^{U}\mathbf{p}_{\mathrm{o_p}} = \mathbf{v}_P$

Figure 6.1: MPF set-up considering a non-rotating path

The question that arises is how to make a group of vehicles converge and then follow the moving path with a desired speed profile along that same path.

As a first step, we focus on the frame $\{P\}$ and solve the PF problem in $\{P\}$, that is, in local coordinates. The solution will then be transferred to the inertial reference frame. Furthermore, the solution that we propose is done at a kinematic level. We start with a single vehicle. Let ${}^{P}\mathbf{p}$ denote the position of an agent expressed in $\{P\}$. The objective is to derive a local guidance law for the vehicle's velocity in $\{P\}$, ${}^{P}\mathbf{v}$, such that the vehicle follows the path $\mathcal{P}$ in $\{P\}$. The kinematic solution (guidance law) is necessarily of the form ${}^{P}\mathbf{v} = {}^{P}\mathbf{v}({}^{P}\mathbf{p}, \mathcal{P})$ where the latter expression means that the vehicle computes an instantaneous velocity vector in the local coordinates that is a function of the position of the vehicle in $\{P\}$, the geometry of the path $\mathcal{P}$, and the desired speed profile for the vehicle along $\mathcal{P}$. For the local control law we adopted the P. Maurya's guidance law, presented in 3.4.2. Now, it remains to define the guidance law in global coordinates, $\{U\}$, since frame $\{P\}$ is moving over time.

The inertial position of a vehicle can be expressed as the sum of the position of the origin of $\{P\}$, ${}^{U}\mathbf{p}_{\mathrm{o_p}}$ with the position of the vehicle in $\{P\}$, ${}^{P}\mathbf{p}$, properly multiplied by the rotation matrix from $\{P\}$ to $\{U\}$, that is,

$$ {}^{U}\mathbf{p} = {}^{U}\mathbf{p}_{\mathrm{o_p}} + {}^{U}_{P}\mathrm{R} \cdot {}^{P}\mathbf{p}. \tag{6.1}$$

Considering that frame $\{P\}$ is moving with linear velocity but does not rotate, the rotation matrix ${}^{U}_{P}\mathrm{R}$ is equal to the identity and, therefore, equation (6.1) can be rewritten as

$$ {}^{U}\mathbf{p} = {}^{U}\mathbf{p}_{\mathrm{o_p}} + {}^{P}\mathbf{p}. \tag{6.2}$$

By computing the derivative of both sides of equation (6.2) we obtain the equation for the global guidance law given by

$$\underset{\substack{\downarrow\\ \text{Guidance law}\\ \text{in } \{U\}}}{{}^{U}\mathbf{v}} = \frac{d}{dt}{}^{U}\mathbf{p} = \underset{\substack{\downarrow\\ \text{Linear}\\ \text{velocity}\\ \text{of } \{P\}}}{\mathbf{v}_P} + \underset{\substack{\downarrow\\ \text{Guidance}\\ \text{law in } \{P\}}}{{}^{P}\mathbf{v}} \ . \tag{6.3}$$

To use the global guidance law from (6.3) the vehicle must perform the following steps:

1. Compute its position ${}^{U}\mathbf{p}$ in $\{U\}$;
2. Compute its position ${}^{P}\mathbf{p} = {}^{U}\mathbf{p} - {}^{U}\mathbf{p}_{\mathrm{o_p}}$ in $\{P\}$;
3. Compute the local kinematic guidance law ${}^{P}\mathbf{v} = {}^{P}\mathbf{v}({}^{P}\mathbf{p}, \mathcal{P})$;
4. Transfer the guidance law to the inertial reference frame as ${}^{U}\mathbf{v} = \mathbf{v}_P + {}^{P}\mathbf{v}$.

6.3 Target Translating and Rotating

Throughout this section we will extend the previous results to the scenario where the geometric path has both linear and angular velocities, $\mathbf{v}_P$ and ω_P, respectively. As done in 6.2 we consider that $\mathcal{P}$ is rigidly attached to $\{P\}$. The set-up for MPF, considering that the path rotates over time is represented in figure 6.2. The PF problem is once again solved in local coordinates (expressed in $\{P\}$) and P. Maurya's guidance law is used in the local guidance law.

As done previously, we start by defining the inertial position of a vehicle as a sum of the inertial position of the origin of $\{P\}$, ${}^{U}\mathbf{p}_{\mathrm{o_p}}$, with the position of the vehicle in the frame $\{P\}$ multiplied by the rotation matrix from $\{P\}$ to $\{U\}$. Thus, the inertial position of the agent is given by

$$ {}^{U}\mathbf{p} = {}^{U}\mathbf{p}_{\mathrm{o_p}} + {}^{U}_{P}\mathrm{R} \cdot {}^{P}\mathbf{p}. \tag{6.4}$$

By looking at figure 6.2 it is possible to see that the reference frame $\{P\}$ has different orientations over time. Thus, and considering that the inertial reference frame is fixed on any place on earth it is clear that the orientation of $\{P\}$ is not the same as the orientation of $\{U\}$ for every time instant. For this reason, the rotation matrix is now different from the identity. We may now derive both sides of equation

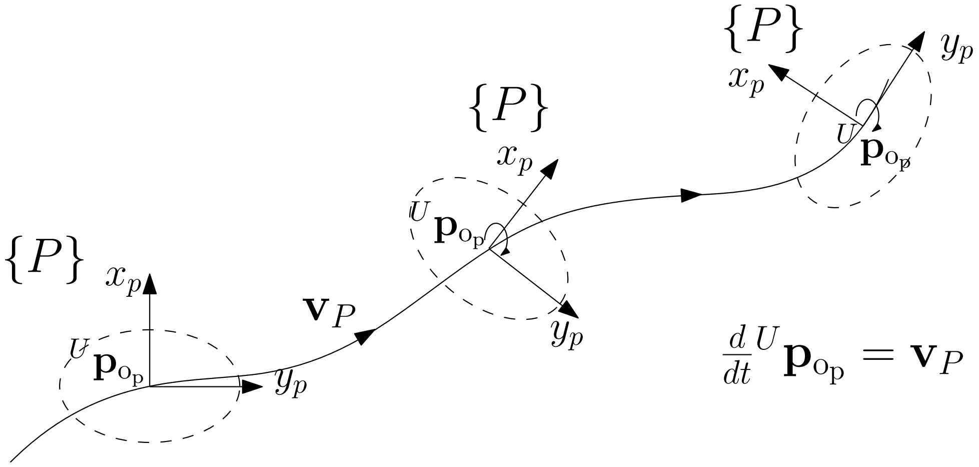

{P}
x_p
y_p
{P}
x_p
y_p
{P}
x_p
y_p
${}^{U}\mathbf{p}_{\mathrm{o_p}}$
${}^{U}\mathbf{p}_{\mathrm{o_p}}$
${}^{U}\mathbf{p}_{\mathrm{o_p}}$
$\mathbf{v}_P$
$\frac{d}{dt}{}^{U}\mathbf{p}_{\mathrm{o_p}} = \mathbf{v}_P$

Figure 6.2: MPF set-up considering a rotating path

(6.4) to obtain the equation for the global guidance law as follows:

$$ {}^{U}\mathbf{v} = \frac{d}{dt}{}^{U}\mathbf{p} = \mathbf{v}_P + {}^{U}_{P}\dot{\mathrm{R}} \cdot {}^{P}\mathbf{p} + {}^{U}_{P}\mathrm{R} \cdot {}^{P}\mathbf{v}. \tag{6.5} $$

The derivative of the rotation matrix, ${}^{U}_{P}\dot{\mathrm{R}}$, is equal to $R\mathcal{S}(\omega_P)$, where ω_P is the speed of rotation of $\{P\}$ with respect to $\{U\}$ expressed in $\{P\}$. Based on a property of the term $\mathcal{S}(\omega_P)$, $\mathcal{S}(\mathrm{w})\mathrm{a} = \mathrm{w} \times \mathrm{a}$, we may rewrite the guidance law in $\{U\}$ as

$$ \underset{\substack{\downarrow \\ \text{Guidance law} \\ \text{in } \{U\}}}{{}^{U}\mathbf{v}} = \underset{\substack{\downarrow \\ \text{Linear} \\ \text{velocity} \\ \text{of } \{P\}}}{\mathbf{v}_P} + {}^{U}_{P}\mathrm{R} \cdot \underset{\substack{\downarrow \\ \text{Angular} \\ \text{velocity} \\ \text{of } \{P\}}}{\omega_P} \times {}^{P}\mathbf{p} + {}^{U}_{P}\mathrm{R} \cdot \underset{\substack{\downarrow \\ \text{Guidance} \\ \text{law in } \{P\}}}{{}^{P}\mathbf{v}} . \tag{6.6} $$

In (6.6) we defined a generic global guidance law that can be used for both non-rotating and rotating paths. Considering a non-rotating path it is clear that the rotation matrix from $\{P\}$ to $\{U\}$ corresponds to the identity. In the same line of though, the angular velocity is zero, thus $\omega_P = 0$. In this case, the global guidance law degenerates into the one defined in (6.3).

6.4 Robustness to External Disturbances

In the scenarios where the vehicles are deployed, external disturbances, such as the ocean currents, affect the vehicles and cause the heading angle to be different from the course angle. Due to the maneuver's nature, and considering a non-zero curvature path, the local guidance law that generates ${}^{P}\mathbf{v}$ does not guarantee convergence of the vehicles to the geometric path. Thus, when we transfer the guidance law to $\{U\}$, a good execution of the MPF maneuver cannot be guaranteed. To overcome this situation, it is necessary to take into consideration the presence of the external disturbances in the global guidance law. Fortunately, the influence of a constant water current can be easily taken into account at the kinematic level.

Consider a practical situation, where the ocean currents are not known beforehand and that the vehicles do not have direct access to the ocean currents. We use the estimates provided by the complementary filter from (3.25). We deal with the ocean currents in the local reference frame $\{P\}$. Thus, in the global guidance law from (6.6) we simply subtract the estimates of the ocean currents. The global guidance law, taking into consideration the presence of external disturbances, can be rewritten as

$$^{U}\mathbf{v} = \mathbf{v}_P + {}^{U}_{P}\mathrm{R} \cdot \omega_P \times {}^{P}\mathbf{p} + {}^{U}_{P}\mathrm{R} \cdot {}^{P}\mathbf{v} - \hat{\mathbf{v}}_c. \tag{6.7}$$

Using the global guidance law from (6.7) the vehicles will converge to the geometric path $\mathcal{P}$ even in the presence of ocean currents.

It is very important to remark that the analysis above was done for the case of a single vehicle. A straightforward extension of the methodology, using the results in the chapter on cooperative path following, can be done to derive the kinematic guidance laws for each and every vehicle, where the local velocity vector of a given vehicle, say i, is now computed as

$$^{P}\mathbf{v}_i = {}^{P}\mathbf{v}\left({}^{P}\mathbf{p}_i, \mathcal{P}\right), \tag{6.8}$$

where now the local guidance law takes into account the interaction with its neighbours, with a view to achieving cooperative path following with a given geometric configuration and a common speed profile along the path. In this work, we will focus on the case where the paths are circumferences.

6.5 Problem Formulation Using Virtual Reference Frame

In the previous sections, we solved the PF problem in $\{P\}$ using a local guidance law. After this, considering that frame $\{P\}$ is moving over time, we transferred the local guidance law to a global guidance law defined in the inertial reference frame. The convergence of the vehicles to the moving-path is ensured, regardless of the distance between the vehicles and the moving-path. However, considering that the target is initially far from the vehicles, the distance is seen as a large cross-track error in $\{P\}$. Thus, the P. Maurya's guidance law (used in the local guidance law) may yield undesired transients. Motivated by this situation, we introduce a virtual target on which we fix a reference frame named $\{VP\}$. In $\{VP\}$ we rigidly attach a virtual path, $\mathcal{VP}$, that is an exact copy of the path $\mathcal{P}$. We formulate the PF task in $\{VP\}$ which implies that the vehicles converge to $\mathcal{VP}$ instead. Furthermore, we will make sure that the frame $\{VP\}$ will tend asymptotically to $\{P\}$. This will make the vehicle's approach to the desired path more smooth and orderly. The two reference frames, $\{P\}$ and $\{VP\}$, are represented in figure 6.3.

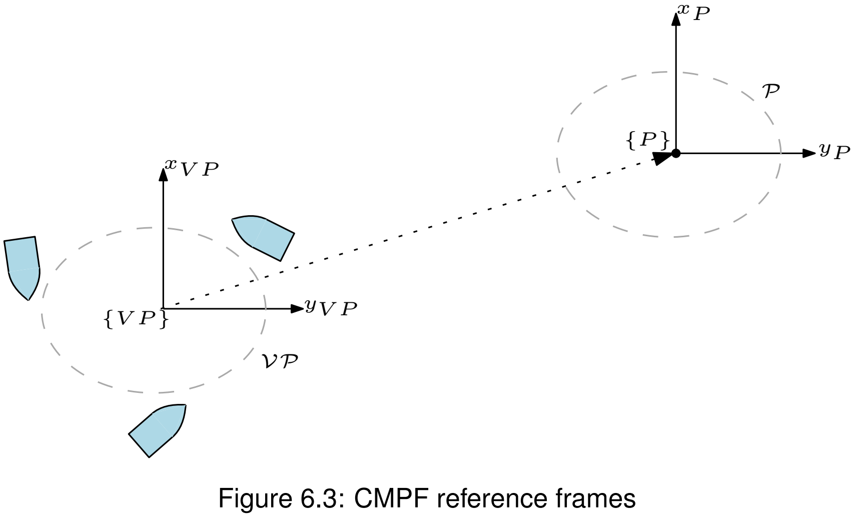


Figure 6.3: CMPF reference frames

To avoid redundant calculations, we will not derive the steps necessary to obtain the global guidance law, considering that the PF is solved in $\{VP\}$. The fact that we are formulating the problem in a different frame does not affect the mathematical derivation of the global guidance law. Instead, the difference lies in the notation used during the formulation. In (6.7), the subscript and superscript $_P$ and ^{P}, respectively, become $_{VP}$ and ^{VP} and instead of using the linear and angular velocity of $\{P\}$ we use the linear and angular velocity of $\{VP\}$, $\mathbf{v}_{VP}$, and ω_{VP}. Thus, from (6.7) the guidance law is rewritten as

$$^{U}\mathbf{v} = \mathbf{v}_{VP} + {}^{U}_{VP}\mathrm{R} \cdot \omega_{VP} \times {}^{VP}\mathbf{p} + {}^{U}_{VP}\mathrm{R} \cdot {}^{VP}\mathbf{v} - \hat{\mathbf{v}}_{\mathrm{c}}, \tag{6.9}$$

where $^{VP}\mathbf{p}$ denotes the position of a generic vehicle in $\{VP\}$. The inertial guidance law from (6.9) guarantees convergence of the vehicle to the geometric path $\mathcal{VP}$, known as a virtual path. Still to be defined is the motion of $\{VP\}$, and how to guarantee convergence of frame $\{VP\}$ to $\{P\}$. Regarding the orientation of $\{VP\}$, this reference frame is initialized with the same orientation as $\{P\}$. Thus, during the maneuver we set the angular velocity of $\{VP\}$ equal to the angular velocity of $\{P\}$, that is,

$$\omega_{VP} = \omega_{P}. \tag{6.10}$$

From the fact that the two reference frames are initialized with the same orientation and have equal angular velocities we conclude that the orientation of the two reference frames will be the same over the maneuver. In practice, synchronization of the two frames must of course be ensured by some means of communication between the agents that update the information on the motions of both targets. This is simply an implementation issue.

Notice that the origin of frame $\{VP\}$ is not initialized at the origin of $\{P\}$ (at the beginning of the circular tracking maneuver, the underwater targets may be quite far from the trackers at the surface). Thus, to guarantee convergence of the two reference frames, $\mathbf{v}_{VP}$ must be chosen to guarantee the correction of the mismatch between the origin of $\{VP\}$ and $\{P\}$, $^{U}\mathbf{p}_{\mathrm{o_{vp}}}$ and $^{U}\mathbf{p}_{\mathrm{o_{p}}}$, respectively. The strategy is to provide to $\mathbf{v}_{VP}$ the linear velocity of $\{P\}$ plus a correction term

$$\mathbf{v}_{VP} = \mathbf{v}_{P} + \mathbf{v}_{\mathrm{corr}}, \tag{6.11}$$

where $\mathbf{v}_{\mathrm{corr}}$ denotes the correction term that is based on proportional feedback and is simply given by

$$\mathbf{v}_{\mathrm{corr}} = K \tanh\left({}^{U}\mathbf{p}_{\mathrm{o_{p}}} - {}^{U}\mathbf{p}_{\mathrm{o_{vp}}}\right). \tag{6.12}$$

Note that the controller defined in (6.12) is a straightforward adaptation of a proportional controller. The use of the hyperbolic tangent function imposes a bound on the velocity correction term as determined by K.

For practical implementation purposes, it is important to initialize the positions of the surface vehicles properly. In situations where the latter and and the underwater vehicles start far apart, it makes no sense for the trackers to follow the moving path $\mathcal{VP}$, for this would be a waste of energy. In this situation, the trackers should try and move fast to a defined neighborhood of the target, at which point they start executing moving path following along the path $\mathcal{VP}$. This can be simply done by estimating, by some means, the distance between the target and the "center of mass" of the surface vehicles. Based on this distance, we adjust the vehicles' tasks, i.e., if this distance is higher than a defined threshold, the vehicles perform PF along a straight line that connects the target with the formation mass center. Once this distance becomes smaller than the defined threshold, we initialize the virtual target on the formation mass center, and the vehicles perform MPF with respect to $\mathcal{VP}$.

Considering real missions where these guidance laws can be applied, such as ocean seismic surveys, it is important to know the role of each agent involved in the maneuver. For this purpose, the vehicle (or set of vehicles) that carry the receiving devices (streamers with hydrophones) will correspond to one or multiple targets. If only one vehicle is considered, then this vehicle will execute PF over a predefined path that is mission dependent. If at least two vehicles carry receiving devices, then they will be performing cooperative path following, synchronizing their positions along the underwater pre-defined paths. In addition, the origin of the virtual path $\mathcal{VP}$ will be executing another CPF maneuver with a target or, in the case of multiple targets, with the center of the underwater formation as a means to synchronize their positions over time. Notice that by endowing the surface and underwater vehicles to perform CPF, some form of compliance will be introduced in the general scheme.

6.6 Simulation Results

Throughout this section we will consider a fleet composed by three agents that are required to converge to a common moving geometric path and, at the same time, converge to a uniformly distributed formation along the path. To synchronize the vehicles during the maneuver, the local guidance law in $\{VP\}$ is adjusted and, the absolute values of ${}^{P}\mathbf{v}$ is set to v_d defined in (4.21). The orientation of ${}^{P}\mathbf{v}$ is provided by P. Maurya's guidance law. Some of the parameters involved in the control algorithms were fixed trough all simulations:

- $K = 0.05 \rightarrow$ Gain of the correction term in the reference frame velocity, (6.12);
- $k_e = 0.2 \rightarrow$ Gain of the coordination correction term, (4.22);
- $v_r = 0.6\,\mathrm{m\,s^{-1}} \rightarrow$ Formation speed reference.

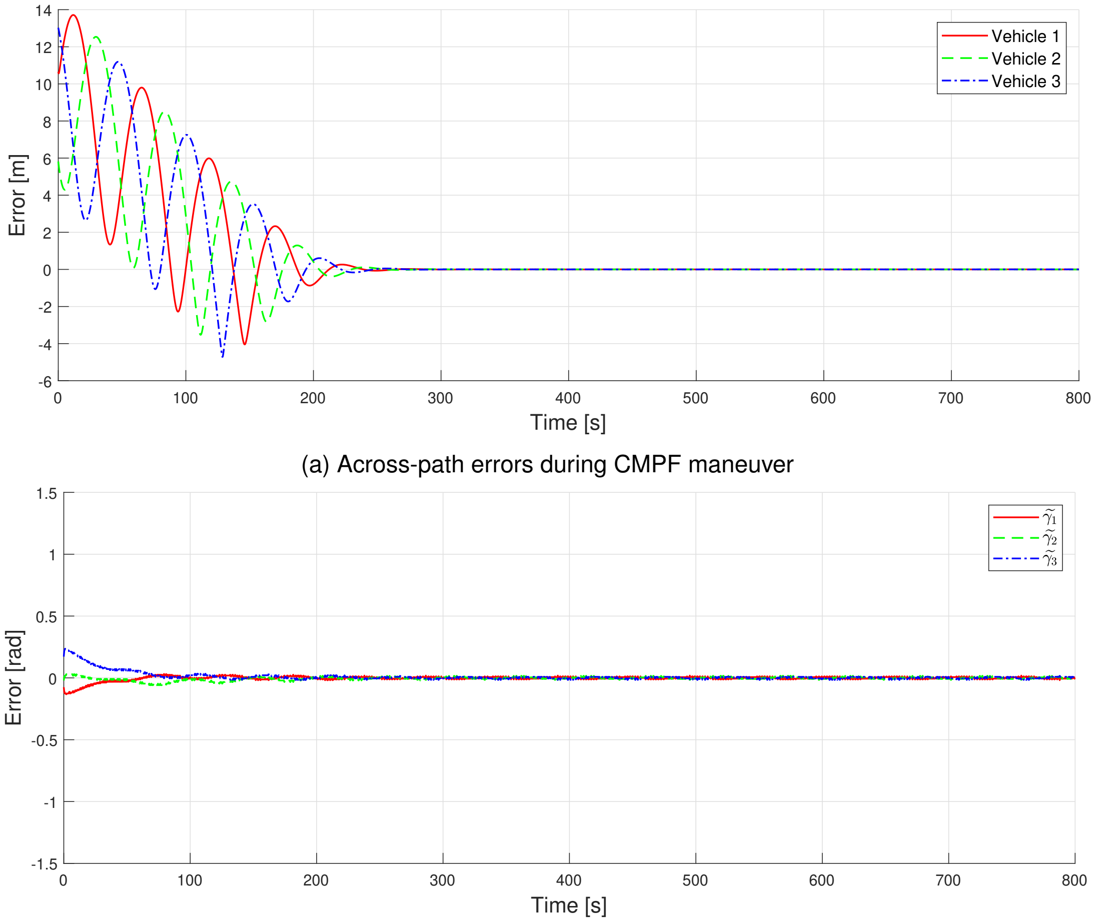
6.6.1 Non-Rotating Path

We start by considering a non-rotating path and evaluate the performance of the vehicles using the control law from (6.9). Before analyzing the performance of the MEDUSA class vehicles we start by testing the control algorithms considering a set of unicycle-like agents. This model does not consider the dynamics, but only the kinematics. For this reason, the use of this model will provide useful information about the performance expected of the CMPF control law.

For the first simulation we considered an arbitrary target moving with a linear velocity of $\mathbf{v}_P = [0.05; 0.05]\,\mathrm{m\,s^{-1}}$. The specified geometric path is a circumference with radius $R = 5$m. The results obtained are represented in figure 6.4.

By looking at figure 6.4a we conclude that, as expected, the use of the global guidance law from (6.9) guaranteed convergence of the cross-track error to the origin for the unicycle-like agents. The target was initialized at approximately 10m away from the vehicles. For the simulation, we considered that the origin of the virtual frame was initialized at the start of the simulation at the average initial position of the vehicles. Thus, the vehicles start following $\mathcal{VP}$ right after the beginning of simulation. For this reason, the coordination is achieved early in the simulation. As the vehicles synchronize their positions, the coordination error represented in 6.4b, converges to zero. During the maneuver, the vehicles navigate synchronously as the coordination error remains at the zero.

After running the simplified model, we replaced the unicycle model with the MEDUSA vehicle model. We performed a simulation under the same condition as 6.4. The results are represented in figure

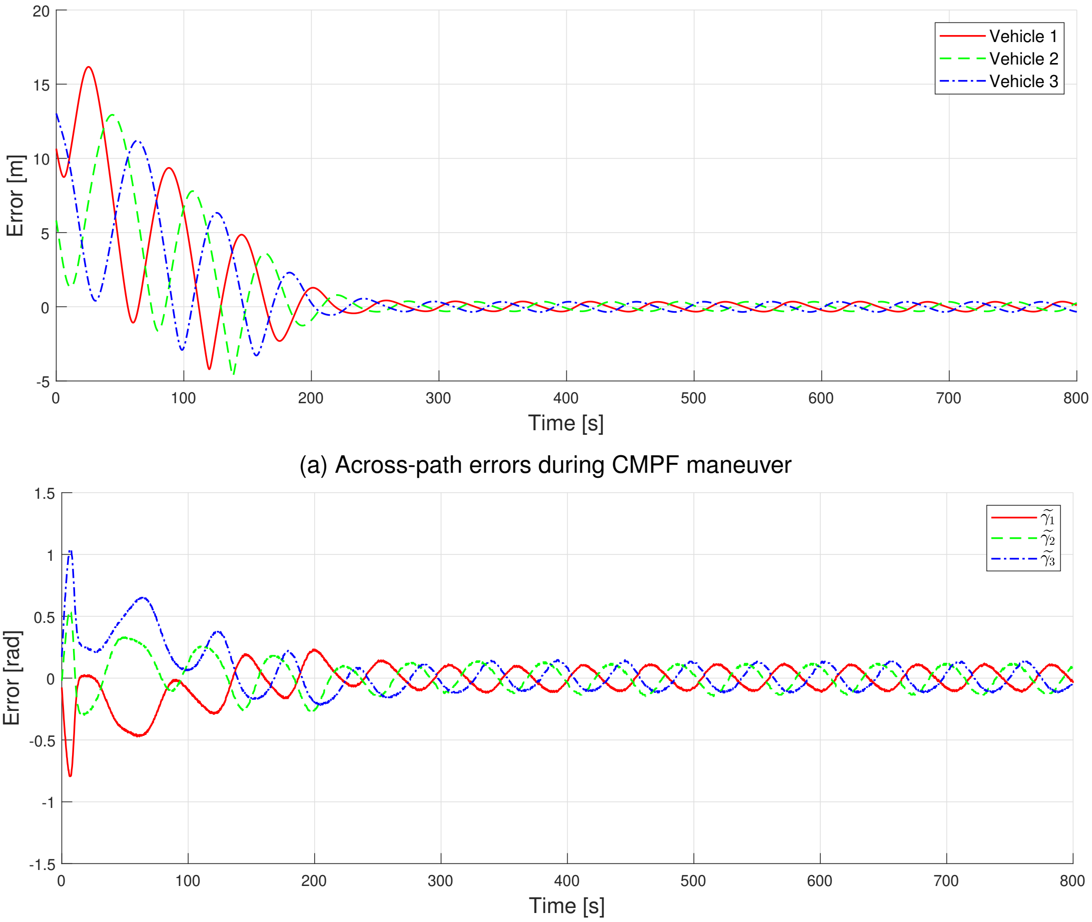


(a) Across-path errors during CMPF maneuver

(b) Coordination error: difference between synchronization parameters

Figure 6.4: Simulation results for a circular moving path using a simplified model

6.5. By comparing the results from 6.5 with the results from 6.4, there are some notorious differences. When the MEDUSA vehicles model is used, the cross-track error, represented in figure 6.5a, does not converge to the origin. Instead, it converges to a region centered at the origin, delimited by $[-0.4,\, 0.4]\mathrm{m}$. After reaching the convergence region, the cross-track error oscillates inside this region. The oscillation is related with the fact that in the MEDUSA vehicles there exists a finite heading rate as well as a finite acceleration. These finite rates cause a delay between the instant where the control inputs are received and the time instant where the vehicle's state reflects these inputs. The delay is a physical phenomenon that can not be eliminated. The fact that the cross-track error oscillates without converging to the origin causes the coordination error to behave similarly. As in the case of the cross-track error, the coordination error also converges to a region centered at zero, delimited by $[-0.2,\, 0.2]\,\mathrm{rad}$. The bounds of the convergence region can be seen as residual, as they correspond to approximately $\pm 10^{\circ}$.



(a) Across-path errors during CMPF maneuver

(b) Coordination error: difference between synchronization parameters

Figure 6.5: Simulation results for a circular moving path using the MEDUSA vehicle models

The vehicles' speed is represented in figure 6.6. Contrarily to what happened throughout chapter 4, the speeds do not converge to a steady-state value. Instead the speeds assume an oscillatory behavior centered on the nominal speed reference for the formation, $v_r = 0.6\,\mathrm{m\,s^{-1}}$. The nature of the oscillations is related with the nature of the encircle maneuver. Throughout the maneuver the vehicles perform complete turns around the moving target. Thus, its orientation varies over time. At some instant the vehicles are aligned with the target velocity vector. These instants correspond to the maximum values in the vehicles' speeds. On the opposite side, when the vehicles are aligned in the opposite direction to the target velocity vector, the speeds reach their minimum values.

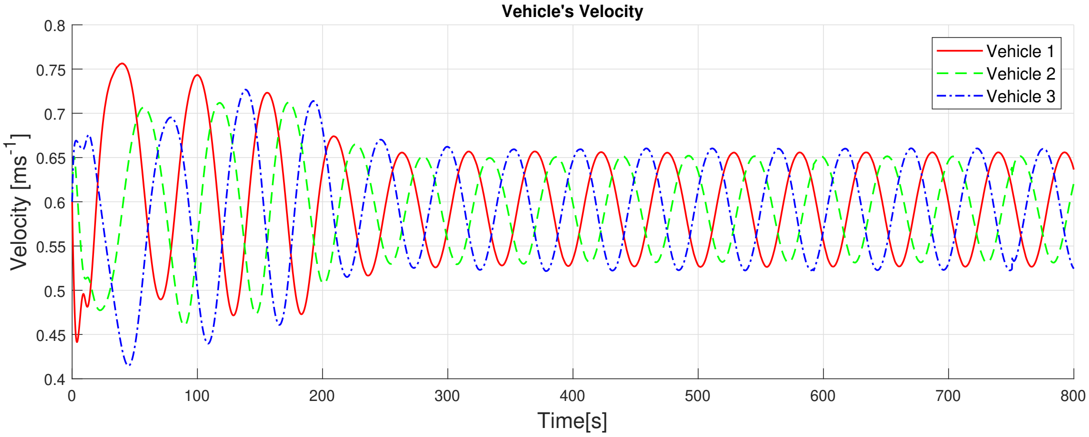


Figure 6.6: Vehicles velocity for a circular moving path using MEDUSA vehicle models

In the results represented in 6.5, the cross-track error converged to a symmetric region centered at zero. These results are not valid for all paths. The convergence region of the cross-track error for different circumference radius, R, is represented in table 6.1. In the simulations, we considered a target moving with linear velocity $\mathbf{v}_P = [0.05; 0.05]$ m s^{-1}.

Circumference Radius [m]	**Cross-Track Error Convergence Region** [m]	
	Lower bound	Upper bound
2	-0.12	1.25
3	-1.50	1.31
4	-0.47	0.45
5	-0.40	0.40
6	-0.30	0.29
7	-0.26	0.24
8	-0.23	0.21
9	-0.20	0.18
10	-0.18	0.16

Table 6.1: Cross-track error convergence region for a circular path with different radii

From the analysis of the results presented in table 6.1 we conclude that, for a circular path with radius equal or higher than 4m, the convergence region is centered at the origin. For smaller radii, the convergence region is not symmetric about the origin. This phenomenon is related to the path dimensions. The path covers a small area that, with the target linear velocity considered, makes it difficult for the vehicles to follow the paths. Nevertheless, in practical applications the geometric path intended for the vehicles will not have radius lower than 4m. For a higher radius, the vehicles see a smaller local displacement of the path and perform the maneuver with a cross-track error centered at zero. The convergence region is not precisely symmetric as it is possible to detect a residual difference in the bound value.

The other parameter that impacts the convergence region of the cross-track error is the target linear velocity. In table 6.2 are represented the bounds of the convergence region of the cross-track error for different target linear velocities. We considered a circular path with a radius $R = 5m$.

From the analysis of table 6.2, we conclude that there exists a direct relationship between the target linear velocity and the bounds of the cross-track error convergence region. The higher the target's linear velocity, the less time the vehicles have to perform the maneuver, which reduces the precision at which they can follow the moving path. Apart from a small variation, the symmetry of the convergence region

Target Velocity $\mathbf{v}_P$ $[\mathrm{m\,s^{-1}}]$	**Cross-track Error Convergence Region** $[\mathrm{m}]$	
	Lower bound	Upper bound
$[0.05;\ 0.05]$	-0.40	0.40
$[0.07;\ 0.07]$	-0.63	0.63
$[0.10;\ 0.10]$	-0.95	0.95
$[0.12;\ 0.12]$	-1.22	1.20
$[0.15;\ 0.15]$	-1.63	1.56

Table 6.2: Cross-track error convergence region for a circular path with different linear velocities

is verified for the different target velocities.

6.6.2 Rotating Path

To evaluate the performance of the control law from (6.9) for a path rotating with a known angular velocity we will consider both constant curvature paths and non-constant curvature paths. For non-constant curvature paths, we consider elliptical paths with similar axis lengths.

6.6.2.1 Constant Curvature Paths

For the analysis of the vehicles' behavior for a constant curvature path, we consider a circular path with radius $R = 5\mathrm{m}$. As it was done previously, we start by simulating the situation where the set of vehicles are unicycle-like agents.

The intended geometric path is fixed to a target moving with linear velocity $\mathbf{v}_P = [0.05;\ 0.05]\,\mathrm{m\,s^{-1}}$. We may now introduce in the path an angular velocity of $\omega_P = 0.02\mathrm{rad\,s^{-1}}$. Both position and synchronization errors were similar to the ones obtained in 6.4. The difference between simulations lies in the introduction of the path rotation. Considering the path shape, it becomes clear that, for an observer fixed on the inertial reference frame, the path angular velocity in not observable, i.e., the observer only sees the path translating in time with linear velocity $\mathbf{v}_P$. For this reason, the introduction of a small angular velocity does not affect the cross-track error. Instead it only affects the vehicles speeds that are adjusted to ”keep-up” with the path rotation. In figure 6.7 is represented the velocity of one of the vehicles for simulation 6.4 (non-rotating path) and for the rotating path.

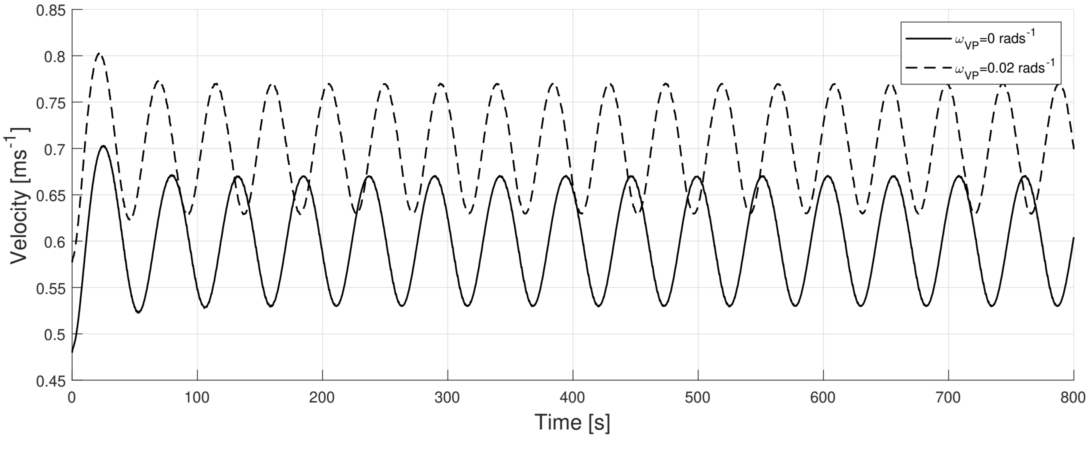
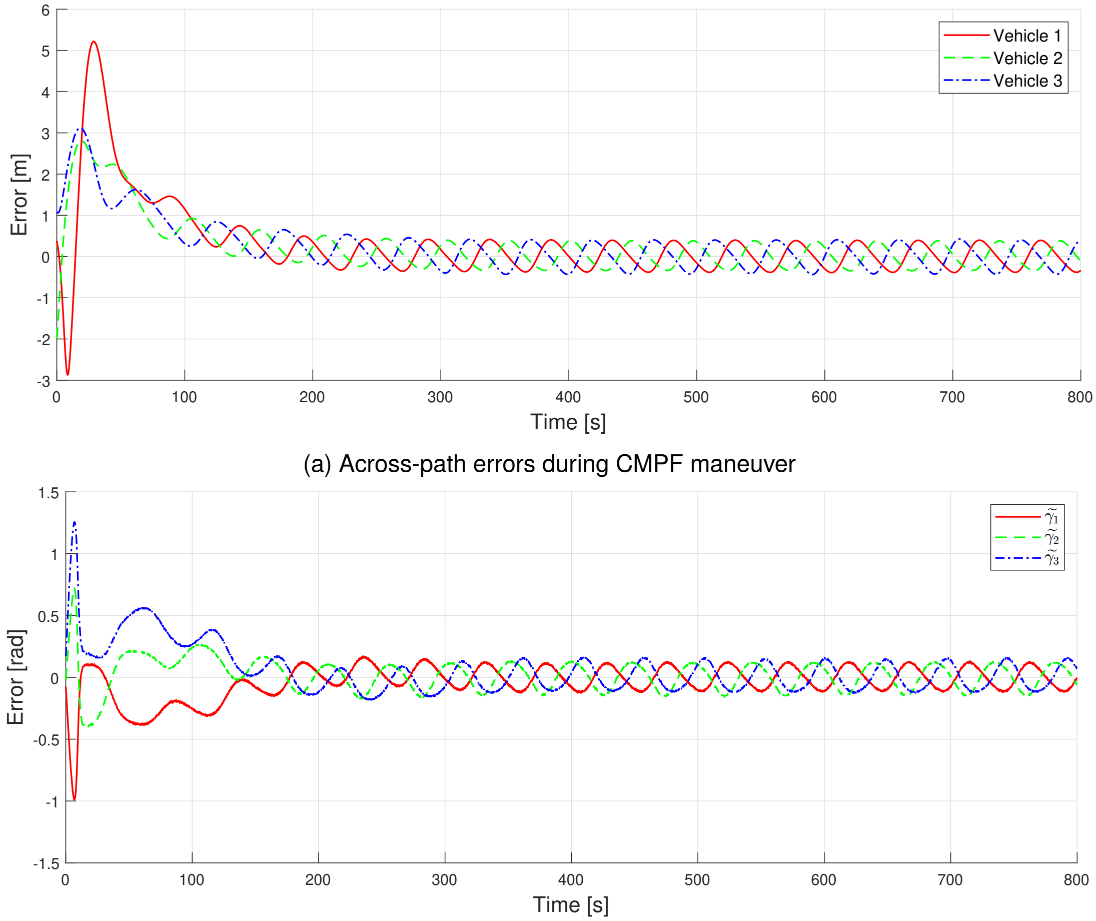


Figure 6.7: Vehicle velocity with and without path rotation

Form the analysis of figure 6.7, it becomes clear the difference in the mean speed for both simulations. For the non-rotating path, represented by a solid line, the speed oscillations are centered at $0.6\mathrm{m\,s^{-1}}$. For the rotating path, represented by a dashed line, the speed oscillations are centered at $0.7\mathrm{m\,s^{-1}}$, which corresponds to an increase of $0.1\mathrm{m\,s^{-1}}$. The introduction on an angular velocity of $\omega_P = 0.02\,\mathrm{rad\,s^{-1}}$ on a circumference with $R = 5\mathrm{m}$ causes a local linear velocity increase of $0.1\mathrm{m\,s^{-1}}$, equal to $R \times \omega_P$. The overall behavior of the speed is not modified with the introduction of a non-zero angular velocity. Nevertheless, the increase in the mean vehicle's speed causes the vehicles to perform a complete turn around the circumference in less time. Thus, the frequency of the oscillations in the vehicle's speed is slightly higher with the introduction of a positive angular velocity. If the angular velocity is contrary to the vehicle's navigation direction then, the vehicle's mean speed would decrease (with respect to the non-rotating situation) and the frequency on the oscillation would decrease.

The next step is to verify the behavior of the MEDUSA class vehicles using the control law from (6.9). The simulation conditions are not modified and we consider a circular path, with radius $R = 5\mathrm{m}$, attached to a target moving with a linear velocity of $\mathbf{v}_P = [0.05;\, 0.05]\,\mathrm{m\,s^{-1}}$. The angular velocity of the path was kept at $\omega_P = 0.02\,\mathrm{rad\,s^{-1}}$. The results are represented in figure 6.8.



(a) Across-path errors during CMPF maneuver

(b) Coordination error: difference between synchronization parameters

Figure 6.8: Simulation results for a circular moving and translating path using a MEDUSA model

The results from 6.5 can be directly compared with the results of 6.8. The simulations were conducted in the same conditions, except for the angular velocity that was only introduced in the latter. The origin of the cross-track error oscillation has been explained above. The convergence region of the cross-track error is not modified with the introduction of an angular velocity of $\omega_P = 0.02\mathrm{rad\,s^{-1}}$. Even though, these results are not valid for every value of ω_P, i.e., the convergence region of the cross-track error obtained without path angular velocity is not always the same as the one obtained when the simulation conditions are the same, but an arbitrary angular velocity is introduced. In the control law, there is a term, $\omega_{VP} \times {}^{VP}\mathbf{p}$, proportional to the path radius and angular velocity. This term adds a velocity vector that is tangent to the vehicle's projection on the path to the other velocity terms. If the angular velocity is too high, the vehicles are not capable of achieving the required linear velocity.

To allow for a better understanding of the impact of ω_P on the convergence region of the cross-track error, we conducted a series of simulations covering a number of values for this parameter. For the path chosen, the convergence region of the cross-track error is symmetric. In figure 6.9 is represented the absolute value of the bounds of the convergence region.

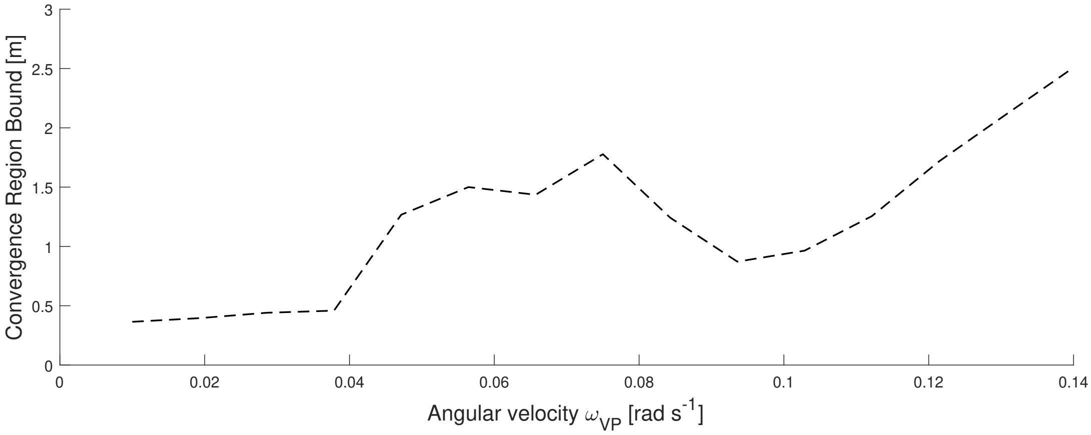


Figure 6.9: Cross-track error convergence region bound for different angular velocities

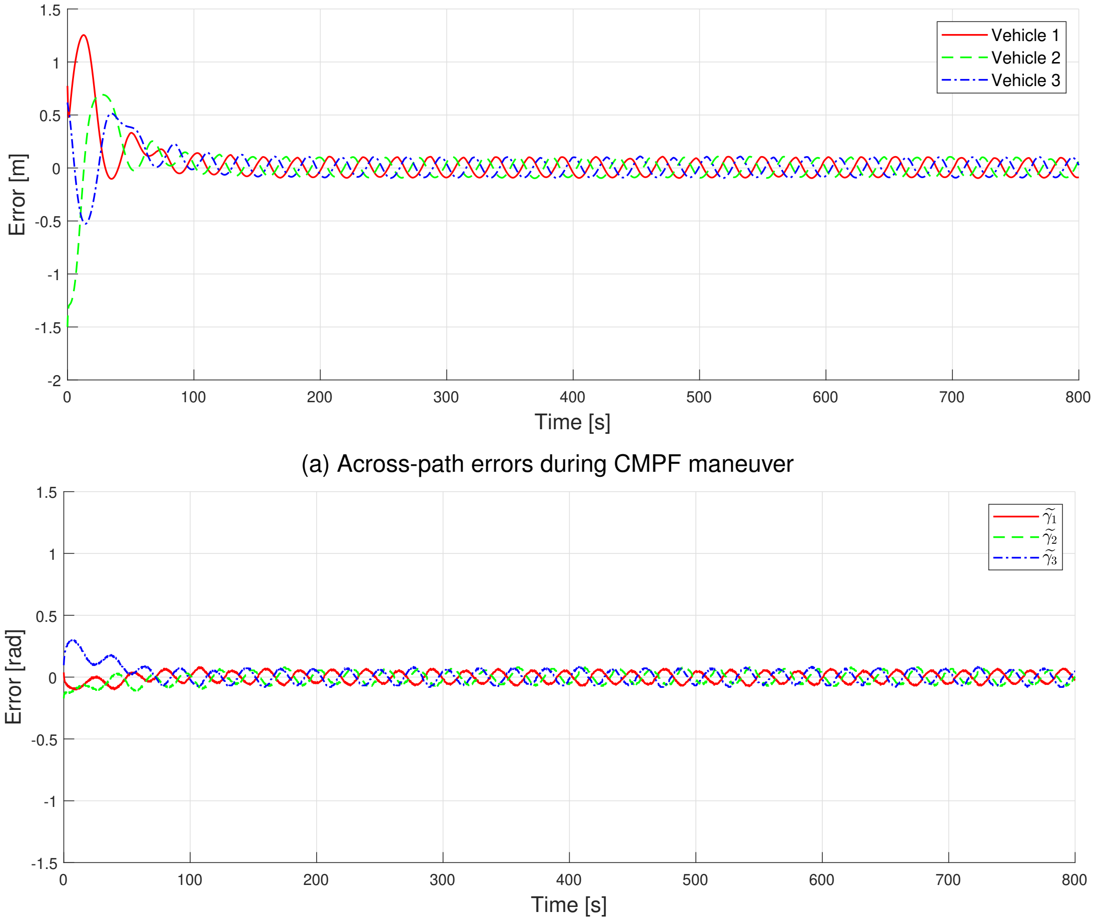
From the analysis of figure 6.9, we conclude that, in general, increasing the path angular velocity increases the bound of the cross-track error convergence region. For values of ω_P higher than 0.08rad s^{-1}, the bounds of the convergence region decrease. This decrease is related with a resonance that occurs in the path for such angular velocities. For an angular velocity higher than 0.1rad s^{-1}, the bound of the convergence region has a significant increase. This increase is related with the fact that this angular velocity causes the total velocity vector to reach amplitudes higher than the maximum speeds the vehicle is capable to achieve. Thus, the vehicle's behavior gets unpredictable, and the model obtained for the vehicles is no longer valid.

6.6.2.2 Non-Constant Curvature Paths

To evaluate the performance of the control law for a non-constant curvature path, we consider an elliptical path with axis lengths of 4.5m and 5.5m. P. Maurya's guidance law was designed for straight paths. For a non-constant curvature path, the convergence of the cross-track error to the origin is not guaranteed.

To have a means of comparison, we start by simulating the unicycle model with a static path (without linear and angular velocity). For this scenario, the cross-track error converged to the interval $[-0.05, 0.05]$m. The bounds of the convergence region are low, as the vehicles are able to respond almost instantaneously to the control inputs from the guidance law. Then, we introduce a linear velocity of $\mathbf{v}_P = [0.05; 0.05]$ m s^{-1} and an angular velocity of $\omega_P = 0.02$ rad s^{-1}. The results obtained are represented in figure 6.10.

In both figures, 6.10a and 6.10b, it is possible to see an oscillation in the results. The use of P. Maurya's guidance law for a non-constant curvature path is responsible for the oscillation on the cross-track



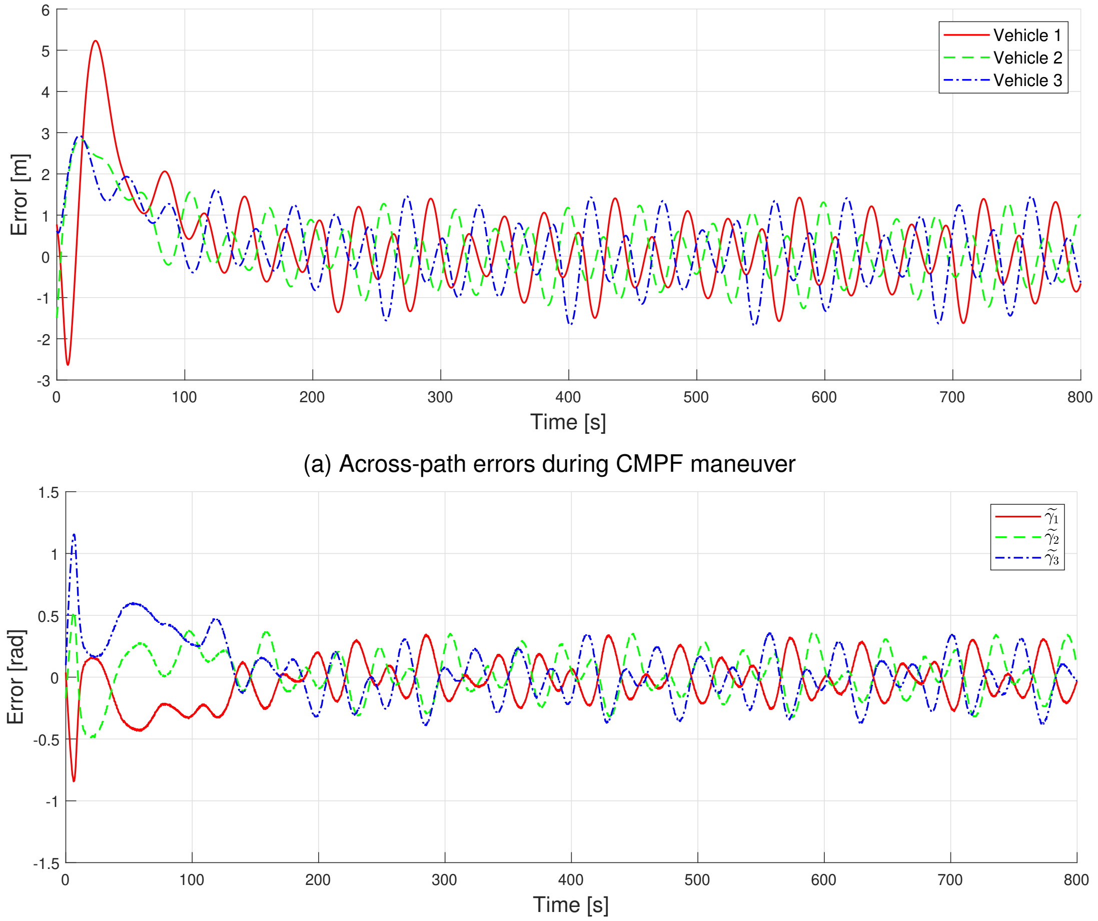
(a) Across-path errors during CMPF maneuver

(b) Coordination error: difference between synchronization parameters

Figure 6.10: Simulation results for an ellipsoid moving and translating path using a simplified model

error, represented in 6.10a, inside the region $[-0.05, 0.05]$m. Once again, the bound on the convergence region is not affected by the introduction of motion in the target. These results prove that the use of the control law from (6.9) provides the appropriate compensation to take into consideration the target motion. The oscillation in the coordination error, 6.10b, is caused by the fact that the cross-track error does not converge to the origin.

The next step is to evaluate the performance of the MEDUSA vehicles guided by (6.9). Once again, we start by simulating a non-moving target. The results obtained showed a convergence region delimited by $[-0.9, 0.9]$m. This region is considerably larger than the one obtained in the same simulation conditions for the unicycle model. We are simulating a model that takes into consideration both kinematics and dynamics. Thus, the vehicles have a finite yaw rate and acceleration as well as a delay between input and actuation, as seen previously. We will use these results as a basis for later comparisons. To see how the vehicles behave when this path is moving with both linear and angular velocity we introduce $\mathbf{v}_P = [0.05; 0.05]\,\mathrm{m\,s^{-1}}$ and $\omega_P = 0.02\mathrm{rad\,s^{-1}}$. The results obtained are represented in figure 6.11.



(a) Across-path errors during CMPF maneuver

(b) Coordination error: difference between synchronization parameters

Figure 6.11: Simulation results for an elliptical moving and translating path using the MEDUSA model

Analyzing figure 6.11a, it is possible to notice that, contrarily to what occurs for circular paths, the variations in the cross-track error do not have a sinusoidal-like shape. This behavior occurs due to the fact that the path curvature is not constant. The variations in the curvature of the path cause the cross-track error to have different amplitudes and variations along the path. The bounds of the convergence region of the cross-track error are considerably higher than the ones obtained for the stationary path. The convergence region is roughly symmetric around the origin and is bounded by $[-1.75, 1.75]$m. To achieve the equally spaced formation along an elliptical path the vehicles' velocity differ along the path. Thus, even for a stationary path, the speeds would not converge to a steady-state value. The coordination error reflects the increase in the cross-track error as it reaches values of approximately $\pm 0.4\,\mathrm{rad\,s^{-1}}$. This error is considerably high as it corresponds to more than $20°$.

To better analyze the impact of the linear and angular velocities on the convergence region of the cross-track error for a non-constant curvature path, a series of simulations were conducted. The results are represented in table 6.3.

Analyzing the results presented in table 6.3, we observe that for the elliptical path, regardless of the simulation conditions, the bounds on the convergence region of the cross-track error are considerably high. The target linear velocity has a direct relationship with the convergence region of the cross-track error. As for the angular velocity, it also impacts directly the convergence region. However, for higher target linear velocities, the impact of the angular velocity is less noticeable. The reduction on the impact of ω_P on the convergence region is related to the fact that the bounds of the convergence region are already high, which means that the vehicles are already performing the maneuver with less precision.

Simulation Conditions		**Cross-Track Error Convergence Region** $[\mathrm{m}]$	
$\omega_P[\mathrm{rad\,s^{-1}}]$	$\mathbf{v}_P[\mathrm{m\,s^{-1}}]$	Lower bound	Upper bound
0.00	$[0.00; 0.00]$	-0.90	0.90
0.00	$[0.05; 0.05]$	-1.00	1.00
0.02	$[0.05; 0.05]$	-1.75	1.50
0.02	$[0.10; 0.10]$	-2.30	2.00
0.02	$[0.15; 0.15]$	-2.80	2.60
0.06	$[0.05; 0.05]$	-1.75	1.50
0.10	$[0.05; 0.05]$	-1.75	1.85

Table 6.3: Cross-track error convergence region for an elliptical path with different linear and angular velocities

From all the results obtained, we conclude that for a circular path, the guidance law developed presents outstanding results as it ensures the right level of precision and coordination over the maneuver. For a non-constant path, the results are less promising. However, in the practical scenario presented, the path intended for the vehicles is a circular path. Thus, we conclude that the control law developed presents very satisfactory results.

6.6.3 Virtual Reference Frame Initialization

In section 6.5 we presented an alternative way to formulate the problem using a reference frame fixed on the virtual target's mass center. Computationally, the simplest approach consists of initializing the virtual target at the start of the maneuver. However, this approach is not the most advantageous as it requires more time for the vehicles to reach a neighborhood of the target. In the results from 6.5, where we initialized the virtual target at the start of the simulation, the vehicles required approximately $200\mathrm{s}$ to reach the steady-state behavior, considering an initial distance of approximately $10\mathrm{m}$. The other approach consists of providing, temporarily, as path, a straight line that connects the formation mass center with the target. Once the formation reaches a neighborhood of the target, we initialize the virtual target and the vehicles start performing CMPF with respect to $\mathcal{VP}$.

To compare the two different approaches, we conducted a simulation considering a large initial distance between the vehicles and the target and each one of the approach methods. The threshold that defines the tasks performed by the vehicles was set to 7m. The results are represented in figure 6.12. From the analysis of figure 6.12 we conclude that initializing the virtual target right at the beginning of the simulation caused the vehicle to navigate far from the target during 1000s. The cross-track error in relation to the path $\mathcal{P}$ oscillates while it decreases to the steady state. It is important to notice that the time necessary for the virtual target to reach the target can be adjusted by tuning the value of K. Nevertheless, if we set a high value for this parameter, the virtual path $\mathcal{VP}$ starts moving with a linear velocity at which the vehicles are not capable of navigating while following the path. Thus, the best approach is to initialize the virtual target only after the vehicles reach the neighborhood of the target and to make them follow the straight line that leads to the target otherwise. The use of this improved strategy resulted in a much smaller period of time where the vehicles were far from the target. In fact, it only took about 150s for the vehicles to reach the target. The results lead to a clear choice among the strategies. It is crucial to notice that the steady-state results are the same, as in the end the vehicles will be following $\mathcal{VP}$ that is converging to $\mathcal{P}$.

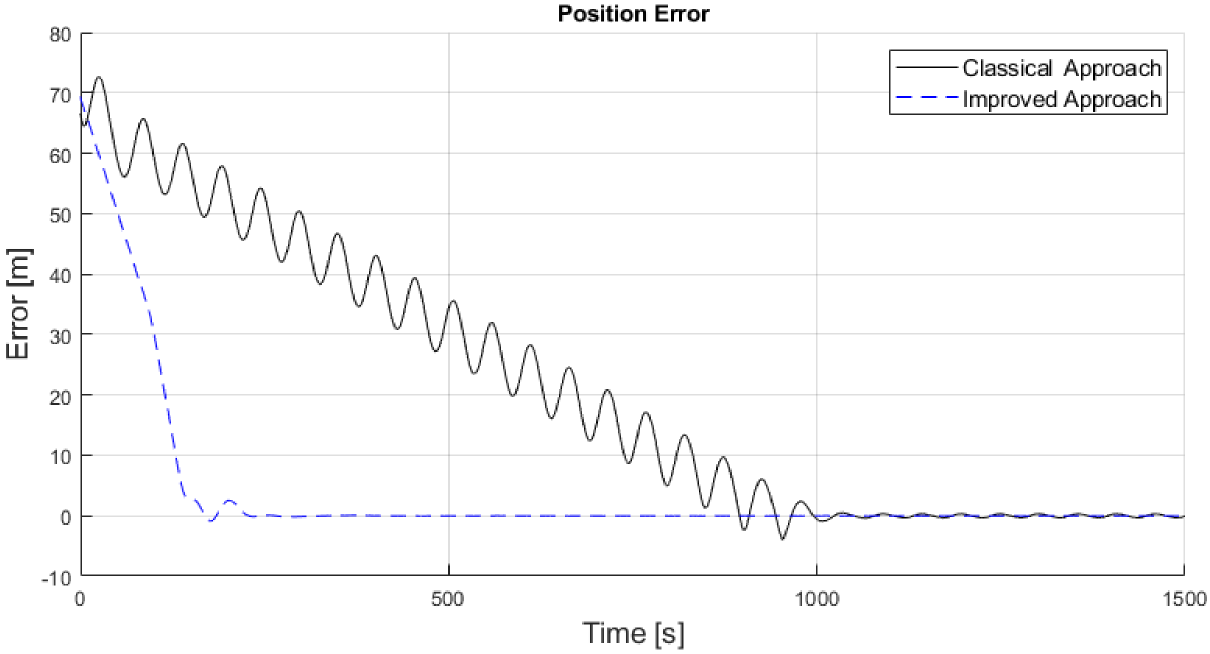
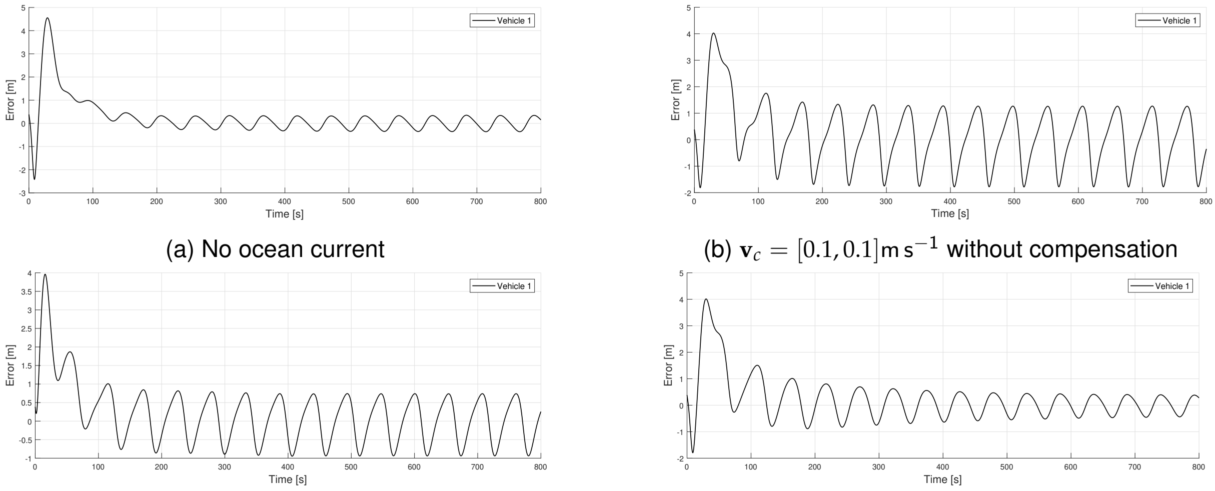


Figure 6.12: Cross-track error in relation to $\mathcal{P}$ using classic and improved path approaches

6.6.4 Robustness to External Disturbances

In 6.4 we introduced the compensation for external disturbances such as ocean currents. A term corresponding to the ocean current velocity estimate was subtracted from the global guidance law so that the total velocity vector would be aligned with the desired course angle. To evaluate the results, we performed several simulations considering different ocean currents. The results obtained are represented in figure 6.13.



(a) No ocean current

(b) $\mathbf{v}_c = [0.1, 0.1]\,\mathrm{m\,s^{-1}}$ without compensation

(c) $\mathbf{v}_c = [0.1, 0]\,\mathrm{m\,s^{-1}}$ with compensation

(d) $\mathbf{v}_c = [0.1, 0.1]\,\mathrm{m\,s^{-1}}$ with compensation

Figure 6.13: Position error obtained for different scenarios and current velocity vectors

To have a means of comparison we started by simulating a scenario with no ocean currents. The results are represented in 6.13a. At steady state, the cross-track error converged to the region $[-0.4, 0.4]$m. Throughout the remaining simulations all the parameter involved in the control algorithms remained unmodified. The only differences lie in the introduction of ocean currents and the respective compensation in the global guidance law. Figure 6.13b shows the cross-track error obtained considering a non-null ocean current misaligned with the target velocity vector and without compensation. As expected, the cross-track error converged to a region larger than the original one, delimited by $[-1.8, 1.3]$m. Figure 6.13c shows the cross-track error obtained by considering a weaker ocean current without proper compensation. The cross-track error converges to a region larger than the original, but not as large as the region obtained in 6.13b. Finally, figure 6.13d shows the cross-track error obtained when the proper current compensation is added. The values of the estimates are initialized at the origin and only after some time are they accurate. For this reason, it is possible to see the cross-track error bounds converging to the region originally obtained. As the values of the estimates become more accurate, the total velocity vector gets aligned with the desired course vector, which means that the correct compensation is provided.

6.6.5 Target Velocity Adjustment

Considering the possible application for the control laws designed, in a scenario where a vehicle malfunction occurs it is necessary to reduce the target's velocity so that the vehicles affected have a higher chance of resuming the maneuver. In this situation, the cross-track error of the vehicle affected by the malfunction increases. Once the central vehicle of the surface formation receives the information, it transmits the information to the underwater vehicle. To this end, we define the formation error as the maximum value of the surface vehicles' cross-track error

$$\textit{formation cross-track error} \; = \max(|\textit{cross-track error}_i|), \tag{6.13}$$

where $\max(\cdot)$ denotes the maximum value and $|\cdot|$ denotes the absolute value. From the previous equations, we conclude that if one of the vehicles has its cross-track error increased the formation cross-track error will also increase. At the leading vehicle on the surface, the auxiliary algorithm defined in 1 is running.

Algorithm 1 Target velocity controller

```
function VELOCITY ADJUST($formation\ cross\text{-}track\ error$, $current\ speed$, $nominal\ speed$)
    if $formation\ cross\text{-}track\ error < Threshold_2$ and $formation\ cross\text{-}track\ error > Threshold_1$ then
        $target\ speed = current\ speed \times (1 - decresase\ rate)$.
    else
        $target\ speed = nominal\ speed$.
    if $target\ speed < Treshold_3$ then
        $target\ speed = Treshold_3$
    return $target\ speed$.
```

We define a region for which the algorithm is executed. by doing this, the algorithm will not reduce the target speed during the initial phase of the target approach. The actuation zone is limited by $[Threshold_1, Threshold_2]$. A decrease rate is imposed, causing the velocity to be successively reduced until the cross-track error reaches a value outside the actuation zone. Due to the limited communication rate between the surface and the underwater formation, the information is only broadcasted at fixed time intervals. To prevent the target from completely stop or even reaching negative velocity references, we define a lower threshold for its velocity, given by $Threshold_3$. To further improve the actuation on the target velocity, we use an hysteresis to prevent the controller from being turned on and off successively. To evaluate the performance of the algorithm designed, we conducted a simulation where an error in vehicle 1 occurs, causing the total velocity vector to not have the desired orientation. The results obtained without the use of the control algorithm are represented in 6.14.

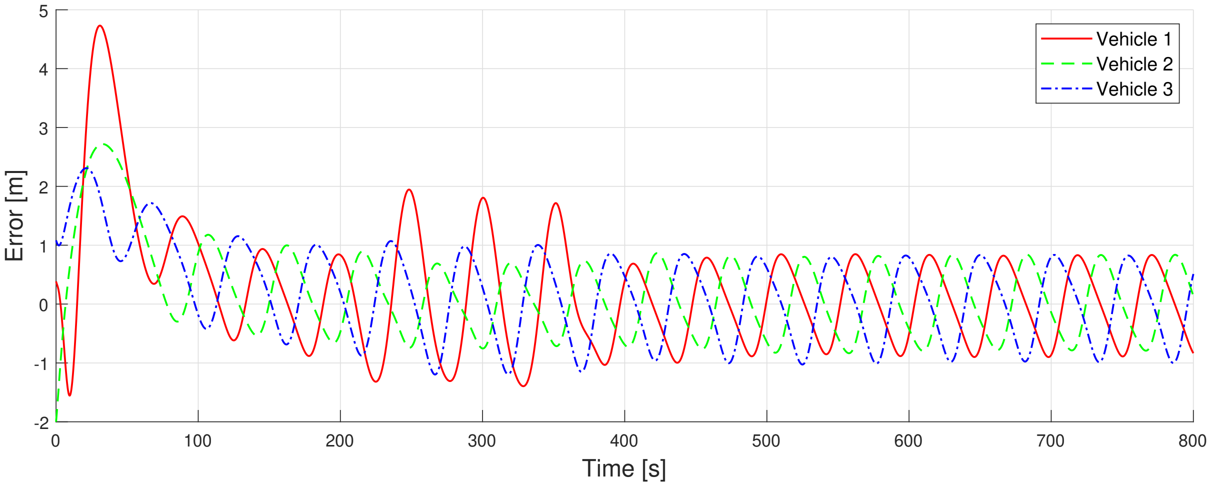
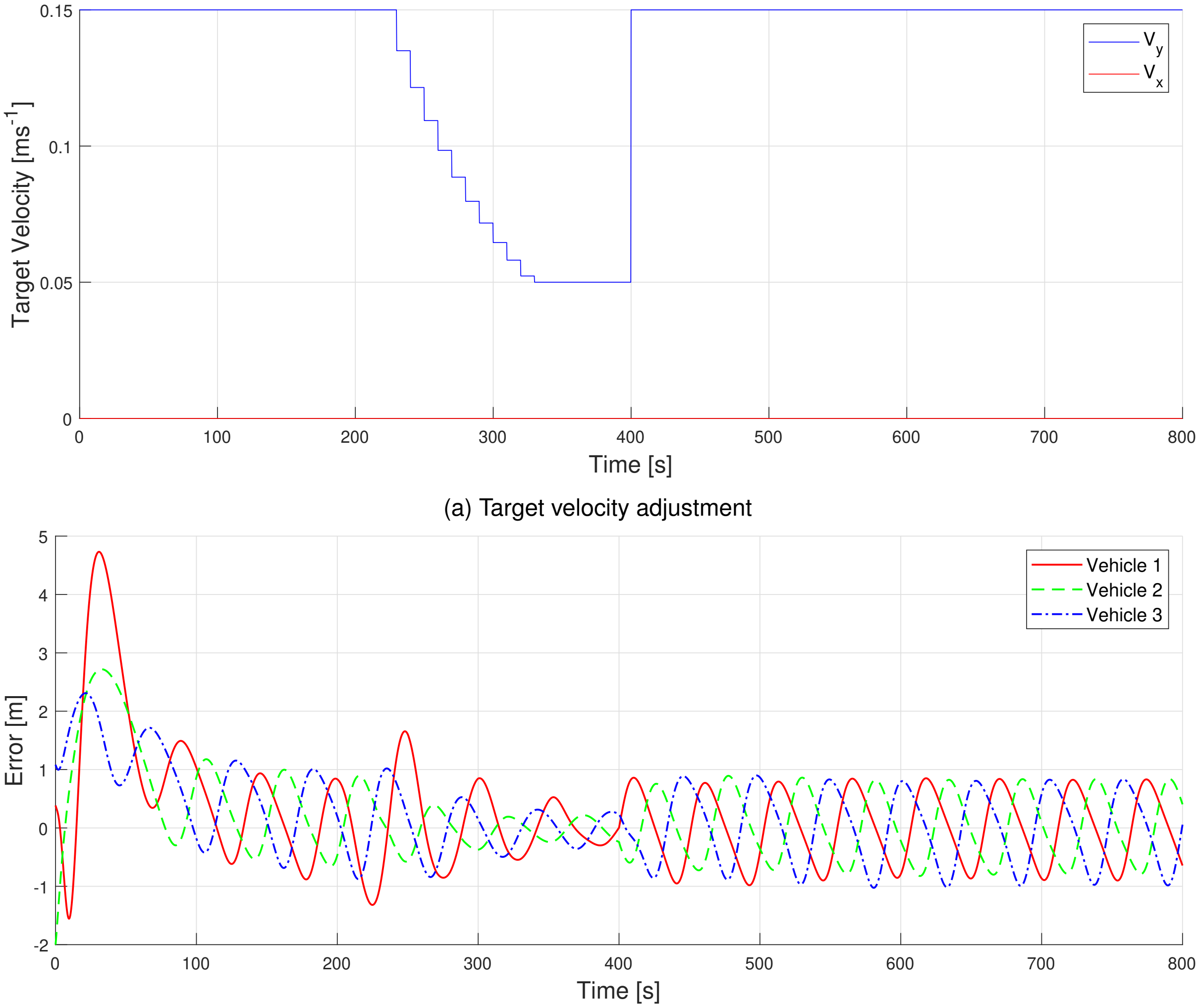


Figure 6.14: Cross-track error without target velocity compensation

A malfunctioning occurred in the time interval $[200, 350]$s, which caused a significant increase in the cross-track error of vehicle one. During the malfunctioned period the local guidance law, computed by vehicle one, is incorrectly transferred to the inertial reference frame. At the highest peak, the position error reached approximately 2 meters. To evaluate the controller performance, the same malfunction was simulated, but the target velocity is now adjusted by the control algorithm 1. The decreased rate was set to 10%, $Threshold_1$ and $Treshold_2$ were set to 1.2m and 2.0m, respectively. The lower bound for the amplitude of the target velocity vector was set to $0.05\mathrm{m\,s^{-1}}$. The simulation results are represented in figure 6.15.



(a) Target velocity adjustment

(b) Cross-track error

Figure 6.15: Target velocity controller performance

Analyzing figure 6.15a it is possible to see that the target velocity was adjusted during the malfunctioning. At approximately 220s the absolute value of the error got higher than the $Threshold_1$, which turned on the controller. The target velocity was successively reduced at a rate of 10%. The decrease in the target velocity was sufficient for the vehicle to retake the maneuver. The last two peaks visible in 6.14 are not seen in 6.15b. In fact, the cross-track error reached values lower than the values obtained in steady-state, as the target linear velocity was $\mathbf{v}_P = [0.00; 0.05]$m s^{-1}. As mentioned, to prevent the controller from turning on and of repeatedly a hysteresis was used. The controller would reset the target velocity once the formation error got lower than 0.65m, which occurred at 400s. The parameters involved in the controller must be chosen according to the maneuver conditions as the path shape, radius, and the target velocity influence the region of convergence of the cross-track error. Nevertheless, the controller provides an efficient strategy to detect an anomaly during the maneuver. Instead of using a fixed correction term on the target velocity that depends on the formation cross-track error, we use a cumulative correction term, defined in percentage terms. The fact that the correction is cumulative allows successively correction on the target linear velocity. In scenarios where at least one of the vehicles is having difficulties in performing the maneuver, the target slows down, which gives vehicles easier to resume the maneuver.

6.6.6 Logic-Based Communications

The communications among vehicles can be reduced and optimized by using an event-triggered communication mechanism. Such mechanisms have already been studied in literature [16] [15]. Each vehicle runs an estimates itself and its neighborhoods. Instead of using the neighborhood coordination states, the vehicles use the estimates computed locally, using the estimator from (4.26).

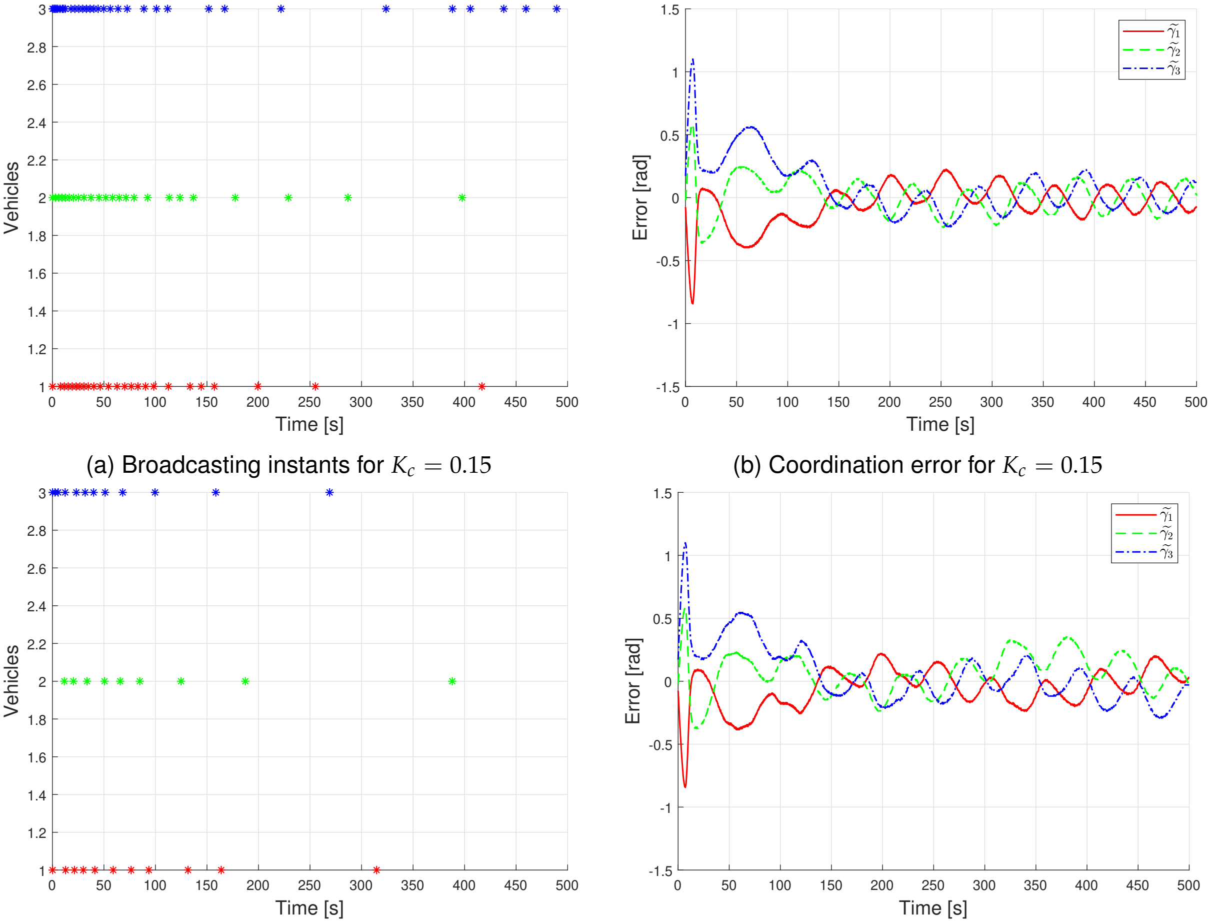
The triggering conditions dictate the instants where communications take place. Let $\tilde{\gamma}_i = |\gamma_i - \hat{\gamma}_i|$ be the estimation error, i.e., the absolute values of the difference between the real synchronization parameter of an arbitrary vehicle and its estimate of its synchronization parameter. It remains defining the triggering condition for each vehicle to send its synchronization parameter to its neighboring vehicles. The logic function that dictates when should a vehicle broadcast its state compares the estimation error with a defined threshold, is given by

$$|\tilde{\gamma}_i| = |\gamma_i - \hat{\gamma}_i| \geq k_c. \tag{6.14}$$

where k_c is the estimation error threshold. The key idea is to run synchronized estimators, i.e., at every broadcast instant of vehicle i, vehicle i own estimate is reset and, for every vehicle j for which there exists a communication link with vehicle i its estimate of vehicle i state is also reset. In the results obtained in [16] the author proved that the coordination error is bounded, by assuming that communication delays are negligible and that the normalized desired speed is common for all vehicles.

The constant k_c plays an important role in the mechanism and in the performance of the vehicles. We conducted several simulations to analyze the performance of the control structure. We considered a circular path with radius $R = 5$m centered on a target moving with a linear velocity of $\mathbf{V}_P = [0.05; 0.05]\,\mathrm{m\,s^{-1}}$. The results obtained are represented in figure 6.16.

In the results presented in 6.16 it is possible to see that the coordination error does not converge to the origin. The oscillation nature has been analyzed above. In 6.16a and 6.16c it is possible to see that during the first instant of simulation there exist more broadcasting instant as the vehicles are still converging to the path an to the desired formation. Once the vehicles synchronize their positions, the interval between broadcasts increases. In the simulation with a higher k_c it is possible to see a lower number of broadcasts in the second half of the simulations. These findings were expected, as a higher estimation error is allowed. However, it is possible to see in 6.16b and 6.16d, that setting a higher value for the threshold causes the coordination error to reach higher values. The higher the interval between communications, the lower the precision of the estimations. Thus, the vehicles are using inaccurate values to adjust their velocities. The bounds on the coordination error for the simulation with $k_c = 0.15$ are the same as those obtained above, 6.5b when there existed continuous communications, which is a very promising result, as the communication was drastically reduced with the introduction of the event-triggered mechanism.



(a) Broadcasting instants for $K_c = 0.15$

(b) Coordination error for $K_c = 0.15$

(c) Broadcasting instants for $K_c = 0.30$

(d) Coordination error for $K_c = 0.15$

Figure 6.16: Position error obtained for different scenarios and current velocity vectors

CHAPTER 7

Graphics Software

This chapter presents a program, built using the software *Unity* [36] that gathers the data from the vehicle's simulation (running in *Simulink*) and displays, in real time, a three-dimensional environment. This tool allows for a clean visual perception of the vehicles' behavior during the simulation of a mission in the water.

7.1 Programs Interaction and Information Flow

The *UNITY* program runs in parallel with the simulation program, in *Simulink*, and is configured to read an odometry message published in ROS. A package from ROS is used, *rosbrige_suite*, that creates a websocket, allowing subscribing to a topic that is being published in the ROS network. For a general overview of ROS, the reader is referred to [32]. The ROS network can be described with the help of the following figure.

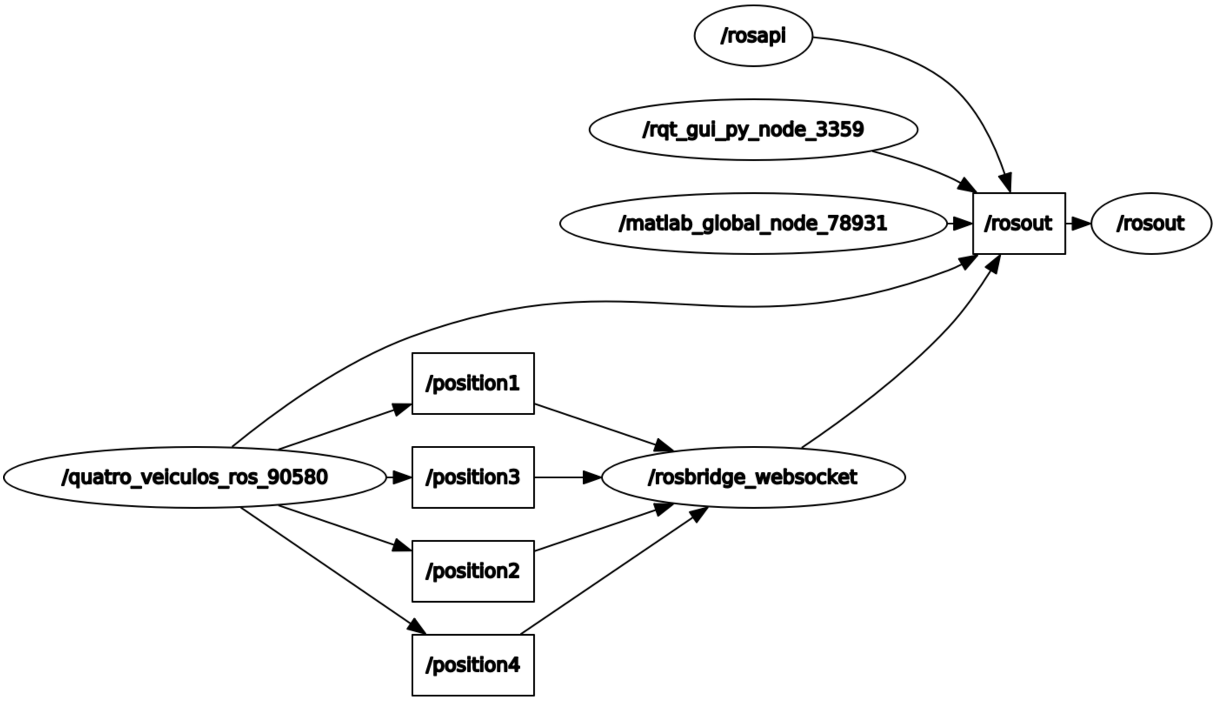


Figure 7.1: ROS network architecture

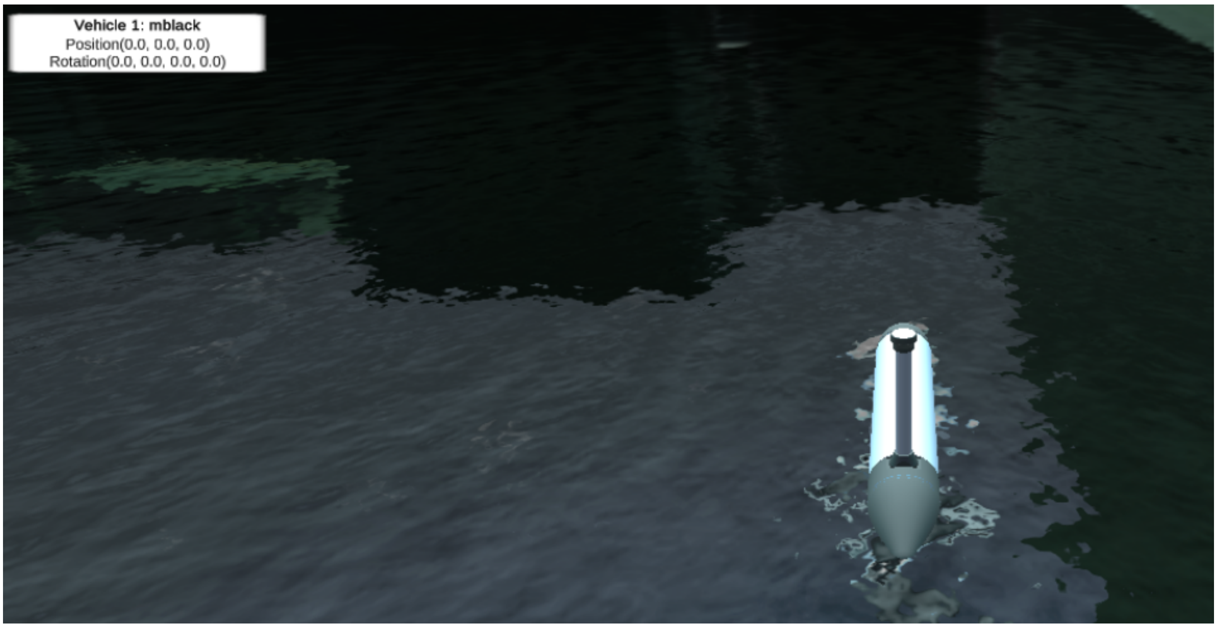
In figure 7.1, it is possible to see the main nodes and topics in the ROS network considered. Throughout this section we will consider a formation composed of four MEDUSA vehicles. One MEDUSA performs a PF maneuver, while the remaining three MEDUSA vehicles perform CMPF with respect to a circular path that in the limit will be centered at the origin of frame $\{P\}$ that is attached to the fourth vehicle center of mass. Each vehicle publishes a *ROS* odometry message denoted by ”/position”, followed by its respective number. The messages published by the vehicles are then subscribed by the websocket node, which publishes the messages in a specific format, the JSON format. The messages in the latter format are then subscribed by node ”/rqt_gui_py_node_2625”, which represents the graphics software.

7.2 Simulation Results

Throughout the thesis we have represented the vehicles' orientation using Euler angles. In *Unity* however, the orientation is expressed in terms of quaternions, defined by a 4-dimensional vector q. This representation is less intuitive than the Euler representation. Mathematically it is possible to convert between representations. The conversion from Euler to quaternion gets even simpler when the vehicles operate in the xy plane, meaning that both θ and ψ are zero. The conversion for this particular case is defined as

$$\begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{bmatrix} = \begin{bmatrix} \cos(\psi/2) \\ 0 \\ 0 \\ \sin(\psi/2) \end{bmatrix}. \tag{7.1}$$

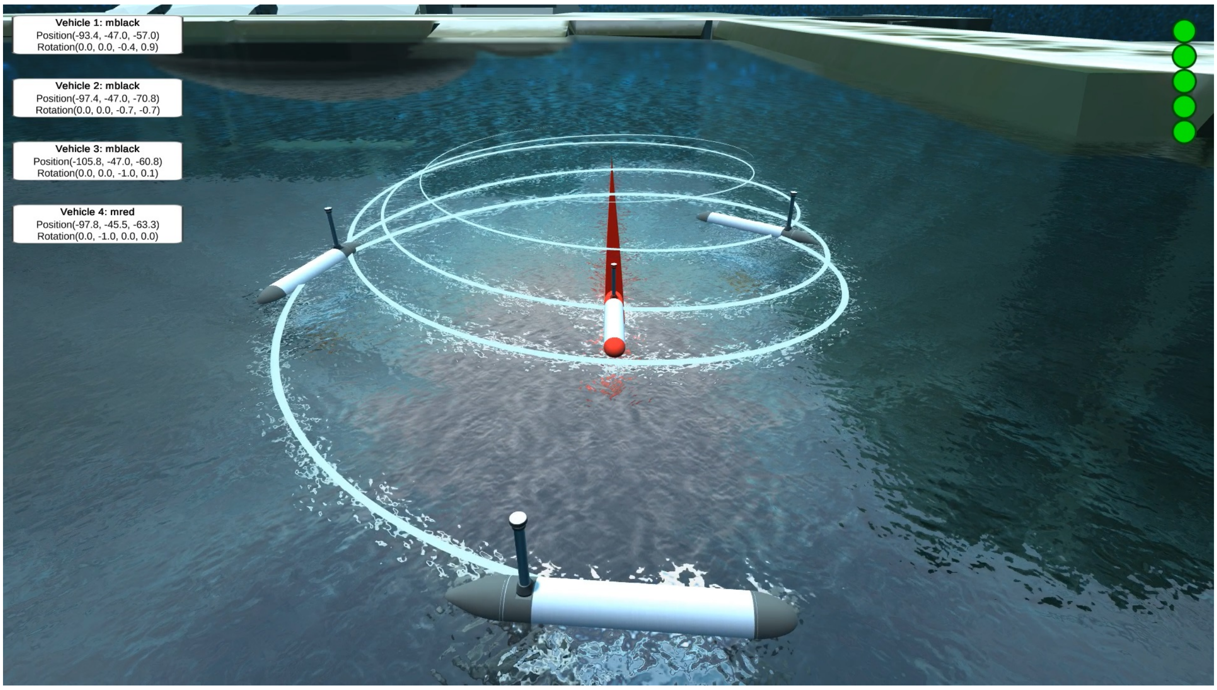
In the graphic software, the user will have the opportunity to see in real time, the position and orientation of the vehicle, as well as the vehicle's trajectory. Figure 7.2 is a snapshot of the computer's screen during simulation of a single vehicle.



Vehicle 1: mblack
Position(0.0, 0.0, 0.0)
Rotation(0.0, 0.0, 0.0, 0.0)

Figure 7.2: Single vehicle representation using UNITY graphic software

In the simulation, the vehicle's trajectories are represented by colored lines. The line, known as trail, can be adjusted, by specifying the desired color in *RGB* format and its "fading" rate. Represented in figures 7.3 and 7.4, are some screenshots obtained using the UNITY program, considering one MEDUSA vehicle performing PF in a lawnmowing maneuver and three MDEUSA vehicles performing CMPF with respect to the a circular path expressed in a frame fixed to the vehicle referred before.



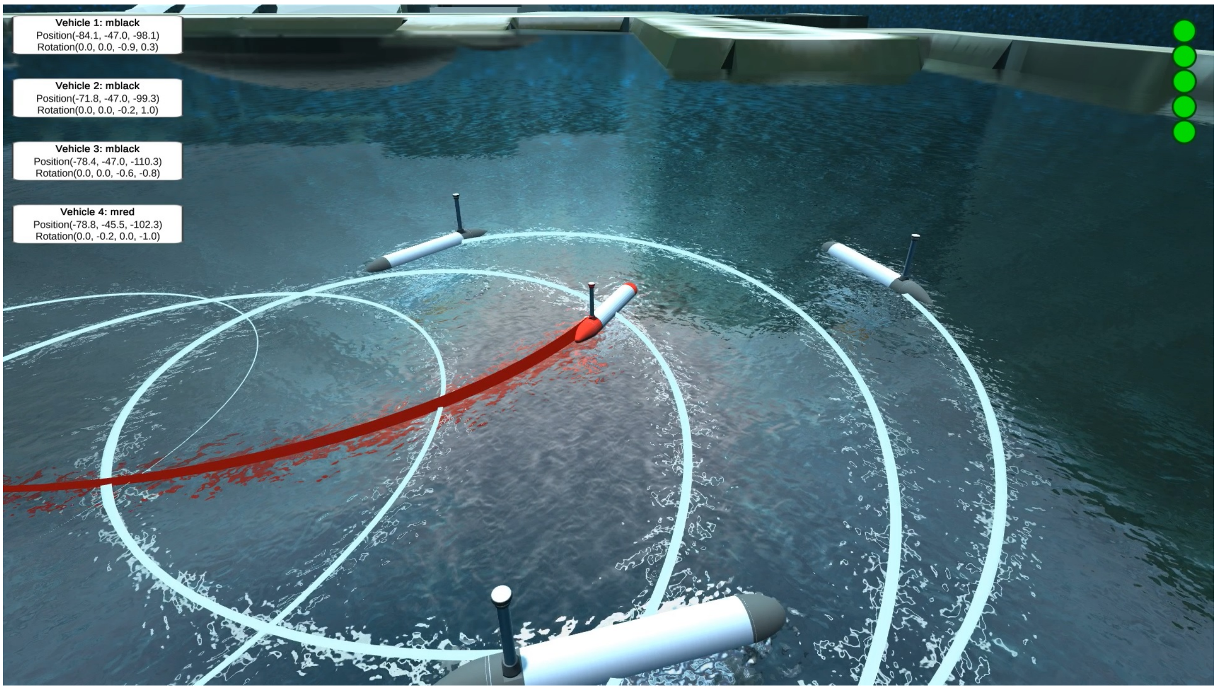
Vehicle 1: mblack
Position(-93.4, -47.0, -57.0)
Rotation(0.0, 0.0, -0.4, 0.9)

Vehicle 2: mblack
Position(-97.4, -47.0, -70.8)
Rotation(0.0, 0.0, -0.7, -0.7)

Vehicle 3: mblack
Position(-105.8, -47.0, -60.8)
Rotation(0.0, 0.0, -1.0, 0.1)

Vehicle 4: mred
Position(-97.8, -45.5, -63.3)
Rotation(0.0, -1.0, 0.0, 0.0)

Figure 7.3: UNITY graphic software representation of four vehicles performing a cooperative mission



Vehicle 1: mblack
Position(-84.1, -47.0, -98.1)
Rotation(0.0, 0.0, -0.9, 0.3)

Vehicle 2: mblack
Position(-71.8, -47.0, -99.3)
Rotation(0.0, 0.0, -0.2, 1.0)

Vehicle 3: mblack
Position(-78.4, -47.0, -110.3)
Rotation(0.0, 0.0, -0.6, -0.8)

Vehicle 4: mred
Position(-78.8, -45.5, -102.3)
Rotation(0.0, -0.2, 0.0, -1.0)

Figure 7.4: Three vehicles performing a circular motion centered at a moving target in red: representation using the UNITY graphic software

The visualization of the vehicles' trajectories is very useful as it transmits a general overview of the vehicles' behavior, otherwise we would only see their current positions. The vehicles' positions, expressed in the inertial reference frame, are also presented and updated in real time, so that the user has access to the exact information about the vehicles' status. In figures 7.3 and 7.4 it is possible to see four green circles at the top right corner. The top circle corresponds to the correct functioning of the program. The remaining four circles correspond to the four vehicles included in the simulation and. The green color means that the websocket was successfully created and is receiving the four odometry messages from the *MATLAB* node.

CHAPTER 8

Conclusion

8.1 Summary

MOTIVATED by real mission scenarios, this thesis focused on the cooperative motion control of multiple autonomous marine vehicles. We started by presenting a generic vehicle model for a marine vehicle moving in a three-dimensional space, thus requiring a six DOF description. After this, and considering this thesis context, we made some simplifications to the model, which allowed to obtain a vehicle model with three DOF. This model served as a basis for the control work made throughout this thesis.

We started by examining the problem of cooperative path following. For its basic building block, the path following controller is composed of an inner-outer loop control architecture. Among other advantages, this strategy allows decoupling the design of the kinematic and dynamic loops. For the inner loop we designed an LQR controller, properly modified to turn the feedback solution into a servomechanism to make the vehicle track a heading reference. For the outer loop, responsible for generating heading references, we analyzed two distinct guidance laws. Due to the results obtained throughout a series of simulations, where we analyzed the performance of each law considering paths formed by straight lines, constant curvatures, and the presence of external disturbances we opted to use P. Maurya's guidance law for the latter work.

Once we were able to control the motion of a single vehicle, the next step was to control the motion of multiple vehicles. We used the controllers previously developed to individually control the motion of each vehicle and introduced an external controller to ensure that the vehicles would converge to a defined geometric pattern. We adopted a distributed control architecture where each vehicle makes its own decisions based on the interaction with its neighbouring vehicles. The idea behind the coordination controller is to adjust the speed of each vehicle so that the ensemble will converge to the desired formation. A correct parametrization of the paths is essential to ensure that the vehicles will reach and then maintain the geometric formation. The communication network was an aspect that the controller should take into consideration, since in practical missions, each vehicle can only communicate with a small set of vehicles. To this end, we borrowed tools from graph theory and represented the communication network using an undirected graph. Both scenarios of continuous and discrete communications were analyzed. In the latter, we developed an estimator for the coordination parameter that each vehicle should run in order to estimate its own and its neighbor's status.

After this initial phase, we focused on the target encircling and moving-path following maneuvers. We started by analyzing a strategy based on trajectory tracking that allows a set of agents to track and encircle a moving target. The adopted strategy only requires the knowledge of the vehicles' relative position with respect to the target. A decentralized control structure was implemented, where each vehicle would run an independent exosystem to generate relative reference positions. A potential function that introduces a repulsion force between each agent of the exosystem was used. The potential function was introduced to ensure that the vehicles would converge to the uniform pattern, i.e., they would be equally spaced over the encircling maneuver. A modification on the control architecture was proposed to replace the exosystem by a block that only requires trigonometric concepts to generate relative reference positions. Both in the original and the modified strategy there exists no feedback from the real vehicles, but only from the values generated by the exosystems. The simulations results are very promising, as the vehicles have a high performance and are capable of encircle the target with a good accuracy. Some simulations were also conducted by inserting a feedback from the vehicles' state in the exosystem from the original strategy. The results showed that the vehicles required more time to converge to the encircling maneuver and consequently to the desired formation. At steady-state the results showed no differences from the original strategy. Thus, the results are once more promising as the coordination was achieved by using the vehicles' state. However, we must stress that the use of the real vehicles' variables has no theoretical support.

Lastly, we proposed an intuitive control law for moving-path following. The strategy requires the knowledge of the target's velocity. The path following problem is solved on the moving frame, and the correct compensation that takes into consideration the frame motion is provided. In simulation results, we were able to guarantee the convergence of the cross-track and coordination error to the origin for a set of unicycle agents. For the MEDUSA vehicles we were able to guarantee the convergence to a region centered at the origin. To synchronize the vehicles the concepts of path parametrization and the coordination controller previously developed were used. The robustness with respect to external disturbances was also analyzed. Using values provided by a complementary filter designed to estimate the ocean currents we were able to compensate for the existence of a constant ocean current. A strategy to detect and react in compliance with anomalies during the maneuver was developed. Monitoring the cross-track error during the maneuver allows for the detection of these anomalies. Once an anomaly is detected, the target velocity is decreased so that the vehicles have a higher chance of resuming the maneuver.

In summary, this thesis is a contribution to a general problem of cooperative motion control whereby a group of vehicles, called trackers, encircle a moving target while ensuring compliance with respect to temporary communication losses and vehicle malfunctions. The solution proposed is conceptually simple, and shows considerable promise for implementation in a real life scenario in the area of geotechnical surveying. A software tool was developed to afford systems designers the capability to visualize the missions being simulated.

8.2 Future Work

All the controllers developed throughout this work were implemented in *Simulink* and their performance was evaluated in simulation. However, real tests with the vehicles are the ultimate proof of good performance. During the simulations we tried to reproduce the scenario of an actual mission to the fullest. In spite of this, the true performance of the controllers can only be evaluated with the implementation of the algorithms in the robots and by conducting real tests. Thus, one of the next steps is the implementation of the developed software.

Throughout this thesis we considered that communications between the surface and underwater segments are done acoustically. However, if communications could be made using an optical channel, high transmission rates would be possible. However, there is an outstanding issue that stems from the fact that optical transmissions is still highly directional. For this reasons, considerable work must be done to enable two vehicles to communicate optically in the course of a mission. In fact, with current technology, the use of optical communications will require that if two vehicles decide to communicate, then both must have access to reasonably good estimates of their relative positions and point the optical beams accordingly, possibly by performing a local sweep so that their conical angles of communication will lock. The use of such mechanism, coupled with a system for relative positioning based on distributed estimation concepts would drastically improve the cooperative capabilities of the vehicles.

The detection of anomalies is a highly importance issue. To increase the autonomy of the vehicles and to allow for self-correction of occasional errors it is essential that the vehicles detect the occurrence of an anomaly and take actions to mitigate their effects. For this purpose, an algorithm that adjusts both the target's velocity and the nominal formation speed could be helpful.

The above are just representative examples of a mixture of theoretical and practical problems that must be addressed in the future. Clearly, distributed navigation and control for practical mission that include automated geotechnical surveys warrant substantial research and development effort.

Bibliography

[1] Cooperative Cognitive Control for Autonomous Underwater Vehicles. *Deliverable D 1.1. Requirements Analysis Report*. 2009. URL: `http://robotics.jacobs-university.de/projects/Co3-AUVs/publicdeliverables/D11-RequirementsAnalysis.pdf` (visited on 09/21/2019).

[2] Norman Biggs. *Algebraic Graph Theory*. second. Cambridge Mathematical Library. Cambridge University Press, Aug. 2012.

[3] L. Briñón-Arranz, A. Seuret, and C. Canudas-de-Wit. “Cooperative Control Design for Time-Varying Formations of Multi-Agent Systems”. In: *IEEE Transactions on Automatic Control* 59.8 (Aug. 2014), pp. 2283–2288. ISSN: 2334-3303. DOI: `10.1109/TAC.2014.2303213`.

[4] Lara Briñón-Arranz, Alexandre Seuret, and Antonio Pascoal. “Circular formation control for cooperative target tracking with limited information”. In: *Journal of the Franklin Institute* 356 (Mar. 2019), pp. 1771–1788. DOI: `10.1016/j.jfranklin.2018.12.011`.

[5] Arthur Bryson, Y.-C Ho, and George Siouris. “Applied Optimal Control: Optimization, Estimation, and Control”. In: *Systems, Man and Cybernetics, IEEE Transactions on* 9 (July 1979), pp. 366–367. DOI: `10.1109/TSMC.1979.4310229`.

[6] James Cameron. *The Mariana Trench*. Mar. 2012. URL: `http://www.deepseachallenge.com/the-expedition/mariana-trench` (visited on 11/23/2018).

[7] Michael Scott Cavers. “The Normalized Laplacian Matrix and General Randic Index of Graphs”. PhD thesis. University of Regina, July 2010.

[8] CORDIS. *WiMUST project*. Jan. 2015. URL: `https://cordis.europa.eu/project/rcn/194287_en.html` (visited on 11/23/2018).

[9] CORDIS. *GREX project ID: 035223*. Mar. 2016. URL: `https://cordis.europa.eu/project/rcn/79338/factsheet/en` (visited on 11/23/2018).

[10] DSOR. *MORPH*. Jan. 2012. URL: `http://dsor.isr.ist.utl.pt/projects/morph` (visited on 11/23/2018).

[11] T. I. Fossen, K. Y. Pettersen, and R. Galeazzi. “Line-of-Sight Path Following for Dubins Paths With Adaptive Sideslip Compensation of Drift Forces”. In: *IEEE Transactions on Control Systems Technology* 23.2 (July 2015), pp. 820–827. ISSN: 2374-0159. DOI: `10.1109/TCST.2014.2338354`.

[12] Thor I. Fossen, Morten Breivik, and Roger Skjetne. “Line-of-sight path following of underactuated marine craft”. In: *IFAC Proceedings Volumes* 36.21 (Sept. 2003). 6th IFAC Conference on Manoeuvring and Control of Marine Craft (MCMC 2003), Girona, Spain, 17-19 September, 1997, pp. 211–216. ISSN: 1474-6670. DOI: `https://doi.org/10.1016/S1474-6670(17)37809-6`. URL: `http://www.sciencedirect.com/science/article/pii/S1474667017378096`.

[13] Reza Ghabcheloo et al. “Coordinated Path-Following in the Presence of Communication Losses and Time Delays”. In: *SIAM J. Control and Optimization* 48 (Jan. 2009), pp. 234–265. DOI: `10.1137/060678993`.

[14] Andreas J. Hausler. *Cooperative Autonomous Robotics at Sea*. Sept. 2014. URL: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.717.2921&rep=rep1&type=pdf (visited on 09/21/2019).

[15] N. T. Hung, F. C. Rego, and A. M. Pascoal. "Event-Triggered Communications for the Synchronization of Nonlinear Multi Agent Systems on Weight-Balanced Digraphs". In: *2019 18th European Control Conference (ECC)*. IEEE, June 2019, pp. 2713–2718. DOI: 10.23919/ECC.2019.8796277.

[16] Nguyen T. Hung and António M. Pascoal. "Cooperative Path Following of Autonomous Vehicles with Model Predictive Control and Event Triggered Communications". In: *IFAC-PapersOnLine* 51.20 (2018). 6th IFAC Conference on Nonlinear Model Predictive Control NMPC 2018, pp. 562–567. ISSN: 2405-8963. DOI: https://doi.org/10.1016/j.ifacol.2018.11.031. URL: http://www.sciencedirect.com/science/article/pii/S2405896318326855.

[17] Giovanni Indiveri and João Gomes. "Geophysical Surveying with Marine Networked Mobile Robotic Systems: The WiMUST Project". In: *Proceedings of the International Conference on Underwater Networks & Systems*. WUWNET '14. Rome, Italy: ACM, 2014, 46:1–46:2. ISBN: 978-1-4503-3277-4. DOI: 10.1145/2671490.2677084. URL: http://doi.acm.org/10.1145/2671490.2677084.

[18] Woods Hole Oceanographic Institution. *Sentry AUV*. Mar. 2018. URL: https://www.whoi.edu/main/sentry (visited on 11/23/2018).

[19] Woods Hole Oceanographic Institution. *AUV Sentry*. 2019. URL: https://www.whoi.edu/what-we-do/explore/underwater-vehicles/sentry/ (visited on 09/21/2019).

[20] Isaac Kaminer et al. *Time-Critical Cooperative Control of Autonomous Air Vehicles*. English (US). first. Elsevier, Aug. 2017. ISBN: 9780128099469.

[21] Anastasios Lekkas and Thor Fossen. "Line-of-Sight Guidance for Path Following of Marine Vehicles". In: first. Lambert Academic, June 2013. Chap. 5, pp. 63–92. ISBN: 978-3-659-41689-7.

[22] P. Maurya, A. Pedro Aguiar, and A. Pascoal. "Marine Vehicle Path Following Using Inner-Outer Loop Control". In: *IFAC Proceedings Volumes* 42.18 (2009). 8th IFAC Conference on Manoeuvring and Control of Marine Craft, pp. 38–43. ISSN: 1474-6670. DOI: https://doi.org/10.3182/20090916-3-BR-3001.0071. URL: http://www.sciencedirect.com/science/article/pii/S1474667016318687.

[23] *MEDUSA class vehicles*. http://dsor.isr.ist.utl.pt/vehicles/medusa. 2018.

[24] NOAA. *How deep is the ocean?* June 2018. URL: https://oceanservice.noaa.gov/facts/oceandepth.html (visited on 11/23/2018).

[25] NOAA. *How Many Species Live in the Ocean?* June 2018. URL: https://oceanservice.noaa.gov/facts/ocean-species.html (visited on 11/23/2018).

[26] C. O'Donnell et al. *Celtic Sea Herring Acoustic Survey Cruise Report 2018, 08 - 28 October, 2018*. 2018. URL: http://hdl.handle.net/10793/1385.

[27] D. O'Donnell et al. *Blue Whiting Acoustic Survey cruise report, March 19- April 11, 2017*. 2017. URL: http://hdl.handle.net/10793/1319.

[28] Katsuhiko Ogata. *Modern Control Engineering*. 4th. Upper Saddle River, NJ, USA: Prentice Hall PTR, 2001. ISBN: 0130609072.

[29] Tiago Miguel Monteiro de Oliveira. "Moving Path Following Control System for Fixed-Wing Unmanned Aerial Vehicles". MA thesis. Faculdade de Engenharia da Universidade do Porto, Apr. 2017.

[30] A. Pascoal, I. Kaminer, and P. Oliveira. “Navigation system design using time-varying complementary filters”. In: *IEEE Transactions on Aerospace and Electronic Systems* 36.4 (Oct. 2000), pp. 1099–1114. ISSN: 2371-9877. DOI: 10.1109/7.892661.

[31] António Pascoal. “Autonomous Vehicles: An Introduction to Modeling”. Instituto Superior Técnico. 2018.

[32] Morgan Quigley et al. “ROS: an open-source Robot Operating System”. In: *ICRA workshop on open source software*. Vol. 3. Jan. 2009.

[33] Guilherme Manuel Vilela Sanches. “Sensor-Based Formation Control of Autonomous Marine Robots”. MA thesis. Instituto Superior Técnico, 2015.

[34] João Miguel dos Santos Ribeiro. “Motion Control of Single and Multiple Autonomous Marine Vehicles”. MA thesis. Instituto Superior Técnico, 2011.

[35] STREP. *CO3 Project*. Jan. 2015. URL: http://robotics.jacobs-university.de/projects/Co3-AUVs (visited on 11/23/2018).

[36] Unity Technologies. *Unity*. 2019. URL: https://unity.com/pt.

[37] Francesco Vanni Vanni. “Coordinated Motion Control of Multiple Autonomous Underwater Vehicles”. MA thesis. Instituto Superior Técnico, 2007.

[38] WiMUST. *Widely scalable Mobile Underwater Sonar Technology*. 2014. URL: http://www.wimust.eu/ (visited on 09/21/2019).