POLITECNICO DI TORINO

Master's Degree in Mechatronic Engineering

Master's Degree Thesis

Cooperative Circumnavigation and Tracking of Irregular Shape Target

Supervisors

Prof. Marcello CHIABERGE

Prof. Karl H. JOHANSSON

Prof. Joana F. G. FONSECA

Candidate

Cesare CAPUTI

Academic Year 2019/2020

Abstract

Collective behaviour happens not only among humans but also across several species of the animal kingdom. It depends on local interactions among individuals, which coordination is driven by the spread of a general information.
By taking inspiration from biological systems, researchers and engineers are trying to reproduce the cooperative behaviour of humans and animals like ants and birds, giving birth to multi-vehicle systems, with the aim of solving several problems.
A multi-vehicle system is composed of interconnected vehicles coordinated to act collectively towards global objective.
When controlling such systems, the goal is to obtain a coordinated behaviour through local interactions among vehicles and the surrounding environment.
The aim of the degree project is to perform a multi-vehicle circumnavigation and tracking of an irregular shape target, by implementing estimation and control algorithms.
The degree project is based on the licentiate thesis [1] performed by PhD student Joana Filipa Gouveia Fonseca, concerning the multi-vehicle target tracking.
The mentioned thesis has been developed in particular to monitor and study colonies of algae which grow out of control and produce toxic or harmful effects. These are known as harmful algal blooms, or HABs. They appear frequently and have a substantial negative effect, mainly on the environment but also on humans, causing large-scale mortality of fish, mammals and birds.
Through this master thesis, we aim at the achievement of a solution to the HABs problem, by applying the collective behaviour concept to a multi-agent system.
Starting from the literature, a preliminary work has been developed in Matlab/Simulink to get familiar with the algorithms presented in the licentiate thesis.
Afterwards, it is introduced the estimation and control's algorithm, developed in a

simulating environment, provided by the Gazebo simulator.
The approach presented in this thesis consists of three important steps: the estimation of the target, the agent's arrangement in a regular configuration, the circumnavigation of the target by the system.
This method involves the estimation of an irregular shape target, on the basis of the agents local measurements.
Then, after the determination of a leader agent, to which the remaining robots are subordinate, each agent is positioned around the target.
Finally, the whole system assumes a regular configuration, and circumnavigate the target.
Computer vision and estimation algorithms have been developed by using the computer vision library OpenCV in the ROS framework.
Circumnavigation has been achieved thanks to the development of Python code.
Final results are discussed, presenting pros and weakness of this work.

Keywords

Multi-Vehicle system, UAV, USV, HAB, optimal estimation, formation control, swarm robot, circumnavigation.

Acronyms

ASV Autonomous Surface Vehicles

AUV Autonomous Underwater Vehicles

HAB Harmful Algal Blooms

ISME Interuniversity Center of Integrated Systems for the Marine Environment

MAS Marine Autonomous Systems

OpenCV Open Computer Vision Library

ROS Robot Operating System

ROV Remotely Operated Vehicles

SITL Software in the loop

UAV Unmanned Aerial Vehicle

USV Unmanned Surface Vehicle

List of Figures

1.0.1 Collective Behaviours 1
1.1.1 A harmful algal bloom offshore of San Diego County, California 3
1.2.1 Blue-green algal blooms, Wisconsin 4

2.1.1 UAV/UGV Autonomous Cooperation 8
2.1.2 Spatially targeted communication 9
2.2.1 MAS for monitoring of decommissioned oil fields 10
2.2.2 Simulation of Multi-Robot Underwater Intervention 11
2.3.1 2018 Winter Olympics Opening Ceremony - 1,200 Drones, Intel 12
2.3.2 Four major formation type 13
2.4.1 Circumnavigation with target-centric formation around a moving target. 15
2.4.2 Simulation of multi-agent circumnavigation 16

3.1.1 Multi-vehicle system circumnavigating the target 19
3.3.1 Differential drive 22
3.4.1 Control scheme 23
3.4.2 Overall scheme 24
3.5.1 Multi-vehicle system circumnavigating the target 24
3.7.1 Pinhole camera model 27
3.7.2 Otsu's thresholding 28
3.7.3 Canny method 29
3.7.4 Fit ellipse function 29

4.1.1 Control system 33
4.2.1 ROS architecture 33
4.2.2 SINMOD Simulation 34
4.2.3 3DR Iris 35
4.2.4 Turtlebots 36

4.2.5 CVBridge - ROS package 37
4.2.6 RGB - HSV color spaces 37
4.2.7 Reference frames 38
4.2.8 Determination of contour point for optimal estimation 40
4.2.9 Determination of velocity command 41

5.1.1 Heading and Angular Velocity 44
5.1.2 Heading and Angular Velocity corrupted by WN 44
5.1.3 Coordinates and heading angle corrupted by WN 45
5.2.1 World 46
5.2.2 Optimal estimation with USVs covering one side 46
5.2.3 Optimal estimation with evenly spaced USVs 48
5.2.4 Optimal estimation with unevenly spaced USVs 48
5.2.5 Optimal estimation with USVs covering one side 49
5.2.6 Formation 50
5.2.7 Circumnavigation 51

Contents

1 Introduction **1**

1.1 Problem 2

1.2 Purpose 3

1.3 Benefits, Ethics and Sustainability 4

1.4 Methodology 5

2 Background **7**

2.1 Multi-vehicle systems 7

2.2 Multi-vehicle control for marine sensing 10

2.3 Multi-robot formation 12

2.4 Cooperative multi-vehicle circumnavigation and target tracking 15

3 Methodologies **18**

3.1 Problem statement 18

3.1.1 System Description 19

3.2 Estimation and control algorithms 20

3.3 Robot model 21

3.3.1 Kinematic Model 21

3.3.2 Dynamic Model 22

3.4 Control strategy 23

3.5 Problem Statement 24

3.6 PX4 Autopilot for drones 25

3.7 Computer vision 26

3.7.1 Pinhole camera model 26

3.7.2 Object detection - Thresholding 27

3.7.3 Edge detection - Canny filter 28

3.7.4 Optimal ellipse estimation - Fit ellipse 29

4 Algorithm implementation **31**

4.1 Preliminary work 31
4.2 ROS implementation 33
4.2.1 Target 34
4.2.2 Unmanned Aerial Vehicle 35
4.2.3 Unmanned Surface Vehicles 35
4.2.4 Computer Vision 36
4.2.5 Optimal Estimation 39
4.2.6 Formation and Circumnavigation 40

5 Simulation results **43**
5.1 Preliminary work's results 43
5.2 ROS/Gazebo implementation's results 45
5.2.1 Computer vision's results 46
5.2.2 Optimal Estimation's results 47
5.2.3 Formation and Circumnavigation's results 50

6 Conclusions **52**
6.1 Future Works 53

References **55**

Chapter 1

Introduction

Collective behaviour happens in nature not only among humans but also across several species of the animal kingdom. It depends on local interactions among individuals, which coordination is driven by the spread of a general information.
In the animal kingdom, collective behaviours occur frequently. For example, in Fig. 1.0.1 are shown animals aggregates such as fish shoal, birds flock and insect swarm. Animals get benefits from collective behaviours such as defence against predators, enhanced foraging success and higher success in finding a mate.

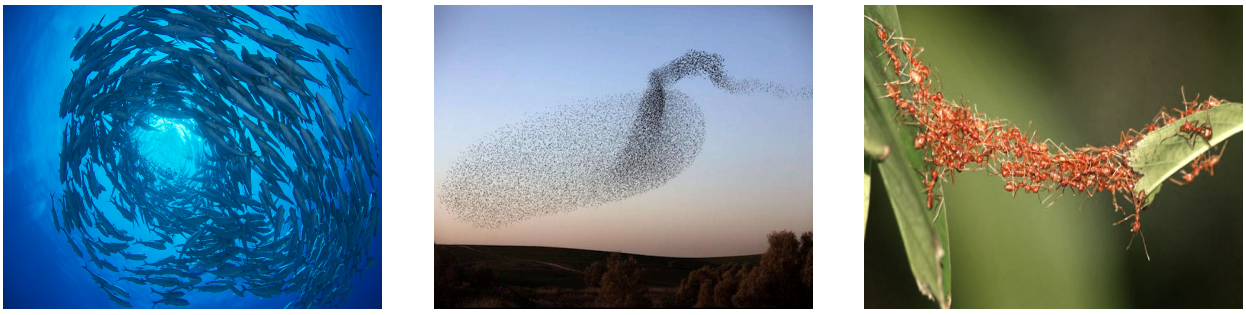


Figure 1.0.1: Collective Behaviours

Also human crowds behaviours can be assessed as collective, even though they gather for different reasons. A crowd doing the wave at a football game, a group of people forming around a street preacher, or even construction workers building up a building, show all the necessary requirements to be marked as cooperative behaviours.
Inspired by nature, researchers and engineers are trying to reproduce the cooperative behaviour of humans and animals like ants and birds, giving birth to multi-vehicle systems, with the aim of tackling several problems.

A multi-vehicle system is composed of interconnected vehicles coordinated to act collectively towards global objective.
When controlling such systems, the goal is to obtain a coordinated behaviour through local interactions among vehicles and the surrounding environment.
During the past decades, multi-vehicle systems have been subject of several researches. The interest in this field grew out of the multiple use cases, the robustness of the systems, the implications of costs reduction and the efficiency if compared with a more complex and expensive single vehicle.
Albeit the positive features of multi-vehicle systems, the dynamic nature of the vehicles and of the environment influences the complexity of the problem which depends on the number of variables to take into account. The achievement of the solution results to be thus not trivial. But, despite of this, a successful implementation can bring innovative solutions to real issues.
One motivating application is the monitoring of algal blooms. This phenomenon occurs frequently in all types of water (around coasts, in lakes and streams), and has a substantial negative effect mainly on the environment such as large-scale mortality of fish, mammals and birds, but also on humans.

1.1 Problem

Harmful algal blooms, or HABs, occur when colonies of algae — simple plants that live in the sea and freshwater — grow out of control and produce toxic or harmful effects on people, fish, shellfish, marine mammals and birds. The human illnesses caused by HABs, though rare, can be debilitating or even fatal, accordingly to the National Oceanic and Atmospheric Administration (NOAA) [2].
HABs occur naturally, but human activities that disturb ecosystems seem to play a role in their more frequent occurrence and intensity. Increased nutrient loadings and pollution, food web alterations, water flow modifications and climate change all play a role.
Indeed, in [3], they infer that climate change will influence marine planktonic system globally, and thus algal blooms may increase in frequency and severity. Increasing temperatures and ocean stratification are the perfect elements for the spreading of algal blooms. Therefore the need of tracking and studying the HABs is of paramount importance.

Figure 1.1.1: A harmful algal bloom offshore of San Diego County, California

In the past years, many studies and simulations have been carried out on dynamics of algal blooms, more specifically on diatoms and flagellates which are two species of algal blooms. Results show that the simulated abundance and distribution of diatoms and flagellates changes remarkably not only during the highly dynamic spring bloom but also during the summer [4].
Thus, periodic data collection of algal blooms are highly encourage in order to build accurate models. This would enhance the understanding of such an unpredictable, highly dynamic target.

1.2 Purpose

The lack of accurate information about Harmful Algal Blooms causes the lack of models which would allow researchers a better understanding of this more and more frequent phenomenon.
Multiple are the reasons why few models are available. First of all, high mission costs tend to reduce the number of measurements sessions carried out over a year. Then, algal blooms are sometimes located in hazardous areas which prevent the data collection.
Nowadays there are few ways of collecting HABs data. This operation can be carried out via satellite. For example, the National Centers for Coastal Ocean Science has developed the Algal Bloom Monitoring System to routinely deliver near-real-time images for use in locating, monitoring and quantifying algal blooms in coastal and lake

regions of the US. In Fig. 1.2.1 is shown the Winnebago Waterways/Green Bay, one of seven locations monitored via satellite.
Another way of collecting data is via monthly mission carried out by researchers, using a manned research vessel. However, both methods have drawbacks like the presence of clouds for the data collection through satellite and high costs related to the missions with vessels, causing a reduction of collected data, sometimes classified as unreliable.

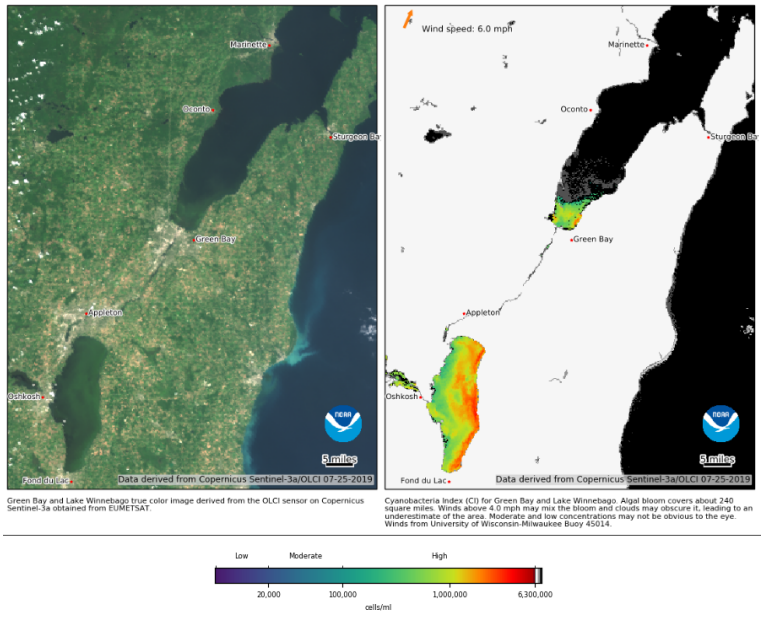


Figure 1.2.1: Blue-green algal blooms, Wisconsin

Therefore the questions rise spontaneously: is it possible to locally measure and track a dynamic target such as the HABs? If yes, how to increase the reliability and lower the costs of measurements missions?
Starting from [1], in this thesis we propose a novel approach to provide frequent, reliable and local measurements of HABs by means of an autonomous multi-vehicle system. Indeed, we investigate the control of multiple unmanned surface vehicles (USVs), for circumnavigation purposes, and the use of an unmanned aerial vehicle (UAV) for the detection and tracking of an algal bloom area.

1.3 Benefits, Ethics and Sustainability

The success of this degree project and of its future works may be very helpful for the scientific community, to better understand the dynamic behaviour of the HABs and to analyze, through specific measurements, their chemical composition and impact on

the surrounding environment.
Moreover, the implementation of this work can be helpful to preserve the environment, useful to prevent mortality of fish, birds, mammals and improving also the human health.
Out of the HABs specific case and besides the environmental aspects, the algorithms presented in this degree project can be applied to more general scenarios. Indeed, they can find room in many other use cases, in which the identification of a non-regular shape target is required and where multiple vehicles, eventually of different types (i.e. aerial and ground robots) can be deployed.
Thus, the implementation of the algorithms presented in this thesis can be used to demonstrate the ability of autonomous systems of helping the humans in dangerous tasks or extremely expensive missions, highlighting efficiency and cost reductions. Indeed, nowadays, there is a high focus on multi-vehicle systems because they would make possible the achievement of a more secure and efficient human-robot cooperation.

1.4 Methodology

The work developed in this master thesis is based on the algorithms presented by Fonseca et al. [1]. The author purports to introduce novel methodologies for the tracking and circumnavigation of HABs. Indeed, three interesting algorithms are proposed to account for the HABs problem.
On the basis of those algorithms, in this thesis, the work has been divided into two sections.
In the first part, the adaptive estimation and control algorithms have been analyzed and tested. The problem is introduced in chapters 3.1, where the functions of each agent of the system are defined. The irregular shape target is approximated to a circle so that a protocol based on local measurements, provided by an Unmanned Aerial Vehicle (UAV), can be used to estimate the target center and radius, as discussed in chapter 3.2.
In chapter 3.3, the model of the Unmanned Surface Vehicle (USV), relative to each agent of the system, is defined.
Finally, a control strategy is derived in chapter 3.4 to bring the vehicle system towards the target. For this purpose, Matlab/Simulink have been used.

In the second part of this master thesis, we focus on the optimal estimation and control strategy for cooperative circumnavigation, presented in chapter 4 of [1].
In section 3.5 the problem is presented. Differently from the first sections, the target shape is non-convex, as discussed in section 4.2.1; the UAV is used only for target detection while the USVs are used for circumnavigation purposes, respectively introduced in sections 4.2.2 and 4.2.3. The whole system then, cooperates to perform the target optimal estimation, as explained in section 4.2.5, carried out by means of computer vision's algorithm, deeply discussed in sections 3.7 and 4.2.4.
Finally, circumnavigation of the estimated target, while preserving the desired formation, can be carried out by the multi-vehicle, accordingly to section 4.2.6.
The Robot Operating System (ROS) and the simulator Gazebo have been used to simulate the dynamics of the agents, to develop the computer vision algorithms, useful to detect and estimate the target, and to implement the control strategy for circumnavigation and formation purposes.

Chapter 2

Background

In the past years, multi-vehicle systems have achieved resounding interest in the research field, due to their great ability to operate in different environments for various use cases.
In this chapter, a detailed description about the theoretical background is presented, upon which the degree project has been developed.
The main focus is on the literature concerning the cooperative behaviour of multi-vehicle systems and their marine applications, paying close attention to formation strategies, an important subject to study when dealing with multi-agents systems.
Some examples are also introduced to show the potentiality of such systems.
A detailed discussion concerning the cooperative multi-vehicle circumnavigation and target tracking is presented by analyzing some related works.
Moreover, a general idea of the algorithms presented in [1] will be provided, pinpointing the adaptive and optimal estimations and their relative control strategies.

2.1 Multi-vehicle systems

A multi-vehicle system consists of vehicle teams acting as autonomous agents. The control of complex systems such as multi-vehicle systems is an issue at the center of a growing need of distributed operations in a wide range of cooperative applications such as planetary exploration, search and rescue robot teams and multi agents cooperative carrying.
According to [5], the cooperative control concerns with engineered systems that

can be characterized as a collection of interconnected decision-making components with limited processing capabilities, locally sensed information and limited inter-component communications, all seeking to achieve a collective (global) objective.
Key element of such systems, composed by a variable number of agents, is the capability of each agent of taking decisions. Each robot is responsible for its own local motion, with the aim of achieving the whole group's goal.
This important concept is called decentralized control and, instead of the presence of one decision-maker that has access to all information incoming from all agents, each agent is capable of compute locally the actions to perform, accordingly to the information that it gathers.
Another feature that thrusts the study in this field is the scalability. This system's property consists in managing a growing amount of work by adding a large and not fixed number of interconnected resources to the system. Due to this, from a complexity point of view, both computation and communication problems may arise.
Murayama et al. [6] purposes to find a solution for the connectivity problem, besides the one concerning the collision avoidance.
Due to drastic changes in vehicle formation, the risk of having a failure in the network communication is high.
Therefore, it is proposed a decentralized trajectory planning method that guarantees continuous network connectivity and collision avoidance.

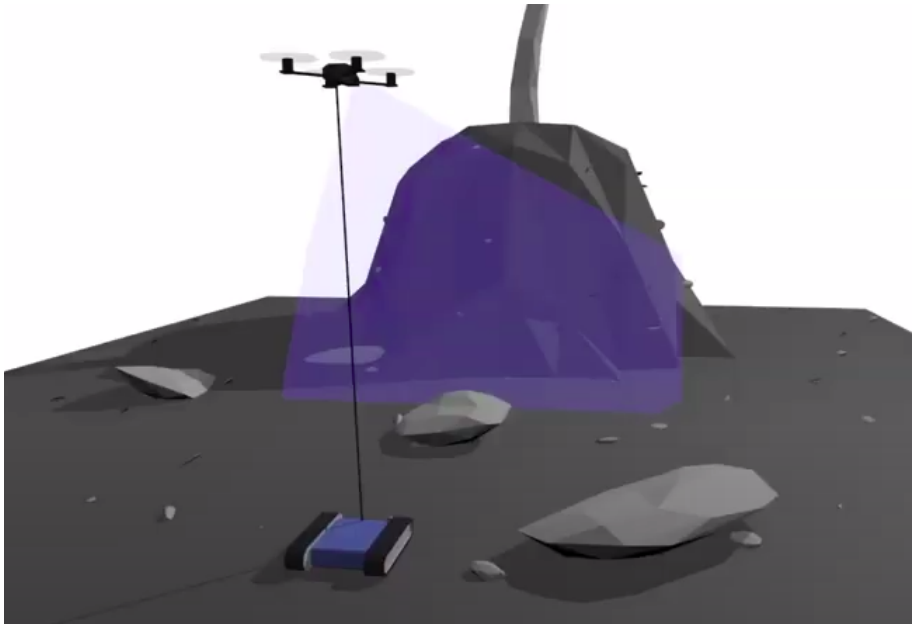
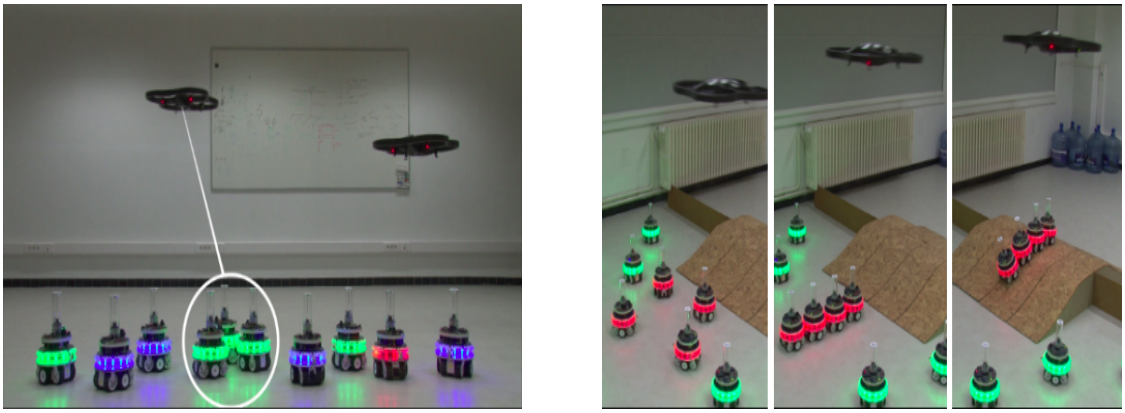


Figure 2.1.1: UAV/UGV Autonomous Cooperation

Another recent research concerning multi-vehicle cooperation is presented by Takahiro Miki et al. [7].
In this novel approach, a key feature of multi-vehicle system is highlighted, which allows to take advantage of the good qualities of each type of vehicle.
The proposed cooperative system is comprising of two different kind of robots, an Unmanned Aerial Vehicle (UAV) and an Unmanned Ground Vehicle (UGV).
Indeed, the positive factors of each vehicle are exploited, such as high mobility and wide range of sensor coverage for the UAV, large payload and long battery duration for the UGV.
As a result, in an unknown environment, as shown in Fig. 2.1.1, the UAV employs visual inertial navigation with 3D voxel mapping and obstacle avoidance planning whereas the UGV generates, through the voxel map, an elevation map to execute path planning based on a traversability analysis.
A communication method, which includes the use of a heterogeneous multi-robot system, is presented by [8]. A spatially targeted communication is presented when flying robots and ground-based self-assembling robots are used.
As already discussed before, flying robots' privileged view over the environment is used to determine and communicate information to groups of ground-based robots (Fig. 2.3.2b) on what morphologies to form to carry out upcoming tasks (Fig. 2.3.2d).
So far we have introduced only few of the recent researches concerning multi-vehicle systems, composed or not of different kind of robots, but their use can be useful in many other applications.



(a) Subgroup of robots chosen by the UAV

(b) Robot self-assembling into a linear morphology to safely cross a hill

Figure 2.1.2: Spatially targeted communication

2.2 Multi-vehicle control for marine sensing

The Earth is a watery place. About 71 percent of the Earth's surface is water-covered, and the oceans hold about 96.5 percent of all Earth's water. Monitoring and measuring such large and arduous areas in a reliable and cheap way is nowadays a problem of relevant importance.

Multi-vehicle unmanned systems can be a suitable solution due to their precision and cost efficiency in hazardous environments [9] albeit many challenges can be encountered when dealing with them, such as robustness of control strategies, uncertainty in inter-vehicle communications and distributed operation, system complexity due to the problem size and coupling between tasks.

Such systems can rely on the versatility and scalability of their members, that could be UAV, USV and ASV: indeed, different vehicles' skills can be exploited for different purposes, such as long time period measurements [10].

Jones [11] discusses about alternative solutions to conventional monitoring systems of marine environments. Already available marine autonomous systems (MAS) offer a wide range of solutions for supporting environmental assessment and monitoring of decommissioned oil fields. Data can be directly collected using acoustic, visual, and oceanographic sensors deployed on MAS. In this way, there would be cost savings and a substantial improvement in the temporal and spatial resolution of environmental monitoring.

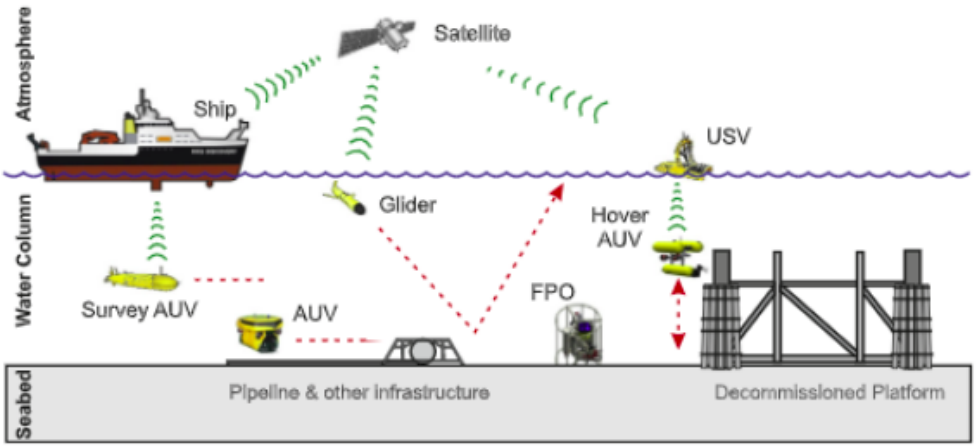


Figure 2.2.1: MAS for monitoring of decommissioned oil fields

Prototyping and testing of such robots through realistic simulations can help to validate several aspects of the projects, customizing the physical characteristics of the vehicles

model and the dynamic of the environment to evaluate the systems performances and reliability.
Thus, developers and researchers are working on software or plugins compatible with Gazebo simulator for carrying out simulations of unmanned underwater vehicles, such as ROVs, remotely operated vehicles, and AUVs, autonomous underwater vehicles [12], [13], [14].
Fig. 2.2.2 shows a simulation's example of multiple underwater robots in an offshore wind park scenario for intervention purposes [15].

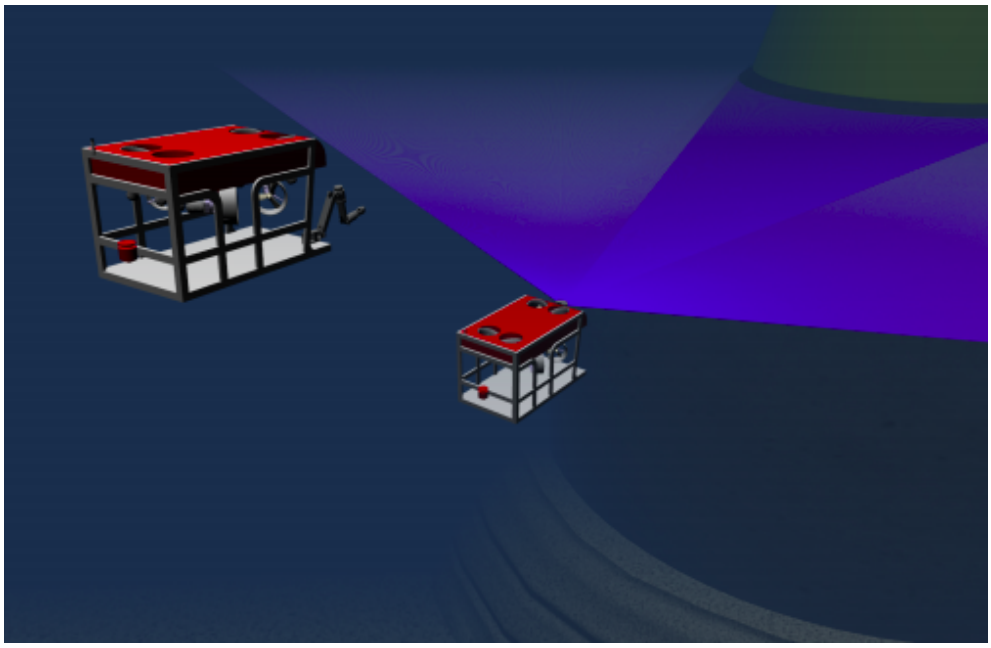


Figure 2.2.2: Simulation of Multi-Robot Underwater Intervention

Multi-vehicle system can also perform other activities beyond the ones concerning the marine environmental monitoring and intervention.
The problem of safeguarding civilian harbours against terroristic attacks, coming from the so-called "blue border" (i.e. the sea-side), is receiving an increasing interest in the recent years. In this context, the use of a multi-vehicle team of "protecting" autonomous marine vehicles certainly represents a valid solution for reducing the harbour vulnerability, keeping safe human lives.
A research activities of the Italian Interuniversity Center of Integrated Systems for the Marine Environment, ISME, concerns the harbour protection with autonomous marine vehicles [16]. Under normal conditions, the vehicles can perform patrolling surveys of the more crucial water ways but, if an unauthorized vessel or a vessel moving in a suspect way is detected, one vehicle can be used for "intercepting" the

menace, allowing to determine whether the suspect vessel is “hostile” or “friend” without exposing humans directly to risk.
In the literature there are many other marine applications in which multi-vehicle systems can be used and many researchers are still developing new approaches.

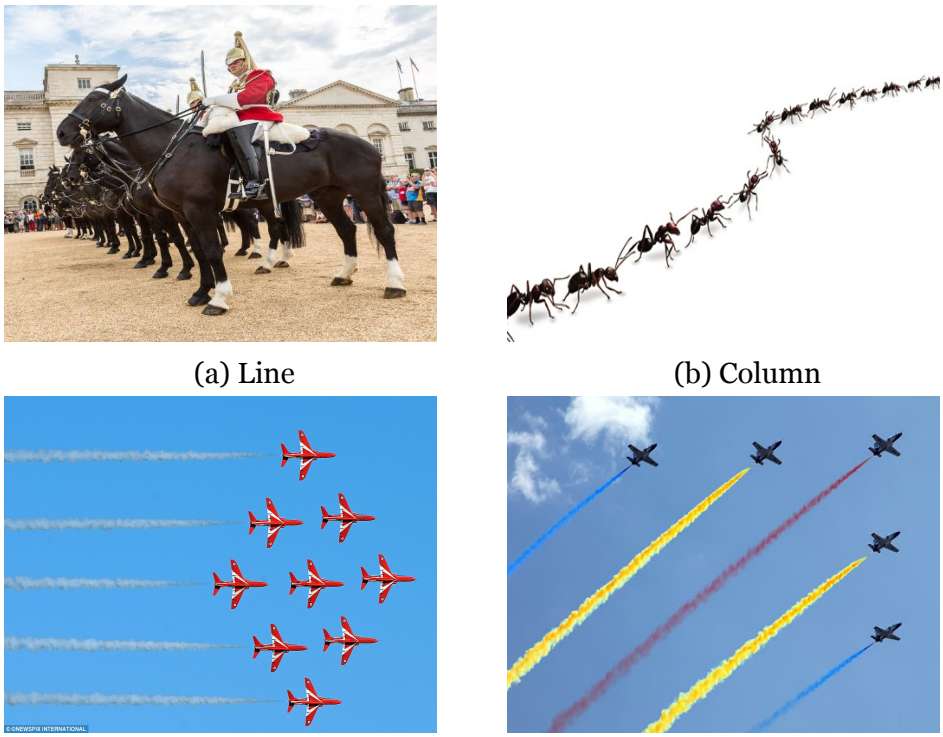
2.3 Multi-robot formation

In nature, the animals group themselves to combine their senses in order to maximize the chance of detecting predators or to more efficiently forage for food. Sometime it helps them to maximize the opportunity of finding a partner.
By grouping, like schooling for fish and flocking for birds, animals benefit from the presence of their counterpart.
This phenomenon belong to the subject of the formation behaviors.
In robotics, multi-vehicle systems could also benefit from formation tactics. Therefore, in the past decades, thanks to the enhancements in communication and computational power, formation control has gained interest in the scientific community and several strategies for controlling multi-vehicle systems, moving in formation, have been developed.

Figure 2.3.1: 2018 Winter Olympics Opening Ceremony - 1,200 Drones, Intel

According to Ji et al. [17], a formation control problem is defined as “to drive the agents to a special configuration such that their relative position satisfy a desired

topological (spatial) and physical constraints". In other words, a formation control problem requires that, while all the robots collectively move along a path, each agent of the system stands in a specific configuration, such that the whole system is arranged in a desired shape, usually regular.



(a) Line (b) Column

(c) Diamond (d) Wedge

Figure 2.3.2: Four major formation type

Desired formation can be defined in 2D or in 3D configuration and it should be preserved even in the presence of obstacles. In Fig. 2.3.2, four major formation type (line, column, diamond, wedge) are shown.

Besides, formation control allows individual team agents to concentrate their sensors across a determined portion of the environment, while their partners cover the rest. This feature can be exploited and adopted in several applications such as military applications where sensor ranges are limited, marine applications where it is required to cover wide water areas, search and rescue where many robot scout can be deployed to quicken the activities, cooperative transportation by small autonomous vehicles or cooperative surveillance.

The use of multi-vehicle system, specially drones, in formation scheme is making its way also in the entertainment field. As shown in Fig. 2.3.1, drones are currently deployed during ceremonies, sometimes taking the place of fireworks, to enchant the

crowd with astonishing exhibitions.

According to Balch et al. [18], the approaches to formation generation in robots may be distinguished by their sensing requirements, their method of behavioral integration, and their commitment to preplanning. Thus, at the end of the last century, reactive behaviours have been implemented in multi-robot teams, by integrating the formation behaviors with other navigational behaviors to enable a robotic team to reach navigational goals, avoid hazards and simultaneously remaining in formation [18].

In the literature, three techniques for formation position have been determined: unit-center-referenced, neighbor-referenced and leader-referenced.

About the last one mentioned, Barca et al. [19] purports to overcome, through the use of graph theory, the high communication load, high energy usage and lack of robustness of conventional techniques.

The improvement of the state-of-the-art formation control schemes for leader-follower type is achieved by employing mechanisms that enable groups of robots to move in two-dimensional formations without the need for inter robot communication.

On the contrary, it worths to take into account a possible drawback when relying only on a leader vehicle.

Motion planning in the formation control structure is typically achieved by firstly finding a trajectory for the leader and then controlling the other vehicles to maintain a desired relative position with respect to the leader. Albeit this may perfectly work, a hardware failure by the leader or a loss of communication among the agents may affect irreversibly the whole system, questioning the robustness of the leader-referenced formation technique.

By relying on the communication strength, Van Parys et al. [20] overcomes the concept of leader-reference method and introduces a novel approach in which all members of the formation are equal. This implies that each vehicle determines its own trajectory. By allowing communication, the vehicles are able to adapt their trajectories in order to satisfy the formation constraints.

2.4 Cooperative multi-vehicle circumnavigation and target tracking

Target tracking and circumnavigation with a network of autonomous agents is a particular problem within cooperative multi-vehicle control. Huge interest about this topic is due to the numerous applications in which it can used.
The literature provides a wide range of researches, such as [21], [22], [23] concerning formation control or cooperative circumnavigation of a known target, formation and estimation for tracking a moving target defined as a unit point.
Deghat et al. [24] faces the problem of localization and circumnavigation of a slowly moving target with unknown speed, when the agent only knows its own position with respect to its initial frame, and the bearing angle to the target in that frame.
When dealing with the circumnavigation problem, the scalability property can be exploited and thus the employment of a various number of agents occur.
Indeed, Boccia et al. [25] present a distributed algorithm to estimate the position of the target with a network of planar autonomous agents. Thus, the agents are driven to rotate around the target while forming a regular polygon and keeping a desired distance.

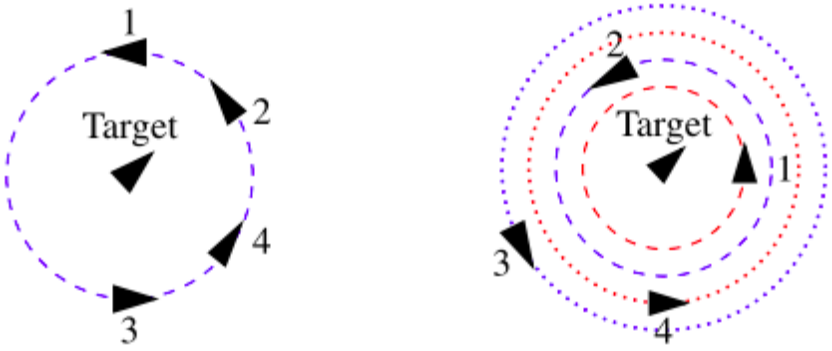


Figure 2.4.1: Circumnavigation with target-centric formation around a moving target.

Sen et al. [26], instead, proposes a cooperative protocol for multi-circular circumnavigation with any desired angular spacing around a non-stationary target. The formation technique refers to the unit-center-referenced, as shown in Fig. 2.4.1, where it is shown an unevenly spaced single orbit (left) and multiple circles with unequal angular spacing (right). The target information and desired formation

parameters are assumed to be known to only one agent partially.
Similarly, Wang et al. [27] focuses on the problem of tracking a moving target with a group of mobile robots with the constraint of allowed local communication only between neighbor robots.
Analogously, Zhong et al. [28], propose a distributed control strategy for the multi-agent system to achieve three objectives: reaching the target plane with set orientation, circumnavigate around the target with prescribed radius, and avoiding collisions among agents.
By increasing the number of targets without changing their nature, still considered as unit points, Shao et al. [29] introduces an interesting algorithm to perform, through a multi-agent system, the localisation and circumnavigation in the multi-target scenario, by employing 2D bearing measurements. By means of an estimation algorithm, each agent cooperatively estimate the centroid of the multi-target in order to guarantee that each agent circumnavigates all targets around the multi-target centroid with a desired radius.
Until now, the nature of the target has been considered as a unit point. Fonseca et al. [1] introduces a remarkable difference from the previous researches: the target is no more assumed as unit point but it is considered as an irregular dynamic bidimensional shape.

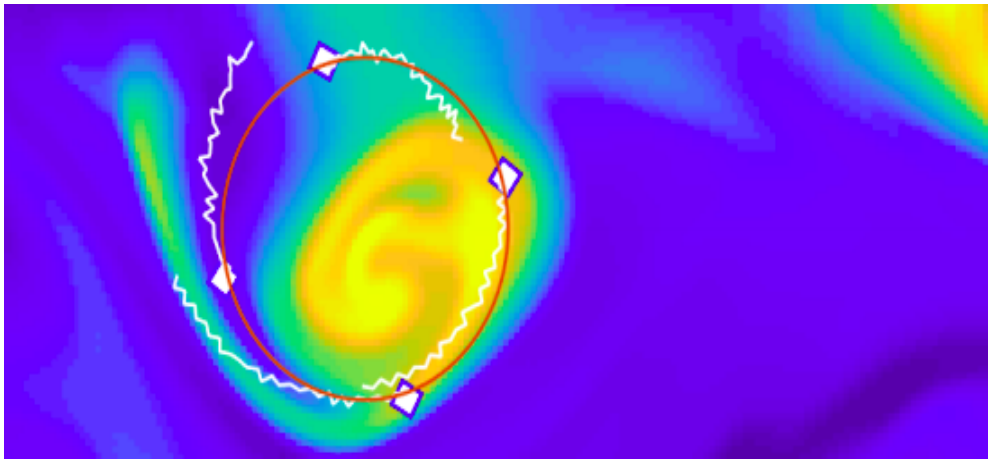


Figure 2.4.2: Simulation of multi-agent circumnavigation

Starting from this not negligible feature, it is introduced a novel method for cooperative multi-vehicle circumnavigation and tracking of a mobile target, taking also into account its shape shifting that requires constant measuring and estimation while performing formation control.

A protocol based on local measurements provided by the vehicles is developed to detect and estimate the target's location and shape. Then, a control strategy is derived to bring the multi-vehicle system close to the target. Afterwards, the target circumnavigation occurs while forming a regular polygon. Fig. 2.4.2 shows a simulation of cooperative multi-vehicle circumnavigation and target tracking where the target estimate (red) and the agents path (white) are highlighted.
The degree project is based on [1], therefore a deep analysis has been carried out.
In the licentiate thesis, estimation and control algorithms are proposed for multi-vehicle system target tracking and circumnavigation.
Cooperative multi-vehicle circumnavigation of an irregular dynamic target using adaptive estimation considers the problem of tracking a mobile target estimating, through the use of only one of the agents, its characteristics. For this purpose, both UAV and USVs are deployed.
Differently, with distributed sensing, the measurement process is decentralised by having all the vehicles capable of collecting and sharing data to perform the target estimation.
An irregular shape target is still considered. The multi-vehicle system is again composed by an UAV and n USVs. The former's function is to to track through the computer vision algorithms the target, whereas the latter have to measure their distance from the target's boundary.
Finally, optimal estimation is carried out allowing the target circumnavigation while preserving the desired formation scheme.

Chapter 3

Methodologies

In this chapter, the research methodologies, tools and procedure are introduced to address the cooperative multi-vehicle circumnavigation and target tracking problem. The main issue that we aim to solve is the local measurement and tracking of an irregular shape target, such as the HABs, by means of a multi-vehicle system, by exploiting estimation and control algorithms.
A preliminary work has been done to get familiar with the adaptive estimation and control algorithm. Sections 3.1, 3.2, 3.3 and 3.4 refer to the problem statement, its description and the presentation of the robot model. The equations governing the estimation and control algorithms are discussed. Finally, the schematic of the feedback control strategy is introduced.
In section 3.5, the same problem is considered but it is faced with a different approach. Indeed the goal is achieved in a simulated environment through the implementation of the optimal estimation and its relative control algorithm. Methods described in sections 3.6 and 3.7 refer to the implementation performed in Gazebo through the ROS tool.

3.1 Problem statement

We consider the problem of tracking a target using a multi-USV system and a single UAV, as shown in Fig. 3.1.1. The UAV purpose is to constantly measure its distance from the target whereas the USVs goal is to process the estimates and circumnavigate the target.

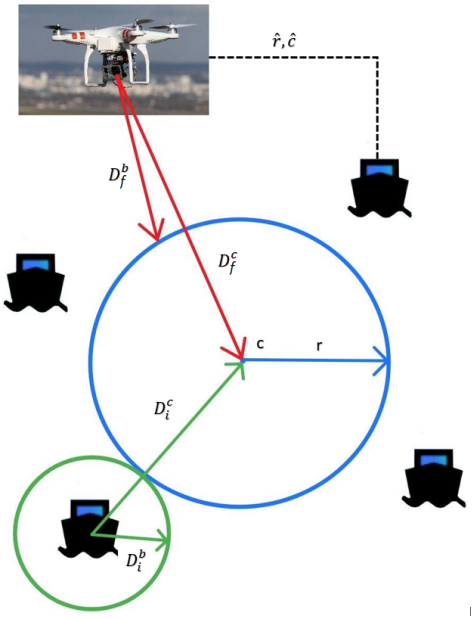


Figure 3.1.1: Multi-vehicle system circumnavigating the target

Measurements are processed by the UAV estimator, which provides the estimates to the multi-USV system.
We assume to work with a system of n vehicles circumnavigating a target shaped as a circle, as shown if Fig. 3.1.1.

3.1.1 System Description

We define the circle as

$$(\mathbf{c}, r) \in R^3 \tag{3.1}$$

where $\mathbf{c} = (x, y)$ and r are the center and radius of the circle. We define the estimate of the circle as $(\hat{\mathbf{c}}, \hat{r}) \in R^3$ and the position of each vehicle as

$$\mathbf{p}_i = [x_i, y_i] \in R^2 \tag{3.2}$$

The UAV would provide initial estimates $\mathbf{c}(0) = (x(0), (0))$ and $r(0))$ beside the initial position $\mathbf{p}_i(0)$ of each vehicle. Each vehicle has access to its GPS position and to the position of the one in front of it. The robots constantly measure their distances to the target's center and boundary, D_i^c and D_i^b respectively, and share it with the other robots. With the available data we can implement the feedback control system such that the

kinematics of each agent is

$$\dot{\mathbf{p}}_i = \mathbf{u}_i, \qquad i \in [1, ..., n] \tag{3.3}$$

where $\mathbf{u}_i \in R^2$ is the control input.
We would like to space the agents equally along the defined circle to avoid them from concentrating in some region, which would mean losing information about some algal bloom front. Therefore we define the counterclockwise angle β_i between two consecutive vehicles having position $\mathbf{p}_i - \hat{\mathbf{c}}$ and $\mathbf{p}_{i+1} - \hat{\mathbf{c}}$ such that

$$\beta_i(0) \geq 0 \qquad \sum_{i=1}^{n} \beta_i(0) = 2\pi \qquad i \in [1, ..., n] \tag{3.4}$$

3.2 Estimation and control algorithms

The UAV estimator block processes the measurements taken by the drone flying above the target to provide estimates of the center and radius of the target. We denote with $\mathbf{p}_f$ the position of the UAV and with D_f^c and D_f^b the distances measured by the UAV from the center and boundary of the target. We can estimate the radius and the center of the target accordingly to the following expressions

$$\dot{\hat{r}} = -\gamma_1 \dot{D}_f^c \left[\frac{1}{2} \left(\frac{d}{dt} \left(D_f^b \right)^2 - \frac{d}{dt} \left(D_f^c \right)^2 \right) + \dot{D}_f^c \hat{r} \right] \tag{3.5}$$

$$\dot{\hat{\mathbf{c}}} = -\gamma_2 \dot{\mathbf{p}}_f \left[\frac{1}{2} \left(\frac{d}{dt} \left(D_f^c \right)^2 - \frac{d}{dt} ||\mathbf{p}_f||^2 \right) + \dot{\mathbf{p}}_f^\top \hat{\mathbf{c}} \right] \tag{3.6}$$

where γ_1 and γ_2 are scalar quantities. In the above equations, the derivatives of measurements occur and it is required the explicit differentiation of measured signals with accompanying noise amplification. To prevent this, we adopt the state variable filtering and the equations 3.5 and 3.6 can be expressed as

$$\dot{\hat{r}} = -\gamma_1 V \left[\eta - m + V \hat{r} \right] \tag{3.7}$$

$$\dot{\hat{\mathbf{c}}} = -\gamma_2 V_2 \left[\eta_2 - m_2 + V_2^\top \hat{\mathbf{c}} \right] \tag{3.8}$$

We can now derive the expression of the desired control input. We define the direction of each USV towards the estimated center of the target as the bearing ψ_i and its perpendicular $E\psi_i$. The desired control input can be expressed as

$$\mathbf{u}_i = \hat{\mathbf{c}} + \left(\hat{D}_i^b - \frac{1}{\delta}\dot{\hat{r}} \right) \psi_i + \beta_i \hat{D}_i^c E\psi_i \tag{3.9}$$

The control actuation of a USV is limited, therefore we introduce actuation bounds so that for a specific $\mathbf{u}_i$ it is possible to have $\mathbf{U}_i$ within some specified bounds

$$\mathbf{U}_i = \delta \mathbf{u}_i \tag{3.10}$$

where δ is a parameter to be defined.
The variable *r* can be estimated under the persistent excitation condition on D_i^c.
Persistent excitation plays a key role in establishing parameter convergence in adaptive identification, as discussed in [1].

3.3 Robot model

We model each vehicle as a unicycle, a vehicle with a single steerable wheel.
In this section, the dynamic model of the unicycle-like mobile robot proposed by Martins et al. [30] is reviewed. As shown in Fig. 3.3.1, the configuration of a unicycle is described by

$$\mathbf{q}_i = [\mathbf{p}_i, \theta_i] \in R^3 \tag{3.11}$$

where $\mathbf{p}_i = [x_i, y_i] \in R^2$ is the position of each vehicle and θ_i is the heading angle.

3.3.1 Kinematic Model

Due to the pure rolling constraint, the component of the velocity of the vehicle along the perpendicular of the heading direction is set to 0; this leads to the definition of the unicycle kinematic model

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos\theta \\ \sin\theta \\ 0 \end{bmatrix} v + \begin{bmatrix} -a\sin\theta \\ -a\cos\theta \\ 1 \end{bmatrix} \omega \tag{3.12}$$

where v and w are the driving and the steering velocity respectively, G is the center of mass, C is the position of the castor wheel, h is the point of interest having coordinates (x_i, y_i), θ is the heading angle between the robot orientation and the x axis, a is the distance between the point of interest and the central point of the virtual axis linking the traction wheels (point B).

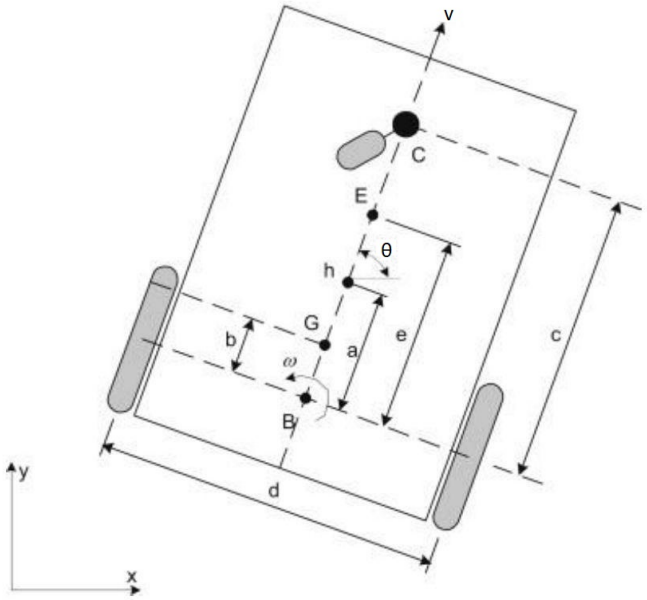


Figure 3.3.1: Differential drive

3.3.2 Dynamic Model

The complete mathematical model adopted from De La Cruz and Carelli [31], neglecting disturbances, is written as:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \\ \dot{v} \\ \dot{\omega} \end{bmatrix} = \begin{bmatrix} v\cos\theta - a\omega\sin\theta \\ v\sin\theta + a\omega\cos\theta \\ \omega \\ \frac{\phi_3}{\phi_1}\omega^2 - \frac{\phi_4}{\phi_1}v \\ -\frac{\phi_5}{\phi_2}\omega v - \frac{\phi_6}{\phi_2}\omega \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ \frac{1}{\phi_1} & 0 \\ 0 & \frac{1}{\phi_2} \end{bmatrix} \begin{bmatrix} v_{ref} \\ \omega_{ref} \end{bmatrix} \tag{3.13}$$

where v_{ref} and ω_{ref} are the desired value of the linear and angular velocities, representing the input signals of the system. The vector $\phi \in R^6$ takes into account

physical parameters of the robot.

3.4 Control strategy

The control strategy adopted to control the velocities of each vehicle is shown below. For this purpose, we may adopt PID controllers to handle each dynamic parameter. The resulting feedback control scheme is shown in Fig. 3.4.1.
The control scheme consists of two closed-loop feedbacks, used to control the vehicle velocity and to determine the angle between two successive vehicles. The reference inputs are v_d and w_d, the desired driving and steering velocities derived from the equation 3.9, which instead provides v_x, v_y. The control input u, provided by the controller, enters the dynamic model of the system, which gives as outputs, the vehicle's velocities. These enter the kinematic model's block giving as output the coordinates of the vehicle.
The scheme of the whole system is presented in Fig. 3.4.2.

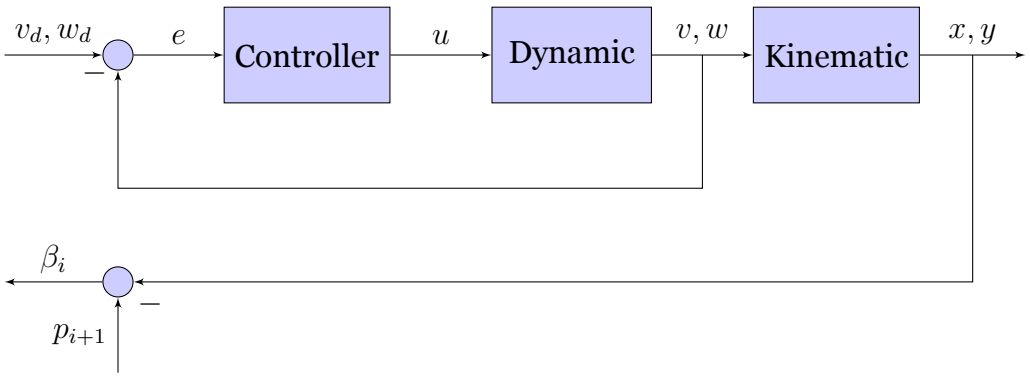


Figure 3.4.1: Control scheme

Starting from the measurement carried out by the UAV, the estimator block processes the data sent by the UAV to provide the estimations of radius and center coordinates. The latter are used to get the heading direction ψ_i by means of the blocks "Heading direction" whereas the angle between two consecutive vehicles, β_i, is obtained by means of the center coordinates, the robot position and the one of of the successive vehicle. This is performed in the "Robots angle" respectively.
It is possible to determine for each USV the desired velocity, which would be the input of the control scheme previously introduced in Fig. 3.4.1, represented by the "Controller/Robot model" block.

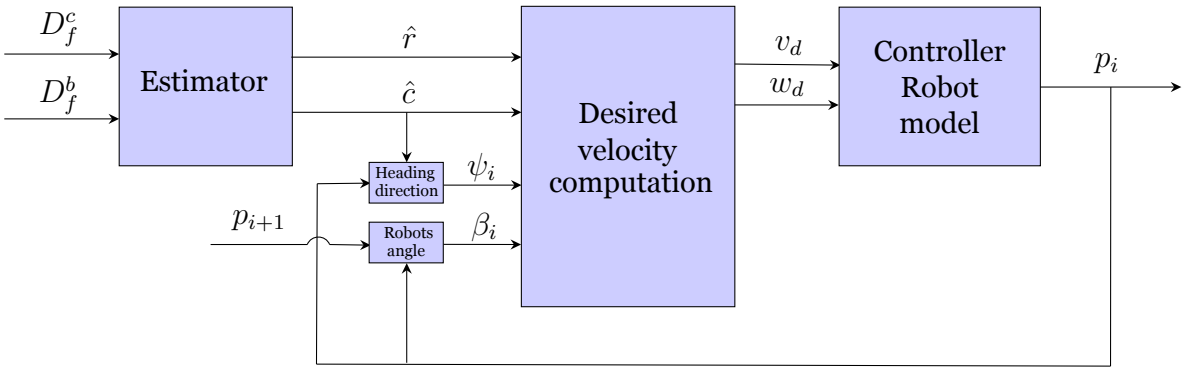


Figure 3.4.2: Overall scheme

3.5 Problem Statement

From this section we deal with the same problem of tracking and circumnavigation, facing it with a different approach.

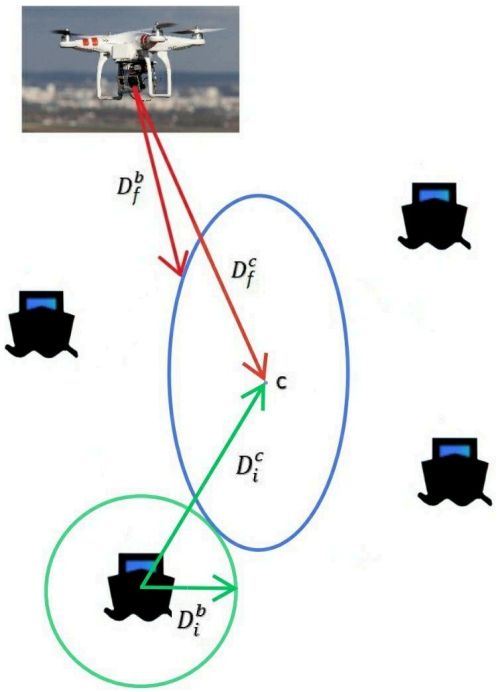


Figure 3.5.1: Multi-vehicle system circumnavigating the target

We define a different setup and, therefore, a different algorithm to solve the multi-vehicle target tracking problem. We do not rely on a single vehicle for the measurement

process, which is now decentralized since all vehicles are capable of collecting and sharing data to achieve target estimation and thus circumnavigation.
We consider the target with an irregular shape approximated, through the optimal estimation algorithm, to an ellipse.
In this approach, the USV system is composed of n USVs, all measuring their distance to the boundary of the target. After the target estimation, the USV system must circumnavigate the target while forming a regular polygon.
Fig. 3.5.1 shows the described system, which appears to be similar to the one depicted in Fig. 3.1.1, with a different target shape and measurement capabilities of each USV.

3.6 PX4 Autopilot for drones

Core of the degree project is the use of a UAV, a drone equipped with a mono camera placed below it and facing downward. Its goal is to track the algal bloom, processing the image through computer vision algorithm to get the target estimation.
A drone is an unmanned robotic vehicle that can be remotely or autonomously controlled.
Drones are used for many applications as aerial photography/video, carrying cargo, racing, search and rescue etc. Different types of drones can be used in different scenarios; it could be air, ground, sea, and underwater. These are referred to as Unmanned Aerial Vehicles (UAV), Unmanned Aerial Systems (UAS), Unmanned Ground Vehicles (UGV), Unmanned Surface Vehicles (USV), Unmanned Underwater Vehicles (UUV).
The ”brain” of the drone is called ”autopilot”. It consists of flight stack software running on vehicle controller hardware, the ”flight controller”.
PX4 Autopilot [32] is a powerful open source autopilot flight stack, suitable for our purpose since simulators allow to control vehicles in simulated environment. Furthermore, software in the loop (SITL) can be carried out on an already built in UAV model.
In this way, algorithms can be tested within a modelling environment on a model representing the real system. This can help to test and improve the software.

3.7 Computer vision

Vision is one of the most powerful sense of the human being. From vision, we contactless extract an extremely high informative content.
Analogically, artificial vision in robotics is defined as the process of extraction, characterization and interpretation of information coming from images of a tridimensional world.
By exploiting artificial vision, we can perform the detection of the target and the determination of its center and boundaries positions with respect to the multi-vehicle system. This plays a key role in the fulfillment of this work.
The suitable tool for this purpose in robotic applications is OpenCV [33]. OpenCV, Open Source Computer Vision Library, is a library of programming functions mainly aimed at real-time computer vision. It is used for several application as facial and gesture recognition, Human–Computer interaction (HCI), mobile robotics and object identification, Augmented reality, etc.
In the following sections we propose the work developed to detect, analyze and estimate the target.

3.7.1 Pinhole camera model

A pinhole camera is the simplest imaging device which captures the geometry of perspective projection.
This model allows the mapping from a three dimensional world onto a two dimensional plane. The working principle is based on rays of light entering the camera through an infinitesimally small opening. The intersection of light rays with the image plane form the image of the object.
As shown in Fig. 3.7.1, a scene view is formed by projecting 3D points into the image plane using a perspective transformation.
By neglecting extrinsic parameters and distortion coefficients, the mathematical model of the pinhole, expressed in the equation 3.14, can be exploited to find the 3D coordinates of our points of interest, which are the target contour points. Camera parameters can be determined by means of camera calibration procedure [34].

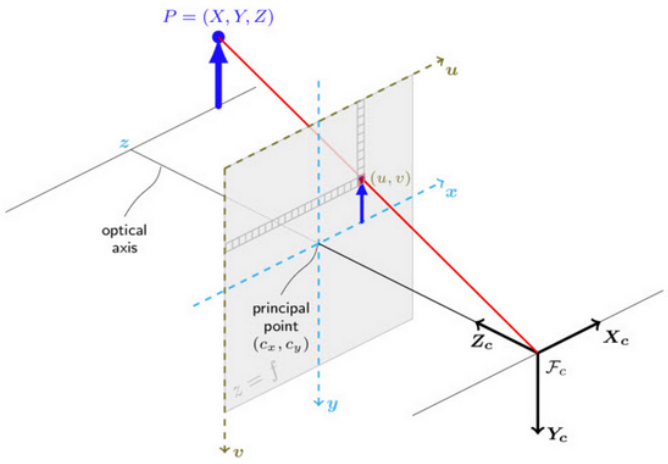


Figure 3.7.1: Pinhole camera model

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \tag{3.14}$$

where:

- u, v are the coordinates of the projection point in pixels;
- f_x, f_y the focal lengths expressed in pixel units;
- c_x, c_y is a principal point that is usually at the image center;
- X, Y, Z are the coordinates of the point in the 3D world.

3.7.2 Object detection - Thresholding

In image processing, segmentation is the process that subdivides the scene in its constituent parts, or objects, and it is one of the most important steps because, at this stage, objects are extracted from a representation toward a subsequent identification and analysis.
One of the simplest method and most used techniques to detect objects, separating them from the background, is the thresholding [35].
Thresholding is a way to create a binary image from a grayscale or full-color image.
The simplest thresholding methods replace each pixel in an image with a black pixel if the image intensity is less than some fixed constant T or a white pixel if the image intensity is greater than that constant.

However, the choice of the constant T is sometimes critical and it can be variable, as it could occur in a general scenario for our purpose. Thus, for the detection of the target, we make use of a method to perform automatic image thresholding, the Otsu's method. Otsu's method avoids having to choose the value of the constant T and determines it automatically an optimal global threshold. If we consider a bimodal image (with only two distinct image values), where the histogram would only consist of two peaks, a good threshold would be in the middle of those two values.
Therefore, Otsu's method exhibits the relatively good performance if the histogram can be assumed to have bimodal distribution.
On the other hand, if the object area is small compared with the background area, the histogram no longer exhibits bimodality. Fig. 3.7.2 shows also that, if the variances of the object and the background intensities are large compared to the mean difference, or the image is severely corrupted by additive noise, the determination of the threshold through the Otsu's method is compromised.

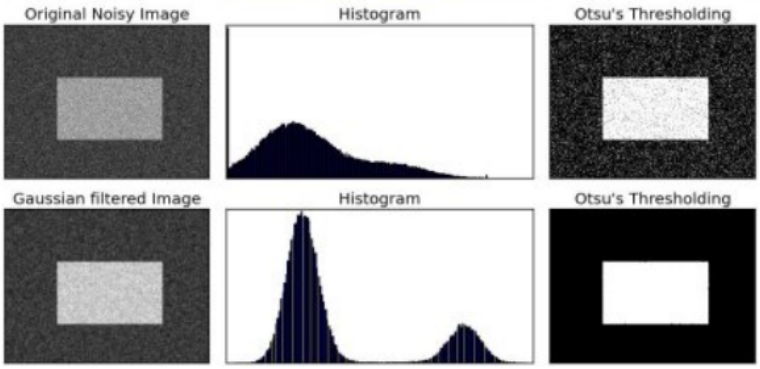


Figure 3.7.2: Otsu's thresholding

3.7.3 Edge detection - Canny filter

The Canny edge detector [36] is one of the most robust methods for edge detection. It is a multi-stage algorithm and it is based on the image gradient after the application of a Gaussian filter. In general the Gaussian reduces and smooths the image to remove the noise. Thereafter, two thresholds are defined, which lead to the definition of strong and weak edges. This is done to remove any unwanted pixels which may not constitute the edge.
In this thesis, the determination of the target contour has been accomplished by processing the binary image obtained through the Otsu's thresholding, so that the black and white contrast can be exploited.

Fig. 3.7.3 shows an example of the processing of a grayscale image through the Canny filter.

Figure 3.7.3: Canny method

3.7.4 Optimal ellipse estimation - Fit ellipse

The estimation algorithm concerning the cooperative estimation using distributed sensing requires an optimal estimation of the target shape to accomplish the circumnavigation task.
In this thesis, we have decided to approximate the shape of the non-convex target to an ellipse, since it is the general case of the circle, previously used.

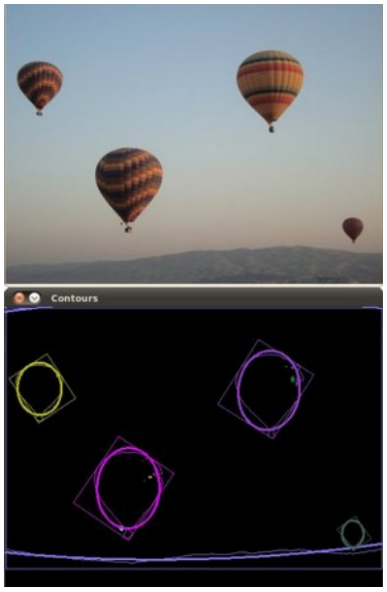


Figure 3.7.4: Fit ellipse function

For this purpose, the $fitEllipse()$ function [37] of the OpenCV library has been used.

The function calculates the ellipse that fits (in a least-squares sense) a set of 2D points best of all.
To perform the target estimate, we have picked as many contour's points as the number of USVs deployed. Those points are the closest to the USVs positions, as described in section 4.2.5. As result, the function returns the rotated rectangle in which the ellipse is inscribed, as shown if Fig. 3.7.4.

Chapter 4

Algorithm implementation

The work developed in this degree project aims to extends the state of the art of the research by adding a relevant dowel to this innovative and continuously growing field. In the following chapter, the implementation of the algorithms in two different frameworks is discussed. The methodologies and tools previously introduced are required for the successful achievement of the degree project.
Section 4.1 concerns the implementation in Matlab/Simulink of the adaptive estimation algorithm and control strategy, when an unmanned surface vehicles (USV) moves around a moving target having a circular shape.
Section 4.2 refers to the implementation of the multi-vehicle system in the simulation environment through the ROS framework. The target is static and has an irregular shape. It has been adopted the optimal estimation algorithm along with the control strategy introduced in chapter 5 of [1].

4.1 Preliminary work

In this section we investigate the control of an unmanned surface vehicles (USVs), modelled as a unicycle, for mobile target circumnavigation and tracking, where the target is an algal bloom area. Assuming that the target shape can be approximated to a circle, a protocol based on local measurements provided the USV is developed to estimate the target center and radius. Then a control strategy is derived to bring the vehicle to the target. This has been carried out as a preliminary work to validate the convergence of the estimation and control algorithms.
The overall work relative to this section has been carried out in Matlab/Simulink.

First of all, by neglecting the robot dynamic, we simulate the motion, affected by noise, of the target and of the USV moving around it.
Equations 4.1 and 4.2 describe them respectively.

$$\begin{aligned} x_c(t) &= x_c(t-1) + \alpha(randn + \beta) \\ y_c(t) &= y_c(t-1) + \alpha(randn + \beta) \end{aligned} \tag{4.1}$$

$$\begin{aligned} \theta(t) &= \theta(t-1) + \frac{2\pi}{n} \\ x_{USV}(t) &= x_c(t) + D_i^c(t)cos\theta(t) \\ y_{USV}(t) &= y_c(t) + D_i^c(t)sin\theta(t) \end{aligned} \tag{4.2}$$

where $randn$ is a normally distributed variable, n is the time taken by the USV to make a loop around the target, α and β are adjustable parameters.
Afterwards, estimation algorithms presented in section 3.2, equations 3.7 and 3.8, have been implemented to get $\hat{r}$ and $\hat{c}$, the estimation of the radius and of the coordinates of the center of the target respectively. Then, by means of equation 3.9, the control input can be obtained. Once the estimation is accomplished, the control input can be used to perform the control strategy.
Note that equation 3.9 provides v_x, v_y whereas in the feedback control strategy the reference inputs are v_d and w_d are required. For the control, the robot dynamic has been taken into account. Starting from equation 3.13, model parameters has been chosen to consider its physical characteristics.
Fig. 4.1.1 shows the control strategy scheme. Two main part can be highlighted: the control block which embeds two PIDs and two blocks, the dynamic and kinematic blocks, which describe the nature of the robot model through the use of the parameters previously chosen.
The control block processes the reference inputs and the feedback signals to provide the control inputs to the block of the dynamic model of the system. Then, the kinematic model's block processes the vehicle's velocities provided by the dynamic model block giving as output the coordinates of the vehicle.
At last, these could be compared with the position of the successive vehicle to calculate the angle between the vehicles, β_i. Additionally, the control strategy has been analyzed in the case of model's uncertainty, modeled as a white noise.

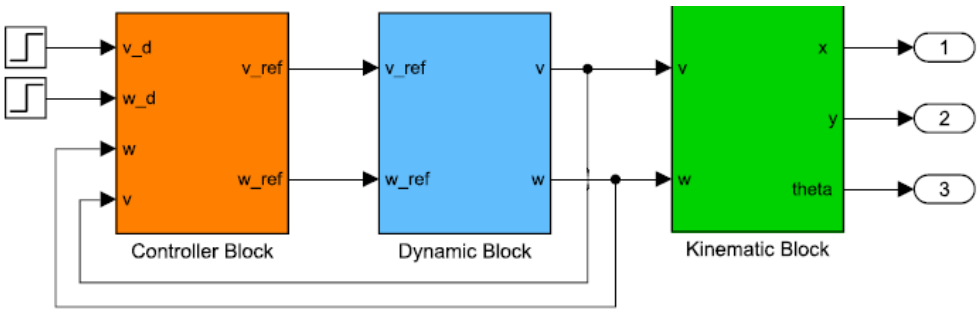


Figure 4.1.1: Control system

4.2 ROS implementation

In this section we discuss a way to simulate the multi-vehicle system presented in section 3.5, with the aim of detecting, tracking and circumnavigation of a target.
The work has been developed in the Robot Operating System (ROS) framework [38], one of the most adopted when dealing with robotics.

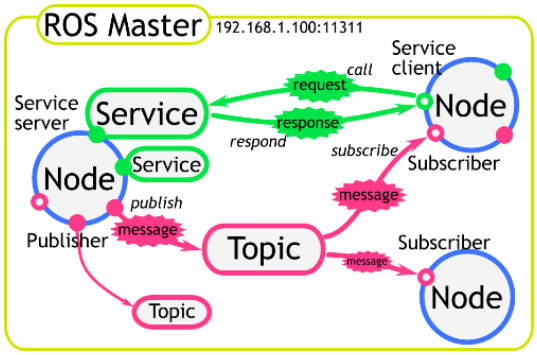


Figure 4.2.1: ROS architecture

ROS is an open-source robotic middleware for the large scale development of complex robotic systems as it provides hardware abstraction, low-level device control, inter-processes message-passing and package management. It provides tools and libraries for obtaining, building, writing and running code across multiple computers. It also allows the manipulation of sensor data of the robot as a labeled abstract data stream called *topic*. Furthermore, it provides high level applications such as arm controllers, face tracking, mapping, localization and path planning.
Fig. 4.2.1 shows the ROS architecture, which is based on the concept of several nodes communicating asynchronously, exchanging information by means of topics.

ROS node is an executable whose code is written in C++, Python and is created for a specific purpose.
Another important tool, essential for robotics application, is the simulator. One of the most jointly used with ROS is Gazebo [39]. It allows to rapidly test algorithms, design robots, perform tests, accurately and efficiently simulate populations of robots in complex environments. Last but not least, it allows to reproduce the physics and dynamic of robots and environments. In this way, it is possible to test the efficiency of systems when disturbances as wind or sensor noise occur.
As discussed in previous sections, the system to reproduce in the ROS/Gazebo environment consists of a UAV and multiple USVs. The target has been chosen accordingly to the goal of the thesis.
In sections 4.2.1, 4.2.2, 4.2.3 and 4.2.4 the work relative to the implementation is discussed.

4.2.1 Target

The target plays a key role in the simulation. In a general scenario, the target is a dynamic shape-changing object. For simplicity, we have decided to start with a static target having a constant shape.

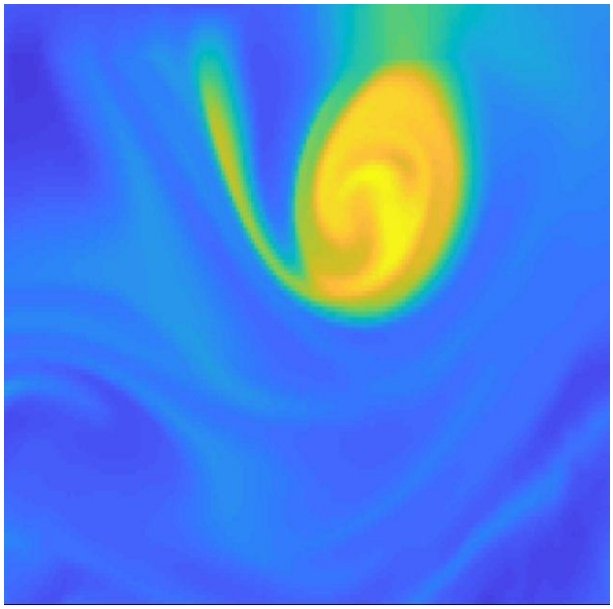


Figure 4.2.2: SINMOD Simulation

Fig. 4.2.2 shows a snapshot of a SINMOD simulation of flagellates near the Norwegian

sea coast [40]. Each pixel is about 100 meters so the image is about 10km in latitude and longitude. The yellow spot indicates a high level of algae concentration.
It is important to highlight that the chosen target has an irregular profile. It could be approximated to a circular shape, i.e. drawing a convex hull, but in this way we could lose or affect data, decreasing the efficiency of the system. Therefore, we have decided to analyze the non convex target, performing estimations directly on it.

4.2.2 Unmanned Aerial Vehicle

The Unmanned Aerial Vehicle, shortly UAV, is the element of the system in charge of detecting the target through a camera. The chosen UAV is the 3DR Iris, as shown in Fig. 4.2.3.
Usually, drones come with an embedded Autopilot and we have decided to rely on the PX4 drone Autopilot because:

- it provide an already existing model of the chosen drone, the 3DR Iris;
- it allows us to perform SITL;
- allows to perform Offboard control.

Note that the drone in the Fig. 4.2.3 is not equipped with a camera. Therefore, without loss of generality, we have imported the model of the default calibrated mono camera provided by Gazebo. It faces downward because the UAV flies over the target.
Thanks to PX4 Autopilot, UAV engines starts and takeoff occurs. We have tuned and set the UAV pose in order to let the camera detect the whole algae bloom spot.

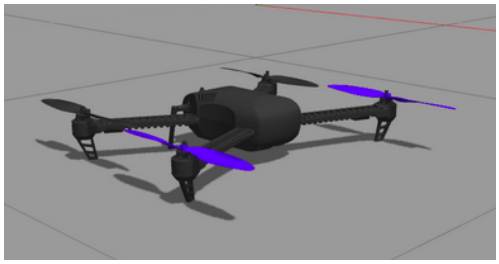


Figure 4.2.3: 3DR Iris

4.2.3 Unmanned Surface Vehicles

Another important element in the whole multi vehicle system is the group of Unmanned Surface Vehicles, or USVs. Their main goal is to circumnavigate the

detected target.
In this work, a group of n Turtlebots Waffle are used to simulate the group of Unmanned Surface Vehicles (USVs).
TurtleBot [41] is a ROS standard platform robot. Turtle is derived from the Turtle robot, which was driven by the educational computer programming language Logo in 1967. In addition, the turtlesim node, which first appears in the basic tutorial of ROS, is a program that mimics the command system of the Logo turtle program. It is also used to create the Turtle icon as a symbol of ROS. TurtleBot has become the standard platform of ROS, which is the most popular platform among developers and students. Fig. 4.2.4 shows 3 types of turtlebots, which differentiate from physical characteristics and embedded sensors. From left to right: Turtlebot3 Burger, Turtlebot3 Waffle and Turtlebot3 Waffle Pi. Even though the Turtlebot is a differential drive robot, its analysis can be simplified to a unicycle, as supposed in section 3.3.
Their purpose is to receive the commands to reach the target, identified through the estimated optimal ellipse, and thus circumnavigate it.

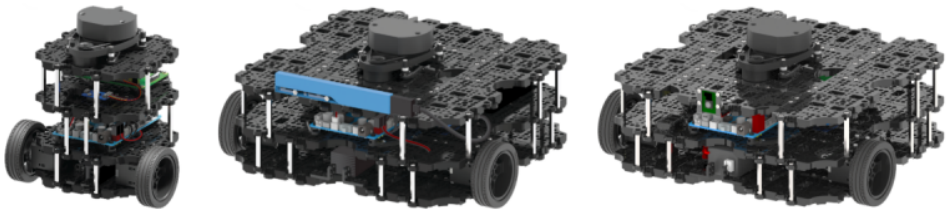


Figure 4.2.4: Turtlebots

4.2.4 Computer Vision

Gazebo is a powerful simulator for testing computer vision algorithms when dealing with autonomous robots moving in 3D space. Our goal is to detect through the camera, mounted on the UAV, the target. The camera streams a video with a frequency of 60Hz. Each frame of the video is processed as if it is an image.
Image processing and target detection have been done through the ROS package CVBridge [42], which converts OpenCV images to ROS Image messages and viceversa, as shown in Fig. 4.2.5.
The image processing algorithm, developed by means of OpenCV functions already introduced in section 3.7, is discussed in the following. It consists in:

- image blurring;

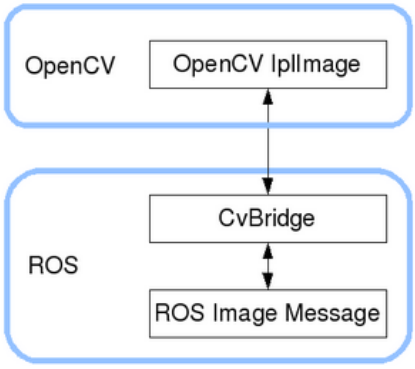


Figure 4.2.5: CVBridge - ROS package

- color space conversion;
- object and edge detection;
- mathematical interpretation of the recognized objects.

Firstly, the blur of the image, captured by the camera, has been applied. This step is needed to delete the noise affecting the image.
Then, the function $cvtColor()$ has been implemented to convert the image from one color space to another. This has been done because usually images are stored in RGB (Red, Blue, Green) or BGR (Blue, Green,Red) color space, where each pixel is represented by the amount of the three colors. However, we use thresholding method which does not require the RGB. Due to this, we change the current color space from RGB to HSV (Hue, Saturation, Value). Fig. 4.2.6 shows the difference between the two color spaces.

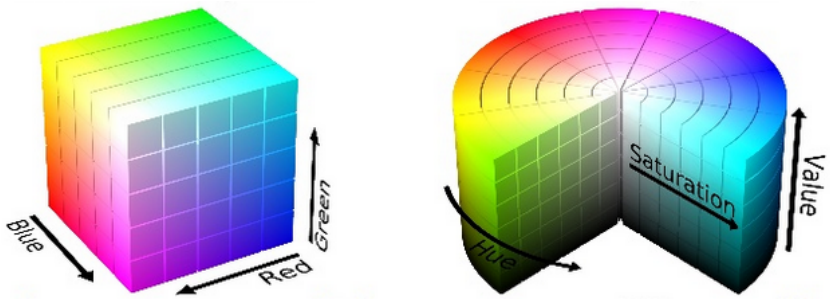


Figure 4.2.6: RGB - HSV color spaces

Afterwards, in order to process the image and perform the object detection, we apply the Otsu's thresholding method by means of $threshold()$ function. A binary image will be returned with a clear identification of the target spot.
Then, Canny filter is applied to perform the edge detection. From the previously

obtained binary image is returned an image showing the detected contours of the algal bloom.

After the detection, now we have to assign a mathematical meaning to the found contours. This is done through the $findContours()$ method. It is suggested to use binary images for better accuracy, applying threshold or canny edge detection.

This function allows us to check the number of detected contours and their pixel's coordinates in the image frame. The same holds for the identification of the center of the contour's object, which is mathematically calculated. Moreover it allows to assign an hierarchy accordingly to parameters as area, perimeters, etc, to decide the order of contour analysis.

Note that the identified pixels corresponding to contours points and center point are related to the image frame. Therefore, by exploiting the Pinhole camera model, discussed in section 3.7.1, equation 3.14, and simple reference frame transformations, we have obtained the coordinates of the points of interest in the world frame.

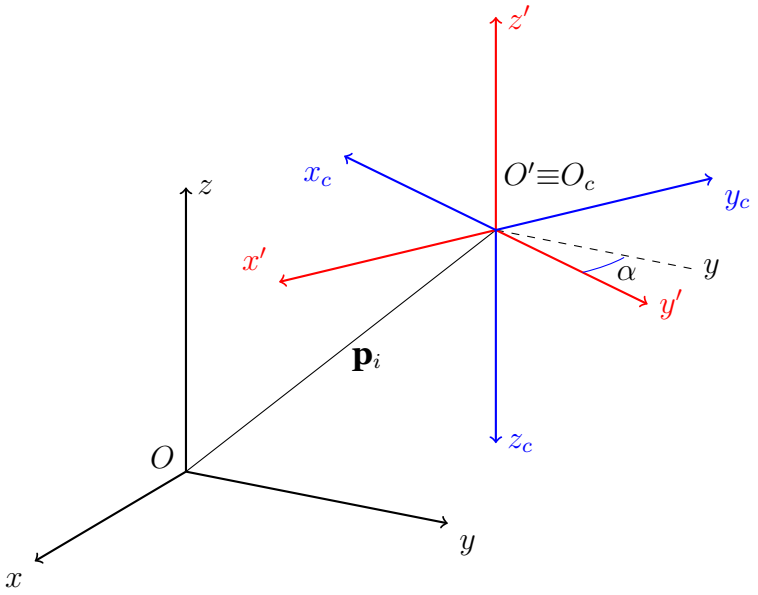


Figure 4.2.7: Reference frames

Fig. 4.2.7 shows the reference frames where:

- R_W=$\{O, x, y, z\}$ is the fixed world reference frame;
- R_{UAV}=$\{O, x', y', z'\}$ is the mobile UAV reference frame;
- R_C=$\{O, x_C, y_C, z_C\}$ is the camera reference frame;

In this way, we find the position of the contour and center coordinates in the camera frame R_C. Then, through the homogeneous transformation expressed in matrix form

in 4.4, we get the coordinates of these points in the world reference frame R_W.

$$\begin{bmatrix} X_W \\ Y_W \\ Z_W \\ 1 \end{bmatrix} = T_C^W \begin{bmatrix} X_C \\ Y_C \\ Z_C \\ 1 \end{bmatrix} \tag{4.3}$$

$$T_C^W = \begin{bmatrix} -s_\alpha & -c_\alpha & 0 & x_i \\ -c_\alpha & s_\alpha & 0 & y_i \\ 0 & 0 & -1 & z_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{4.4}$$

where $\mathbf{p}_i$=(x_i, y_i, z_i) is the UAV position in the 3D space, α is the rotation angle with respect to the z' axis, assuming that only yaw motion can happen, $s(\cdot)$ and $c(\cdot)$ stand respectively for sin and cos functions and $[X_C, Y_C, Z_C] \equiv [X, Y, Z]$ expressed in equation 3.14.

4.2.5 Optimal Estimation

The optimal ellipse estimation concerns the approximation of the irregular target shape to an ellipse, given as many 2D points as the number of the n USVs involved.
In the following, it is explained the process that each USV performs to provide the data needed for the optimal estimation.
Once the contour points have been defined in the world reference frame, the optimal ellipse estimation can begin.
Each USV, whether it is inside or outside the bloom, constantly measures the distance between its position and all the points of the target's boundary. The chosen point for the estimation process is given by the one providing the minimum distance.
Fig. 4.2.8 shows the measurement process done by each USV to determine the contour point to be picked for the optimal estimation. In this case, the red point, b_j, with $1 < j < m$, is the selected contour point provided from the USV for estimation of the optimal ellipse.
The described process can be mathematically expressed by the equation 4.5

$$\min \mathbf{d}(t) = \min_{\forall i \in [1,\dots,m]} \|\mathbf{p}_i(t) - \mathbf{b}_i(t)\| \tag{4.5}$$

where $\mathbf{d}$ is the vector containing all the time-varying distances, $\mathbf{b}_i$ is the time-varying 2D position of the i-th boundary point and $\mathbf{p}_i$ is the position of the USV.

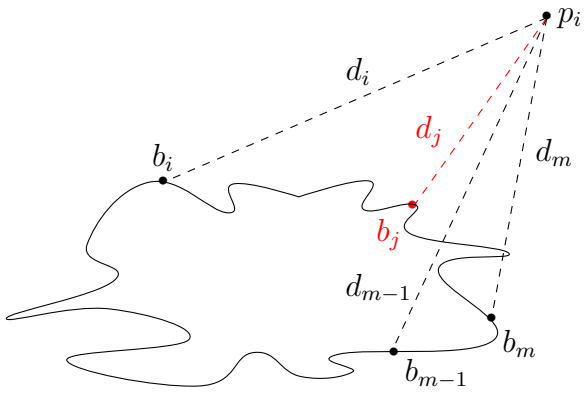


Figure 4.2.8: Determination of contour point for optimal estimation

Since the multi-vehicle system consists of n USVs, n contour points will be selected and given as input of the function discussed in 3.7.4, used for the fitting of the ellipse.

4.2.6 Formation and Circumnavigation

This work presents a control algorithm for composing a specific robot formation and perform circumnavigation of the detected target.
In this field, vehicles can be named *swarm robots*. The swarm robots are expected to arrange, evenly spaced, around the estimated ellipse.
At the beginning, the robots are randomly placed around the target so that the optimal estimation can be performed. Once an estimate is obtained, formation process can begin.
The chosen formation technique concerns the leader-referenced approach.
A specific robot called ”leader” is chosen among the swarm. The goal of this robot is to determine the position of the other USVs around the target.
In this work, the leader is simply chosen as the first robot listed by the algorithm in the swarm list. The leader, then, chooses the first ”leader’s follower”, or simply ”follower”, which will be the leader’s closest robot. Iteratively, the robots are tidied up according to this simple working principle.
This specific order is very useful when circumnavigation is performed since it allows to fulfill the evenly spaced constraint. In this way, each front of the target area is analyzed, avoiding the gathering of two or more robots in the same area.

After each USV has reached its own target goal, established by the leader, circumnavigation is performed.
Circumnavigation consists in the clockwise motion of the robot swarm, laying above the estimated optimal ellipse, around the target.
The velocity command assigned to each USV depends on its distance from the estimated ellipse boundary, and thus indirectly to the distance from the target center. Moreover, it also depends on its distance from the previous and next USVs.

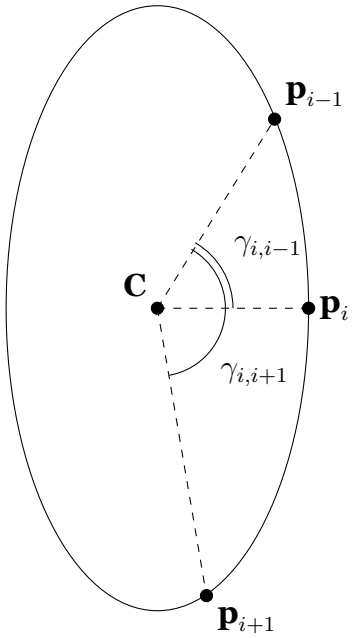


Figure 4.2.9: Determination of velocity command

As shown in Fig. 4.2.9, each vehicle determines the distance from the previous and next vehicles, in order to get a velocity command suitable for the fulfillment of the evenly spaced constraint. In this work, we refer to the control strategy expressed in chapter 5.2 of [1].
The chosen control law is:

$$\mathbf{u}_i = D_i\psi_i + \frac{D_{i,i+1}}{D_{i,i-1}}E\psi_i \tag{4.6}$$

where:

$$\begin{aligned} D_{i,i+1} &= \mathbf{p}_i - \mathbf{p}_{i+1}, \qquad i = 1, ..., n-1 \\ D_{i,i-1} &= \mathbf{p}_i - \mathbf{p}_{i-1}, \qquad i = 2, ..., n \\ D_{n,1} &= \mathbf{p}_n - \mathbf{p}_1 \end{aligned}$$

The heading angle is defined by ψ_i, which defines the direction of each vehicles towards the center of the target. Matrix E stands for a rotation of 90° around the z axis.
The velocity command is therefore calculated and applied to the USVs in order to reduce the difference between two consecutive angles, $\gamma_{i,i+1}$ and $\gamma_{i,i-1}$.
In this work, we have fulfilled this requirement as:

$$\gamma_{i,i+1} \approx \gamma_{i,i-1} \qquad \Rightarrow \qquad \gamma_{i,i+1} - \gamma_{i,i-1} \leq \eta \tag{4.7}$$

where η is a tunable angle representing a threshold angle.

Chapter 5

Simulation results

In this chapter, the simulation results are presented, obtained by implementing the algorithms, and an acute assessment through an accurate analysis is carried out.
Section 5.1 refers to the preliminary work developed in the Matlab/Simulink environment. The obtained outcomes, both the ideal and the noise corrupted case, are displayed and discussed.
Section 5.2 refers to the main part of this work, which concerns the implementation of the multi-vehicle system in the simulation environment.
The outcomes of the computer vision algorithm, the result of the optimal estimate, of the circumnavigation and formation allow us to evaluate the developed work.

5.1 Preliminary work's results

We disclose the results obtained from the simulation of the feedback control strategy applied to the robot's model. Fig. 5.1.1 shows the heading and angular velocities of the vehicle, expressed respectively in m/s and rad/s, where:

$$v_d = 0.5 \qquad w_d = 0.5$$
$$v(0) = 0.1 \qquad w(0) = 0.2$$

The transients show no overshoot and we achieve null steady state tracking error in the heading velocity whereas there is a small but negligible in the angular velocity plot. Below the plot of the velocities, their relative control input has been shown. Of course this is only a qualitative response of the system but further analysis has to be done,

taking into account the physical characteristics and limits of the model on which this control strategy will be implemented.

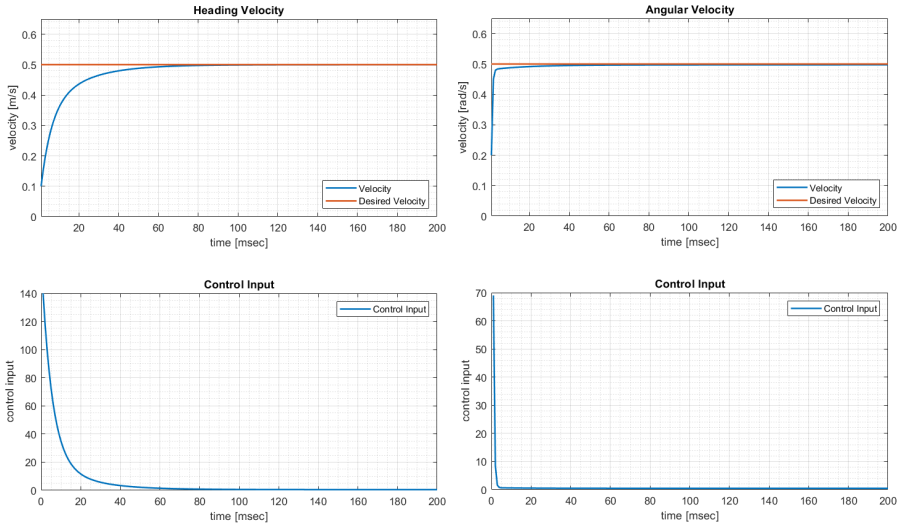


Figure 5.1.1: Heading and Angular Velocity

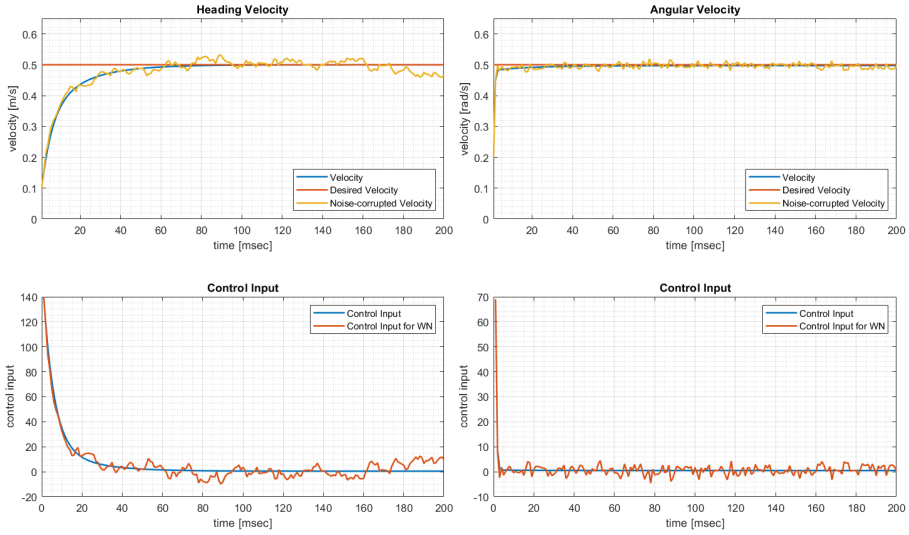


Figure 5.1.2: Heading and Angular Velocity corrupted by WN

Then, we analyze the response of a noise-corrupted system so that we can check if the overall system is robust.

Fig.5.1.2 shows that, even by adding a white-noise to the mathematical model expressed in the equation 3.13, the control system response is bounded in an acceptable range.

The effect of the noise acting on the kinematic block can be also analyzed. Indeed,

Fig. 5.1.3 shows that also the coordinates and heading angle of the robot are not badly affected by the white noise entering the system as additive noise.

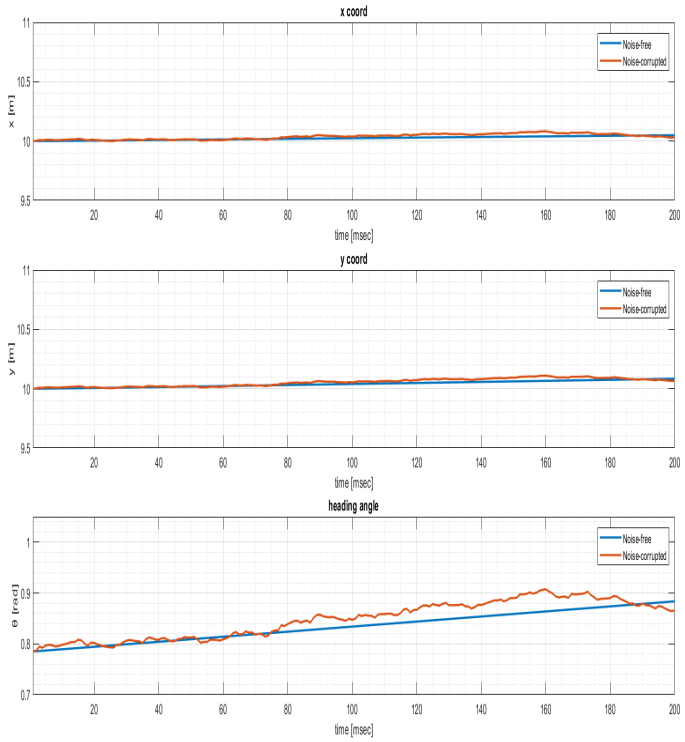


Figure 5.1.3: Coordinates and heading angle corrupted by WN

In conclusion, acceptable results have been obtained from this preliminary, yet qualitative, work.

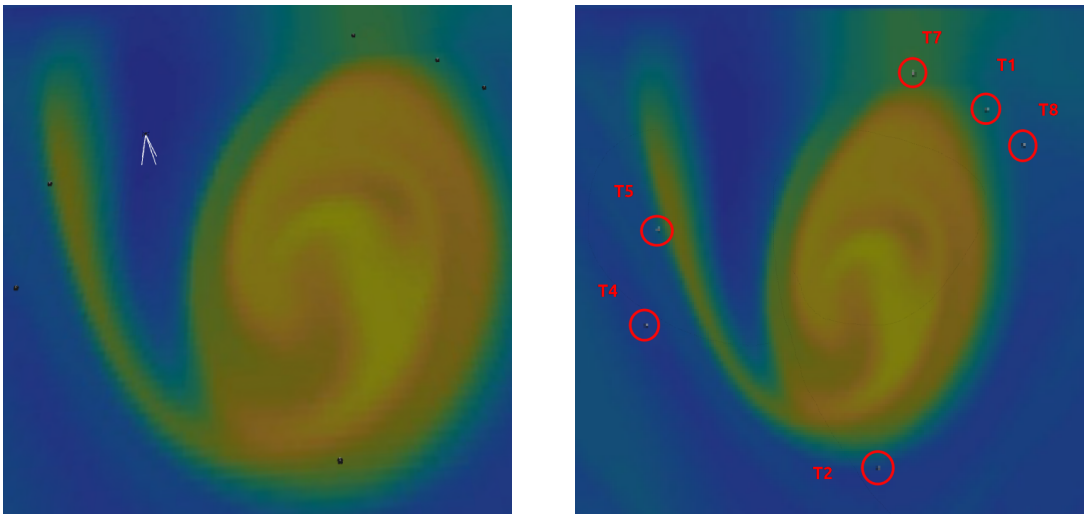
5.2 ROS/Gazebo implementation's results

The implementation of the algorithm in the simulated environment, provided by Gazebo simulator, leads to the creation of the world depicted in Fig. 5.2.1. Here we can identify the target, the UAV, the 3DR Iris, and six USVs. The number of agents employed in the simulation can vary and this can improve the accuracy of the estimation and enhance the performance of the circumnavigation. A takeoff command can be given to the UAV in off-board mode, setting the desired pose. We set the UAV position as $\mathbf{p}_{UAV}$ = (-6, -1, 48).

Thus, we refer to the USVs pose as $\mathbf{T}_i$, which have a randomly chosen 2D initial position, since the z coordinate is null:

$\mathbf{T}_1$ = (6, -21), $\mathbf{T}_2$ = (-24, -12), $\mathbf{T}_4$ = (-12, 7), $\mathbf{T}_5$ = (-4, 6), $\mathbf{T}_7$ = (9, -15), $\mathbf{T}_8$ = (3, -24).

For sake of simplicity we have used the target image on a smaller scale, reducing it of scale factor equal to 1:250.

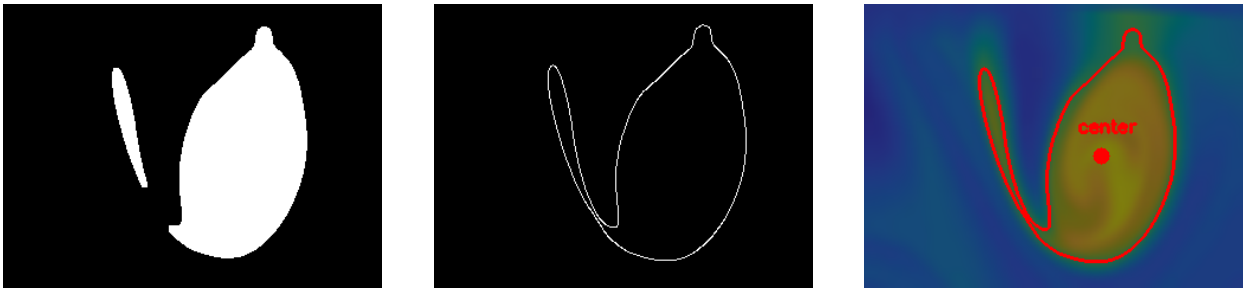


(a) External view (b) UAV camera view

Figure 5.2.1: World

5.2.1 Computer vision's results

The camera mounted on the 3DR Iris UAV allows to constantly process the images and analyze the target. Indeed, by means of the computer vision functions presented in section 3.7, object and edge detection are carried out.



(a) Object Detection (b) Edge Detection (c) Object Interpretation

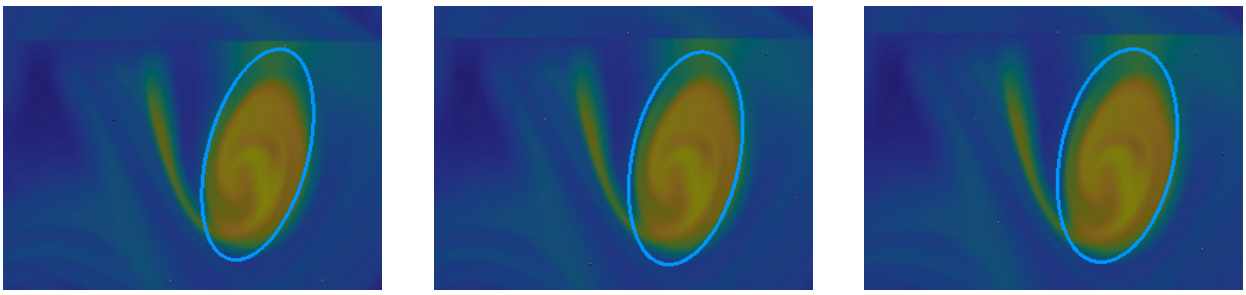
Figure 5.2.2: Optimal estimation with USVs covering one side

This leads firstly to the detection of the bloom mask (Fig. 5.2.2a), in which there is the separation of the object of interest from the background. Then, only the bloom profile is obtained through the edge detection, by applying the Canny filter (Fig. 5.2.2b).

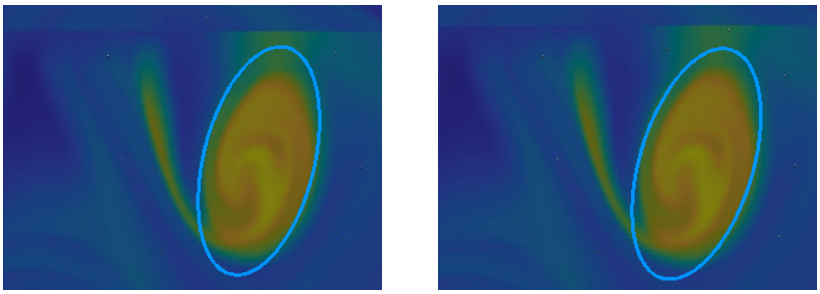
Here, we get only a contour having no mathematical meaning. To achieve the last part of the computer vision algorithm, the object interpretation has to be carried out, by identifying the contour and center information (Fig. 5.2.2c).
In conclusion, we can state that the OpenCV functions allows us to perfectly analyze and detect, in simulated environment, the target, albeit its complex shape. It is important also to remember that errors in the detection of the target can occur, or caused by the environment (wind, sun, etc) or by an unexpected failure of the camera. This would prevent the correct working of the detection algorithm, by blocking the whole system.

5.2.2 Optimal Estimation's results

The optimal estimation is an important key point of the whole algorithm and an error in the estimate of the target can affect the smooth functioning of the system.
Thus, a feature to carefully take under examination is the accuracy of the estimation process.
The accuracy of the estimation may depends on the number of USVs involved in the measurement process, whether they are inside or outside the algal bloom, but also on their position with respect to the target. A further analysis has been carried out by varying the number and position of the USVs.
An evident difference can be highlighted by comparing Fig. 5.2.3 and Fig. 5.2.4 with Fig.5.2.5. The comparison is based on the arrangement of the vehicles with respect to the target.
The first two figures show the target estimations when the robots are placed all around it, without taking into account the constraint of spacial equality. On the contrary, Fig.5.2.5 shows the result of the estimation process when the robots cover only one front of the target.
While the first two figures show reasonable estimations, in the third one the estimate is highly affected by the position of the vehicles with respect to the target, resulting in a worsen of estimation, which affects the performance of the algorithm.
Another comparison can be carried out by analysing the estimation results in the case of evenly spaced or unevenly spaced USVs.
For this purpose, Fig. 5.2.3 and Fig. 5.2.4 show the estimation results of the optimal ellipse with a varying number of employed vehicles, by analysing the effect of their arrangement around the target.

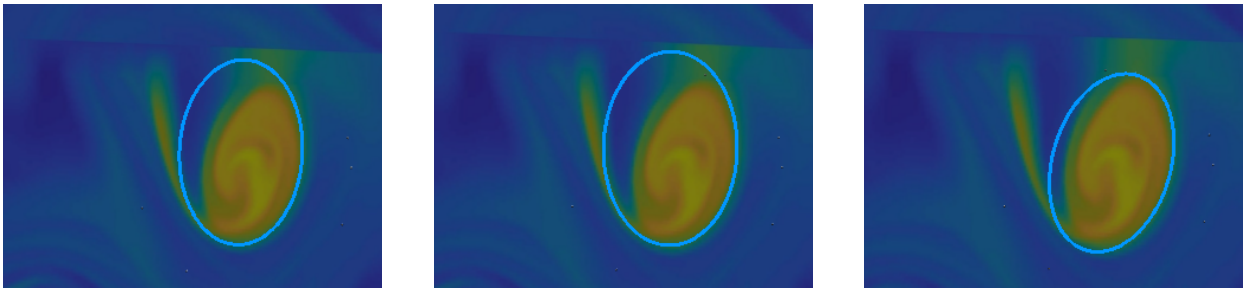


(a) 5 USVs (b) 6 USVs (c) 7 USVs

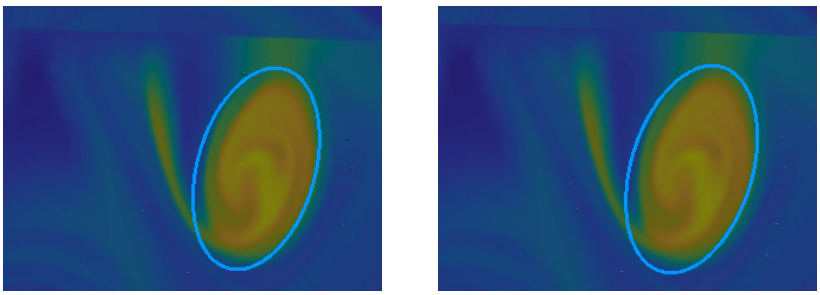


(d) 8 USVs (e) 9 USVs

Figure 5.2.3: Optimal estimation with evenly spaced USVs



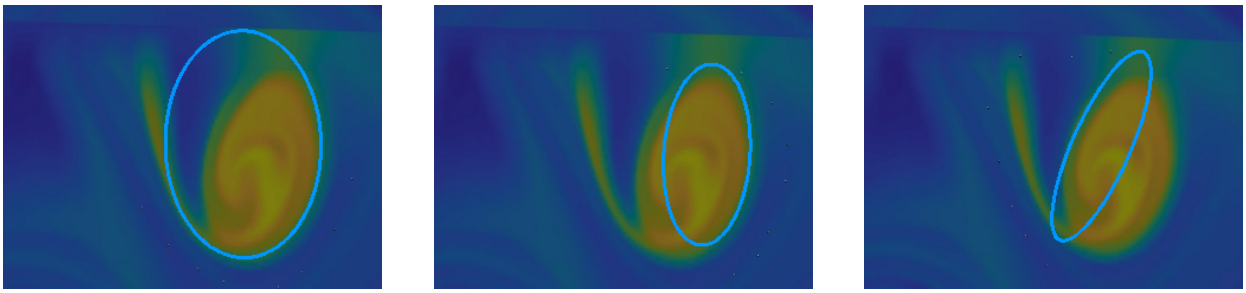
(a) 5 USVs (b) 6 USVs (c) 7 USVs



(d) 8 USVs (e) 9 USVs

Figure 5.2.4: Optimal estimation with unevenly spaced USVs

By comparing the figures, it is possible to state that a limited number of USVs deployed around the algal bloom can return a non accurate estimate of the target, as shown in Fig. 5.2.4a and Fig. 5.2.4b.



(a) 9 USVs (b) 9 USVs (c) 9 USVs

Figure 5.2.5: Optimal estimation with USVs covering one side

The estimation results slightly changes when the number of vehicles increases, meaning that the spacial equality constraint can be overcome by increasing the number of deployed robots.
In a real implementation, even though performance and accuracy can benefit from it, the increasing number of used robots can affect the cost of the mission, taking into account that a large algal bloom may required a huge number of vehicles. Therefore, a good trade-off among accuracy, performance and costs is highly suggested.

5.2.3 Formation and Circumnavigation's results

The estimation process is followed by a further process consisting of two parts: formation and circumnavigation. The former is responsible of the arrangement of the vehicles around the target by fulfilling the evenly spaced constraint, the latter is relative to the clockwise circumnavigation of the target.
The formation process, as explained in section 4.2.6 begins as soon as the first target estimate is obtained.
Fig. 5.2.6 shows the evolution of the multi-vehicle system while accomplishing the formation task.
$T8$ is the leader and, accordingly to the closest robot principle, the other vehicles are listed in the following order: $T1$, $T7$, $T5$, $T4$, $T2$.
As can be seen from Fig. 5.2.6, USVs arrange counter-clockwise in this specific order. Note that the order assigned during the formation process is preserved and also the evenly spaced constraint is always guaranteed.

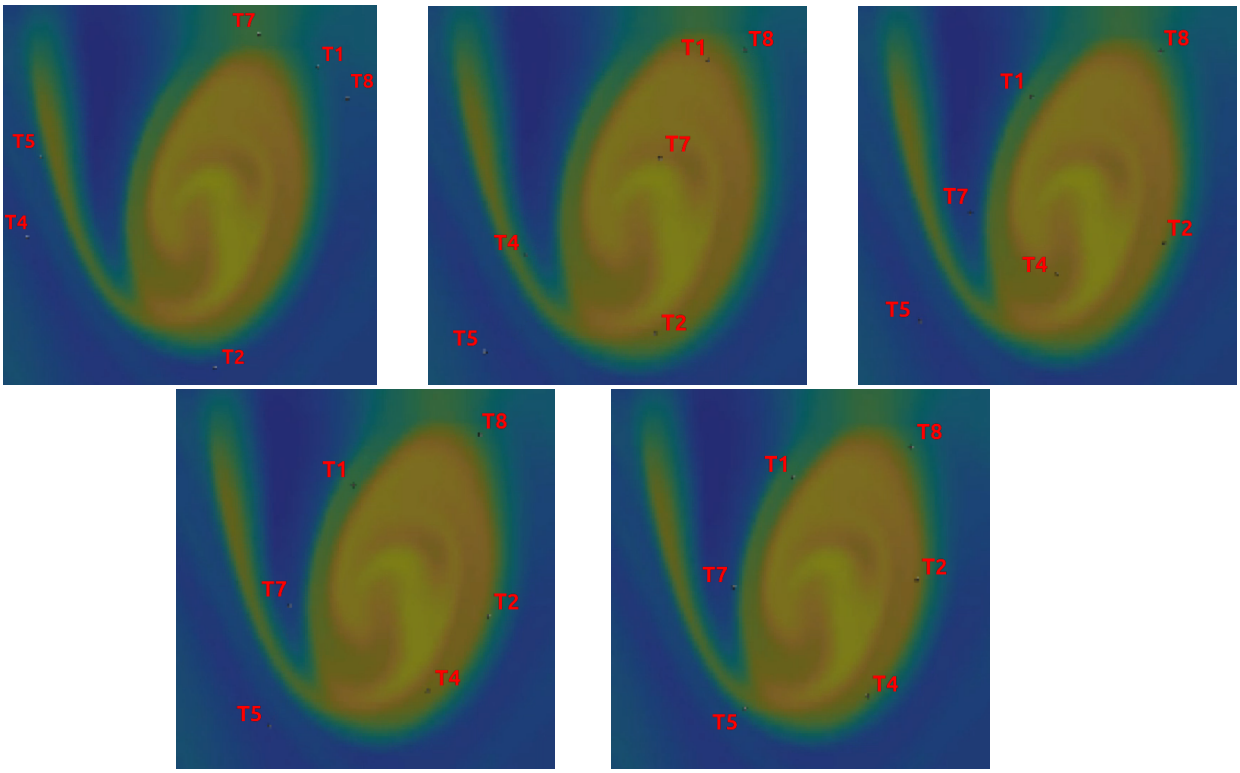


Figure 5.2.6: Formation

Once all the USVs reach the target goal assigned by the leader, and they are close enough to the optimal ellipse estimated, the circumnavigation process can begin. This occurs only if the condition expressed in the equation 4.7 is satisfied. If not, the USVs arrange themselves in order to fulfill the constraint.

The input velocity of each USV is described by the equation 4.6.

Fig. 5.2.7 represents 5 consecutive instants in which is depicted the clockwise circumnavigation carried out by the robots around the target.

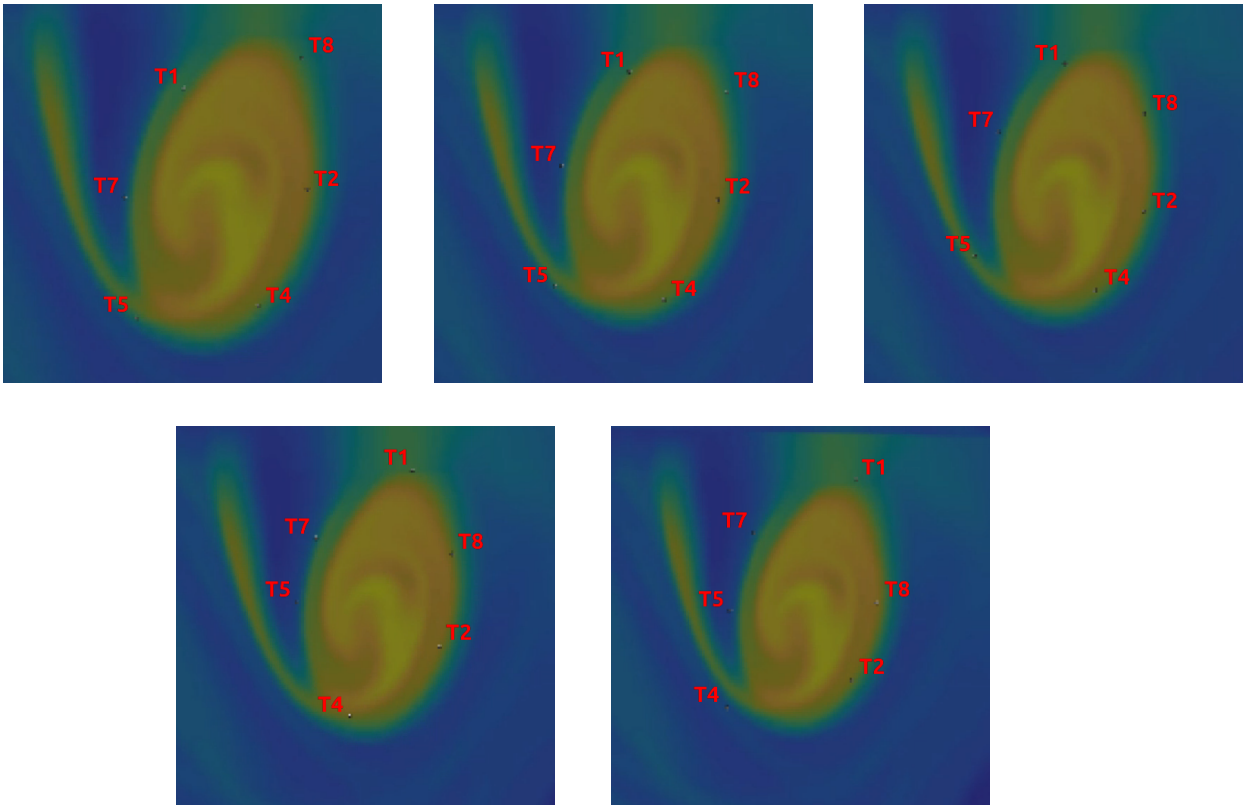


Figure 5.2.7: Circumnavigation

As a result, target circumnavigation is performed by the agents by preserving the formation and always respecting the constraint.

Chapter 6

Conclusions

This degree project has at its core the development of an algorithm to demonstrate that collective behaviour among vehicles can occur, aiming to the accomplishment of a common goal.
The purpose determined at the beginning of this degree project has been fully achieved, positively demonstrating the correct working and absolute reliability of the algorithm. In the first part, it has been demonstrated that the estimation performed by the multi-vehicle system achieves the convergence also in the case of corrupted robot model, and in the second part, at least in simulation, that a multi-vehicle system can be employed for the detection and tracking of an irregular shape target.
From a technical point of view, this work has contributed to confirm that specific tools as OpenCV and ROS, widely used in robotics, are suitable for our purpose. Indeed, thanks to them, in first place we have been able to easily manage the hardware elements composing our systems, the UAV ans USVs. Secondly, we have also developed software to perform image processing, coordinate and connect each entity of the multi-vehicle system with the aim of fulfilling the prefixed goal.
From a theoretical point of view, we have further confirmed the excellent work done by [1], realizing, although in simulation environment, what has been introduced only in theory, neglecting robots' dynamics and environmental variables.
Furthermore, this work has allowed a deep analysis of the good results obtained from the simulation, but also an investigation on the side effects that can be encountered when dealing with this kind of algorithm in this specific scenario.
We have presented the pros and weakness of estimation process, which performance can be worsen in the presence of few vehicles or not suitably arranged robots. When

this is avoided, estimates are good enough to perform an excellent circumnavigation.
We have also taken a look at the working of the formation and circumnavigation processes. They rely on the target estimation, which ignites the whole process and thus their performance are strictly bounded to it. When a good estimation of the target is provided to the system, robots formation guided by a leader takes place and after multi-vehicle circumnavigation can occur.
Moreover, the execution velocity of these two processes can be tuned and can depend on many features. Indeed, the overall system can be affected by many variables such as number of deployed vehicles or robots position with respect to the target. Besides, also dynamic or generic quantities of the environment (wind, waves, obstacles, the effect of the sun) rather than physical properties of robots (noisy sensors, damages, etc), neglected in this degree project, can affect the correct working of the algorithm.
In conclusion, we are satisfied of the obtained results, being the first ever work developed in this field, which aims at dealing with a problem that has long been ignored even though it affects many lives, both the animal kingdom and the human species.
The results obtained in this degree project are meant to encourage other students and researchers to carry out further studies to improve the developed algorithms. This would allow local measurement and tracking of target such as the HABs in hazardous environments, the reduction of mission costs and an increasing in the number of monthly measurement missions, as well as an improvement in efficiency, reliability and performance.

6.1 Future Works

Collective behaviours concerns an interesting and innovative topic that will be at the center of the future technologies in many domains (marine, military, aeronautic, agriculture, etc).
Core of a challenging research engineer work in the field of cooperative multi-vehicle system can be the continuation of this advanced implementation.
A cutting-edge research work can lead to be pioneer in this field, gaining precious individual skills for the development of an application aimed at the human-robot cooperation.
Being a newly discovered field, there is plenty of implementations with a wide range of possible solutions.

On the basis of the work developed in this degree project, possible future works may concern a further development of the estimation algorithms, taking into account all the non-ideal quantities that can be present in a real dynamic target. In this thesis, only the case of a static target has been shown. An implementation in a simulated environment to analyzed a scenario with a dynamic irregular shape target can add an exceptional contribution, improving significantly the performance of the system, helping to better understand the limits and advantages of such a multi-vehicle system.

Key element of a further simulation, perhaps by using the UUV simulators, may be the analysis of the effect of waves and sea currents on the dynamic of the target and on the overall system. It would be useful to determine how they affect the performance and the behaviour of the system.

Another implementation can include the use of multiple UAVs when the target's size is extremely huge, and it is quite difficult to analyze and process all the information with a single UAV. This implementation would involve the merging of data coming from two different sources, then their processing to finally provide a good information to the USVs.

On the other hand, beyond other simulations, a physical implementation, in lab or outdoor, is highly suggested to fully validate the results obtained in simulation environment. This would be very challenging due to the hardware implementation, both on UAV and USVs, of the developed algorithms and the management of all the vehicles comprising the system.

Bibliography

[1] Gouveia Fonseca, Joana F. “Cooperative Multi-Vehicle Circumnavigation and Tracking of a Mobile Target”. Kungliga Tekniska högskolan, 2020.

[2] *National Oceanic and Atmospheric Administration - NOOA*. URL: `https://www.noaa.gov/what-is-harmful-algal-bloom`.

[3] Wells, Mark L, Trainer, Vera L, Smayda, Theodore J, Karlson, Bengt SO, Trick, Charles G, Kudela, Raphael M, Ishikawa, Akira, Bernard, Stewart, Wulff, Angela, Anderson, Donald M, et al. “Harmful algal blooms and climate change: Learning from the past and present to forecast the future”. In: *Harmful algae* 49 (2015), pp. 68–93.

[4] Wassmann, Paul, Slagstad, Dag, Riser, Christian Wexels, and Reigstad, Marit. “Modelling the ecosystem dynamics of the Barents Sea including the marginal ice zone: II. Carbon flux and interannual variability”. In: *Journal of Marine Systems* 59.1-2 (2006), pp. 1–24.

[5] Mohammad Shahab, ID227598. “Cooperative Control of Multi-Vehicle Systems”. In: *KFUPM, Dhahran, Saudi Arabia* (2008).

[6] Murayama, Toru. “Online trajectory planning method for multi-vehicle system considering network connectivity and collision avoidance simultaneously”. In: *SICE Journal of Control, Measurement, and System Integration* 8.1 (2015), pp. 15–21.

[7] Miki, Takahiro, Khrapchenkov, Petr, and Hori, Koichi. “UAV/UGV autonomous cooperation: UAV assists UGV to climb a cliff by attaching a tether”. In: *2019 International Conference on Robotics and Automation (ICRA)*. IEEE. 2019, pp. 8041–8047.

[8] Mathews, Nithin, Christensen, Anders Lyhne, O'Grady, Rehan, and Dorigo, Marco. "Spatially targeted communication and self-assembly". In: *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE. 2012, pp. 2678–2679.

[9] Lucieer, Arko, Turner, Darren, King, Diana H, and Robinson, Sharon A. "Using an Unmanned Aerial Vehicle (UAV) to capture micro-topography of Antarctic moss beds". In: *International journal of applied earth observation and geoinformation* 27 (2014), pp. 53–62.

[10] Millet Plumet, Dern. "Autonomous surface vehicle for oceanographic survey". In: *International Autonomous Surface Ship Symposium*. 2008.

[11] Jones, Daniel OB, Gates, Andrew R, Huvenne, Veerle AI, Phillips, Alexander B, and Bett, Brian J. "Autonomous marine environmental monitoring: Application in decommissioned oil fields". In: *Science of the total environment* 668 (2019), pp. 835–853.

[12] Kermorgant, Olivier. "A dynamic simulator for underwater vehicle-manipulators". In: *International Conference on Simulation, Modeling, and Programming for Autonomous Robots*. Springer. 2014, pp. 25–36.

[13] Dhurandher, Sanjay K, Misra, Sudip, Obaidat, Mohammad S, and Khairwal, Sushil. "UWSim: A simulator for underwater sensor networks". In: *Simulation* 84.7 (2008), pp. 327–338.

[14] *Unmanned Underwater Vehicle Simulator*. URL: `https://uuvsimulator.github.io/`.

[15] Manhães, Musa Morena Marcusso, Scherer, Sebastian A, Voss, Martin, Douat, Luiz Ricardo, and Rauschenbach, Thomas. "UUV simulator: A gazebo-based package for underwater intervention and multi-robot simulation". In: *OCEANS 2016 MTS/IEEE Monterey*. IEEE. 2016, pp. 1–8.

[16] Antonelli, Gianluca, Arrichiello, Filippo, Casalino, Giuseppe, Chiaverini, Stefano, Marino, Alessandro, Simetti, Enrico, and Torelli, Sandro. "Harbour protection strategies with multiple autonomous marine vehicles". In: *International Workshop on Modelling and Simulation for Autonomous Systems*. Springer. 2014, pp. 241–261.

[17] Ji, Meng and Egerstedt, Magnus. “Distributed coordination control of multiagent systems while preserving connectedness”. In: *IEEE Transactions on Robotics* 23.4 (2007), pp. 693–703.

[18] Balch, Tucker and Arkin, Ronald C. “Behavior-based formation control for multirobot teams”. In: *IEEE transactions on robotics and automation* 14.6 (1998), pp. 926–939.

[19] Barca, Jan Carlo, Sekercioglu, A, and Ford, Adam. “Controlling formations of robots with graph theory”. In: *Intelligent Autonomous Systems 12*. Springer, 2013, pp. 563–574.

[20] Van Parys, Ruben and Pipeleers, Goele. “Distributed MPC for multi-vehicle systems moving in formation”. In: *Robotics and Autonomous Systems* 97 (2017), pp. 144–152.

[21] Dimarogonas, Dimos V and Johansson, Karl H. “On the stability of distance-based formation control”. In: *2008 47th IEEE Conference on Decision and Control*. IEEE. 2008, pp. 1200–1205.

[22] Cao, Ming, Morse, A Stephen, Yu, C, Anderson, Brian DO, and Dasguvta, S. “Controlling a triangular formation of mobile autonomous agents”. In: *2007 46th IEEE Conference on Decision and Control*. IEEE. 2007, pp. 3603–3608.

[23] Sun, Zhiyong. *Cooperative coordination and formation control for multi-agent systems*. Springer, 2018.

[24] Deghat, Mohammad, Shames, Iman, Anderson, Brian DO, and Yu, Changbin. “Localization and circumnavigation of a slowly moving target using bearing measurements”. In: *IEEE Transactions on Automatic Control* 59.8 (2014), pp. 2182–2188.

[25] Boccia, Antonio, Adaldo, Antonio, Dimarogonas, Dimos V, Bernardo, Mario di, and Johansson, Karl H. “Tracking a mobile target by multi-robot circumnavigation using bearing measurements”. In: *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*. IEEE. 2017, pp. 1076–1081.

[26] Sen, Arijit, Sahoo, Soumya Ranjan, and Kothari, Mangal. “Circumnavigation on multiple circles around a nonstationary target with desired angular spacing”. In: *IEEE Transactions on Cybernetics* (2019).

[27] Wang, Zongyao and Gu, Dongbing. “Cooperative target tracking control of multiple robots”. In: *IEEE Transactions on Industrial Electronics* 59.8 (2011), pp. 3232–3240.

[28] Zhong, Hang, Wang, Yaonan, Miao, Zhiqiang, Tan, Jianhao, Li, Ling, Zhang, Hui, and Fierro, Rafael. “Circumnavigation of a moving target in 3D by multi-agent systems with collision avoidance: an orthogonal vector fields-based approach”. In: *International Journal of Control, Automation and Systems* 17.1 (2019), pp. 212–224.

[29] Shao, JingPing and Tian, Yu-Ping. “Multi-target localisation and circumnavigation by a multi-agent system with bearing measurements in 2D space”. In: *International Journal of Systems Science* 49.1 (2018), pp. 15–26.

[30] Martins, Felipe N, Celeste, Wanderley C, Carelli, Ricardo, Sarcinelli-Filho, Mário, and Bastos-Filho, Teodiano F. “An adaptive dynamic controller for autonomous mobile robot trajectory tracking”. In: *Control Engineering Practice* 16.11 (2008), pp. 1354–1363.

[31] De La Cruz, Celso and Carelli, Ricardo. “Dynamic modeling and centralized formation control of mobile robots”. In: *IECON 2006-32nd Annual Conference on IEEE Industrial Electronics*. IEEE. 2006, pp. 3880–3885.

[32] *PX4 Autopilot*. URL: `https://docs.px4.io/master/en/getting_started/px4_basic_concepts.html`.

[33] *OpenCV - Open Computer Vision Library*. URL: `https://opencv.org/`.

[34] *Camera calibration and 3D reconstruction*. URL: `https://docs.opencv.org/2.4/modules/calib3d/doc/camera_calibration_and_3d_reconstruction.html`.

[35] *Image Thresholding*. URL: `https://docs.opencv.org/master/d7/d4d/tutorial_py_thresholding.html`.

[36] *Canny Edge Detector*. URL: `https://docs.opencv.org/2.4/doc/tutorials/imgproc/imgtrans/canny_detector/canny_detector.html`.

[37] *Fit Ellipse*. URL: `https://docs.opencv.org/2.4/modules/imgproc/doc/structural_analysis_and_shape_descriptors.html?highlight=fit%20ellipse#cv2.fitEllipse`.

[38] *Robot Operating System - ROS*. URL: http://wiki.ros.org/ROS/Tutorials.

[39] *Gazebo Simulator*. URL: http://gazebosim.org/.

[40] *SINMOD Simulation data*. URL: https://www.sintef.no/en/ocean/initiatives/sinmod/.

[41] *Turtlebot3*. URL: https://emanual.robotis.com/docs/en/platform/turtlebot3/overview/.

[42] *vision_opencv*. URL: http://wiki.ros.org/cv_bridge.