Visual servoing on deformable objects : an application to tether shape control

Matheus Laranjeira Moreira

UNIVERSITÉ DE TOULON

RÉGION SUD PROVENCE ALPES CÔTE D'AZUR

SUBSEA TECH Marine and underwater technologies

ÉCOLE DOCTORALE 548 - MER ET SCIENCES

Laboratoire COSMER EA 7398

THÈSE

présentée par

Matheus LARANJEIRA MOREIRA

soutenue publiquement le : 26 avril 2019

pour obtenir le grade de Docteur en automatique, signal, productique, robotique

Visual Servoing on Deformable Objects

An Application to Tether Shape Control

JURY :

M. CHAUMETTE François – Directeur de Recherche, Inria	Rapporteur
M. JAULIN Luc – Professeur, ENSTA Bretagne	Rapporteur
M. PETILLOT Yvan – Professeur, Heriot-Watt University	Président
Mme CADENAT Viviane – MCF-HDR, Université de Toulouse III	Examinatrice
M. CREUZE Vincent – MCF-HDR, Université de Montpelier	Examinateur
M. CHERUBINI Andrea – MCF-HDR, Université de Montpelier	Examinateur
M. HUGEL Vincent – Professeur, Université de Toulon	Directeur de thèse
Mme DUNE Claire – MCF, Université de Toulon	Co-encadrement

Membres invités :
M. NAVARRO-ALARCON David – Assistant Professor, The Hong Kong Polytechnic University
M. CHARDARD Yves – Directeur de la société Subsea Tech
M. OPDERBECKE Jan – Directeur de l'unité systèmes sous-marins, Ifremer
M. BRIGNONE Lorenzo – Responsable du service PRAO, Ifremer

Acknowledgements

First of all, I would like to thank each member of my jury for giving me the honor of assessing my work. Thanks to Yvan Petillot, professor at the Heriot-Watt University, for chairing my jury. Thanks to Luc Jaulin, professor at ENSTA Bretagne, and to François Chaumette, Inria senior research scientist, for the time they spent analyzing this thesis and for all the remarks and high-quality feedbacks given in their reports. Thanks to Viviane Cadenat, associate professor at Université de Toulouse III, to Vincent Creuze, associate professor at Université de Montpelier, and to Andrea Cherubini, associate professor at Université de Montpelier, for accepting to be members of the jury and for their interest in my work. Thanks to the invited members of my jury, David Navarro-Alarcon, assistant professor at The Hong Kong Polytechnic University, Yves Chardard, CEO of the Subsea Tech company, Jan Opderbecke, head of unit for underwater systems at Ifremer, and Lorenzo Brignone, head of the underwater robotics laboratory at Ifremer, for their useful comments.

I also thank the French Sud region and the Subsea Tech company for their financial support.

I would like to thank my supervisor, Vincent Hugel, and advisor, Claire Dune. They introduced me to the scientific and academic world. Their advices and criticism allowed me to advance during the thesis. I thank them for their availability and support. Thanks to all the members of the laboratory for all the discussions and interesting feedbacks. Thanks to Sabine Seillier and Cédric Anthierens for their assistance during the experimental phase of this thesis.

I thank my former colleagues at the office: Minh, Jérome, Nicolas, Ornella, Jean-Baptiste, Hoang Anh... The pleasant working environment and good mood helped me a lot to keep my optimism even in harder times. Thanks to Minh for his support during my first experiments in the summer of 2016. Thank you Anh for your kindness and providential help during the experiments with underwater robots. Thanks for the cayenne peppers. It will be amazing to see them grow and have fresh peppers this summer. Thank you Jean-Baptiste for presenting me this wonderful region that is the Provence and the Mediterranean coast and for the interesting discussions about the latin languages. Thanks for the Slivovica. It will be nice to appreciate it next winter.

I would also like to thank my family. My father, Agnaldo, my sister, Paloma, and my bother, Saulo, for all the good times we spent together and their support, even at a distance. A special thanks to my mother, Magali, for being present at some key moments in this thesis.

Thanks to Salima and Edmond for the restoring weekends and for being present in my thesis defense. Thanks to Marisa for all the support during the last months.

I have to enormously thank Dorothea, my partner and great love. Thank you for always listening to me and providing endless support and encouragement. Thanks also for accepting to be the official graphic designer of this thesis. Your work significantly improved the aesthetic quality of this manuscript. For all your help in everyday life, I do not know how to thank you. I believe you will have to do a thesis as well. But I ask the good Lord not to.

Contents

Glossary

General rules

a	a scalar in the image plane is represented by a lowercase letter
A	a scalar in $\mathbb{E}^3$ is represented by an uppercase letter
$\mathbf{a}$	a vector is represented by a bold lowercase letter
$\mathbf{A}$	a matrix is represented by a bold uppercase letter

Mathematics

$\mathbb{E}^n$	Euclidean space with n dimensions
$\mathcal{F}_a$	a generic Euclidean reference frame $\mathcal{F}_a(\mathbf{P}_a, \mathbf{i}_a, \mathbf{j}_a, \mathbf{k}_a)$ consists of an origin $\mathbf{P}_a$ and an orthogonal basis $(\mathbf{i}_a, \mathbf{j}_a, \mathbf{k}_a)$
$\mathbf{p}$	the bold lowercase letter $\mathbf{p}$ is reserved to represent a vector containing the normalized coordinates of an image point. For example, $\mathbf{p}_a = [x_a \ y_a \ 1]^T$ represents an image point a in the camera frame
$\mathbf{P}_a$	a generic point in the Euclidean space $\mathbb{E}^3$ is noted $\mathbf{P}_a$, where the subscript a identifies the point
$\mathbf{P}_b\mathbf{P}_a$	a vector defined by starting point $\mathbf{P}_b$ and end point $\mathbf{P}_a$
${}^b\overline{\mathbf{P}}_a$	the overlined bold uppercase letter $\overline{\mathbf{P}}$ is reserved to represent a vector containing the Cartesian coordinates of a 3-D point. ${}^b\{\mathbf{P}_b\mathbf{P}_a\} = {}^b\overline{\mathbf{P}}_a = [{}^bX_a \ {}^bY_a \ {}^bZ_a]^T$ represents the 3-D point $\mathbf{P}_a$ in frame $\mathcal{F}_b$
${}^a\left\{\frac{d\mathbf{P}_2\mathbf{P}_1}{dt}\right\}_{\mathcal{F}_b}$	the derivative of vector $\mathbf{P}_2\mathbf{P}_1$ with respect to frame $\mathcal{F}_b$ and expressed in frame $\mathcal{F}_a$
${}^b\dot{\overline{\mathbf{P}}}_a$	the derivative of vector $\mathbf{P}_b\mathbf{P}_a$ with respect to frame $\mathcal{F}_b$ and expressed in frame $\mathcal{F}_b$
${}^b\mathbf{P}_a$	the bold uppercase letter $\mathbf{P}$ is reserved to represent a vector containing the homogeneous coordinates of a 3-D point. ${}^b\mathbf{P}_a = [{}^b\overline{\mathbf{P}}_a \ 1]^T$ represents a 3-D point $\mathbf{P}_a$ in frame $\mathcal{F}_b$
${}^b\mathbf{R}_a$	a 3-D rotation matrix from frame $\mathcal{F}_a$ to frame $\mathcal{F}_b$
${}^b\mathbf{M}_a$	an homogeneous transformation matrix from frame $\mathcal{F}_a$ to frame $\mathcal{F}_b$
${}^b\mathbf{v}_a$	the velocity screw vector ${}^b\mathbf{v}_a = [{}^b\boldsymbol{\nu}_a, {}^b\boldsymbol{\omega}_a]^T$ of point $\mathbf{P}_a$ expressed in the coordinate frame $\mathcal{F}_b$. ${}^b\boldsymbol{\nu}_a$ and ${}^b\boldsymbol{\omega}_a$ are, respectively, the linear and angular components of the velocity vector
$\mathbf{v}_a$	the generic velocity screw vector $\mathbf{v}_a = [\boldsymbol{\nu}_a, \boldsymbol{\omega}_a]^T$ of point $\mathbf{P}_a$ not projected in any coordinate frame. The linear velocity of $\mathbf{P}_a$ is noted

$\boldsymbol{\nu}_a$ and the angular velocity of frame $\mathcal{F}_a$ with respect to the world frame $\mathcal{F}_W$ is noted $\boldsymbol{\omega}_a$.

Introduction

Unmanned mobile robots are well suited to explore environments considered too costly, time consuming, and hazardous for human inspection. They execute a wide variety of tasks such as gathering samples or monitoring and maintenance of structures in inaccessible sites. In some cases, when the explored environment is well structured, these robots can operate in autonomy, without any human intervention, being mostly employed in survey applications. However, autonomous navigation in unknown spaces and autonomous manipulation requiring physical contacts with unstructured environments without human supervision represent technical and technological challenges that remain to be addressed.

Many robotic tasks require continuous human intervention to be carried out. In such cases, the decision-making and intelligence of operation is the human responsibility, while the robot is limited to execute the low level tasks. Teleoperation systems are required to allow human operators to control the robots with the aid of visual and other sensory feedback. Teleoperation systems can be divided into two main categories: wired and wireless communication. Wireless teleoperation is preferable in some robotic applications, since it offers the robot more freedom of motion and flexibility in navigation. Nevertheless, a wireless link may be subject to interference and signal loss that degrade the reliability of data exchange with the robot. Therefore, the use of tethers has advantages in applications where robust data communication is a priority and its interruption can lead to the loss of the robot and mission failure.

Applications where the use of tethers is preferable instead of wireless communication include, for example, urban search and rescue operations (Fukushima et al., 2000; Perrin et al., 2004), planetary geologic survey (Tsai et al., 2013), underwater missions (McKerrow and Ratner, 2007) and sewerage (Reverte et al., 2011). Specifically in the case of underwater operations, wireless communication is even more complicated, since electromagnetic waves are strongly attenuated in water and acoustic subsea communications have a very low bandwidth and significant time delay, which represents a considerable obstacle to robust teleoperation (Marani et al., 2009).

The main advantage of using tethers is the provision of a fast and stable communication link. In addition, tethers can transfer power, which makes it possible to carry out longer operations with energy consuming payload and enable vehicle downsizing thanks to the absence of batteries. Another advantage of tethers is that they can be used as a mechanical support for robots exploring hard-to-reach areas as cliffs, caves, crevices and other steep terrains (McGarey et al., 2016b). Nevertheless, tethering also

have important shortcomings. Tethers are known to limit the robot workspace and they may become entangled with obstacles or with other fellow robots, leading to immobilization. The energy transfer by wire may also represent a constraint to large robot displacements since a very long cable increases the losses by Joule heating. A compromise must be found between the power demanded by the robot and the size of the cable: for a better power flow, thicker cables are needed. However, the thicker the cable, the more rigid it is and greater torsion efforts it applies on the robot, which will require more power to compensate them.

In this thesis, we are interested in the use of small tethered underwater robots, also called mini remotely operated vehicles (mini-ROVs), in the context of exploration of shallow waters, with less than 10 meters of depth. The ROVs are linked to a surface vessel by a tether (also called umbilical) that ensures energy supply and data transfer. Typical ROVs generally weigh some hundred kilograms and are powerful electromechanical machines that require significant human and material resources to be deployed. They have a working depth between 1,000 and 6,000 meters and can move away several tenths of kilometers from their base. The logistical difficulties of deploying large underwater robots led to the development of mini-ROVs, that weigh a few kilograms and are low-power demanding. They are able to explore shallow waters, caves and some wrecks that are not accessible to typical ROVs. However, these light-weight and less powerful vehicles are much more sensitive to the disturbances engendered by a long tether, which can cause unbalance and make them difficult to maneuver. Mini-ROVs are thus limited to displacements around the surface vessel, which consequently limits the exploration of shallow waters, since the vessel cannot get too close to the coast.

The Cosmer Laboratory and the SUBSEA-TECH company, a partner and co-funder of this thesis, came together in reflections and discussions on the possible strategies of underwater robotic teleoperation in coastal areas. Thereby, the idea of using mini-ROVs serially linked with a tether arose: a group of small-size identical robots are distributed all along the tether (see Figure 1). All robots forming the fleet have the same architecture, the same set of sensors and actuators, hence the same motion capability. Together, the small robots can compensate the disturbances of a long tether and allow the leader robot to behave as if it were in the vicinity of the base. Moreover, this chain of robots can work as a sensor network that have a distributed perception of the underwater environment. Mapping of currents and seabed can be achieved with this fleet of robots that can collect data and share informations in real time thanks to the wired connection. Depending on the situation, the fleet could also be organized in ways other than serial, as triangular or square grids, in order to enlarge the covered area.

The objective of this PhD thesis is to investigate how to control the shape of a section of tether linking a pair of robots, a leader and a follower, as it is illustrated by Figure 1. The general idea is that the leader robot freely explores its surrounding, as if it was not attached to anything, while the follower robot is expected to manage the tether shape from its embedded sensors feedback so that the tether does not hamper the leader movements. We propose to use only standard sensors commonly embedded in the mini-ROVs while the tether itself will not be modified nor instrumented.

The thesis is structured as follows. Chapter 1 is dedicated to the state-of-the-

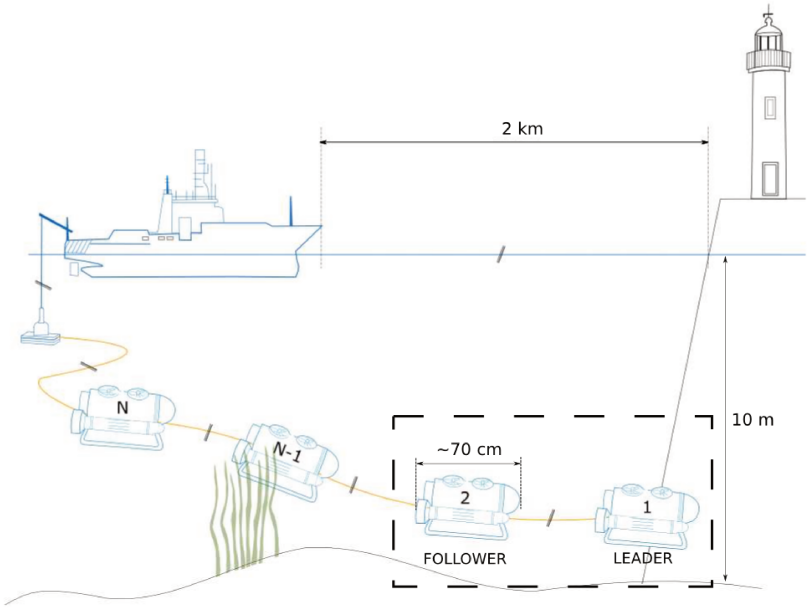


Figure 1: A chain of N small tethered robots used to explore shallow waters.

art in matter of tethered robots, detailing the existing strategies used to control the tether shape during robotic missions. At the end, a new concept of tether handling is presented, the chain of robots, which is the scientific focus of the thesis. Chapter 2 provides the model of the tether and the robots linked by the tether. Chapter 3 presents the procedure used to detect and estimate the tether 3D shape, while Chapter 4 presents the control scheme proposed to regulate the tether shape from the robot's sensory feedback. Simulation results are also presented and discussed. Finally, we conclude with Chapter 5, which provides closing remarks, possible extensions to this work and perspectives on underwater tether shape control. A French summary of this document is available in appendix H.

Chapter 1

State of the Art on Tether Management

There are two types of unmanned underwater vehicles: remotely operated vehicles (ROVs) and autonomous underwater vehicles (AUVs). ROVs communicate continuously with a surface vessel thanks to a physical link called umbilical or tether, due to the low-bandwidth of underwater wireless communication. AUVs are completely autonomous and previously programmed to execute specific mission with very little communication with the surface vessel.

Despite the recent development of techniques of autonomous navigation and task execution, a large number of underwater robotic missions are still carried out by teleoperation. The tether provides a reliable communication link and power supply that enables the fulfillment of long-term missions. There are different types of tethers, depending on the type of mission and deployed robot. Heavy, thick and rigid tethers are used to transfer data and huge amounts of energy to work-class ROVs that operate in deep sea. Thinner and more flexible tethers are used to provide data exchange to large ROVs that have embedded batteries, or also to provide energy to low power-consuming ROVs. Tethers are thus an important element in underwater robotics operations, and a large number of missions that require continuous human intervention could not be achieved without them. Tethers, however, also bring with them additional problems to robotics operation, since they limit the robot workspace and can be entangled with objects in the environment, leading to robot immobilization and, consequently, mission abort.

Tether management is one of the most challenging issues in operations using tethered robots. In this Chapter we address the current strategies that deal with the control of umbilicals. We start by describing the underwater coastal environment where our system will be deployed. Then, we continue by defining the different types of robots and tethers that are used in underwater operation. Next, we give some details about the existing strategies of tether and deformable objects handling in the context of underwater application, which can be also extended to others domains of application. Finally, we present our strategy of tether management for mini-ROVs that operate in

underwater cluttered coastal zones (less than 10 meters deep) while being as far as possible from the surface vessel.

1.1 The Underwater Environment

Coastal areas are the boundaries between the marine and continental domains. These are important areas in terms of both marine and terrestrial biodiversity as well as in terms of convergence of oceanic and continental processes that play a major role in the determination of geomorphological, geochemical and climatic changes. The coasts are very attractive, as evidenced by the strong anthropomorphization of the littoral zone. It is estimated that currently more than 60% of the world's population (3.8 billion people) live in coastal area (less than 150km from the coast). This high concentration of population is due to a considerable economic activity, particularly in the fisheries sector, aquaculture, trade and energy, but also due to the booming touristic sector.

With respect to the underwater environment, the abundance of light in the first meters of depth makes the environment the home of a large range of species. It is also an important area of reproduction, as it is, for example, the case of estuaries and coral reefs. The coastal marine environment shelters fragile ecosystems that should be constantly monitored with the aim of preserving their balance. This surveillance is all the more important since coastal ecosystems are often affected by pollutants generated by ship traffic and liquid waste emission of cities.

Seawater can be characterized by thermophysical properties as density, temperature and salinity. The density of surface seawater ranges from about 1020 to 1029kg/m^3, depending on the temperature and salinity. At a temperature of 25°C, salinity of 35g/kg and pressure of 1atm, the density of seawater is 1023.6kg/m^3 (Nayar et al., 2016). Deep in the ocean, under high pressure, seawater can reach a density of 1050kg/m^3 or higher. In the Mediterranean sea, the surface temperature ranges from 13°C to 25°C annually. Salinity varies between 38.4 and 41.2g/kg and its density is around 1027kg/m^3. Water density slightly varies with depth and can be considered constant in the first 30 meters.

The sea profile can be schematically divided into different areas based on the wave propagation, and the hydromorphodynamic processes involved (see Figure 1.1). The wave base is the depth from which waves can interact with the seabed moving sand shoreward. It is normally characterized as a half of the wavelength. At depths greater than half the wavelength, water motion is less than 4% of its value at water surface and can be neglected. The wave base depth is generally between 1 and 30 meters. The nearshore zone extends from the wave base depth until the waves break point. The surf zone (or breaker zone) extends from the waves break point until the shore line. This is a dynamic area associated with wave-generated currents, including longshore, undertow and seaward moving rip currents. Finally the swash zone represents the portion of beach alternately covered and discovered by the water.

We are interested in exploring the nearshore and breaker zones, which are both areas where intense sea currents take place. As we will see in the next Sections, most

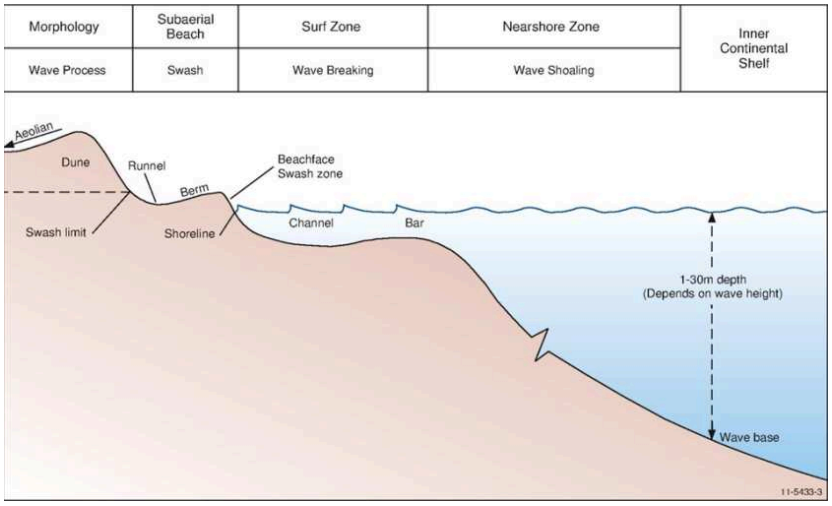


Figure 1.1: Profile or cross-section of a typical wave-dominated beach showing the nearshore zone, the surf zone, and finally the swash zone. Courtesy of Short and Woodroffe (2009).

of the time, unmanned underwater vehicles are equipped with proprioceptive and exteroceptive sensors, such as inertial measurement units (IMU), sonars, cameras, doppler velocity-log (DVL), pressure gauge, etc. In coastal environments, inertial measurements become noisier due to accelerations produced by the currents on the robot. This leads to a gradual drift of position estimation in inertial navigation systems. The image feedback of the cameras are also affected by the presence of current, since they put the seabed sediments in suspension, rendering the water more turbid. Camera visibility is already limited to some tenths of meters in clear waters due to light absorption (all colors are absorbed, except blue). In turbid waters, the presence of sediments can further degrade the image quality of cameras. The following Sections deal with a survey of existing solutions in order to design a relevant robotic system that could be used to explore such challenging environments.

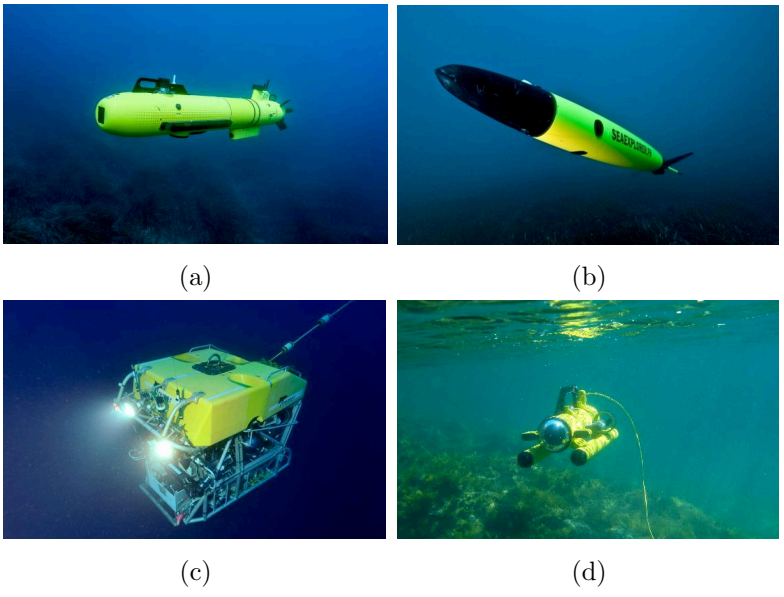
1.2 Underwater Robots

There are different categories of underwater robots that are each one suited for the execution of specific tasks at a given depth range (see some examples in Figure 1.2).

1.2.1 Autonomous Underwater Vehicles

In survey missions, where large areas shall be covered and inspected by robots (a few square kilometers in a day, for example), the AUVs (Autonomous Underwater Vehicles) are commonly employed. They are untethered robots designed to cover large travel distances thanks to their hydrodynamic shape (usually in torpedo form) and to collect data in autonomy for post processing. Examples of application are localization

missions of wreckages of missing airplanes or detailed mosaicking of the seafloor for the oil and gas industries (Wynn et al., 2014). An important subclass of AUVs are the underwater gliders, that use small changes in their buoyancy in order to move up and down in the ocean. This vertical displacement generates a forward motion thanks to the use of flying wings and to the control of the vehicle pitch by movable internal ballast (usually battery packs). Vehicle steering is accomplished either with a rudder or by moving internal ballast to control roll. They are generally used in oceanography research to collect chemical and physical measurements of seawater in ocean sampling missions that range from hours to weeks or months, with up to hundreds of kilometers of range (Inzartsev and Alexander Pavin, 2009, chap. 26). Since they are deployed for missions of long duration (up to several months) and large displacement (hundreds of kilometers), their localization is less accurate than other AUVs during the cyclic diving period.



(a) (b)

(c) (d)

Figure 1.2: Some examples of French underwater robots. (a) the AUV A9 from ECA robotics. (b) the AUV glider Sea-Explorer from Alseamar. (c) Victor 6000, an Ifremer work-class ROV dedicated to scientific ocean research in deep waters. (d) the observation-class mini-ROV observer from Subseatech.

1.2.2 Remotely Operated Vehicles

ROVs are employed in missions where continuous human intervention is required. They are highly maneuverable vehicles that are operated by a crew aboard a vessel or floating platform. The connection between the robot and the crew is made possible thanks to a tether or an umbilical cable that is used along with a tether management system (TMS). ROVs are often designed to execute inspection, repair and maintenance of subsea structures in deep water. Some fields of applications are the offshore hydrocarbon

extraction and the scientific research studying deep sea ecosystems and archaeological sites. There are three main categories of ROVs, based on the vehicle size and capabilities (Christ and Wernli Sr, 2013):

1. **Work class ROVs**: Vehicles in this category are generally heavy electromechanical machines (over 1000 kg). These robots are used to execute maintenance and construction tasks of huge underwater structures, typically in the oil and gas industry or in the heavy civil engineering. Due to the necessity of large amounts of force, the vehicle propellers and manipulators are hydraulically powered. They are designed to be deployed in waters deeper than 3000 meters. This deep depth can be achieved thanks to bulky pressure housings that protect the whole system from high pressure. The size and weight of the vehicle implies that they have to be deployed from large surface vessels with a specific launch and recovery system (LARS) that looks like a crane. In such operations, the amount of exposed cable is so important that a dedicated tether management system (TMS) is required to reduce the disturbances generated by the cable on the robot.

2. **Mid-sized ROVs**: These vehicles weigh from 100 kg up to 1000 kg. They are designed to operate in intermediate depths around 1000 meters, being thus more compact than the work-class ROVs. They are also generally all-electric vehicles (powering locomotion and sensors) with some hydraulic power for the operation of manipulators. Their missions are similar to work-class ROVs, but operating at lower depths. They also need to be deployed by a LARS and depending on their weight and size, smaller surface vessel and fewer crew members are needed, reducing the operation costs.

3. **Observation class ROVs**: These robots go from the smallest vehicles to those wighting 100 kg. They are employed in depths lower than 300 meters. They are completely electric machines, DC-powered and much less expensive than the other class of ROVs. They are mainly used for observation missions, for inspection in offshore industry or for oceanographic and archaeological research. Some of them are equipped with electrical manipulator arms to perform dexterous tasks (as biological sampling or archaeological excavation). The arm is equipped with specific tools such as vacuum gripper, brushes or pliers. The lighter vehicles within this class are typically hand launched, and can move freely from the surface with manual handling of the tether.

In this thesis we focus on the use of **mini-ROVs**, a sub-category of the observation class. Usually, mini-ROVs are used in places such as a sewer, pipeline, small cavities, very shallow waters and other cluttered spaces. One person can carry the complete ROV system (robot and tether) up to a small boat, deploy it and complete the job without outside help. A remarkable advantage of using mini-ROVs is the low cost of the operation (Wernli and Christ, 2009). Actually, the reduction of the operating costs is a main goal in underwater robotics research, since the mobilization of huge structures (host ship and crew) represents an important part of the mission charges. In the cases

of survey or light manipulation missions, mini-ROVs can be used in cooperation with USVs (unmanned surface vessels), as described in Shimono et al. (2015); Conte et al. (2018). This kind of system eliminates the need of an on-board crew, that can in return operate onshore via a radio-link. The reduced costs facilitate the access to underwater intervention for safety and security operations in civil structures such as ports, channels and dams (Molchan, 2005), as well as for many other scientific domains, such as oceanography (Smolowitz et al., 2015) and archaeological (L'Hour and Creuze, 2016) research. Normally, in such fields, the tasks of surveying and sampling are made by a professional diver. However, the operating complexity, medical problems, high costs and limited time of diving have led to the development of alternative robotic solutions in which the use of mini-ROVs plays a relevant role. These robots are operated up to some hundreds of meters away from the surface vessel.

At these distances, important dragging forces are applied to the tether which considerably disturbs the motion of the mini-ROV. Therefore, even for these small robots, a system that could manage the tether is required. Besides, this system should be able to detect or prevent tether entanglement. This necessity is even more important if the mini-ROV is deployed in confined or cluttered environments. In order to achieve a functional tether management for mini-ROV we must have some knwoledge about the tether properties. Another objective is to minimize the forces that the tether applies on the ROV. In the next section we focus on the different types of tether and their physical properties.

1.3 Tethers

In order to understand the forces that a tether applies on a mini-ROV, the interaction between the robot motion and the cable shape should be modeled. First we will give some definitions about ROV usual tethers. Secondly we will investigate the influence of the tether cross-sectional area on the system design and then present some existing cable models.

1.3.1 Utility

ROVs require an electromechanical cable that provides mechanical support, power supply and data exchange with the surface vessel. In work-class and middle-class ROVs, a tether management system (TMS) is needed to regulate the thousands of meters of cable deployed and disturbances caused by the surface vessel motion. The cable linking the surface to TMS is termed the **umbilical**, while the cable from the TMS to the submersible is termed the **tether**. The umbilical cable is used to reach the operational depth while the tether cable allows the ROV to make excursions at that depth for a distance of around a few hundred meters away from the TMS. ROVs of various sizes and categories can use a TMS, with the constraint of having a crane on board the ship that can move the whole system (robot and TMS) from the deck to the splash zone.

In the case of mini-ROVs, since their operational depth is limited to some hundreds of meters, no TMS is needed and the robot remains in the vicinity of the surface vessel.

Thus, we will always use the term tether when making reference to a cable linking a mini-ROV to a surface vessel, base station or fellow robot. The diameter of the cable is the dominant factor in overall vehicle drag. Therefore, minimizing cable diameter is an important part of ROV design and operation. Some advanced ROVs carry their own power sources and only require a communication link to the surface vessel through an expendable fiber-optic cable (Brignone et al., 2015). The umbilical cable is generally steel jacketed, while the tether cable uses synthetic fibers to maintain the required buoyancy (neutral, slightly negative or positive).

1.3.2 Buoyancy

The tether buoyancy in water is also an important aspect to consider. There are three buoyancies to consider:

- Positive: when the tether has a lower density than water and therefore rises.
- Neutral: when the tether has the same density than water and therefore remains in suspension in the water column.
- Negative: when the tether has a higher density than water and sinks.

Some examples of tethers are depicted in Figure 1.3 and their physical properties can be found in Table 1.1.

Which tether to choose depends on the application. For example if the ROV is to operate on the sea floor it would be preferable to use a neutral or positive tether to keep it from dragging along the sea floor and potentially becoming entangled with objects. A negatively buoyant tether has fewer applications and are commonly employed in deep water environments, or areas with snagging hazards near the surface of the water, such as under-ice teleoperations (Bowen et al., 2012; Katlein et al., 2017).

Tether	diam. (m)	lin. density (kg/m)	buoyancy
Fathom	7.6×10^{-3}	4.3×10^{-2}	neutral
Novasub	7.4×10^{-3}	5.4×10^{-2}	negative
μlinx	8.0×10^{-4}	7.0×10^{-4}	negative

Table 1.1: Tethers physical properties shown in Figure 1.3: diameter, linear density and buoyancy in fresh water.

1.3.3 Cross-section

Work class ROVs have power provided in alternating current (AC), which is transmitted through the umbilical from the surface vessel to the cage and then from the cage to the ROV through the tether. When it reaches the robot, the electrical power is converted to direct current (DC) to power the electronics. Video and data are often transmitted through optical fibers in order to reduce the electromagnetic disturbances caused by the AC power transmission.

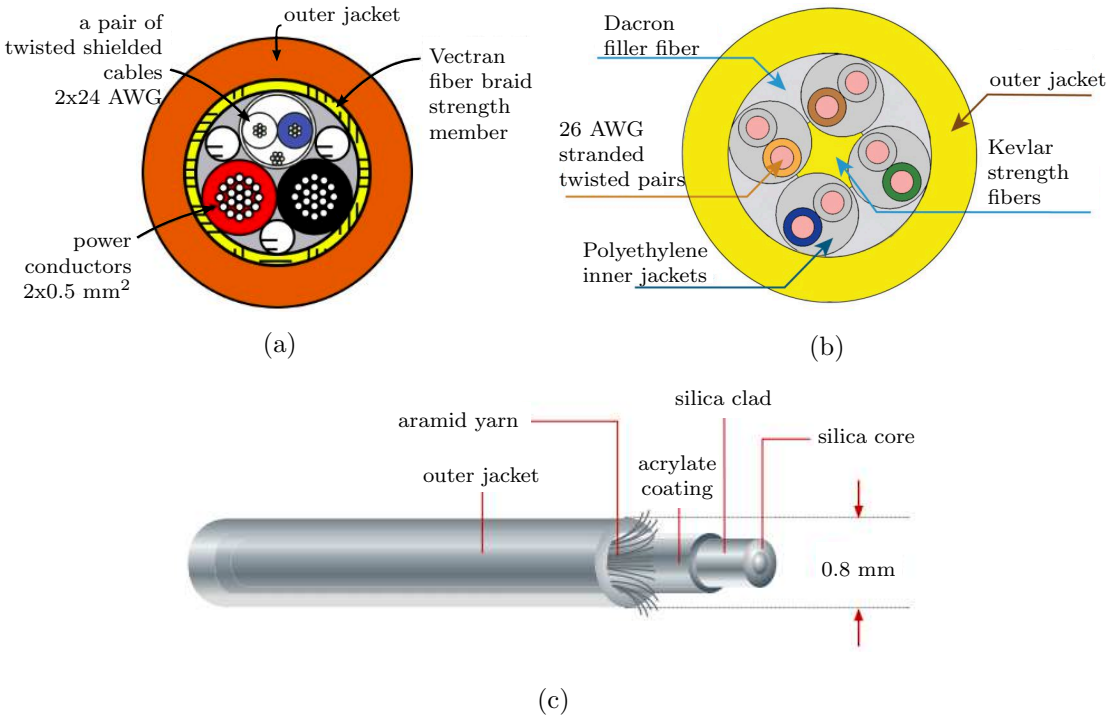


Figure 1.3: Some examples of tethers used by mini-ROVs. (a) the tether model DLR-1P20-2C50 from Novasub for power supply and data transfer. (b) the Fathom tether from BlueRobotics that carries four pairs of 26 AWG wire for data exchange. (c) the μlinx 50 OM3 tether from OFS optics is composed of a multimode 50 μm optical fiber for data transfer.

While mid-sized and work class ROVs use AC current for power transmission through the umbilical and then a combination (AC and DC) along the tether, most observation class ROV systems use exclusively DC current for power transmission. The delivered power should be sufficient to operate all the electronics and thrusters. The propulsion of the vehicle must be powerful enough to overcome its own drag force but also that of the tether. Therefore, the tether length may be critical in determining the available power for vehicle displacement. A too long tether can generate too much drag force that would lead to the vehicle immobilization. In addition, long cables also decreases the available power due to losses caused by Joule effect. The maximum tether length for a given power requirement is a function of the voltage, the size and the resistance of the conductor:

$$P = \frac{U^2 A}{\rho L} \tag{1.1}$$

where P is the delivered power, U is the voltage, A is the cross-sectional area of the conductor, ρ is its electrical resistivity and L is the tether length. The maximum range of the robot is hence limited by a maximum tether length above which the most part of the delivered power is dissipated in heat through the cables (see Figure 1.4). In the

case of mini-ROVs, this limitation is even more restrictive due to the low voltage of the embedded electronic devices and thrusters (about 12 volts).

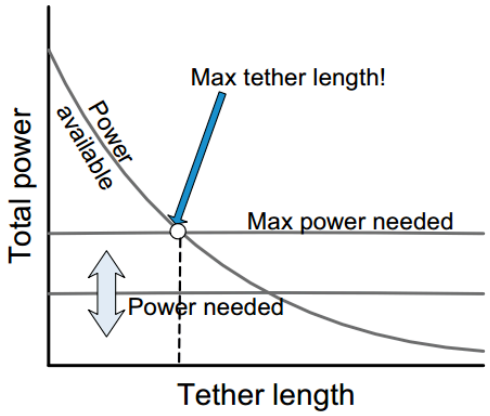


Figure 1.4: The maximum tether length for a given voltage V and a given cable cross-sectional area A. Extracted from Christ and Wernli Sr (2013).

The ROV's tether is typically the highest drag item on the ROV system. The drag produced by the ROV is based upon the following formula (Christ and Wernli Sr, 2013):

$$D_{tether} = \frac{\sigma A_t L_t V_t^2 C_{d_t}}{2} \tag{1.2}$$

where σ is the ratio between the density of water and the gravitational acceleration; C_d is the non-dimensional drag coefficient, ranging from 0.8 to 1 based on the cross-sectional area of the tether profile (A); L is the tether length; V is the tether relative velocity with respect to water and AL is the characteristic area on which C_d is applied. The robot drag is calculated in the same way and the system total drag is then:

$$D_{total} = \frac{\sigma A_t L_t V_t^2 C_{d_t}}{2} + \frac{\sigma A_v L_v V_v^2 C_{d_v}}{2} \tag{1.3}$$

The total drag increases proportionally with the tether length for a given cross-section (see Figure 1.5).

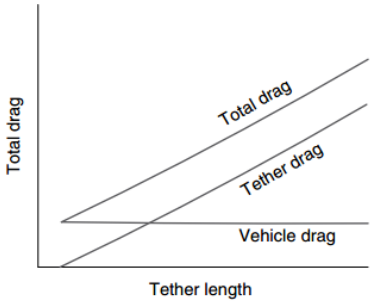


Figure 1.5: Tether drag versus tether length. Extracted from Christ and Wernli Sr (2013).

The demanded power needed to compensate the total drag is calculated as follows:

$$P_D = D_v V_v + D_t V_t. \tag{1.4}$$

It is linearly proportional to the tether length and to the cube of the velocity, which means that for a given maximum velocity, the longest the cable, the highest the power needed to displace the vehicle and tether. From equation 1.1, in order to increase the available power, for a given length, we have to increase the cable cross-sectional area, which directly impacts the power needed to compensate the dragging force. Then, a trade-off has to be found between the maximum velocity and maximum reach of the system with reasonable power requirements.

1.3.4 Models

In order to be able to manage the motion of a cable that is freely deployed in seawater in presence of currents, tides and waves, we need to estimate the tether shape. To do so, there are two solutions. The first one is to add sensors at the attachment points and all along the cable, as is the case of Smart Tethers (Frank et al., 2013). These tethers are designed to offer a real-time GPS location of the ROV using non-acoustic tether shape measurements. Several nodes are embedded along the tether cable to provide local orientation and depth. This information is used to estimate the current tether shape and ROV heading and position with a 10-30 Hz refresh rate. Similarly, optical fiber may also be used to give visual feedback of the tether shape (Childers et al., 2010).

The second solution, if we consider a non-equipped tether, is to use a parametrized model of the cable. A cable can be physically described by its attachment points, a cross-sectional area, length, linear density and axial, flexural and torsional rigidities. Cable modeling is useful to help to understand cabled structure statics and dynamics as well as to design carrying cables for suspension bridges, intercontinental communication cables, cable-driven robots, and to design umbilicals and tethers providing power supply and data transfer for teleoperated robots.

The most complete cable model in the air relies on the Irvine equation (Irvine, 1981) that takes into account both the elasticity and the deformation of the cable due

to its own mass and has been shown to be very realistic (Merlet, 2018c). This model assumes that the cable lies in a vertical plane, the cable plane, and is therefore a 2D model. However, this model is complex to use in a kinematic analysis (Merlet, 2018c). In Howell (1992), the problem of low-tension cables is investigated for underwater applications and several models that consider or ignore the cables elasticity are studied. The equations of motion for a cable are nonlinear and strongly coupled for complete cable models. Other numerical models of underwater umbilicals and tether were addressed in Buckham (2003); Buckham et al. (2003), where a tether lumped mass model is addressed. If we consider inextensible (nonelastic), perfectly flexible and subjected to a uniformly distributed load, a widely spread cable model is the catenary curve (Irvine, 1981). Analytic solutions are difficult to obtain for complete models, and numerical approximation techniques are often necessary. In the simplified case of a catenary, analytic solutions are available.

In the next section we will see that other simplified models were used for tether and deformable objects shape control strategies such as parabolas, splines and circle arcs. For real-time robotic control these approximated parametric models have the great advantage of speeding up the computational time of the control loop, although being limited to some environmental and operational assumptions.

1.4 Tethers and Deformable Objects Management

This section focuses on existing strategies of tether management for underwater and mobile robots in general. Since the tether can also be seen as a deformable object, this section will also address some techniques of deformable objects handling that could be applied to tether management.

1.4.1 Underwater Applications

This subsection deals with tether management systems (TMS) that are typically used for work-class and middle-class robots. As depicted in Figure 1.6, the TMS is placed at the junction between the end of the umbilical and the beginning of the tether (Hawkes and Jeffrey, 1987; Abel, 1994). This junction is made by a simple clump/depressor weight or by a vehicle handling system (a protection cage or a top hat mechanism), as it is shown in Figure 1.7.

The use of a clump weight is prevalent in the observation-class category. The clump weight absorbs the cross-section drag of the current in a passive way, relieving the submersible of the umbilical drag from the surface to the working depth. Therefore, the ROV only needs to drag a portion of the tether length between the clump weight and the vehicle. Similarly to clump weights, cages also function as a negatively buoyant anchor to overcome the drag imposed by the umbilical, and additionally, they are used to protect the vehicle against abrasions and deployment damage due to the instability of most surface vessels. Clump weights and cages can be used without a TMS for reasons of simplicity. The addition of a TMS is considered by operators as a similar complexity of having a second ROV concurrently in the water.

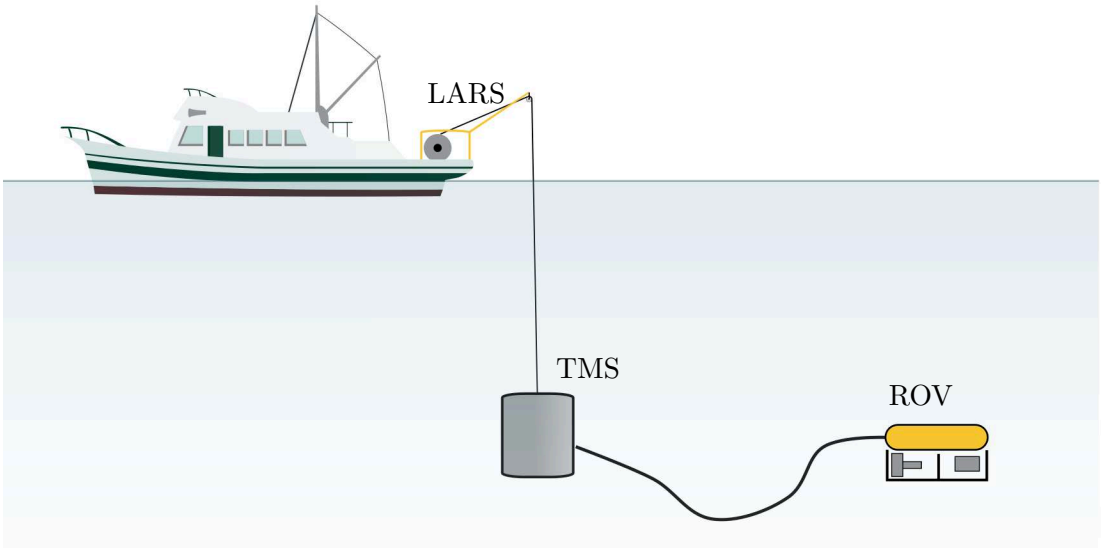
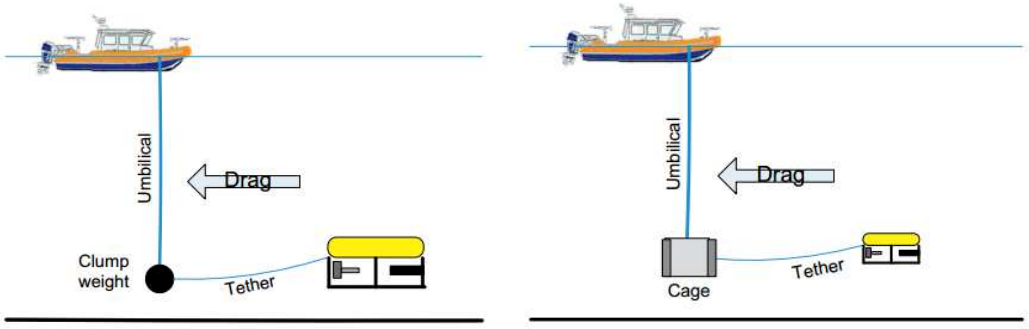


Figure 1.6: Main components for teleoperation in underwater robotics. A remotely operated vehicle (ROV), a tether management system (TMS) and a launch and recovery system (LARS). Adapted from (Salgado-Jimenez et al., 2010).

The TMS aims to control the amount of unwound tether between the ROV and a cage housing or clump weight. With respect to the umbilical, the amount of unwound cable can be controlled by a heave compensator system (see Figure 1.8). The objective is to reduce the umbilical slackness by reeling in the cable or to unwind additional cable as the sea dynamics dictates. If these heave dynamics are not compensated for then the vehicle at the end of the umbilical will not be stable and/or the umbilical can become overstressed and damaged. Heave compensation systems can be active (Yang et al., 2008) or passive (Driscoll et al., 2000). With active heave compensation (AHC), motion sensors measure vessel movement along the vessel's vertical axis and direct the AHC unit to reel in or reel out the cable. With passive heave compensation (PHC), a system of sheaves is used for shortening or lengthening the exposed cable length in concert with the vessel's motion. AHC is much more accurate for achieving a constant load position as it works on a position reference frame (rather than weight which has no direct reference to motion).

In most offshore applications, the ROV umbilical will be handled by a winch system that can have a heave compensator embedded to wind/unwind the umbilical following the ocean dynamics. The tether connecting the ROV to the cage is handled by the TMS, which increases or decreases the tether length according to the operator's commands.

Another strategy for tether management more rarely addressed in the literature is the use of the surface vessel for the dynamic positioning of a control point on the tether in order to reduce the disturbance force exerted on the ROV. In Triantafyllou and Grosenbaugh (1991), a control scheme is proposed to manage an underwater tether from



(a) (b)

Figure 1.7: Two examples of a ROV operation with a negatively buoyant anchor to overcome the drag imposed by the umbilical: (a) a clump weight/depressor and (b) a cage used to protect the ROV against abrasions and deployment damage while lowering and heaving the ROV. Both options can be used with a tether management system (TMS) in order to reduce the drag force the vehicle has to deal with. Courtesy of Christ and Wernli Sr (2013).

a dynamically positioned surface ship. The dynamic positioning system uses the surface ship's thrusters to provide the control force and hydrophones and submerged pingers to position itself, while the ROV position is calculated through a tether numerical model using the ship position as input. The main challenge of this control scheme is to take into account the important time delay that the system exhibits: the underwater cable motion reacts much slower to the ship displacement than to the reeling activity of the winch. In Prabhakar and Buckham (2005), this approach is evolved and a decoupled controller is proposed for the horizontal positioning and depth regulation of a tether point near the ROV. Since the transverse motions propagate much more slowly in the tether than motions along its vertical profile, the horizontal tether positioning is accomplished via the ship displacement and the depth regulation is made by the winch.

In Khatib et al. (2016), the concept of using an observation-class ROV as a relay station between the surface vessel and an humanoid underwater robot was addressed. The objective was that the middle robot could reduce the disturbances caused by the tether on the humanoid robot during the execution of delicate tasks of archaeological pieces collection from the seabed. In addition, the relay station provided nearby recharge and reduced the risks of tether entanglement, allowing the humanoid robot to achieve wider and safer operations.

1.4.2 Terrestrial and Aerial Applications

Tether and deformable objects handling is a challenge also found in some fields of application other than underwater environment. In the context of planetary exploration, tethers are used to connect a base station to a rover dedicated to explore hard-to-reach

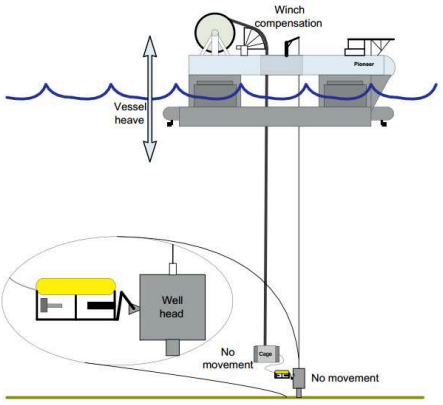


Figure 1.8: Illustration of heave compensation. Courtesy of Christ and Wernli Sr (2013)

terrains (Krishna et al., 1997; McGarey et al., 2016b; Brown et al., 2018). In such applications, the tether works as both a mechanical support and a transmission link of power and data. In Iqbal et al. (2008), the tether also serves as a guide line so that the rover can autonomously track it in order to find the way back to the base station. Frontal infra-red sensors mounted on the front of the rover are used to track the tether while it is rewound into the rover reel. The tether lays on the ground and entanglement with obstacles was an issue unfortunately not addressed in this paper.

When the terrain is to steep, the tether can be kept under tension and function as an anchor for the rover to explore planetary craters. In Tsai et al. (2013), the problem of vision-based tether-assisted docking of a daughter rover to its base station (central module) is addressed. The docking strategy relies on an algorithm running on the rover that uses stereo cameras to detect fiducials markers mounted on the central module and then estimate its relative pose. The rover uses a motion planner to position and orient itself such that it aligns its arm with the docking cone to be retracted back to the mother station at the end of the mission. Proper tether tension is maintained during motion thanks to the vision-based relative pose estimation that enables to geometrically calculate the instantaneous exposed tether length from the tip of the rover's arm to the mother station. The fact that the tether itself is not used to provide information for the docking process limits the robot workspace since the docking station should permanently stay in the rover's field of view.

Objects in the environment can also be seen as anchor points instead of obstacles. In Vishnu et al. (2008) and Rajan et al. (2016), force sensors were used to detect tether anchor points. On-board tilt sensors together with the exposed tether length measurement allowed to estimate the robot position in a line-of-sight. The disentanglement technique proposed was based on an algorithm of tether-following that allows the robot to bypass the obstacles without being clung to them. Experiments with two wheeled robots serially linked together by a tether to a base station were carried out. The robots were organized in a leader/slave configuration and performed tether disentanglement around a single obstacle. Tether aided localization was then further improved by gath-

ering multiple sensory feedback. In Murtra and Tur (2013), wheel odometry, tether length measurements, and an IMU (inertial measurement unit) were used to localize a pipe-inspection robot, where tether length was used to limit uncertainty about the distance traveled in a pipe. This work was only concerned with localization of the robot and no attempt was made to detect and map tether contact points (i.e., the anchor point) within the pipe. In McGarey et al. (2016a) and McGarey et al. (2017), the pose of a tethered robot and the positions of the intermediate tether anchor points were estimated using tether length, bearing-to-anchor angle, and odometry gathered along the trajectory. The objective of this work was the formulation of a *tethered simultaneous localization and mapping* (TSLAM) problem whose solution would allow the robot to safely return to its base along an outgoing trajectory while avoiding tether entanglement. The motivation was to use TSLAM as a building block to aid conventional, camera and laser-based approaches to SLAM, which tend to fail in dark and or dusty environments.

Multiple mobile robots can be used for cooperative transportation of large objects, that can be rigid (Huntsberger et al., 2004) or flexible. In Echegoyen et al. (2010), three terrestrial robots were used to transport a flexible hose modeled by Geometrically Exact Dynamic Splines (GEDS). A camera with a global view of the scene was used to detect the robots and the hose by color segmentation. The leader robot pursued a predefined trajectory while the follower-robots' command velocities were computed from a fuzzy-heuristic local controller. The curvature of the hose segment in front of each robot was used as a visual feature by the controller that was regulated in order to avoid the hose of being taut.

Aerial robots were also used in the transportation of flexible cables (Estevez and Graña, 2015; Estevez et al., 2015). The criteria for the transportation was that all the robots should carry the same load. Thus, the hose was modeled by a catenary and the IMU of the robots were used to estimate their relative height, which was regulated by a PID controller with the aim of evenly distribute the hose weight among the robots.

Tethers are used by unmanned aerial vehicles (UAV) for long-term missions with high-speed communication between the operator and the robot in an wide range of applications, such as robot-assisted search and rescue (Pratt et al., 2008) and coastal and environmental remote sensing (Klemas, 2015). Most published work in the field of tethered flight are restricted to the taut tether case. In these systems no tether management is employed while the UAV maintains tension. Otherwise, a winch mechanism placed in a fixed or mobile base station continuously reels in any slack tether length (Nicotra et al., 2017). The dynamics and control of a quadrotor unmanned aerial vehicle connected to a fixed point on the ground via a tether was addressed in Lee (2015). The tether was considered as a collection of an arbitrary number of rigid links that are serially interconnected via ideal ball joints. The motion equations of the system (robot and tether) were obtained from Hamilton's principle. A control law based on inertial sensing was designed through feedback linearizion and used the UAV pose with the aim of maintaining a desired tether state (orientation and tension). This control law was evaluated in two numerical examples of station keeping and predefined path tracking. A reactive tether management approach, where the tension and

departure angle are measured at the winch, showed moderate winch controller results that can be further improved by incorporating knowledge of the UAV position (Zikou et al., 2015). Another work used the measured tether length, tension, and departure angle as a means for non-GPS position estimation of the UAV based on a catenary cable model (Kiribayashi et al., 2017). In Talke et al. (2018), the quasi-static catenary curve of a hanging tether between an essentially stationary UAV and a small unmanned surface vehicle (USV) is investigated and characterized. The objective is to develop in a near future a winch controller that could maintain the tether slack and compensate the USV heave in order to minimize the tether traction and the risk of being in contact with the water surface. A multi-agent extension of the tethered aerial robot problem was investigated through numerical simulations in Tognon and Franchi (2015), where a chain of two flying robots was considered. The goal was to independently control the elevation angles of the two tether segments as well as their internal stress.

Cable management is also an important issue in cable-driven parallel robots (CDPR) domain. In Dallej et al. (2011) and Dallej et al. (2012), a vision-based controller was introduced and validated on simulation. The cable model relied on a simplified model of an inextensible hefty cable in which the profile was considered to be a parabolic curve. A multi-camera setup allowed to measure the direction of the cable tangents as well as the pose of a visual target attached to the robot's mobile platform, whose pose was then controlled through visual servoing. Recently, a novel method for workspace planning for a cable-control robot in cluttered environments was introduced (Wang and Bhattacharya, 2018). More complex cable models were also considered and their use was proved to enlarge the workspace border of CDPR (Merlet, 2018a).

1.4.3 Synthesis of Existing Cable Management Strategies

In this Section we will present a short summary of the cable perception and handling strategies previously addressed. We will conclude by analyzing which solutions would be adapted to operations with mini-ROVs navigating in the underwater coastal environment.

1.4.3.1 Passive and Active Cable Management Strategies

The ROV systems are composed of a surface vessel, a linking cable and a teleoperated robot. The solutions for tethers and cables management presented in Sections 1.4.1 and 1.4.2 are grouped in the following list:

A. Passive management

A1. add an intermediate clump weight: the heave motion of the surface vessel is absorbed by a submerged massive body placed at the junction between the umbilical cable and the ROV tether (Christ and Wernli Sr, 2013, chap. 9).

A2. add a passive heave compensator: shock absorbers, drill string compensators or more sophisticated hydraulic and mechanical systems of

winches (Huster et al., 2009) are used to absorb and dissipate the energy generated by the ship heave motion.

B. Active management

B1. actuate the tether extremities: winches are used in underwater missions to compensate heave motion of the vessel and manage the amount of tether between the clump weight and the robot. The winch mechanism can be placed on the ship (Yang et al., 2008), on the robot (Brignone et al., 2015) or inside the TMS cage (Hawkes and Jeffrey, 1987; Abel, 1994). They are also used by cable-driven parallel robots (Dallej et al., 2011, 2012), terrestrial rovers (Tsai et al., 2013; Vishnu et al., 2008; Rajan et al., 2016) and unmanned aerial robots (Lee, 2015; Talke et al., 2018) to regulate the length of exposed cable.

B2. actuate the surface vessel: regulate the ship-robot relative position in order to reduce the amount of tether deployed to allow the ROV horizontal displacement (Triantafyllou and Grosenbaugh, 1991; Prabhakar and Buckham, 2005).

B3. actuate the tether: add an intermediate robot that works as a relay station and is commanded by an onboard operator (Khatib et al., 2016) or **use a team of robots** that transport together deformable objects (Echegoyen et al., 2010; Estevez et al., 2015; Rajan et al., 2016).

The most simple strategy to reduce the disturbances generated by the cable on the robot is to use a clump weight that absorbs the cable oscillations caused by the heave motion of the ship (item **A1** of the list). This is a passive tether management solution where neither the ship nor the cable are instrumented. This option is commonly used by work-class and middle-class ROVs that operate in mostly wide and empty environments and is therefore not adapted to the type of mission we focus on that use mini-ROVs to explore coastal underwater environments.

The cable oscillations can be compensated by equipping its extremities with intelligent winch mechanisms that can either be aboard the surface vessel, embedded in the robot or submerged at the depth of operation (item **B1**). When the exposed tether length is controlled aboard the ship it is called heave compensator. When the system is submerged and it is commanded by an onboard operator it is called tether management system (TMS). Heave compensator could be an interesting solution for tether management of mini-ROVs. However, a shortcoming of this solution is that the mini-ROVs would be limited to explore a zone in the vicinity of the ship since they cannot pull a large amount of cable (over 300 meters). On the other hand, the choice of adding submerged winch mechanisms is not adapted to coastal zones due to the shallow depth and to the presence of obstacles in the marine relief such as stones and coral reefs. Finally, equipping the ROV with winches increases its weight, making it bulkier. Moreover, the reeling activity generates non-negligible variation of the mini-ROV mass repartition, affecting its buoyancy and whole dynamics.

Other studies proposed to use a combined control scheme that regulates both the ship positioning and the amount of deployed tether by the surface winch (item **B2** item of the list). However, the mini-ROVs would remain limited to the exploration around the surface vessel, which is also constrained itself not to get too close to the coast.

1.4.3.2 Classification according to Cable Perception and Modeling techniques

In the case of active tether management, an estimation of the tether state must be computed to determine the action the system has to take: reel in/out the tether or move its extremities, for example. The models used to represent the cable shape vary depending on the physical properties of the cable and the hypothesis of operation. The sensors used to perceive the cable and to provide input information for the model computation also vary depending on the type of robot used and the domain of application.

The list here below summarizes cable handling strategies with respect to the absence or presence of cable models:

A. Without tether model: the tether shape can be regulated without considering any cable model. This can be done by a human operator, by a passive mechanical system or by an active compensator based on sensory feedback. Unless otherwise noted, the three subclasses below are related to underwater operations.

A1. teleoperation: human operators can command a tether management systems (TMS) reeling in/out the tether based on visual information or on ROV-ship relative position estimation (Abel, 1994). Another alternative is to add an intermediate robot that works as a relay station near the main ROV. It was the case of the operations of the underwater humanoid robot Ocean One (Khatib et al., 2016).

A2. passive regulation: shock absorbers, string compensators or more sophisticated hydraulic and mechanical winch system can be used to passively absorb the heave motion of the ship and thus reduce the disturbances generated on the umbilical (Huster et al., 2009).

A3. active regulation: the disturbances caused by the ship heave motion on the umbilical can also be compensated by a control unit that manages the speed of a hydraulic winch from information provided by an inertial measurement unit (IMU) (Yang et al., 2008). In the context of planetary exploration, the exposed tether length was also regulated through active winches mounted on rovers (Tsai et al., 2013). The tether connected the rover to a base station and it was maintained taut during the entire operation. Fiducial markers mounted on the front of the base station allowed the rover to estimate its relative pose and then maintain the tether taut during the entire operation.

B. With geometric tether models: cables and deformable objects can be represented by geometric models that may offer regulation parameters to be used

in a control loop.

B1. Parabolic curves were used to model the cable profile for cable-driven parallel robots. In Dallej et al. (2012), a set of cameras was used to estimate the relative position of the robotic platform and the control winches in order to regulate the cable length deployed.

B2. Spline curves were used to model a flexible hose that is transported by a team of terrestrial robots (Echegoyen et al., 2010). A camera observing the global scene was used to return to the vehicles velocity controllers the robots position and the spline curvature (that represented the hose 1-D deformation).

B3. Catenaries were used to model tethers and flexible wires in the context of unmanned aerial vehicles (UAV) operations. In Estevez and Graña (2015), the relative pose of three simulated quadrotors was regulated so that the hose being carried has its weight equally distributed among the robots. In Lee (2015), a control scheme was proposed to manage the shape of a simulated tether linking an UAV to a grounded anchor point. No winch system was used and the tether shape was managed through the UAV-anchor relative position. In Talke et al. (2018), a tether with known length and attachment points relative position was modeled by catenary curve. The tether was linking UAV and USV (unmanned surface vessel) prototypes. The catenary model was useful to investigate the system robustness to the USV heave motion in order to minimize the risks of the tether touching the water surface.

C. With tether lumped mass models: more complex numeric schemes are also used to represent the tether shape in underwater applications (Prabhakar and Buckham, 2005; Triantafyllou and Grosenbaugh, 1991). These approaches are often grounded in the knowledge of the relative position between ship and ROV, which implies the use of external acoustic positioning systems. In Eidsvik and Schjølberg (2016), the cable was modeled by finite-elements methods using Beam equations.

The amount and type of information given by the sensory feedback determines the cable model that will be used. For example, force and angle sensors can be used to regulate the tension and orientation of a cable without necessarily having to resort to an accurate model. Otherwise, vision sensors are preferably used to return information that can be easily applied to geometric models of the object in the observed scene. Positioning systems are used in the case of more precise models where the physical properties of cables are well-known.

The sensory payload available in a mini-ROV is very restrictive because of its small size and its low propulsive power. These robots are mainly used for observation tasks and are, therefore, often equipped with cameras that give visual feedback to the operator, and with an IMU that gives feedback about the robot orientation and that can be used to get a rough location estimation. Since visual sensors are available, representing

the cable by parametrized geometric curves sounds a natural choice. That is actually part of the solution we investigate in this thesis with the aim of proving the feasibility of a tether management strategy that allows mini-ROVs to be deployed far away from the surface vessel without being limited by an important drag force of a long tether. This will be discussed in the following Section.

1.5 Our Scientific Focus: Vision Servoing of a Pair of Robots in a Chain of Mini-ROVs

The long term objective of this project is to design an active tether management solution for mini-ROV missions of long range displacements within cluttered environments and shallow waters.

We choose to control the tether shape by adding several robots linked together all along it. We call this concept the **chain of mini-ROVs** (see Figure 1.9). The robots play the role of actuators and change the whole tether shape depending on the situation at hand. Our strategy is to avoid any contact with obstacles. If the depth is too narrow, it would be better to maintain the tether more taut in order to prevent it from dragging on the seabed. Otherwise, if the environment is more spacious, the tether can be more slack in order to give more freedom of motion to the robots.

The robots that compose the chain are compact, lightweight and with a limited sensory payload. Thus, we choose to investigate the use of the onboard camera to perceive and estimate a parameterized geometric model of the cable. The main focus of this thesis is to manage the tether linking two robots through visual feedback. These robots are named leader, for the front vehicle, and follower, for the rear vehicle. The thesis objective can be therefore summarized by the following question:

How can we manage a tether link between two successive robots, a leader and a follower, within the chain through visual sensory feedback?

The proposed chain of mini-ROVs will operate under the set of assumptions listed below:

A.1.1 the maximum distance between robots is about 10 meters;

A.1.2 the robots can navigate at slightly different depths (difference less than 5 meters, depending on the tether length);

A.1.3 the roll and pitch motion of the robots are mechanically stabilized or regulated at low level to keep the vehicle horizontal;

A.1.4 the robots are equipped with a frontal and/or a rear camera that films the tether;

A.1.5 each robot within the chain must manage the tether segment preceding it;

A.1.6 the leader robot should not manage any portion of the tether, being free to explore its surroundings and execute other tasks;

A.1.7 the leader robot may be outside the follower camera field of view, but a portion of the tether is always visible.

A.1.8 the tether is detectable in the camera image flow;

A.1.9 the tether linking both robots is negatively buoyant and the tether plane remains in the vertical. The tether lowest point is always situated between both robots;

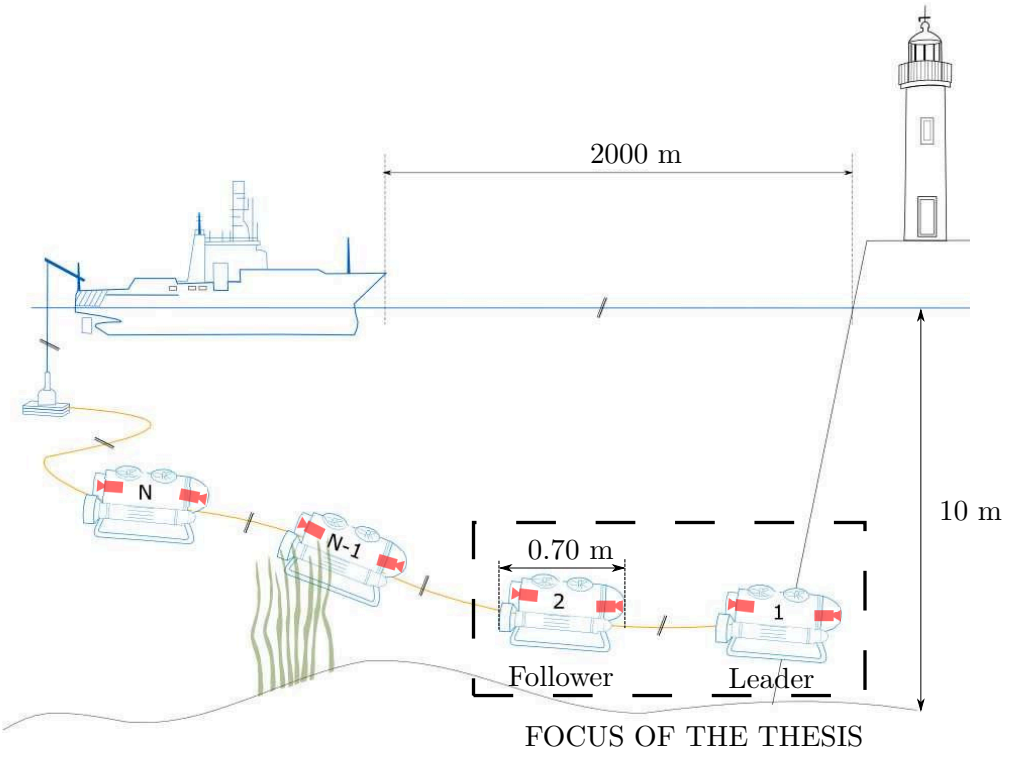
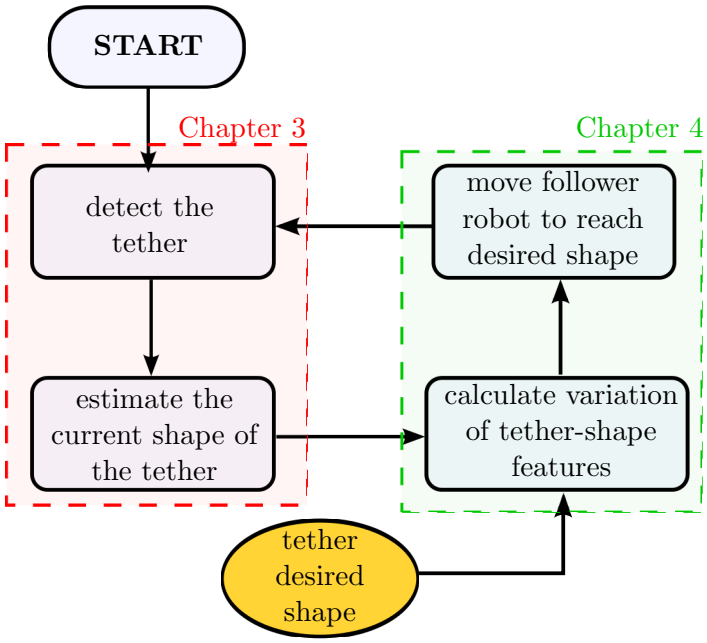


Figure 1.9: A chain of N compact tethered robots used to explore shallow waters. The thesis focuses on the shape control of the tether link between the first two robots, namely the leader and its immediate follower. The robots that compose the chain are light-weight mini-ROVs equipped with a frontal and rear cameras and an IMU.

The maximum distance between the robots is limited because the modeling errors would be propagated over a too long length of tether and then affect our management strategy. A camera mounted on the front of the robot will be used to give real-time informations about the current tether shape. The camera is supposed to be near the tether attachment point on the robot so that at least a portion of the tether can be captured all along the mission. This also gives more freedom of motion to the leader robot that is not required to be in the field of view of the follower.

As depicted in Figure 1.10, the tether management strategy we proposed can be decomposed into three main steps. First, the tether is detected in the camera image. Secondly, the detected points are used to estimate a parametrized geometric model that fits the tether observed shape. Third, the current tether parameters are entered in a control loop that will displace the tether attachment points so that a desired shape of the tether is reached. Actually, the tether desired shape is obtained through the regulation of robots relative pose. Both robots could enter in the control loop, but we took the choice of leaving the leader robot free to move in the environment and execute other tasks. The follower robot, in turn, will be in charge of moving itself in order to regulate the tether shape. Each step mentioned above will be developed in the following three Chapters of this document, as presented in right side of Figure 1.10.



Chapter 2: **system modeling**
the tether and robot models are introduced

Chapter 3: **tether perception**
the proposed method for tether perception and shape estimation is introduced

Chapter 4: **tether shape control**
the proposed method for tether shape control is developed

Chapter 5: **Conclusions**
concluding remarks are given and work perspectives are analyzed

Figure 1.10: Left: simplified algorithm chart flow of the proposed vision-based tether shape control scheme. Right: short presentation of the following Chapters content.

Chapter 2

System Modeling

This Chapter presents the model of the tether used and the robotic system composed of two underwater robots linked together by the tether, which is under study in this thesis. This set is the initial portion of the chain of tethered robots presented in the previous Chapter. The robots do not have the same task in this system: one is the leader, and has as main task the exploration of its surroundings; the other is the follower, behind the leader, whose main task is the control of the tether shape in order not to hamper the leader movements. The follower robot should manage the tether shape primarily using its own sensory feedback. In this thesis, the objective is to use the tether as it is, and not considering additional sensors that could equip the tether to obtain sensory feedback on its current shape. The Chapter is organized in three parts. The first part deals with the catenary equations, the second part presents the model of the robots and the last part gives the relation between the kinematics of the robots and the kinematics of the attachment points.

2.1 Tether Model

The underwater tether we use in this work is slightly negative buoyant, which is frequently the case of tethers and umbilicals that transfer power. They are used in maintenance, survey (see Section 1.3.2) and archaeological missions (Khatib et al., 2016). We assume that our tether is a perfectly inextensible and flexible cable with a constant transversal section and constant linear density. From these assumptions, we choose to model the tether as a catenary.

2.1.1 Catenary Equation

The catenary curve is the shape of a perfectly flexible, inextensible hanging cable that is subject to uniform load distribution along itself and that is supported at its extremities (Johnston et al., 2009).

Figure 2.1 presents a scheme of a hanging cable attached at points $\mathbf{P}_1$ and $\mathbf{P}_2$. The cable lowest point, $\mathbf{P}_0$, is supposed to be at the center of the coordinate frame

$\mathcal{F}_0(\mathbf{P}_0, \mathbf{i}_0, \mathbf{j}_0, \mathbf{k}_0)$, where $\mathbf{k}_0$ is vertical, $\mathbf{i}_0$ and $\mathbf{j}_0$ are horizontal axes. Appendix A summarizes the notations used for position and orientation representation of points and solids in the Euclidean space.

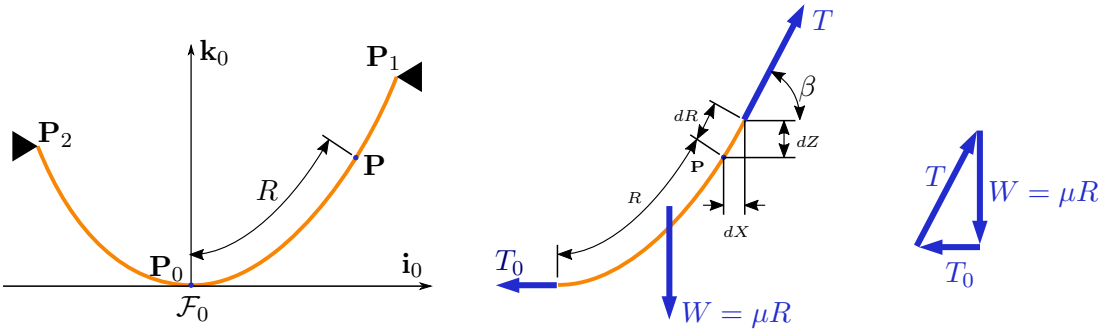


Figure 2.1: A cable hanging under its own weight and attached to points $\mathbf{P}_2$ and $\mathbf{P}_1$. The cable linear weight is given by μ, expressed in N/m.

We denote the load per unit length by μ, expressed in N/m. The magnitude W of the total load carried by a portion of cable with a length of R, extending from the lowest point $\mathbf{P}_0$ to some generic point $\mathbf{P}$, is:

$$W = \mu R. \tag{2.1}$$

Therefore, the tension at $\mathbf{P}$ can be calculated as

$$T = \sqrt{T_0^2 + \mu^2 R^2}, \tag{2.2}$$

where T_0 is the tension at point $\mathbf{P}_0$. In order to simplify, let us introduce the catenary constant

$$C = \frac{\mu}{T_0}. \tag{2.3}$$

Replacing it in equation (2.2) yields:

$$T = \mu\sqrt{1/C^2 + R^2}. \tag{2.4}$$

Let us now define the horizontal projection of an small element of cable length dR as:

$$dX = \cos\beta dR. \tag{2.5}$$

Observing from Figure 2.1 that $\cos\beta = \frac{T_0}{T}$ and using equations (2.3), (2.4) and (2.5) we have

$$dX = \frac{T_0}{T} dR = \frac{dR}{\sqrt{1 + R^2C^2}}. \tag{2.6}$$

Integrating equation (2.6) from $\mathbf{P}_0$ to $\mathbf{P}$, we obtain

$$X = \int_0^R \frac{dR}{\sqrt{1 + R^2C^2}} = \frac{1}{C}\sinh^{-1}(RC) \tag{2.7}$$

that can be rewritten as

$$R = \frac{1}{C}\sinh\left(CX\right), \tag{2.8}$$

which is the geometric expression of the catenary cable half-length.

We can now write that:

$$dZ = \tan\beta dX. \tag{2.9}$$

Observing from Figure 2.1 that $\tan\beta = \frac{W}{T_0}$ and using equations (2.1), (2.3), (2.8) and (2.9), we have:

$$dZ = \sinh\left(CX\right)dX. \tag{2.10}$$

Integrating from $\mathbf{P}_0$ to $\mathbf{P}$ we obtain

$$Z = \int_0^X \sinh\left(CX\right)dX, \tag{2.11}$$

which reduces to the equation of the catenary cable in the coordinate frame $\mathcal{F}_0$ centered at the lowest point of the catenary, namely $\mathbf{P}_0$:

$$Z = \frac{1}{C}\left[\cosh\left(CX\right) - 1\right]. \tag{2.12}$$

2.1.2 Catenary Parameter

The catenary parameter C depends on the height difference between attachment points ΔH, the cable slackness H and the cable total length L. Figure 2.2 depicts the notation used for the catenary model.

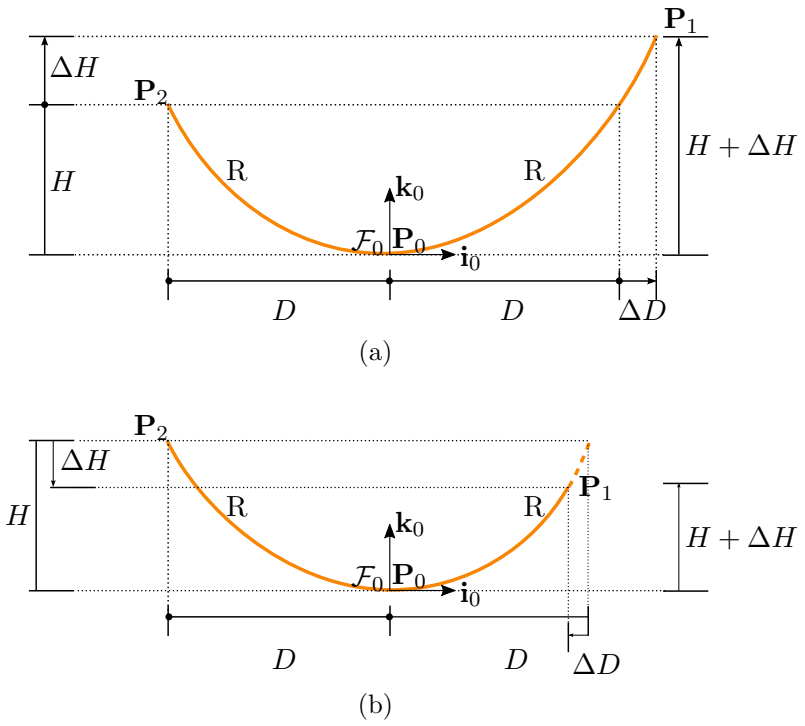


Figure 2.2: Illustration of a catenary with attachment points $\mathbf{P}_2$ and $\mathbf{P}_1$, and lowest point $\mathbf{P}_0$. (a) an example of catenary with $\Delta H > 0$. (b) an example of catenary with $\Delta H < 0$.

In order to calculate the catenary parameter C as a function of the H, ΔH and L we express that the attachment points belong to the catenary and that the total length of the catenary is constant. Knowing that the coordinates of the attachment points in frame $\mathcal{F}_0$ are ${}^0\overline{\mathbf{P}}_2 = [-D,\ 0,\ H]^T$ and ${}^0\overline{\mathbf{P}}_1 = [D+\Delta D,\ 0,\ H+\Delta H]$, we obtain, thanks to equation (2.12), that:

$$H = \frac{1}{C}\left[\cosh(CD) - 1\right] \text{ and} \tag{2.13}$$

$$H + \Delta H = \frac{1}{C}\left[\cosh(C(D + \Delta D) - 1\right]. \tag{2.14}$$

Using equation (2.8) we can calculate the total length of the catenary:

$$L = \frac{1}{C}\left[\sinh(C(D + \Delta D)) + \sinh(CD)\right]. \tag{2.15}$$

The analytical solving of this equation leads to a second order equation for C. One of the solutions is a catenary where the lowest point is between the cable attachment points and therefore belongs to the cable. The other solution is a catenary where the lowest point does not belong the cable.

Appendix B gives the details of the calculation of the catenary parameter C, whose expression related to the lowest point belonging to the catenary is:

$$C = \frac{2}{L^2 - (2H + \Delta H)^2} \left(2H + \Delta H + 2L\sqrt{H\frac{H + \Delta H}{L^2 - \Delta H^2}} \right) \tag{2.16}$$

which can be rewritten as:

$$C = 2\frac{C_n}{C_d} = 2\frac{2H + \Delta H + 2L\sqrt{H\frac{H+\Delta H}{L^2-\Delta H^2}}}{L^2 - (2H + \Delta H)^2} \tag{2.17}$$

In the case where the attachment points are at the same height, $\Delta H = 0$, and we obtain the following expression for C:

$$C = \frac{2H}{R^2 - H^2}, \tag{2.18}$$

where R is the catenary half-length.

2.1.3 Catenary Parameter Constraints

The conditions of study of the catenary relative to cable length L, slackness H and difference of attachment points height ΔH are:

$$H > 0 \tag{2.19}$$

$$H + \Delta H > 0 \tag{2.20}$$

$$2H + \Delta H < L. \tag{2.21}$$

These inequalities ensure the validity of equation (2.17), where the term below the square root must be positive, as well as the denominator.

Equations (2.19) and (2.20) state that the catenary lowest point is between the attachment points. If ΔH is positive (see Figure 2.2a), the cable should be long enough to cumulate twice the slackness length H and the difference of height ΔH. The limit case, $H = 0$, is when the attachment point $\mathbf{P}_2$ is the lowest point of the catenary. If ΔH is negative (see Figure 2.2b), the slackness H should be greater than the difference of height so that the catenary lowest point still remains between the attachment points. The limit case, $H + \Delta H = 0$, is when the attachment point $\mathbf{P}_1$ is the lowest point of the catenary.

The inequalities (2.21) ensures that parameter C remains positive which means that the catenary is not degenerated. The degenerated case, $L = 2H + \Delta H$, occurs when both attachment points and the lowest point are on the same vertical.

2.2 Robots Model and Configuration

Small teleoperated underwater robots, classified as mini-ROVs, are preferable to the execution of missions in confined environments or in shallow coastal waters thanks to their high maneuverability and small size. In this thesis, we choose to use the 6-thruster model of mini-ROVs inspired from the BlueROV experimental platform from Blue Robotics. There are two models, the first one is related to the BlueROV1 and the second one to the BlueROV2 (see Figure 2.3). A comparison between both robots is presented in table 2.1.

(a) (b)

Figure 2.3: The experimental robots used in this thesis: (a) the BlueROV1 and (b) the BlueROV2, both from Blue Robotics.

	BlueROV1	**BlueROV2**
Thrusters configuration	direct 3 vertical 2 horizontal 1 lateral	vectorial 2 vertical 4 horizontal
number of controllable DOF	6	5
number of watertight enclosures	one	two (electronic and battery)
weight	8 kg	9 kg
dimensions (LxWxH)	483 x 330 x 267 (mm)	457 x 338 x 254 (mm)
architecture	Raspiberry 3 model with Pixhawk autopilot	
sensors	IMU, Raspicam and external pressure gauge	

Table 2.1: A comparison of the some characteristics of the two robots used in this thesis.

The architecture of the robots is composed of a main board Raspberry Pi 3 Model (Ethernet, micro SD, GPIO ports, CPU Quad Core 1.2GHz ARM Cortex A53 (ARMv8)

Broadcom BCM2837 64bit, 1GB of RAM), which is connected via USB to a general purpose microcontroller, namely a Pixhawk autopilot microcontroller that was designed for aerial drones. The microcontroller has the following internal sensors: a 3-axis accelerometer and gyroscope (Invensense MPU6000), a 3-axis gyroscope (ST Micro L3GD20H), a 3D e-compass (ST Micro LSM303D) and a barometer (MEAS MS5611).

Both robots are equipped with an external pressure sensor (the Bar30 model from BlueRobotics) that is connected to the Pixhawk and used to measure water pressure. The Pixhawk controls the six ESCs (electronic speed controllers) that manage the power and rotation velocity commands sent to the thrusters. Each robot is also equipped with a Raspberry Pi Camera. The onboard computer communicates with a remote workstation via a Fathom-X Tether Interface that provides robust high-speed Ethernet connection. The tether attached to the robot only provides data exchange with the workstation. The power is delivered by an embedded battery (a LiPo 3S 11.1V 5000mAh) that ensures an average autonomy of 1 hour. A schematic overview of the robot components is presented in Figure 2.4.

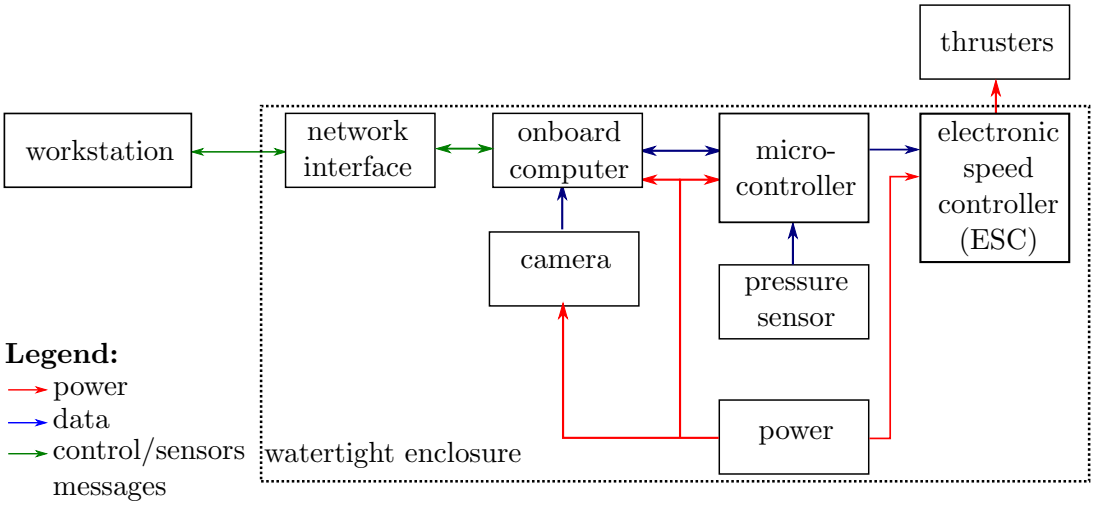
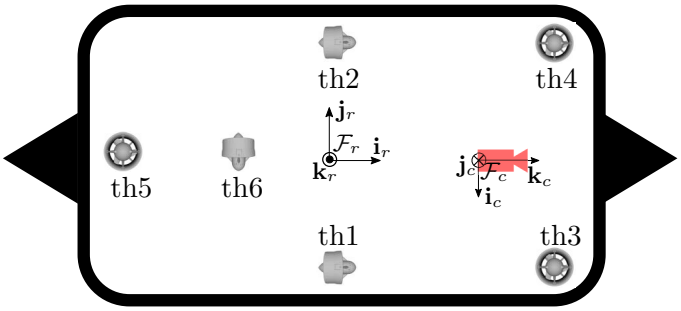


Figure 2.4: A scheme presenting the components of the BlueROVs. The watertight zone is delimited by the dashed rectangle.

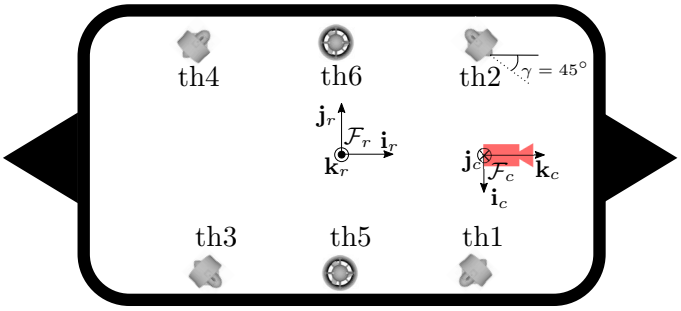
2.2.1 Thruster Configuration and Allocation Matrix

The main difference between the two models is the thruster configuration: the BlueROV1 has a **direct** thrusters configuration and the BlueROV2 has a **vectored** thrusters configuration.

Figure 2.5 presents, for both robots, a scheme containing the camera and thrusters positions and orientations. We note $\mathcal{F}_r(\mathbf{P}_r, \mathbf{i}_r, \mathbf{j}_r, \mathbf{k}_r)$ the coordinate frame linked to robot r. $\mathbf{P}_r$ is the center of the robot, $\mathbf{i}_r$, $\mathbf{j}_r$, $\mathbf{k}_r$ are respectively the robot's longitudinal, lateral and vertical axes. For the BlueROV1, $\mathbf{P}_r$ is at the middle of the segment linking



(a)



(b)

Figure 2.5: A top view scheme with thrusters and camera positions for (a) BlueROV1 and (b) BlueROV2. The BlueROV2 horizontal thrusters orientation angle is given by $\gamma = 45°$.

thrusters th_1 and th_2, which are used to generate longitudinal and yaw motion. For the BlueROV2, $\mathbf{P}_r$ is at the center of the rectangle whose vertices are the center of the thrusters th_1, th_2, th_3 and th_4, which are used to generate longitudinal, lateral and yaw motion. The camera frame is noted $\mathcal{F}_c(\mathbf{P}_c, \mathbf{i}_c, \mathbf{j}_c, \mathbf{k}_c)$ and details about its orientation with respect to the robot frame are given in Chapter 3.

Tables 2.2 and 2.3 give the corresponding thrusters and cameras orientations and positions for BlueROV1 and BlueROV2, respectively. The thruster configuration of the BlueROV1 enables to control the 6 DOF of the robot, whereas the thruster configuration of BlueROV2 does not allow the control of the pitch DOF. All the motors are T200 models from BlueRobotics (350 W, 25 A of max. current, 5.1 kgf of max. thrust, 300-3800 rev/min of rotation speed). Details about the motors characteristics can be found in BlueRobotics (2019).

Positions	X	Y	Z
${}^{r}\overline{\mathbf{P}}_{th1}$	0	-0.115	0
${}^{r}\overline{\mathbf{P}}_{th2}$	0	0.115	0
${}^{r}\overline{\mathbf{P}}_{th3}$	0.175	-0.115	0
${}^{r}\overline{\mathbf{P}}_{th4}$	0.175	0.115	0
${}^{r}\overline{\mathbf{P}}_{th5}$	-0.175	0	-0.100
${}^{r}\overline{\mathbf{P}}_{th6}$	-0.05	0	-0.120
${}^{r}\overline{\mathbf{P}}_{c}$	0.17	0	0

Table 2.2: BlueROV1 thrusters and camera positions with respect to the robot frame. The measurements are in meters. An illustration showing the robot thrusters and camera positions is given in Figure 2.5a.

Positions	X	Y	Z
${}^{r}\overline{\mathbf{P}}_{th1}$	0.135	-0.115	0.0
${}^{r}\overline{\mathbf{P}}_{th2}$	0.135	0.115	0.0
${}^{r}\overline{\mathbf{P}}_{th3}$	-0.135	-0.115	0.0
${}^{r}\overline{\mathbf{P}}_{th4}$	-0.135	0.115	0.0
${}^{r}\overline{\mathbf{P}}_{th5}$	0	-0.115	0.070
${}^{r}\overline{\mathbf{P}}_{th6}$	0	0.115	0.070
${}^{r}\overline{\mathbf{P}}_{c}$	0.17	0	0.070

Table 2.3: BlueROV2 thrusters and camera positions with respect to the robot frame. The measurements are in meters. An illustration showing the robot thrusters and camera positions is given in Figure 2.5b.

In model-based control of underwater robots, the vehicle motion is regulated through the thrust forces generated by the propellers. Thus, a thruster allocation system maps the command forces calculated by the vehicle motion controller into the command of each actuator (Fossen et al., 2009).

Figure 2.6 describes the conversion chain from the desired body thrust commands to the pulse width modulation (PWM) signals sent to the ESC circuits. This chain is implemented inside the microcontroller. The first step is the conversion from the body thrusts to the actuators thrusts. The second step consists of converting the actuator thrusts into motor commands in the shape of PWM signals. The body thrust vector can be denoted, in the body-fixed frame, by (Antonelli et al., 2008):

$$\boldsymbol{\tau}_{body} = \begin{bmatrix} F_x \\ F_y \\ F_z \\ \Gamma_x \\ \Gamma_y \\ \Gamma_z \end{bmatrix}, \tag{2.22}$$

where (F_x, F_y, F_z) and $(\Gamma_x, \Gamma_y, \Gamma_z)$ are, respectively, the components of the resultant forces and moments acting on the center of the vehicle, namely $\mathbf{P}_r$. The actuator thrust vector can be denoted by:

$$\boldsymbol{\tau}_{actuators} = \begin{bmatrix} F_{th_1} \\ F_{th_2} \\ F_{th_3} \\ F_{th_4} \\ F_{th_5} \\ F_{th_6} \end{bmatrix}, \tag{2.23}$$

where F_{th_i} is the thrust force generated by the thruster th_i.

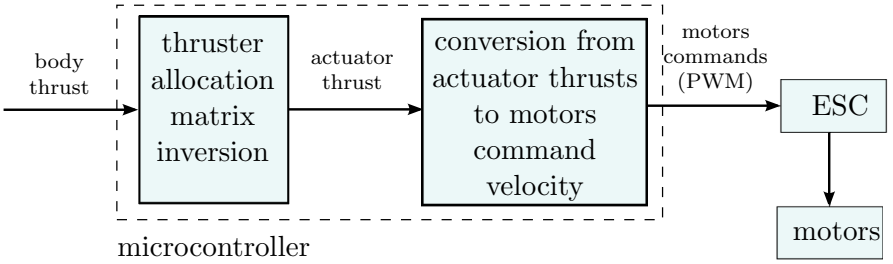


Figure 2.6: A block diagram illustrating the low level thrust conversion from the vehicle desired body thrust to the PWM commands sent to the motors. The steps implemented in the robot micro-controller are surrounded by a black dashed line.

The relation between $\boldsymbol{\tau}_{body}$ and $\boldsymbol{\tau}_{actuators}$ can be expressed as (Creuze, 2014):

$$\boldsymbol{\tau}_{body} = \mathbf{T}\mathbf{W}\boldsymbol{\tau}_{actuators}, \tag{2.24}$$

where $\mathbf{T}$ is called the thruster allocation matrix and $\mathbf{W}$ is a weighting matrix that takes into account the thrusters specificities. As the thrusters are all the same for the BlueROVs, we have $\mathbf{W} = \mathbf{I}_{6\times 6}$.

The thruster allocation matrix is given by:

$$\mathbf{T} = \begin{bmatrix} \mathbf{u}_1 & \cdots & \mathbf{u}_i & \cdots & \mathbf{u}_n \\ {}^r\overline{\mathbf{P}}_{th_1} \times \mathbf{u}_1 & \cdots & {}^r\overline{\mathbf{P}}_{th_i} \times \mathbf{u}_i & \cdots & {}^r\overline{\mathbf{P}}_{th_n} \times \mathbf{u}_n \end{bmatrix} \tag{2.25}$$

where ${}^r\overline{\mathbf{P}}_{th_i}$ is the thruster th_i position with respect to the robot center, and $\mathbf{u}_i$ is the unit vector giving the thrust direction of thruster th_i. The total number of thrusters is n.

For the BlueROV1 the allocation matrix is given by

$$\mathbf{T}_{BR1} = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 \\ 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & -d_Y & d_Y & 0 & -d_{Z_6} \\ 0 & 0 & -d_X & -d_X & d_X & 0 \\ d_Y & -d_Y & 0 & 0 & 0 & d_{X_6} \end{bmatrix}, \tag{2.26}$$

where $d_X = 0.175\text{m}$, $d_Y = 0.115\text{m}$, $d_{X_6} = 0.050\text{m}$ and $d_{Z_6} = 0.175\text{m}$ (see table 2.2). The inverse matrix is given by:

$$\mathbf{T}_{BR1}^{-1} = \begin{bmatrix} 0.5 & \frac{d_{X_6}}{2d_Y} & 0 & 0 & 0 & \frac{1}{2d_Y} \\ 0.5 & -\frac{d_{X_6}}{2d_Y} & 0 & 0 & 0 & -\frac{1}{2d_Y} \\ 0 & \frac{d_{Z_6}}{2d_Y} & 0.25 & -\frac{1}{2d_Y} & -\frac{1}{4d_X} & 0 \\ 0 & -\frac{d_{Z_6}}{2d_Y} & 0.25 & \frac{1}{2d_Y} & -\frac{1}{4d_X} & 0 \\ 0 & 0 & 0.5 & 0 & \frac{1}{2d_X} & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 \end{bmatrix}. \tag{2.27}$$

The motors configuration of the BlueROV1 allocates only one motor – thruster 6 – for the lateral motion. As a consequence, the single lateral thrust will also generate roll and yaw motion, since the motor is not positioned at center of the robot. In case the positioning of the motor is not accurate, it can generate disturbing roll and yaw motion. Moreover, the single lateral thruster is also responsible for a disturbing pitch motion because the propeller rotation exerts a torque on the robot body frame, according to the action/reaction principle. These disturbances could be compensated by adding counteracting roll/pitch/yaw thrust terms that depend on the lateral thrust. The compensation matrix that could be applied after the $\mathbf{T}_{BR1}$ matrix is:

$$\mathbf{C}_{BR1} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & \delta_x & 0 & 1 & 0 & 0 \\ 0 & \delta_y & 0 & 0 & 1 & 0 \\ 0 & \delta_z & 0 & 0 & 0 & 1 \end{bmatrix}, \tag{2.28}$$

where δ_x, δ_y and δ_z have to be determined empirically. The equation (2.24) can be hence rewritten as:

$$\boldsymbol{\tau}_{body} = \mathbf{C}_{BR1}\mathbf{T}_{BR1}\boldsymbol{\tau}_{actuators}. \tag{2.29}$$

If we want to control the actuator, we need to inverse the previous equation to get:

$$\boldsymbol{\tau}_{actuators} = (\mathbf{C}_{BR1}\mathbf{T}_{BR1})^{-1}\boldsymbol{\tau}_{body}, \tag{2.30}$$

where

$$(\mathbf{C}_{BR1}\mathbf{T}_{BR1})^{-1} = \begin{bmatrix} 0.5 & \frac{-\delta_z + d_{X_6}}{2d_Y} & 0 & 0 & 0 & \frac{1}{2d_Y} \\ 0.5 & -\frac{-\delta_z + d_{X_6}}{2d_Y} & 0 & 0 & 0 & \frac{-1}{2d_Y} \\ 0 & \frac{\delta_y}{4d_X} - \frac{-\delta_x - d_{Z_6}}{2d_Y} & 0.25 & \frac{-1}{2d_Y} & \frac{-1}{4d_X} & 0 \\ 0 & \frac{\delta_y}{4d_X} + \frac{-\delta_x - d_{Z_6}}{2d_Y} & 0.25 & \frac{1}{2d_Y} & \frac{-1}{4d_X} & 0 \\ 0 & \frac{-\delta_y}{2d_X} & 0.5 & 0 & \frac{1}{2d_X} & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 \end{bmatrix}. \tag{2.31}$$

The values of δ_x and δ_z are used to compensate the inaccuracies of measurements of d_{X_6} and d_{Z_6}, while δ_y is mainly used to compensate the disturbing pitch motion due to the propeller rotation of thruster 6.

For the BlueROV2 the allocation matrix is given by:

$$\mathbf{T}_{BR2} = \begin{bmatrix} \cos\gamma & \cos\gamma & -\cos\gamma & -\cos\gamma & 0 & 0 \\ \sin\gamma & -\sin\gamma & \sin\gamma & -\sin\gamma & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & -d_Y & d_Y \\ d_\perp & -d_\perp & -d_\perp & d_\perp & 0 & 0 \end{bmatrix}, \tag{2.32}$$

where γ is the angle between $\mathbf{i}_r$ and the thruster direction axis, $d_\perp = d_X s\gamma + d_Y c\gamma$, $d_X = 0.135$m, $d_Y = 0.115$m, $s\gamma = \sin\gamma$ and $c\gamma = \cos\gamma$ (see table 2.3). The body thrust vector is now given by:

$$\boldsymbol{\tau}_{body} = \begin{bmatrix} F_x \\ F_y \\ F_z \\ \Gamma_x \\ \Gamma_z \end{bmatrix}, \tag{2.33}$$

since the pitch motion is not controllable. The allocation matrix can be decoupled into two submatrices, one that deals with longitudinal, lateral and yaw motion, and the other one that deals with vertical and roll motion:

$$\begin{bmatrix} F_x \\ F_y \\ \Gamma_z \end{bmatrix} = \begin{bmatrix} \cos\gamma & \cos\gamma & -\cos\gamma & -\cos\gamma \\ \sin\gamma & -\sin\gamma & \sin\gamma & -\sin\gamma \\ d_\perp & -d_\perp & -d_\perp & d_\perp \end{bmatrix} \begin{bmatrix} F_{th_1} \\ F_{th_2} \\ F_{th_3} \\ F_{th_4} \end{bmatrix} = \mathbf{M} \begin{bmatrix} F_{th_1} \\ F_{th_2} \\ F_{th_3} \\ F_{th_4} \end{bmatrix} \tag{2.34}$$

$$\begin{bmatrix} F_z \\ \Gamma_x \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ -d_Y & d_Y \end{bmatrix} \begin{bmatrix} F_{th_5} \\ F_{th_6} \end{bmatrix} \tag{2.35}$$

The equation (2.35) can be directly inverted to get

$$\begin{bmatrix} F_{th_5} \\ F_{th_6} \end{bmatrix} = \begin{bmatrix} 0.5 & -0.5/d_Y \\ 0.5 & 0.5/d_Y \end{bmatrix} \begin{bmatrix} F_z \\ \Gamma_x \end{bmatrix}. \tag{2.36}$$

The matrix $\mathbf{M}$ is not square. However, since $\gamma = 45°$, matrix $\mathbf{M}\mathbf{M}^T$ is invertible and we can use the Moore-Penrose pseudo-inverse to express the actuator vector as a function of the desired body thrust vector:

$$\begin{bmatrix} F_{th_1} \\ F_{th_2} \\ F_{th_3} \\ F_{th_4} \end{bmatrix} = \mathbf{M}^+ \begin{bmatrix} F_x \\ F_y \\ \Gamma_z \end{bmatrix} \tag{2.37}$$

with $\mathbf{M}^+ = \mathbf{M}^T(\mathbf{M}\mathbf{M}^T)^{-1}$.

Another solution to obtain the actuators commands is to exploit the vectorial configuration of motors th_1 to th_4. If we consider a longitudinal displacement, and we choose to have the same thrust produced by th_1 and th_2 for the one part, and th_3 and th_4 for the other part, we can set:

$$F_{th_1} + F_{th_2} = -\left(F_{th_3} + F_{th_4}\right). \tag{2.38}$$

In the same way, for the lateral displacement we have:

$$F_{th_1} + F_{th_3} = -\left(F_{th_2} + F_{th_4}\right). \tag{2.39}$$

Finally, for the yaw displacement, we have:

$$F_{th_1} + F_{th_4} = -\left(F_{th_2} + F_{th_3}\right). \tag{2.40}$$

Actually, the equations (2.38) to (2.40) are the same:

$$F_{th_1} + F_{th_2} + F_{th_3} + F_{th_4} = 0. \tag{2.41}$$

This equation can be used as an additional equation to invert the system of equations defined by (2.34). Hence, the system to be inverted becomes:

$$\begin{bmatrix} 0 \\ F_x \\ F_y \\ \Gamma_z \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ \cos\gamma & \cos\gamma & -\cos\gamma & -\cos\gamma \\ \sin\gamma & -\sin\gamma & \sin\gamma & -\sin\gamma \\ d_\perp & -d_\perp & -d_\perp & d_\perp \end{bmatrix} \begin{bmatrix} F_{th_1} \\ F_{th_2} \\ F_{th_3} \\ F_{th_4} \end{bmatrix}, \tag{2.42}$$

whose inversion leads to:

$$\begin{bmatrix} F_{th_1} \\ F_{th_2} \\ F_{th_3} \\ F_{th_4} \end{bmatrix} = \begin{bmatrix} 0.25 & 0.25/\cos\gamma & 0.25/\sin\gamma & 0.25/d_\perp \\ 0.25 & 0.25/\cos\gamma & -0.25/\sin\gamma & -0.25/d_\perp \\ 0.25 & -0.25/\cos\gamma & 0.25/\sin\gamma & -0.25/d_\perp \\ 0.25 & -0.25/\cos\gamma & -0.25/\sin\gamma & 0.25/d_\perp \end{bmatrix} \begin{bmatrix} 0 \\ F_x \\ F_y \\ \Gamma_z \end{bmatrix} \tag{2.43}$$

$$\begin{bmatrix} F_{th_1} \\ F_{th_2} \\ F_{th_3} \\ F_{th_4} \end{bmatrix} = \begin{bmatrix} 0.25/\cos\gamma & 0.25/\sin\gamma & 0.25/d_\perp \\ 0.25/\cos\gamma & -0.25/\sin\gamma & -0.25/d_\perp \\ -0.25/\cos\gamma & 0.25/\sin\gamma & -0.25/d_\perp \\ -0.25/\cos\gamma & -0.25/\sin\gamma & 0.25/d_\perp \end{bmatrix} \begin{bmatrix} F_x \\ F_y \\ \Gamma_z \end{bmatrix} \tag{2.44}$$

The thrusters of the BlueROVs are controlled through Pulse-Width Modulation (PWM) signals that are sent to an ESC (electronic speed control) circuit that regulates the power delivered to the motor. The relation between the desired actuator thrusts calculated by the high-level control law and the PWM signals sent to the ESC circuit is governed by the motor characteristics depicted in Figure 2.7. This involves values of PWM signals that are bounded between 1100 μs and 1900 μs. In order to deal with the deadband around the null velocity, an interpolation of the curve can be used to determine the PWM commands from the thrust desired input (Bessa et al., 2006).

Figure 2.7: T200 motor curve featuring thrust versus PWM commands. Courtesy of BlueRobotics (2019).

Regarding the vectorial configuration like in the BlueROV2, it is possible to deal with the dead-band by applying different thrusts to the pair of propellers (longitudinal: forward/rear pairs, lateral: left/right pairs, yaw: left/right diagonal paris), which comes to introduce a motor regime r_m in the equation (2.41), as:

$$F_{th_1} + F_{th_2} + F_{th_3} + F_{th_4} = r_m. \tag{2.45}$$

This solution involves more power consumption. However, the motor regime r_m can be optimized as proposed by Ropars et al. (2015).

2.2.2 Kinematic Model of a mini-ROV

This subsection looks at the three-dimensional modeling of a mini-ROV, which is a solid (non-articulated) robot.

The robot's state vector is composed of its position and orientation with respect to a world reference frame $\mathcal{F}_W(\mathbf{P}_W, \mathbf{i}_W, \mathbf{j}_W, \mathbf{k}_W)$, with origin denoted $\mathbf{P}_W$, that is supposed to be world-fixed and inertial, and $\mathbf{i}_W = [1,0,0]^T, \mathbf{j}_W = [0,1,0]^T, \mathbf{k}_W = [0,0,1]^T$ (see Figure 2.8). In the nautical field, the axes orientation is frequently used with the world-frame vertical axis ($\mathbf{k}_W$) pointing downward, following the SANAME (Society of Naal and Marine Engineers) convention. In this thesis we have chosen to use the vertical axis pointing upwards, which is the convention commonly used in the others robotic fields. The reason for this is that the control scheme that we propose in Chapter 4 is also applicable to domains other than underwater robotics.

We note $\mathcal{F}_{r_i}(\mathbf{P}_{r_i}, \mathbf{i}_{r_i}, \mathbf{j}_{r_i}, \mathbf{k}_{r_i})$ the coordinate frame linked to robot r_i. $\mathbf{P}_{r_i}$ is the center of the robot, $\mathbf{i}_{r_i}$, $\mathbf{j}_{r_i}$, $\mathbf{k}_{r_i}$ are respectively the robot's longitudinal, lateral and vertical axes.

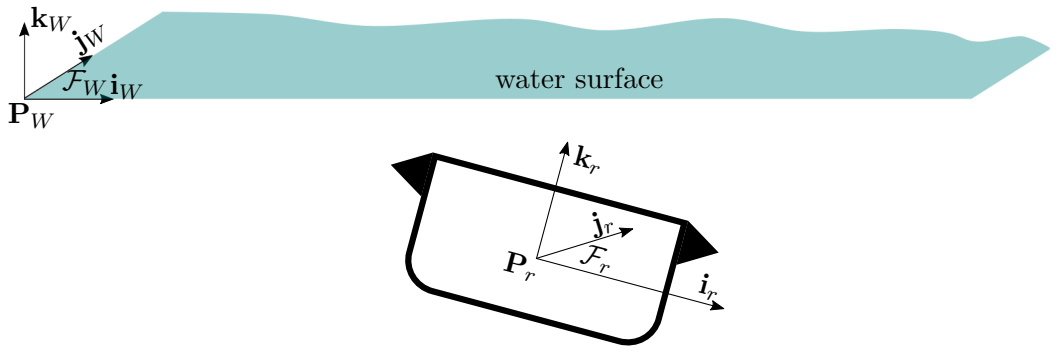


Figure 2.8: The world-fixed and robot-fixed coordinate frames: $\mathcal{F}_W$ and $\mathcal{F}_r$, respectively.

Kinematic Equations in the World Frame Let us define

$$^W\overline{\mathbf{P}}_r = \begin{bmatrix} ^WX_r \\ ^WY_r \\ ^WZ_r \end{bmatrix}, \tag{2.46}$$

the vector containing the position coordinates of center of the robot expressed in the world-fixed reference frame $\mathcal{F}_W$, and

$$^W\boldsymbol{\Omega}_r = \begin{bmatrix} \phi \\ \theta \\ \psi \end{bmatrix}, \tag{2.47}$$

the vector containing the Euler angle coordinates in $\mathcal{F}_W$.

The time derivative of the robot position vector expressed in the world-fixed frame is $^W\dot{\overline{\mathbf{P}}}_r$. If we define

$$^r\boldsymbol{\nu}_r = \begin{bmatrix} \nu_x \\ \nu_y \\ \nu_z \end{bmatrix} \tag{2.48}$$

as the linear velocity vector of the robot-fixed frame $\mathcal{F}_r(\mathbf{P}_r, \mathbf{i}_r, \mathbf{j}_r, \mathbf{k}_r)$ with respect to the world-fixed frame $\mathcal{F}_W$, expressed in the robot-fixed frame, the following relation between the defined linear velocities holds

$$^W\dot{\overline{\mathbf{P}}}_r = {}^W\mathbf{R}_r\,{}^r\boldsymbol{\nu}_r, \tag{2.49}$$

where $^W\mathbf{R}_r \in SO(3)$ is the rotation matrix expressing the transformation from the robot-fixed frame to the world frame.

The time derivative of the orientation vector expressed in the world frame is $^W\dot{\boldsymbol{\Omega}}_r$. Let us define

$$^r\boldsymbol{\omega}_r = \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} \tag{2.50}$$

as the angular velocity vector of the robot frame $\mathcal{F}_r$ with respect to the world-fixed frame $\mathcal{F}_W$, expressed in the robot frame. The relation between ${}^W\dot{\boldsymbol{\Omega}}_r$ and the robot angular velocity is given by:

$$ {}^W\dot{\boldsymbol{\Omega}}_r = {}^W\mathbf{J}_r {}^r\boldsymbol{\omega}_r, \tag{2.51} $$

where ${}^W\mathbf{J}_r \in \mathbb{R}^{3\times 3}$ is a proper Jacobian matrix that can be expressed in terms of the Euler angles as

$$ {}^W\mathbf{J}_r(\boldsymbol{\Omega}) = \frac{1}{\cos\theta} \begin{bmatrix} 1 & \sin\phi\sin\theta & \cos\phi\sin\theta \\ 0 & \cos\phi\cos\theta & -\cos\theta\sin\phi \\ 0 & \sin\phi & \cos\phi \end{bmatrix}, \tag{2.52} $$

which is a singular matrix for $\theta = \frac{\pi}{2}(2l+1)$, with $l \in \mathbb{N}$, i. e., for a pitch angle of $\pm\pi/2$ rad. The inverse of matrix ${}^r\mathbf{J}_W(\boldsymbol{\Omega})$, when possible, is calculated by

$$ {}^r\mathbf{J}_W(\boldsymbol{\Omega}) = {}^W\mathbf{J}_r^{-1}(\boldsymbol{\Omega}) = \begin{bmatrix} 1 & 0 & -\sin\theta \\ 0 & \cos\phi & \cos\theta\sin\phi \\ 0 & -\sin\phi & \cos\theta\cos\phi \end{bmatrix}. \tag{2.53} $$

By bringing together equations (2.49) and (2.51), we obtain the following kinematic model for the robot:

$$ \begin{cases} {}^W\dot{\overline{\mathbf{P}}}_r = {}^W\mathbf{R}_r {}^r\boldsymbol{\nu}_r \\ {}^W\dot{\boldsymbol{\Omega}}_r = {}^W\mathbf{J}_r {}^r\boldsymbol{\omega}_r \end{cases}. \tag{2.54} $$

It is often useful to collect the kinematic equations in six-dimensional matrix forms (Siciliano and Khatib, 2016, chap. 43). Let us thus define the pose vector ${}^W\boldsymbol{\eta}_r \in \mathbb{R}^6$ as

$$ {}^W\boldsymbol{\eta}_r = \begin{bmatrix} {}^W\overline{\mathbf{P}}_r \\ {}^W\boldsymbol{\Omega}_r \end{bmatrix}, \tag{2.55} $$

and the velocity vector ${}^r\mathbf{v}_r \in \mathbb{R}^6$ as

$$ {}^r\mathbf{v}_r = \begin{bmatrix} {}^r\boldsymbol{\nu}_r \\ {}^r\boldsymbol{\omega}_r \end{bmatrix}. \tag{2.56} $$

2.3 Pair of Robots Connected by a Tether

In this Section, the physical system under study is modeled. It is composed of two mini-ROVs linked together by a sagging tether, which corresponds to the first pair of robots that composes the chain of robots described in the previous Chapter. First, the geometric model of a hanging tether linking both robots is introduced. Then, the equation relating the robot velocity with the tether attachment point relative position variation is given.

Figure 2.9 presents the generic nomenclature pattern of coordinate frames inside the chain of robots. The index $i \in [0, n]$ is the robot index inside the chain composed of n robots. The robot's front and rear camera frames are named $\mathcal{F}_{c_{i,1}}$ and $\mathcal{F}_{c_{i,2}}$. Similarly, the tether attachment point frames in the robots are named $\mathcal{F}_{i,1}$ and $\mathcal{F}_{i,2}$, respectively for front and rear attachment. This is a generic nomenclature. The detailed description of our system composed of the first pair of robots and the tether segment linking them is given in the following.

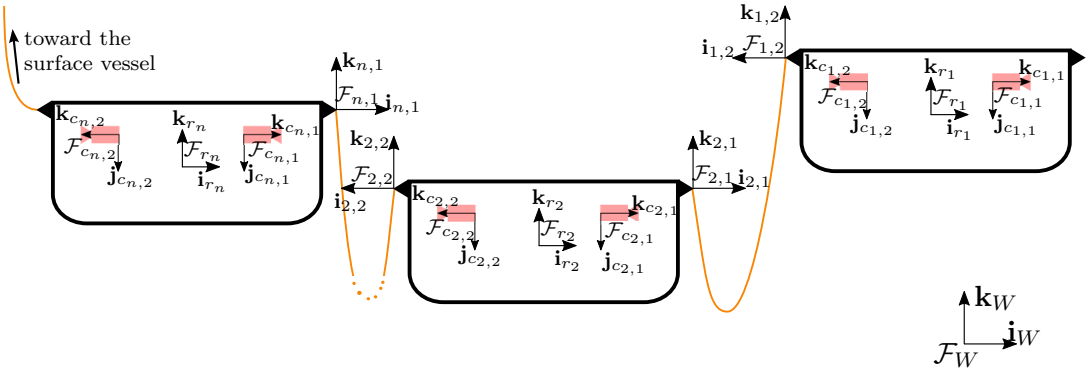


Figure 2.9: An illustration presenting the generic nomenclature pattern of coordinate frames inside the chain of robots.

2.3.1 Catenary Model Applied for Tethered Robots

The catenary model of a tether is suitable for quasi-static analyses (Milutinović et al., 2014). This means that the system has a slow dynamic behavior and the tether attachment points move at a low relative velocity (lower than 0.5 m/s). In such conditions, we can assume that the tether is in static equilibrium at successive instants of time while the robots, and consequently the attachment points, move.

We consider the case of the first two robots of the chain linked by a sagging tether catenary (see Figure 2.10). The robots are named r_1 and r_2, respectively for leader and follower. The tether attachment points frames are named $\mathcal{F}_1$ and $\mathcal{F}_2$, respectively for attachment on the leader and on the follower robot. The coordinate frame named $\mathcal{F}_0$ is at the tether lowest point.

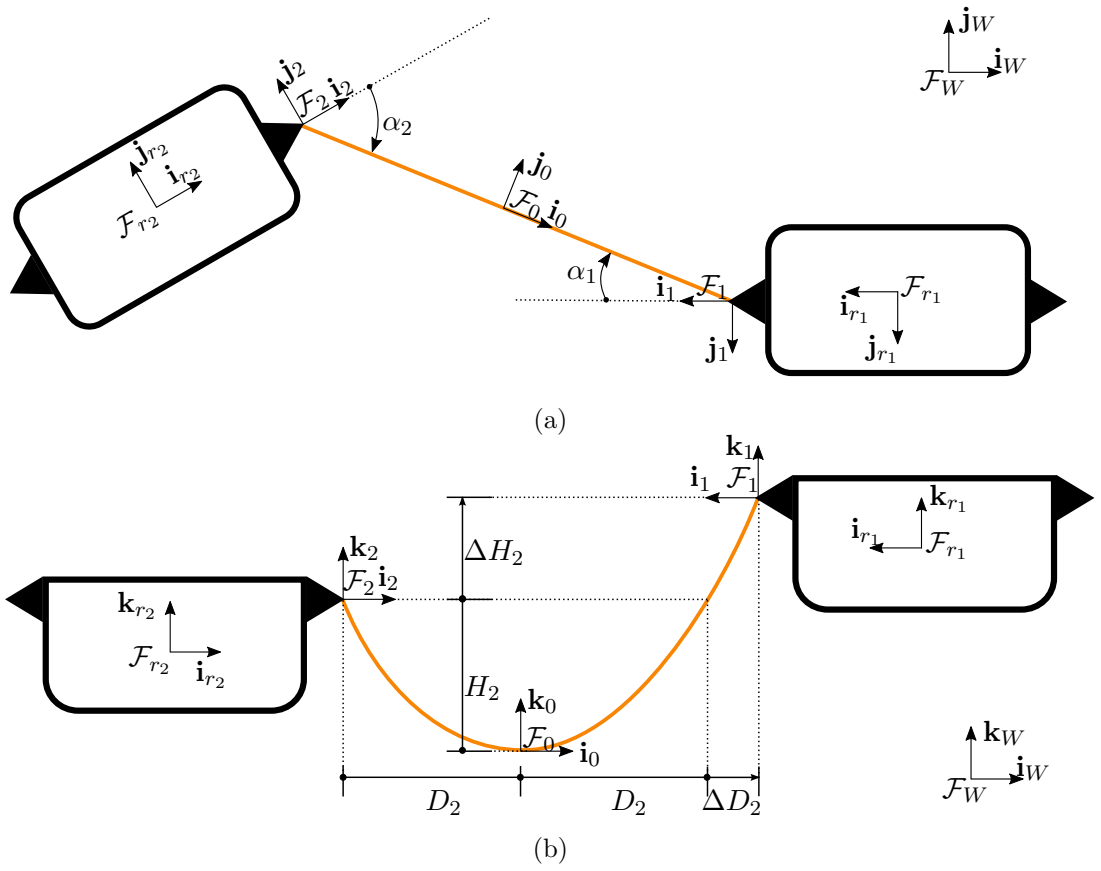


Figure 2.10: Two illustrations presenting the (a) top view and (b) side view of the first pair of robots composing the chain of robots. The tether linking the robots is modeled by a catenary whose slackness is noted H_2. The difference of height between the attachment points from $\mathbf{P}_2$ is ΔH_2. The H_2-slack catenary half-span is noted D_2 and ΔD_2 is the supplementary span due to the height difference. The tether orientation angles with respect to its attachment point frames are noted α_1 and α_2, respectively for leader and follower attachment points.

The tether attachment point coordinate frames are noted $\mathcal{F}_i(\mathbf{P}_i, \mathbf{i}_i, \mathbf{j}_i, \mathbf{k}_i)$, with $i = 1, 2$, and are defined as follows:

$$\mathbf{i}_i = \frac{\mathbf{j}_{r_i} \times \mathbf{k}_W}{\|\mathbf{j}_{r_i} \times \mathbf{k}_W\|} \tag{2.57}$$

$$\mathbf{k}_i = \mathbf{k}_W \tag{2.58}$$

$$\mathbf{j}_i = \mathbf{k}_i \times \mathbf{i}_i. \tag{2.59}$$

This means that $\mathbf{k}_i$ is vertical, $\mathbf{i}_i$ is the intersection of the sagittal plane of the robot r_i with the horizontal plane, and $\mathbf{j}_i$ is horizontal and orthogonal to $\mathbf{i}_i$.

The coordinate frame $\mathcal{F}_1(\mathbf{P}_1, \mathbf{i}_1, \mathbf{j}_1, \mathbf{k}_1)$ is at the tether attachment point on the leader robot r_1. The axis $\mathbf{k}_1$ is vertical, $\mathbf{i}_1$ is parallel to the longitudinal axis of the

robot ($\mathbf{i}_{r_1}$) and points backwards. Axis $\mathbf{j}_1$ is horizontal and orthogonal to $\mathbf{i}_1$. The oriented angle from $\mathbf{i}_1$ to $\mathbf{i}_0$ is named α_1.

The coordinate frame $\mathcal{F}_2(\mathbf{P}_2, \mathbf{i}_2, \mathbf{j}_2, \mathbf{k}_2)$ is at the tether attachment point on the follower robot r_2. The axis $\mathbf{k}_2$ is vertical, $\mathbf{i}_2$ is parallel to the longitudinal axis of the robot ($\mathbf{i}_{r_2}$) and points forwards. Axis $\mathbf{j}_2$ is horizontal and orthogonal to $\mathbf{i}_2$. The oriented angle from $\mathbf{i}_2$ to $\mathbf{i}_0$ is named α_2.

The coordinate frame $\mathcal{F}_0(\mathbf{P}_0, \mathbf{i}_0, \mathbf{j}_0, \mathbf{k}_0)$ is attached to the catenary lowest point (see Section 2.1.1). The axis $\mathbf{k}_0$ is vertical and $\mathbf{i}_0$ is horizontal inside the vertical plane where the catenary is included. The axis $\mathbf{j}_0$ is horizontal and orthogonal to $\mathbf{i}_0$.

The catenary equation can be expressed in the lowest point frame $\mathcal{F}_0$ or in the attachment point frames $\mathcal{F}_1$ and $\mathcal{F}_2$. Since we are interested in this work to estimate and control the tether shape evolution due to the robots motion, we choose to express the catenary equation in the attachment point frames. Actually, $\mathcal{F}_1$ and $\mathcal{F}_2$ are symmetric, and the catenary slackness and half-span presented in Figure 2.10 with respect to $\mathcal{F}_2$ can also be defined with respect to $\mathcal{F}_1$. Figure 2.11 presents the symmetry of the catenary features according to the frame of reference ($\mathcal{F}_1$ or $\mathcal{F}_2$).

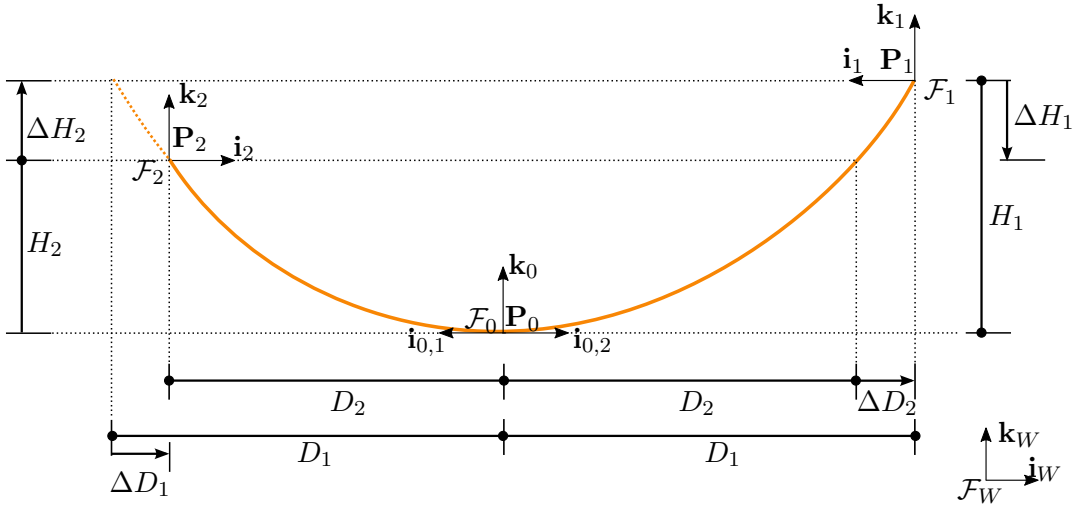


Figure 2.11: An illustration presenting the symmetry of the catenary features according to the reference frame chosen ($\mathcal{F}_1$ or $\mathcal{F}_2$). The catenary slackness are $H_1 = {}^0Z_1$ and $H_2 = {}^0Z_2$. The attachment points difference of height is $\Delta H_1 = {}^1Z_2$ and $H_2 = {}^2Z_1$. The catenary parameter $C(H_1, \Delta H_1) = C(H_2, \Delta H_2)$ is invariant whatever the point $\mathbf{P}_i$ we choose for the starting attachment point of the cable. Thanks to the symmetry we have: $\Delta H_1 = -\Delta H_2$, $H_1 = H_2 + \Delta H_2$, $H_2 = H_1 + \Delta H_1$, $\Delta D_1 = -\Delta D_2$, $D_1 = D_2 + \Delta D_2$, $D_2 = D_1 + \Delta D_1$. As a reminder: $H_i > 0$, $H_i + \Delta H_i > 0$, $D_i > 0$ and $D_i + \Delta D_i > 0$.

Here below we develop the catenary equation in $\mathcal{F}_2$, centered at the follower robot attachment point. The same procedure could be applied to express this equation in $\mathcal{F}_1$.

From equation (2.12), a generic point with homogeneous coordinates ${}^0\mathbf{P} = \left[{}^0X,\ {}^0Y,\ {}^0Z,\ 1\right]^T$

defined in frame $\mathcal{F}_0$, belongs to the catenary if its coordinates are:

$$ {}^0\mathbf{P} = \begin{bmatrix} {}^0X \\ 0 \\ \frac{1}{C}\left[\cosh(C\ {}^0X) - 1\right] \\ 1 \end{bmatrix}. \tag{2.60} $$

We can write the same equation in frame $\mathcal{F}_2$ though the following homogeneous transformation:

$$ {}^2\mathbf{P} = {}^2\mathbf{M}_0\,{}^0\mathbf{P}, \tag{2.61} $$

where

$$ {}^2\mathbf{M}_0 = \begin{bmatrix} \cos\alpha_2 & -\sin\alpha_2 & 0 & D_2\cos\alpha_2 \\ \sin\alpha_2 & \cos\alpha_2 & 0 & D_2\sin\alpha_2 \\ 0 & 0 & 1 & -H_2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.62} $$

is the homogeneous transformation matrix from $\mathcal{F}_0$ to $\mathcal{F}_2$. Developing equation 2.61, we obtain the following equations:

$$ {}^2X = \cos\alpha_2\left({}^0X + D_2\right) \tag{2.63} $$

$$ {}^2Y = \sin\alpha_2({}^0X + D_2) \tag{2.64} $$

$$ {}^2Z = \frac{1}{C}\left[\cosh(C\ {}^0X) - 1\right] - H_2. \tag{2.65} $$

From equations (2.63) and (2.64) we obtain

$$ {}^2X = \cot\alpha_2\ {}^2Y, $$

and from equations (2.64) and (2.65) we obtain

$$ {}^2Z = \frac{1}{C}\left[\cosh\left(C\left(\frac{{}^2Y}{\sin\alpha_2} - D_2\right)\right) - 1\right] - H_2. $$

Both equations form the catenary equation in frame $\mathcal{F}_2$ that is given by:

$$ {}^2\mathbf{P} = \begin{bmatrix} \cot\alpha_2\ {}^2Y \\ {}^2Y \\ \frac{1}{C}\left[\cosh\left(C\left(\frac{{}^2Y}{\sin\alpha_2} - D_2\right)\right) - 1\right] - H_2 \\ 1 \end{bmatrix}. \tag{2.66} $$

2.3.2 Tether Attachment Points and Robots Kinematics

In this section we focus on the tether attachment point kinematics in relationship with robots kinematics. In order to design the robot motion control (Chapter 4), we will need to take into account the kinematics of the tether attachment points with respect to the robot.

The vector defined from $\mathbf{P}_2$ to $\mathbf{P}_1$ is noted $\mathbf{P}_2\mathbf{P}_1$. The time-derivative with respect to frame $\mathcal{F}_2$ is developed to introduce the robots velocities in the following way:

$$\left\{\frac{d\mathbf{P}_2\mathbf{P}_1}{dt}\right\}_{\mathcal{F}_2} = \left\{\frac{d\mathbf{P}_2\mathbf{P}_1}{dt}\right\}_{\mathcal{F}_W} - \boldsymbol{\omega}_2 \times \mathbf{P}_2\mathbf{P}_1,$$

where, $\boldsymbol{\omega}_2$ is the angular velocity of $\mathcal{F}_2$ with respect to $\mathcal{F}_W$. From the Chasles relation we have:

$$\begin{aligned}\left\{\frac{d\mathbf{P}_2\mathbf{P}_1}{dt}\right\}_{\mathcal{F}_2} = &\left\{\frac{d\mathbf{P}_2\mathbf{P}_{r_2}}{dt}\right\}_{\mathcal{F}_W} + \left\{\frac{d\mathbf{P}_{r_2}\mathbf{P}_W}{dt}\right\}_{\mathcal{F}_W} + \left\{\frac{d\mathbf{P}_W\mathbf{P}_{r_1}}{dt}\right\}_{\mathcal{F}_W} \\ &+ \left\{\frac{d\mathbf{P}_{r_1}\mathbf{P}_1}{dt}\right\}_{\mathcal{F}_W} - \boldsymbol{\omega}_2 \times \mathbf{P}_2\mathbf{P}_1.\end{aligned} \tag{2.67}$$

Noting $\left\{\frac{d\mathbf{P}_W\mathbf{P}_{r_1}}{dt}\right\}_{\mathcal{F}_W} = \boldsymbol{\nu}_{r_1}$ and $\left\{\frac{d\mathbf{P}_{r_2}\mathbf{P}_W}{dt}\right\}_{\mathcal{F}_W} = -\boldsymbol{\nu}_{r_2}$, we have:

$$\left\{\frac{d\mathbf{P}_2\mathbf{P}_1}{dt}\right\}_{\mathcal{F}_2} = \left\{\frac{d\mathbf{P}_2\mathbf{P}_{r_2}}{dt}\right\}_{\mathcal{F}_W} - \boldsymbol{\nu}_{r_2} + \boldsymbol{\nu}_{r_1} + \left\{\frac{d\mathbf{P}_{r_1}\mathbf{P}_1}{dt}\right\}_{\mathcal{F}_W} - \boldsymbol{\omega}_2 \times \mathbf{P}_2\mathbf{P}_1$$

Expressing the derivatives $\left\{\frac{d\mathbf{P}_2\mathbf{P}_{r_2}}{dt}\right\}_{\mathcal{F}_W}$ and $\left\{\frac{d\mathbf{P}_{r_1}\mathbf{P}_1}{dt}\right\}_{\mathcal{F}_W}$ in the robot's frames we have:

$$\begin{aligned}\left\{\frac{d\mathbf{P}_2\mathbf{P}_1}{dt}\right\}_{\mathcal{F}_2} = &\left\{\frac{d\mathbf{P}_2\mathbf{P}_{r_2}}{dt}\right\}_{\mathcal{F}_{r_2}} + \boldsymbol{\omega}_{r_2} \times \mathbf{P}_2\mathbf{P}_{r_2} - \boldsymbol{\nu}_{r_2} + \boldsymbol{\nu}_{r_1} \\ &+ \left\{\frac{d\mathbf{P}_{r_1}\mathbf{P}_1}{dt}\right\}_{\mathcal{F}_{r_1}} + \boldsymbol{\omega}_{r_1} \times \mathbf{P}_{r_1}\mathbf{P}_1 - \boldsymbol{\omega}_2 \times \mathbf{P}_2\mathbf{P}_1,\end{aligned} \tag{2.68}$$

and knowing that $\left\{\frac{d\mathbf{P}_2\mathbf{P}_{r_2}}{dt}\right\}_{\mathcal{F}_{r_2}} = 0$ and $\left\{\frac{d\mathbf{P}_{r_1}\mathbf{P}_1}{dt}\right\}_{\mathcal{F}_{r_1}} = 0$, we finally obtain

$$\left\{\frac{d\mathbf{P}_2\mathbf{P}_1}{dt}\right\}_{\mathcal{F}_2} = \boldsymbol{\nu}_{r_1} - \boldsymbol{\nu}_{r_2} + \boldsymbol{\omega}_{r_2} \times \mathbf{P}_2\mathbf{P}_{r_2} - \boldsymbol{\omega}_{r_1} \times \mathbf{P}_1\mathbf{P}_{r_1} - \boldsymbol{\omega}_2 \times \mathbf{P}_2\mathbf{P}_1 \tag{2.69}$$

If the robot remains horizontal (see assumption **A.1.3** in the Introduction), $\mathcal{F}_2$ and $\mathcal{F}_{r_2}$ have the same orientation, and $\boldsymbol{\omega}_2 = \boldsymbol{\omega}_{r_2}$.

Otherwise, if there are slight pitch and roll movements, $\boldsymbol{\omega}_2$ can be approximated with $\boldsymbol{\omega}_2 = (\boldsymbol{\omega}_{r_2}.\mathbf{k}_W)\mathbf{k}_W$, with $\mathbf{k}_W$ the unit vector along the vertical axis of the world frame. This is valid as long as pitch and roll angles remain small enough to make a first order approximation with respect to the horizontal orientation of the robot.

$$\boxed{\left\{\frac{d\mathbf{P}_2\mathbf{P}_1}{dt}\right\}_{\mathcal{F}_2} = \boldsymbol{\nu}_{r_1} - \boldsymbol{\nu}_{r_2} + \boldsymbol{\omega}_1 \times \mathbf{P}_{r_1}\mathbf{P}_1 - \boldsymbol{\omega}_{r_2} \times \mathbf{P}_{r_2}\mathbf{P}_2 - (\boldsymbol{\omega}_{r_2}.\mathbf{k}_W)\mathbf{k}_W \times \mathbf{P}_2\mathbf{P}_1} \tag{2.70}$$

If the leader robot is considered motionless,

$$\boxed{\left\{\frac{d\mathbf{P}_2\mathbf{P}_1}{dt}\right\}_{\mathcal{F}_2} = -\boldsymbol{\nu}_{r_2} - \boldsymbol{\omega}_{r_2} \times \mathbf{P}_{r_2}\mathbf{P}_2 - (\boldsymbol{\omega}_{r_2}.\mathbf{k}_W)\mathbf{k}_W \times \mathbf{P}_2\mathbf{P}_1} \tag{2.71}$$

The inertial measurement unit (IMU) of the robot can give the quaternion of orientation of the robot and therefore the world coordinates of the robot axes, hence ${}^W\mathbf{R}_{r_2}$. Then, the orientation of $\mathcal{F}_2$ in $\mathcal{F}_W$ can be calculated thanks to equations (2.57) to (2.59), and therefore the rotation matrix between $\mathcal{F}_2$ and $\mathcal{F}_{r_2}$, noted ${}^2\mathbf{R}_{r_2}$.

We assume we also have ${}^W\mathbf{R}_{r_1}$ thanks to the IMU of the leader robot, which allows to get ${}^1\mathbf{R}_{r_1}$, thanks to equations (2.57) to (2.59). The relative orientation between the robots is given by ${}^{r_2}\mathbf{R}_{r_1} = {}^W\mathbf{R}_{r_2}^T\,{}^W\mathbf{R}_{r_1}$.

If pitch and roll angles are considered as zero (see assumption **A.1.3** in the Introduction),

$${}^2\mathbf{R}_{r_2} = \mathbf{I} \tag{2.72}$$

$${}^1\mathbf{R}_{r_1} = \mathbf{I} \tag{2.73}$$

$${}^{r_2}\mathbf{R}_{r_1} = \begin{bmatrix} -\cos(\alpha_2 - \alpha_1) & \sin(\alpha_2 - \alpha_1) & 0 \\ -\sin(\alpha_2 - \alpha_1) & -\cos(\alpha_2 - \alpha_1) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2.74}$$

Then,

$${}^2\boldsymbol{\nu}_2 = {}^2\mathbf{R}_{r_2}\,{}^{r_2}\boldsymbol{\nu}_2 \tag{2.75}$$

$${}^2\boldsymbol{\nu}_1 = {}^2\mathbf{R}_{r_2}\,{}^{r_2}\mathbf{R}_{r_1}\,{}^{r_1}\boldsymbol{\nu}_1 \tag{2.76}$$

$${}^2\boldsymbol{\omega}_2 = {}^2\mathbf{R}_{r_2}\,{}^{r_2}\boldsymbol{\omega}_2 \tag{2.77}$$

$${}^2\boldsymbol{\omega}_1 = {}^2\mathbf{R}_{r_2}\,{}^{r_1}\boldsymbol{\omega}_1 \tag{2.78}$$

$${}^2\left\{\mathbf{P}_{r_2}\mathbf{P}_2\right\} = {}^2\mathbf{R}_{r_2}\,{}^{r_2}\overline{\mathbf{P}}_2 \tag{2.79}$$

$${}^2\left\{\mathbf{P}_{r_1}\mathbf{P}_1\right\} = {}^2\mathbf{R}_{r_2}\,{}^{r_2}\mathbf{R}_{r_1}\,{}^{r_1}\overline{\mathbf{P}}_1 \tag{2.80}$$

$${}^2\left\{\mathbf{P}_2\mathbf{P}_1\right\} = {}^2\overline{\mathbf{P}}_1 \tag{2.81}$$

$${}^2\left\{\mathbf{P}_{r_2}\mathbf{P}_1\right\} = {}^2\left\{\mathbf{P}_{r_2}\mathbf{P}_2\right\} + {}^2\left\{\mathbf{P}_2\mathbf{P}_1\right\} \tag{2.82}$$

$$= {}^2\mathbf{R}_{r_2}\,{}^{r_2}\overline{\mathbf{P}}_2 + {}^2\overline{\mathbf{P}}_1 \tag{2.83}$$

We note ${}^2\dot{\overline{\mathbf{P}}}_1$ the velocity of $\mathbf{P}_1$ with respect to frame $\mathcal{F}_2$ expressed in $\mathcal{F}_2$:

$${}^2\dot{\overline{\mathbf{P}}}_1 = {}^2\left\{\frac{d\mathbf{P}_2\mathbf{P}_1}{dt}\right\}_{\mathcal{F}_2} \tag{2.84}$$

We also note ${}^{r_i}\mathbf{v}_{r_i} = [{}^{r_i}\boldsymbol{\nu}_{r_i}, {}^{r_i}\boldsymbol{\omega}_{r_i}]^T$, the velocity vector defined at $\mathbf{P}_{ri}$, center of the robot r_i, and expressed in frame $\mathcal{F}_{r_i}$. In order to simplify the writing, we note for now on this ego-centered velocity as:

$$ {}^{r_i}\mathbf{v} = \left[{}^{r_i}\boldsymbol{\nu}, {}^{r_i}\boldsymbol{\omega}\right]^T \tag{2.85} $$

We can rewrite Eq. (2.70) in matrix form:

$$ \begin{aligned} {}^2\dot{\overline{\mathbf{P}}}_1 = &\left[{}^2\mathbf{R}_{r_2}{}^{r_2}\mathbf{R}_{r_1}, \quad -\left[{}^2\mathbf{R}_{r_2}{}^{r_2}\mathbf{R}_{r_1}{}^{r_1}\overline{\mathbf{P}}_1\right]^\times {}^2\mathbf{R}_{r_2}{}^{r_2}\mathbf{R}_{r_1}\right]{}^{r_1}\mathbf{v} \\ &- \begin{bmatrix} {}^2\mathbf{R}_{r_2}, & -\left[{}^2\mathbf{R}_{r_2}{}^{r_2}\overline{\mathbf{P}}_2\right]^\times {}^2\mathbf{R}_{r_2} - \left[{}^2\overline{\mathbf{P}}_1\right]^\times \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} {}^2\mathbf{R}_{r_2} \end{bmatrix} {}^{r_2}\mathbf{v}, \end{aligned} \tag{2.86} $$

where $[\mathbf{X}]^\times$ is the skew matrix defined from vector $\mathbf{X}$.

We note $\mathbf{T}_{r_1,2}$ and $\mathbf{T}_{r_2,2}$ and rewrite equation (2.86) such that:

$$ \boxed{{}^2\dot{\overline{\mathbf{P}}}_1 = \mathbf{T}_{r_1,2}{}^{r_1}\mathbf{v} + \mathbf{T}_{r_2,2}{}^{r_2}\mathbf{v}.} \tag{2.87} $$

Symmetrically, we can write

$$ \boxed{{}^1\dot{\overline{\mathbf{P}}}_2 = \mathbf{T}_{r_1,1}{}^{r_1}\mathbf{v} + \mathbf{T}_{r_2,1}{}^{r_2}\mathbf{v}.} \tag{2.88} $$

Equations (2.87) and (2.88) link the kinematics of the tether attachment points $\mathbf{P}_i$ with respect to the kinematics of both robots. These kinematic equations will be used for the control scheme described in Chapter 4.

If we assume that the robots remain horizontal (${}^{r_2}\mathbf{R}_2 = \mathbf{I}$), we can write:

$$ \begin{aligned} \mathbf{T}_{r_2,2} &= -\left[\mathbf{I}_{3\times 3}, \quad -\left\{\left[{}^{r_2}\overline{\mathbf{P}}_2\right]^\times - \left[{}^2\overline{\mathbf{P}}_1\right]^\times\right\}\right] = -\left[\mathbf{I}_{3\times 3}, \quad -\left[{}^{r_2}\overline{\mathbf{P}}_1\right]^\times\right] \\ \mathbf{T}_{r_1,2} &= \left[{}^{r_2}\mathbf{R}_{r_1}, \quad -\left[{}^{r_2}\mathbf{R}_{r_1}{}^{r_1}\overline{\mathbf{P}}_1\right]^\times {}^{r_2}\mathbf{R}_{r_1}\right] \end{aligned} \tag{2.89} $$

Otherwise we can extract the robot attitude angle from the IMU sensors. The frames $\mathcal{F}_{r_2}$ and $\mathcal{F}_2$ have the same azimuthal orientation. Therefore, the rotation matrix ${}^{r_2}\mathbf{R}_2$ is a function of only roll (ϕ) and pitch (θ) angles of robot r_2, as follows:

$$ {}^{r_2}\mathbf{R}_2 = \begin{bmatrix} \cos\theta & \sin\theta\sin\phi & \sin\theta\cos\phi \\ 0 & \cos\phi & -\sin\phi \\ -\sin\theta & \cos\theta\sin\phi & \cos\theta\cos\phi \end{bmatrix}. \tag{2.90} $$

In addition, $\mathbf{T}_{r_1,2}$ can be expressed as a function of $\mathbf{T}_{r_2,2}$. Indeed we have:

$$ \mathbf{T}_{r_1,2} = -\mathbf{T}_{r_2,2}{}^{r_2}\mathbf{V}_{r_1} \tag{2.91} $$

where

$$ {}^{r_2}\mathbf{V}_{r_1} = \begin{bmatrix} {}^{r_2}\mathbf{R}_{r_1}, & \left[{}^{r_2}\overline{\mathbf{P}}_{r_1}\right]^\times {}^{r_2}\mathbf{R}_{r_1} \\ \mathbf{0} & {}^{r_2}\mathbf{R}_{r_1} \end{bmatrix} \tag{2.92} $$

is the velocity twist matrix that allows to get the velocity of the center of another coordinate frame ($\mathcal{F}_{r_1}$), while making the frame transformation from $\mathcal{F}_{r_1}$ to $\mathcal{F}_{r_2}$ (see appendix C). We need the coordinates of the previous frame center, $\overline{\mathbf{P}}_{r_1}$, expressed in the new frame ($\mathcal{F}_{r_2}$).

Using the twist matrix, we can rewrite the equation (2.87) as:

$$\boxed{{}^{2}\dot{\overline{\mathbf{P}}}_1 = \mathbf{T}_{r_2,2}\left(-{}^{r_2}\mathbf{V}_{r_1}\,{}^{r_1}\mathbf{v} + {}^{r_2}\mathbf{v}\right).} \tag{2.93}$$

Finally, if we assume that the robot is horizontal and that the leader robot is stationary (${}^{r_1}\mathbf{v} = \mathbf{0}$), the equation that links the kinematics of attachment point $\mathbf{P}_1$ to the kinematics of the follower robot is reduced to:

$${}^{2}\dot{\overline{\mathbf{P}}}_1 = \mathbf{T}_{r_2,2}\,{}^{r_2}\mathbf{v} = \begin{bmatrix}-\mathbf{I}_{3\times3}, & [{}^{r_2}\overline{\mathbf{P}}_1]^{\times}\end{bmatrix}\begin{bmatrix}{}^{r_2}\boldsymbol{\nu}\\ {}^{r_2}\boldsymbol{\omega}\end{bmatrix} \tag{2.94}$$

$$\boxed{{}^{2}\dot{\overline{\mathbf{P}}}_1 = -{}^{r_2}\boldsymbol{\nu} - {}^{r_2}\boldsymbol{\omega} \times {}^{r_2}\overline{\mathbf{P}}_1} \tag{2.95}$$

2.4 Conclusions

In this Chapter we focused on the model description of the first pair of robots of the chain. The pair is connected by a tether segment and it is composed of a leader robot, that should freely explore its surroundings, and a follower, that is expected to control the tether shape in order not ot hamper the leader movements. Two different types of robots are used in this work: one with vectored thrusters configuration and another with direct configuration. Both types of robots are described, a kinematic model is developed and details about their architecture, embedded sensors and thrusters allocation are given. A model of the tether linking the leader and the follower robots is also introduced. The tether is supposed to be slightly negative buoyant, perfectly flexible and inextensible, being, therefore, modeled by a catenary cable. The catenary model of the tether is developed in order to express the tether equations in the follower robot attachment frame. The complete system composed of both robots (leader and follower) and the linking tether is also modeled.

The next Chapter addresses the strategies of perception of the tether, explaining how it is detected in the image and how its shape is estimated from the camera feedback.

Chapter 3

Underwater Perception of the Tether

A camera is a sensor commonly present in mini-ROVs since it can provide a great amount of comprehensive and detailed information about the environment to the human operator. Moreover, cameras are light, compact and have low power consumption, which is perfectly suitable to be embedded to small robots. We use the camera as the main sensor for tether perception inside the chain of robots.

In this Chapter we develop the method used to detect the tether and to estimate its shape parameters from visual feedback provided by embedded cameras. First, we introduce some camera and tether positioning assumptions on which our algorithms of tether detection and shape estimation are based. Then, the camera model is presented, followed by the calculation of the catenary projection equation in the image plane. This equation, that contains the catenary shape parameters, is used for the curve fitting procedure that estimates the tether 3-D shape. Finally, simulation results are presented.

3.1 Camera Configurations and Assumptions

An illustration of the first pair of the chain of robots is presented in Figure 3.1, which is actually the same as Figure 2.10 with two cameras added that film the tether extremities. Our objective is to detect the tether and estimate its shape by analyzing the portion of tether that is visible by the cameras.

As presented in Section 2.3, the robots are named r_i, with i being the robot index inside the chain. Since we focus on the first pair of robots we will use the simplified nomenclature of the cameras. The cameras are named c_i, with i being the index of the robot where the camera is mounted.

To implement our vision-based procedure of tether shape estimation, some assumptions are made with respect to the camera position and tether properties. They are listed here below:

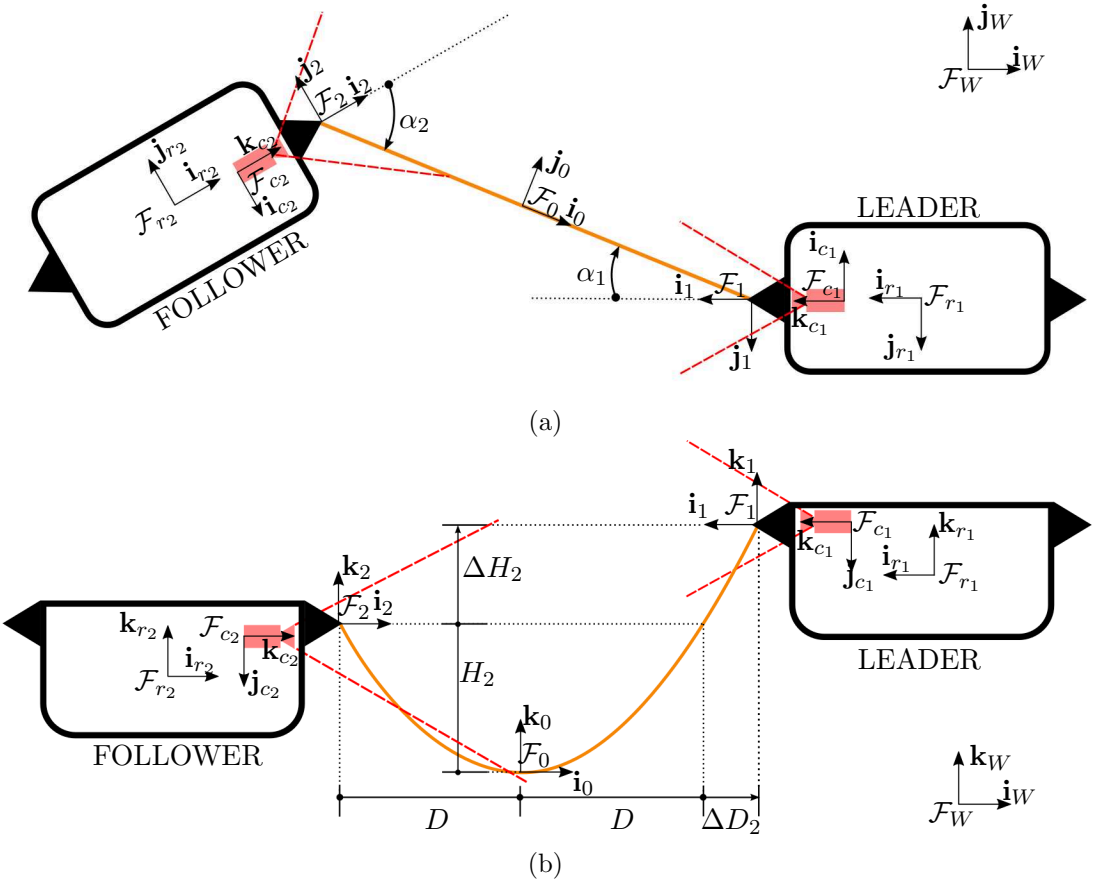


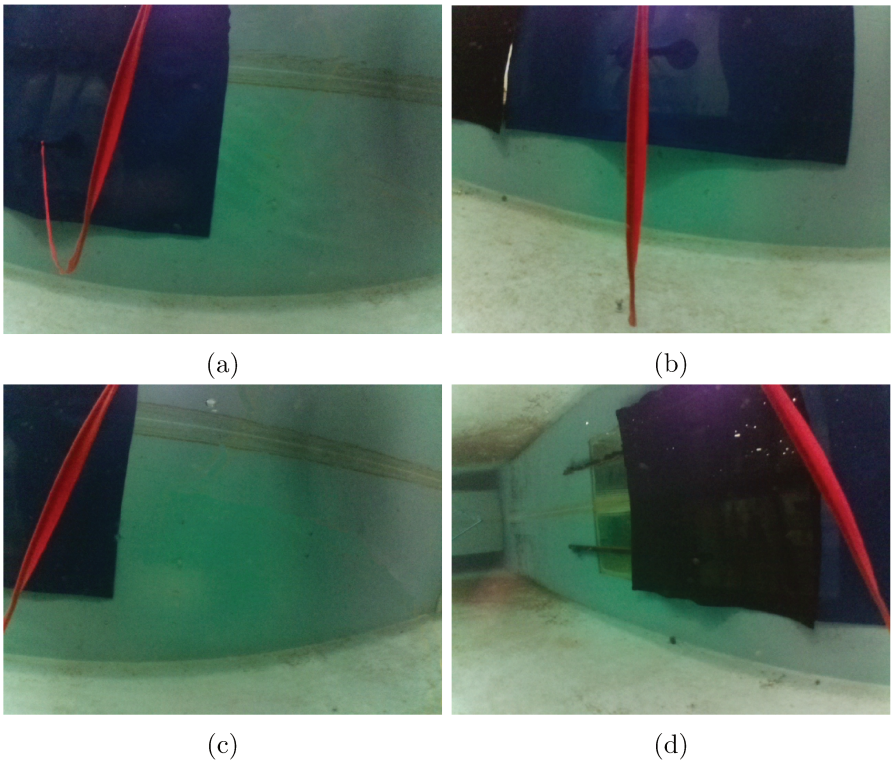
Figure 3.1: Illustration of a pair of tethered robots where the field of view of the cameras is highlighted by red dashed lines. (a) top view and (b) side view.

A.3.1 The cameras have a fixed and known pose with respect to the robot frame, which can be expressed as the following homogeneous matrix: ${}^{r_i}\mathbf{M}_{c_i}$. We assume that the camera optical axis ($\mathbf{k}_{c_i}$) is aligned with the robot longitudinal axis ($\mathbf{i}_{r_i}$). The optical axis points forwards for frontal camera and backwards for rear camera. The camera vertical axis ($\mathbf{j}_{c_i}$) points downwards and it is aligned with the robot vertical axis ($\mathbf{k}_{r_i}$). In other words, we have: $\mathbf{k}_{c_i} \parallel \mathbf{i}_{r_i}$ and $\mathbf{j}_{c_i} \parallel \mathbf{k}_{r_i}$;

A.3.2 The tether attachment points have a fixed and known positions with respect to the robots. This means that ${}^{r2}\mathbf{P}_2$ and ${}^{r1}\mathbf{P}_1$ are constant and known. The tether attachment point is above the camera in the robot frame. For a robot r_i and attachment point ${}^{r_i}\mathbf{P}_i$ we have: ${}^{r_i}Z_i > {}^{r_i}Z_{c_i}$;

A.3.3 The difference of height between the tether attachment points can be estimated through pressure sensors embedded on the robots. ΔH_i is thus directly measured.

A.3.4 The tether has a known length;



(a) (b)

(c) (d)

Figure 3.2: Some examples of the mock tether images captured by the BlueROV1 camera in different configurations. (a) the tether is completely visible. (b) the tether plane is aligned with the camera optical axis. (c) and (d) the tether is partially visible. The mock tether is made of a leaded curtain wire wrapped by an orange cotton ribbon.

A.3.5 The tether has a known and uniform color;

A.3.6 The tether portion near the camera is detectable up to a few meters away.

Assumptions **A.3.1** and **A.3.2** imply that the relative position between the camera and the tether attachment point is known. Considering also the tether negative buoyancy and that the robots remain horizontal (respectively assumptions 9 and 3 in Chapter 1), the tether always enters on the camera field from the top.

Figure 3.2 shows images taken with the BlueROV1 in the pool that gives some example images of a mock tether. In these images, the BlueROV1 simulates the follower robot. One extremity of mock tether is attached to the robot front and the other is attached to the pool wall. The mock tether is 1.50 meter long and about 0.10 meter thick. It is made of a leaded curtain wire wrapped by an orange cotton ribbon. The tether attachment point position in the robot frame is ${}^{r}\overline{\mathbf{P}}_2 = [0.23, 0.0, 0.065]^T$ meters and the camera position is ${}^{r}\overline{\mathbf{P}}_{c_2} = [0.17, 0.0, 0.0]^T$ meters (see Section 2.2 for details on BlueROV1 model).

Considering assumption A.3.3, there are only two parameters left to be estimated through camera feedback: the tether slacknesses H_i and the orientation angles α_i.

3.2 Camera Modeling

In this section we will introduce the pinhole camera model, which is a standard camera model used in computer vision applications.

3.2.1 Image Formation

A camera is a device composed of a set of lenses used to direct the light on a sensitive surface in order to form an image of the observed scene. Digital cameras are equipped with a light sensitive surface composed of metal-oxide-semiconductors (MOS) capacitors that allows the conversion of incoming photons into electron charges that are read by a charge-coupled device (CCD) to form the image. In order to have a quality image, some camera parameters related to the photometry must be set. Among these settings, we will highlight, in the next subsection, the adjustment of the camera exposure and the white balance, which are the precondition for image sharpness and color fidelity.

An image is a two-dimensional brightness array. Mathematically, an image is a function I, defined on a compact region S of a two-dimensional surface, taking values in the positive real numbers(Ma et al., 2012):

$$I : S \in \mathbb{R}^2 \to \mathbb{R}^+; (x, y) \mapsto I(x, y), \tag{3.1}$$

for a single channel or grayscale image. Images capture two kinds of information: photometric, which are related to intensity and color of images, and geometric, which are related to positions, lines and curves of the objects in the scene.

If the image is colored, it is actually composed of three arrays also called channels (Ma et al., 2012). The RGB (red, green, blue) model is a widely used color model in which these basic colors are added to form a colored image. An alternative representation is the HSV (hue, saturation, value) model, which is widely used in computer vision algorithms to perform color detection of objects since the color can be directly filtered by the hue channel. In this thesis we will consider the hue channel to segment the tether.

3.2.2 Camera Exposure and White Balance

Camera exposure determines the amount of light entering in the device, that is to say, how dark or light the image will appear when it is captured by the camera. The camera exposure is defined by three camera parameters that compose the exposure triangle: the aperture, the ISO setting and the shutter speed. Each parameter controls the exposure differently.

The aperture controls the area over which light can enter in the camera and it is usually specified in terms of a f-number N, which is given by the ratio of focal length

to effective aperture diameter.

$$N = \frac{f}{d}$$

where f is the focal length and d is the diameter of the entrance pupil (effective aperture).

The shutter speed (or exposure time) determines when the camera sensor will be open or closed to incoming light from the camera lens. It is measured in seconds and range from 1/400 to 30 seconds.

The ISO setting determines how sensitive the camera's sensor is to incoming light, which is measured on a numerical scale defined by the ISO standard.

Many combinations of the above three settings can be used to achieve the same exposure. The key, however, is knowing which trade-offs to make, since each setting also influences other image properties. For example, shutter speed affects motion blur, ISO setting affects image noise and aperture affects depth of field (the range of distance that objects in the image appear acceptably sharp). Most digital cameras have standardized exposure modes that automatically control the aperture, ISO setting and shutter speed. For robotic applications, this means that the camera does not have a constant configuration and its exposure can change during the operation. Such behavior is mostly undesirable. Therefore, in this thesis, the camera exposure is manually set by the operator.

White balance (WB) is the process of removing unrealistic color casts, so that objects which appear white in the scene are rendered white in the photo. The human eyes are very good at judging what is white under different light sources, but digital cameras often have great difficulty with automatic white balance (AWB) and can create unsightly blue, orange, or even green color casts. Such changes can cause disturbances in the execution of robotic tasks that uses image feedback for the detection or recognition of objects. Many algorithms relay on the objects hue information to perform detection, classification and recognition in the computer vision. Therefore, an incorrect adjustment of white balance or an AWB that incorrectly compensate to change of lighting can degrade the performance of vision-based detection algorithms, once the hue of objects will change in the image. In some robotic tasks, AWB is avoided and, in this thesis, the white balance of the camera is manually set by the human operator.

The camera exposure and the white balance determine the brightness, sharpness and colors (hue) of objects in the image. These parameters have to be set carefully to obtain an accurate and robust color detection of the tether.

In the next subsection we will address the geometric aspects of image formation in order to obtain the relationship between the position of objects in the scene and its correspondence in the image. In computer vision, the most used geometric camera model is the pinhole model, since it allows a simple and linear modeling of image formation in homogeneous coordinates.

3.2.3 The Pinhole Camera Model

The pinhole model considers that the reflection and diffraction effects can be neglected during light propagation in the camera's lens system, and only light refraction is considered (Ma et al., 2012). The model also assumes that the light enters in the camera through a minimal aperture so that all rays are undeflected from the emitting source towards the optical center. Moreover, the intensity measured at a pixel is identical to the amount of energy radiated at the corresponding point in space, i.e., a Lambertian surface (Ma et al., 2012). Under these conditions, the image formation process can be reduced to an ideal perspective projection, which is a good geometric approximation of a well-focused system.

The pinhole model (Figure 3.3) is composed of a frame $\mathcal{F}_c(\mathbf{P}_c, \mathbf{i}_c, \mathbf{j}_c, \mathbf{k}_c)$, whose origin $\mathbf{P}_c$ is the center of projection. The projection plane or image plane π_i is parallel to the plane $(\mathbf{i}_c, \mathbf{j}_c)$ and it is placed at a distance f (the focal length) from the origin. The axis $\mathbf{k}_c$ is called the optical axis and its intersection with the image plane is called the principal point $\mathbf{p}_c$.

A point a in the space is defined in the camera frame $\mathcal{F}_c$ by the vector ${}^c\mathbf{P}_a = [{}^cX_a \ {}^cY_a \ {}^cZ_a \ 1]^T$ in metric coordinates. It is projected on the image plane π_i as a point with coordinates $\mathbf{p}_a = [x_a, y_a, 1]^T$, that are calculated using the Thales theorem:

$$\mathbf{p}_a = \frac{f}{{}^cZ_a} \begin{bmatrix} {}^cX_a \\ {}^cY_a \\ {}^cZ_a \end{bmatrix}, \tag{3.2}$$

where ${}^cZ_a \neq 0$. This equation can be rewritten in matrix form:

$$\mathbf{p}_a = \frac{1}{{}^cZ_a} \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} {}^c\mathbf{P}_a. \tag{3.3}$$

If the focal length is known, it can be normalized to 1 and the equation (3.3) becomes:

$$\mathbf{p}_a = \frac{1}{{}^cZ_a} \mathbf{A}\, {}^c\mathbf{P}_a, \tag{3.4}$$

where

$$\mathbf{A} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \tag{3.5}$$

is called the standard (or canonical) projection matrix.

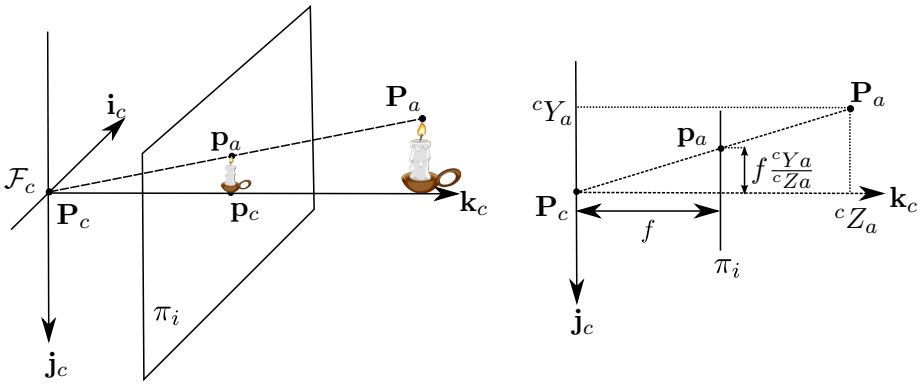


Figure 3.3: Frontal pinhole imaging model: the perspective projection of a 3-D point $\mathbf{P}_a$ on the image plane π_i is the point $\mathbf{p}_a$. The camera center is noted $\mathbf{P}_c$.

3.2.4 Intrinsic Parameters

In digital cameras, the measurements are made in terms of pixels (u, v) with respect to a frame $\mathcal{F}_{\pi_i}(\mathbf{P}_{\pi_i}, \mathbf{i}_{\pi_i}, \mathbf{j}_{\pi_i}, \mathbf{k}_{\pi_i})$ whose origin $\mathbf{P}_{\pi_i}$ is in the image upper-left corner (see Figure 3.4). Therefore, the transformation from the normalized metric to pixel coordinates of $\mathbf{p}_a$ takes into account the change of units followed by a translation of origin:

$$\begin{bmatrix} u_a \\ v_a \\ 1 \end{bmatrix} = \begin{bmatrix} \frac{1}{l_x} & 0 & u_0 \\ 0 & \frac{1}{l_y} & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_a \\ y_a \\ 1 \end{bmatrix} \tag{3.6}$$

where l_x and l_y are the pixel size and (u_0, v_0) are the pixel coordinates of the principal point in frame $\mathcal{F}_{\pi_i}$.

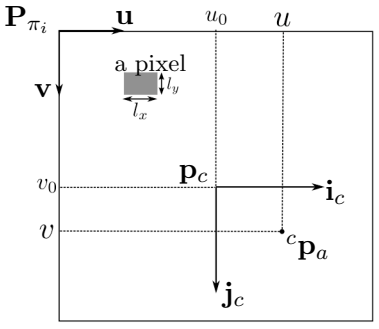


Figure 3.4: Transformation from pixels to meters coordinates.

Now, combining the perspective projection model (equation (3.3)) with the above equation yields a realistic model of transformation between the homogeneous coordinates of 3-D point and its corresponding pixel coordinates in the image plane:

$$\begin{bmatrix} u_a \\ v_a \\ 1 \end{bmatrix} = \frac{1}{{}^cZ_a} \begin{bmatrix} \frac{1}{l_x} & 0 & u_0 \\ 0 & \frac{1}{l_y} & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} {}^cX_a \\ {}^cY_a \\ {}^cZ_a \\ 1 \end{bmatrix}. \tag{3.7}$$

This equation is frequently rewritten placing all the camera intrinsic parameters in only one matrix $\mathbf{K}$ called the intrinsic parameter matrix or the calibration matrix of the camera. This amounts to considering a camera with a focal length normalized to 1, whose pixel size is $\left(p_x = \frac{f}{l_x}\right) \times \left(p_y = \frac{f}{l_y}\right)$:

$$\begin{bmatrix} u_a \\ v_a \\ 1 \end{bmatrix} = \frac{1}{{}^cZ_a} \mathbf{KA} \begin{bmatrix} {}^cX_a \\ {}^cY_a \\ {}^cZ_a \\ 1 \end{bmatrix} \tag{3.8}$$

where

$$\mathbf{K} = \begin{bmatrix} p_x & 0 & u_0 \\ 0 & p_y & v_0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{3.9}$$

The conversion between the 3-D point coordinates and its corresponding pixel coordinates is an important step to achieve the estimation of 3D models from monocular camera images. Firstly, this conversion is made by projecting the 3D points of the scene in the image plane. We obtain the ideal metric coordinates of the projected point. Secondly, the conversion between metric and pixel coordinates is made by the intrinsic parameter matrix, which is estimated through a calibration procedure. In our case, the camera calibration procedure is based on a Levenberg-Marquardt optimization algorithm that minimizes the reprojection error (Tsai, 1987).

3.3 Tether Detection

Underwater images are mostly blue due to color absorption. The longest wavelengths with lowest energy are absorbed first: red, then orange, yellow, green and finally blue (see Figure 3.5).

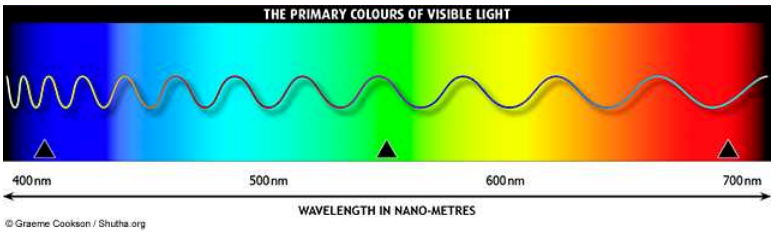


Figure 3.5: Visible light spectrum for varying wavelengths. Extracted from shutha.org.

In order to ease the tether detection in the image we select the color of the mock tether so that it is far enough from the main blue color in the hue representation (see figure 3.6). We chose orange which is expected not to be present in a natural scene. The tether is then detected in each image thanks to a simple segmentation method in the HSV space. We chose detection rather than tracking to increase the algorithm robustness.

We use the assumptions **A.3.1** and **A.3.2** about the tether attachment point position with respect to the camera as a heuristic to detect a starting point of the tether in the image. Then we propagate the color segmentation to the rest of the tether, starting from this seed point. Once the orange pixels belonging to the tether are detected, we use the following morphomathematical transformation to shrink the tether detection to a single line: first a closing to fill in the holes, then a skeletisation. The transformation from pixel to metric coordinates in the image plane is given in equation (3.4).

Two examples of this tether detection algorithm are presented in Figure 3.7. The source images are depicted in Figures 3.7a and 3.7d. Their respective hue histograms are shown in Figures 3.7b and 3.7e. The histograms are shifted so that the orange color is placed at the middle of the variation range. The results on tether detection are presented in Figures 3.7c and 3.7f.

The first example corresponds to an easy aerial setup with an homogeneous blue background. The second one is an underwater scene acquired in the experimental pool. The segmentation thresholds are manually selected during initial setup for each experiment. In the first experiment that is carried out in the air, we clearly observe two histogram modes. The first one corresponds to the blue background and the other one corresponds to the tether. In the second experiment, the scene is more complex and has a wider hue variation. Even mainly blue, the scene background is composed of several hues. We still observe a mode corresponding to the tether hue, yet it is less isolated. In both cases the algorithm manages to segment the tether.

Wrong tether detection may occur due to lighting changes or when the tether cannot be distinguished from background objects of similar color. A minimum number of

detected points is used to validate the tether detection.

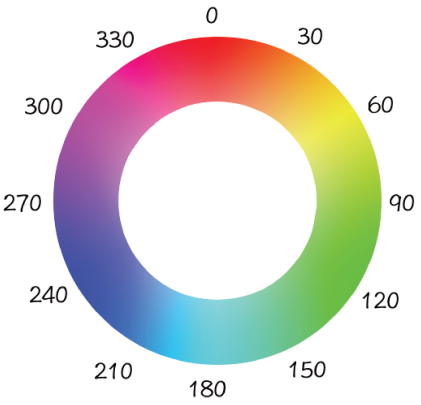


Figure 3.6: Hue representation. Courtesy of NYC Departement of Education.

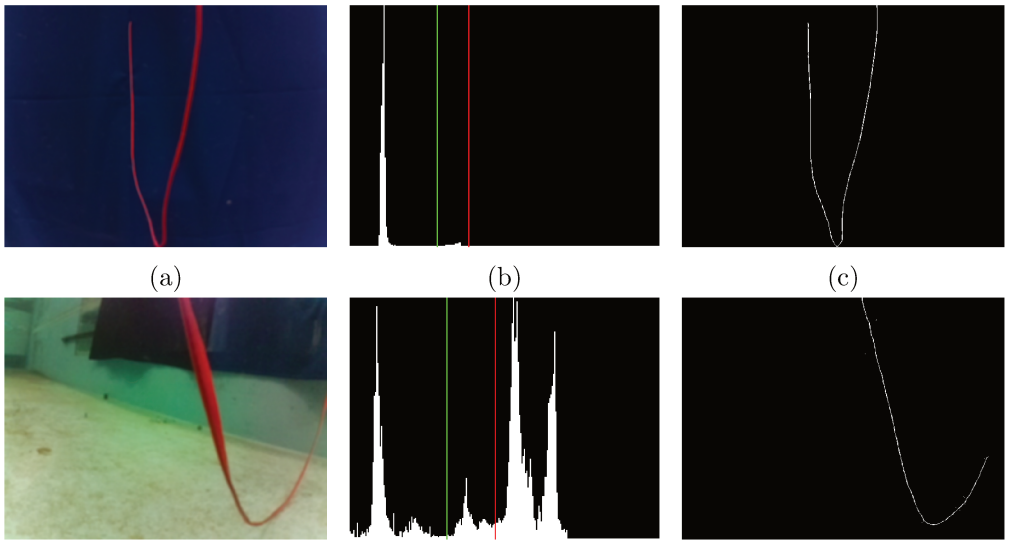


Figure 3.7: Tether detection examples in both aerial (a,b,c) and underwater (d,e,f) experiments. The mode corresponding to the specific color of the tether is selected by two thresholds in the hue histogram. The minimum and maximum thresholds are displayed in green and red lines, respectively (b,e). The detected points are refined thanks to morphomathematical closing and skeletization (c,f).

3.4 Catenary Features Estimation

Once the 2D points that belong to the tether are detected in the image, we can use them to estimate the tether features that are related to its 3D shape: H and α. First, we have to express the 3D equation of the catenary in the camera frame. Then, its perspective projection in the image plane can be expressed using the pinhole model described in Section 3.2. Finally, the 3D catenary features can be estimated by matching the 2D projection to the detected points through a curve fitting algorithm.

3.4.1 Catenary Equation in the Camera Frame

The expression of the tether catenary in frame $\mathcal{F}_2$ was given by equation (2.66), in the previous Chapter, as follows:

$$ {}^{2}\mathbf{P} = \begin{bmatrix} \cot\alpha_2 \ {}^{2}Y \\ {}^{2}Y \\ \frac{1}{C}\left[\cosh\left(C\left(\frac{{}^{2}Y}{\sin\alpha_2} - D_2\right)\right) - 1\right] - H_2 \\ 1 \end{bmatrix}. \tag{3.10} $$

The transformation between $\mathcal{F}_2$ and the robot centered frame $\mathcal{F}_{r_2}$ is then made by

$$ {}^{r2}\mathbf{P} = {}^{r_2}\mathbf{M}_2 {}^{2}\mathbf{P}, \tag{3.11} $$

where ${}^{r_2}\mathbf{M}_2$ the homogeneous matrix expressing the pose of $\mathcal{F}_2$ in $\mathcal{F}_{r_2}$:

$$ {}^{r_2}\mathbf{M}_2 = \begin{bmatrix} {}^{r_2}\mathbf{R}_2 & {}^{r_2}\overline{\mathbf{P}}_2 \\ \mathbf{0}^{T}_{3\times1} & 1 \end{bmatrix}. \tag{3.12} $$

As presented in Section 2.3.1, the frames $\mathcal{F}_{r_2}$ and $\mathcal{F}_2$ have the same azimuthal orientation. Therefore, the rotation matrix ${}^{r_2}\mathbf{R}_2$ is a function of only roll (ϕ) and pitch (θ) angles of robot r_2, as follows:

$$ {}^{r_2}\mathbf{R}_2 = \begin{bmatrix} \cos\theta & \sin\theta\sin\phi & \sin\theta\cos\phi \\ 0 & \cos\phi & -\sin\phi \\ -\sin\theta & \cos\theta\sin\phi & \cos\theta\cos\phi \end{bmatrix}. \tag{3.13} $$

If we suppose that the robot r_2 is mechanically stabilized in roll and pitch by floaters, we have that ${}^{r_2}\mathbf{R}_2 = \mathbf{I}$ and the previous homogeneous transformation becomes a simple translation:

$$ {}^{r_2}\mathbf{M}_2 = \begin{bmatrix} 1 & 0 & 0 & {}^{r_2}X_2 \\ 0 & 1 & 0 & {}^{r_2}Y_2 \\ 0 & 0 & 1 & {}^{r_2}Z_2 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{3.14} $$

The catenary equation transformation between the follower robot camera frame $\mathcal{F}_{c_2}$ and the robot frame $\mathcal{F}_{r_2}$ then made as follows:

$$ {}^{c_2}\mathbf{P} = {}^{c_2}\mathbf{M}_{r_2} {}^{r_2}\mathbf{M}_2 {}^{2}\mathbf{P}. \tag{3.15} $$

If we consider the frontal camera is fixed to the robot body, ${}^{c_2}\mathbf{M}_{r_2} = {}^{r_2}\mathbf{M}_{c_2}^{-1}$ is constant. Considering the alignment between the camera and robot frame axes given by the assumptions **A.3.1** and **A.3.2** in the previous section (and also depicted in Figure 3.1) we have that

$$ {}^{c_2}\mathbf{M}_{r_2} = \begin{bmatrix} 0 & -1 & 0 & {}^{r_2}Y_{c_2} \\ 0 & 0 & -1 & {}^{r_2}Z_{c_2} \\ 1 & 0 & 0 & -{}^{r_2}X_{c_2} \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{3.16} $$

Replacing equations (3.16) and (3.14) in (3.15) we obtain:

$$ {}^{c_2}\mathbf{P} = \begin{bmatrix} 0 & -1 & 0 & {}^{r_2}Y_{c_2} \\ 0 & 0 & -1 & {}^{r_2}Z_{c_2} \\ 1 & 0 & 0 & -{}^{r_2}X_{c_2} \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & {}^{r_2}X_2 \\ 0 & 1 & 0 & {}^{r_2}Y_2 \\ 0 & 0 & 1 & {}^{r_2}Z_2 \\ 0 & 0 & 0 & 1 \end{bmatrix} {}^{2}\mathbf{P}, \tag{3.17} $$

that yields

$$ {}^{c_2}\mathbf{P} = \begin{bmatrix} 0 & -1 & 0 & -{}^{r_2}Y_2 + {}^{r_2}Y_{c_2} \\ 0 & 0 & -1 & -{}^{r_2}Z_2 + {}^{r2}Z_{c_2} \\ 1 & 0 & 0 & {}^{r_2}X_2 - {}^{r_2}X_{c_2} \\ 0 & 0 & 0 & 1 \end{bmatrix} {}^{2}\mathbf{P}. \tag{3.18} $$

The position of the tether attachment point $\mathbf{P}_2$ in the camera frame $\mathcal{F}_{c_2}$ is given by:

$$ {}^{c_2}\mathbf{P}_2 = {}^{c_2}\mathbf{M}_{r_2}\, {}^{r_2}\mathbf{P}_2. \tag{3.19} $$

Using equation (3.16) in (3.19) we obtain

$$ {}^{c_2}\mathbf{P}_2 = \begin{bmatrix} {}^{c_2}X_2 \\ {}^{c_2}Y_2 \\ {}^{c_2}Z_2 \\ 1 \end{bmatrix} = \begin{bmatrix} -{}^{r_2}Y_2 + {}^{r_2}Y_{c_2} \\ -{}^{r_2}Z_2 + {}^{r_2}Z_{c_2} \\ {}^{r_2}X_2 - {}^{r_2}X_{c_2} \\ 1 \end{bmatrix}, \tag{3.20} $$

and replacing equation (3.20) in (3.18) yields:

$$ {}^{c_2}\mathbf{P} = \begin{bmatrix} 0 & -1 & 0 & {}^{c_2}X_2 \\ 0 & 0 & -1 & {}^{c_2}Y_2 \\ 1 & 0 & 0 & {}^{c_2}Z_2 \\ 0 & 0 & 0 & 1 \end{bmatrix} {}^{2}\mathbf{P}. \tag{3.21} $$

Finally, replacing equation (3.10) in (3.21) we obtain:

$$ {}^{c_2}\mathbf{P} = \begin{bmatrix} -{}^{2}Y + {}^{c_2}X_2 \\ -\frac{1}{C}\left[\cosh\left(C\left(\frac{{}^{2}Y}{\sin\alpha_2} - D_2\right)\right) - 1\right] + H_2 + {}^{c_2}Y_2 \\ \cot\alpha_2\, {}^{2}Y + {}^{c_2}Z_2 \\ 1 \end{bmatrix}. \tag{3.22} $$

From the first line of equation (3.22) we have the following relationship:

$$^{2}Y = -^{c_2}X + {}^{c_2}X_2. \tag{3.23}$$

Hence, we can rewrite equation (3.22) as:

$$^{c_2}\mathbf{P} = \begin{bmatrix} {}^{c_2}X \\ -\frac{1}{C}\left[\cosh\left(C\left(\frac{-{}^{c_2}X+{}^{c_2}X_2}{\sin\alpha_2} - D_2\right)\right) - 1\right] + H_2 + {}^{c_2}Y_2 \\ \cot\alpha_2\left(-{}^{c_2}X + {}^{c_2}X_2\right) + {}^{c_2}Z_2 \\ 1 \end{bmatrix}. \tag{3.24}$$

Symmetrically, the equation of the tether catenary can also be written in frames $\mathcal{F}_1$ and $\mathcal{F}_{c_1}$. This leads to an equation similar to (3.24), but with parameters H_1, ΔH_1, D_1 and ΔD_1 (see Section 2.3.2 and Figure 2.11). From now on, we develop the catenary projection equation in the camera frame $\mathcal{F}_{c_2}$. The frame $\mathcal{F}_{c_2}$ is noted $\mathcal{F}_c$, the angle α_2 is noted α, the catenary half-span and slackness are respectively noted D and H, and the tether attachment points difference of height is noted ΔH. Hence, the notation of equation (3.24) is simplified to:

$$^{c}\mathbf{P} = \begin{bmatrix} {}^{c}X \\ -\frac{1}{C}\left[\cosh\left(C\left(\frac{-{}^{c}X+{}^{c}X_2}{\sin\alpha} - D\right)\right) - 1\right] + H + {}^{c}Y_2 \\ \cot\alpha\left(-{}^{c}X + {}^{c}X_2\right) + {}^{c}Z_2 \\ 1 \end{bmatrix}. \tag{3.25}$$

3.4.2 Catenary Projection on the Image Plane

A generic 3-D point belonging to the catenary with homogeneous coordinates $^{c}\mathbf{P}$ in $\mathcal{F}_c$, is projected on the image plane as a 2-D point with coordinates $\mathbf{p} = [x \;\; y \;\; 1]^T$ using equation (3.4):

$$\mathbf{p} = \frac{1}{{}^{c}Z}\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} {}^{c}\mathbf{P} \tag{3.26}$$

that yields

$$\begin{cases} x = \dfrac{{}^{c}X}{{}^{c}Z} & \text{(3.27a)} \\ y = \dfrac{{}^{c}Y}{{}^{c}Z} & \text{(3.27b)} \end{cases}$$

From the second line of the matrix of equation (3.25), let us define that

$$\zeta = \frac{-{}^{c}X + {}^{c}X_2}{\sin\alpha}, \tag{3.28}$$

and from the third line of the same equation we have that

$$^{c}Z = \cot\alpha\left(-{}^{c}X + {}^{c}X_2\right) + {}^{c}Z_2. \tag{3.29}$$

Then, replacing equation (3.29) in (3.27a) yields:

$$^{c}X = \frac{x\,(^{c}X_2 \cot\alpha + {}^{c}Z_2)}{1 + x\cot\alpha}, \tag{3.30}$$

that replaced back in equations (3.28) and (3.29) gives, respectively:

$$\zeta = \frac{^{c}X_2 - x^{c}Z_2}{\sin\alpha + x\cos\alpha} \tag{3.31}$$

and

$$^{c}Z = \frac{\cot\alpha^{c}X_2 + {}^{c}Z_2}{1 + x\cot\alpha}. \tag{3.32}$$

Finally, from equations (3.31) and (3.32), and replacing the second line of equation (3.25) in (3.27b) we obtain the catenary projection equation on the image plane $y = f(x, H, \Delta H, \alpha)$, knowing that C and D depend on H and ΔH (see equations (2.14) and (2.17)):

$$y = \frac{1}{^{c}Z}\left[-\frac{cosh\,(C\zeta - CD) - 1}{C} + H + {}^{c}Y_2\right] \tag{3.33}$$

where ζ and ^{c}Z are defined by equations (3.31) and (3.32), respectively.

3.4.3 Catenary Curve Fitting

The catenary 3D shape is estimated through a nonlinear least-square fitting procedure based on a Guass-Newton algorithm: the catenary projection model (equation (3.33)) is fitted to the tether points detected in the camera image. The algorithm of tether detection was addressed in Section 3.3. In this current Section we focus on the fitting procedure, assuming that the tether is detectable.

The catenary shape is described by the following parameters:

$$a = \frac{H}{H_{max}} \tag{3.34a}$$

$$b = \sin\alpha \tag{3.34b}$$

$$\Delta H = {}^{2}Z_1 \tag{3.34c}$$

with H_{max} being a maximum acceptable tether slackness. We assume that the difference of height between the tether attachments points (ΔH) is estimated through the robot's external pressure sensors (see assumption **A.3.3**). Therefore, only the first two parameters of equation (3.34) are estimated through curve fitting. These parameters form a feature vector $\mathbf{s}$ defined as:

$$\mathbf{s} = \begin{bmatrix} a \\ b \end{bmatrix}. \tag{3.35}$$

The parameters a and b are respectively related to the tether slackness and orientation with respect to the follower robot. Since $0 < H < H_{max}$, the a-parameter ranges in the interval $]0, 1[$.

For the b-parameter, we will consider that it ranges in the same interval $]0,1[$. Therefore, **we always consider the case** $\alpha > 0$. When $\alpha < 0$, we will use the mirror image in order to comply with these boundaries.

The tether is completely taut when $a = 0$ and it reaches the maximum acceptable slackness H_{max} when $a = 1$. When $b = 0$ the tether belongs to the plane $\pi\left(\mathbf{P}_2, \mathbf{i}_2, \mathbf{k}_2\right)$ and when $b = 1$ it belongs to the plane $\pi\left(\mathbf{P}_2, \mathbf{j}_2, \mathbf{k}_2\right)$.

Rewriting equation (3.33) with these new parameters yields:

$$y\left(\mathbf{s}, x\right) = \frac{1}{{}^{c}Z}\left[-\frac{cosh\left(C\zeta - CD\right) - 1}{C} + aH_{max} + {}^{c}Y_2\right], \tag{3.36}$$

where

$$\zeta = \frac{{}^{c}X_2 - x\,{}^{c}Z_2}{b + x\sqrt{1-b^2}} \tag{3.37}$$

and

$${}^{c}Z = \frac{\sqrt{1-b^2}\,{}^{c}X_2 + b\,{}^{c}Z_2}{b + \sqrt{1-b^2}\,x}. \tag{3.38}$$

3.4.3.1 Gauss-Newton Algorithm

The Gauss-Newton algorithm is a modification of the Newton's method for solving nonlinear least squares problems of continuous and differentiable functions. The Gauss-Newton algorithm has the advantage of not requiring second derivatives, therefore, it is commonly used in curve fitting problems, where a sum of squared functions called residuals is minimized.

A residual can be defined as the distance between a point in the curve we want to fit and its correspondent in the curve issued from the current parameters estimation. In our case, we computed the residuals as follows:

$$r_i\left(\mathbf{s}\right) = y_i - y(\mathbf{s}, x_i) \tag{3.39}$$

where(x_i, y_i) are the metric coordinates of the i-th detected point of the tether in the image and $y(\mathbf{s}, x_i)$ is the tether model given by equation (3.36).

The sum of residuals is commonly called the cost function and our objective is to minimize it. Here below is the cost function we designed for the catenary fitting:

$$\Gamma(\mathbf{s}) = \frac{1}{M}\sum_{i=1}^{M} r_i^2(\mathbf{s}), \tag{3.40}$$

where $M \in \mathbb{N}$ is the total number of detected points.

The minimum of the cost function $\Gamma(\mathbf{s})$ is found through an iterative process that is usually initialized at a starting point in the middle of variation range of the feature vector. In our case, since $a \in]0;1[$ and $b \in]0;1[$, we use

$$\mathbf{s}^0 = [0.5, 0.5]^T \tag{3.41}$$

as starting point (in Section 3.4.3.5 a procedure of initial guess is presented). The feature vector is iterated as follows:

$$\mathbf{s}^{k+1} = \mathbf{s}^k - \lambda \mathbf{J}_r^+ \mathbf{r}\left(\mathbf{s}^k\right), \tag{3.42}$$

where $k \in \{1, ..., N\}$ is the iteration index with $N \in \mathbb{N}$, $\lambda \in \mathbb{R}^+$, $\mathbf{r}\left(\mathbf{s}^k\right)$ is a column vector that stacks the residuals $r_i\left(\mathbf{s}^k\right)$ and $\mathbf{J}_r^+$ is the Moore-Penrose pseudo-inverse of the Jacobian matrix $\mathbf{J}_r$. Since the catenary is continuous and differentiable, so is the cost function $\Gamma(\mathbf{s})$. The Jacobian matrix $\mathbf{J}_r$ of the residuals vector is thus given by

$$\mathbf{J}_r = \frac{1}{M}\frac{d\mathbf{r}}{d\mathbf{s}}, \tag{3.43}$$

whose i-th row has the following form:

$$\mathbf{J}_r\left(i\right) = \frac{1}{M}\left[-\frac{\partial y(x_i,\mathbf{s})}{\partial a} \quad -\frac{\partial y(x_i,\mathbf{s})}{\partial b}\right], \tag{3.44}$$

where

$$\begin{aligned}\frac{\partial y(x_i,\mathbf{s})}{\partial a} = \frac{1}{Z}\Bigg[&-\frac{\partial\left(\frac{1}{C}\right)}{\partial a}\left[\cosh\left(C\zeta - CD\right) - 1\right] \\ &- \frac{\sinh\left(C\zeta - CD\right)}{C}\left(A_1 + A_2 + A_3\right) + H_{max}\Bigg]\end{aligned} \tag{3.45}$$

and

$$\begin{aligned}\frac{\partial y(x_i,\mathbf{s})}{\partial b} = &-\frac{1}{C}\frac{\partial\frac{1}{Z}}{\partial b}\left[\cosh\left(C\zeta - CD\right) - 1 - C\left(H + {}^cY_2\right)\right] \\ &- \frac{1}{Z}\frac{\partial\zeta}{\partial b}\sinh\left(C\zeta - CD\right),\end{aligned} \tag{3.46}$$

with $A_1 = \zeta\frac{\partial C}{\partial a}$, $A_2 = -D\frac{\partial C}{\partial a}$ and $A_3 = -C\frac{\partial D}{\partial a}$. The calculation details of the Jacobian derivatives are presented in appendix D.

3.4.3.2 Study of the Gauss-Newton Jacobian Singularities

A singular matrix is a matrix whose determinant is zero and it is hence non-invertible. This occurs when the matrix is not full ranked or when there are elements in the matrix that tend to infinity. We will study the singularities of the Jacobian matrix defined in equation (3.43) with respect to parameters a and b.

Firstly, let us focus on the partial derivative with respect to the a-parameter, which is given by equation (3.45). We note that this equation is singular in the following case:

$$2H + \Delta H = L. \tag{3.47}$$

However, this corresponds to the degenerated case and it is excluded as defined in the conditions of existence of the catenary presented in Section 2.1.3. Therefore, such a

case will never occur, since we choose the values of H_{max}, ΔH and the range of the a-parameter that avoid it.

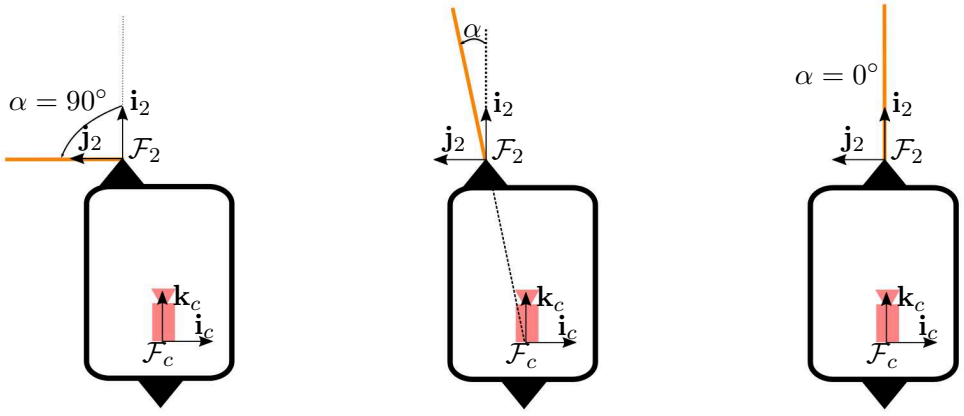
Secondly, focusing on the partial derivative with respect to the b-parameter, which is given by equation (3.46), we find the following singularities:

$$\sqrt{1-b^2} = 0 \tag{3.48}$$

$$\sqrt{1-b^2}\ {}^cX_2 + b\ {}^cZ_2 = 0 \tag{3.49}$$

$$b + \sqrt{1-b^2}x_i = 0. \tag{3.50}$$

Equation (3.48) represents the case where $b = \sin\alpha = \pm 1$, thus $\alpha = \pm 90°$ and the tether plane is perpendicular to the robot longitudinal plane, as shows Figure 3.8a. Considering the experimental setup, the cases where $\alpha \geq 90°$ or $\alpha \leq -90°$ will be avoided since the tether is less visible (limited camera field of view) and it risks of becoming entangled with the robot.



(a) (b) (c)

Figure 3.8: Top view of a robot-tether system presenting the singularities on catenary shape estimation with respect to the tether orientation parameter $b = \sin\alpha$. (a) the tether is perpendicular to the robot longitudinal axis. (b) the tether plane is aligned with the segment $\mathbf{P}_c\mathbf{P}_2$ and the catenary curve degenerates into a straight line in the image. The tether attachment point has an offset with respect to the camera axis (${}^cX_2 \neq 0$). (c) the specific case where ${}^cX_2 = 0$.

Equation (3.49) represents the case where the catenary plane is aligned with the segment linking the camera center $\mathbf{P}_c$ to the tether attachment point $\mathbf{P}_2$, as shows Figure 3.8b. The catenary shape is therefore degenerated into a straight line in the image. Indeed, we have:

$$\tan\alpha = \frac{b}{\sqrt{1-b^2}} = -\frac{{}^cX_2}{{}^cZ_2}.$$

If ${}^cX_2 = 0$, this situation occurs when the catenary plane is also aligned with the robot longitudinal axis and thus $b = 0$ (see Figure 3.8c).

Equation (3.50) leads to:

$$\frac{b}{\sqrt{1-b^2}} = -x_i$$

and from the perspective projection given by equation (3.4), we obtain:

$$\tan\alpha = -\frac{{}^cX_i}{{}^cZ_i}.$$

Regarding Figure 3.9b and from the definition of $\tan\alpha$, we obtain

$$\frac{{}^cX_i - {}^cX_2}{{}^cZ_i - {}^cZ_2} = \frac{{}^cX_i}{{}^cZ_i}.$$

which simplifies to (assuming ${}^cZ_i \neq 0$):

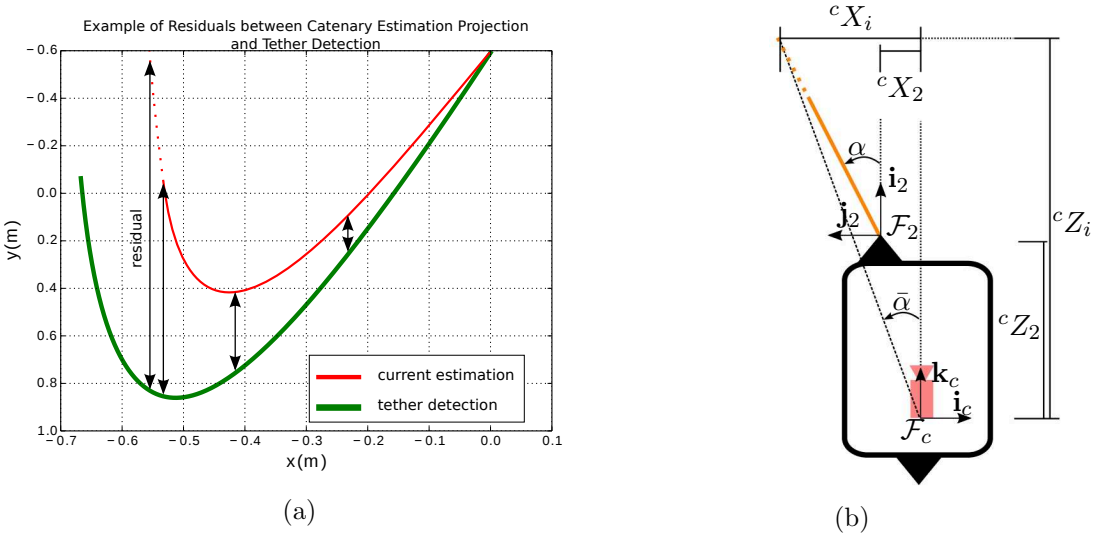
$$x_i = \frac{{}^cX_2}{{}^cZ_2} = x_2.$$

This means that the points $\mathbf{P}_i$ and $\mathbf{P}_2$ have the same abscissa in the image plane. Therefore, this occurs when the catenary projection degenerates into a vertical line in the image. This is the generalization of equation (3.49). These cases are depicted in Figures 3.8b and 3.8c.

3.4.3.3 Particular Case of Remote Points

The more remote the catenary points from the image plane, the more tightened the x-coordinates of their projection in the image plane. This is illustrated in Figure 3.9 by the left part of the curves that tend to be more and more vertical as the 3-D points get further away from the image plane. As a consequence, derivatives can take very high values, which can lead to numerical issues such as ill-conditioning of the Gauss-Newton Jacobian matrix. Another issue is occurrence of high residuals in the cost function.

In order to cope with these problems, a filter that selects only bounded residuals is used. This filter and other improvements added to the standard Gauss-Newton algorithm are described in the following Section.



(a) (b)

Figure 3.9: Case of remote points and large residuals. (a) an illustration of residuals (black arrows) between tether detected points (in green) and the projection of the catenary current estimation (in red). The tether length is $L = 1.50$m and the height difference between the attachment points is $\Delta H = 0$. The tether shape is defined by the feature vector $\mathbf{s} = [0.6, 0.6]^T$, with $H_{max} = 0.70$m. The red dashed line represents the catenary points which are beyond the tether length ($L = 1.50$m). (b) a top view illustration of catenary remote points that become more and more tight and vertical in the image plane as $\bar{\alpha} \to \alpha$.

3.4.3.4 Gauss-Newton Improved Algorithm

The general scheme of the improved Gauss-Newton algorithm proposed for the catenary fitting procedure is presented in Figure 3.10. It comprises four new functions that are described below. The tether shape estimation starts from the acquisition of a new image and the detection of the tether points. Then, the residuals are calculated between the tether detected points and the catenary projection generated by the starting feature vector $\mathbf{s}^0$. Subsequently, the residuals vector is filtered by the functions **unbounded residuals removal** (1) and **2-D points subsampling** (2). Next, a residual weighting matrix, calculated through the method of the **M-estimators** (3), is applied. Then, the feature vector is iterated ensuring that it remains inside its bounds of definition (function **bounded iteration**) (4). Finally, the stopping criteria are checked. If the algorithm converges or the maximum number of iterations is reached, the estimated feature vector is returned and a new image is acquired. The improved Gauss-Newton is also represented in pseudo-code by Algorithm 1.

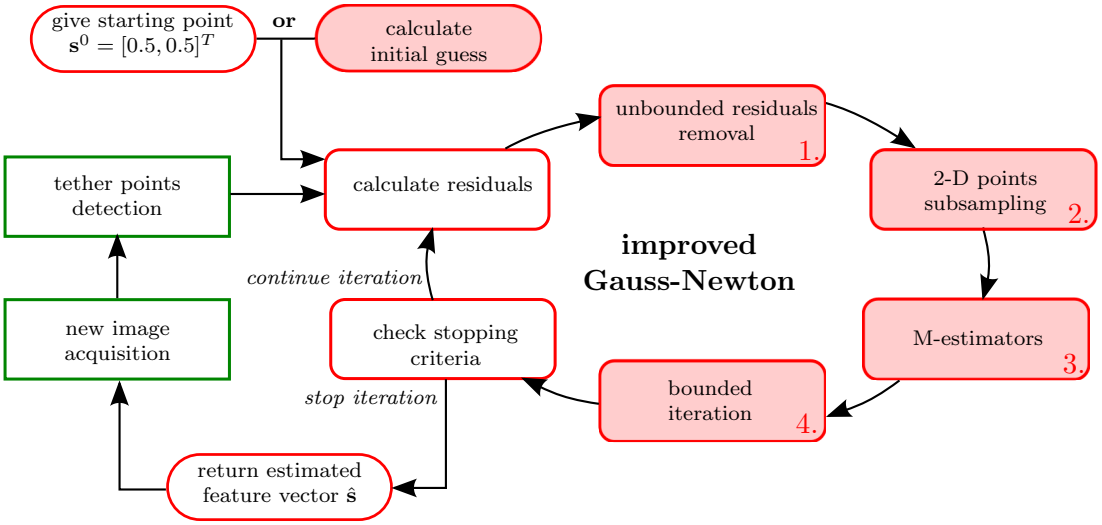


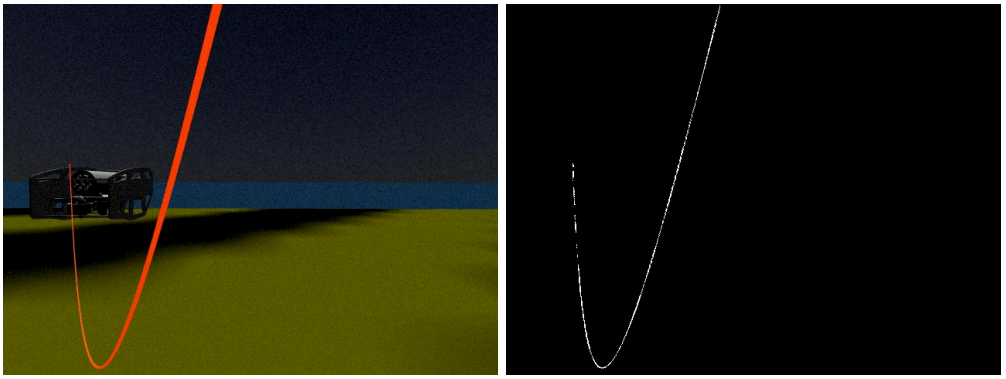
Figure 3.10: Chart flow of the improved Gauss-Newton algorithm. The four new functions added to the algorithm flow are numbered and highlighted by a pink background.

Algorithm 1 improved Gauss-Newton Algorithm

procedure GNA($\mathbf{T}_{in}, \mathbf{s}^0, k_{max}$)
  $k \leftarrow 0$
  $\mathbf{s}^k \leftarrow \mathbf{s}^0$
  while $k \leq k_{max}$ **do**
    $\mathbf{T} \leftarrow \mathbf{T}_{in}$ ▷ retrieve tether points
    $\mathbf{r} \leftarrow calculateResiduals(\mathbf{T}, \mathbf{s}^k)$
    $\mathbf{r}, \mathbf{T} \leftarrow unboundedResidualsRemoval(\mathbf{T}, \mathbf{s}^k, \mathbf{r})$
    $\mathbf{r}, \mathbf{T} \leftarrow 2DPointsSubsampling(\mathbf{T}, \mathbf{r})$
    $\mathbf{D} \leftarrow robustEstimation(\mathbf{r})$
    $\mathbf{J} \leftarrow calculateJacobian(\mathbf{T}, \mathbf{s}^k)$
    $\mathbf{s}^{k+1} \leftarrow boundedIteration(\mathbf{B}, \mathbf{J}, \mathbf{D}, \mathbf{r}, \mathbf{s}^k)$
    if $\Gamma\left(\mathbf{s}^{k+1}\right) - \Gamma\left(\mathbf{s}^k\right) < 0$ **and** $\left|\Gamma\left(\mathbf{s}^{k+1}\right) - \Gamma\left(\mathbf{s}^k\right)\right| < \epsilon$ **then**
      return $\mathbf{s}^{k+1}$ ▷ precision threshold reached
    else if $\Gamma\left(\mathbf{s}^{k+1}\right) - \Gamma\left(\mathbf{s}^k\right) > 0$ **then**
      return $\mathbf{s}^k$ ▷ cost function increased
    $\mathbf{s}^k \leftarrow \mathbf{s}^{k+1}$
  return $\mathbf{s}^{k+1}$ ▷ max. number of steps exceeded

The first two functions are illustrated with the tether detected points obtained from the simulated image shown in Figure 3.11. In this simulation the feature vector is $\mathbf{s} = [0.6, 0.6]^T$, $H_{max} = 0.70$ meter and $\Delta H = 0$ meter. All the steps of detection and fitting are run. Thus, it includes noise due to image segmentation and pixel/meter conversion.

The residuals are calculated from the difference between the current tether detected points and the estimated catenary projection, whose feature vector is $\hat{\mathbf{s}} = [0.5, 0.5]^T$. This is the starting point of the fitting iteration process (see equation (3.41)).



(a) (b)

Figure 3.11: An example of tether points detection from simulated images. The simulated tether is 1.50 meter long, 3 milimeter thick, with feature vector $\mathbf{s} = [0.6, 0.6]^T$, $H_{max} = 0.70$ meter and $\Delta H = 0$ meter. (a) the raw image obtained from follower robot camera. (b) the tether detected points obtained from image processing (described in Section 3.3) and used as input data by the Gauss-Newton algorithm.

1. Unbounded Residuals Removal The objective of this function is to remove the residuals that are related to catenary points that physically do not exist, being an extension of the tether length. Regarding Figure 3.9, for example, only the residuals between the red and green full lines should be taken into account. Therefore, the residual related to the detected point $\mathbf{p}_i = [x_i, y_i, 1]^T$ is accepted if the following conditions are satisfied.

First, from equation (3.50), we extract the following condition:

$$\widehat{b} + x_i\sqrt{1 - \widehat{b}^2} \neq 0. \tag{3.51}$$

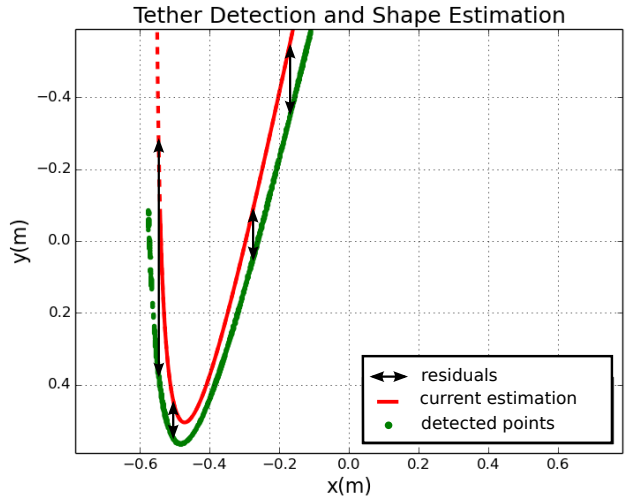
Secondly, we bound the point depth in the scene:

$$^c\widehat{Z}_i > {}^cZ_2 \tag{3.52}$$

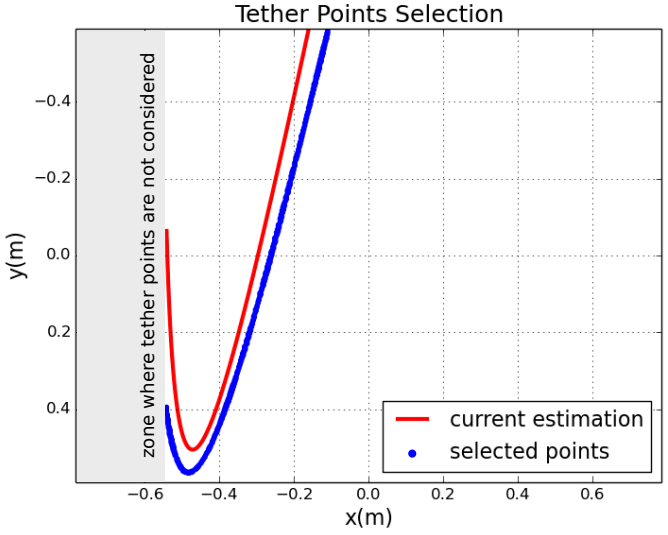
$$^c\widehat{Z}_i < L + {}^cZ_2, \tag{3.53}$$

As a reminder, L is the tether total length and the depth $^c\widehat{Z}_i$ is calculated from equation (3.38).

The unbounded residuals removal function is applied to the tether points presented in Figure 3.11b and the result is shown in Figure 3.12.



(a)



(b)

Figure 3.12: An example of **unbounded residuals removal** applied to tether points depicted in Figure 3.11b. (a) the tether points are drawn in green, an example of catenary shape estimation with $\hat{\mathbf{s}} = [0.5, 0.5]^T$ is drawn in red and some residuals are shown by black arrows. The red dashed line represents the catenary points that do not satisfy the bounding conditions and are therefore eliminated. (b) the selected points resulting from the **unbounded residuals removal** are drawn in blue.

The full red line represents the catenary points that satisfy the conditions given by equations (3.51), (3.52) and (3.53). These equations give the coordinate of the projection of the attachment points $\mathbf{P}_1$, which bounds the coordinate of the tether in the image plane. The red dashed line represents the catenary points that do not satisfy these conditions and that will be therefore filtered out. The selected points resulting from the filtering are down in blue in Figure 3.12b.

2-D Points Subsampling The residuals are not evenly distributed throughout the tether projection. As shows Figure 3.13a, the 2-D detected points are more concentrated, with respect to the abscissa axis, in zones where the tether cable is closer to the vertical. In the zone near to the tether lowest point, the points are more spaced. This has a great impact on the residual computation since we use a vertical distance. Algorithm 2 introduces a procedure allowing to obtain a more uniform distribution of detected points along the tether.

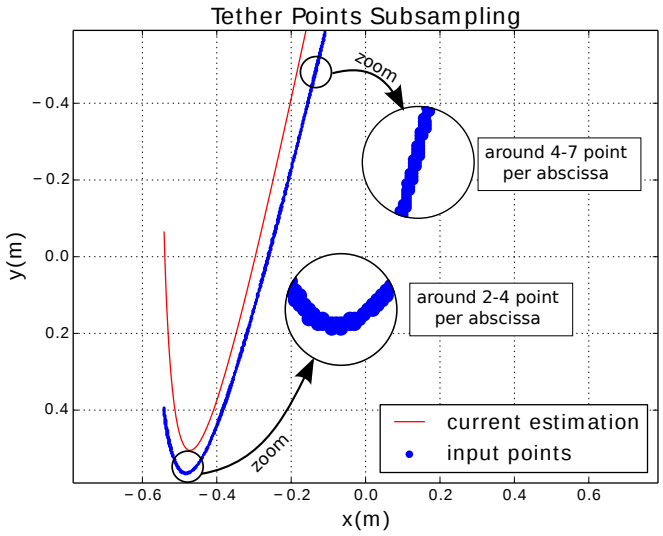
The algorithm takes as input the vector of residuals $\mathbf{r}_{in}$ and the matrix $\mathbf{T}_{in}$ containing the tether points resulting from the unbounded residuals filtering described previously. $\mathbf{T}_{in}$ is a $2 \times m$ matrix, where m is the number of input points. The algorithm returns the vector $\mathbf{r}_{out}$ containing the smallest residual for each abscissa, and the matrix $\mathbf{T}_{out}$ that contains the tether points used to calculate the residuals.

Algorithm 2 2-D points subsampling

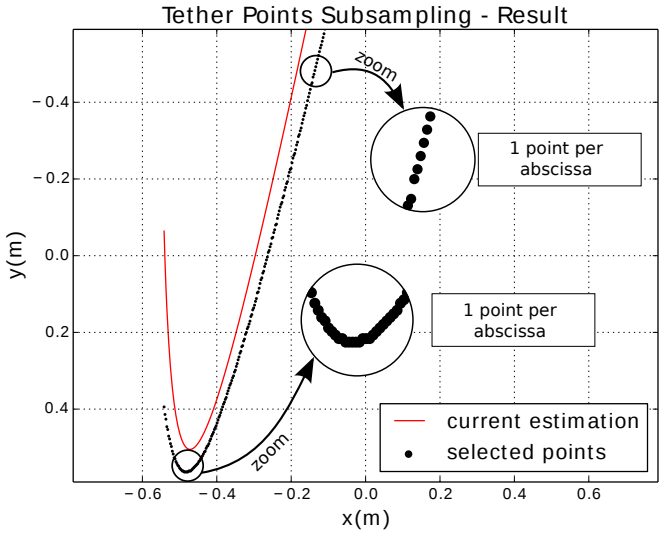
procedure 2DPS($\mathbf{r}_{in}$,$\mathbf{T}_{in}$) ▷ inputs: residuals vector, tether points
  $i \leftarrow 0$, $k \leftarrow 0$
  $m \leftarrow getColumns(\mathbf{T}_{in})$ ▷ get number of measurements
  while $i < m$ **do**
    if $|\mathbf{T}_{in}(0,i) - \mathbf{T}_{in}(0,i+1)| > \epsilon$ **then** ▷ different abscissa
      if use_{min} **then**
        $use_{min} \leftarrow false$
        $\mathbf{r}_{out}(k) \leftarrow \mathbf{r}_{min}$
        $\mathbf{T}_{out}(:,k) \leftarrow \mathbf{T}_{in}(:,i_{min})$
        $\mathbf{r}_{min} \leftarrow \infty$
        $k \leftarrow k+1$
      else
        $\mathbf{r}_{out}(k) \leftarrow \mathbf{r}_{in}(i)$
        $\mathbf{T}_{out}(:,k) \leftarrow \mathbf{T}_{in}(:,i)$
        $k \leftarrow k+1$
    else
      $use_{min} \leftarrow true$ ▷ same abscissa
      $r_i \leftarrow \mathbf{r}_{in}(i)$
      $r_{i+1} \leftarrow \mathbf{r}_{in}(i+1)$
      if $|r_i| < |r_{min}|$ **then** ▷ choose the smallest residual
        $r_{min} \leftarrow r$
        $i_{min} \leftarrow i$
      else if $|r_{i+1}| < |r_{min}|$ **then**
        $r_{min} \leftarrow r_{i+1}$
        $i_{min} \leftarrow i+1$
    $i \leftarrow i+1$
  return $\mathbf{r}_{out}, \mathbf{T}_{out}$

The resulting subsampled residuals are shown in Figure 3.13b in black dots. The number of computed residuals is reduced from 738 to 177, which also reduces the

Jacobian matrix size and allows a faster computation of the Gauss-Newton iterations.



(a)



(b)

Figure 3.13: An example of **2-D points subsampling** applied to tether points that were already filtered by the unbounded residuals removal function, as presented in Figure 3.12b. (a) the input points are drawn in blue and the current catenary shape estimation ($\hat{\mathbf{s}} = [0.5, 0.5]^T$) is drawn in red. The unevenly tether points distribution is demonstrated by two zoomed regions. (b) the selected points resulting from the 2-D points subsampling are drawn in black. Only one tether point per abscissa is selected (the one with the smallest residual).

Robust Estimation Following the residuals subsampling, we apply a procedure of residuals weighting called M-estimators (Pressigout, 2006, chap. 3). This procedure aims to make the tether shape estimation more robust to detection noise (color segmentation and pixel/meter conversion). The general idea is to give a higher weight to smaller residuals, which are associated to a more reliable measure. The new residuals vector is thus weighted by the diagonal matrix $\mathbf{D}$ as follows:

$$\mathbf{r}_M = \mathbf{D}\mathbf{r} \tag{3.54}$$

where

$$\mathbf{D} = \begin{bmatrix} w_1 & \cdots & 0 \\ & \ddots & \\ 0 & \cdots & w_M \end{bmatrix}. \tag{3.55}$$

The weights w_i, with $i \in \{1, ..., M\}$ and M being the number of residuals, range in the interval $[0, 1]$. They are calculated according to Huber (1981), as follows:

$$w_i = \frac{f_M(r_i)}{r_i}, \tag{3.56}$$

where $f_M(r_i)$ is the weighting function defined as (Marchand et al., 2005):

$$f_M(r_i, S) = \begin{cases} r_i(r_i^2 - S^2)^2 & \text{if } |r_i| < S \\ 0 & \text{else} \end{cases} \tag{3.57}$$

where $S = 4.7\hat{\sigma}$ and $\hat{\sigma} = 1.48 Med_i(|r_i - Med_i(r_i)|)$ is the median absolute deviation.

Once we added a new weight matrix $\mathbf{D}$, the feature vector iteration rule defined by equation (3.42) should be updated as follows:

$$\mathbf{s}^{k+1} = \mathbf{s}^k - \lambda \left(\mathbf{D}\mathbf{J}_r\right)^+ \mathbf{D}\mathbf{r}\left(\mathbf{s}^k\right). \tag{3.58}$$

Bounded Iteration The feature vector, as defined by equation (3.35), is bounded: $\mathbf{s} \in]0, 1[\times]0, 1[$. Feature vectors outside these bounds are associated to catenary projections that do not exist. Therefore, we have to ensure that the feature vector remains inside this bounded zone throughout the iteration process.

In order to manage the boundedness of the feature vector, the function described by Algorithm 3 is applied to each iteration step. This function takes as input the Jacobian matrix $\mathbf{J}$, the weighting matrix $\mathbf{D}$, the residual vector $\mathbf{r}$, the feature vector at the current iteration $\mathbf{s}^k$ and the matrix $\mathbf{B}$ that contains the bounds of the feature vector.

The general idea of the algorithm is to reduce the gains inside matrix $\mathbf{\Lambda}$ corresponding to features that are out of bounds. These gains are reduced until the feature return to the bounds. Matrix $\mathbf{\Lambda}$ is diagonal and it is defined as follows:

$$\mathbf{\Lambda} = \begin{bmatrix} \lambda_1 & \cdots & 0 \\ & \ddots & \\ 0 & \cdots & \lambda_N \end{bmatrix}, \tag{3.59}$$

where N is the size of the feature vector.

Algorithm 3 Bounded Iteration Algorithm

procedure BI($\mathbf{J}, \mathbf{D}, \mathbf{r}, \mathbf{s}^k, \mathbf{B}$)
  $\mathbf{s}^{k+1} = \mathbf{s}^k - \mathbf{\Lambda} \left(\mathbf{D}\mathbf{J}_r\right)^+ \mathbf{D}\mathbf{r}\left(\mathbf{s}^k\right)$
  while $isOutofBounds(\mathbf{s}^{k+1}, \mathbf{B})$ **do** ▷ if vector is out of bounds
    $i \leftarrow 0$
    while $i < size(\mathbf{s}^{k+1})$ **do** ▷ find feature that is out of bounds
      if $isOutofBounds(\mathbf{s}^{k+1}(i), \mathbf{B}(i,:))$ **then**
        $\mathbf{\Lambda}(i,i) = 0.1\mathbf{\Lambda}(i,i)$ ▷ reduce the corresponding gain
      $i \leftarrow i + 1$
    $\mathbf{s}^{k+1} = \mathbf{s}^k - \mathbf{\Lambda} \left(\mathbf{D}\mathbf{J}_r\right)^+ \mathbf{D}\mathbf{r}\left(\mathbf{s}^k\right)$ ▷ re-iterate
  return $\mathbf{s}^{k+1}$

Iteration Stopping Criteria The stopping criteria for the iteration process we choose are related to the evolution of the cost function. They are enumerated here below.

1. If the cost function decrement is positive and smaller than the precision threshold ϵ:

$$\Gamma\left(\mathbf{s}^{k+1}\right) - \Gamma\left(\mathbf{s}^k\right) < 0$$
$$\text{and}$$
$$\left|\Gamma\left(\mathbf{s}^{k+1}\right) - \Gamma\left(\mathbf{s}^k\right)\right| < \epsilon.$$

   The found solution that minimizes the cost function is:

$$\widehat{\mathbf{s}} = \mathbf{s}^{k+1}.$$

2. If the cost function decrement is positive, the iteration process is stopped:

$$\Gamma\left(\mathbf{s}^{k+1}\right) - \Gamma\left(\mathbf{s}^k\right) > 0.$$

   The found solution that minimizes the cost function is:

$$\widehat{\mathbf{s}} = \mathbf{s}^k.$$

3. If the number of iterations exceed a maximum threshold k_{max}:

$$k \geq k_{max},$$

   where k is the number of the current iteration.

Cost Functions Comparison The improvements we introduced to the fitting procedure modify the shape of the original cost function. Let us examine here below the differences these improvements brought. The case of study is the cost function for a tether whose feature vector is $\mathbf{s} = [0.6, 0.6]^T$, which means slackness $H = 0.42\,\text{meter}$ and orientation angle $\alpha = 37°$. The projection of this tether in the image plane is drawn in Figure 3.9a.

The original cost function is depicted in Figure 3.14. We observe a sharp variation at $b = 0.6$. This is produced by the computation of large residuals, since the remote points (see Section 3.4.3.3) are not filtered. We note some local minima at $b < 0.6$ and $a \to 0$. Indeed, one of the solution to minimize the large residuals generated by the remote points is to tighten the tether.

The new cost function with the modifications we propose for the improved Gauss-Newton is depicted in Figure 3.15. It is much smoother than the original one and no sharp variation is observed. Local minima also exist. Therefore, the use of an initial guess of the tether shape will be important to achieve the global minimum, leading to a correct shape fitting and parameters estimations.

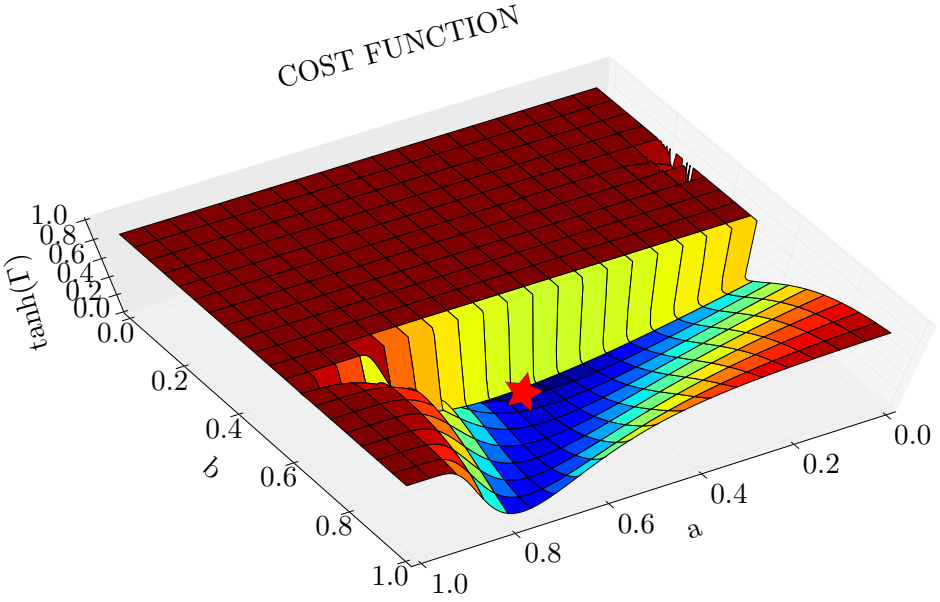


Figure 3.14: The hyperbolic tangent of the cost function of the catenary curve presented in Figure 3.9, whose feature vector is $\mathbf{s} = [0.6, 0.6]^T$, the tether total length is $L = 1.50\,\text{m}$ and $\Delta H = 0\,\text{m}$. The hyperbolic tangent was used to saturate the highest values of the cost function to 1. The function minimum is marked by the red star at $a = 0.6$, $b = 0.6$.

A more detailed study of cases on tether fitting is presented in the section dedicated to the simulation results, at the end of this Chapter. Next, we present the calculation

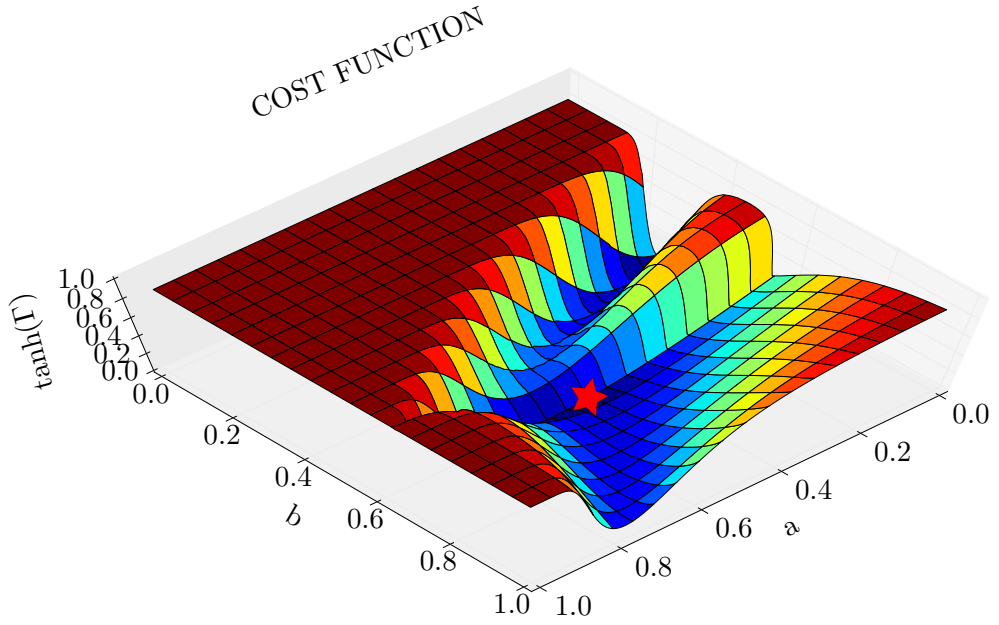


Figure 3.15: The hyperbolic tangent of the improved Gauss-Newton cost function of the catenary curve presented in Figure 3.9. The hyperbolic tangent was used to saturate the highest values of the cost function to 1. The function minimum at $a = 0.6$, $b = 0.6$, marked by the red star.

procedure of the tether shape initial guess.

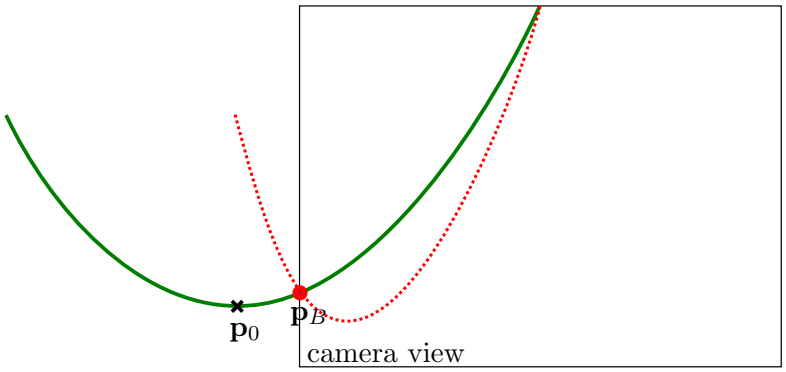
3.4.3.5 Initial Guess of Catenary Shape for Gauss-Newton

The Gauss-Newton algorithm proposed in the previous Section is always initialized at the starting point $\mathbf{s}^0 = [0.5, 0.5]^T$, which is not necessarily near the solution that minimizes the cost function. Since global convergence is not assured by the Gauss-Newton algorithm, local minima may be found. This would lead to the algorithm convergence but to wrong parameters estimation. In order to avoid these local minima, an initial guess near the global minimum should be used as starting point of the Gauss-Newton algorithm.

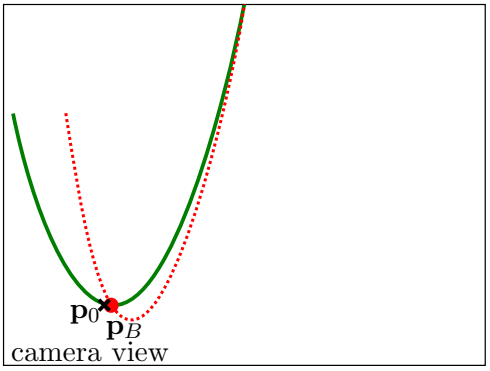
In the case of catenary fitting, we propose to use an initial guess based on the following approximation (see Figure 3.16):

The projection $\mathbf{p}_0$ of the tether 3D lowest point $\mathbf{P}_0$ is approximated with the tether lowest point detected in the image $\mathbf{p}_B$.

Our objective is to calculate an initial estimation, an initial guess, of the tether shape. This is achieved by calculating a catenary projection that passes through the tether lowest point in the image considering the approximation mentioned here above ($\mathbf{p}_B \approx \mathbf{p}_0$).



(a)



(b)

Figure 3.16: An illustration showing the difference between the projection of the tether 3D lowest point, $\mathbf{p}_0$ marked by a black cross, and the tether lowest point detected in the image, $\mathbf{p}_B$ marked by a red dot, is highlighted. The observed tether is drawn in green and the catenary curve corresponding to the initial guess is drawn by a red dashed line. (a) an embedded camera view where $\mathbf{p}_0$ is outside of the camera field of view. (b) another situation where $\mathbf{p}_0$ is inside camera field of view and $\mathbf{p}_B \approx \mathbf{p}_0$.

Evaluating equations (3.36), (3.37) and (3.38) at $\mathbf{p}_B \approx \mathbf{p}_0$ ($\zeta = D$) leads to:

$$H = y_B \frac{\sqrt{1-b^2}\,{}^cX_2 + b\,{}^cZ_2}{b + \sqrt{1-b^2}\,x_B} - {}^cY_2 \tag{3.60}$$

and

$$D = \frac{{}^cX_2 - x_B\,{}^cZ_2}{b + x_B\sqrt{1-b^2}}, \tag{3.61}$$

where x_B and y_B are the metric-coordinates of the tether lowest point in the image: $\mathbf{p}_B = [x_B, y_B, 1]^T$.

The relation between D and H can be easily obtained evaluating equation (2.60) at $\mathbf{P}_2$, yielding:

$$H = \frac{1}{C}\left[\cosh(CD) - 1\right] \tag{3.62}$$

In order to facilitate writing, let us redefine equations (3.60) and (3.61) as

$$H = \frac{G_2}{G_3} \tag{3.63}$$

and

$$D = \frac{G_1}{G_3}. \tag{3.64}$$

where

$$\begin{aligned} G_1 &= {}^cX_2 - {}^cZ_2 x_B \\ G_2 &= y_B \left(\sqrt{1-b^2}\,{}^cX_2 + b\,{}^cZ_2 \right) - {}^cY_2 G_3 \\ G_3 &= b + x_B \sqrt{1-b^2} \end{aligned}$$

Then, replacing equations (3.63) and (3.64) in (3.62) we obtain the new function

$$g(b) = \frac{CG_2}{G_3} - \cosh\left(\frac{CG_1}{G_3}\right) + 1 = 0. \tag{3.65}$$

Replacing $H = G_2/G_3$ in the expression of the catenary parameter C in equation (2.17) we obtain:

$$C = 2\frac{2\frac{G_2}{G_3} + \Delta H + 2L\sqrt{\frac{G_2}{G_3}\frac{\frac{G_2}{G_3}+\Delta H}{L^2-\Delta H^2}}}{L^2 - \left(2\frac{G_2}{G_3} + \Delta H\right)^2} \tag{3.66}$$

$$C = 2G_3\frac{2G_2 + G_3\Delta H + 2L\sqrt{G_2\frac{G_2+G_3\Delta H}{L^2-\Delta H^2}}}{G_3^2L^2 - (2G_2 + G_3\Delta H)^2}. \tag{3.67}$$

Replacing equation (3.67) in (3.65) yields:

$$g(b) = \widetilde{C}G_2 - \cosh\left(\widetilde{C}G_1\right) + 1 = 0, \tag{3.68}$$

where

$$\widetilde{C} = 2\frac{2G_2 + G_3\Delta H + 2L\sqrt{G_2\frac{G_2+G_3\Delta H}{L^2-\Delta H^2}}}{G_3^2L^2 - (2G_2 + G_3\Delta H)^2}.$$

The equation (3.68) can be used to find a root value b_{init} for the initial estimation of the orientation parameter (b) that corresponds to a catenary that passes through $\mathbf{p}_B$. The calculation of the slackness parameter (a) is obtained replacing equation (3.60) in Eq. (3.34a):

$$a = \frac{G_2}{G_3 H_{max}}. \tag{3.69}$$

In order to search for the root value of equation (3.68), we must use a numerical method. Among the possible numerical solving algorithms, we use the bisection (or

dichotomy) algorithm. We could have used other methods like an interval analysis method. However this method requires to find bounds of the function derivative, which is more complicated to implement. In all these methods the functions must be continuous in order to avoid numerical drift generated by function derivatives that tend to infinity.

Bounds for initial guess The root of equation (3.65) should be explored by the dichotomy algorithm within a bounded interval $b \in [b_{low}, b_{upp}]$ that ensures the existence of the catenary. This interval is given by the following constraints:

1. The tether lowest point $\mathbf{p}_B$ is in front of the robot, which means that $D > 0$;
2. The tether lowest point $\mathbf{p}_B$ is below the attachment points which means $H > 0$ and $H + \Delta H > 0$. The constraint is $H > \max(0, -\Delta H)$;
3. The tether lowest point $\mathbf{p}_B$ is between the attachment points and the catenary shape is not degenerated, which requires that $L > 2H + \Delta H$. Taking into account $H < H_{max}$, the constraint is $H < \min(H_{max}, \frac{L-\Delta H}{2})$.

The first constraint imposes that the catenary half-span between the tether attachment points is a positive quantity: $D > 0$. Knowing that the tether lowest point $\mathbf{p}_B$ is considered on the left of the tether attachment point ${}^c\overline{\mathbf{P}}_2$ for the study ($b > 0$), we have:

$$x_B < x_2 \tag{3.70}$$

$$x_B < \frac{{}^cX_2}{{}^cZ_2}, \tag{3.71}$$

which yields

$${}^cX_2 - {}^cZ_2 x_B > 0$$

and, from equation (3.64), we have:

$$G_1 > 0.$$

In order to have $D > 0$, since $G_1 > 0$, we get, from the denominator of equation (3.64):

$$G_3 > 0$$

$$b + x_B\sqrt{1 - b^2} > 0$$

Knowing that

$$\tan\alpha = \frac{b}{\sqrt{1 - b^2}},$$

we finally obtain:

$$\tan\alpha > -x_B.$$

The first bound is thus given by:

$$\alpha_{L1} = \tan^{-1}(-x_B) \tag{3.72}$$

The second constraint imposes $H > 0$ and $H > -\Delta H$. Let us write the slackness lower bound inequality from equation (3.63):

$$H > \max(0, -\Delta H) \tag{3.73}$$

$$\frac{G_2}{G_3} > \max(0, -\Delta H) \tag{3.74}$$

$$\frac{y_B\left(\sqrt{1-b^2}{}^cX_2 + b^cZ_2\right)}{b + x_B\sqrt{1-b^2}} - {}^cY_2 > \max(0, -\Delta H) \tag{3.75}$$

Let us introduce the tether slackness lower bound as:

$$\xi_{max} = \max(0, -\Delta H).$$

Replacing it in equation (3.75) we obtain:

$$E(\xi_{max})\tan\alpha > F(\xi_{max}) \tag{3.76}$$

where

$$F = -y_B{}^cX_2 + x_B(\xi_{max} + {}^cY_2)$$
$$E = y_B{}^cZ_2 - (\xi_{max} + {}^cY_2)$$

If $E > 0$, equation (3.76) yields a lower bound for α:

$$\alpha_{L2} = \tan^{-1}\left(\frac{F(\xi_{max})}{E(\xi_{max})}\right). \tag{3.77}$$

Otherwise, if $E < 0$, it yields an upper bound:

$$\alpha_{U2} = \tan^{-1}\left(\frac{F(\xi_{max})}{E(\xi_{max})}\right). \tag{3.78}$$

The third constraint imposes $H < \frac{L-\Delta H}{2}$ and $H < H_{max}$. From equation (3.63), let us write the inequality that gives the tether slackness upper bound:

$$H < \min\left(H_{max}, \frac{L-\Delta H}{2}\right) \tag{3.79}$$

$$\frac{y_B\left(\sqrt{1-b^2}{}^cX_2 + b^cZ_2\right)}{b + x_B\sqrt{1-b^2}} - {}^cY_2 < \min\left(H_{max}, \frac{L-\Delta H}{2}\right) \tag{3.80}$$

Let us introduce the tether slackness upper bound as:

$$\xi_{min} = \min\left(H_{max}, \frac{L-\Delta H}{2}\right).$$

Similarly to equation (3.76), we obtain the bounds for α as:

$$E(\xi_{min}) \tan \alpha < F(\xi_{min}) \tag{3.81}$$

If $E > 0$, equation (3.81) yields an upper bound:

$$\alpha_{U3} = \tan^{-1} \left(\frac{F(\xi_{min})}{E(\xi_{min})} \right) \tag{3.82}$$

Otherwise, if $E < 0$, it yields a lower bound:

$$\alpha_{L3} = \tan^{-1} \left(\frac{F(\xi_{min})}{E(\xi_{min})} \right). \tag{3.83}$$

The procedure that gives the upper and lower bounds for α is described by Algorithm 4, which finally yields:

$$b_{low} = \sin (\alpha_{low}) \tag{3.84}$$

and

$$b_{upp} = \sin (\alpha_{upp}). \tag{3.85}$$

The definition of the bounds ensures that the root searching algorithm will find a solution for the initial guess of the catenary shape that meets the conditions of existence of the catenary. These conditions were discussed in Section 2.1.3 and state, in summary, that the tether lowest point remains between the attachment points. The uniqueness of the solution was not established but we are only interested in getting one solution for the initial guess.

An example of the initial guess function $g(b)$, defined by equation (3.68), is drawn in Figure 3.17. This function was generated by a catenary curve with feature vector $\mathbf{s} = [0.6, 0.6]^T$. The root of $g(b)$ is searched inside the bounds $b_{low} = 0.54$ and $b_{upp} = 1.0$, which are marked by green vertical lines. For this case, the initial guess of the catenary shape is given by $\mathbf{s} = [0.66, 0.57]^T$.

Algorithm 4 Bounds for Initial Guess

procedure BIG(${}^c\overline{\mathbf{P}}_2, \mathbf{p}_B, L, \Delta H, H_{max}$)
  $x_B, y_B \leftarrow extractCoordinates\,(\mathbf{p}_B)$
  ${}^cX_2, {}^cY_2, {}^cZ_2 \leftarrow extractCoordinates\,\left({}^c\overline{\mathbf{P}}_2\right)$
  $\alpha_{low} \leftarrow 0$
  $\alpha_{upp} \leftarrow \pi/2$
  $\alpha_{L1} \leftarrow \tan^{-1}(-x_B)$ ▷ constraint 1 lower bound
  if $\alpha_{L1} > \alpha_{low}$ **then**
    $\alpha_{low} \leftarrow \alpha_{L1}$
  $\xi_{max} \leftarrow \max(0, -\Delta H)$
  $E \leftarrow y_B{}^cZ_2 - (\xi_{max} + {}^cY_2)$
  $F \leftarrow -y_B{}^cX_2 + x_B(\xi_{max} + {}^cY_2)$
  if $E > 0$ **then**
    $\alpha_{L2} \leftarrow \tan^{-1}\left(\frac{F}{E}\right)$
    if $\alpha_{L2} > \alpha_{low}$ **then**
      $\alpha_{low} \leftarrow \alpha_{L2}$ ▷ constraint 2 lower bound
  else
    $\alpha_{U2} \leftarrow \tan^{-1}\left(\frac{F}{E}\right)$
    if $\alpha_{U2} < \alpha_{upp}$ **then**
      $\alpha_{upp} \leftarrow \alpha_{U2}$ ▷ constraint 2 upper bound
  $\xi_{min} \leftarrow \min(H_{max}, \frac{L-\Delta H}{2})$
  $E \leftarrow y_B{}^cZ_2 - (\xi_{min} + {}^cY_2)$
  $F \leftarrow -y_B{}^cX_2 + x_B(\xi_{min} + {}^cY_2)$
  if $E > 0$ **then**
    $\alpha_{U3} \leftarrow \tan^{-1}\left(\frac{F}{E}\right)$
    if $\alpha_{U3} < \alpha_{upp}$ **then**
      $\alpha_{upp} \leftarrow \alpha_{U3}$ ▷ constraint 3 upper bound
  else
    $\alpha_{L3} \leftarrow \tan^{-1}\left(\frac{F}{E}\right)$
    if $\alpha_{L3} > \alpha_{low}$ **then**
      $\alpha_{low} \leftarrow \alpha_{L3}$ ▷ constraint 3 lower bound
  $b_{low} = \sin(\alpha_{low})$
  $b_{upp} = \sin(\alpha_{upp})$
  return b_{low}, b_{upp}

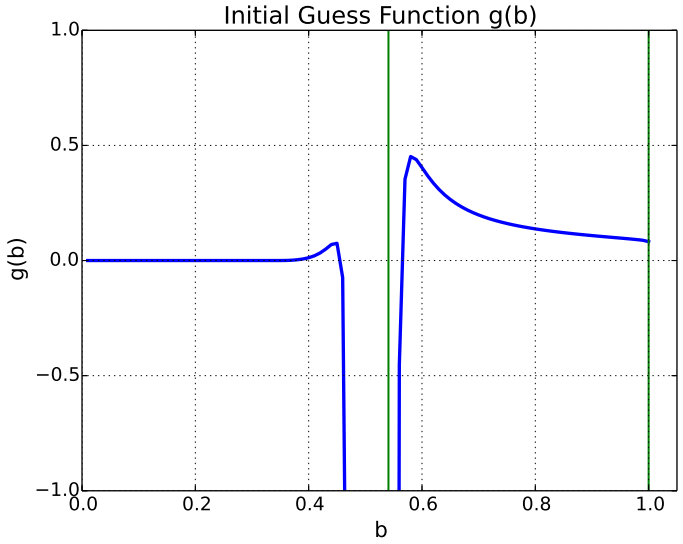


Figure 3.17: The initial guess function $g(b)$ given by equation (3.68) for a catenary whose feature vector is $[0.6, 0.6]^T$. The root of $g(b)$ is searched in the interval defined by the green vertical lines.

3.5 Results

We present in this Section the simulation results we obtained for tether shape estimation. Three methods of catenary fitting are tested: the standard Gauss-Newton algorithm (GN), the improved Gauss-Newton without initial guess (IGN) and improved Gauss-Newton with initial guess (IGN + IG).

First, the estimation error of the feature vector is evaluated in the entire range of definition ($\mathbf{s} \in]0,1[\times]0,1[$), considering non-noisy data and an infinite camera field of view, which means that the 2-D points are generated from the theoretical projection equation of the catenary into the image (see equation (3.36)), and that all points of the tether are taken into account.

Next, these methods are tested in a simulation environment comprising all aspects of tether shape estimation, from detection to fitting procedure. Two study cases where the camera is static are presented to demonstrate the improvements our modifications brought to catenary curve fitting and, consequently, to tether shape estimation. In addition, to evaluate the fitting robustness, the feature vector is estimated all along a predefined path of the camera.

3.5.1 Focus on Feature Vector Estimation Error

In this study, image tether points are generated through the catenary projection equation. The tether detection is not simulated and the camera field of view is assumed as infinite. Then, these points are used as input for the studied fitting procedures, whose accuracy is compared through the following error measurement:

$$\mathbf{e} = |\mathbf{s}_{sim} - \widehat{\mathbf{s}}|$$

where $\mathbf{s}_{sim}$ is the feature vector used to simulate the tether points, $\widehat{\mathbf{s}}$ is the estimation provided by the fitting procedure, and $|.|$ is the Euclidean norm. Since the input data is not noisy, this study allows to check the estimation error arising from the optimization procedure itself.

The simulated tether is composed of 1000 points, which is a constant value. The tether is 1.50 meter long, the height difference between the attachment points is zero ($\Delta H = 0$) and the maximum authorized slackness is $H_{max} = 0.70$ meter. The fitting algorithm gain is set to $\lambda = 0.5$ (see equation (3.42)).

Figure 3.18 presents the feature vector estimation error and the exit status for the standard Gauss-Newton (GN). This algorithm produces accurate results when the solution is near the starting point $\mathbf{s}^0 = [0.5, 0.5]^T$. Around $b = 0.5$, we observe that the shape estimation is more accurate for low values of a-parameter and worse for large values. This occurs because, for a same orientation angle, the slacker the tether is ($a \rightarrow 1$), the tighter it becomes in the image. Both tether portions near the attachment points get closer and more vertical in the image and the catenary shape is less characterized.

Analyzing Figure 3.18a, we could incorrectly conclude that the GN algorithm estimation error is moderately low for most of cases, particularly in the zone around $\mathbf{s} = [0.5, 0.5]^T$. However, the algorithm does not converge towards the solution and

remains stuck near the departing point, which is in the middle of the feature vector variation range. The iteration process is interrupted at the beginning due to an increase in the cost function or singular Jacobian computation, as shows Figure 3.18b. This behavior is expected because, as seen at the end of Section 3.4.3.4, the existence of large residuals produces sharp variations in the cost function of the standard Gauss-Newton. Moreover, the feature vector is not bounded and it may be evaluated in regions where our catenary model is not defined.

Figure 3.19a presents the estimation error for the improved Gauss-Newton without initial guess (IGN). This algorithm may reach the solution when $0.4 < b < 0.8$. We observe that the error increases for low values of the b-parameter. In such cases, the cable is very tight in the image and catenary shape is not well characterized. If the a-parameter approaches zero, the estimation accuracy becomes even worse since the tether cable is taut. The catenary is thus degenerated into a straight line in the image.

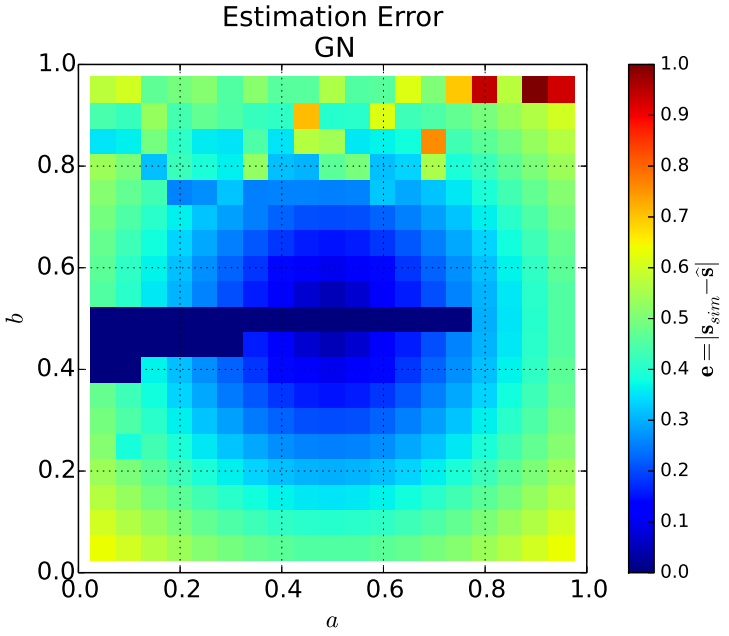
The exit status for the IGN algorithm is depicted in Figure 3.19b. Globally, the algorithm converges when the solution is near the starting point $\mathbf{s}^0 = [0.5, 0.5]^T$, specially with respect to the b-parameter. The reason is that the b-parameter has a direct influence on the amount of points to be removed due to large residuals computation (see Section 3.4.3.3). If too many points are filtered, the iteration process cannot be continued and the algorithm stops. These cases are signalized by the red squares in the legend and typically occur for tight tether projections in the image (low values of b-parameter).

Figure 3.20a presents the estimation error for the improved Gauss-Newton with initial guess (IGN + IG). The error is kept close to zero throughout the whole plane a, b, except for low values of a and b parameters. In these few cases, the tether is taut ($a \to 0$) and its projection in the image is tight ($b \to 0$). The catenary shape in the image plane is thus degenerated into a straight line and the fitting procedure is not able to improve the shape estimation given by the initial guess.

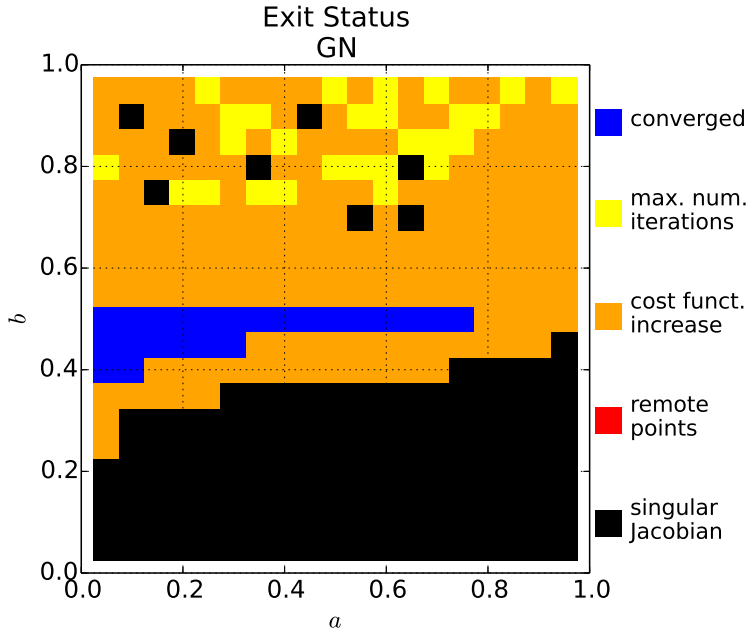
Since the camera field of view is assumed to be infinite, the projection of the tether lowest point is always present in the image plane. As a result, the approximation used to calculate the initial guess on the tether shape and initialize the IGN + IG algorithm is always near the solution. This greatly enhances the probability of the global minimum to be reached. That is the reason why the IGN +IG has such a better accuracy in comparison with the IGN (see Figures 3.19a and 3.20a).

Regarding the exit status for the IGN + IG (Figure 3.20b), the algorithm converges in the vast majority of cases. This does not occur for some few cases of low values of a and b parameters, as expected. There is no exit due to excessive points removal (red squares), since the starting point is always close to the solution.

In the next Section we will study the whole fitting chain including image processing and pixel/meter conversion. Taking into account a limited camera field of view, 3-D projection of the tether lowest point will not always be inside the image. In these cases, we expect the initial guess to be less accurate, nonetheless ensuring the convergence. Moreover, we may find additional error due to image processing.

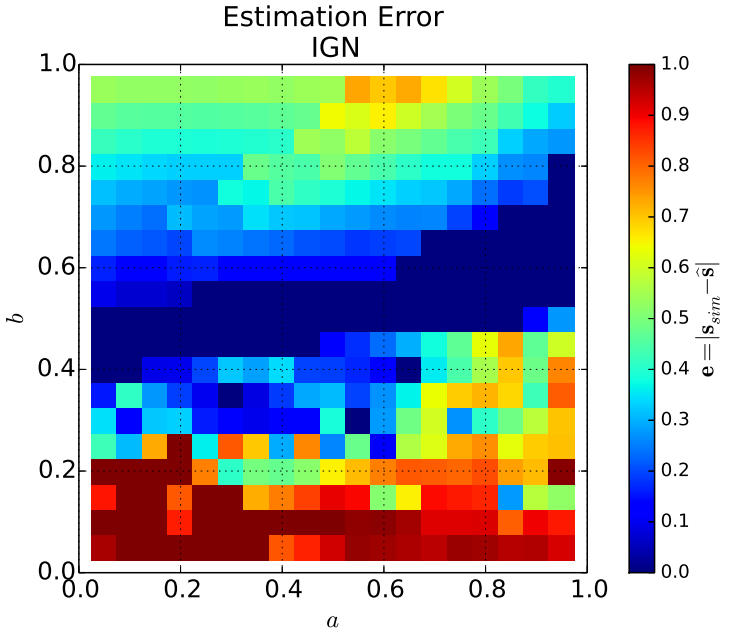


(a)

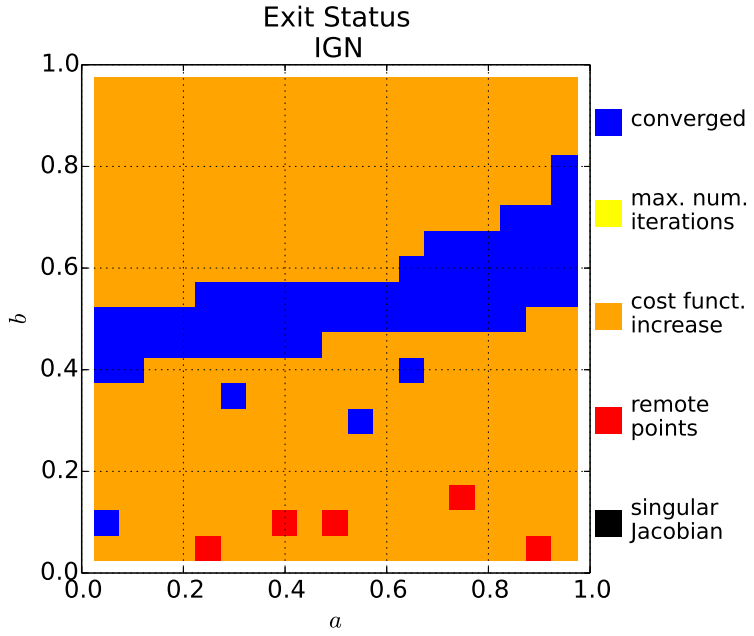


(b)

Figure 3.18: Simulation results for the standard Gauss-Newton algorithm (GN). (a) the feature vector estimation error is represented in color scale. (b) the algorithm exit status is depicted in color code.

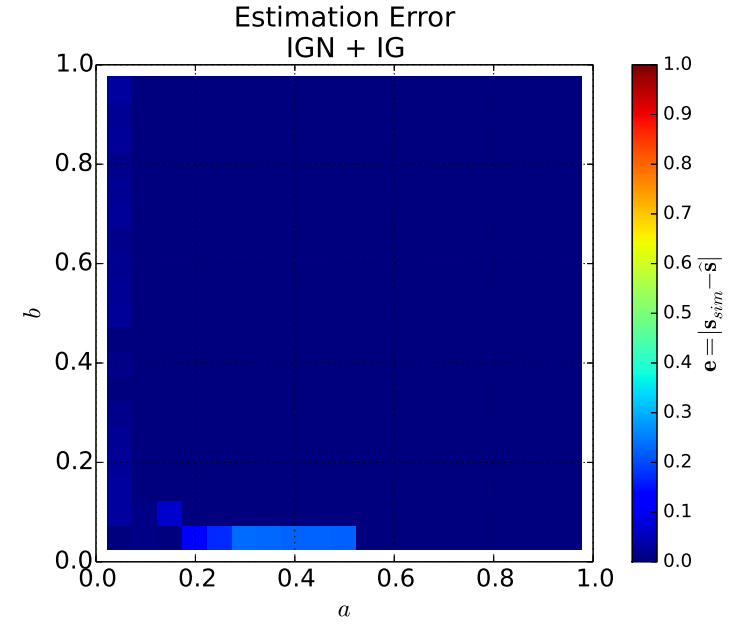


(a)

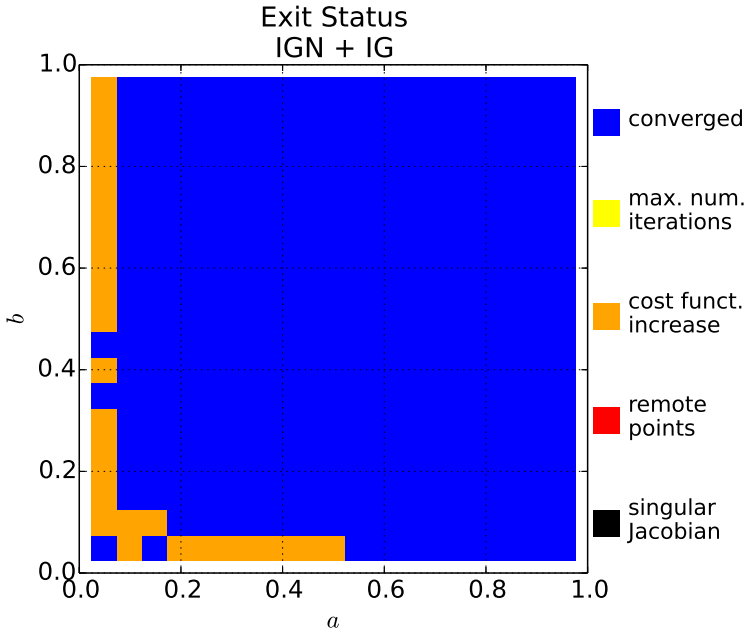


(b)

Figure 3.19: Simulation results for the improved Gauss-Newton algorithm without initial guess (IGN). (a) the feature vector estimation error is represented in color scale. (b) the algorithm exit status is depicted in color code.



(a)



(b)

Figure 3.20: Simulation results for the improved Gauss-Newton algorithm without initial guess (IGN + IG). (a) the feature vector estimation error is represented in color scale. (b) the algorithm exit status is depicted in color code.

3.5.2 Study Cases in a Simulated Environment

The simulations are carried out on Gazebo (Koenig and Howard, 2004) using the UUV (unmanned underwater vehicle) package (Manhães et al., 2016). This simulation environment comprises all aspects of tether shape estimation, from detection to fitting.

The simulation scenario is composed of two BlueROV1 connected by a sagging orange tether (see Figure 3.21). The robots are equipped with simulated pressure sensors, frontal and rear cameras. Hydrodynamics aspects were only considered in the robots motions, modeled through the Fossen equations (Fossen, 2011), and viscous forces of water on the tether cable were neglected. The robot's hydrodynamic parameters used in the simulations are available in Aili and Ekelund (2016).

The simulated tether is composed of 150 links of 0.01 m length connected by spherical ball joints. The tether is 3 millimeter thick and has a total length of 1.50 m and total mass of 0.150 kg. The maximum authorized slackness is selected to be $H_{max} = 0.70$ m.

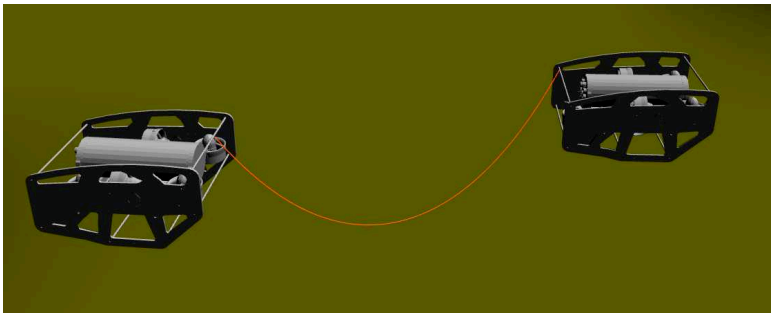
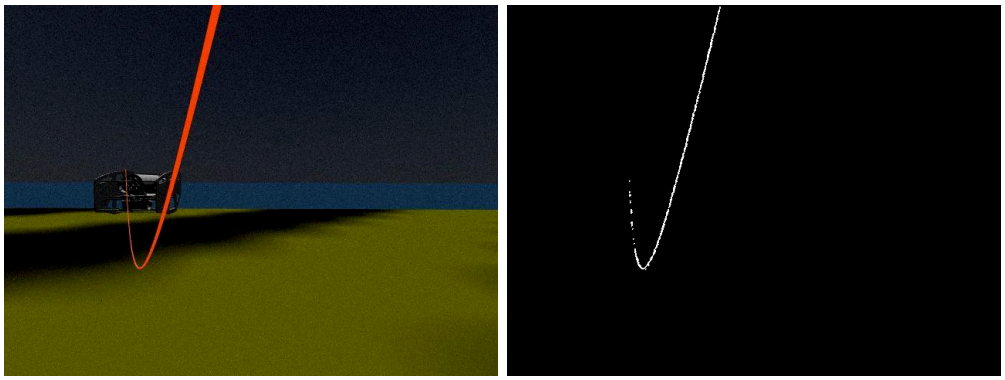


Figure 3.21: The simulation of two BlueROV 1 linked by an orange sagging tether on Gazebo.

Three cases of tether shape estimation are studied. In the first case, all three algorithms, namely GN, IGN and IGN + IG, converge to the solution. In the second case, only the IGN + IG algorithm reaches the solution. The IGN + IG algorithm robustness is tested in the third study case, where the follower robot executes a predefined path, varying the tether slackness, orientation and height difference between attachment points. All algorithms use the same step gain $\lambda = 0.5$ (see equation (3.42) for the Gauss-Newton iteration process definition).

The simulation is useful to have a ground truth for the features to be compared with their estimation. The tether slackness H results from the search for the lowest link and the Z-coordinates of the tether attachment point on the follower. The orientation angle α is calculated thanks to the 3-D position of the robots. The value of ΔH is directly obtained from the difference of height between the first and last link of the tether. The ground truth will be used in the third study case.

Case 1 The image source and tether detected points of case 1 are presented in Figure 3.22. The tether is entirely visible in the image and its shape given by the feature vector $\mathbf{s} = [0.40, 0.45]^T$, which is the solution of our minimization problem. This feature vector corresponds to a tether with slackness $H = 0.28$ meter and orientation angle $\alpha = 27°$.



(a) (b)

Figure 3.22: The (a) source image and (b) tether detected points of study case 1. The tether shape is given by the feature vector $\mathbf{s}[0.40, 0.45]^T$, which corresponds to a slackness $H = 0.28$ meter and orientation angle $\alpha = 27°$.

The cost function and the steps of the standard Gauss-Newton (GN) are drawn in Figure 3.23. As we can observe, the algorithm starting point is in the blue zone, where the cost function values are lowest. This allows to converge quickly to the solution, after 8 iterations. The cost function, however, is not smooth everywhere. We observe a sharp variation around $b = 0.45$ due to the presence of large residuals, as previously explained in Section 3.4.3.3.

Both IGN and IGN + IG algorithms converge to the solution, as expected. The cost function and steps of both optimization algorithms are drawn in Figure 3.24. The cost function is much smoother than that of the standard Gauss-Newton. No sharp variation is observed, thanks to the function of large residuals removal (see Section 3.4.3.4). Both starting points are near the solution (blue zone). The IGN and IGN + IG algorithms converge after 7 and 8 iterations, respectively. Comparing IGN and GN optimizations, we observe that both realize a large first step. Otherwise, the convergence of IGN + IG is more gradual since the departure point is closer to the solution.

The initial and final tether shape estimation for the three studied procedures are depicted in Figure 3.25. As expected, the estimated catenary fits the tether detected points using any of the three methods. Notably, the tether shape initial guess is very close to the solution.

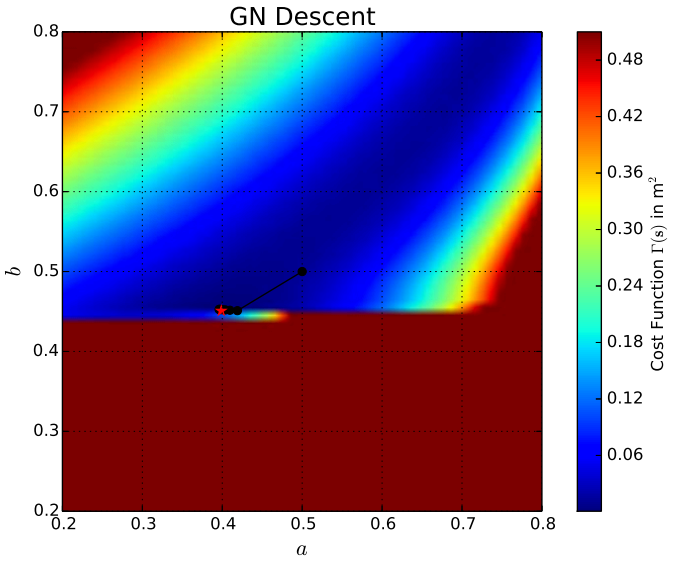


Figure 3.23: The descent of the GN algorithm marked by black dots. The starting point is at $\mathbf{s}^0 = [0.5, 0.5]^T$ and the solution is at $\mathbf{s} = [0.40, 0.45]^T$, marked by a red star. The cost function values are plotted in the background using the color scale depicted on the right.

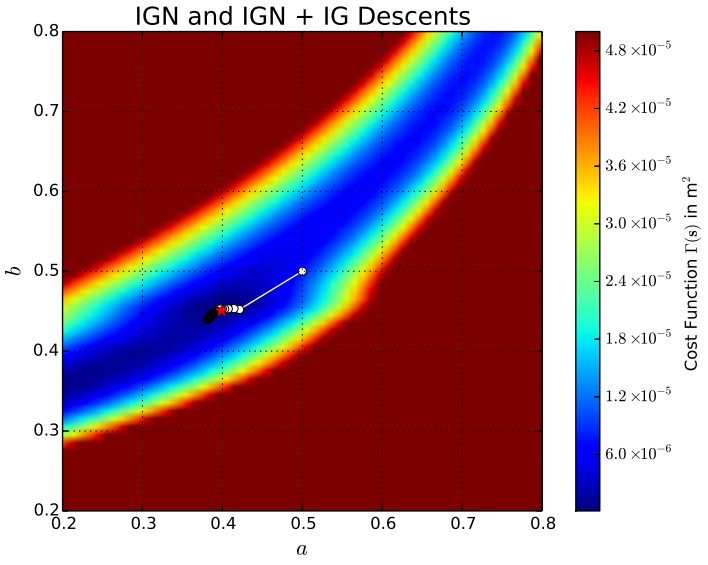
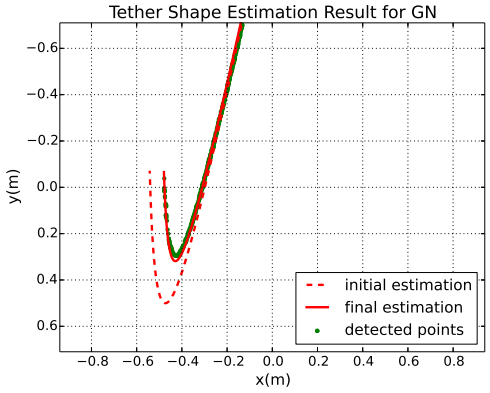
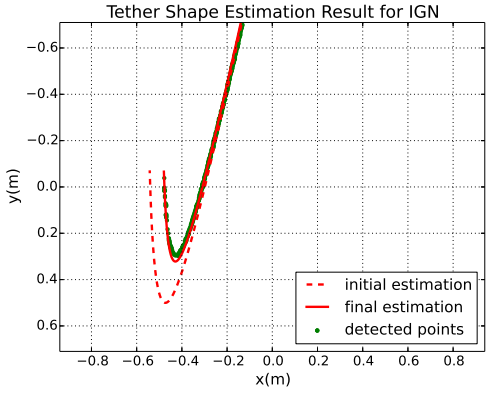


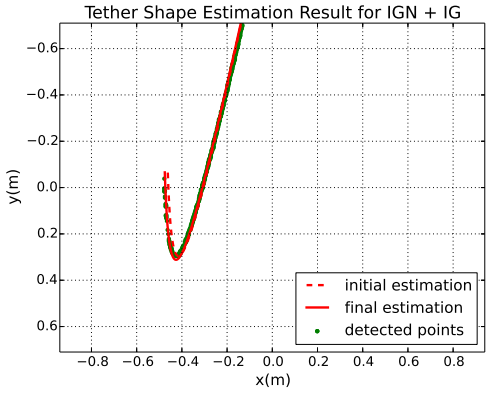
Figure 3.24: Comparison of the descent of IGN (white dots) and IGN + IG (black dots) algorithms. The solution is at $\mathbf{s} = [0.40, 0.45]^T$, marked by a red star. The cost function values are plotted in the background using the color scale depicted on the right.



(a)



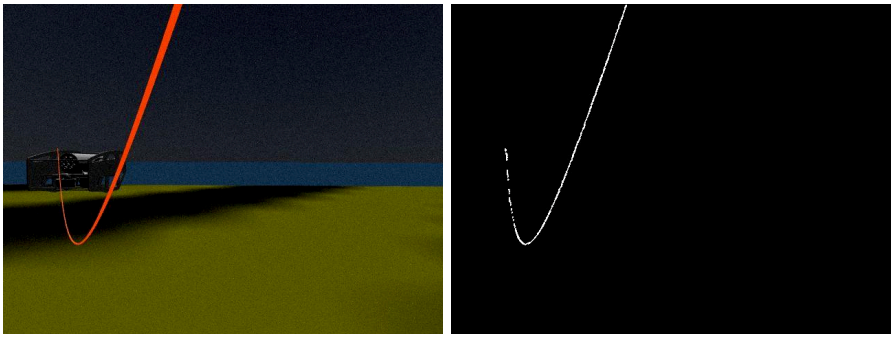
(b)



(c)

Figure 3.25: Catenary curve fitting problem depicted in the image plane. The tether detected points are drawn in green dots, the tether catenary initial estimation is drawn by a red dashed line and the final estimation is drawn by a red full line. The tether shape estimation is achieved by the three methods: (a) GN, (b) IGN and (c) IGN + IG algorithms. The initial guess of the tether shape is given by $\mathbf{s}^0 = [0.38, 0.44]^T$. The solution is at $\mathbf{s} = [0.40, 0.45]^T$.

Case 2 The image source and tether detected points of case 2 are presented in Figure 3.26. The tether is also entirely visible in the image and its shape is given by the feature vector $\mathbf{s} = [0.4, 0.6]^T$, which corresponds to $H = 0.28$ meter and orientation angle $\alpha = 37°$.



(a) (b)

Figure 3.26: The (a) source image and (b) tether detected points of study case 2. The tether shape is given by the feature vector $\mathbf{s}[0.4, 0.6]^T$, which corresponds to $H = 0.28$ meter and $\alpha = 37°$.

The cost function and the steps of the standard Gauss-Newton are drawn in Figure 3.27. The algorithm is initialized in a region with large residuals, which involves the computation of large numerical values in the Jacobian and leads to an ill-conditioned matrix. As a consequence, the algorithm cannot converge to the solution. It reaches $\mathbf{s} = [0.02, 0.50]^T$ after 172 steps and exits due to an increase of the cost function. The final feature vector corresponds to a tight tether ($a \approx 0$). The initial and final tether shape estimations are depicted in Figure 3.29a.

The cost function and steps of the IGN and IGN+IG algorithms are depicted in Figure 3.28. The standard starting point $\mathbf{s}^0 = [0.5, 0.5]^T$ is far away from the solution, in the red zone. As a result, the IGN fitting procedure reaches a local minimum, whereas IGN + IG algorithm converges to the solution. Indeed, the Gauss-Newton algorithm does not ensure global convergence.

Without the initial guess, the IGN algorithm reaches a local minimum at $\mathbf{s} = [0.27, 0.52]^T$ after 7 steps. As presented in Figure 3.29b, this feature vector corresponds to a catenary that minimizes the residuals related to the portion of the detected tether near the robot attachment point (top middle of the image).

With the initial guess ($\mathbf{s}^0 = [0.47, 0.57]^T$), the IGN+IG algorithm reaches the solution after 14 steps. We observe in Figure 3.29c that the final estimation of the catenary fits the tether detected points. The initial estimation is close to the solution and passes through the tether lowest point in the image, as explained in Section 3.4.3.5. This limits the number of large residuals to be filtered and helps the algorithm to converge to the global minimum.

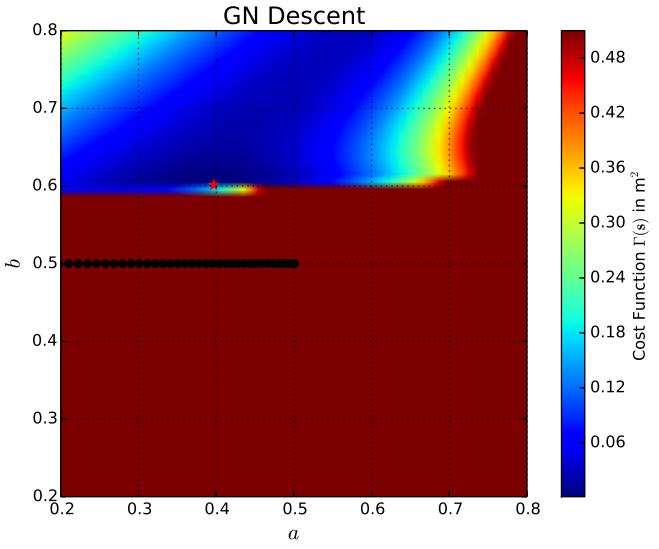


Figure 3.27: The descent of a standard Gauss-Newton algorithm marked by black dots. The starting point is at $\mathbf{s}^0 = [0.5, 0.5]^T$ and the solution is at $\mathbf{s} = [0.4, 0.6]^T$, marked by a red star. The cost function values are plotted in the background using the color scale depicted on the right. The algorithm does not reach the solution.

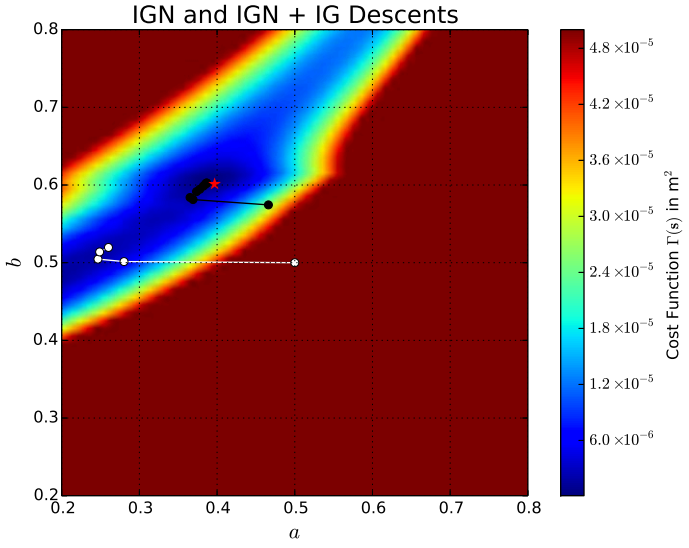
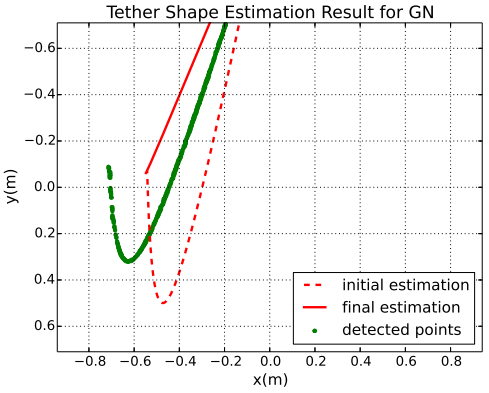
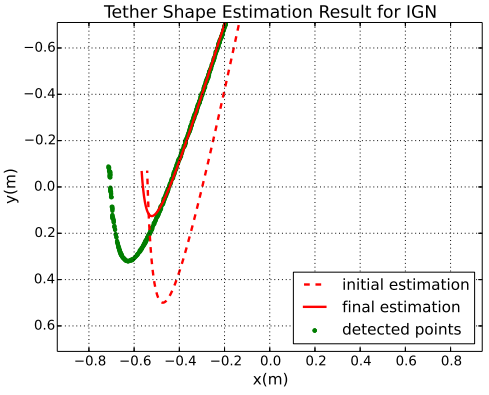


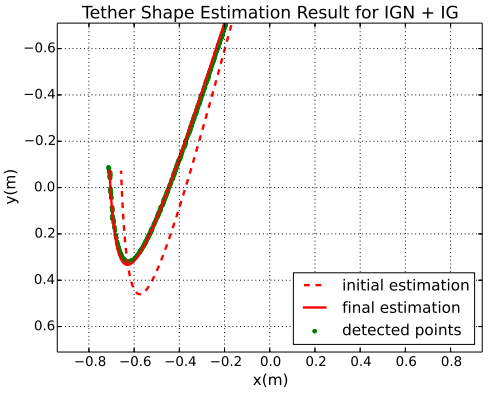
Figure 3.28: Comparison of the descent of the IGN (white dots) and IGN+IG (black dots) algorithms. The solution is at $\mathbf{s} = [0.4, 0.6]^T$, marked by a red star. The cost function values are plotted in the background using the color scale depicted on the right. The solution is only reached if an initial guess ($\mathbf{s}^0 = [0.47, 0.57]^T$) of the tether shape is given.



(a)



(b)



(c)

Figure 3.29: Catenary curve fitting problem depicted in the image plane for the three methods studied: (a) GN, (b) IGN and (c) IGN+IG. The tether detected points are drawn in green, the initial estimation is drawn by a red dashed line and the final estimation is drawn by a red full line. The correct tether shape estimation is only achieved by the improved Gauss-Newton with initial guess (IGN+IG).

A summary with the number of computed steps and exit status of study cases 1 and 2 is given on Table 3.1 for all the three algorithms under study. The exit code 0 corresponds to algorithm convergence and the code -1 corresponds to iteration interrupted due to increase in cost function.

	GN		**IGN**		**IGN+IG**	
	steps	exit	steps	exit	steps	exit
case1 $\mathbf{s} = [0.40, 0.45]^T$	8	0	7	0	8	0
case2 $\mathbf{s} = [0.40, 0.60]^T$	172	-1	8	-1	14	0

Table 3.1: A summary of study cases 1 and 2. The exit code 0 corresponds to algorithm convergence and the code -1 corresponds to iteration interrupted due to increase in cost function.

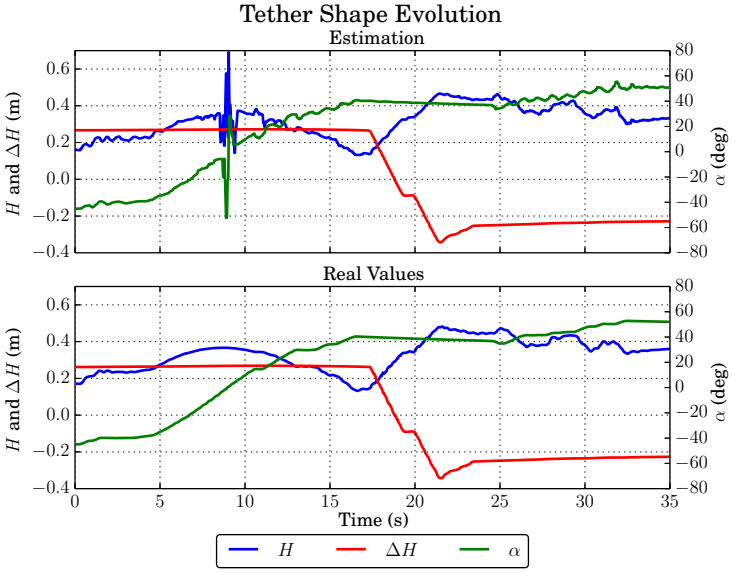
Case 3 The tether shape estimation is tested in a case where the follower robot executes a predefined path, while the three features, namely H, α and ΔH, are made variable. Moreover, during the motion, the robot will have to cross the singularity $b = 0$. At the beginning $(0 - 17\text{s})$, $\alpha = -45^\circ$ and $\Delta H = 0.26\,\text{m}$, the follower robot executes a lateral motion in order to pass through the singularity $\alpha = 0^\circ$. Then $(17 - 24\text{s})$, the robot goes downwards and stops when $\Delta H = -0.23\,\text{m}$. At the end $(24 - 35\text{s})$, one more lateral motion is executed until $\alpha = 52^\circ$ so that the tether lowest point is outside the camera field of view.

The value of ΔH is estimated from pressure sensors mounted on the robots, while the tether parameters H and α are estimated through the curve fitting procedure we propose. The evolution of the estimated and real values of these variables throughout the executed path are depicted in Figure 3.30.

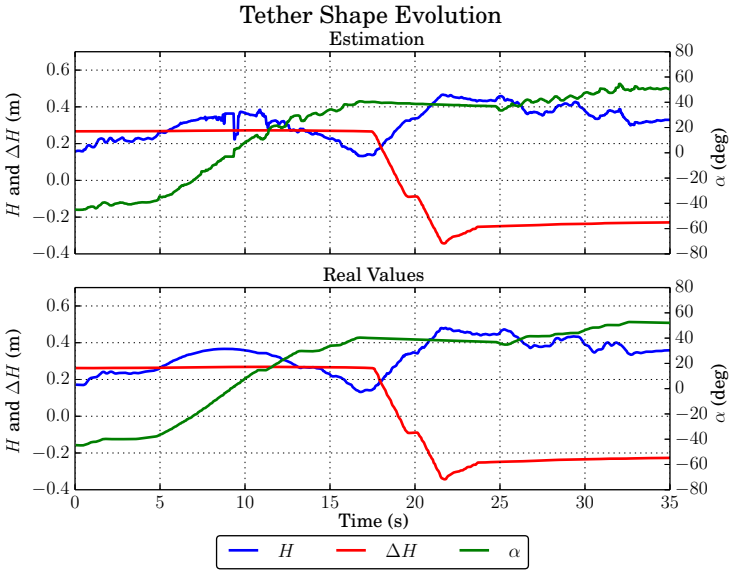
We observe that the tether shape estimation procedure yields accurate results all along the path, except around the singularity $b = 0$. Some oscillations are noted in the estimated values of H and α. This is due to the natural motion of the tether when it is displaced by the robot. In the simulation the cable is not subject to the damping effect of the water. Back-and-forth movements on the cable add some noise to the estimation of H, while the swinging motion of the tether plane around the vertical is responsible for the oscillations on α. Actually, the catenary model assumes a static cable. Therefore, the tether shape variations caused by the cable motion are not taken into account in our fitting procedure. The tether cable does not behave as a perfect catenary all the time, which generates slight inaccuracies in the shape estimation.

The crossing of the singularity $\alpha = 0^\circ$ occurs around 8 s, when we observe a poor estimation of H and α (see Figure 3.30a). In such cases, the tether is very tight in the image, which can be noted by examining the standard deviation of the detected points abscissa. In such cases, instead of launching the fitting procedure, the previously estimated values of the tether parameters can be maintained (see Figure 3.30b). Thus, the estimated curves of H and α are flat around the singularity, instead of being

unstable. Furthermore, the tether shape control law, that will be presented in the next Chapter, can be used to predict the shape features evolution from the robot motion (Laranjeira et al., 2017). This prediction can help to cross the singularity.



(a)



(b)

Figure 3.30: Estimation and real values of the tether shape during the execution of a path by the follower robot. (a) the fitting procedure is used during the whole path execution. (b) during the singularity crossing (around 8 second), the fitting procedure is stopped and the previously estimated values of the tether shape are maintained.

3.5.3 Discussion

The improved Gauss-Newton with an initial (IGN+IG) guess has been demonstrated to yield accurate results in tether shape estimation. The modifications proposed make the cost function much smoother, avoiding singularities in the Jacobian computation and allowing the algorithm to reach the solution. Moreover, the feature vector is bounded, which ensures that the catenary projection equation is always evaluated within the definition range of the shape parameters a and b. As soon as the initial guess provides a fine first approximation of the tether shape, the algorithm converges to the solution and local minima are avoided. This occurs when the tether lowest point is inside the camera field of view.

Some enhancements on the proposed procedure of tether shape estimated can be discussed. They are addressed here below.

Detection Improvements The catenary projection equation may be used to implement a model tracker (Marchand and Chaumette, 2005). The nonlinear minimization process would thus start from the estimated tether shape resulting from the previous image. This would lead to a faster computation of the estimation loop. However, the drawback of such tracking methods is the loss of robustness in environments where there are objects the tether can be confused with. A strategy of tracking reinitialization, with a new the tether detection, should thus be additionally implemented.

Choice of Residual Distance Function We use a vertical distance between points to compute the residuals. This distance is not very discriminating in terms of shape variation when the tether is tightened in the image and its projection is similar to two vertical line segments connected. An alternative would be to use a horizontal distance, but the same problem would be faced in cases where the tether is taut and its projection in the image plane approaches a horizontal line. The orthogonal distance between the curves could be also computed. However, this increases the complexity (and thus computing time) of the fitting procedure, since another minimization problem included inside the estimation loop in order to find the orthogonal distance.

Optimization Algorithm Other optimization algorithms could be investigated. For example, the feature vector boundaries could be dealt with a barrier function, as is done in interior points methods. Moreover, inequality constraints could be added to the problem. The information provided by the tether lowest point in the image could be thus further explored, in addition to the initialization step. The minimizing catenary would be constrained to pass near the detected tether lowest point in order to lead the fitting procedure to reach the global minimum. This would be interesting in cases where the tether lowest point projection is outside the camera field of view and the initial guess is not close to the solution.

Tether Attachment Point Influence The first points of the detected tether could be used to online refine the measured position of the tether attachment point ${}^{c}\mathbf{P}_2$. This

point has a non-negligible influence on the fitting, since it defines the departing point of the tether in the image. It is assumed to be a constant, but, in fact, variations on its actual position may occur. A solution would be to estimate it as an additional parameter of the curve fitting. However, this would lead to three more derivatives and to additional local minima in the cost function.

Additional Sensors The catenary model may be degenerated into a vertical line in the image when the tether plane is aligned with the camera optical axis. This singularity could be avoided through the addition of another camera that films the tether from another point of view. Hence, this degenerated case will never occur simultaneously in both images, and the fusion of such complementary visual information would lead to an accurate tether shape estimation in an wider workspace.

Other sensors, such as 3-D accelerometers could be added to the tether ends and provide additional information about its shape. The cable plane vertical orientation and its departure angles could be estimated from the accelerometers inclination with respect to the gravitational field. This additional data can be used together with the visual feedback in order to produce a more robust shape estimation, namely in complex scenes where tether detection may be less accurate. In this thesis, we assumed that the tether is not instrumented, as it is the case with most of the tethers found in the market. However, new products have been recently introduced where inertial measurements units are distributed throughout the cable (Frank et al., 2013), which are promising.

Model Improvements The cable model studied is the catenary curve, which is adapted for sagging cable modeling. Negatively buoyant tethers are sagging cables, and the catenary curve was used to estimate their 3-D shape from curve fitting. The features issued from the catenary model, the slackness and orientation parameters, give a physical and intelligible notion of the cable shape. In this thesis, the tether plane is assumed to remain in the vertical. This degree of freedom can be integrated to the tether model in future works. Moreover, the physical interaction between tether, robots and surrounding water can be studied in an hydrodynamic modeling of the system. Other models could also be investigated, such as parabolas and splines, although their parameters reflect a less physical notion of the cable shape. Finite element models have been recently proposed a system composed of a ROV and a cable. The main advantage of the catenary model is the reduced number of parameters, which allows a fast and analytical computation of the cable shape and its projection in the image. As long as the disturbances in water are limited, the catenary model is sufficient to design a local vision-based estimator and controller of the tether shape and regulate the relative positions of robots inside a chain.

3.6 Conclusions

In this Chapter, the procedure of tether 3-D shape estimation from camera feedback was presented. First, the pinhole camera model and the procedure of tether points detection in the image were introduced. Then, the catenary 3-D equation of the tether cable was developed in the camera frame, and its projection in the image plane was calculated. The tether 3-D shape is given by a feature vector that expresses the tether slackness and orientation with respect to the robot. The objective is to estimate the feature vector, and consequently the tether 3-D shape, from the detected points in the image through a curve fitting process. Therefore, a Gauss-Newton-like algorithm was used to fit the catenary projection to the tether detected points and thus estimate its 3-D shape.

Four new functions were added to the standard Gauss-Newton algorithm in order to improve the fitting procedure performance. Three residual filtering functions were introduced and used to avoid singularities in the Gauss-Newton Jacobian, speed up its computation, and to enhance the algorithm robustness to detection noise. The fourth function aims to bound the feature vector throughout the iteration process. Moreover, an initialization heuristic was introduced in order to obtain a starting point closer to the solution and hence avoid local minima.

The tether shape estimation procedure was validated in simulations and it has been shown to produce accurate results. The advantage of the simulation is to provide a ground truth to check the validity of the algorithm. Experimental results on real images were carried out with terrestrial robots in Laranjeira et al. (2017). These experiments have shown the robustness of the algorithm in presence of light change and detection noise. The proposed improvement functions added to the standard Gauss-Newton algorithm (GN) make the cost function much smoother, avoiding singularities in the Jacobian computation. The algorithm initialization heuristic has been shown to ensure the global minimum is reached as soon as the tether lowest point projection is visible in the image. Possible improvements in tether detection and shape estimation procedures were also addressed and the use of alternative methods and sensors were discussed.

Chapter 4

Tether Shape Control

In this Chapter the control scheme used to regulate the tether shape from visual feedback is introduced. We propose a new visual servoing scheme for parametric deformable objects. This new vision based control is innovative in two ways: firstly because it deals with deformable objects, secondly because the object is linked to the robot and its deformation is directly induced by the robot motion itself.

We will see in the literature survey why only a few research work on visual servoing deals with deformable objects, whereas the literature dealing with rigid objects is very extensive. We will especially point out the differences between the control of rigid objects and deformable objects shape. Furthermore, only a few works envision the case where the objects are directly attached to the robot and modified by the robot's motion.

Next, the general control scheme is presented: a low level controller based on IMU (inertial measurement unit) data is dedicated to high speed control of the pitch and roll and a high level controller is designed for the positioning of the follower robot with regards to tether shape.

We will show how the standard visual servoing scheme, proposed by Chaumette and Hutchinson (2006, 2007), can be adapted to control a robot relatively to deformable objects linked with it. Hence we will develop the equations of our visual servoing control scheme based on the catenary model of the tether and the motion of the tether attachment points.

Firstly, an interaction matrix related to the follower robot motion with regards to the catenary features variation is established. Preliminary results on tether shape control obtained with terrestrial robots are presented.

Secondly, keeping in mind that both leader and follower are equipped with vision sensors, it follows another symmetrically built interaction matrix related to the leader point of view. We will show that the fusion of both controllers allows to fully constrain the pair of robots. The catenary-based interaction matrices are then used to build different proportional control laws: simple stack of matrices and hierarchical control schemes. These controllers are tested in simulation.

Finally, the results are discussed and concluding remarks are given.

4.1 State of the Art on Vision-Based Control

In this thesis, we want to use the cameras embedded on the robots to estimate and control the shape of the tether. The use of visual information to control the pose of an object in the camera frame through the movements of a robot is part of the visual servoing framework of study (Shirai and Inoue, 1973; Weiss et al., 1987; Feddema and Mitchell, 1989; Chaumette and Hutchinson, 2006, 2007). In underwater applications, visual servoing has been used in a wide range of situations such as station keeping from the observation of landmarks in the sea bottom (van der Zwaan et al., 2002), alignment and tracking with respect to airplane black box (Prats et al., 2012), pipeline tracking (Krupinski et al., 2015), autonomous docking (Park et al., 2009) and valve tuning (Carreras et al., 2015).

In this section we will give details about some uses of visual servoing that can be applied to the pose control of rigid objects as well as to the shape control of deformable objects, which enters more specifically in our framework of study. More details about the visual servo control methodology will by given in Section 4.3, where the visual command aspects of the thesis are addressed.

4.1.1 Rigid Objects

The objective of vision-based control, or visual servo control, is to regulate the pose of a robot in order to reach 2-D desired image features or 3-D desired geometrical features (Espiau et al., 1992; Chaumette and Hutchinson, 2006, 2007). This is achieved by minimizing an error function typically defined by

$$\mathbf{e}(t) = \mathbf{s}(\mathbf{m}(t), \mathbf{a}) - \mathbf{s}^*. \tag{4.1}$$

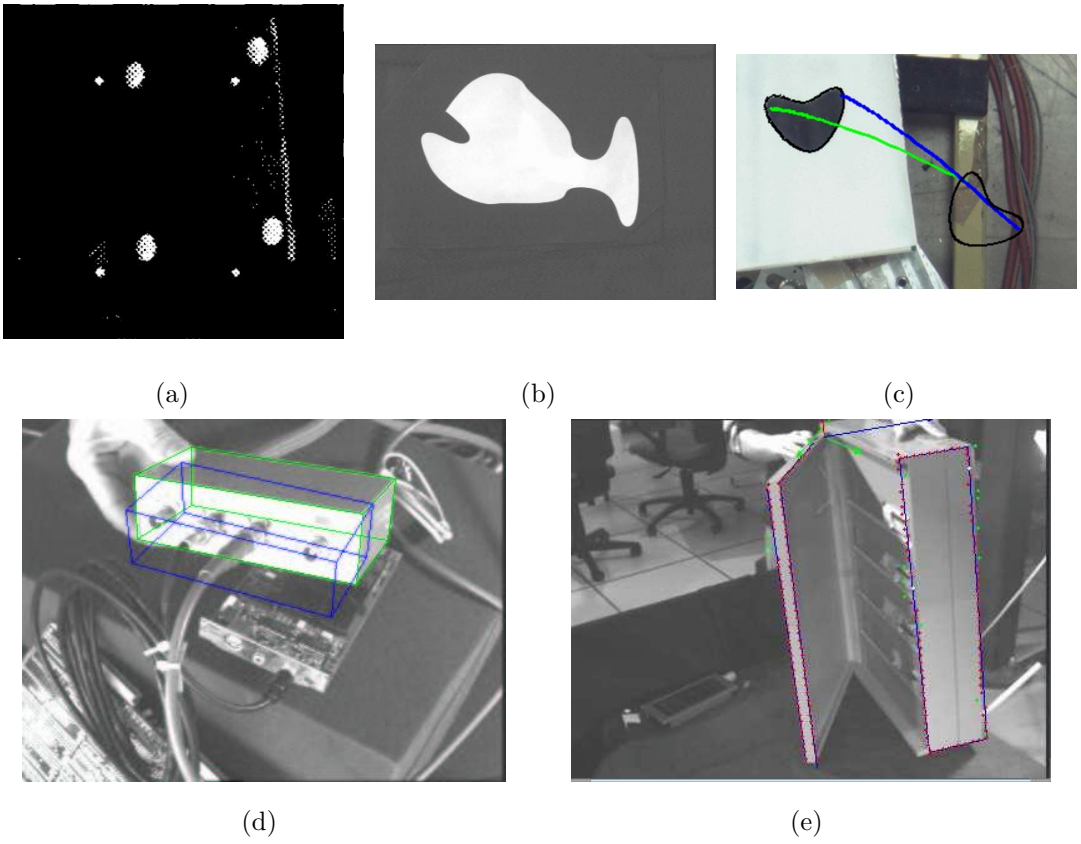
The vector $\mathbf{m}(t)$ contains some measurements made on the image (e.g. the tether detected points) while the vector $\mathbf{a}$ contains some additional knowledge about the system, as camera intrinsic parameters and 3-D models of objects in the scene(Chaumette and Hutchinson, 2006). This information is used to compute the feature vector $\mathbf{s}$ whose desired values are stored in $\mathbf{s}^*$, which is supposed to be constant. The most straightforward approach to regulate the visual features towards the desired values is to design a velocity controller. In order to do so, it is necessary to know the relationship between the time variation of $\mathbf{s}$ and the camera velocity $\mathbf{v}_c$:

$$\dot{\mathbf{s}} = \mathbf{L}_s \mathbf{v}_c, \tag{4.2}$$

where $\mathbf{L}_s$ is called the interaction matrix.

Visual servoing is used to regulate a wide range of visual primitives (also called visual features). First works addressed the use of image features composed of a set of image point coordinates (see Figure 4.1a) which have to reach a particular value in the image in order to achieve the specified task (Allen et al., 1993; Feddema et al., 1991; Hashimoto, 1993; Papanikolopoulos et al., 1993). This approach was then generalized to more complex visual features defined upon the projection in the image of geometrical

primitives such as lines, spheres and cylinders (Chaumette, 1994). These works were concerned with known and simple objects. They assumed that the objects in the scene can be expressed with simple geometrical features (points, straight lines, ellipses etc.). The group of objects that these methods can be applied to is thus limited.



(a) (b) (c)

(d) (e)

Figure 4.1: Example of visual servo control on rigid and articulated objects. (a) an experiment of positioning in front of a square extracted from Espiau et al. (1992). Detected feature points in large dots and their desired positions are drawn in small white dots. (b) an example of an amorphous planar shape used to validate the use of image moments in visual servoing; extracted from Tahri and Chaumette (2003). (c) an example of shape alignment using a technique of contour fitting to estimate object feature points to be used in visual servoing (extracted from Yazicioglu et al. (2009)). The feature points trajectories are drawn in green and blue lines. (d) an example of visual servoing used to regulate the robot's pose with respect to a rigid object whose CAD model is known (from Comport et al. (2004)). (e) another similar example but applied to articulated objects (from Comport et al. (2007)).

Positioning tasks through 2-D visual servoing with respect to objects of unknown shape was achieved in Collewet and Chaumette (2002) and Dune (2009). In Collewet and Chaumette (2002), the object was assumed to be planar and motionless and the

proposed method was based on the use of points as visual features. The object was modeled by a triangle formed by three points extracted from the image. The task was then executed in two steps: first, the object is maintained in the camera field of view and then the positioning task is performed so that the triangle base length and height are maximized and the object desired position is achieved. The image moments were also investigated in visual servoing positioning tasks with respect to planar objects of unknown shape (Tahri and Chaumette, 2003), as presented in Figure 4.1b. A method to determine the analytical form of the interaction matrix relating the camera velocity to any image moment time variation was presented in Chaumette (2004). The visual features were selected by combinations of moments and presented nice decoupling and stability properties. However, this visual servoing scheme is adequate in cases where it is possible to consider that the object is planar and that the object and the camera plane are parallel at the desired position. The obtained behavior was satisfactory for small displacements, but could be unsatisfactory for large displacements around the camera axes Chaumette (2004).

More recently, algebraic curves were also applied to shape alignment tasks in visual servo control (Yazicioglu et al., 2009), as it is illustrated in Figure 4.1c. The boundary of planar objects of unknown shape were modeled by algebraic equations obtained through the regularized 3L fitting algorithm (Sahin and Unel, 2004). These algebraic equations were then decomposed into a unique sum of product of lines, whose intersection were used as feature points in a visual servoing control scheme.

If the object is known, then its model can be used to perform positioning tasks (see an example in Figure 4.1d). This model can be a CAD (Computer-Aided Design) model (Lowe, 1992; Gennery, 1992; Comport et al., 2004) or a 2D template of the object (Kervrann and Heitz, 1998). In Marchand et al. (1999), polyhedral models were used to approximate the shape of known objects. A visual tracking based on the estimation, between two successive images, of a 2D global affine transformation was used to maintain the object in sight and the coordinates of the model corners where used as features in the visual servoing control scheme. This model-based approach evolved and was also applied to more complex, articulated objects (Nickels and Hutchinson, 2001; Comport et al., 2007) as presented in Figure 4.1e.

The visual servo control methods presented above were applied to the control of the robot's pose with respect to rigid or articulated objects that were not attached to the robot and could move independently in the space. Therefore, such approaches seem not to be adapted to tether handling, where the cable itself is used as a visual feature, since the tether is attached to the robot and its 3-D shape changes with the robot motion. This type of visual servoed task where the object deformation is taken into account is addressed in the next subsection.

4.1.2 Deformable Objects

The feedback control of soft object deformations has many applications in a wide range of fields, such as medical robotics for the manipulation of soft tissues (Mallapragada et al., 2011; Torabi et al., 2009), the automated food processing for shaping compliant

food materials (Wang and Hirai, 2011; Pettersson et al., 2011), 3C (Computers, Consumer Electronics and Communications) industry for cable positioning (Wakamatsu et al., 2006; Bretl and McCarthy, 2014), and the garment industry for folding fabrics (Bell, 2010). To servo-control deformations, a robotic system must continuously measure the object's configuration, typically with a vision sensor, and use its feedback data to compute a trajectory that deforms the object into a desired shape. This type of manipulation task aims to achieve a desired object shape, an approach that contrasts with standard visual servoing on rigid objects, where the servo loop is formulated in terms of the robot's desired pose with respect to the object.

Contrary to research studies on visual servoing of rigid objects, there does not exist many research work on visual servoing on deformable objects. Actually, some shape parametrization of deformable objects is required in order to achieve visual servoing, which can be a difficult challenge.

The active deformation control of compliant objects by robot manipulators was addressed in Navarro-Alarcon et al. (2013a) and Navarro-Alarcon et al. (2013b), where geometric visual features such as points, angles and curvatures were used to automatically deform soft objects toward a desired target shape (see Figure 4.2a). Only objects that exhibit or can be modeled based on purely elastic deformation were considered. A feature vector that expresses the desired deformation (a desired angle or curvature) is built from the observation of points of interest of the object. The time derivative of this feature vector is then related to the manipulator end-effector velocity by a deformation Jacobian matrix. A velocity controller was thus designed in order to bring the object to the desired state of deformation.

Since the proposed method was used to handle objects of unknown shape in a model-free approach, the deformation Jacobian matrix is unknown. It is numerically computed through the Broyden update rule (Chaumette and Hutchinson, 2007) at the control algorithm initialization and at each loop iteration. Besides the fact that the proposed method is only adapted to elastic deformation, other limitations were highlighted. The online estimation method presented requires slow motion of the robot manipulators, low-pass filtering of the observation points, and small control gains in order to provide smooth deformation trajectories.

An improved version of this controller was proposed in Navarro-Alarcon et al. (2014), that combined a minimum of offline information with online adaptation of deformation parameters. An analytical solution for the deformation Jacobian matrix was also proposed, which contrasts with the previous purely numeric algorithm. Moreover, the earlier approach only considers the affine camera model, whereas in this updated version the authors presented the complete solution to estimate the deformation Jacobian matrix with perspective camera model.

An innovative representation of the object's shape was introduced in Navarro-Alarcon and Liu (2018), where the closed-loop deformation task was based on a truncated Fourier series of 2-D image contours (see Figure 4.2b).

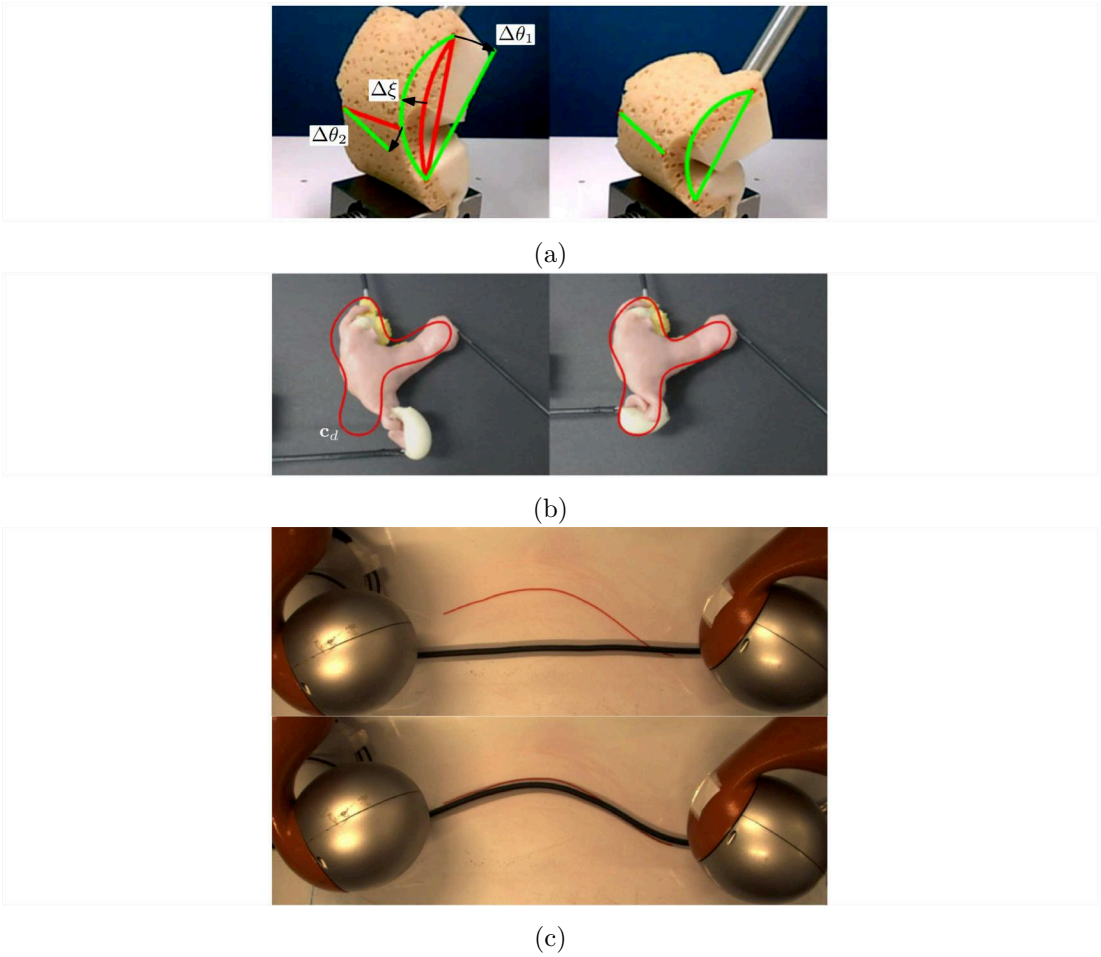


Figure 4.2: Some examples of visual servoing on deformable objects. (a) shape control of a sponge with current and desired shape feature drawn in red and green, respectively. Initial shape on the left-hand side and final shape on the right-hand side. Extracted from Navarro-Alarcon et al. (2013b). (b) manipulation of an uterus model though visual feedback where the object contour is modeled by a truncated Fourier series. The desired contour is drawn in red and the figure was extracted from Navarro-Alarcon and Liu (2018). (c) the same technique of shape modeling through Fourier series applied to 1-D objects. Image extracted from Zhu et al. (2018).

This approach contrasts with previous methods that typically use the displacements of feature points to define the deformation task (reach a desired angle or curvature). The controller design relied on an explicit shape servo loop by feeding back a vector of Fourier coefficients. This design provided more flexibility for the description of the feedback shape and its target configuration (the desired contour can be roughly sketched on a screen), and allowed the manipulation of soft objects with unstructured surfaces without special fiducial markers. To avoid the full parametric identification of the object's model, an online algorithm that approximates the deformation properties was

proposed. This algorithm iteratively re-calibrates an approximated linear deformation model computed from multiple data points collected along the local trajectory. The presented method has some limitations. First, the segmentation of the manipulated object may be complicated to perform for scenes with poor illumination, similar background/object colors or blurred edges. Similarly to other model-free approaches, this method is unable to determine whether a final target shape is physically reachable or not before the task is performed. Although the method does not require to use fiducial markers, it does need to observe one clearly distinguishable feature point to define the origin of the feedback contour (the authors used the position of the static gripper). It must be also noticed that since the estimator computes a linearized model that is valid only locally, a single approximation cannot, in general, be used to model the whole properties of the object. Finally, although the design of the robot motion controller is not restricted to plane motions, the proposed method can only control 2-D image projections of the observed shapes. An application to linear deformation on flexible cables was recently reported in Zhu et al. (2018). An example is depicted in Figure 4.2c.

4.2 General Control Scheme

In this thesis, we assume that the robots are horizontal, i.e., with a regulation of pitch and roll to zero. We assume that the roll and pitch angles are regulated by a low level controller or mechanically stabilized. These degrees of freedom will be thus excluded from the visual servoing control scheme. The general scheme of the vision-based tether shape control proposed in this thesis is depicted in Figure 4.3.

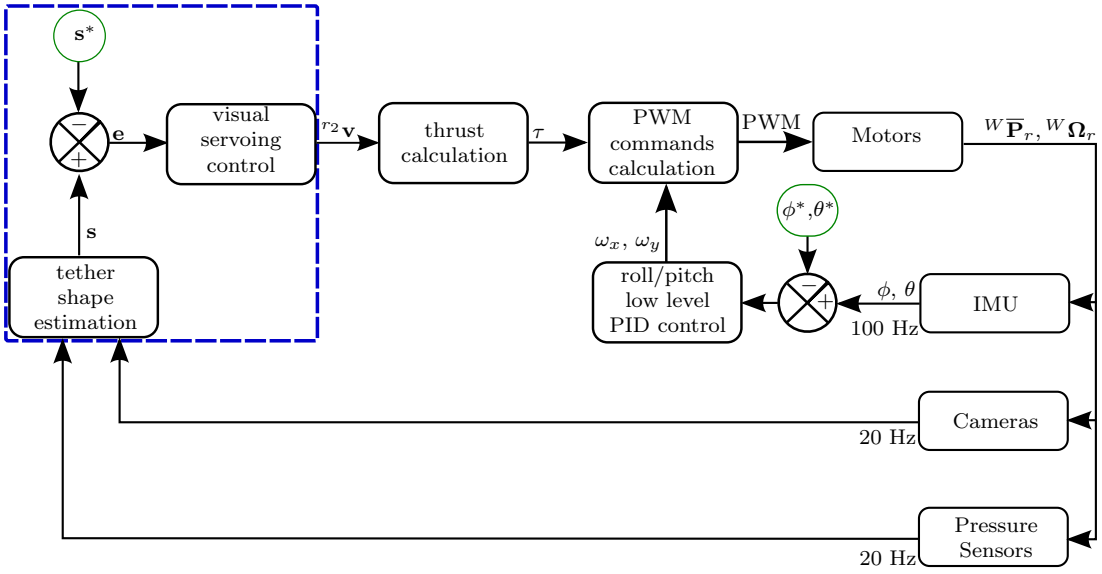


Figure 4.3: Chart flow of the vision-based tether shape control scheme. Details about the visual servoing control loop (blue frame) are given in Figure 4.4.

The objective of this control scheme is to regulate the tether features given by the feature vector $\mathbf{s}$. First, the error vector $\mathbf{e}$ between the desired ($\mathbf{s}^*$) and estimated ($\mathbf{s}$) feature vectors is computed. This error vector is used as input to calculate the follower robot velocities given in vector ${}^{r_2}\mathbf{v}$ that will regulate the tether shape. The robot thrust and then the PWM (Pulse Width Modulation) commands for each motor are calculated, as explained in Chapter 2. A low level control is in charge of regulating roll and pitch angles at high frequency from data provided by the IMU (inertial measurement unit). Once the thrusts are applied, new sensory data is acquired in order to calculate new velocity commands. The depth sensors provide information about the difference of height between the tether attachment points, whereas camera feedback provides tether detected points used for tether shape estimation through image fitting techniques, as explained in Chapter 3.

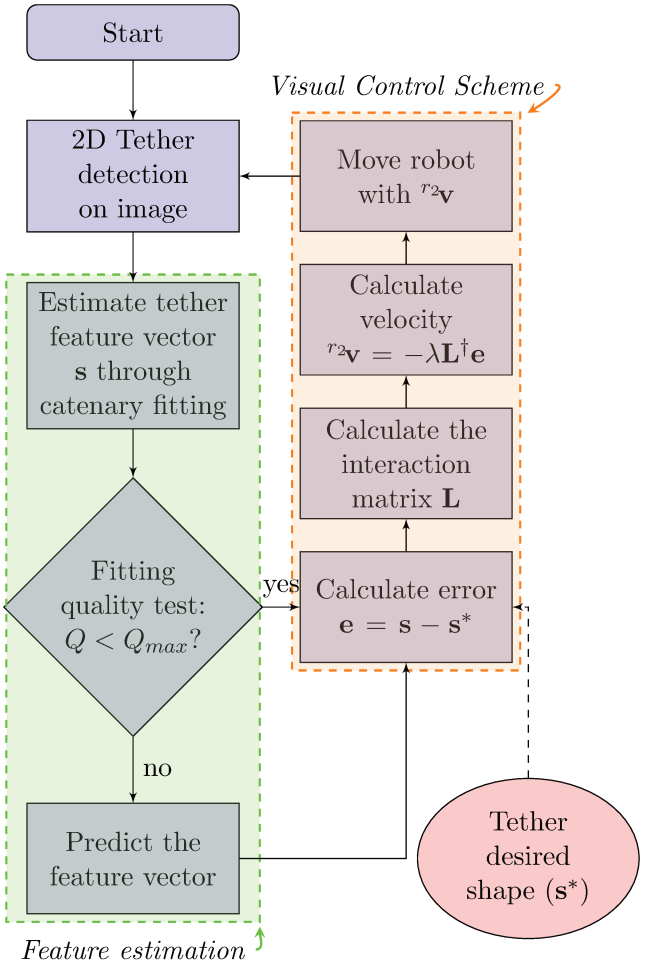


Figure 4.4: Algorithm chart flow for catenary-based visual servoing.

The detailed steps of the visual servoing control loop are given in Figure 4.4. First, the tether detected points in the image are used to estimate the tether shape through

a catenary fitting algorithm. Then, the cost function value is used to calculate a fitting quality index. If the shape estimation is not accurate, the tether features are predicted from the previous command velocities and interaction matrix evaluation. If the shape estimation is enough accurate, the error vector between current and desired features are computed. Next, the interaction matrix and the new command velocities are calculated before sending them to the embedded micro-controller. The calculation of the interaction matrix is detailed in the following Section.

4.3 Catenary-Based Interaction Matrices

The objective of this Section is to express the velocities of the follower robot as a function of features extracted from the catenary model of the tether, i.e., its slackness, orientation, and height difference between attachment points. As explained in previous Section, the tether slackness and orientation are estimated thanks to image fitting techniques whereas the attachment points height difference is calculated from depth sensors feedback.

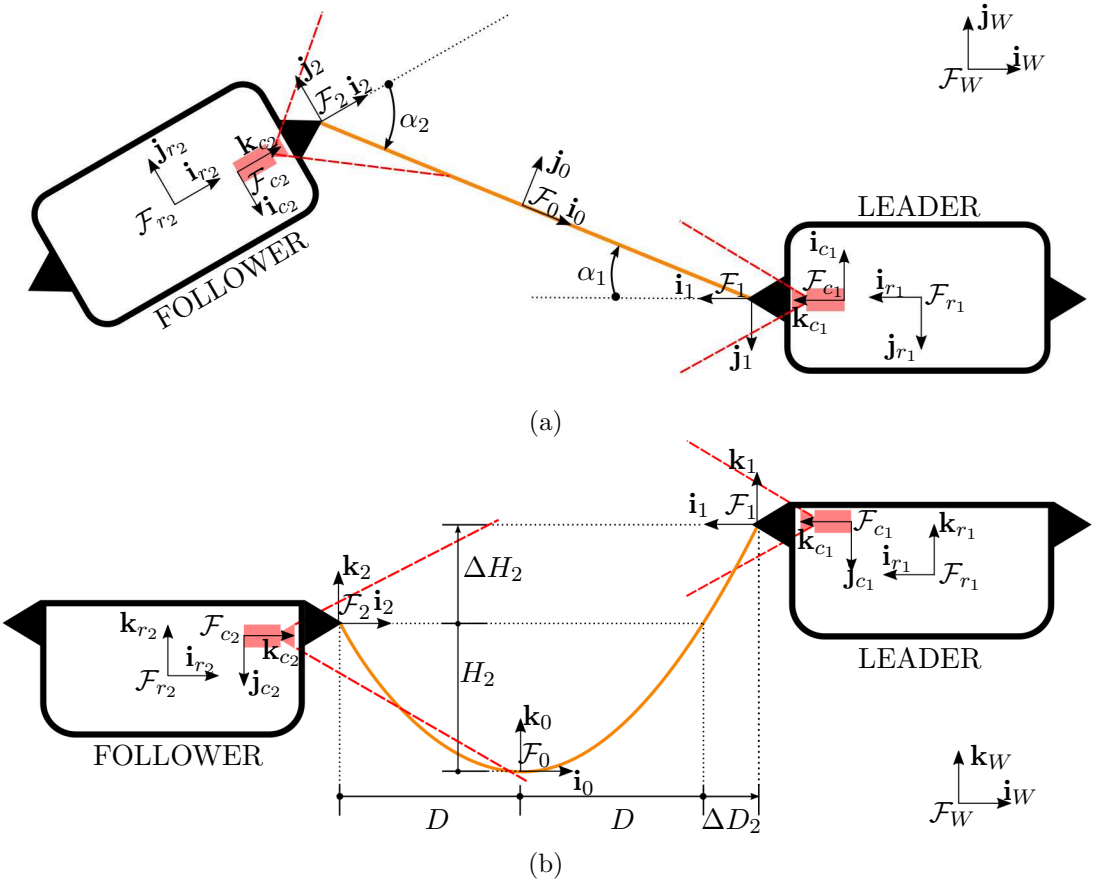


Figure 4.5: Figure 3.1 is reproduced to remind the features definition. (a) top view and (b) side view.

The catenary features (see Figure 4.5) from the follower robot side, as introduced in Chapter 3 in equations (3.34a) and (3.34b), are noted

$$a_2 = \frac{H_2}{H_{max}} \tag{4.3}$$

$$b_2 = \sin \alpha_2, \tag{4.4}$$

and range in the interval $]0, 1[$. In addition to these features, we add a third one related to ΔH_2:

$$d_2 = \frac{\Delta H_2 + \Delta H_{max}}{2\Delta H_{max}} \tag{4.5}$$

This new feature is also normalized and ranges in the interval $]0, 1[$. If $\mathbf{P}_1$ is above $\mathbf{P}_2$, then $d_2 \in [0.5, 1]$ else $d_2 \in [0, 0.5]$, which means that $\Delta H_2 \in [-\Delta H_{max}, \Delta H_{max}]$.

We then define the feature vector as:

$$\mathbf{s}_2 = \begin{bmatrix} a_2 \\ b_2 \\ d_2 \end{bmatrix}. \tag{4.6}$$

Similarly, from the leader robot side, we can set the feature vector as:

$$\mathbf{s}_1 = \begin{bmatrix} a_1 \\ b_1 \\ d_1 \end{bmatrix}, \tag{4.7}$$

where $a_1 = \frac{H_1}{H_{max}}$, $b_1 = \sin \alpha_1$ and $d_1 = \frac{\Delta H_1 + \Delta H_{max}}{2\Delta H_{max}}$.

From equations (2.87) and (2.88) in Chapter 2, we have the relation between the robots velocities and the variation of the tether attachment points relative position:

$$^2\dot{\overline{\mathbf{P}}}_1 = \begin{bmatrix} ^2\dot{X}_1 \\ ^2\dot{Y}_1 \\ ^2\dot{Z}_1 \end{bmatrix} = \mathbf{T}_{r_1,2}{}^{r_1}\mathbf{v} + \mathbf{T}_{r_2,2}{}^{r_2}\mathbf{v} \tag{4.8}$$

$$^1\dot{\overline{\mathbf{P}}}_2 = \begin{bmatrix} ^1\dot{X}_2 \\ ^1\dot{Y}_2 \\ ^1\dot{Z}_2 \end{bmatrix} = \mathbf{T}_{r_1,1}{}^{r_1}\mathbf{v} + \mathbf{T}_{r_2,1}{}^{r_2}\mathbf{v}.$$

Now, we need to find the relationship between the feature vector time derivative and the coordinates of the attachment points time derivative. We will conduct the calculation from the follower side using $\dot{\mathbf{s}}_2 = \left[\dot{a}_2, \dot{b}_2, \dot{d}_2\right]^T$ and $^2\dot{\overline{\mathbf{P}}}_1 = \left[^2\dot{X}_1, {}^2\dot{Y}_1, {}^2\dot{Z}_1\right]^T$. The expression regarding the leader robot side will be obtained symmetrically.

We start the calculation with

$$^2\overline{\mathbf{P}}_1 = \begin{bmatrix} ^2X_1 \\ ^2Y_1 \\ ^2Z_1 \end{bmatrix} = \begin{bmatrix} (2D_2 + \Delta D_2)\cos\alpha_2 \\ (2D_2 + \Delta D_2)\sin\alpha_2 \\ \Delta H_2 \end{bmatrix}_{\mathcal{F}_2} = \begin{bmatrix} (2D_2 + \Delta D_2)\sqrt{1 - b_2^2} \\ (2D_2 + \Delta D_2)b_2 \\ 2d_2\Delta H_{max} - \Delta H_{max} \end{bmatrix}_{\mathcal{F}_2}. \tag{4.9}$$

We know from Figure 2.10 that:

$$\sin\alpha_2 = b_2 = \frac{{}^2Y_1}{2D_2 + \Delta D_2} \tag{4.10}$$

$$\cos\alpha_2 = \sqrt{1 - b_2^2} = \frac{{}^2X_1}{2D_2 + \Delta D_2}. \tag{4.11}$$

The values of a_2 and b_2 are obtained from the vision system, and d_2 from the depth sensors. The value of D_2 is obtained thanks to the following equation:

$$D_2 = \frac{1}{C}\cosh^{-1}\left(Ca_2H_{max} + 1\right), \tag{4.12}$$

which states that $\mathbf{P}_2$ belongs to the catenary, as introduced by equation (2.13) in Section 2.1.2.

The value of $D_2 + \Delta D_2$ is obtained thanks to the equation:

$$D_2 + \Delta D_2 = \frac{1}{C}\cosh^{-1}\left(C\left(a_2H_{max} + 2d_2\Delta H_{max} - \Delta H_{max}\right) + 1\right), \tag{4.13}$$

which states that $\mathbf{P}_1$ belongs to the catenary, as introduced by equation (2.14).

Let us differentiate equation 4.9:

$${}^2\dot{\overline{\mathbf{P}}}_1 = \begin{bmatrix} \left[\dot{D}_2 + (\dot{D}_2 + \dot{\Delta D}_2)\right]\sqrt{1-b_2^2} - (2D_2 + \Delta D_2)\frac{b_2\dot{b}_2}{\sqrt{1-b_2^2}} \\ \left[\dot{D}_2 + (\dot{D}_2 + \dot{\Delta D}_2)\right] b_2 + (2D_2 + \Delta D_2)\dot{b}_2 \\ 2\dot{d}_2\Delta H_{max} \end{bmatrix}_{\mathcal{F}_2} \tag{4.14}$$

$$= \begin{bmatrix} \left[\dot{D}_2 + (\dot{D}_2 + \dot{\Delta D}_2)\right]\frac{{}^2X_1}{2D_2+\Delta D_2} - {}^2Y_1\frac{\dot{b}_2}{\sqrt{1-b_2^2}} \\ \left[\dot{D}_2 + (\dot{D}_2 + \dot{\Delta D}_2)\right]\frac{{}^2Y_1}{2D_2+\Delta D_2} + (2D_2 + \Delta D_2)\dot{b}_2 \\ 2\dot{d}_2\Delta H_{max} \end{bmatrix}_{\mathcal{F}_2} \tag{4.15}$$

We then need to explicit the derivative of the features a_2, b_2 and d_2 as function of ${}^2\dot{X}_1, {}^2\dot{Y}_1$ and ${}^2\dot{Z}_1$.

Expression of $\dot{d}_2$ as function of ${}^2\dot{X}_1, {}^2\dot{Y}_1$ and ${}^2\dot{Z}_1$

From equation (4.5), we directly have:

$$\boxed{\dot{d}_2 = \frac{\Delta\dot{H}_2}{2\Delta H_{max}} = \frac{{}^2\dot{Z}_1}{2\Delta H_{max}}} \tag{4.16}$$

Expression of $\dot{b}_2$ as function of ${}^2\dot{X}_1, {}^2\dot{Y}_1$ and ${}^2\dot{Z}_1$

From the first two lines of the matrix given in equation (4.15) we get:

$${}^2\dot{X}_1 = (\dot{D}_2 + (\dot{D}_2 + \dot{\Delta D}_2))\sqrt{1-b_2^2} - (2D_2 + \Delta D_2)\frac{b_2\dot{b}_2}{\sqrt{1-b_2^2}} \tag{4.17}$$

$${}^2\dot{Y}_1 = (\dot{D}_2 + (\dot{D}_2 + \dot{\Delta D}_2))b_2 + (2D_2 + \Delta D_2)\dot{b}_2 \tag{4.18}$$

which leads to:

$$2\dot{D}_2 + \dot{\Delta D}_2 = \sqrt{1-b_2^2}\,{}^2\dot{X}_1 + b_2\,{}^2\dot{Y}_1 = \frac{{}^2X_1\,{}^2\dot{X}_1 + {}^2Y_1\,{}^2\dot{Y}_1}{2D_2 + \Delta D_2} \tag{4.19}$$

$$\boxed{\dot{b}_2 = \frac{\sqrt{1-b_2^2}}{2D_2 + \Delta D_2}\left[-b_2\,{}^2\dot{X}_1 + \sqrt{1-b_2^2}\,{}^2\dot{Y}_1\right]} \tag{4.20}$$

Expression of $\dot{a}_2$ as function of ${}^2\dot{X}_1, {}^2\dot{Y}_1$ and ${}^2\dot{Z}_1$

From the definition of parameter a_2 in equation (4.3) we obtain:

$$\dot{a}_2 = \frac{\dot{H}_2}{H_{max}} \tag{4.21}$$

The equation (4.19) is useful to get $\dot{H}_2$ as a function of the coordinates derivatives ${}^2\dot{X}_1, {}^2\dot{Y}_1$ and ${}^2\dot{Z}_1$ because $\dot{D}_2$ and $\dot{\Delta D}_2$ depend on H_2 and ΔH_2.

From equation (4.12) we have

$$CH_2 = \left[\cosh(CD_2) - 1\right]. \tag{4.22}$$

The differentiation of this equation yields:

$$\dot{D}_2 = p_2\dot{H}_2 + q_2\dot{\Delta H}_2 \tag{4.23}$$

with

$$p_2 = \frac{C + L_{D_2}\frac{\partial C}{\partial H_2}}{C\sinh(CD_2)} \tag{4.24}$$

$$q_2 = \frac{L_{D_2}\frac{\partial C}{\partial \Delta H_2}}{C\sinh(CD_2)} \tag{4.25}$$

$$L_{D_2} = H_2 - D_2\sinh(CD_2) \tag{4.26}$$

From equation (4.13) we have:

$$C\left(H_2 + \Delta H_2\right) = \left[\cosh(C(D_2 + \Delta D_2) - 1\right]. \tag{4.27}$$

The differentiation of this equation yields:

$$\dot{D}_2 + \dot{\Delta D}_2 = u_2\dot{H}_2 + v_2\dot{\Delta H}_2 \tag{4.28}$$

with:

$$u_2 = \frac{C + L_{\Delta D_2}\frac{\partial C}{\partial H_2}}{C\sinh(C(D_2 + \Delta D_2))} \tag{4.29}$$

$$v_2 = \frac{C + L_{\Delta D_2}\frac{\partial C}{\partial \Delta H_2}}{C\sinh(C(D_2 + \Delta D_2))} \tag{4.30}$$

$$L_{\Delta D_2} = H_2 + \Delta H_2 - (D_2 + \Delta D_2)\sinh(C(D_2 + \Delta D_2)) \tag{4.31}$$

The partial derivatives of the catenary C parameter are calculated from differentiation of equation (2.16):

$$\dot{C}(H_2, \Delta H_2) = \frac{\partial C}{\partial H_2}\dot{H}_2 + \frac{\partial C}{\partial \Delta H_2}\dot{\Delta H}_2. \tag{4.32}$$

The expression of the partial derivatives $\frac{\partial C}{\partial H_2}$ and $\frac{\partial C}{\partial \Delta H_2}$ are given in appendix D.

Finally, combining equations (4.19), 4.21, (4.23) and (4.28) leads to:

$$2\dot{D}_2 + \dot{\Delta D}_2 = (u_2 + p_2)\dot{H}_2 + (v_2 + q_2)\dot{\Delta H}_2 \tag{4.33}$$

$$\boxed{\dot{a}_2 = \frac{1}{(u_2 + p_2)H_{max}}\left(\frac{{}^2X_1\,{}^2\dot{X}_1 + {}^2Y_1\,{}^2\dot{Y}_1}{2D_2 + \Delta D_2} - (v_2 + q_2)\,{}^2\dot{Z}_1\right)} \tag{4.34}$$

Catenary Interaction Matrix

In summary, we can write equations (4.34), (4.20) and (4.16) as:

$$\dot{a}_2 = \frac{1}{(u_2 + p_2)H_{max}}\begin{bmatrix}\sqrt{1 - b_2^2}, & b_2, & -(v_2 + q_2)\end{bmatrix}\begin{bmatrix}{}^2\dot{X}_1 \\ {}^2\dot{Y}_1 \\ {}^2\dot{Z}_1\end{bmatrix} \tag{4.35}$$

$$\dot{b}_2 = \frac{\sqrt{1 - b_2^2}}{2D_2 + \Delta D_2}\begin{bmatrix}-b_2, & \sqrt{1 - b_2^2}, & 0\end{bmatrix}\begin{bmatrix}{}^2\dot{X}_1 \\ {}^2\dot{Y}_1 \\ {}^2\dot{Z}_1\end{bmatrix} \tag{4.36}$$

$$\dot{d}_2 = \frac{1}{2\Delta H_{max}}\begin{bmatrix}0 & 0 & 1\end{bmatrix}\begin{bmatrix}{}^2\dot{X}_1 \\ {}^2\dot{Y}_1 \\ {}^2\dot{Z}_1\end{bmatrix}, \tag{4.37}$$

where the expressions of b_2 and $\sqrt{1 - b_2^2}$ were given in equations (4.10) and (4.11).

Rewriting equations (4.35), (4.36) and (4.37) in matrix form yields:

$$\begin{bmatrix}\dot{a}_2 \\ \dot{b}_2 \\ \dot{d}_2\end{bmatrix} = \mathbf{M}_2\begin{bmatrix}{}^2\dot{X}_1 \\ {}^2\dot{Y}_1 \\ {}^2\dot{Z}_1\end{bmatrix} \tag{4.38}$$

with

$$\mathbf{M}_2 = \begin{bmatrix}\frac{\sqrt{1-b_2^2}}{(u_2+p_2)H_{max}} & \frac{b_2}{(u_2+p_2)H_{max}} & -\frac{v_2+q_2}{(u_2+p_2)H_{max}} \\ \frac{-b_2\sqrt{1-b_2^2}}{2D_2+\Delta D_2} & \frac{1-b_2^2}{2D_2+\Delta D_2} & 0 \\ 0 & 0 & \frac{1}{2\Delta H_{max}}\end{bmatrix}$$

Introducing equation (2.87) into (4.38) we obtain:

$$\dot{\mathbf{s}}_2 = \begin{bmatrix}\dot{a}_2 \\ \dot{b}_2 \\ \dot{d}_2\end{bmatrix} = \mathbf{L}_{r_1,2}\,{}^{r_1}\mathbf{v} + \mathbf{L}_{r_2,2}\,{}^{r_2}\mathbf{v}, \tag{4.39}$$

with

$$\mathbf{L}_{r_1,2} = \mathbf{M}_2 \mathbf{T}_{r_1,2}$$
$$\mathbf{L}_{r_2,2} = \mathbf{M}_2 \mathbf{T}_{r_2,2}.$$

Using the twist matrix (see appendix C), we have:

$$\mathbf{L}_{r_1,2} = -\mathbf{M}_2 \mathbf{T}_{r_2,2} {}^{r_2}\mathbf{V}_{r_1}$$
$$\mathbf{L}_{r_1,2} = -\mathbf{L}_{r_2,2} {}^{r_2}\mathbf{V}_{r_1},$$

which allows us to rewrite equation (4.39) as follows:

$$\dot{\mathbf{s}}_2 = \begin{bmatrix} \dot{a}_2 \\ \dot{b}_2 \\ \dot{d}_2 \end{bmatrix} = \mathbf{L}_{r_2,2} \left({}^{r_2}\mathbf{v} - {}^{r_2}\mathbf{V}_{r_1} {}^{r_1}\mathbf{v} \right), \tag{4.40}$$

The matrices $\mathbf{L}_{r_1,2}$ and $\mathbf{L}_{r_2,2}$ are called interaction matrices and link, respectively, the velocities of leader (r_1) and follower (r_2) robots with the time derivative of the feature vector $\mathbf{s}_2$. Symmetrically, we have:

$$\dot{\mathbf{s}}_1 = \begin{bmatrix} \dot{a}_1 \\ \dot{b}_1 \\ \dot{d}_1 \end{bmatrix} = \mathbf{L}_{r_2,1} {}^{r_2}\mathbf{v} + \mathbf{L}_{r_1,1} {}^{r_1}\mathbf{v} = \mathbf{L}_{r_2,1} \left({}^{r_2}\mathbf{v} - {}^{r_2}\mathbf{V}_{r_1} {}^{r_1}\mathbf{v} \right) \tag{4.41}$$

with

$$\mathbf{L}_{r_2,1} = \mathbf{M}_1 \mathbf{T}_{r_2,1}$$
$$\mathbf{L}_{r_1,1} = \mathbf{M}_1 \mathbf{T}_{r_1,1}$$
$$\mathbf{L}_{r_1,1} = -\mathbf{M}_1 \mathbf{T}_{r_2,1} {}^{r_2}\mathbf{V}_{r_1} = -\mathbf{L}_{r_2,1} {}^{r_2}\mathbf{V}_{r_1}$$

Generalizing equations (4.39) and (4.41), we can write, for $i \in \{1, 2\}$, that:

$$\dot{\mathbf{s}}_i = \begin{bmatrix} \dot{a}_i \\ \dot{b}_i \\ \dot{d}_i \end{bmatrix} = \mathbf{L}_{r_2,i} {}^{r_2}\mathbf{v} + \mathbf{L}_{r_1,i} {}^{r_1}\mathbf{v} \tag{4.42}$$

with

$$\mathbf{L}_{r_2,i} = \mathbf{M}_i \mathbf{T}_{r_2,i} \tag{4.43}$$
$$\mathbf{L}_{r_1,i} = \mathbf{M}_i \mathbf{T}_{r_1,i}$$

and

$$\mathbf{M}_i = \begin{bmatrix} \frac{\sqrt{1-b_i^2}}{(u_i+p_i)H_{max}} & \frac{b_i}{(u_i+p_i)H_{max}} & -\frac{v_i+q_i}{(u_i+p_i)H_{max}} \\ \frac{-b_i\sqrt{1-b_i^2}}{2D_i+\Delta D_i} & \frac{1-b_i^2}{2D_i+\Delta D_i} & 0 \\ 0 & 0 & \frac{1}{2\Delta H_{max}} \end{bmatrix}$$

Interaction Matrices for 4 Degrees of Freedom

In order to avoid the control of roll and pitch rotations through visual servoing, but rather to let a low level control scheme regulate them, we choose to remove these degrees of freedom from the interaction matrix.

As a matter of fact, the calculation of the pseudo-inverse of the complete interaction matrix involves a minimization of the norm of ${}^{r_2}\mathbf{v} = [{}^{r_2}\nu_x, {}^{r_2}\nu_y, {}^{r_2}\nu_2, {}^{r_2}\omega_x, {}^{r_2}\omega_y, {}^{r_2}\omega_z]^T$, and of the norm of $\|\mathbf{s}_2 - \mathbf{L}_{r2,2}{}^{r_2}\mathbf{v}\|$, which can generate values for ω_x and ω_y that may disturb the low level control used for the stabilization of the vehicle. Therefore, instead of considering all the degrees of freedom of the robot, we can use a trimmed interaction matrix:

$$\begin{bmatrix}\dot{a}_2\\ \dot{b}_2\\ \dot{d}_2\end{bmatrix} = \begin{bmatrix}\frac{\sqrt{1-b_2^2}}{(u_2+p_2)H_{max}} & \frac{b_2}{(u_2+p_2)H_{max}} & -\frac{v_2+q_2}{(u_2+p_2)H_{max}}\\ \frac{-b_2\sqrt{1-b_2^2}}{2D_2+\Delta D_2} & \frac{1-b_2^2}{2D_2+\Delta D_2} & 0\\ 0 & 0 & \frac{1}{2\Delta H_{max}}\end{bmatrix}\begin{bmatrix}-1 & 0 & 0 & {}^{r_2}Y_1\\ 0 & -1 & 0 & -{}^{r_2}X_1\\ 0 & 0 & -1 & 0\end{bmatrix}\begin{bmatrix}{}^{r_2}\nu_x\\ {}^{r_2}\nu_y\\ {}^{r_2}\nu_z\\ {}^{r_2}\omega_z\end{bmatrix}$$

with the new interaction matrix $\mathbf{L}_{r2,2}^{3\times4}$, excluding the roll and pitch of the robot:

$$\mathbf{L}_{r2,2}^{3\times4} = \begin{bmatrix}\frac{\sqrt{1-b_2^2}}{(u_2+p_2)H_{max}} & \frac{b_2}{(u_2+p_2)H_{max}} & -\frac{v_2+q_2}{(u_2+p_2)H_{max}}\\ \frac{-b_2\sqrt{1-b_2^2}}{2D_2+\Delta D_2} & \frac{1-b_2^2}{2D_2+\Delta D_2} & 0\\ 0 & 0 & \frac{1}{2\Delta H_{max}}\end{bmatrix}\begin{bmatrix}-1 & 0 & 0 & {}^{r_2}Y_1\\ 0 & -1 & 0 & -{}^{r_2}X_1\\ 0 & 0 & -1 & 0\end{bmatrix}$$

that yields, in its developed form:

$$\mathbf{L}_{r2,2}^{3\times4} = \begin{bmatrix}-\frac{\sqrt{1-b_2^2}}{(u_2+p_2)H_{max}} & -\frac{b_2}{(u_2+p_2)H_{max}} & \frac{v_2+q_2}{(u_2+p_2)H_{max}} & {}^{r_2}Y_1\frac{\sqrt{1-b_2^2}}{(u_2+p_2)H_{max}} - {}^{r_2}X_1\frac{b_2}{(u_2+p_2)H_{max}}\\ \frac{b_2\sqrt{1-b_2^2}}{2D_2+\Delta D_2} & -\frac{1-b_2^2}{2D_2+\Delta D_2} & 0 & {}^{r_2}Y_1\frac{-b_2\sqrt{1-b_2^2}}{2D_2+\Delta D_2} - {}^{r_2}X_1\frac{1-b_2^2}{2D_2+\Delta D_2}\\ 0 & 0 & -\frac{1}{2\Delta H_{max}} & 0\end{bmatrix}. \tag{4.44}$$

In a similar way, we can calculate the matrices $\mathbf{L}_{r2,1}^{3\times4}$, $\mathbf{L}_{r1,2}^{3\times4}$ and $\mathbf{L}_{r1,1}^{3\times4}$, removing the roll and pitch degrees of freedom from matrices $\mathbf{T}_{r2,1}$, $\mathbf{T}_{r1,2}$ and $\mathbf{T}_{r1,1}$. A procedure of test for the interaction matrices is presented in appendix F.

Singularity on the Interaction Matrix

The maximum rank of matrix $\mathbf{L}_{r2,2}^{3\times4}$ is 3. The minimum rank of matrix $\mathbf{L}_{r2,2}^{3\times4}$ is 2, since there are at least two independent columns including the third one. The fourth column is a linear combination of the first and second columns. Therefore, the rank is reduced to 2 when the first and second columns are linked, which occurs when the determinant of the submatrix $\mathbf{L}_{r_2,2}^{3\times4}(1:2,1:2)$ is zero:

$$\frac{\sqrt{1-b_2^2}}{(u_2+p_2)H_{max}(2D_2+\Delta D_2)} = 0,$$

which corresponds to

$$1 - b^2 = 0. \tag{4.45}$$

The equation (4.45) represents the cases where $b = \sin\alpha = \pm 1$, thus $\alpha = \pm 90°$ and the tether plane is perpendicular to the robot longitudinal plane. In such cases, the first column of the interaction matrix is zero. This produces a loss of control of the longitudinal velocity. As discussed in Section 3.4.3.2, these cases will be avoided since the tether is less visible (limited camera field of view) and it risks of becoming entangled with the robot.

Analyzing the matrix $\mathbf{L}_{r_2,2}^{3\times 4}$ we note that it can be singular due to infinite values, if:

$$2D_2 + \Delta D_2 = 0 \tag{4.46}$$

$$u_2 + p_2 = 0 \tag{4.47}$$

The singularity given in equation (4.46) corresponds to a tether whose span is zero (attachments points on the same vertical), which is not possible.

The equation (4.47) corresponds to a singularity that never occurs because $u_2 + p_2 \neq 0$ and keeps a negative sign. If $u_2 + p_2$ were to become zero, it would mean that the span between the tether attachment points could be changed whatever the tether slackness, which is impossible (see equation (4.33)).

Another explanation can be given by looking at equation (4.33), and considering a fixed ΔH_2, which yields:

$$2\dot{D}_2 + \Delta\dot{D}_2 = (u_2 + p_2)\dot{H}_2.$$

If we extend the catenary span, $2D_2 + \Delta D_2$ increases, and the slackness H_2 decreases, which means that $2\dot{D}_2 + \Delta\dot{D}_2$ is positive and $\dot{H}_2$ is negative. Therefore, $(u_2 + p_2) < 0$. Conversely, if we reduce the catenary span, $2D_2 + \Delta D_2$ decreases and the slackness H_2 increases, which means that $2\dot{D}_2 + \Delta\dot{D}_2$ is negative and $\dot{H}_2$ is positive, also leading to $(u_2 + p_2) < 0$.

4.4 Follower Robot Control using Tether Visual Feedback

In this Section, we present the vision-based control scheme used to command the follower robot motion in order to regulate the tether shape. As introduced in Chapter 2, the leader robot is free to explore its surroundings, while the follower robot main task is to ensure, from its onboard camera feedback, that the tether cable does not hamper the leader movements. Therefore, the follower must maintain the cable slack enough not to pull the leader. Moreover, it must keep a desired orientation between tether and onboard camera in order to ensure that the tether is well visible in the image and that its shape can be accurately estimated from the curve fitting algorithm presented in Chapter 3. The tether attachment points difference of height must also be regulated, since it impacts the tether slackness and shape in the image.

Here, only the follower robot frontal camera is used in the visual servoing control scheme (both leader and follower cameras will be used in Section 4.5). The feature vector to be regulated from the follower robot motion is defined as:

$$\mathbf{s}_2 = \begin{bmatrix} a_2 \\ b_2 \\ d_2 \end{bmatrix} = \begin{bmatrix} \frac{H_2}{H_{max}} \\ \sin\alpha_2 \\ \frac{\Delta H_2 + \Delta H_{max}}{2\Delta H_{max}} \end{bmatrix}.$$

The error vector $\mathbf{e}_2$ between the current and the desired feature vector $\mathbf{s}_2^*$ is defined as:

$$\mathbf{e}_2 = \mathbf{s}_2 - \mathbf{s}_2^*. \tag{4.48}$$

The error vector time derivative is therefore:

$$\dot{\mathbf{e}}_2 = \dot{\mathbf{s}}_2. \tag{4.49}$$

As explained in the previous Section, the roll and pitch angles are assumed to be fixed thanks to a low level control or to a mechanical self-stabilization. We use hence 3 by 4 interaction matrices, removing the columns related to roll and pitch motions of the robot. For example, the interaction matrix $\mathbf{L}_{r_2,2}^{3\times4}$, given by the equation (4.44), is a 3 by 4 matrix that links the variations of the feature vector $\mathbf{s}_2$ to the follower robot velocities ${}^{r_2}\mathbf{v}^{3\times4} = [\nu_x, \nu_y, \nu_z, \omega_z]^T$. In order to simplify the writing, we will note from now on:

$$\begin{aligned}
\mathbf{L}_{r_2,2} &= \mathbf{L}_{r_2,2}^{3\times4} \\
\mathbf{L}_{r_1,2} &= \mathbf{L}_{r_2,2}^{3\times4} \\
\mathbf{L}_{r_2,1} &= \mathbf{L}_{r_2,2}^{3\times4} \\
\mathbf{L}_{r_1,1} &= \mathbf{L}_{r_2,2}^{3\times4} \\
{}^{r_2}\mathbf{v} &= {}^{r_2}\mathbf{v}^{3\times4} \\
{}^{r_1}\mathbf{v} &= {}^{r_1}\mathbf{v}^{3\times4}.
\end{aligned}$$

In the sequel, preliminary results with 2-D terrestrial robots are given. Then a number of tether shape control schemes for 3-D underwater robots are tested in simulation. Two types of situations are addressed. First, the leader robot is stopped and the

follower robot brings the tether from its initial to its desired shape. Second, the leader robot executes a predefined path and the follower robot must maintain the tether at its desired shape. The leader robot velocities may be considered or not in the follower robot control law.

The simulations are carried out on Gazebo (Koenig and Howard, 2004). In the case of underwater robots, the UUV (unmanned underwater vehicle) package (Manhães et al., 2016) is used, as described in the previous Chapter. The robots are equipped with simulated pressure sensors, frontal and rear cameras. The simulated tether is 1.50 meter long and 3 millimeter thick. The maximum slackness is set at $H_{max} = 0.70$ meter. The maximum attachment point difference of height is set at $\Delta H_{max} = 0.50$ meter. The control loop runs at a rate of 30 Hz.

4.4.1 Preliminary Results with Terrestrial Robots

In this Section we present a summary of our preliminary research results on tether shape control using terrestrial robots. More details are given in appendix E.

First, we introduce the work that was presented at the 2017 IEEE International Conference on Robotics and Automation (ICRA 2017), whose title is **Catenary-based Visual Servoing for Tethered Robots**. This work addressed the use of catenary-model-based features to achieve visual servoing of an umbilical linking two terrestrial robots.

Second, we introduce another work that compared catenary-based with image-based visual servoing controllers. More details are available in the article entitled **Tether Shape Control through Catenary-Model-Based Visual Servoing**, in appendix E.

4.4.1.1 Catenary-Based Visual Servoing for Terrestrial Robots

The new visual servoing scheme introduced for catenary shaped deformable objects was tested on terrestrial robots to control the tether parametric shape by properly moving its fixation point.

The interaction matrix is a full 2×6 matrix (see equation (23) of the corresponding paper in appendix E), assuming that the tether attachment points are at the same height. Only the features from the follower camera image view were taken into account.

The experimental system is composed of two terrestrial mobile robots of the same motion capabilities linked with a slack rope.

One simulation and two real experiments were carried out in order to validate the control law with two Turtlebots (Gerkey and Conley, 2011) with 2 DOF (translation v_x and rotation w_z). They were equipped with an onboard RGBD camera and linked by a 1.4 meter long and 3 millimeter thick orange rope. The video frame rate was fixed to 10Hz. The same robot motion capabilities, rope length, thickness and maximum sag were used in simulation and real experiments.

Simulation and Experimental Results One simulation and two real experiments are presented in order to validate the control law. Two Turtlebots Gerkey and Conley (2011) with 2 DOF (translation $\boldsymbol{\nu}_x$ and rotation $\boldsymbol{\omega}_z$) are used as experimental robots. They are equipped with a Kinect device and linked by a 1.4 meter long and 3 millimeter thick orange rope (see Figure 4.6). The rope maximum sag is $H_{max} = 0.40$m, corresponding to its attachment point's height. The servoing algorithm runs in the follower robot computer, which is equipped with a Intel Core i5-2410M @ 2.3 GHz processor. The video frame rate is fixed to 10Hz. The same robot motion capabilities, rope length, thickness and maximum sag are used in simulation and real experiments.

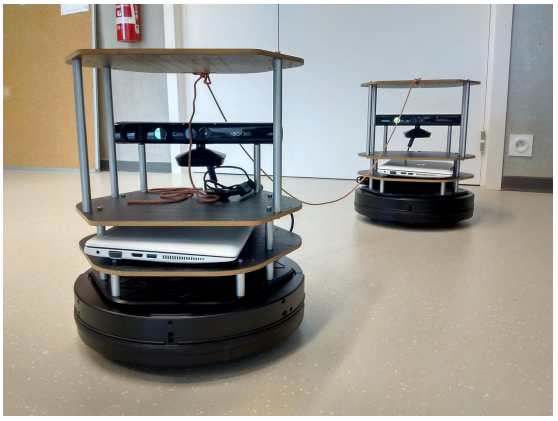


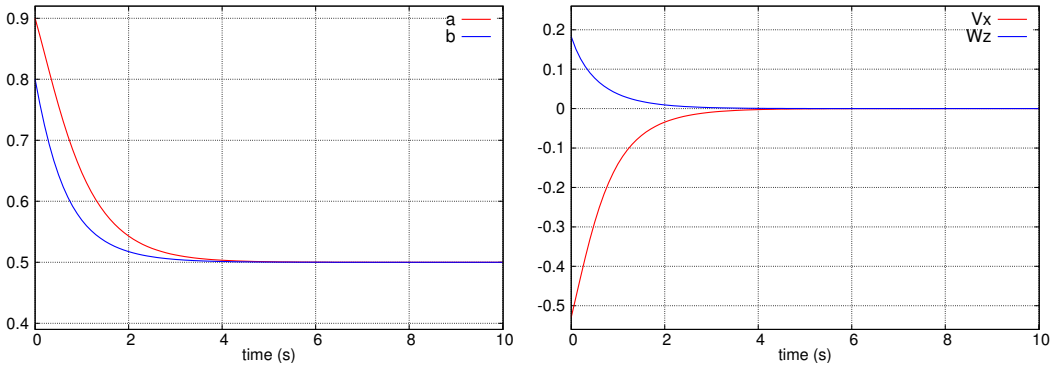
Figure 4.6: Experimental setup: two Turtlebots (Gerkey and Conley, 2011) simulate a tether handling system for remotely operated robots. The leader robot freely explores its surroundings while the follower robot is expected to maintain the tether slack enough not to hamper the leader movements

The objective of the simulation is that the follower robot moves the tether from a very slackened to a moderately tight shape. The tether orientation is controlled, passing from a large angle θ[1] to a desired smaller angle. The initial and desired values of the feature vector are $\mathbf{s}_o = (0.9, 0.8)$ and $\mathbf{s}^* = (0.5, 0.5)$, respectively. We assume that the robots can perfectly estimate the tether parameters, so the fitting process is not simulated. Figure 4.7 presents the simulation results. As expected, both parameters have an exponential decay and converge to the desired value within 4 seconds for a gain $\lambda = 0.75$ and a sampling period of 0.1 seconds.

In the first experiment, the simulation initial conditions are repeated and the follower robot moves the tether from an initial shape $\mathbf{s}_o = (0.9, 0.8)$ to a desired shape $\mathbf{s}^* = (0.5, 0.5)$. We used the same gain as for simulation ($\lambda = 0.75$) and a fitting quality threshold $Q_{max} = 1$. Results are summarized in Figure 4.8.

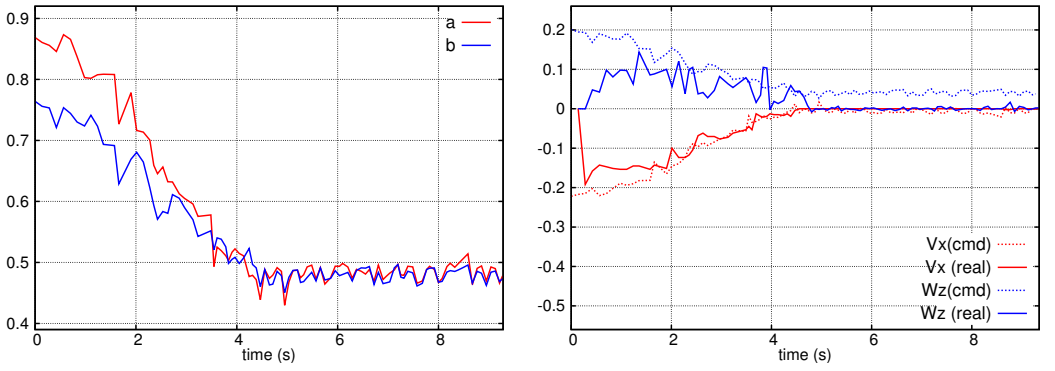
In the second experiment (see Figure 4.9), the leader robot is freely displaced while the follower robot ensures that the rope keeps a desired shape $\mathbf{s}^* = (0.7, -0.5)$. First, the leader robot moves forward, then turns left (24s). Next, it moves about 1.25 meters forward (38s), makes a half-turn (50s) and moves about 1.25 meters forward

[1]for the Turtlebots, we note the orientation angle as θ, instead of α



(a) (b)

Figure 4.7: Simulation results for tether shape control. (a) the parameters evolution. The tether goes from an initial to a desired shape ($\mathbf{s}_o = (0.9, 0.8)$ and $\mathbf{s}^* = (0.5, 0.5)$, respectively). (b) the control velocities. Linear velocity ($\boldsymbol{\nu}_x$) in m/s and angular velocity ($\boldsymbol{\omega}_z$) in rad/s

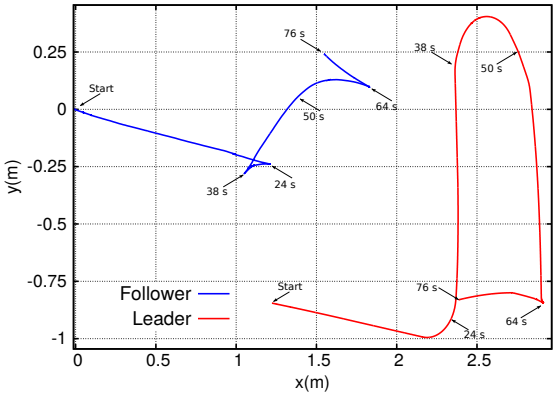


(a) (b)

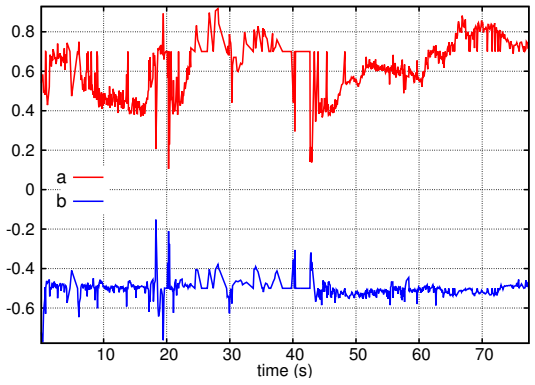
Figure 4.8: Results of a real experiment for tether shape control. (a) the tether parameters evolution. The tether goes from an initial to a desired shape ($\mathbf{s}_o = (0.9, 0.8)$ and $\mathbf{s}^* = (0.5, 0.5)$, respectively). (b) the control velocities. Linear velocity ($\boldsymbol{\nu}_x$) in m/s and angular velocity ($\boldsymbol{\omega}_z$) in rad/s

again (64s). At the end, it makes a quarter turn and moves about 0.50 meters backward (76s). Compared with the first experiment, a higher value was set to the tether first parameter in order to give more freedom of maneuvering to the leader robot. We used $Q_{max} = 1$ and two different gains for linear and angular velocities: $\lambda_l = 1.0$ and $\lambda_\omega = 6.0$, respectively (see discussion below). Figure 4.9c presents the fitting quality index evolution during the experiment. The feature prediction was used 20 times in cases of wrong rope detection and more 38 times due to inaccurate fitting.

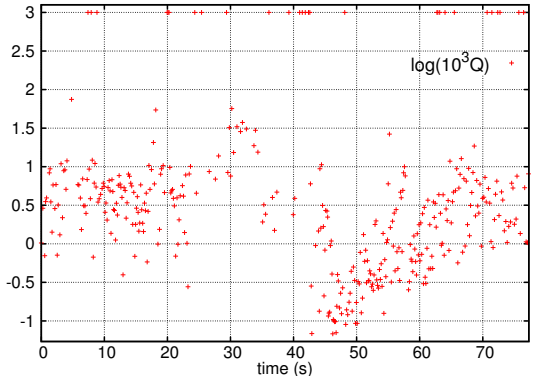
In Figure 4.8, the angular velocity command does not converge to zero, and remains quasi constant. This is due to the limitations of our experimental platform that cannot



(a)



(b)



(c)

Figure 4.9: Two robots are linked by a tether. The leader robot freely moves while the follower robot maintains a desired tether shape $\mathbf{s}^* = (0.7, -0.5)$. (a) the leader and follower trajectories with time indications in seconds. (b) the tether parameters evolution. (c) the fitting quality index Q evolution during the experiment. Feature prediction is used in cases of wrong rope detection and inaccurate fitting ($Q = 1$ i.e. $log(10^3 Q) = 3$)

achieve a rotation velocity lower than 0.05 rad/s. In the second experiment, we selected a higher gain for the angular velocity compared to the linear velocity gain in order to overcome this problem. As a future improvement, an integrator compensator to deal with the low level control of the velocity can be added to the servoing loop.

Looking at Figure 4.8, we can note that the real experimentation curves are noisy compared with those obtained in simulation. This is mainly due to wrong rope detection that can affect the tether feature estimation. Another reason is the possible rope oscillation during the robot motion, which can occur when the robot halts and restarts motion. This could be taken into account by designing a dynamic controller that takes the rope inertia into account. In addition, the rope can be tracked in the image by a gradient-guided algorithm.

Despite the problem of nonholonomy, the control scheme was validated by simulation and by two series of real experiments. The comparison of simulation and real experiments showed that the tether converged to their desired values, validating the proposed control law. More details about this work can be found in appendix E, where the paper **Catenary-based Visual Servoing for Tethered Robots**, presented at ICRA 2017, is reproduced. These experiments constituted our first results obtained with catenary-based visual servoing of umbilicals. This work showed the feasibility of the catenary-based visual servoing proposed in the framework of this thesis.

4.4.1.2 Comparison of Catenary-Based Control with Image-Based Visual Servoing

In a second series of studies, we compared the catenary-based controllers with standard image-based visual servoing techniques. We considered some salient points such as the highest and lowest tether points in the image, or we approximated the first portion of the tether with a line segment (see Figure 4.10). Such features were successfully used to control a robot motion by regulating the shape of rigid objects in the image (Collewet and Chaumette, 2002; Cherubini et al., 2008). The tether shape evolution, velocity commands and condition number of interaction matrices were studied in normal and limit cases for two image-based controllers and both catenary-based controllers proposed:

- **image points**: coordinates of the tether highest and lowest points ($\mathbf{p}_A$ and $\mathbf{p}_B$) detected in the image without taking into account that the points belong to a catenary projection (Espiau et al., 1992)
- **image line segment** : center, length and orientation of the segment $\overline{\mathbf{p}_A\mathbf{p}_B}$ linking the highest and the lowest points of the tether detected in the image, without taking into account that the points belong to a catenary projection (Chaumette et al., 1993)
- **catenary parameters**: two features $\mathbf{s} = [a\ b]^T$, related to tether slackness and orientation that parameterize the catenary curve to fit the tether 3D shape (Laranjeira et al., 2017).

- **catenary-projection lowest point**: coordinates of the tether lowest point ($\mathbf{p}_B$) detected in the image, making the approximation that this point is the projection of the 3D catenary lowest point .

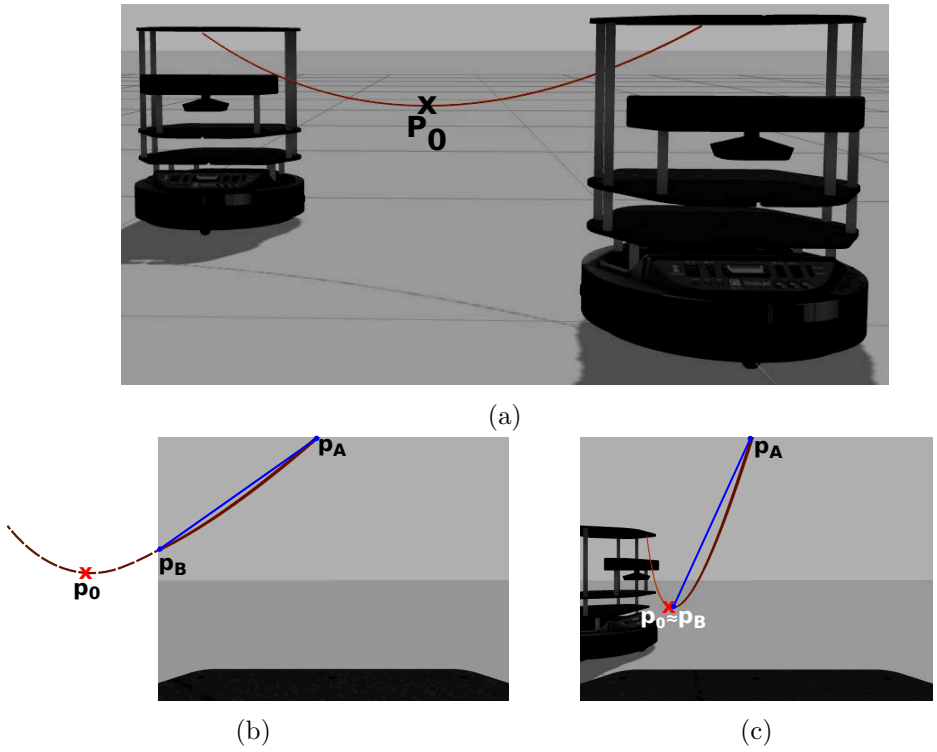


Figure 4.10: Image features used to manage the tether shape: its highest and lowest points in the image ($\mathbf{p}_A$ and $\mathbf{p}_B$) and the line segment $\overline{\mathbf{p}_A\mathbf{p}_B}$ in blue. (a) 3D simulated scene where the tether 3D lowest point $\mathbf{P}_0$ is out of the follower's camera field of view. (b) corresponding embedded view with $\mathbf{p}_0$ being the perspective projection of $\mathbf{P}_0$. (c) Another situation where $\mathbf{P}_0$ is inside the follower's camera field of view

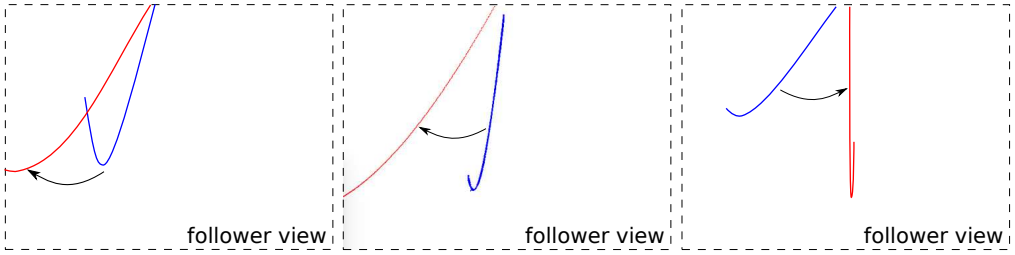
Simulation are performed using Gazebo Koenig and Howard (2004) to test the whole control loop including computer vision algorithms. The setup is composed of two Turtlebots Gerkey and Conley (2011) linked by a 1 meter long and 3 mm thick orange rope that simulates the tether. The robots have 2 DOF (one translation and one rotation) and the follower is equipped with a camera that monitors an extremity of the rope. The maximum authorized rope sag is fixed to $H_{max} = 0.40m$, corresponding to its attachment points height.

The rope is detected in the follower's camera by a color segmentation algorithm, that is refined by morphological skeletonization. The segmented points are then used to compute the visual servoing features.

Three simulations are carried out in order to compare the performances of vision-based tether shape controllers (see Figure 4.11). The objective is to study the controllers

behavior in normal conditions, when the tether shape is clearly visible, as well as in limit cases, when the tether is not completely visible or when it degenerates into a vertical line in the image.

The first simulation (Figure 4.11a) is a typical example where the tether is completely visible and not degenerated in the image. The orientation angle θ[2] is required to vary from 20° to 45° while the tether slackness must remain the same (initial $H_0 = 0.20m$ and desired $H^* = 0.20m$). In the second simulation (4.11b), the angle θ is required to vary from 10° to 60° to check controllers performance when the tether is partially visible (θ close to 60°). The initial and desired slackness are the same ($H_0 = H^* = 0.25m$). In the third simulation (4.11c), the angle θ is required to vary from 20° to 0° to check the controller performance when the tether plane is near to the camera optical axis (θ close to 0). The initial and desired slackness are $H_0 = 0.10m$ and $H^* = 0.24m$, respectively.



(a) normal case (b) limit case 1 (c) limit case 2

Figure 4.11: Follower robot camera view for (a) first, (b) second and (c) third simulation. Initial and desired tether shape in the image are drawn in blue and red lines, respectively

Figures 4.12, 4.13 and 4.14 show, for the three studied cases, the tether shape parameters error and the velocity commands of the follower robot as a function of time. Figure 4.15 presents the trajectory of the tether lowest point in the image. Figure 4.16 presents the condition number evolution for the controllers interaction matrices and the Gauss-Newton Jacobian used for the catenary curve fitting. The condition number of a matrix $\mathbf{A}$ is defined as follows Feddema et al. (1989):

$$cond(\mathbf{A}) = |\mathbf{A}||\mathbf{A}^+| \tag{4.50}$$

where $|.|$ is the L_2 norm.

In the normal case, the tether shape control is achieved by all the controllers. They also allow to reach the desired position in the image plane (see Figure 4.12, and 4.15a). The tether parameters (slackness H and orientation θ) converge to their desired values even using image-based features. Image-based controllers are very sensitive to the detection of the lowest part of the tether, and work well when this part is well visible in the image. Compared with pure image-based controllers, the catenary lowest-point controller appears to execute a more balanced regulation between slackness (Figure 4.12a)

[2]for the Turtlebots, we note the orientation angle as θ, instead of α

and orientation (Figure 4.12b). In the image plane, the resulting trajectory of the lowest point is more regular than pure image-based trajectories (Figure 4.15a). In addition, the catenary parameter controller is less sensitive to noisy tether detection whereas the other controllers are very dependent on the accurate detection of the lowest point in the image.

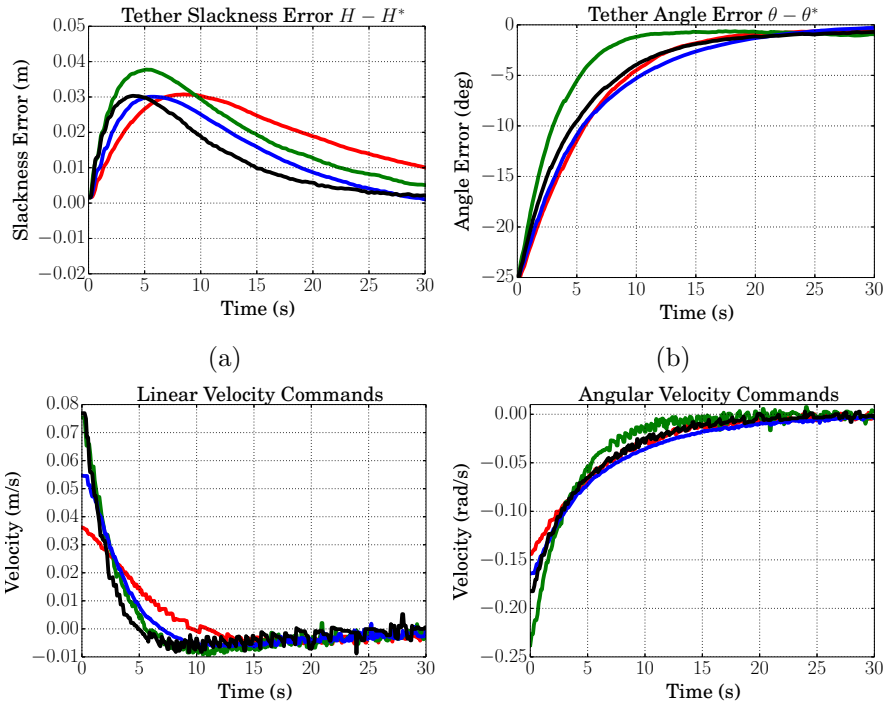


Figure 4.12: Normal case: error evolution for (a) tether slackness and (b) orientation as well as (c) linear and (d) angular command velocities. Tether initial and desired shape are respectively: $(H_0 = 0.20m, \theta_0 = 20°)$ and $(H^* = 0.20m, \theta^* = 45°)$. Legend in Fig. 4.15d. All controllers use a gain $\lambda = 1.0$.

In the first limit case, the tether desired shape is only achieved by the catenary parameter controller (Figure 4.13). The other controllers cannot reach the desired tether shape leading to a residual offset of slackness (Figure 4.13a) and angle (Figure 4.13b). However, these controllers converge as expected in the image plane, while we observe a discontinuity of the lowest point trajectory in the case of catenary-parameter controller (Figure 4.15b). This is not surprising since the catenary-parameter controller directly regulates the tether 3D shape, and converges to the real lowest point that is outside the image.

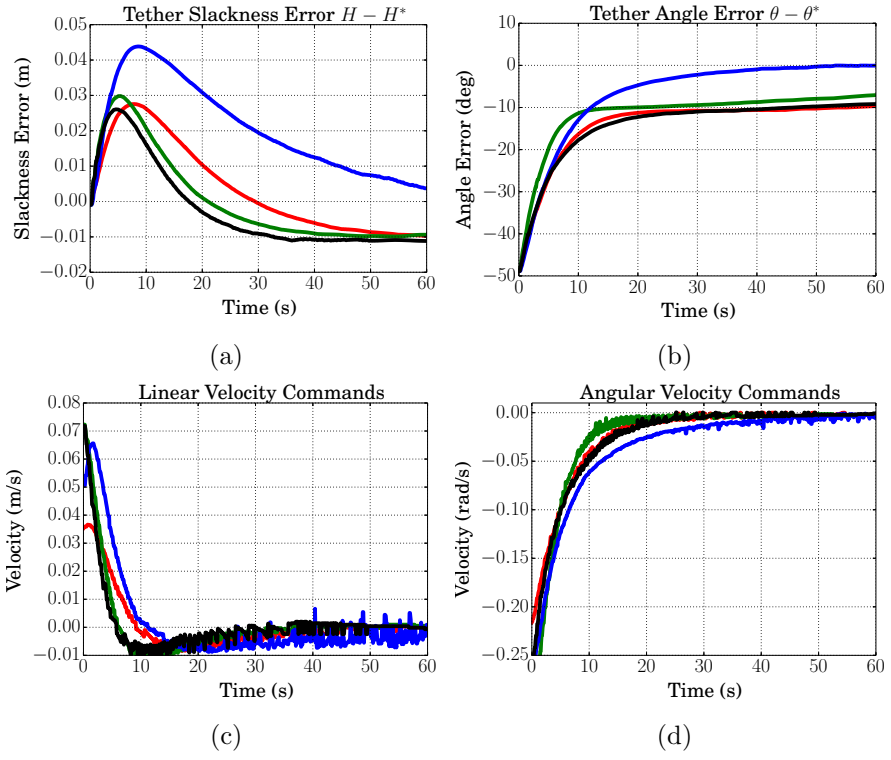


Figure 4.13: Limit case 1: error evolution for (a) tether slackness and (b) orientation as well as (c) linear and (d) angular command velocities. Tether initial and desired shape are respectively: $(H_0 = 0.25m, \theta_0 = 10°)$ and $(H^* = 0.25m, \theta^* = 60°)$. Legend in Fig. 4.15d. All controllers use a gain $\lambda = 1.0$.

In the second limit case (see Figures 4.14a to 4.14d), image-based and catenary lowest point controllers achieve the tether desired shape. The image features can be well estimated since at least half of the tether is visible. However, the catenary parameters controller cannot reach the target shape because of fitting failure when the tether gets close to the camera sagittal plane. This is confirmed by the condition number of the curve fitting Gauss-Newton Jacobian (4.16d) that becomes ill-conditioned at the end of the simulation. Nevertheless, all the interaction matrices are well conditioned (see Figures 4.16a to 4.16c), which means that they are robust to input noise.

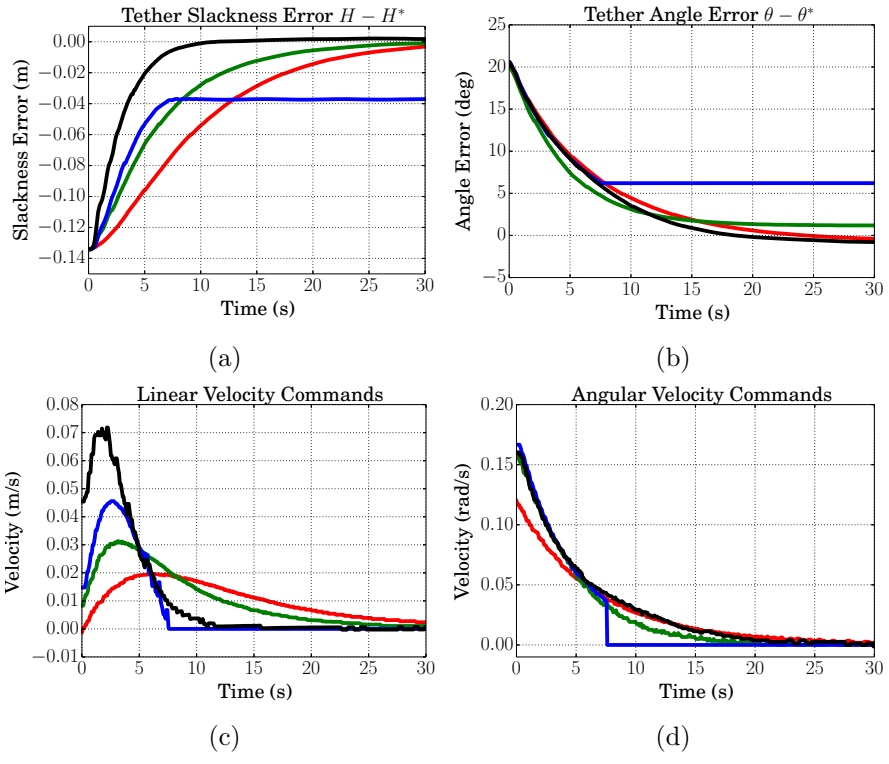
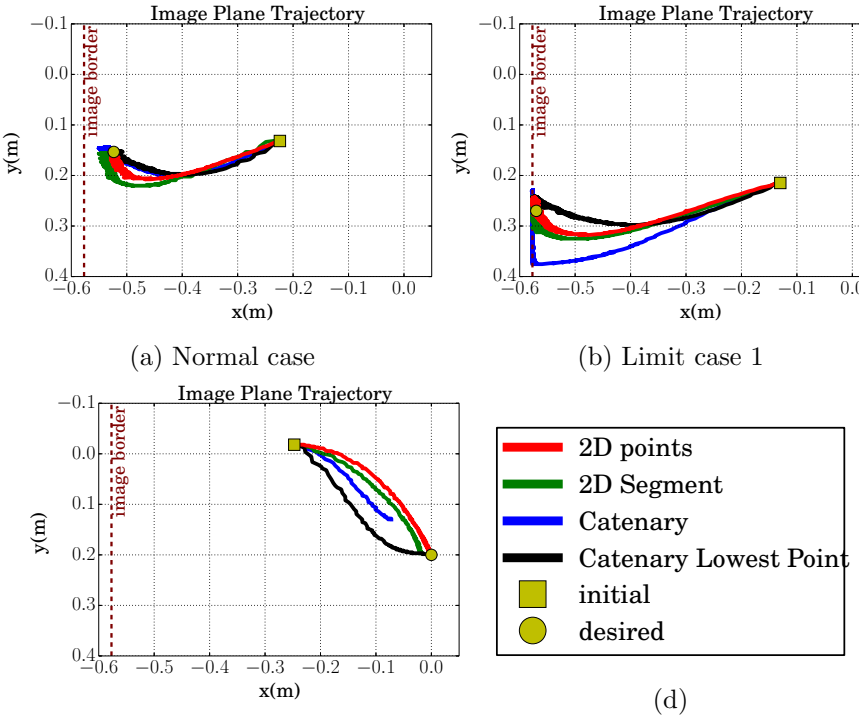


Figure 4.14: Limit case 2: error evolution for (a) tether slackness and (b) orientation as well as (c) linear and (d) angular command velocities. Tether initial and desired shape are respectively: $(H_0 = 0.10m, \theta_0 = 20°)$ and $(H^* = 0.24m, \theta^* = 0°)$. Legend in Fig. 4.15d. All controllers use a gain $\lambda = 1.0$.



(a) Normal case

(b) Limit case 1

(c) Limit case 2

(d)

Figure 4.15: Tether lowest point trajectory in the image plane for (a) normal case, (b) limit case 1 and (c) limit case 2. (d) legend

The catenary lowest-point controller makes use of an initial guess of the tether angle parameter. In the proximity of $\theta = 0$, the slackness parameter is not well estimated due to a singularity in the Jacobian matrix of the Gauss-Newton. In order to improve the catenary parameter controller, a solution will be to directly carry out an initial guess on the slackness parameter.

All four controllers need at least 50% of tether detected points in the image when the tether gets closer to the camera sagittal plane. In real situations it will be difficult to detect half the tether the longer it is. In order to deal with this issue, a solution could be to equip the robot with an additional camera outside its sagittal plane. Another solution would consist of placing fiducial markers along the tether at known length, which could be easily detected in the image and overcome the noisy detection of the lowest point.

More details about this comparison of controllers are given in appendix E. The study showed that the additional knowledge about the tether 3-D shape in the visual servoing control allowed to obtain satisfactory tether handling, provided that the projection of the tether is not degenerated into a straight line in the image. The visual controller

based on catenary parameters performed better in cases where the tether was partially out of the camera field of view. The visual control based on the lowest point could be useful near the degenerated cases when the lowest part of the catenary is visible. The standard Image-Based features failed to converge accurately in the case of partial visibility of the tether in the image. Both catenary-based controllers could be combined to design a more robust controller.

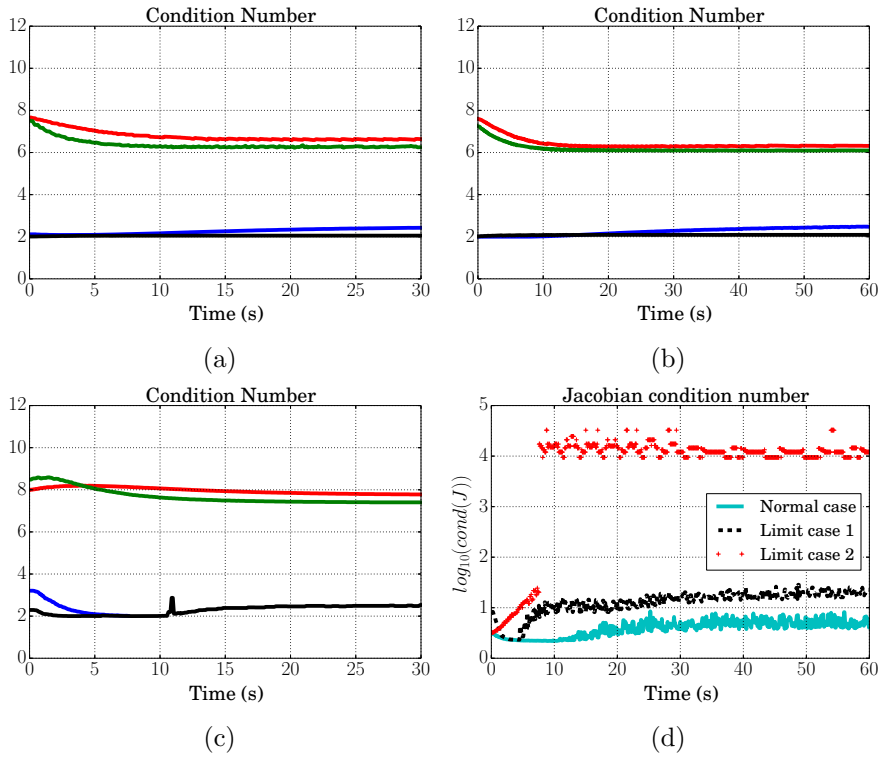


Figure 4.16: Interaction matrices condition number evolution for (Fig. a) normal and (Figs. b and c). Legend in Fig. 4.15d. (d) condition number evolution for the curve fitting Gauss-Newton Jacobian during normal and limit cases

4.4.2 Underwater Tether Shape Regulation while Leader Robot is Motionless

Let us consider the case with underwater robots where the leader robot is motionless. The follower robot should bring the tether from its initial towards the desired shape, as given in Table 4.1. The initial and desired shape of the tether in the image are depicted in Figure 4.17.

	Follower		
	H_2(m)	α_2	ΔH_2(m)
initial	0.50	80°	0.0
desired	0.15	30°	0.20

Table 4.1: Initial and desired tether features for visual servoing control scheme using only the follower robot's camera.

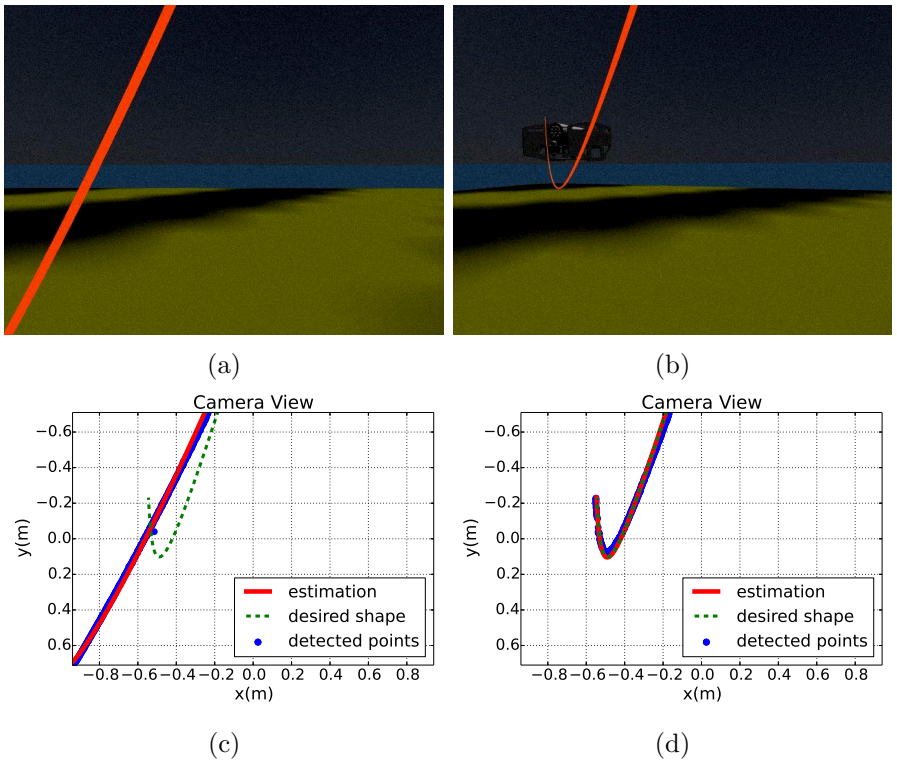


Figure 4.17: Follower onboard view of the tether. The (a,c) initial and (b,d) final images of the visual servoing. Figures (a) and (b) are the camera source images while figures (c) and (d) depict the tether detected points (in blue), the shape estimation (in red) and the desired shape in the image (in green). The tether initial and desired features are given in Table 4.1.

At the beginning, the leader robot is outside the camera field of view and the tether

is only partially visible in the image. The objective is to bring the tether into the camera field of view in order to ensure an accurate estimation of its shape. Moreover, we would like to maintain the tether not too slack in order to prevent it from getting entangled with obstacles in the sea floor, for example.

In the following subsections, a number of control schemes are tested in this scenario. First, the complete 3×4 interaction matrix is used so that the follower robot performs all features regulation (H_2, α_2 and ΔH_2) simultaneously. Second, a sum of controllers set up, splitting the interaction matrix into two, in order to separately regulate the features (H_2, α_2) and (ΔH). Finally, hierarchical control schemes are introduced, where the regulation of H_2 and α_2 is prioritized over the regulation of ΔH_2. The control gain is always the same for all controllers : $\lambda = 1.0$.

4.4.2.1 Visual Servoing Control with 3×4 Interaction Matrix

In this control scheme, the follower robot must regulate the feature vector $\mathbf{s}_2$ as a whole.

Control Law From equations (4.39) and (4.49), and fixing ${}^{r_1}\mathbf{v} = 0$, we have:

$$\mathbf{L}_{r_2,2}{}^{r_2}\mathbf{v} = \dot{\mathbf{e}}_2. \tag{4.51}$$

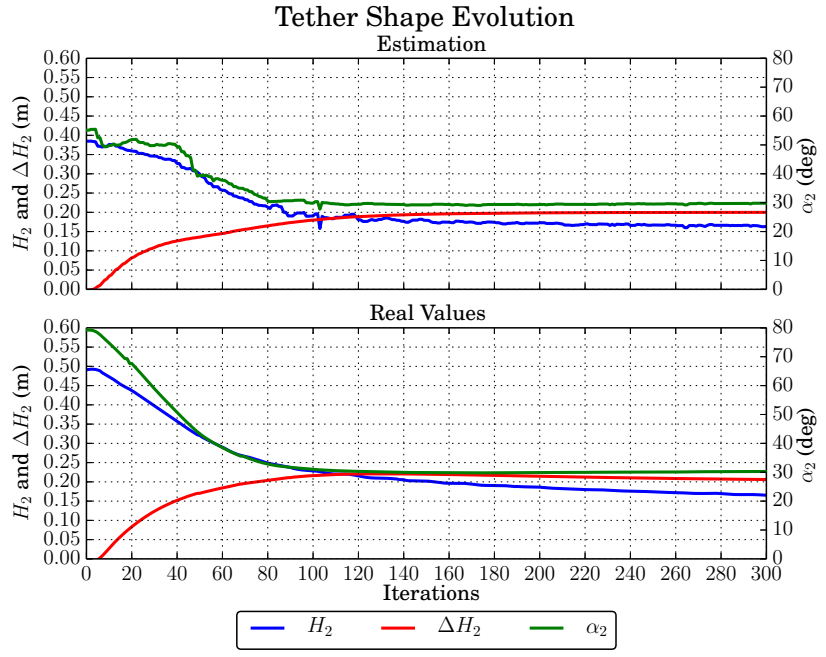
Setting $\dot{\mathbf{e}}_2 = -\lambda \mathbf{e}_2$ for exponential convergence, with $\lambda \in \mathbb{R}^+$, the follower command velocity can be calculated as follows:

$${}^{r_2}\mathbf{v} = -\lambda \mathbf{L}^+_{r_2,2}\mathbf{e}_2. \tag{4.52}$$

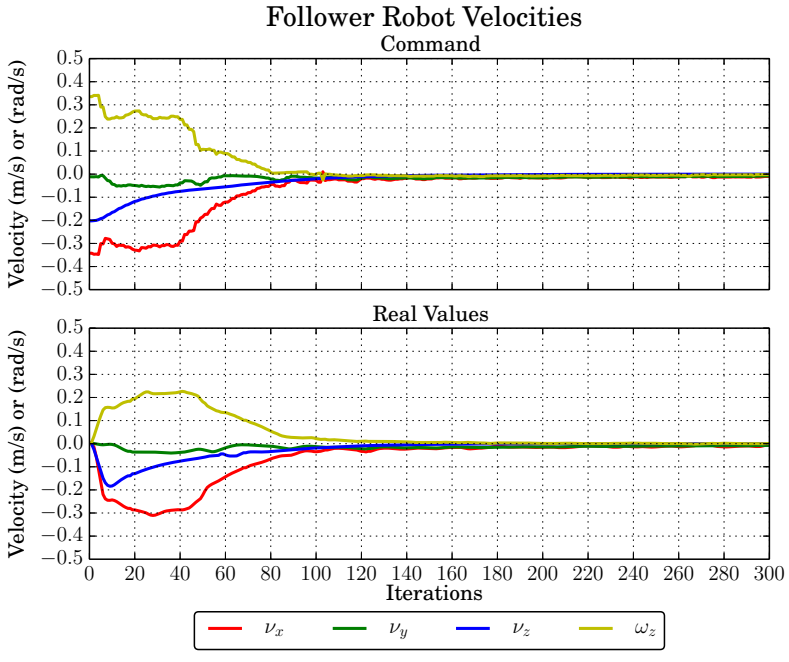
Simulation Results The tether shape evolution, from visual feedback estimation and simulation ground truth, is depicted in Figure 4.18a. The robot's commanded and executed velocities are shown in Figure 4.18b.

At the beginning of the simulation ($0 - 40$ iterations), the tether shape estimation is not accurate, mainly with respect to the parameter α_2 (see Figure 4.18a). Indeed, as presented in Figure 4.17a, the tether is almost a straight line in the image at its initial shape, and the catenary curve fitting procedure cannot yield an accurate estimation of its shape. Despite this, the control loop brings the tether towards its desired shape in the image (see Figure 4.17d) and the tether features (H_2, α_2 and ΔH_2) converge towards their desired values (see Figure 4.18a).

With respect to the command velocities (see Figure 4.18b), we observe a decay after the 40^{th} iteration, the time the tether lowest point takes to come back into the camera field of view. The robot commanded velocities are nearly zero after the 120^{th} iteration, since the tether reached its desired shape.



(a)



(b)

Figure 4.18: Simulation results for a control law using full 3×4 interaction matrix. (a) tether shape and (b) follower robot velocity evolution. The tether desired shape is given by $H_2^* = 0.15\,\text{m}$, $\alpha_2^* = 30^\circ$ and $\Delta H_2^* = 0.20\,\text{m}$.

4.4.2.2 Sum of Controllers

We now introduce another control scheme, splitting the interaction matrix into two, in order to separately regulate the features (H_2, α_2) and (ΔH_2). Thus, we ensure that the control of ΔH_2 only generates vertical velocity commands (ν_z), whereas it might also generate forward velocity commands (ν_x) using the 3×4 interaction matrix.

Control Law Let us note $\mathbf{e}_2^{H,\alpha}$ and $\mathbf{e}_2^{\Delta H}$ the error vectors related to the regulation of (H_2, α_2) and ΔH_2, respectively:

$$\mathbf{e}_2^{H,\alpha} = \begin{bmatrix} a_2 - a_2^* \\ b_2 - b_2^* \end{bmatrix}$$

and

$$\mathbf{e}_2^{\Delta H} = \begin{bmatrix} d_2 - d_2^* \end{bmatrix}.$$

Their interaction matrices are obtained splitting the matrix given in equation (4.44), as follows:

$$\mathbf{L}_{r2,2}^{H,\alpha} = \begin{bmatrix} -\frac{\sqrt{1-b_2^2}}{(u_2+p_2)H_{max}} & -\frac{b_2}{(u_2+p_2)H_{max}} & \frac{v_2+q_2}{(u_2+p_2)H_{max}} & {}^{r_2}Y_1\frac{\sqrt{1-b_2^2}}{(u_2+p_2)H_{max}} - {}^{r_2}X_1\frac{b_2}{(u_2+p_2)H_{max}} \\ \frac{b_2\sqrt{1-b_2^2}}{2D_2+\Delta D_2} & -\frac{1-b_2^2}{2D_2+\Delta D_2} & 0 & {}^{r_2}Y_1\frac{-b_2\sqrt{1-b_2^2}}{2D_2+\Delta D_2} - {}^{r_2}X_1\frac{1-b_2^2}{2D_2+\Delta D_2} \end{bmatrix}$$

and

$$\mathbf{L}_{r2,2}^{\Delta H} = \begin{bmatrix} 0 & 0 & -\frac{1}{2\Delta H_{max}} & 0 \end{bmatrix}.$$

Similarly to equation (4.52), we can calculate the follower robot control law as:

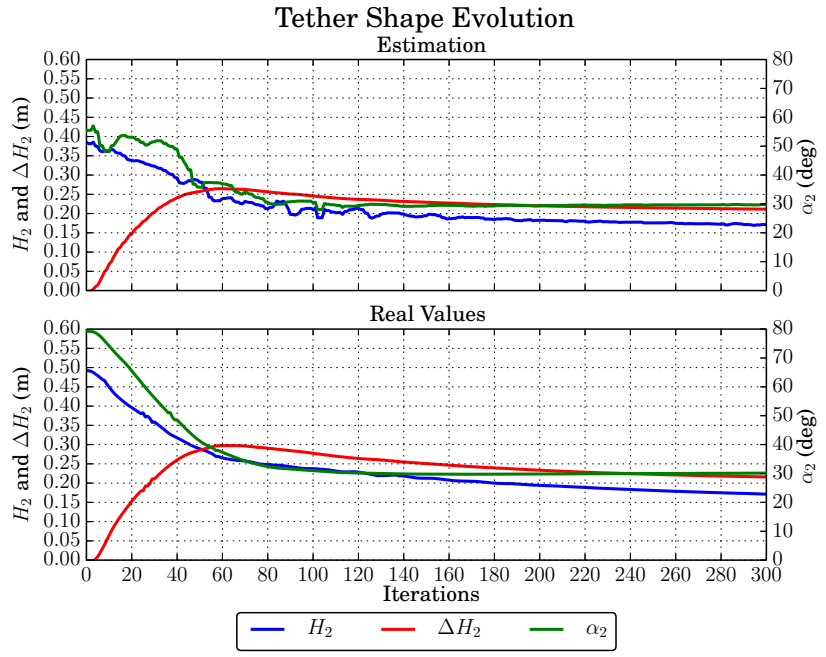
$${}^{r_2}\mathbf{v} = -\lambda_1 \left(\mathbf{L}_{r2,2}^{H,\alpha}\right)^+ \mathbf{e}_2^{H,\alpha} - \lambda_2 \left(\mathbf{L}_{r2,2}^{\Delta H}\right)^+ \mathbf{e}_2^{\Delta H}, \tag{4.53}$$

where $\lambda_1, \lambda_2 \in \mathbb{R}^+$ are the control gains for the regulation of $\mathbf{e}_2^{H,\alpha}$ and $\mathbf{e}_2^{\Delta H}$.

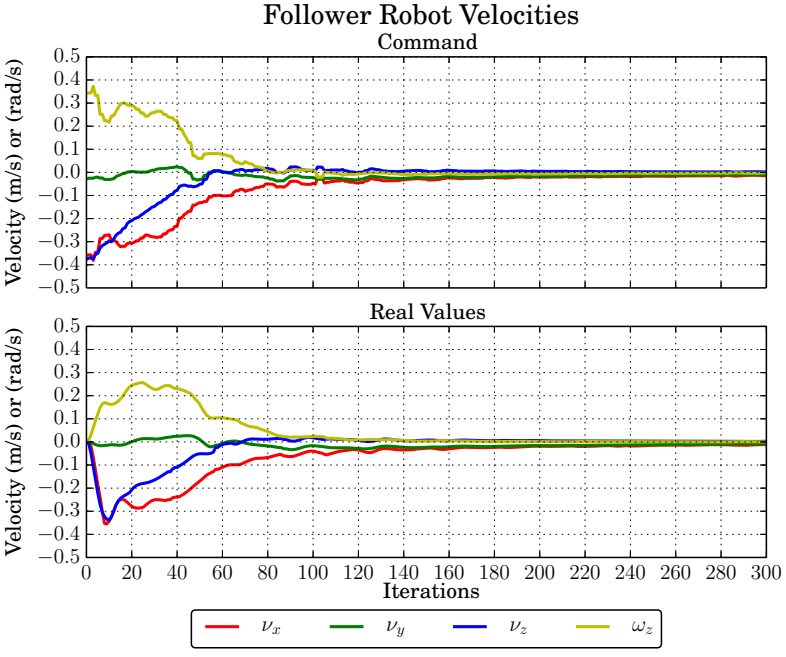
Simulation Results The same simulation presented in the previous subsection is now performed using the sum controller. As a reminder, the tether initial and desired shape are given in Table 4.1. The control gain is set to $\lambda_1 = \lambda_2 = 1.0$. The tether shape evolution, from visual feedback estimation and simulation ground truth, is depicted in Figure 4.19a. The robot's commands and executed velocities are shown in Figure 4.19b.

Analyzing Figure 4.19a, we observe that the tether reaches its desired shape within 300 iterations. However, the feature ΔH_2 takes more time to converge when compared to the controller using the entire interaction matrix (see Figure 4.18a).

Such a control scheme may be interesting to ensure that the regulation of ΔH_2 only implies a vertical motion. Yet, since the first task also implies vertical motion, it can happen that none of the tasks are fully regulated. In order to ensure tasks regulation, we have to set up a hierarchical control scheme.



(a)



(b)

Figure 4.19: Simulation results for the sum of controllers. (a) tether shape and (b) follower robot velocity evolution. The tether desired shape is given by $H_2^* = 0.15\,\text{m}$, $\alpha_2^* = 30^\circ$ and $\Delta H_2^* = 0.20\,\text{m}$.

The interaction matrix presented in equation (4.44) already has nice decoupling properties, which allows the features to be simultaneously regulated without disturbing each other. As a result, we obtain a faster convergence of the tether features using the entire matrix instead of making the sum controllers. Let us look in the sequel the results obtained with hierarchical control schemes.

4.4.2.3 Hierarchical Task Control

Let us now split the follower robot command into two tasks. The first task corresponds to the regulation of H_2 and α_2, The second task corresponds to the regulation of ΔH_2. In order to ensure that the secondary task will not interfere the execution of the priority task, the redundancy formalism can be used (Mansard and Chaumette, 2007).

Control Law Let us note $\mathbf{e}_2^{H,\alpha}$ and $\mathbf{e}_2^{\Delta H}$ as the error vector of the priority and secondary tasks, respectively:

$$\mathbf{e}_2^{H,\alpha} = \begin{bmatrix} a_2 - a_2^* \\ b_2 - b_2^* \end{bmatrix}$$

and

$$\mathbf{e}_2^{\Delta H} = \begin{bmatrix} d_2 - d_2^* \end{bmatrix}.$$

Similarly to equation (4.52), we can calculate the follower robot control law as:

$${}^{r_2}\mathbf{v} = -\lambda \left(\mathbf{L}_{r2,2}^{H,\alpha}\right)^+ \mathbf{e}_2^{H,\alpha} + \mathbf{P}_1 \mathbf{z}, \tag{4.54}$$

where

$$\mathbf{P}_1 = \mathbf{I} - \left(\mathbf{L}_{r2,2}^{H,\alpha}\right)^+ \mathbf{L}_{r2,2}^{H,\alpha} \tag{4.55}$$

is the orthogonal projection operator on the null space of $\mathbf{L}_{r2,2}^{H,\alpha}$, the interaction matrix of the first task. The vector $\mathbf{z}$ is an arbitrary vector that can be used to apply the secondary task. Thanks to the projector operator $\mathbf{P}_1$, the first task is ensured not to be disturbed by the secondary command introduced by $\mathbf{z}$.

We can now introduce the secondary task through:

$$\mathbf{z} = -\lambda_2 \left(\mathbf{L}_{r2,2}^{\Delta H}\right)^+ \mathbf{e}_2^{\Delta H}$$

and rewriting equation (4.54) as:

$${}^{r_2}\mathbf{v} = -\lambda_1 \left(\mathbf{L}_{r2,2}^{H,\alpha}\right)^+ \mathbf{e}_2^{H,\alpha} - \lambda_2 \mathbf{P}_1 \left(\mathbf{L}_{r2,2}^{\Delta H}\right)^+ \mathbf{e}_2^{\Delta H}, \tag{4.56}$$

where $\lambda_1, \lambda_2 \in \mathbb{R}^+$ are the control gains for the first and second tasks, respectively. The interaction matrices $\mathbf{L}_{r2,2}^{H,\alpha}$ and $\mathbf{L}_{r2,2}^{\Delta H}$ are obtained splitting the matrix given in equation (4.44), as made in subsection 4.4.2.2.

Actually, this control law can be seen as the application of the first task commands and those of the second task that do not disturb the first task execution. The task

hierarchy is respected. However, the second task execution is not optimal in the least-square sense.

An alternative to the control law given in equation (4.56) can be designed in order to meet the requirements of optimal control.

Similarly to equation (4.51), the evolution of the second task error is given by:

$$\dot{\mathbf{e}}_2^{\Delta H} = \mathbf{L}_{r2,2}^{\Delta H\, r_2}\mathbf{v}. \tag{4.57}$$

Then, introducing equation (4.57) in (4.54) we obtain:

$$\mathbf{z} = \left(\mathbf{L}_{r2,2}^{\Delta H}\mathbf{P}_1\right)^+ \left[\dot{\mathbf{e}}_2^{\Delta H} - \mathbf{L}_{r2,2}^{\Delta H}\left(\mathbf{L}_{r2,2}^{H,\alpha}\right)^+ \dot{\mathbf{e}}_2^{H,\alpha}\right], \tag{4.58}$$

which can be finally introduced back in equation (4.54), yielding:

$${}^{r_2}\mathbf{v} = \left(\mathbf{L}_{r2,2}^{H,\alpha}\right)^+ \dot{\mathbf{e}}_2^{H,\alpha} + \mathbf{P}_1\left(\mathbf{L}_{r2,2}^{\Delta H}\mathbf{P}_1\right)^+ \left[\dot{\mathbf{e}}_2^{\Delta H} - \mathbf{L}_{r2,2}^{\Delta H}\left(\mathbf{L}_{r2,2}^{H,\alpha}\right)^+ \dot{\mathbf{e}}_2^{H,\alpha}\right], \tag{4.59}$$

Since $\mathbf{P}_1$ is idempotent and Hermitian (it is a projection operator), equation (4.59) can be rewritten as (Mansard and Chaumette, 2007):

$${}^{r_2}\mathbf{v} = \left(\mathbf{L}_{r2,2}^{H,\alpha}\right)^+ \dot{\mathbf{e}}_2^{H,\alpha} + \left(\widetilde{\mathbf{L}}_{r2,2}^{\Delta H}\right)^+ \dot{\tilde{\mathbf{e}}}_2^{\Delta H}, \tag{4.60}$$

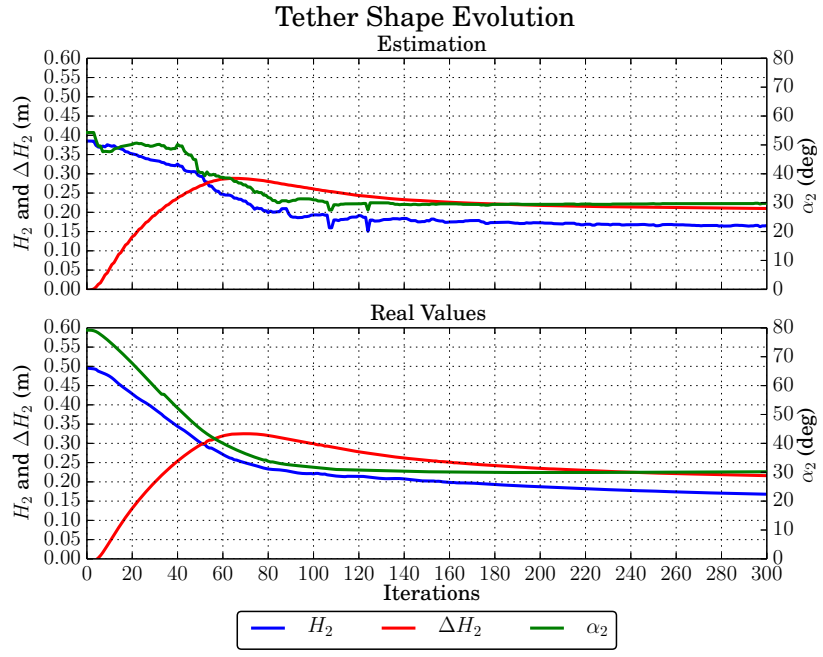
where $\widetilde{\mathbf{L}}_{r2,2}^{\Delta H} = \mathbf{L}_{r2,2}^{\Delta H}\mathbf{P}_1$ and $\dot{\tilde{\mathbf{e}}}_2^{\Delta H} = \dot{\mathbf{e}}_2^{\Delta H} - \mathbf{L}_{r2,2}^{\Delta H}\left(\mathbf{L}_{r2,2}^{H,\alpha}\right)^+ \dot{\mathbf{e}}_2^{H,\alpha}$. We can rewrite this equation as:

$${}^{r_2}\mathbf{v} = -\lambda_1\left(\mathbf{L}_{r2,2}^{H,\alpha}\right)^+ \mathbf{e}_2^{H,\alpha} - \lambda_2\left(\widetilde{\mathbf{L}}_{r2,2}^{\Delta H}\right)^+ \tilde{\mathbf{e}}_2^{\Delta H}, \tag{4.61}$$

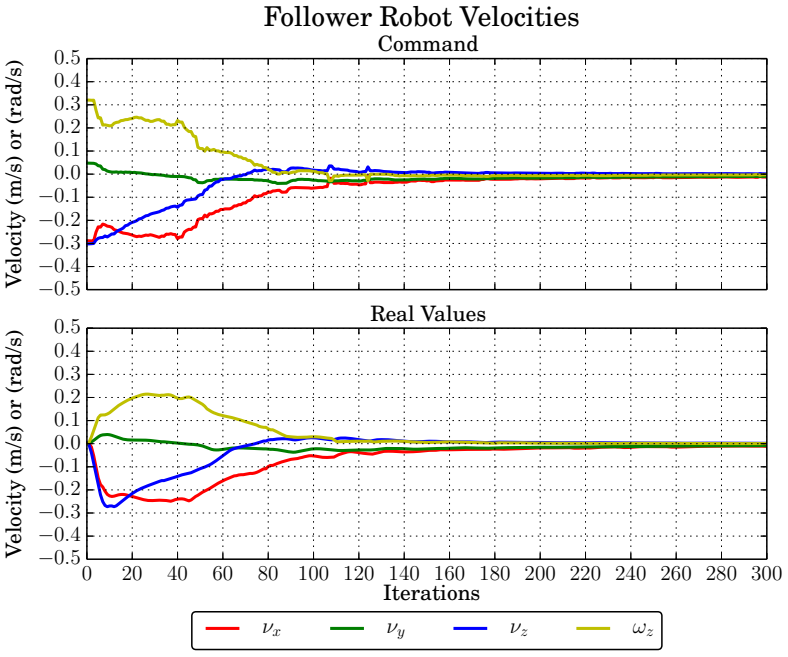
Let us present in the sequel the simulation results obtained from both control laws, given in equations (4.56) and (4.61)

Simulation Results The same simulation presented in the previous subsection is now performed using the hierarchical control laws. As a reminder, the tether initial and desired shape were given in Table 4.1.

The simulation results for the hierarchical non-optimal controller (equation (4.56)) are presented in Figure 4.20. The simulation results for the hierarchical optimal controller (equation (4.61)) are presented in Figure 4.21. The control gains are $\lambda_1 = \lambda_2 = 1.0$ in both cases.

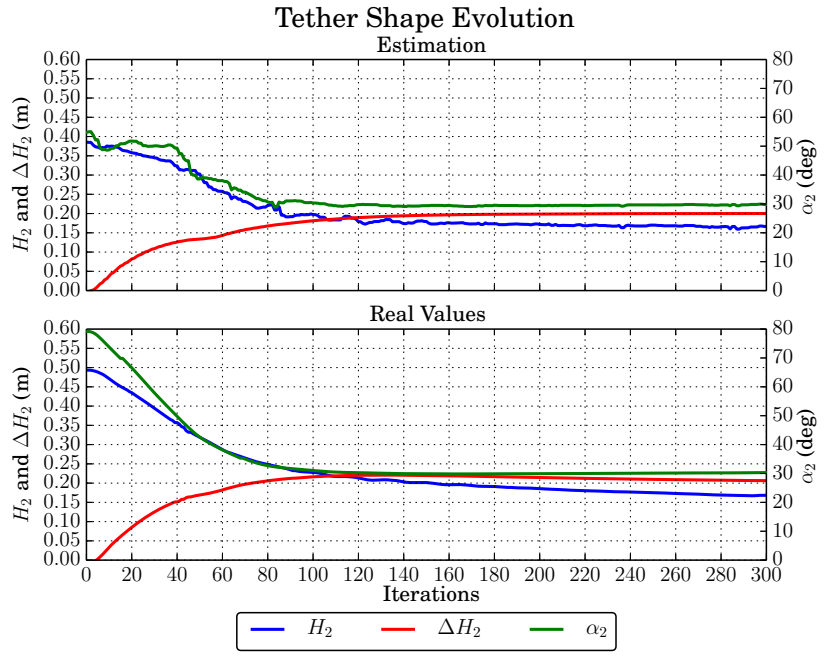


(a)

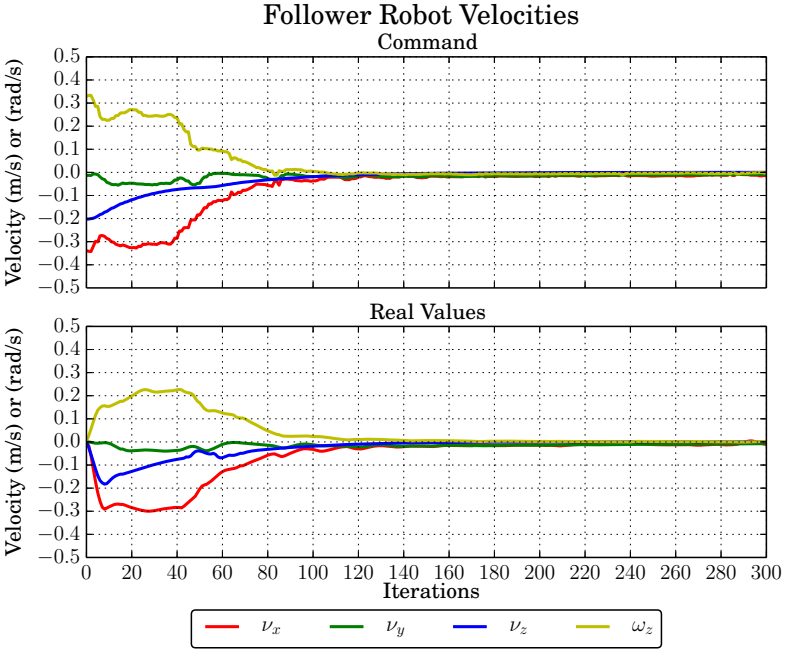


(b)

Figure 4.20: Simulation results for the hierarchical non-optimal control. (a) tether shape and (b) follower robot velocity evolution. The tether desired shape is given by $H_2^* = 0.15\,\text{m}$, $\alpha_2^* = 30^\circ$ and $\Delta H_2^* = 0.20\,\text{m}$.



(a)



(b)

Figure 4.21: Simulation results for the hierarchical optimal control. a) tether shape and (b) follower robot velocity evolution. The tether desired shape is given by $H_2^* = 0.15\,\text{m}$, $\alpha_2^* = 30^\circ$ and $\Delta H_2^* = 0.20\,\text{m}$.

We clearly observe that using the hierarchical non-optimal controller, the secondary task (regulation of ΔH) is not properly regulated at the beginning. The value of ΔH overshoots its desired value by 0.13 m around the 70^{th} iteration (see Figure 4.20a). Using the hierarchical optimal controller, however, the secondary task can be better executed and all the features converge towards their desired values within the 300 iterations without overshoot (see Figure 4.21a). This behavior is expected (Mansard and Chaumette, 2007). In the non-optimal control, the second task commands are just filtered by the projector operator, whereas, in the optimal control, the second task control law allows a more efficient distribution of the control efforts in the remaining degrees of freedom of the robot. The difference on the velocity commands between these two hierarchical control approaches is notable regarding the velocity component ν_z (in blue) in Figures 4.20b and 4.21b. Using the optimal controller, the velocity component ν_z has a lower magnitude, which represents an optimized command in term of control efforts.

4.4.2.4 Comparing Follower Robot Trajectories

The executed trajectories of the follower robot for each of the controllers above studied is depicted in Figure 4.22.

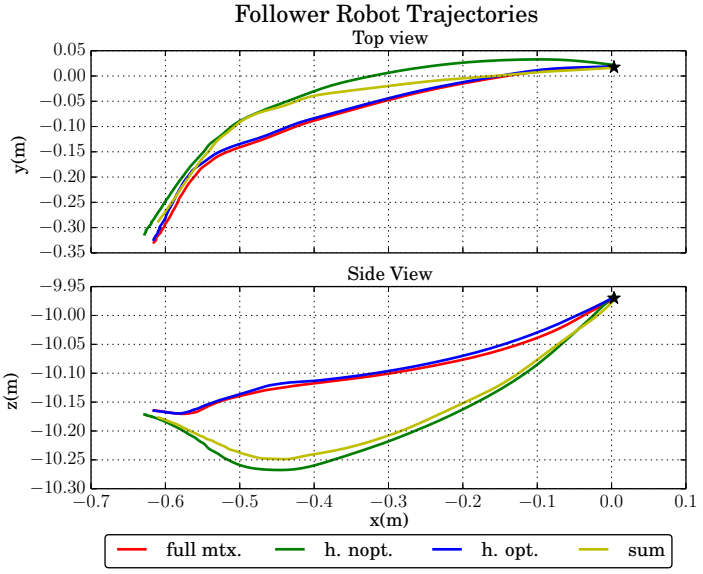


Figure 4.22: Comparing the follower robot trajectories for the four controllers presented in the previous subsections: full 3×4 interaction matrix controller (in red), sum of controllers (in yellow), hierarchical non-optimal (in green) and hierarchical optimal (in blue) controllers. The robot starting position is marked by a black star.

The full interaction matrix and hierarchical optimal controllers generate similar trajectories. These trajectories go more directly towards the final position, when compared to the sum of controllers and the hierarchical non-optimal controller. The full

interaction matrix and hierarchical optimal controllers are more efficient in terms of features regulation. This is also visible regarding the executed trajectories. Moreover, we note that the final positions are not the same for each of the controllers. Actually, the regulated features $(H_2, \alpha_2, \Delta H_2)$ do not constrain the pose of the follower robot, as will be discussed in Section 4.5.

4.4.3 Underwater Tether Shape Regulation while Leader Robot Moves

Let us now consider the case where the leader robot moves and executes a predefined path. This path consists of a constant speed straight forward displacement combined with lateral and vertical motions. The leader robot executed velocities are depicted in Figure 4.23. The lateral (ν_y, in green) and vertical (ν_z, in blue) velocities are executed in parallel with a straight forward motion (ν_x, in red).

All the way through, the follower robot must maintain the tether desired shape given in Table 4.2. The control gain is always the same for all controllers : $\lambda = 2.0$.

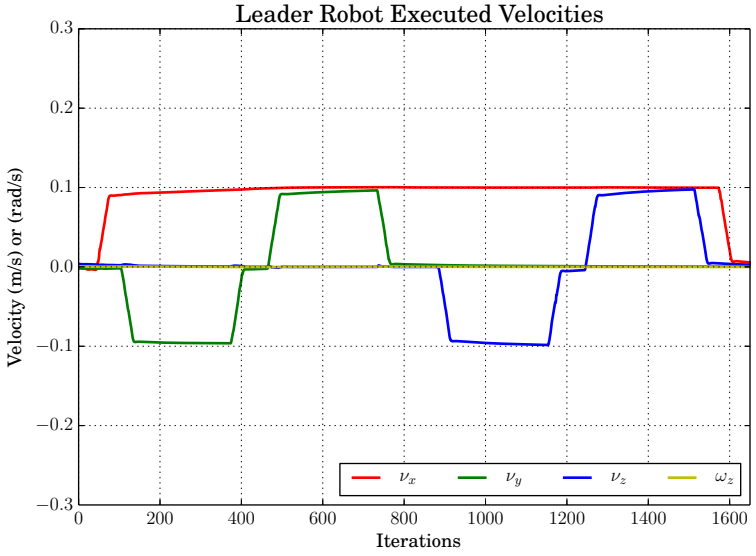


Figure 4.23: The leader robot executed velocities. The executed path consists of a constant straight forward displacement combined with lateral and vertical motions.

	Follower		
	H_2(m)	α_2	ΔH_2(m)
desired	0.35	30°	0.0

Table 4.2: Desired tether features for visual servoing control scheme while the leader robot moves.

Firstly, we present the simulation results where the follower robot tries to regulate the tether shape not being aware of the leader robot motion. Secondly, we include the leader velocity commands in the follower robot control law.

4.4.3.1 Neglecting the Leader Velocity on the Follower Robot Command

Let us analyze the case where the leader robot velocities are not integrated in the follower robot commands. Therefore, the leader robot motion acts as an additional perturbation to the tether shape control task.

Control Law Four control laws are tested:

1. The entire interaction matrix in a non-hierarchical control law, as presented in equation (4.52). As a reminder, the follower command velocities are calculated as:

$$^{r_2}\mathbf{v} = -\lambda \mathbf{L}^{+}_{r_2,2}\mathbf{e}_2.$$

2. A sum of controllers where the features (H_2, α_2) and ΔH_2 are regulated separately, as presented in equation (4.53). As a reminder, the follower command velocities are calculated as:

$$^{r_2}\mathbf{v} = -\lambda_1 \left(\mathbf{L}^{H,\alpha}_{r2,2}\right)^{+} \mathbf{e}^{H,\alpha}_2 - \lambda_2 \left(\mathbf{L}^{\Delta H}_{r2,2}\right)^{+} \mathbf{e}^{\Delta H}_2.$$

3. The hierarchical non-optimal control law applying the projection operator after the second task commands computation, as presented in equation (4.56). As a reminder, the follower command velocities are calculated as:

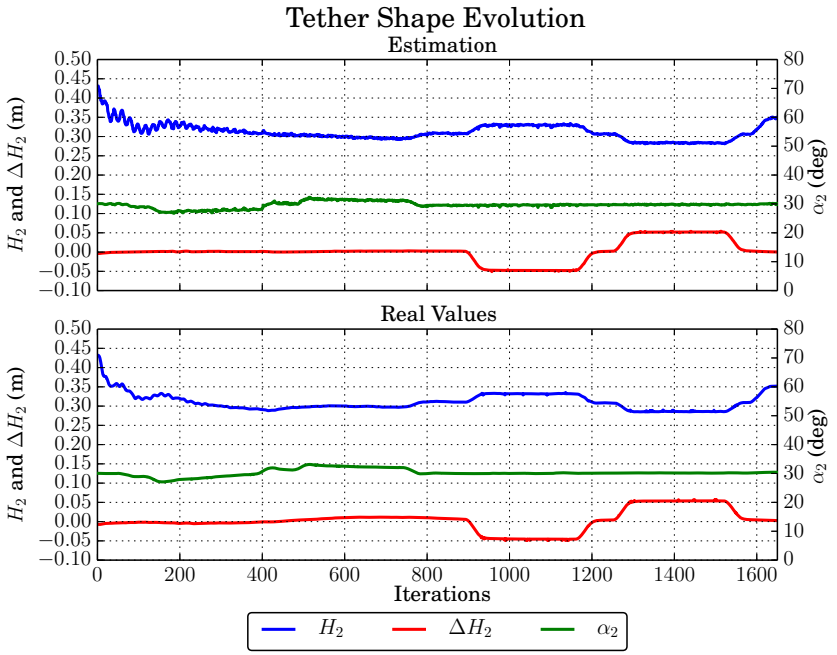
$$^{r_2}\mathbf{v} = -\lambda_1 \left(\mathbf{L}^{H,\alpha}_{r2,2}\right)^{+} \mathbf{e}^{H,\alpha}_2 - \lambda_2 \mathbf{P}_1 \left(\mathbf{L}^{\Delta H}_{r2,2}\right)^{+} \mathbf{e}^{\Delta H}_2.$$

4. The hierarchical optimal controller, as presented in equation (4.61). As a reminder, the follower command velocities are calculated as:

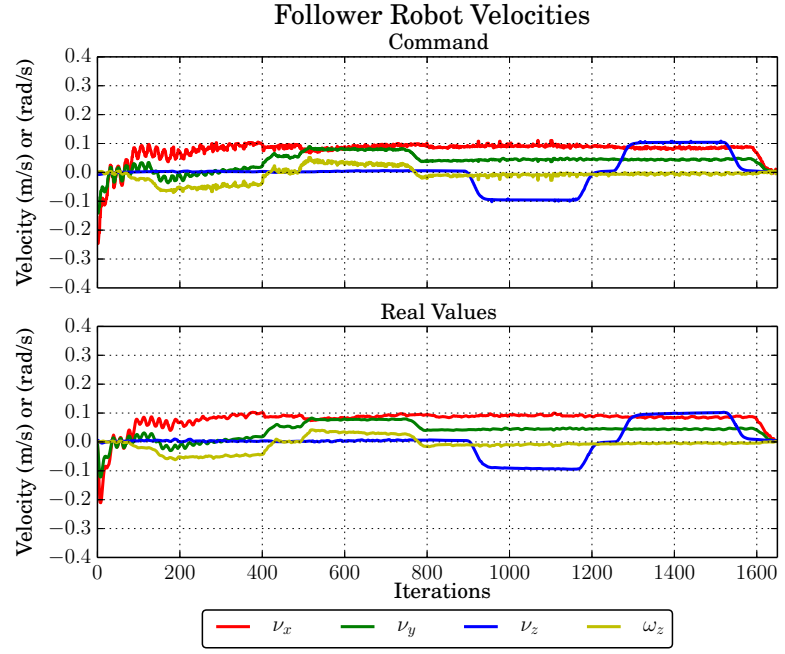
$$^{r_2}\mathbf{v} = -\lambda_1 \left(\mathbf{L}^{H,\alpha}_{r2,2}\right)^{+} \mathbf{e}^{H,\alpha}_2 - \lambda_2 \left(\widetilde{\mathbf{L}}^{\Delta H}_{r2,2}\right)^{+} \widetilde{\mathbf{e}}^{\Delta H}_2,$$

where $\widetilde{\mathbf{L}}^{\Delta H}_{r2,2} = \mathbf{L}^{\Delta H}_{r2,2}\mathbf{P}_1$ and $\dot{\widetilde{\mathbf{e}}}^{\Delta H}_2 = \dot{\mathbf{e}}^{\Delta H}_2 - \mathbf{L}^{\Delta H}_{r2,2}\left(\mathbf{L}^{H,\alpha}_{r2,2}\right)^{+} \dot{\mathbf{e}}^{H,\alpha}_2$.

Simulation Results Figures 4.24, 4.25, 4.26 and 4.27 respectively present the simulation results for the studied control above enumerated as (1), (2), (3) and (4).

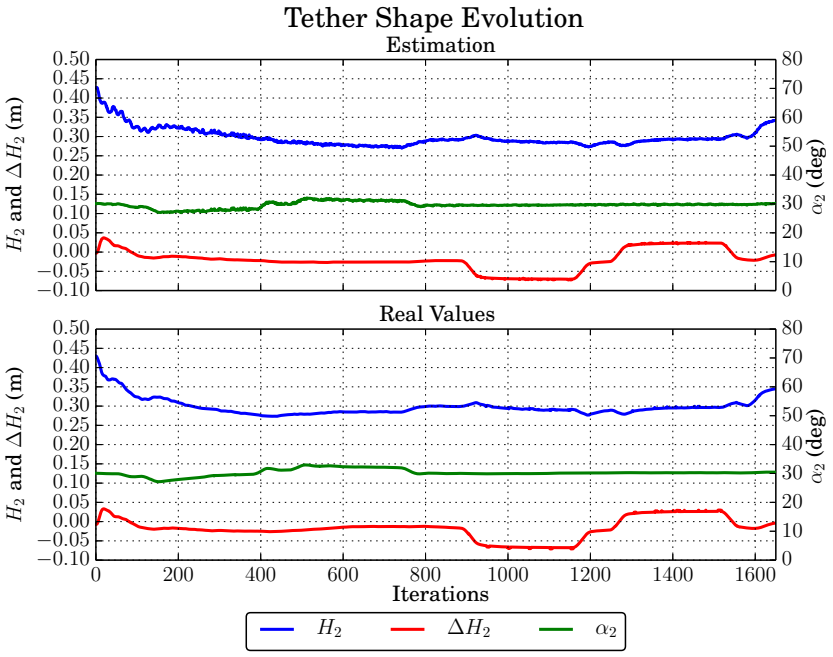


(a)

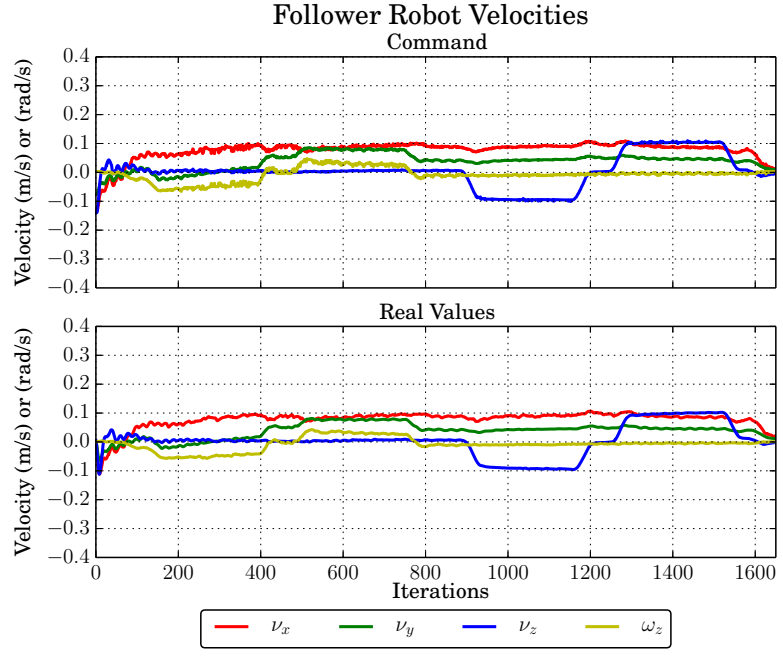


(b)

Figure 4.24: Simulation results for a non-hierarchical control law using the 3×4 interaction matrix. (a) tether shape and (b) follower robot velocity evolution. The tether desired shape is given by $H_2 = 0.35\,\text{m}$, $\alpha_2 = 30^\circ$ and $\Delta H_2 = 0\,\text{m}$.

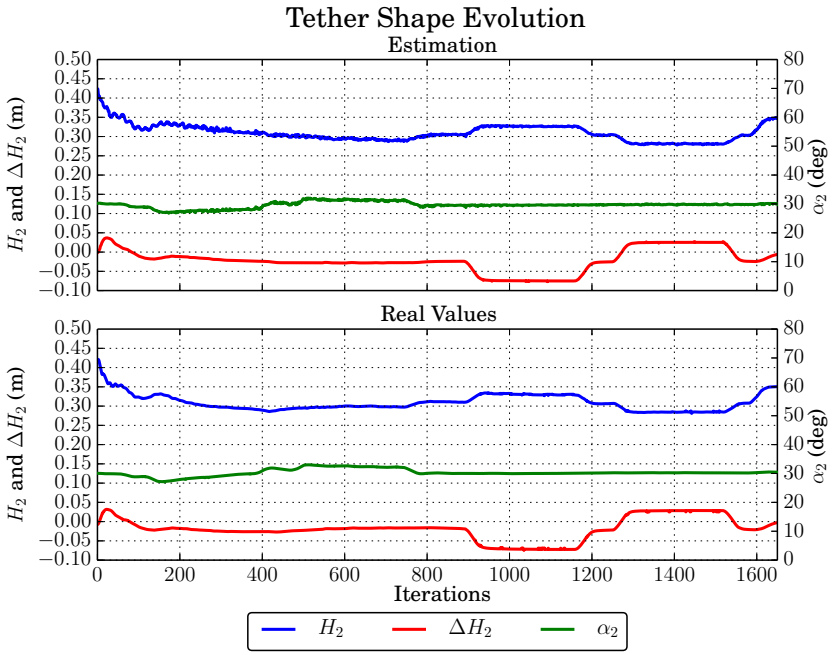


(a)

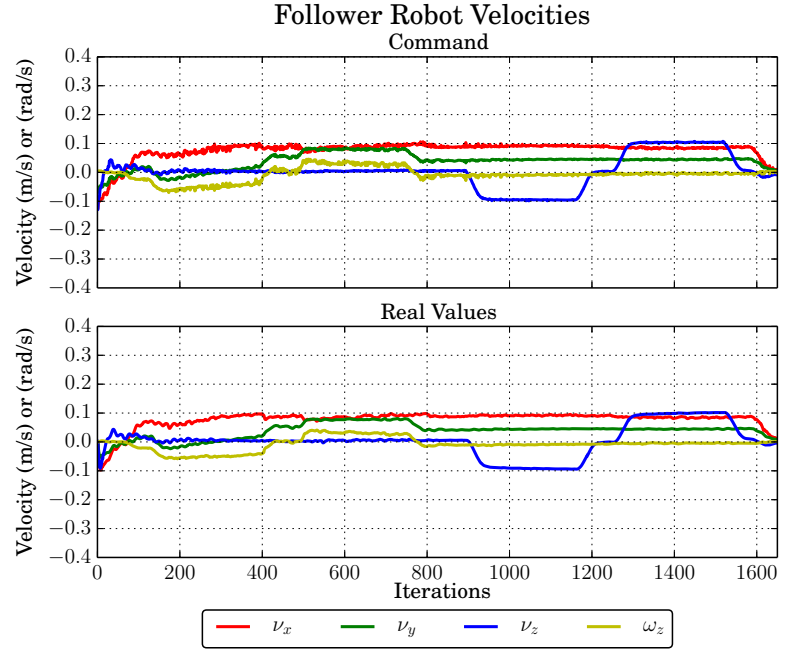


(b)

Figure 4.25: Simulation results for the sum of controllers law with no task priority. (a) tether shape and (b) follower robot velocity evolution. The tether desired shape is given by $H_2 = 0.35\,\text{m}$, $\alpha_2 = 30°$ and $\Delta H_2 = 0\,\text{m}$.

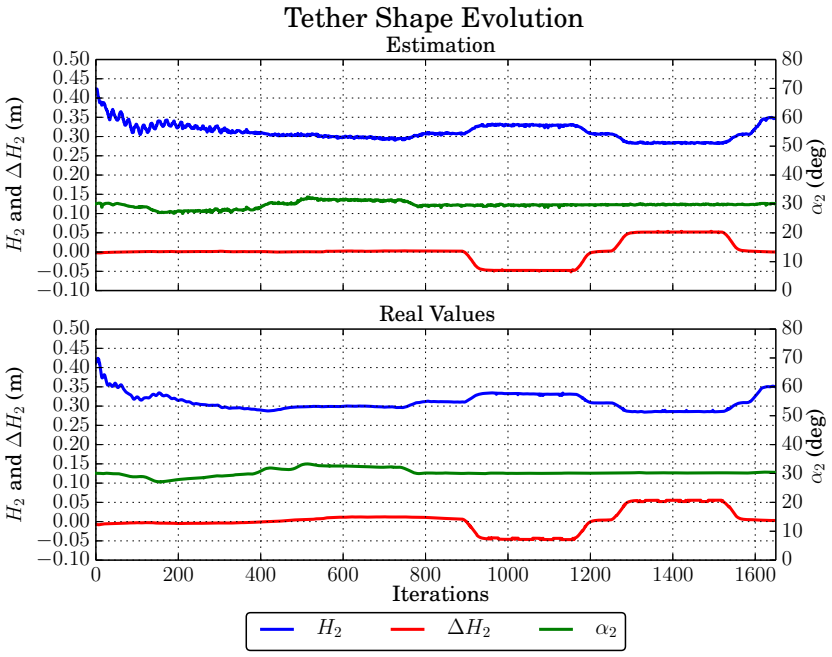


(a)

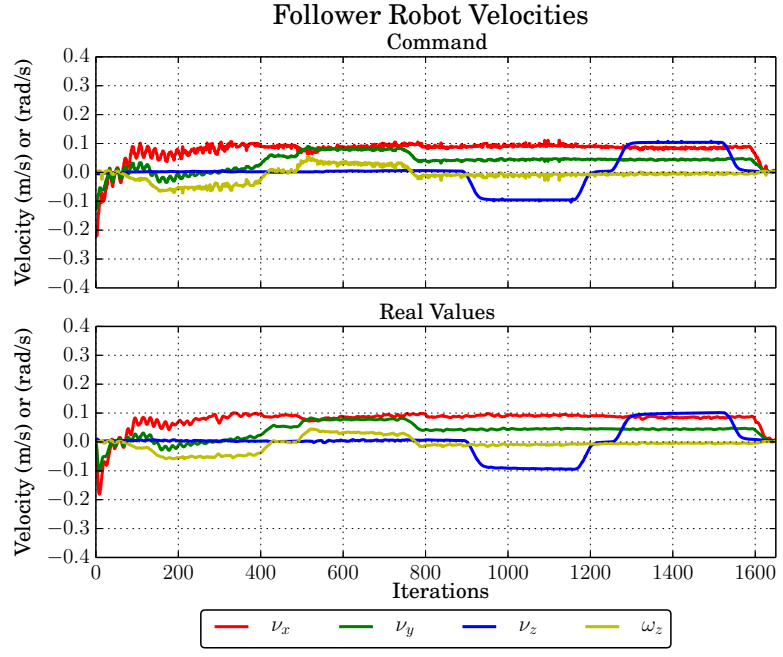


(b)

Figure 4.26: Simulation results for the hierarchical non-optimal controller. (a) tether shape and (b) follower robot velocity evolution. The tether desired shape is given by $H_2 = 0.35\,\text{m}$, $\alpha_2 = 30°$ and $\Delta H_2 = 0\,\text{m}$.



(a)



(b)

Figure 4.27: Simulation results for the hierarchical optimal controller. (a) tether shape and (b) follower robot velocity evolution. The tether desired shape is given by $H_2 = 0.35\,\text{m}$, $\alpha_2 = 30°$ and $\Delta H_2 = 0\,\text{m}$.

At the beginning ($0 - 80$ iterations), we observe, for all control laws, a greater control effort to bring the tether from its initial to its desired slackness: $H^0 = 0.43\,\text{m}$ and $H^* = 0.35\,\text{m}$, respectively. During the lateral motion of the leader ($100 - 750$ iterations), all the four controllers achieve to maintain the tether near its desired shape. A gap between current and desired tether shape is expected, since the leader motion is not considered in the follower commands, which are always lagged.

When the leader executes vertical displacements ($900 - 1500$ iterations), we observe a slight disturbance on the regulation of the tether slackness for the controller using the entire interaction matrix. This occurs because the controller must deal simultaneously with the regulations of H_2 and ΔH_2. Such a disturbance is less remarkable for the sum of controllers, since the regulation of ΔH_2 acts only on the vertical velocity (ν_z), leaving the forward motion (ν_x) free for the slackness regulation. This controller yields, however, a less efficient regulation of ΔH_2. A similar behavior is observed for the hierarchical non-optimal controller, since the second tasks commands are filtered by the projector operator. The hierarchical optimal controller yields similar results to the controller using the entire interaction matrix. Since the robot has enough degrees of freedom to regulate all the features, the execution of the second task can be optimized in the least-square sense.

4.4.3.2 Including the Leader Velocity on the Follower Robot Command

We can also include the leader robot velocities in the follower robots commands. This allows the follower robot to take into account the disturbances generated by the leader motion and thus to regulate the tether shape as if the leader was stopped.

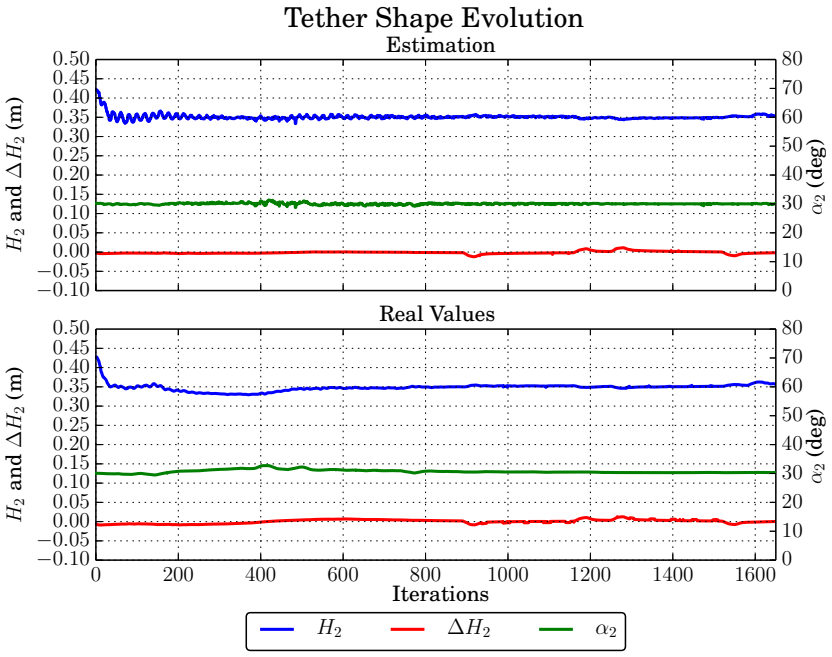
Control Law From equations (4.40) and (4.49), we have:

$$\mathbf{L}_{r_2,2}\left({}^{r_2}\mathbf{v} - {}^{r_2}\mathbf{V}_{r_1}\,{}^{r_1}\mathbf{v}\right) = \dot{\mathbf{e}}_2 \tag{4.62}$$

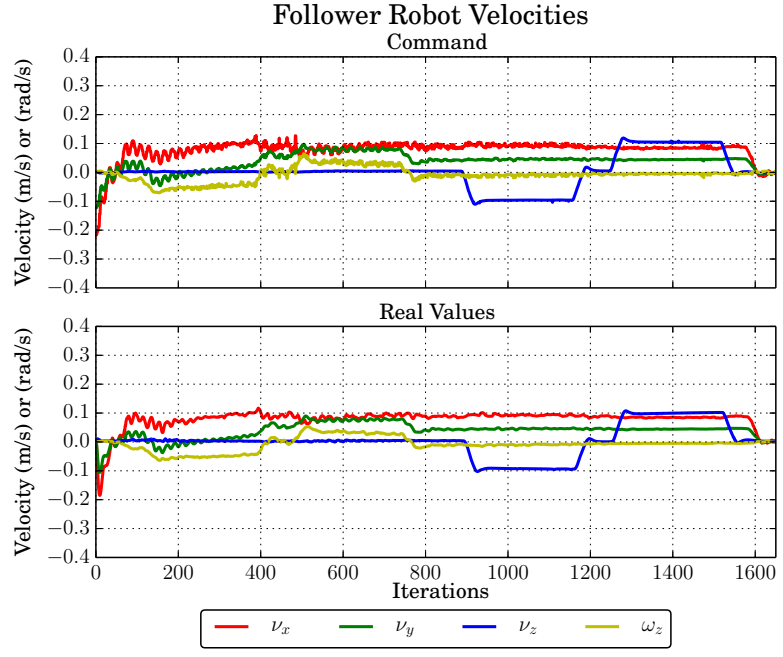
and, setting $\dot{\mathbf{e}}_2 = -\lambda \mathbf{e}_2$ for exponential convergence, with $\lambda \in \mathbb{R}^+$, the follower command velocity can be calculated as follows:

$${}^{r_2}\mathbf{v} = -\lambda \mathbf{L}^+_{r_2,2}\mathbf{e}_2 + {}^{r_2}\mathbf{V}_{r_1}\,{}^{r_1}\mathbf{v}. \tag{4.63}$$

Simulation Results Figure 4.28 presents the simulation results using the control law given in equation (4.63).



(a)



(b)

Figure 4.28: Simulation results for a control law including the leader velocities in the follower commands computation. (a) tether shape and (b) follower robot velocity evolution. The tether desired shape is given by $H_2 = 0.35\,\text{m}$, $\alpha_2 = 30°$ and $\Delta H_2 = 0\,\text{m}$.

Similarly to the previous control schemes, at the beginning of the simulation($0-80$ iterations), we observe a greater control effort of the follower robot to bring the tether from its initial towards its desired slackness: $H_2^0 = 0.45\,\text{m}$ to $H_2^* = 0.35\,\text{m}$. For the rest of the simulation, the tether is maintained near its desired shape. Some little perturbations are observed in the change of direction of the robot leader, around the 400^{th}, 900^{th} and 1200^{th} iteration. The oscillations on the forward velocity command (ν_x) can be reduced through the use of a lower control gain, since the leader motion is included in the follower commands (we used $\lambda = 2.0$ in all simulations).

We note the follower robot executes a forward (ν_x) and lateral (ν_y) velocities during a large part of the simulation (starting from 400^{th} iteration until the end). This occurs because the follower robot must keep its desired orientation with respect to the tether (regulate α_2) while following the leader robot (regulate H_2). Leader and follower robots do not have the same heading.

4.5 Follower Robot Control using Tether Visual Feedback from Both Cameras - Underwater Case

The system we want to control has 4 degrees of freedom (DOF), as presented in Chapter 2 (see Figures 2.10 and 2.11):

1. the tether plane orientation with respect to the leader robot (α_1);

2. the tether plane orientation with respect to the follower robot (α_2);

3. the difference of height between the attachment points ($\Delta H_1 = -\Delta H_2$);

4. the distance between the attachment points ($2D_1 + \Delta D_1 = 2D_2 + \Delta D_2$), which is actually a function of the tether slackness, as defined in equations (4.12) and (4.13).

The control schemes introduced in the previous Section do not allow us to control all the DOF of the system, since the angle α_1 is not regulated. Therefore, the relative pose between leader and follower robot is not fully constrained, as presented in Figure 4.29.

Using only the follower robot camera, we achieve to control three tether features (H_2, α_2 and ΔH_2) through the interaction matrix given in equation (4.44). The rank of this matrix is 3, which does not constrain all the degrees of freedom of the system. In order to constrain the relative pose between the robots, we use the leader camera to obtain additional information and regulate the angle α_1.

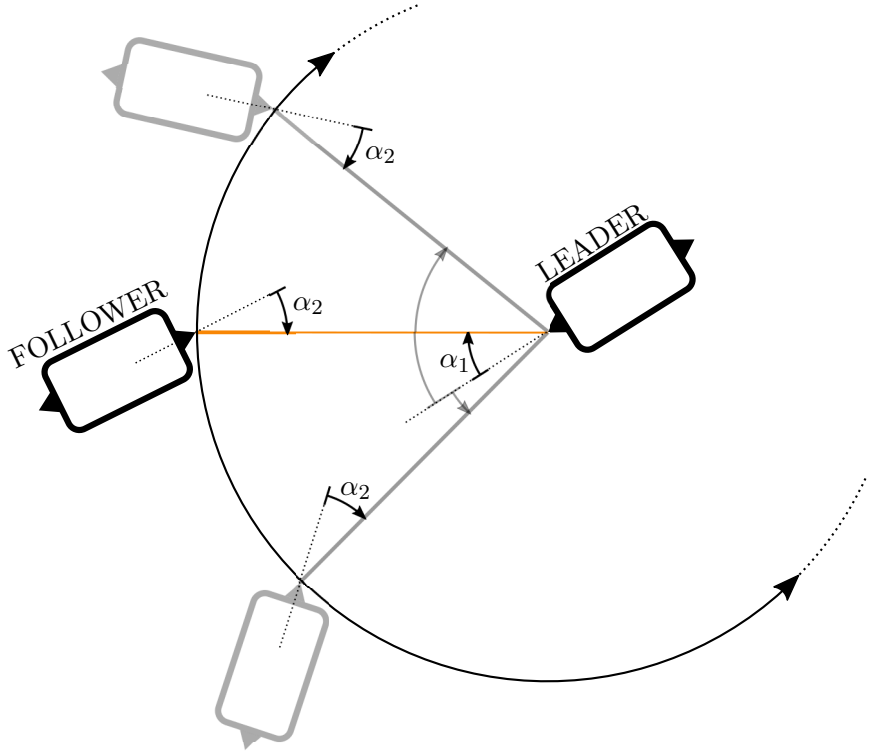


Figure 4.29: Top view of the system composed of leader and follower robots connected by a tether. The relative pose between the robots is not fully constrained without the regulation of angle α_1 between the tether plane and the leader robot longitudinal axis. A circle around the leader attachment point defines an infinity of positions for the follower robot that regulates the features H_2, ΔH_2 and α_2.

In the sequel, we carry one simulation to validate the use of both leader and follower cameras for tether shape regulation. The leader robot is motionless and the simulation environment is the same as described in Section 4.4.

Control Law In order to fully control the 4 DOF of the two robots system, we propose to use the following feature vector:

$$\mathbf{s}_f = \begin{bmatrix} \mathbf{s}_1 \\ \mathbf{s}_2 \end{bmatrix}, \tag{4.64}$$

where $\mathbf{s}_1$ and $\mathbf{s}_2$ were respectively given in equations (4.7) and (4.6). The interaction matrix $\mathbf{L}_{r_2,f}$ relating the follower robot velocity to the feature vector time derivative is given by:

$$\dot{\mathbf{s}}_f = \begin{bmatrix} \mathbf{L}_{r_2,1} \\ \mathbf{L}_{r_2,2} \end{bmatrix} {}^{r_2}\mathbf{v} = \mathbf{L}_{r_2,f}{}^{r_2}\mathbf{v}, \tag{4.65}$$

with

$$\mathbf{L}_{r2,1} = -\mathbf{L}_{r1,1}\,{}^{r_1}\mathbf{V}_{r_2} = \begin{bmatrix} \frac{-\sqrt{1-b_1^2}c_{21}+b_1 s_{21}}{(u_1+p_1)H_{max}} & -\frac{\sqrt{1-b_1^2}s_{21}+b_1 c_{21}}{(u_1+p_1)H_{max}} & -\frac{v_1+q_1}{(u_1+p_1)H_{max}} & -{}^{r_2}X_2\frac{\sqrt{1-b_1^2}s_{21}+b_1 c_{21}}{(u_1+p_1)H_{max}} \\ \frac{b_1\sqrt{1-b_1^2}c_{21}+(1-b_1^2)s_{21}}{2D_1+\Delta D_1} & \frac{b_1\sqrt{1-b_1^2}s_{21}-(1-b_1^2)c_{21}}{2D_1+\Delta D_1} & 0 & {}^{r_2}X_2\frac{b_1\sqrt{1-b_1^2}s_{21}-(1-b_1^2)c_{21}}{2D_1+\Delta D_1} \\ 0 & 0 & \frac{1}{2\Delta H_{max}} & 0 \end{bmatrix}. \tag{4.66}$$

$s_{21} = \sin(\alpha_2 - \alpha_1)$ and $c_{21} = \cos(\alpha_2 - \alpha_1)$. The follower robot velocity commands are calculated through:

$$^{r_2}\mathbf{v} = -\lambda \mathbf{L}^{+}_{r_2,f}\mathbf{e}_f, \tag{4.67}$$

where $\mathbf{e}_f = \mathbf{s}_f - \mathbf{s}^*_f$ and $\mathbf{s}^*_f$ is the desired feature vector.

Simulation Results The following simulation reproduces the same initial scenario of the simulations presented in Section 4.4.2, where only the follower camera is used to regulate the tether shape while the leader robot remains stopped.

The initial tether features are given in Table 4.3. The initial and desired tether shape in the image are depicted in Figure 4.30. At the beginning, the tether is partially visible in both leader and follower cameras.

The tether shape evolution, from visual feedback estimation and simulation ground truth, is depicted in Figure 4.31a. The commanded and executed velocities of the follower robot are shown in Figure 4.31b. The control gain is $\lambda = 0.5$.

	Leader			**Follower**		
	H_1(m)	α_1	ΔH_1(m)	H_2(m)	α_2	ΔH_2(m)
initial	0.50	80°	0.0	0.50	80°	0.0
desired	0.35	30°	−0.20	0.15	30°	0.20

Table 4.3: Initial and desired tether features.

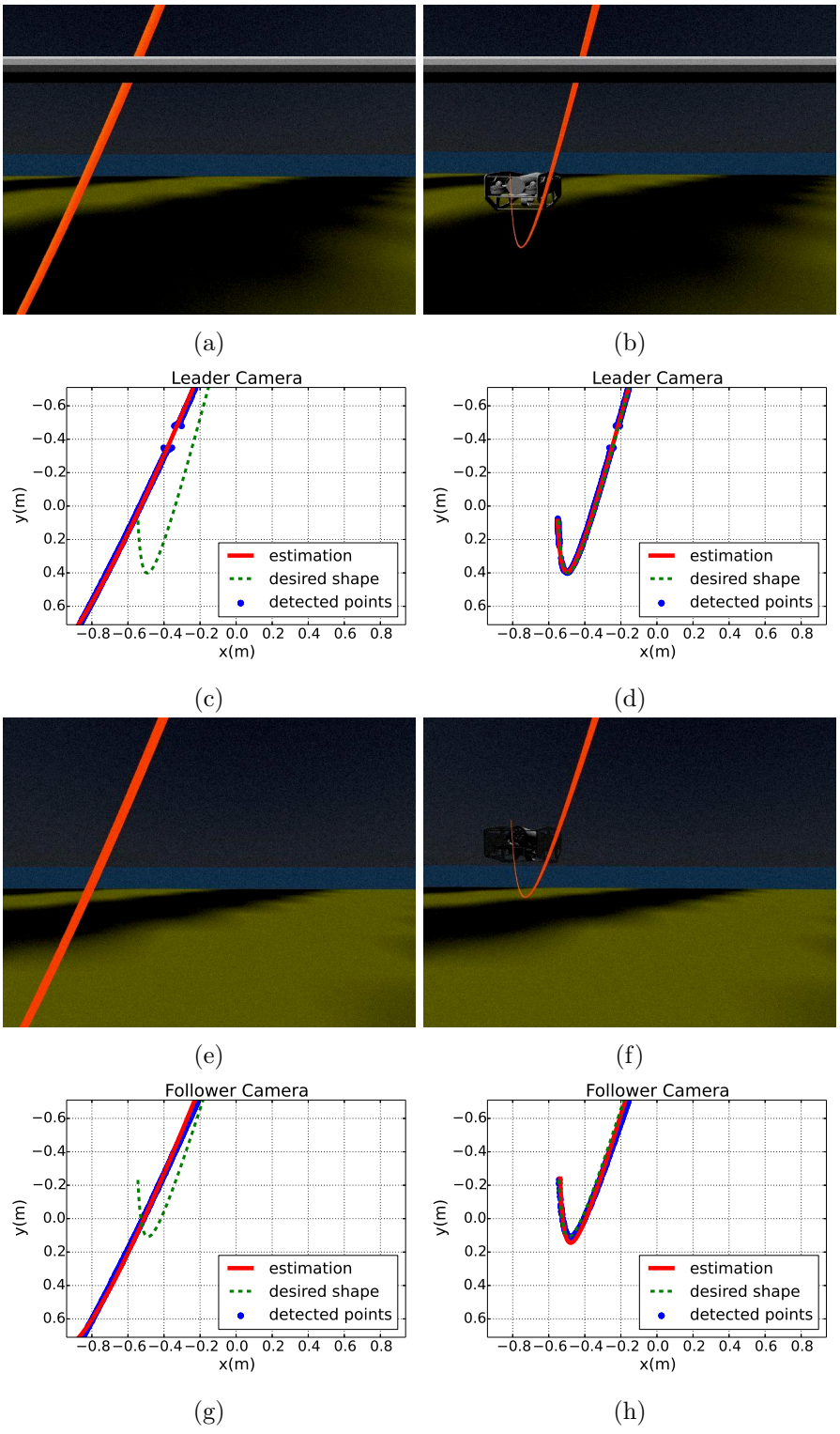
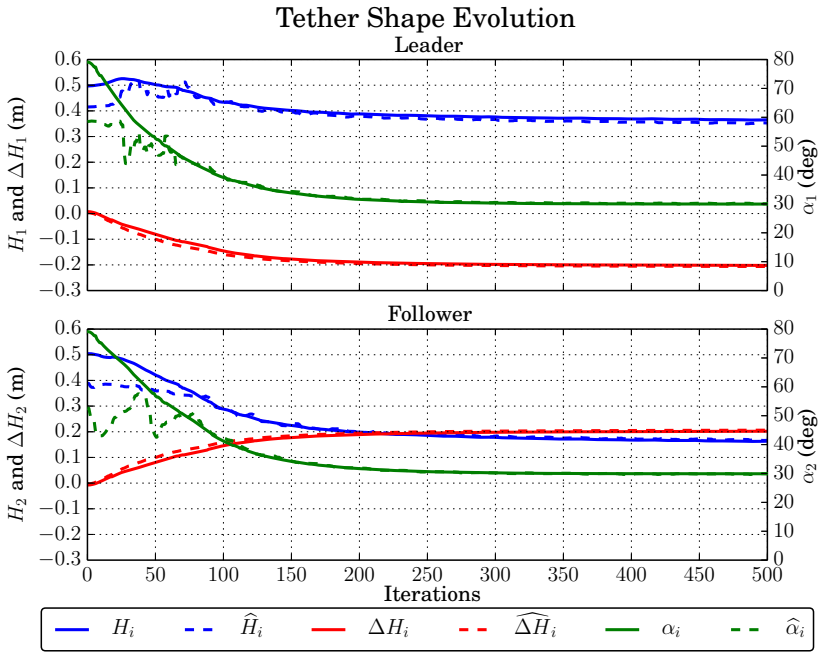
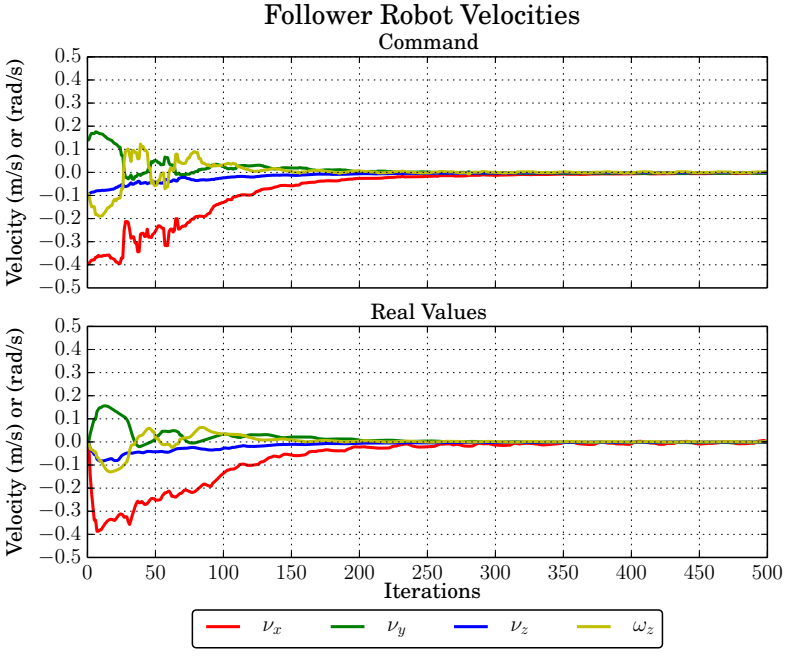


Figure 4.30: Embedded image view of the tether from the (a,b,c,d) leader and (e,f,g,h) follower point of view. The (a,c,e,g) initial and (b,d,f,h) final images of the visual servoing. Figures (a,b,e,f) are the source images while figures (c,d,g,h) depict the tether detected points (in blue), the shape estimation (in red) and the desired shape in the image (in green).



(a)



(b)

Figure 4.31: Simulation results for the controller using both leader and follower cameras. (a) tether shape evolution from leader and follower points of view. Full and dashed lines represent estimated and real simulated values, respectively. (b) follower robot velocity evolution. The tether initial and desired shape are given in Table4.3.

Similar to the results obtained in Section 4.4.2, the tether shape estimation is not accurate at the beginning of the simulation ($0 - 80$ iterations), as presented in Figure 4.31a. Indeed, the tether is almost a straight line in the image at its initial shape (see Figure 4.30e), and the catenary curve fitting procedure cannot yield an accurate estimation of its shape. The curves of estimated and real values of the tether features meet again after the 80^{th} iteration, when the tether is completely visible in both images.

We note a greater oscillation of the estimated values of α_2 when compared to the results obtained in Section 4.4.2.1. This is caused by a greater lateral motion of the follower robot. When only the follower's camera is used to regulate the tether shape, the tether orientation is mainly regulated by a rotation (ω_z) of the follower robot and the lateral motion (ν_y) is almost zero (see Figure 4.18b). When using both cameras, however, the follower robot has to employ a greater lateral motion in order to regulate the angle α_1 simultaneously to α_2. The lateral velocity ν_y causes a swinging motion of the tether plane, which is responsible for the oscillations on α_1 and α_2. These oscillations do not appear on the ground truth curve because the tether attachment points do not oscillate.

Despite the initial shape estimation inaccuracies and the swinging motion of the cable, the control loop brings the tether towards its desired shape in the image (see Figure 4.30h) and the tether features smoothly converge towards their desired values. The robot commanded velocities are nearly zero after the 300^{th} iteration, since the tether reached its desired shape.

4.6 Discussion

The proposed tether shape control scheme has proved useful to properly manage the tether shape. Even in conditions of coarse tether shape estimation, such as cable partial visibility, the controller achieves to bring the tether towards its desired position in the image. The features converge to their desired values as required, with a smooth exponential decay of the error once the tether shape is well fitted in the image. It may occur that at the beginning, the tether is not well fitted due to a lack of tether detected points in the image.

The controller has also proved to be robust to external perturbations, such as unexpected movements of the leader robot. This situation was tested in the case where the leader robot executes a path and the follower must ensure the tether remains at its desired shape. When the leader velocities are neglected by the follower, its commands are lagged, as expected, and the tether is maintained around its desired shape, within a small margin of error. When the leader velocities are included in the follower commands, the tether shape is maintained at its desired shape all along the path.

Four types of controllers were tested. First, the most straightforward approach was implemented, using the entire interaction matrix to regulate all the catenary features simultaneously. This matrix showed to be efficient for the visual servoing control scheme.

Second, a sum of controllers was tested and the features (H_2, α_2) and ΔH_2 were

separately regulated. Such a controller did not show good performances, in particular, the regulation of ΔH_2 introduced an overshoot generated by the vertical velocity commands (ν_z). This is not surprising since both regulations combine their effects in a separate way.

Third, a hierarchical non-optimal control was used for the regulation of (H_2, α_2) as a priority and then ΔH_2 as a secondary task. This solution also showed an overshoot for ΔH_2. Actually, the distribution of the secondary task commands were not optimal with respect to the remaining degrees of freedom of the robot.

Forth, a hierarchical optimal control was tested. This controller optimized the execution of the secondary task, carrying out the best of it without affecting the execution of the priority task. This controller achieved to regulate all the three features properly without overshoot. Convergence performances were observed to be comparable with the controller using the full 3×4 interaction matrix.

Finally, features from the onboard camera of the leader robot were added to the control scheme. This allowed to regulate the angle α_1 between the tether plane and the leader robot longitudinal axis. As a result, the relative position of both robots could be fully constrained. The regulation of the features of both cameras and depth sensor was achieved by stacking the corresponding interaction matrices: the one that regulates the leader features and the other that regulates the follower features. Satisfactory results were also obtained using both cameras, and all the features converged towards their desired values as required, with a smooth exponential decay of the error once the tether was well fitted in the image.

In the following, some perspectives of improvement of our catenary-based control scheme are discussed.

Considering Roll and Pitch Disturbances The robots roll and pitch angles are assumed to be mechanically stabilized or regulated by a low level control law. Disturbances on roll and pitch angles may occur when the robots are deployed in zones with strong disturbances. As our ultimate goal is to deploy a chain of robots in the coastal zone, one of the first possible extensions to be implemented is to consider desired non-zero roll and pitch angles in the control schemes in order to take into account specific diving movements, even though they are not controlled through visual servoing.

Investigate Other Visual Features We investigated the use of the catenary model for a hanging tether cable that links two robots. This model allowed us to parameterize the tether shape accurately, using tether slackness and relative orientation with respect to the robots. Other parametric curves can be investigated. For example, parabolas and splines could be used to fit the tether shape and provide parameters to perform visual control of the tether shape. Recently, Fourier transform coefficients were used in vision-based shape control of linear objects Zhu et al. (2018). This strategy could also be used for tether shape control.

Model-Based Dynamic Control Dynamic interactions between cable and robots can also be modeled. The catenary model can provide information about the forces exerted at its extremities. These could be used in a dynamic control scheme to prevent the robot motion from being disturbed by the cable weight and traction. Moreover, the hydrodynamic modeling of robots and cable can be made in order to achieve a more precise cable handling in water, taking into account its weight and dragging forces. The swinging motion of the tether cable could also be considered.

4.7 Conclusions

In this Chapter we presented 3-D catenary-based visual servoing control schemes used for the shape regulation of a tether linking two robots, a leader and a follower. The tether shape is regulated by the follower robot through the proper displacement of its attachment point.

The tether shape is defined by three parameters: slackness H, orientation angle α and height difference between attachment points ΔH. These parameters are called features, whose values are estimated from sensory feedback data, namely the cameras and pressure sensors embedded on the robots. The pressure sensors give the difference of height between the tether attachment whereas the cameras provide tether detected points used to estimate its slackness and orientation through the curve fitting procedure described in Chapter 3. The tether shape parameters are calculated in relation to the robots. Therefore, there are two slacknesses (H_1 and H_2), two orientations (α_1 and α_2), and two height differences (ΔH_1 and ΔH_2), related to the leader and follower robots. These features are hence used by the control schemes we introduced to regulate the tether shape with respect to one or both robots simultaneously.

The control schemes proposed are proportional control laws based on the interaction matrix relating the follower robot motion to the time derivative of the tether features. Three types of control schemes were tested in the regulation of the tether shape with respect to the follower robot. All the control schemes have proved to properly manage the tether shape, even in conditions of coarse tether shape estimation, such as cable partial visibility. Moreover, they demonstrated to be robust to external perturbations, such as unexpected movements of the leader robot.

First, the entire interaction matrix was used so that all the features were regulated simultaneously. All the features converged towards their desired values as required.

Second, a summing controller was tested and the features (H_2, α_2) and ΔH_2 were separately regulated. As expected, such a controller did not outperform the controller based on the entire interaction matrix.

Third, two hierarchical control laws were tested for the regulation of (H_2, α_2) as a priority and then ΔH_2 as a secondary task. The first one was a hierarchical non-optimal controller whose behavior tended to produce an overshoot of feature ΔH_2, which was also observed with the summing controller. The other one was a hierarchical optimal controller that optimized the execution of the secondary task, carrying out the best of it without affecting the execution of the priority task. All the features were thus

properly regulated, with comparable convergence performances to the controller using the entire matrix.

The simultaneous regulation of the tether features with respect to both leader and follower robots was also achieved with satisfactory results. The relative position of both robots could therefore be fully constrained.

The use of both leader and follower cameras opens interesting perspectives of future works, in the context of the study of high level navigation strategies. Hierarchical control schemes can be used to give priority to the regulation of the tether features with respect to the leader or follower robot, depending on the situation at hand.

Chapter 5

Conclusions

5.1 Summary

In this thesis we investigated the use of camera feedback for the shape control of a tether linking two underwater robots, a leader and a follower. Both robots have the same motion capabilities, and are assumed to remain in the horizontal, with roll and pitch angles autonomously regulated by a high frequency low level controller. Our objective was to prove the feasibility of a visual servoing control scheme that could allow the follower robot to regulate the tether shape releasing the leader robot to explore its surrounding, in autonomy or by teleoperation.

The tether linking the robots is modeled by a catenary. This model is useful to obtain shape parameters of the tether cable. Since our catenary model is symmetric with respect to the attachment point frames, the catenary parameters are calculated relatively to each robot. These parameters, also called catenary features, are defined as: the difference of height between the attachment points, the tether slackness and orientation.

The robots are equipped with depth sensors and cameras positioned near the tether attachment points. The height difference between the attachment points is calculated from the depth sensors measurements, whereas the tether slacknesses and orientations are estimated through an image fitting procedure based on the Gauss-Newton algorithm. Tether points are detected through color segmentation in the images provided by the embedded cameras feedback. The projection of the catenary model in the image plane is thus fitted to the detected points, which yields the estimation of the tether slackness and orientation.

The catenary curve fitting was validated in simulations and it showed to produce accurate results on tether shape estimation. Residual filtering functions were added to the fitting procedure in order to avoid singularities in the Gauss-Newton Jacobian, speed up its computation, and enhance the algorithm robustness to detection noise. Moreover, an initialization heuristic was introduced in order to obtain a starting point closer to the solution and hence avoid local minima.

The estimated catenary features are grouped in a feature vector, which must be

regulated in order to maintain the tether at a desired shape. The tether must be slack enough not to hamper the leader movements. Also, an adequate orientation between the robots and the tether plane must be maintained so that the tether is kept inside the camera field of view. This ensures an accurate shape estimation from camera feedback.

The tether shape is regulated by the follower robot motion, generating a proper displacement of the tether attachment point. The relation between the follower robot motion and the catenary feature time derivatives is given by an interaction matrix, which is the base of the visual servoing control scheme presented in this thesis. A general formalism for the calculation of this interaction matrix was introduced, allowing the regulation of the catenary features with respect to both leader and follower robots.

The catenary-based visual servoing control scheme was validated in simulations. A number of proportional control laws were tested, ranging from the use of one or both robot's cameras to the simultaneous or hierarchical regulation of features. The proposed control scheme has proved to properly manage the tether shape, even in conditions of coarse shape estimation, such as cable partial visibility. Moreover, it has been demonstrated to be robust to external perturbations, such as unexpected movements of the leader robot.

When compared to the control scheme regulating all the catenary features simultaneously, the hierarchical control schemes also yielded satisfactory results, in terms of rapidity, efficiency or robustness to external perturbations. Actually, the interaction matrix we introduced proved to be efficient, since all the features were well tracked by the robot without disturbing each other. The hierarchical control schemes can be useful in the implementation of high level navigation strategies, as will be discussed in the following Section.

In short, the main contributions of this thesis can be enumerated as follows:

1. The real-time estimation of the tether shape through an image fitting procedure based on the catenary model of the tether;

2. The achievement of tether shape regulation through an innovative catenary-based visual servoing control scheme;

3. The study of a number of control laws dedicated to tether shape regulation;

4. The relative positioning of two underwater robots linked by a tether from their embedded cameras and depth sensors feedback.

The scientific publications, workshop participations and scientific popularization activities carried out in this thesis are listed in appendixG.

5.2 Perspectives

Looking forward for tether shape control, one interesting extension of this thesis is to use additional sensory information in order to provide reliable shape estimation for long tether cable lengths (with tens or hundreds of meters length). Cameras have a limited

range, and the tether cable remains visible only up to a few meters away. Moreover, operations in turbid water can make tether point detection difficult, compromising its shape estimation from camera feedback.

As discussed in Chapter 3, additional sensors such as 3-D accelerometers can be added to the tether ends in order to provide additional information about its shape (Merlet, 2018b). These sensors could be useful to measure the departure angles of the tether cables, which can be used to calculate its slackness. Moreover, the accelerometers may provide information about the swinging motion of the tether cable, which was not considered in the control schemes proposed.

Even considering the case of large cables shape regulation, the cameras could yield exploitable information about the cable orientation with respect to the robots. The use of both leader and follower cameras may also provide redundant information about the tether slackness, which can be used together with accelerometers data in order to enhance the reliability of cable shape estimation.

The concept of catenary-based tether shape control for a pair of robots may also be extended for a chain composed of several underwater robots deployed by a surface vessel. The visual servoing control scheme that we introduced could be used to regulate the portion of tether cable connecting successive robots. Since the motion of one robot has an effect on the others, a high level control must be set up to avoid platooning effect.

The navigation of the whole chain of robots is an interesting research theme to be developed. The high level controller must manage the relative position between the robots depending on the situation at hand. If small cavities are to be crossed, the robots must be mostly aligned and the tether not too slack so as not to drag on the floor. Otherwise, when exploring open sites, it would be better to have the robots more spaced laterally in order to have a wider perception of the environment. The hierarchical control schemes can be used so that priority can be given to the orientation or slackness, depending on the situation.

Other geometric configurations than a chain of robots can be used. Triangular and squared grids can be interesting formations to explore open sites and achieve a more distributed perception of the environment.

Possible applications of this work include the exploration of coastal zones by small underwater tethered robots, called mini-ROVs. Small underwater robots are required to carry out exploration missions of these zones due to the low depth. Alone, such robots cannot counterbalance the disturbance caused by tides and currents acting on the tether cable. Together, however, they can more efficiently control the tether shape. Moreover, the use of multiple robots connected by a tether yield a distributed perception of the underwater environment for the human operator on the surface, which is interesting for cartography purposes and synchronized data collection of the environment.

Catenary-based tether shape control can be also useful in the context of cluttered environments exploration, such as underwater caves and submerged mining sites. In such situations, the correct management of the tether shape is crucial for the success of the mission, preventing robots from being lost due to tether entanglement with obstacles.

In a broader sense, this thesis proved that the tether cable can be used for the relative positioning between robots, or between robot and a stationary base. This can be used, for example, by a robot that anchors itself to the seabed relief or underwater structure in order to have its position stabilized for its mission execution. Another example of application would be the relative positioning between a surface vessel and a scout robot.

Appendix A

Position and Orientation Representation

In a 3-dimensional Euclidean space, noted $\mathbb{E}^3$, a coordinate reference frame is defined as $\mathcal{F}_a(\mathbf{P}_a, \mathbf{i}_a, \mathbf{j}_a, \mathbf{k}_a)$ and consists of an origin, denoted $\mathbf{P}_a$, and a triad of mutually orthogonal basis vectors, denoted $(\mathbf{i}_a, \mathbf{j}_a, \mathbf{k}_a)$, that are all fixed within a particular body (Siciliano and Khatib, 2016, chap. 1).

The position of a 3-D point, center of frame $\mathcal{F}_b$, relative to another coordinate frame $\mathcal{F}_a$ can be denoted by the 3×1 vector

$$ {}^a\overline{\mathbf{P}}_b = \begin{bmatrix} {}^aX_b \\ {}^aY_b \\ {}^aZ_b \end{bmatrix} . \tag{A.1} $$

The components of this vector are the Cartesian coordinates of the center of $\mathcal{F}_b$ in the frame $\mathcal{F}_a$, which are the projections the vector ${}^a\overline{\mathbf{P}}_b$ onto the corresponding axes $\mathbf{i}, \mathbf{j}, \mathbf{k}$.

The orientation of frame $\mathcal{F}_b$ relative to frame $\mathcal{F}_a$ can be denoted by expressing the basis vectors $(\mathbf{i}_b, \mathbf{j}_b, \mathbf{k}_b)$ in terms of the basis vectors $(\mathbf{i}_a, \mathbf{j}_a, \mathbf{k}_a)$. This yields $({}^a\mathbf{i}_b, {}^a\mathbf{j}_b, {}^a\mathbf{k}_b)$, which when written together as a 3×3 matrix is known as the rotation matrix

$$ {}^a\mathbf{R}_b = \begin{bmatrix} \mathbf{i}_b \cdot \mathbf{i}_a & \mathbf{j}_b \cdot \mathbf{i}_a & \mathbf{k}_b \cdot \mathbf{i}_a \\ \mathbf{i}_b \cdot \mathbf{j}_a & \mathbf{j}_b \cdot \mathbf{j}_a & \mathbf{k}_b \cdot \mathbf{j}_a \\ \mathbf{i}_b \cdot \mathbf{k}_a & \mathbf{j}_b \cdot \mathbf{k}_a & \mathbf{k}_b \cdot \mathbf{k}_a \end{bmatrix} , \tag{A.2} $$

which is an special orthogonal matrix, such that ${}^a\mathbf{R}_b^T = {}^a\mathbf{R}_b^{-1}$ and $\det {}^a\mathbf{R}_b = 1$.

An elementary rotation of $\mathcal{F}_b$ about the $\mathbf{k}$ axis of $\mathcal{F}_a$ through an angle θ is

$$ {}^a\mathbf{R}_b(\mathbf{k}, \theta) = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} , \tag{A.3} $$

while the same rotation around the $\mathbf{j}$ axis is

$$ {}^a\mathbf{R}_b(\mathbf{j}, \theta) = \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ 0 & 1 & 0 \\ -\sin\theta & \cos\theta & 0 \end{bmatrix} , \tag{A.4} $$

and around **i** axis is

$$ {}^{a}\mathbf{R}_{b}(\mathbf{i},\theta) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta & 0 \end{bmatrix}. \tag{A.5} $$

The Euler angle used to define the orientation of solid bodies in the space are not uniquely defined. Several formulations are proposed in the literature, among which we mainly distinguish roll-yaw-roll, roll-pitch-roll and roll-pitch-yaw. Any rotation matrix of $\mathbb{E}^3$ can be expressed in a form of a product of the three matrices as follows (Jaulin, 2015):

$$ {}^{b}\mathbf{R}_{a} = {}^{a}\mathbf{R}_{b}(\mathbf{k},\theta_k){}^{a}\mathbf{R}_{b}(\mathbf{j},\theta_j){}^{a}\mathbf{R}_{b}(\mathbf{i},\theta_i) \tag{A.6} $$

Position vectors and rotation matrices can be combined together in an homogeneous transformation in order to represent position and orientation in a compact notation. Any 3-D point vector ${}^{i}\overline{\mathbf{P}}_{a}$ expressed relative to frame $\mathcal{F}_i$ can be expressed relative to $\mathcal{F}_j$ if the position and orientation of $\mathcal{F}_i$ are known relative to the $\mathcal{F}_j$:

$$ {}^{j}\overline{\mathbf{P}}_{a} = {}^{j}\mathbf{R}_{i}{}^{i}\overline{\mathbf{P}}_{a} + {}^{j}\overline{\mathbf{P}}_{i}. \tag{A.7} $$

This equation can be written

$$ {}^{j}\mathbf{P}_{a} = {}^{j}\mathbf{M}_{i}{}^{i}\mathbf{P}_{a}, \tag{A.8} $$

where

$$ {}^{j}\mathbf{M}_{i} = \begin{bmatrix} {}^{j}\mathbf{R}_{i} & {}^{j}\overline{\mathbf{P}}_{i} \\ \mathbf{0}^{T} & 1 \end{bmatrix} $$

is the 4×4 homogeneous matrix that transforms vectors from coordinate frame $\mathcal{F}_i$ to coordinate frame $\mathcal{F}_j$, and ${}^{j}\mathbf{P}_{a} = [{}^{j}\overline{\mathbf{P}}_{a}\ 1]^{T}$ and ${}^{i}\mathbf{P}_{a} = [{}^{i}\overline{\mathbf{P}}_{a}\ 1]^{T}$ are homogeneous representations of position vectors ${}^{j}\overline{\mathbf{P}}_{a}$ and ${}^{i}\overline{\mathbf{P}}_{a}$. The homogeneous matrix ${}^{i}\mathbf{M}_{j}$ is invertible and the inverse coordinates transformation from frame $\mathcal{F}_j$ to frame $\mathcal{F}_i$ is represented by:

$$ {}^{j}\mathbf{M}_{i}^{-1} = {}^{i}\mathbf{M}_{j} = \begin{bmatrix} {}^{j}\mathbf{R}_{i}^{T} & -{}^{j}\mathbf{R}_{i}^{T}{}^{j}\overline{\mathbf{P}}_{i} \\ \mathbf{0}^{T} & 1 \end{bmatrix}. $$

The transformation between frames can be expressed as a composition of transformations, such that:

$$ {}^{k}\mathbf{M}_{i} = {}^{k}\mathbf{M}_{j}{}^{j}\mathbf{M}_{i}. $$

Appendix B

Expression of the catenary parameter C

This appendix gives details about the calculation of the catenary parameter C as a function of its slackness H, attachment point difference of height ΔH and cable total length L.

The catenary shape of a hanging cable can be obtained from a set of three equations. The first two equations express that the tether extremities are points belonging to the catenary. The third equation states that the total length of the cable is a known constant. These equations are given here above:

$$H = \frac{1}{C}\left[\cosh(CD) - 1\right] \tag{B.1}$$

$$H + \Delta H = \frac{1}{C}\left[\cosh(C(D + \Delta D) - 1\right] \tag{B.2}$$

$$L = \frac{1}{C}\left[\sinh(C(D + \Delta D)) + \sinh(CD)\right], \tag{B.3}$$

which can be rewritten as:

$$CH + 1 = \cosh(CD) \tag{B.4}$$

$$C(H + \Delta H) + 1 = \cosh(C(D + \Delta D) \tag{B.5}$$

$$CL = \sinh(C(D + \Delta D)) + \sinh(CD). \tag{B.6}$$

We must eliminate D between the equations. Using the identity equation

$$\cosh^2(C(D + \Delta D)) - \sinh^2(C(D + \Delta D)) = 1 \tag{B.7}$$

with equations (B.5) and (B.6) leads to (this means that there can be a minus sign in front of $\sinh(C(D + \Delta D))$ in Eq. (B.6)):

$$\left[C(H + \Delta H) + 1\right]^2 - \left[CL - \sinh(CD)\right]^2 = 1, \tag{B.8}$$

that can be developed to

$$\left[C(H + \Delta H) + 1\right]^2 - \left[(CL)^2 + \sinh^2(CD) - 2CL\sinh(CD)\right] = 1. \tag{B.9}$$

From equations (B.4) and (B.7) we have

$$\sinh^2(CD) = (CH+1)^2 - 1, \tag{B.10}$$

that can be introduced in equation (B.9) yielding:

$$[(CH+1) + C\Delta H)]^2 - (CL)^2 - (CH+1)^2 + 2CL\sinh(CD) = 0. \tag{B.11}$$

This equation can be further developed to:

$$(C\Delta H)^2 + 2C\Delta H(CH+1) - (CL)^2 + 2CL\sinh(CD) = 0 \tag{B.12}$$

$$C(L^2 - \Delta H^2 - 2H\Delta H) - 2\Delta H = 2L\sinh(CD). \tag{B.13}$$

Taking the square of equation (B.13) leads to (this means that there can be a minus sign in front of $\sinh(CD)$ in the equation above but C must remain positive):

$$\left[C(L^2 - \Delta H^2 - 2H\Delta H) - 2\Delta H\right]^2 = 4L^2\left[(CH)^2 + 2CH\right]. \tag{B.14}$$

This is a second order equation in C:

$$\overline{A}.C^2 + \overline{B}.C + \overline{C} = 0, \tag{B.15}$$

with

$$\begin{aligned}\overline{A} &= (L^2 - \Delta H^2 - 2H\Delta H)^2 - 4L^2H^2 \\ &= (L^2 - \Delta H^2)\left[L^2 - (2H + \Delta H)^2\right] \qquad \text{(B.16)}\\ \overline{B} &= -4\Delta H(L^2 - \Delta H^2 - 2H\Delta H) - 8HL^2 \\ &= -4(L^2 - \Delta H^2)(2H + \Delta H) \qquad \text{(B.17)}\\ \overline{C} &= 4\Delta H^2. \qquad \text{(B.18)}\end{aligned}$$

The determinant Δ is:

$$\begin{aligned}\Delta &= \overline{B}^2 - 4\overline{AC} \\ &= 64HL^2(L^2 - \Delta H^2)(H + \Delta H). \qquad \text{(B.19)}\end{aligned}$$

There exist solutions if $\Delta > 0$, i.e. if:

$$L > |\Delta H| \tag{B.20}$$

$$H + \Delta H > 0 \tag{B.21}$$

H is always positive because the attachment point $\mathbf{P}_2$ is always above the lowest point $\mathbf{P}_0$ of the catenary. Therefore, if a solution exists, we necessarily have $\overline{B} < 0$.

In this case the two solutions can be expressed as:

$$C_1 = \frac{-\overline{B} - \sqrt{\Delta}}{2\overline{A}} \tag{B.22}$$

$$C_2 = \frac{-\overline{B} + \sqrt{\Delta}}{2\overline{A}} \tag{B.23}$$

Since $\overline{C} > 0$,

$$sign(C_1 C_2) = sign(\overline{CA}) = sign(\overline{A})$$

- if $\overline{A} < 0$, there is only one positive solution for C, which is C_1 ($\sqrt{\Delta} = \sqrt{\overline{B} - 4\overline{AC}} > -\overline{B}$).
- if $\overline{A} > 0$, there are two positive solutions C_1 and C_2 with $C_1 < C_2$. The catenary with coefficient C_2 is the tightest one ($ch(C_2 X_1)$ behaves like $e^{C_2 X_1}$ when X_1 tends to infinity, which increases much faster than $e^{C_1 X_1}$ as X_1 increases.). Figures B.1 and B.2 illustrate both solutions, the catenary C_1 does not allow the lowest point to belong to the umbilical, whereas the catenary C_2 does.

If we want to have the tightest catenary as a solution, we must have $\overline{A} > 0$, i.e.

$$L > 2H + \Delta H$$

since $2H + \Delta H > 0$.

If $\Delta H = 0$, we get:

$$C_2 = \frac{-\overline{B} + \sqrt{\Delta}}{2\overline{A}} \tag{B.24}$$

$$= \frac{8L^2 H + \sqrt{64H^2 L^4}}{2L^2(L^2 - 4H^2)} \tag{B.25}$$

$$= \frac{16L^2 H}{2L^2(L^2 - 4H^2)} \tag{B.26}$$

$$= \frac{8H}{L^2 - 4H^2} \tag{B.27}$$

If we replace L by $2R$, we have:

$$C_2 = \frac{8H}{4R^2 - 4H^2} = \frac{2H}{R^2 - H^2} \tag{B.28}$$

As a conclusion, the solution for C to have the catenary with the lowest point belonging to the umbilical is:

$$C = \frac{2(L^2 - \Delta H^2)(2H + \Delta H) + 4L\sqrt{H(L^2 - \Delta H^2)(H + \Delta H)}}{(L^2 - \Delta H^2)\left[L^2 - (2H + \Delta H)^2\right]} \tag{B.29}$$

$$= \frac{2}{L^2 - (2H + \Delta H)^2}\left(2H + \Delta H + 2L\sqrt{H\frac{H + \Delta H}{L^2 - \Delta H^2}}\right) \tag{B.30}$$

Let us denote $\alpha = \frac{2H+\Delta H}{L}$,

$$C = \frac{2/L^2}{1-\alpha^2}\left(L\alpha + 2L\sqrt{\frac{(H+1/2.\Delta H)^2 - \Delta H^2/4}{L^2-\Delta H^2}}\right) \tag{B.31}$$

$$C = \frac{2/L}{1-\alpha^2}\left(\alpha + \sqrt{\frac{(2H+\Delta H)^2 - \Delta H^2}{L^2-\Delta H^2}}\right) \tag{B.32}$$

$$C = \frac{2}{1-\alpha^2}\left(\alpha/L + \sqrt{\frac{\alpha^2 - \Delta H^2/L^2}{L^2-\Delta H^2}}\right) \tag{B.33}$$

If we denote $\beta = \Delta H/L$:

$$C = \frac{1}{L}\frac{2}{1-\alpha^2}\left(\alpha + \sqrt{\frac{\alpha^2-\beta^2}{1-\beta^2}}\right) \tag{B.34}$$

with the following conditions:

$$H \geq 0 \tag{B.35}$$

$$H + \Delta H \geq 0 \tag{B.36}$$

$$L \geq max\left\{|\Delta H|, 2H + \Delta H\right\} \tag{B.37}$$

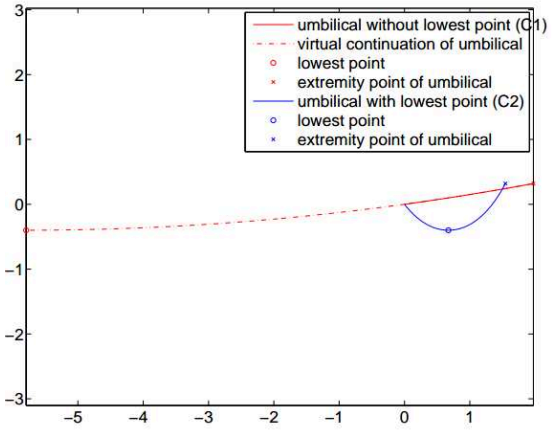


Figure B.1: Catenary solutions with DH positive. $DH = 0.32$. $H = 0.4$. $L = 2$. $(0, 0)$ is the attachment point of the umbilical to the following robot.

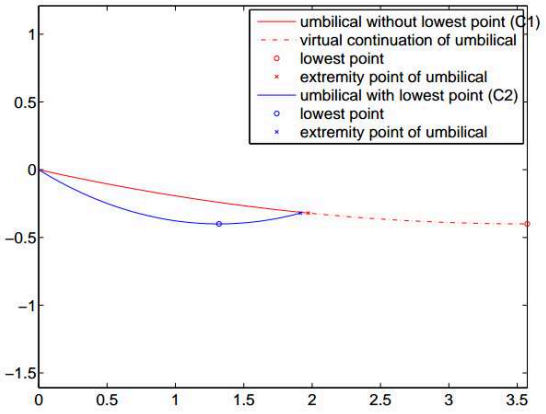


Figure B.2: Catenary solutions with DH negative. $DH = -0.32$. $H = 0.4$. $L = 2$. $(0, 0)$ is the attachment point of the umbilical to the following robot.

Appendix C

Kinematic Equations with Twist Matrix: General Case

Here we establish the relationship between $\mathbf{T}_{r_1,2}$ and $\mathbf{T}_{r_2,2}$. From equation (2.87)) we have:

$$\boxed{{}^2\dot{\overline{\mathbf{P}}}_1 = \mathbf{T}_{r_1,2}{}^{r_1}\mathbf{v} + \mathbf{T}_{r_2,2}{}^{r_2}\mathbf{v}.}$$

where

$$\mathbf{T}_{r_1,2} = \left[{}^2\mathbf{R}_{r_2}{}^{r_2}\mathbf{R}_{r_1}, \quad -[{}^2\mathbf{R}_{r_2}{}^{r_2}\mathbf{R}_{r_1}{}^{r_1}\overline{\mathbf{P}}_1]^{\times 2}\mathbf{R}_{r_2}{}^{r_2}\mathbf{R}_{r_1}\right] \tag{C.1}$$

$$\mathbf{T}_{r_2,2} = -\left[{}^2\mathbf{R}_{r_2}, \quad -[{}^2\mathbf{R}_{r_2}{}^{r_2}\overline{\mathbf{P}}_2]^{\times 2}\mathbf{R}_{r_2} - [{}^2\overline{\mathbf{P}}_1]^{\times 2}\mathbf{R}_{r_2}\right] \tag{C.2}$$

where $[\mathbf{X}]^{\times}$ is the skew matrix defined from vector $\mathbf{X}$.

Let us introduce the twist matrix defined as follows:

$${}^{r_2}\mathbf{V}_{r_1} = \begin{bmatrix} {}^{r_2}\mathbf{R}_{r_1}, & [{}^{r_2}\overline{\mathbf{P}}_{r_1}]^{\times r_2}\mathbf{R}_{r_1} \\ \mathbf{0} & {}^{r_2}\mathbf{R}_{r_1} \end{bmatrix}.$$

$$\begin{aligned}\mathbf{T}_{r_2,2}{}^{r_2}\mathbf{V}_{r_1} &= -\left[{}^2\mathbf{R}_{r_2}, \quad -[{}^2\mathbf{R}_{r_2}{}^{r_2}\overline{\mathbf{P}}_2]^{\times 2}\mathbf{R}_{r_2} - [{}^2\overline{\mathbf{P}}_1]^{\times 2}\mathbf{R}_{r_2}\right]\begin{bmatrix} {}^{r_2}\mathbf{R}_{r_1}, & [{}^{r_2}\overline{\mathbf{P}}_{r_1}]^{\times r_2}\mathbf{R}_{r_1} \\ \mathbf{0} & {}^{r_2}\mathbf{R}_{r_1} \end{bmatrix} \\ &= -\left[{}^2\mathbf{R}_{r_2}{}^{r_2}\mathbf{R}_{r_1}, \quad {}^2\mathbf{R}_{r_2}[{}^{r_2}\overline{\mathbf{P}}_{r_1}]^{\times r_2}\mathbf{R}_{r_1} - [{}^2\mathbf{R}_{r_2}{}^{r_2}\overline{\mathbf{P}}_2]^{\times 2}\mathbf{R}_{r_2}{}^{r_2}\mathbf{R}_{r_1} - [{}^2\overline{\mathbf{P}}_1]^{\times 2}\mathbf{R}_{r_2}{}^{r_2}\mathbf{R}_{r_1}\right]\end{aligned}$$

We have:

$${}^2\mathbf{R}_{r_2}[{}^{r_2}\overline{\mathbf{P}}_{r_1}]^{\times} = [{}^2\mathbf{R}_{r_2}{}^{r_2}\overline{\mathbf{P}}_{r_1}]^{\times 2}\mathbf{R}_{r_2}, \tag{C.3}$$

This allows to apply the skew matrix in frame $\mathcal{F}_2$, instead of $\mathcal{F}_{r2}$.

$$\begin{aligned}
-\mathbf{T}_{r_2,2}{}^{r_2}\mathbf{V}_{r_1} &= \left[{}^2\mathbf{R}_{r_2}{}^{r_2}\mathbf{R}_{r_1}, \quad {}^2\mathbf{R}_{r_2}[{}^{r_2}\overline{\mathbf{P}}_{r_1}]^{\times}{}^{r_2}\mathbf{R}_{r_1} - [{}^2\mathbf{R}_{r_2}{}^{r_2}\overline{\mathbf{P}}_2]^{\times}{}^2\mathbf{R}_{r_2}{}^{r_2}\mathbf{R}_{r_1} - [{}^2\overline{\mathbf{P}}_1]^{\times}{}^2\mathbf{R}_{r_2}{}^{r_2}\mathbf{R}_{r_1}\right] \\
&= \left[{}^2\mathbf{R}_{r_2}{}^{r_2}\mathbf{R}_{r_1}, \quad [{}^2\mathbf{R}_{r_2}{}^{r_2}\overline{\mathbf{P}}_{r_1}]^{\times}{}^2\mathbf{R}_{r_2}{}^{r_2}\mathbf{R}_{r_1} - [{}^2\mathbf{R}_{r_2}{}^{r_2}\overline{\mathbf{P}}_2]^{\times}{}^2\mathbf{R}_{r_2}{}^{r_2}\mathbf{R}_{r_1} - [{}^2\overline{\mathbf{P}}_1]^{\times}{}^2\mathbf{R}_{r_2}{}^{r_2}\mathbf{R}_{r_1}\right] \\
&= \left[{}^2\mathbf{R}_{r_2}{}^{r_2}\mathbf{R}_{r_1}, \quad \left\{[{}^2\mathbf{R}_{r_2}{}^{r_2}\overline{\mathbf{P}}_{r_1}]^{\times} - [{}^2\mathbf{R}_{r_2}{}^{r_2}\overline{\mathbf{P}}_2]^{\times} - [{}^2\overline{\mathbf{P}}_1]^{\times}\right\}{}^2\mathbf{R}_{r_2}{}^{r_2}\mathbf{R}_{r_1}\right] \\
&= \left[{}^2\mathbf{R}_{r_2}{}^{r_2}\mathbf{R}_{r_1}, \quad \left[{}^2\mathbf{R}_{r_2}{}^{r_2}\overline{\mathbf{P}}_{r_1} - {}^2\mathbf{R}_{r_2}{}^{r_2}\overline{\mathbf{P}}_2 - {}^2\overline{\mathbf{P}}_1\right]^{\times}{}^2\mathbf{R}_{r_2}{}^{r_2}\mathbf{R}_{r_1}\right] \\
&= \left[{}^2\mathbf{R}_{r_2}{}^{r_2}\mathbf{R}_{r_1}, \quad \left[{}^2\mathbf{R}_{r_2}{}^{r_2}\overline{\mathbf{P}}_{r_1} - {}^2\mathbf{R}_{r_2}{}^{r_2}\overline{\mathbf{P}}_2 - {}^2\mathbf{R}_{r_2}{}^{r_2}\left(\mathbf{P}_2\mathbf{P}_1\right)\right]^{\times}{}^2\mathbf{R}_{r_2}{}^{r_2}\mathbf{R}_{r_1}\right] \\
&= \left[{}^2\mathbf{R}_{r_2}{}^{r_2}\mathbf{R}_{r_1}, \quad \left[{}^2\mathbf{R}_{r_2}\left\{{}^{r_2}\left(\mathbf{P}_{r_2}\mathbf{P}_{r_1} - \mathbf{P}_{r_2}\mathbf{P}_2 - \mathbf{P}_2\mathbf{P}_1\right)\right\}\right]^{\times}{}^2\mathbf{R}_{r_2}{}^{r_2}\mathbf{R}_{r_1}\right] \\
&= \left[{}^2\mathbf{R}_{r_2}{}^{r_2}\mathbf{R}_{r_1}, \quad \left[{}^2\mathbf{R}_{r_2}{}^{r_2}\mathbf{R}_{r_1}{}^{r_1}\left(\mathbf{P}_1\mathbf{P}_{r_1}\right)\right]^{\times}{}^2\mathbf{R}_{r_2}{}^{r_2}\mathbf{R}_{r_1}\right]
\end{aligned}$$

Since

$$^{r_1}\left(\mathbf{P}_1\mathbf{P}_{r_1}\right) = -{}^{r_1}\overline{\mathbf{P}}_1,$$

we obtain:

$$-\mathbf{T}_{r_2,2}{}^{r_2}\mathbf{V}_{r_1} = \left[{}^2\mathbf{R}_{r_2}{}^{r_2}\mathbf{R}_{r_1}, \quad -\left[{}^2\mathbf{R}_{r_2}{}^{r_2}\mathbf{R}_{r_1}{}^{r_1}\overline{\mathbf{P}}_1\right]^{\times}{}^2\mathbf{R}_{r_2}{}^{r_2}\mathbf{R}_{r_1}\right].$$

Finally,

$$\boxed{-\mathbf{T}_{r_2,2}{}^{r_2}\mathbf{V}_{r_1} = \mathbf{T}_{r_1,2}.} \tag{C.4}$$

Similarly,

$$\boxed{-\mathbf{T}_{r_2,1}{}^{r_2}\mathbf{V}_{r_1} = \mathbf{T}_{r_1,1}.} \tag{C.5}$$

Appendix D

Catenary Derivatives

The catenary projection equation was defined in equation (3.36) and is rewritten here below as a reminder:

$$y\left(\mathbf{s}, x_i\right) = \frac{1}{{}^cZ}\left[-\frac{cosh\left(C\zeta - CD\right) - 1}{C} + aH_{max} + {}^cY_2\right],$$

where

$$\zeta = \frac{{}^cX_2 - x_i{}^cZ_2}{b + x\sqrt{1-b^2}}$$

and

$${}^cZ = \frac{\sqrt{1-b^2}{}^cX_2 + b{}^cZ_2}{b + \sqrt{1-b^2}x_i}.$$

The derivative of the catenary projection equation with respect to the parameters $a = \frac{H}{H_{max}}$ and $b = \sin\alpha$ was introduced by equations 3.45 and 3.46. They are rewritten here below:

$$\begin{aligned}\frac{\partial y(x_i, \mathbf{s})}{\partial a} = \frac{1}{{}^cZ}\Bigg[&-\frac{\partial \frac{1}{C}}{\partial a}\left[\cosh\left(C\zeta - CD\right) - 1\right] \\ &- \frac{\sinh\left(C\zeta - CD\right)}{C}\left(A_1 + A_2 + A_3\right) + H_{max}\Bigg]\end{aligned} \tag{D.1}$$

and

$$\begin{aligned}\frac{\partial y(\mathbf{s}, x_i)}{\partial b} = &-\frac{1}{C}\frac{\partial \frac{1}{{}^cZ}}{\partial b}\left[\cosh\left(C\zeta - CD\right) - 1 - C\left(H + {}^cY_2\right)\right] \\ &- \frac{1}{{}^cZ}\frac{\partial \zeta}{\partial b}\sinh\left(C\zeta - CD\right).\end{aligned} \tag{D.2}$$

In this appendix, we give the calculation details of these derivatives.

First, from equation D.1, the intermediate variables A_1, A_2 and A_3 of are defined as:

$$A_1 = \zeta \frac{\partial C}{\partial a} \tag{D.3}$$

$$A_2 = -D \frac{\partial C}{\partial a} \tag{D.4}$$

$$A_3 = -C \frac{\partial D}{\partial a}. \tag{D.5}$$

The catenary parameter C was defined in equation (2.17), which is presented again here below as a reminder:

$$C = 2\frac{C_n}{C_d} = 2\frac{2H + \Delta H + 2L\sqrt{H\frac{H+\Delta H}{L^2-\Delta H^2}}}{L^2 - (2H + \Delta H)^2}.$$

The partial derivative of C with respect to the a-parameter is:

$$\frac{\partial C}{\partial a} = \frac{\partial C}{\partial H}\frac{dH}{da} \tag{D.6}$$

$$\frac{\partial C}{\partial H} = 2\frac{\frac{\partial C_n}{\partial H}C_d - \frac{\partial C_d}{\partial H}C_n}{C_d^2} \tag{D.7}$$

$$\frac{\partial C_n}{\partial H} = 2 + L\frac{2H + \Delta H}{\sqrt{H(H + \Delta H)(L^2 - \Delta H^2)}} \tag{D.8}$$

$$\frac{\partial C_d}{\partial H} = -4(2H + \Delta H)\ , \tag{D.9}$$

with $\frac{dH}{da} = H_{max}$. The partial derivative of $\frac{1}{C}$ with respect to the a-parameter is:

$$\frac{\partial \frac{1}{C}}{\partial a} = \frac{\partial \frac{1}{C}}{\partial H}\frac{dH}{da} \tag{D.10}$$

$$\frac{\partial \frac{1}{C}}{\partial H} = \frac{\frac{\partial C_d}{\partial H}C_n - \frac{\partial C_n}{\partial H}C_d}{2C_n^2}. \tag{D.11}$$

The partial derivative of D, the catenary half-span, with respect to the a-parameter is:

$$\frac{\partial D}{\partial a} = \frac{\partial \frac{1}{C}}{\partial a}\text{acosh}(CH + 1) + \frac{\frac{\partial C}{\partial a}H + CH_{max}}{C\sqrt{(CH)^2 + 2CH}} \tag{D.12}$$

Finally, the derivative given by equation (D.2) are calculated as follows:

$$\frac{\partial \frac{1}{Z}}{\partial b} = \frac{{}^cX_2 - {}^cZ_2 x_i}{\sqrt{1-b^2}\left(\sqrt{1-b^2}\,{}^cX_2 + b\,{}^cZ_2\right)^2} \tag{D.13}$$

$$\frac{\partial \zeta}{\partial b} = -\frac{({}^cX_2 - {}^cZ_2 x_i)\left(\sqrt{1-b^2} - bx_i\right)}{\sqrt{1-b^2}\left(b + \sqrt{1-b^2}x_i\right)^2}. \tag{D.14}$$

Appendix E

Preliminary Results with Terrestrial Robots

In this appendix is resumed the preliminary work we made on tether shape control using terrestrial robots. Two articles are reproduced in the following pages.

First, the article we presented at the 2017 IEEE International Conference on Robotics and Automation (ICRA 2017) is reproduced. The title of this paper is **Catenary-based Visual Servoing for Tethered Robots**. It addressed the use of catenary model to achieve visual of an umbilical linking two terrestrial robots.

Second, the article **Tether Shape Control through Catenary-Model-Based Visual Servoing** is reproduced. This paper makes a comparison between catenary-based and image-based visual servoing controllers.

Catenary-based Visual Servoing for Tethered Robots

Matheus Laranjeira [1], Claire Dune[1] and Vincent Hugel[1]

***Abstract*— Tethers are used to supply power and transfer data for teleoperated robots. They are known to limit the robot's workspace and could have the effect of hampering its motion. What if we could take advantage of the tether? In this paper a new visual servoing scheme for catenary shaped deformable objects is introduced in order to control the tether parametric shape by properly moving its fixation point. In most of the visual servoing approaches the target object is rigid and distant from the controlled robot. On the contrary, in this paper, the object is deformable and attached to the robot, thus its 3D shape changes while the robot is moving. The experimental system is composed of two terrestrial mobile robots of the same motion capabilities linked with a slack rope. Simulation and real experiments validate the proposed control scheme for proper tether handling.**

I. Introduction

Tethered and umbilical systems are used to provide power, communication and assistance to robots operating in severe environments. Some planetary rovers are connected to a remote station by a cable to secure the exploration of unknown and steep terrains. Their motion can be controlled thanks to cameras that can be mounted either on the rover [1] or on the station [2]. Tethered systems are also designed for underwater missions. The underwater tether is usually slack, and links a remotely operated vehicle (ROV) to a ship on the surface. Some research studies deal with solutions to prevent the ROV from beeing disturbed by the tether due to ship motion [3].

Long flexible sagging objects like tethers, cables, wires, hoses can be modeled by splines [4] and catenaries [5], [6], in the context of object transportation and aerial systems, or by parabolas [7], in the case of cable-driven parallel robots. These objects are subject to gravity and the geometrical equation that matches their shape can be used to control the motion of the load to be transferred from one location to another.

Visual servoing [8], [9] controls a robot motion to regulate some visual features that are usually extracted from rigid objects that can be fixed or moving freely. First, geometrical features such as points, line segments and circles were investigated [10]. Then, 3D models were introduced for known manufactured objects [11], [12], image moments to deal with objects of natural shape [13], splines for shape tracking [14] and algebraic curves for shape alignment [15]. Articulated objects were also considered [16]. Some recent works focus on deformable object shape control where robots push or pull the object to reach a desired configuration [17], [18]. In the context of hose transportation, several robots were attached to a semi-rigid cable and a coordinate visual servoing control was introduced based on the cable curvature derived from splines and GEDS (Geometrically Exact Dynamic Splines) [4]. Unknown deformable objects can be modeled thanks to their viscoelasticity properties [17] or using virtual geometrical features such as image points, distances and curvatures [18].

[1] Cosmer Laboratory EA 7398, University of Toulon, France

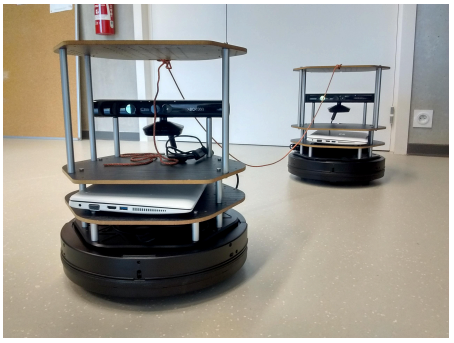


Fig. 1: Experimental setup: two Turtlebots [19] simulate a tether handling system for remotely operated robots. The leader robot freely explores its surroundings while the follower robot is expected to maintain the tether slack enough not to hamper the leader movements

This paper deals with a novel visual servoing strategy based on the deformable shape of a sagging tether that links two terrestrial mobile robots of the same motion capabilities. The objective is to free the leader robot from tether management while the follower robot will be able to position itself to ensure an average sag of the tether and proper orientation with respect to it. The three points below illustrate the contribution of this work:

- The tether is managed between two mobile robots through a visual system embedded on the follower robot that can extract visual information on the tether.
- The tether is modeled by a catenary shape whose parameters are estimated thanks to a real time curve fitting in the image. This allows to take into account the deformation of the tether beyond pixels detected in the image.
- An interaction matrix based on the estimated catenary parameters is calculated and used for the motion control of the follower robot with respect to the deformable tether.

The paper is organized as follows: Section II presents the tether modeling, Section III focuses on the visual servoing strategy used for controlling the position of the follower

robot with respect to the tether. Section IV presents and discusses the experimental results. Finally, Section V concludes the paper and gives directions for future work.

II. Tether modeling

A. Catenary-based geometrical modeling

The experimental setup is presented in Fig. 1. Two terrestrial mobile robots of the same motion capabilities simulate a tether handling system for remotely operated robots. The leader robot freely explores its surroundings while the follower robot is expected to maintain the tether slack enough not to hamper the leader movements. Figure 2 defines the notations and reference frames used in the remainder of the paper. Let Σ_{r_1} and Σ_{r_2} be two Cartesian frames centered at the middle of the robots wheel axis with X-axis pointing towards the front of the robot and a vertical Z-axis. A camera is mounted on the follower robot to track the tether shape. Let Σ_c be a Cartesian frame attached to the camera's optical center, with Z-axis being its optical axis and Y-axis vertically set towards the ground. It is assumed that the optical axis is aligned with the robot X-axis. A 3 millimeter orange rope that links the two robots simulates the tether. Its attachment points are at the same height and define the centers of the Cartesian frames Σ_1 and Σ_2, their orientation being the same as the robot they are attached to. The tether is modeled as a catenary, where R is its half-length, D is the half-span between the attachment points and H is the rope sag. Let Σ_0 be a Cartesian frame attached to the rope. Its center is the center of the rope and the X-axis and Z-axis are in the rope plane.

The classic equation of a catenary, expressed in frame Σ_0, has the following form [20]:

$$\Sigma_0 : \begin{cases} Y = 0 \\ Z = \frac{1}{C}\left[\cosh(Ct) - 1\right] \end{cases} \tag{1}$$

where $C = \frac{2 \cdot H}{R^2 - H^2}$. Expressing the same equation in Σ_2 leads to:

$$\Sigma_2 : \begin{cases} Y = \tan\theta X \\ Z = \frac{1}{C}\left[\cosh\left(C\left(\frac{Y}{\sin\theta} - D\right)\right) - 1\right] - H \end{cases} \tag{2}$$

where θ is the angle between the catenary plane and the Σ_2 X-axis. Finally, the same equation can be written in the camera frame (Σ_c) as:

$$\Sigma_c : \begin{cases} Y = -\frac{1}{C}\left[\cosh\left(C\left(\eta - D\right)\right) - 1\right] + H + Y_2 \\ Z = \cot\theta\left(-X + X_2\right) + Z_2 \end{cases} \tag{3}$$

where $\eta = \frac{-X + X_2}{\sin\theta}$ and (X_2, Y_2, Z_2) are the coordinates of Σ_2 in frame Σ_c.

A 3-D point with coordinates $P = (X, Y, Z)$ in the camera frame is projected on the image plane as a 2-D point with coordinates $p = (x, y)$ through the equation:

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \frac{1}{Z} \begin{bmatrix} f & 0 & c_x & 0 \\ 0 & f & c_y & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \tag{4}$$

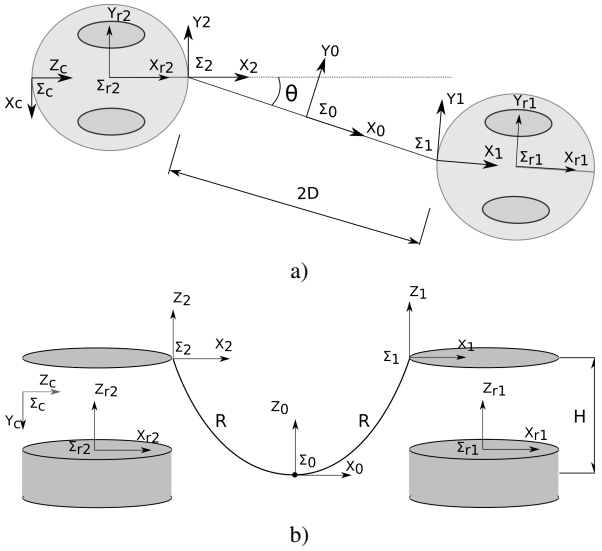


Fig. 2: Scheme of experimental setup. Two terrestrial mobile robots with 2-DOF (rotation and translation) are linked by a passive tether. (a) top view and (b) side view of the scene

where (c_x, c_y) are the coordinates of the principal point and f is the focal length [21]. Therefore, using Eqs. (3) and (4) we obtain the catenary projection on the image plane:

$$y = \eta_1 \left[-\frac{1}{C}\left(cosh\left(C\left(\eta_2 - D\right)\right) - 1\right) + aH_{max} + Y_2 \right] \tag{5}$$

where

$$\eta_1 = \frac{b + \sqrt{1 - b^2}x}{\sqrt{1 - b^2}X_2 + bZ_2} \text{ and } \eta_2 = \frac{X_2 - Z_2 x}{b + x\sqrt{1 - b^2}}$$

As variable shape parameters for the projected curve in the image, we choose:

$$a = \frac{H}{H_{max}} \tag{6}$$

and

$$b = \sin\theta \tag{7}$$

with H_{max} being the tether maximum acceptable sag (i.e. the tether attachment point height). The parameters a and b are respectively relative to the tether slackness and orientation with respect to the follower robot. They range in the interval $[0, 1]$, and the sign of $\sin\theta$ is discriminated by the side on which the tether appears in the image. The tether is horizontal when $a = 0$ and it reaches the ground when $a = 1$. When $b = 0$ the tether belongs to the plane (Σ_2, X_2, Z_2) and when $b = 1$ it belongs to the plane (Σ_2, Y_2, Z_2).

III. Visual servoing algorithm

In this paper, the tether 3D shape is defined by the following feature vector:

$$\mathbf{s} = (a, b)$$

$$\begin{cases} \frac{\partial y(x_i,\mathbf{s})}{\partial a} = \frac{\eta_1}{C}\left[\frac{1}{C}\frac{\partial C}{\partial a}\left(\cosh\left(C\eta_2 - CD\right) - 1\right) - \sinh\left(C\eta_2 - CD\right)\left(A_1 + A_2 + A_3\right) + H_{max}\right] \\ \frac{\partial y(x_i,\mathbf{s})}{\partial b} = -\frac{1}{C}\frac{\partial \eta_1}{\partial b}\left[\cosh\left(C\eta_2 - CD\right) - 1 - C\left(H + Y_2\right)\right] - \eta_1\frac{\partial \eta_2}{\partial b}\sinh\left(C\eta_2 - CD\right) \end{cases} \tag{8}$$

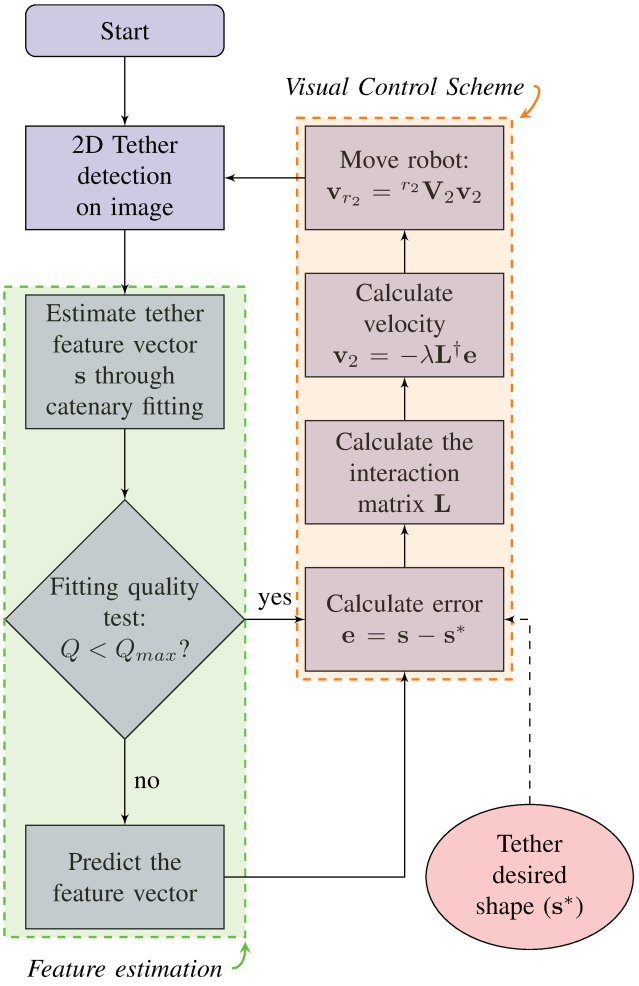


Fig. 3: Algorithm chart flow for catenary-based visual servoing of tethered robots

where a and b were defined in Eq. (6) and (7). The algorithm is depicted on Fig. 3, and is composed of three main steps that are detailed below.

A. Tether detection

The first step deals with the tether detection in the image. It is assumed that the tether has a characteristic color in order to use color segmentation. Knowing that the tether attachment point to the follower robot is above the camera, the detection of pixels belonging to the tether starts in a rectangular region of interest (ROI) in the middle top of the image and iteratively follows the tether shape with a sequence of squares ROIs (Fig. 4a). The location of a subsequent ROI is calculated from the pixel average coordinates of the tether end in the previous ROI. The transformation from pixel to metric coordinates in the image plane is given in Eq. (4).

Wrong rope detection may occur due to lighting changes or when the tether cannot be distinguished from background objects of similar color. A minimum number of detected points and a Pearson coefficient threshold are used to validate the tether detection. The cases of wrong detection are treated with a linear feature prediction as is explained in the next subsection.

B. Estimation of tether shape features

The feature vector is estimated through a non-linear least-square fitting procedure based on a Gauss-Newton algorithm. The following objective function is minimized during the fitting process:

$$\chi^2 = \sum_{i=1}^{N} r_i^2(\mathbf{s}) \tag{9}$$

where $r_i\left(\mathbf{s}\right) = y_i - y(x_i, \mathbf{s})$ is the residual, (x_i, y_i) are the metric coordinates of the i-th detected point of the tether in the image, N is the total number of detected points and $y(x, \mathbf{s})$ is obtained from the current catenary projection in the image (Eq. (5)). The feature vector is iteratively estimated as follows:

$$\mathbf{s}^{k+1} = \mathbf{s}^k - \kappa\mathbf{J}_r^{\dagger}\mathbf{r}\left(\mathbf{s}^k\right)$$

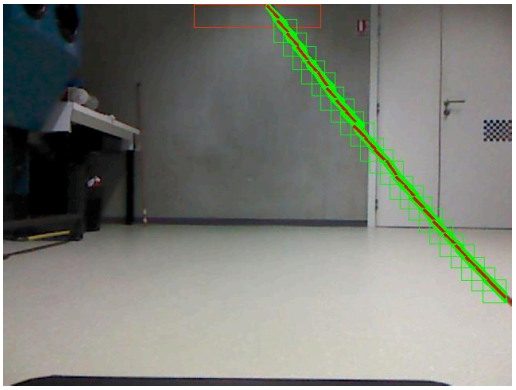
where $\kappa \in \mathbb{R}^+$, $\mathbf{r}\left(\mathbf{s}^k\right)$ is a column vector that stacks the residuals $r_i\left(\mathbf{s}^k\right)$ and $\mathbf{J}_r^{\dagger}$ is the Moore-Penrose pseudo-inverse of the Jacobian matrix $\mathbf{J}_r = \frac{\delta \mathbf{r}}{\delta \mathbf{s}}$, whose i-th row has the following form:

$$\mathbf{J}_r\left(i\right) = \left[-\frac{\partial y(x_i,\mathbf{s})}{\partial a} \quad -\frac{\partial y(x_i,\mathbf{s})}{\partial b}\right]$$

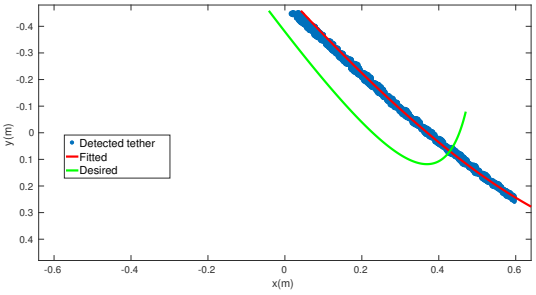
The partial derivatives are given in Eq. (8), with $A_1 = \eta_2\frac{\partial C}{\partial a}$, $A_2 = -D\frac{\partial C}{\partial a}$ and $A_3 = \frac{-A_2+(CR)^2}{a\sqrt{(CR)^2+2CH}}$. The variables η_1, η_2, C, D, H, R and Y_2 are defined in Section II.

The proposed fitting procedure is not suited for small values of parameter $b = \sin\theta$. In such cases, the tether plane is too close to the camera axis, a region where the curve issued from catenary is degenerated and the sum of residuals (Eq. (9)) is no more selective enough. A possible solution to fit smaller values of b would be to use a parabola model $y = f_p(x)$ and to process the fitting using its inverse function relative to the first half of the parabola projection ($x = f_p^{-1}(y)$). Another solution would be to add an extra camera such that the tether plane is never parallel to both camera's axes.

The fitting algorithm was tested in simulation and on real images. As a result, the estimation of $\mathbf{s}$ is exploitable as soon as 30% of the tether is detected. In Fig. 4, the detection finds 70% of the tether, which allows an accurate fitting. The fitting quality is evaluated by the index $Q = \frac{\chi^2}{N}$. For values higher than a empirically tuned threshold, the estimated features are not used in the control velocity calculation. Instead, a first



a)



b)

Fig. 4: Catenary features estimation through a non-linear least-squares fitting. (a) tether detection in the robot image view. (b) tether fitting corresponding to the image view above. The rope detected points are in blue, the fitted and desired catenary curve are in red and green, respectively

order linearization of Eq. (11) is used to predict the feature vector value, as follows:

$$\mathbf{s}(t+dt) = \mathbf{s}(t) + \mathbf{L}(t)\mathbf{v}_2(t) \tag{10}$$

where dt is the sampling period, $\mathbf{L}$ is the catenary interaction matrix and $\mathbf{v}_2 = (v_2, \omega_2)$ is the velocity vector of frame Σ_2 with linear velocity v_2 and angular velocity ω_2. In the current implementation, the number of consecutive iterations used for prediction is not limited.

C. Visual-based control scheme

The relation between the tether features time-derivative and the attachment point Σ_2 motion is given by the following equation:

$$\dot{\mathbf{s}} = \mathbf{L}\mathbf{v}_2 \tag{11}$$

The aim of visual-based control schemes is to minimize the error $\mathbf{e}$ defined as

$$\mathbf{e} = \mathbf{s} - \mathbf{s}^* \tag{12}$$

where $\mathbf{s}$ and $\mathbf{s}^*$ are respectively the current and desired feature vectors (see Fig. 4). In order to ensure an exponential decay of the error, the following control law is designed [8]:

$$\mathbf{v}_2 = -\lambda \mathbf{L}^\dagger \mathbf{e} \tag{13}$$

where $\mathbf{L}^\dagger$ is the Moore-Penrose pseudo-inverse of $\mathbf{L}$ and $\lambda \in \mathbb{R}^+$. The velocity of the follower robot is then obtained through: $\mathbf{v}_{r_2} = {}^{r_2}\mathbf{V}_2 \mathbf{v}_2$, where ${}^{r_2}\mathbf{V}_2$ is the twist transformation matrix from frame Σ_2 to frame Σ_{r_2}.

We used a nonholonomic robot with 2 DOF to experimentally validate our control scheme, as is explained in Section IV. In such case, the conversion between $\mathbf{v}_{r_2}$ and the pair of control variables $\mathbf{u} = (v_x, \omega_z)$ is achieved by the following projection:

$$\mathbf{u} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \mathbf{v}_{r_2}$$

D. Interaction matrix computation

The interaction matrix $\mathbf{L}$ is the tether deformation model. It links the relative motion of the attached points to the derivative of the tether parameters. Contrary to [18], the object has a known parametric shape and this matrix is analytically computed from the definition of $\mathbf{s} = (\mathbf{a}, \mathbf{b})$.

Let $P_0 = (X_0, Y_0, Z_0)$ and $P_1 = (X_1, Y_1, Z_1)$ be the centers of frames Σ_0 and Σ_1, respectively. Evaluating and differentiating Eq. (2) at P_1, we get

$$\dot{H} = -\frac{\dot{C}}{C^2}\left[\cosh(CD) - 1\right] + \sinh(CD)\left(\frac{\dot{C}D}{C} + \dot{D}\right) \tag{14}$$

and differentiating the expression of C in Eq. (1) we have

$$\dot{C} = K_C \dot{H} \tag{15}$$

where $K_C = \frac{2\left(R^2+H^2\right)}{\left(R^2-H^2\right)^2}$. Then, from the definition of a in Eq. (6) and replacing Eq. (15) in (14), we obtain

$$\dot{a} = \frac{K_H}{H_{max}} \dot{D} \tag{16}$$

where

$$K_H = \frac{sinh\,(CD)}{1 + \frac{K_C}{C^2}\left[\cosh(CD) - 1 - CD\sinh(CD)\right]}$$

The relative motion of P_1 in the reference frame Σ_2, due to the velocity of the follower robot $\mathbf{v}_2$ expressed in Σ_2 is:

$$\dot{P}_1 = -v_2 - \omega_2 \times P_1 \tag{17}$$

Geometrically,

$$\dot{P}_0 = \frac{\dot{P}_1}{2} \tag{18}$$

and

$$\dot{D} = \frac{X_0\dot{X}_0 + Y_0\dot{Y}_0}{D} \tag{19}$$

Thus, using Eqs. (17) and (18) in (19) we have:

$$\dot{D} = \frac{1}{2}\begin{bmatrix} -\cos\theta \\ -\sin\theta \\ 0 \\ Z_1\sin\theta \\ -Z_1\cos\theta \\ Y_1\cos\theta - X_1\sin\theta \end{bmatrix}^\top \mathbf{v}_2 \tag{20}$$

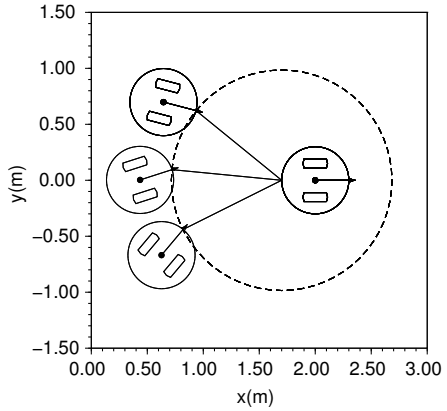


Fig. 5: Multiple positions of the follower robot for the same tether slackness and orientation: the problem only constrains the tether shape and orientation w.r.t. the follower robot

Finally, using Eq. (20) in (16) and given that $b = \sin\theta$,

$$\dot{a} = \frac{K_H}{2H_{max}} \begin{bmatrix} -\sqrt{1-b^2} \\ -b \\ 0 \\ Z_1 b \\ -Z_1\sqrt{1-b^2} \\ Y_1\sqrt{1-b^2} - X_1 b \end{bmatrix}^\top \mathbf{v}_2 \qquad (21)$$

Differentiating $b = \frac{Y_0}{D}$ leads to:

$$\dot{b} = \frac{\dot{Y}_0 - b\dot{D}}{D}$$

From the expressions of $\dot{Y}_0$ and $\dot{D}$ in Eqs. (18) and (20) respectively, we get:

$$\dot{b} = \frac{1}{2D} \begin{bmatrix} b\sqrt{1-b^2} \\ -1+b^2 \\ 0 \\ Z_1(1-b^2) \\ Z_1 b\sqrt{1-b^2} \\ -Y_1 b\sqrt{1-b^2} - X_1(1-b^2) \end{bmatrix}^\top \mathbf{v}_2 \qquad (22)$$

The complete interaction matrix (Eq. (23)) is composed of the rows given in Eqs. (21) and (22) with one more simplification: $Z_1 = 0$ since the attachment points are at the same height. The values of X_1 and Y_1 are geometrically calculated from Eq. (2).

The rank of L is 2, which means that this control law only commands two degrees of freedom of the follower robots, *i.e.* the relative attached points distance and the follower orientation with regards to the tether plane. Figure 5 shows several follower positions for the same set of parameters $s = (a, b)$. Remaining degrees of freedom are thus available for additional tasks, such as obstacle avoidance.

IV. Experimental results

A. Setup

One simulation and two real experiments are presented in order to validate the control law. Two Turtlebots [19] with 2 DOF (translation v_x and rotation w_z) are used as experimental robots. They are equipped with a Kinect device and linked by a 1.4 meter long and 3 millimeter thick orange rope (Fig. 1). The rope maximum sag is $H_{max} = 0.40$m, corresponding to its attachment point's height. The servoing algorithm runs in the follower robot computer, which is equipped with a Intel Core i5-2410M @ 2.3 GHz processor. The video frame rate is fixed to 10Hz. The same robot motion capabilities, rope length, thickness and maximum sag are used in simulation and real experiments.

B. Simulation

The objective of the simulation is that the follower robot moves the tether from a very slackened to a moderately tight shape. The tether orientation is controlled, passing from a large angle θ to a desired smaller angle. The initial and desired values of the feature vector are $\mathbf{s}_o = (0.9, 0.8)$ and $\mathbf{s}^* = (0.5, 0.5)$, respectively. We assume that the robots can perfectly estimate the tether parameters, so the fitting process is not simulated. Figure 6 presents the simulation results. As expected, both parameters have an exponential decay and converge to the desired value within 4 seconds for a gain $\lambda = 0.75$ and a sampling period of 0.1 seconds.

C. Real experiments

In the first experiment, the simulation initial conditions are repeated and the follower robot moves the tether from an initial shape $\mathbf{s}_o = (0.9, 0.8)$ to a desired shape $\mathbf{s}^* = (0.5, 0.5)$. We used the same gain as for simulation ($\lambda = 0.75$) and a fitting quality threshold $Q_{max} = 1$. Results are summarized in Fig. 7.

In the second experiment (Fig. 8), the leader robot is freely displaced while the follower robot ensures that the rope keeps a desired shape $\mathbf{s}^* = (0.7, -0.5)$. First, the leader robot moves forward, then turns left (24s). Next, it moves about 1.25 meters forward (38s), makes a half-turn (50s) and moves about 1.25 meters forward again (64s). At the end, it makes a quarter turn and moves about 0.50 meters backward (76s). Compared with the first experiment, a higher value was set to the tether first parameter in order to give more freedom of maneuvering to the leader robot. We used $Q_{max} = 1$ and two different gains for linear and angular velocities: $\lambda_l = 1.0$ and $\lambda_\omega = 6.0$, respectively (see discussion below). Figure 8c presents the fitting quality index evolution during the experiment. The feature prediction was used 20 times in cases of wrong rope detection and more 38 times due to inaccurate fitting.

In Fig. 7, the angular velocity command does not converge to zero, and remains quasi constant. This is due to the limitations of our experimental platform that cannot achieve a rotation velocity lower than 0.05 rad/s. In the second experiment, we selected a higher gain for the angular velocity compared to the linear velocity gain in order to overcome

$$\mathbf{L} = \begin{bmatrix} -\frac{K_H\sqrt{1-b^2}}{2H_{max}} & -\frac{K_H b}{2H_{max}} & 0 & 0 & 0 & \frac{K_H\left(Y_1\sqrt{1-b^2}-X_1 b\right)}{2H_{max}} \\ \frac{b\sqrt{1-b^2}}{2D} & \frac{-1+b^2}{2D} & 0 & 0 & 0 & -\frac{Y_1 b\sqrt{1-b^2}+X_1(1-b^2)}{2D} \end{bmatrix} \tag{23}$$

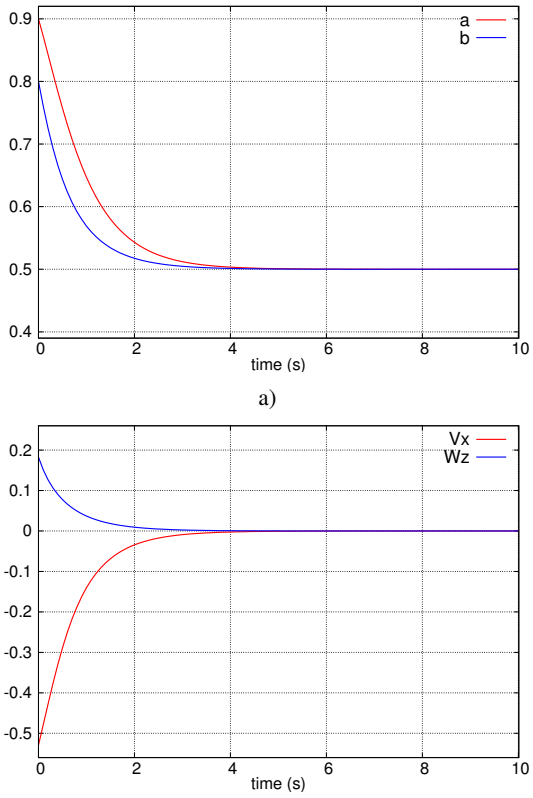


Fig. 6: Simulation results for tether shape control. (a) the parameters evolution. The tether goes from an initial to a desired shape ($\mathbf{s}_o = (0.9, 0.8)$ and $\mathbf{s}^* = (0.5, 0.5)$, respectively). (b) the control velocities. Linear velocity (v_x) in m/s and angular velocity (w_z) in rad/s

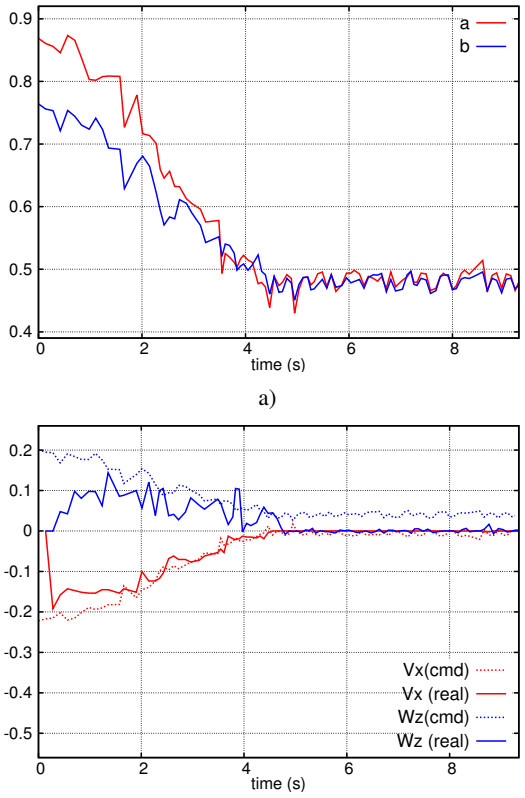


Fig. 7: Results of a real experiment for tether shape control. (a) the tether parameters evolution. The tether goes from an initial to a desired shape ($\mathbf{s}_o = (0.9, 0.8)$ and $\mathbf{s}^* = (0.5, 0.5)$, respectively). (b) the control velocities. Linear velocity (v_x) in m/s and angular velocity (w_z) in rad/s

this problem. As a future improvement, an integrator compensator to deal with the low level control of the velocity can be added to the servoing loop.

Looking at Fig. 7, we can note that the real experimentation curves are noisy compared with those obtained in simulation. This is mainly due to wrong rope detection that can affect the tether feature estimation. Another reason is the possible rope oscillation during the robot motion, which can occur when the robot halts and restarts motion. This could be taken into account by designing a dynamic controller that takes the rope inertia into account. In addition, the rope can be tracked in the image by a gradient-guided algorithm.

V. Conclusions and Perspectives

This paper presents a new visual servoing control scheme to manage a tether linking two terrestrial mobile robots. The tether is modeled by a catenary and its shape parameters are estimated by a non-linear least square fitting. A control law that takes into account the tether deformation was introduced in order to allow the robot to visually control the tether shape. The control scheme is validated by simulation and by two series of real experiments. The comparison of simulation and real experiments shows that both curves converge, validating the proposed control law. The vision-based tether shape controller implemented here gives promising results.

Future work will address the extension of the visual control scheme to multiple 6-D0F robots, and the use of complementary sensors to increase visual feature detection accuracy.

Acknowledgments

This work was achieved thanks to specific funding from the France PACA (Provence-Alpes-Cte-d'Azur) region, and the French SUBSEATECH company, located in Marseille, South-East of France (http://www.subsea-tech.com, pierre.marty@subsea-tech.com).

REFERENCES

[1] D. Tsai, I. A. D. Nesnas, and D. Zarzhitsky, "Autonomous vision-based tethered-assisted rover docking," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Nov 2013, pp. 2834–2841.

[2] J. Iqbal, S. Heikkila, and A. Halme, "Tether tracking and control of rosa robotic rover," in *Control, Automation, Robotics and Vision, 2008. ICARCV 2008. 10th International Conference on*, Dec 2008, pp. 689–693.

[3] S. Prabhakar and B. Buckham, "Dynamics modeling and control of a variable length remotely operated vehicle tether," in *Proceedings of OCEANS 2005 MTS/IEEE*, Sept 2005, pp. 1255–1262 Vol. 2.

[4] Z. Echegoyen, I. Villaverde, R. Moreno, M. Graa, and A. dAnjou, "Linked multi-component mobile robots: Modeling, simulation and control," *Robotics and Autonomous Systems*, vol. 58, no. 12, pp. 1292 – 1305, 2010, intelligent Robotics and Neuroscience. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0921889010001533

[5] J. Estevez and M. Graña, *Robust Control Tuning by PSO of Aerial Robots Hose Transportation*. Cham: Springer International Publishing, 2015, ch. Bioinspired Computation in Artificial Systems: International Work-Conference on the Interplay Between Natural and Artificial Computation, IWINAC 2015, Elche, Spain, June 1-5, 2015, Proceedings, Part II, pp. 291–300.

[6] T. Lee, "Geometric controls for a tethered quadrotor uav," in *2015 54th IEEE Conference on Decision and Control (CDC)*, Dec 2015, pp. 2749–2754.

[7] T. Dallej, M. Gouttefarde, N. Andreff, R. Dahmouche, and P. Martinet, "Vision-based modeling and control of large-dimension cable-driven parallel robots," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Oct 2012, pp. 1581–1586.

[8] F. Chaumette and S. Hutchinson, "Visual servo control. i. basic approaches," *IEEE Robotics Automation Magazine*, vol. 13, no. 4, pp. 82–90, Dec 2006.

[9] ——, "Visual servo control. ii. advanced approaches [tutorial]," *IEEE Robotics Automation Magazine*, vol. 14, no. 1, pp. 109–118, March 2007.

[10] B. Espiau, F. Chaumette, and P. Rives, "A new approach to visual servoing in robotics," *IEEE Transactions on Robotics and Automation*, vol. 8, no. 3, pp. 313–326, Jun 1992.

[11] A. Comport, E. Marchand, and F. Chaumette, "Robust model-based tracking for robot vision," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, IROS'04*, vol. 1, Sendai, Japan, September 2004, pp. 692–697.

[12] A. Petit, E. Marchand, and K. Kanani, "A robust model-based tracker for space applications: combining edge and color information," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, IROS'2013*, Tokyo, Japan, November 2013, pp. 3719–3724.

[13] O. Tahri and F. Chaumette, "Application of moment invariants to visual servoing," in *International Conference on Robotics and Automation*, vol. 3, 2003, pp. 4276–4281.

[14] P. Li, F. Chaumette, and O. Tahri, "A shape tracking algorithm for visual servoing," in *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, April 2005, pp. 2847–2852.

[15] A. Y. Yazicioglu, B. Calli, and M. Unel, "Image based visual servoing using algebraic curves applied to shape alignment," in *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Oct 2009, pp. 5444–5449.

[16] A. Comport, E. Marchand, and F. Chaumette, "Kinematic sets for real-time robust articulated object tracking," *Image and Vision Computing, IVC*, vol. 25, no. 3, pp. 374–391, March 2007.

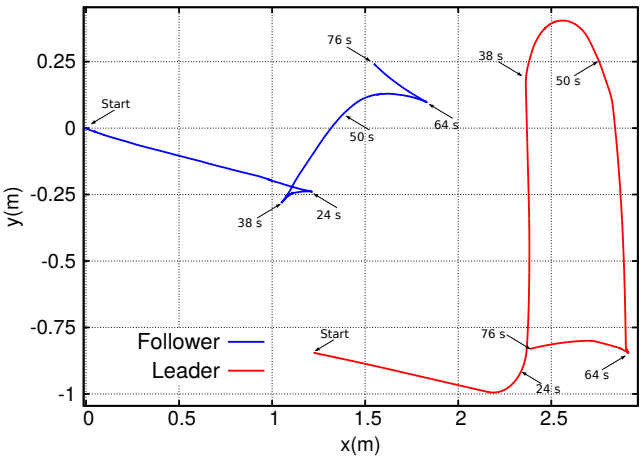
[17] M. Higashimori, K. Yoshimoto, and M. Kaneko, "Active shaping of an unknown rheological object based on deformation decomposition into elasticity and plasticity," in *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, May 2010, pp. 5120–5126.

[18] D. Navarro-Alarcon, Y.-h. Liu, J. G. Romero, and P. Li, "On the visual deformation servoing of compliant objects: Uncalibrated control methods and experiments," *The International Journal of Robotics Research*, vol. 33, no. 11, pp. 1462–1480, 2014.

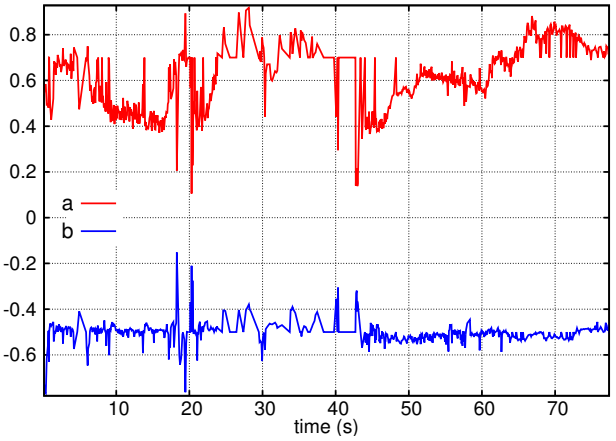
[19] B. Gerkey and K. Conley, "Robot developer kits [ros topics]," *IEEE Robotics Automation Magazine*, vol. 18, no. 3, pp. 16–16, Sept 2011.

[20] J. Stewart, *Calculus*, 6th ed. Thomson Learning, 2007, ch. 7, p. 464.

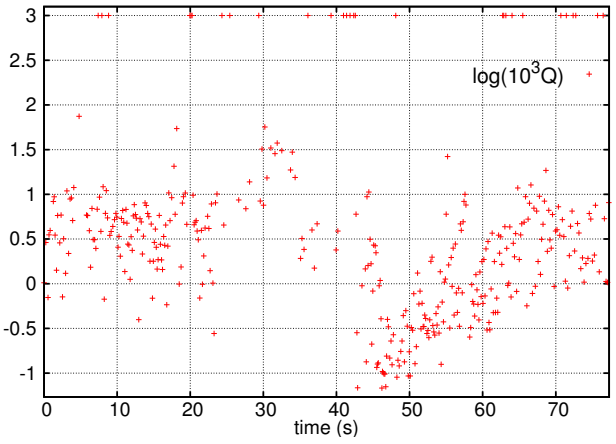
[21] R. I. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, 2nd ed. Cambridge University Press, 2004, ch. 6, p. 155.



a)



b)



c)

Fig. 8: Two robots are linked by a tether. The leader robot freely moves while the follower robot maintains a desired tether shape $\mathbf{s}^* = (0.7, -0.5)$. (a) the leader and follower trajectories with time indications in seconds. (b) the tether parameters evolution. (c) the fitting quality index Q evolution during the experiment. Feature prediction is used in cases of wrong rope detection and inaccurate fitting ($Q = 1$ i.e. $log(10^3 Q) = 3$)

Tether Shape Control through Catenary-Model-Based Visual Servoing

Matheus Laranjeira [1], Claire Dune[1] and Vincent Hugel[1]

***Abstract*— Tethers are usually required by robots operating in hostile environment to ensure power supply and data transfer. Yet, the tether can have an adverse pulling effect on the robot motion. In this work, we envision a chain of robots where vision based control is used between two successive robots to manage the tether slackness. This paper presents a new visual servoing scheme based on the tether lowest point, assuming a catenary shape of the tether. This scheme is compared with an upgraded version of our previous controller based on catenary parameters, and two standard IBVS controllers. Three cases are studied: normal tether visibility, limited tether visibility and tether close to robot's sagittal plane. Simulation results show that traditional visual servoing cannot reach tether desired shape in some limit cases whereas catenary-based controllers can successfully achieve tether shape control as long as the estimation of catenary parameters is enough reliable.**

I. Introduction

Tethered and umbilical systems are used to provide power and communication to robotic operations in hazardous and remote environments such as planetary exploration [1], [2], underwater missions [3], [4] and rescue operations [5]. However, tethers can limit the robots motion and get entangled with obstacles or with other fellow robots. In order to deal with these issues, mechanical systems such as actuated spools [5]–[7] can be used, embedded cameras can give visual feedback to control the tether slackness during docking operations [1] and an intermediate robot can be used to manage the tether during underwater tasks [4].

Actually, a chain of tethered robots can be useful to explore hostile and cluttered environments. Thanks to the chain configuration, the tethered robots can cover larger areas while keeping connection between pairs of robots. Tether force and tether heading sensing capabilities at attachment points can provide means of estimating situations of entanglements [8].

The work presented in this paper focuses on tethered robots inside a chain of robots that have built-in frontal camera, which is used for local detection of the tether. Vision-based control of deformable objects is rarely addressed due to shape estimation complexity or lack of knowledge about the object model. A Geometrically Exact Dynamic Splines controller was introduced in [9] to servo wheeled robots carrying a cable. Virtual geometrical features, such as image points, angles and curvatures were used to control the shape of unknown deformable objects [10]. Cables are also modeled by splines [9], parabolas [11] and catenaries [12], [13].

Recently, we introduced an original visual servoing control scheme for catenary-shaped deformable objects to control the shape of a sagging tether linking two mobile robots [14]. In this paper, this visual servoing control scheme is upgraded for better parameters estimation, and a new visual servoing control scheme based on the catenary lowest point is introduced to combine visual features and catenary parameters. In addition, both catenary-based visual servoing controllers are compared with traditional image-based visual servoing techniques.

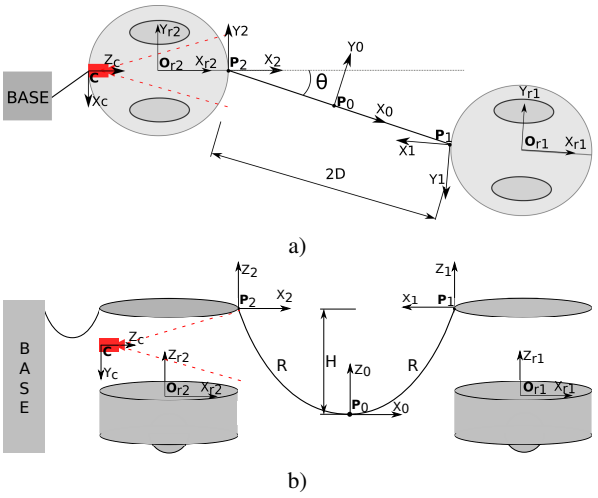


Fig. 1: A leader (r_1) and a follower (r_2) mobile robots are linked by a sagging tether: (a) top view and (b) side view of the scene. Camera, robots and tether frames are respectively represented by: $\Sigma_c = (\mathbf{C}, X_c, Y_c, Z_c)$, $\Sigma_{r_i} = (\mathbf{O}_{r_i}, X_{r_i}, Y_{r_i}, Z_{r_i})$ and $\Sigma_i = (\mathbf{P}_i, X_i, Y_i, Z_i)$. The tether half-length is noted R, its sag is H and the half-span between attachment points is D. The angle between the tether plane and X_2-axis is θ.

The paper is structured as follows. Section II presents the upgraded version of the original catenary-parameter-based visual servoing scheme. Section III describes the new visual servoing scheme based on the catenary lowest point. Section IV deals with the comparison between catenary based visual servoing and traditional image-based visual servoing, in different cases of tether projection shape in the image. Section V concludes the work and presents new perspectives.

II. Upgraded Catenary-Based Visual Servoing

A. Catenary Model

In [14], we introduced an original visual servoing control scheme for tether shape control based on catenary parameters that are estimated through curve fitting of the image projection of the catenary. Parameters are $a = \frac{H}{H_{max}}$ and

[1] Cosmer Laboratory EA 7398, University of Toulon, France

$b = \sin\theta$, respectively related to the tether slackness (H) and to the tether orientation (θ) with respect to the follower robot sagittal plane (Fig. 1). The tether maximum authorized slackness is noted H_{max}. The catenary 3D equation modeling the tether linking both robots is given by:

$$\Sigma_c : \begin{cases} Y = -\frac{1}{C}\left[\cosh\left(C\left(\eta - D\right)\right) - 1\right] + H + {}^cY_2 \\ Z = \cot\theta\left(-X + {}^cX_2\right) + {}^cZ_2 \end{cases} \quad (1)$$

where $C = \frac{2 \cdot H}{R^2 - H^2}$, $\eta = \frac{-X + X_2}{\sin\theta}$ and $\mathbf{P}_2 = ({}^cX_2, {}^cY_2, {}^cZ_2)$ in frame Σ_c. The equation of the catenary projection on the image plane is given by:

$$y = \eta_1 \left[-\frac{1}{C}\left(cosh\left(C\left(\eta_2 - D\right)\right) - 1\right) + aH_{max} + Y_2 \right] \quad (2)$$

where $\eta_1 = \frac{b + \sqrt{1-b^2}x}{\sqrt{1-b^2}{}^cX_2 + b\,{}^cZ_2}$ and $\eta_2 = \frac{{}^cX_2 - {}^cZ_2 x}{b + x\sqrt{1-b^2}}$. The tether is supposed to be only subject to gravity, and that its mass is negligible with respect to the robot's mass. Attachment points are at the same height.

The objective function of the curve fitting algorithm is given by $\chi^2 = \sum_{i=1}^{N} r_i^2(\mathbf{s})$, where $\mathbf{s} = [a\ b]^T$, $r_i(\mathbf{s}) = y_i - y(x_i, \mathbf{s})$ is the residual, (x_i, y_i) are the metric coordinates of the i-th detected point of the tether in the image, N is the total number of detected points and $y(x, \mathbf{s})$ is obtained from the current catenary projection in the image (Eq. (2)). The Gauss-Newton Jacobian used by the fitting algorithm is:

$$\mathbf{J}_r(i) = \left[-\frac{\partial y(x_i, \mathbf{s})}{\partial a} \quad -\frac{\partial y(x_i, \mathbf{s})}{\partial b} \right] \quad (3)$$

The curve fitting algorithm used in this approach has the advantage of only requiring 30% of tether visibility for accurate shape estimation. However, this is a local approach that can converge to a local minimum. In order to find a global minimum, an initial guess can be estimated from the position of the lowest point of the catenary.

B. Upgraded parameters estimation through initial guess

The initial guess for the catenary curve fitting algorithm makes the following approximation: the lowest point $\mathbf{p}_B$ of the tether projection that is detected in the image is approximated with the projection $\mathbf{p}_0 = [x_0, y_0]$ of the tether 3D lowest point $\mathbf{P}_0$ (Fig. 2).

Evaluating Eq. (2) at $\mathbf{p}_0 \approx \mathbf{p}_B$ leads to

$$\Sigma_c : \begin{cases} \hat{D} = \dfrac{{}^cX_2 - {}^cZ_2 x_B}{\sin\hat{\theta} + x_B\cos\hat{\theta}} & (4a) \\ \hat{H} = y_B \dfrac{\cos\hat{\theta}{}^cX_2 + \sin\hat{\theta}{}^cZ_2}{\sin\hat{\theta} + x_B\cos\hat{\theta}} - {}^cY_2 & (4b) \end{cases}$$

The relation between D and H is obtained evaluating Eq. (1) at $\mathbf{P}_1$, yielding:

$$\hat{H} = \frac{1}{\hat{C}}\left[\cosh(\hat{C}\hat{D}) - 1\right] \quad (5)$$

In order to facilitate writing, let's redefine Eq. (4) with:

$$\hat{D} = \frac{G_1}{G_3}$$

and

$$\hat{H} = \frac{G_2}{G_3}$$

where $G_1 = {}^cX_2 - {}^cZ_2 x_B$, $G_2 = y_B\left(\cos\hat{\theta}{}^cX_2 + \sin\hat{\theta}{}^cZ_2\right) - {}^cY_2 G_3$ and $G_3 = \sin\hat{\theta} + x_B\cos\hat{\theta}$. Then, replacing Eq. (4) in (5) yields:

$$g\left(\hat{\theta}\right) = \frac{2G_2^2}{R^2G_3^2 - G_2^2} - cosh\left(\frac{2G_1G_2}{R^2G_3^2 - G_2^2}\right) + 1 \quad (6)$$

The root of $g(\hat{\theta})$ yields an estimation of θ, and $\hat{H}$ is obtained from Eq. (4b). This initial estimation of the catenary parameters is then used by the curve fitting algorithm.

C. Visual Servoing Control

The aim of vision-based control schemes is to minimize an error defined as $\mathbf{e} = \mathbf{s} - \mathbf{s}^*$, where $\mathbf{s}$ and $\mathbf{s}^*$ are respectively the current and the desired visual feature vectors [15], [16]. In our control scheme, the feature vector $\mathbf{s} = [a\ b]^T$ issued from catenary parameters is regulated through the follower robot motion using the following control law [14]:

$$\mathbf{v}_{r2} = -\lambda {}^{r2}\mathbf{V}_2 \mathbf{L}_C^{+}\mathbf{e} \quad (7)$$

where ${}^{r2}\mathbf{V}_2$ is the twist transformation matrix from frame Σ_2 to Σ_{r2} and $\lambda \in \mathbb{R}^+$. The interaction matrix $\mathbf{L}_C$ is given by Eq. (8), where $\widehat{K_C} = \frac{2\left(R^2 + \hat{H}^2\right)}{\left(R^2 - \hat{H}^2\right)^2}$, ${}^2\mathbf{P}_1 = \left({}^2X_1, {}^2Y_1, {}^2Z_1\right)$ in frame Σ_2 is estimated thanks to the fitted catenary, and

$$\widehat{K_H} = \frac{sinh\left(\hat{C}\hat{D}\right)}{1 + \frac{\widehat{K_C}}{\hat{C}^2}\left[\cosh\left(\hat{C}\hat{D}\right) - 1 - \hat{C}\hat{D}\sinh\left(\hat{C}\hat{D}\right)\right]}$$

where $\hat{C} = \frac{2\hat{H}}{R^2 - \hat{H}^2}$ and $\hat{D}$ is given by Eq. (4a).

III. Catenary Lowest-Point Visual Servoing

The first version of the catenary-based controller uses 3D catenary parameters in the feature vector. We propose to design a new controller driven by image features that takes into account the catenary model of the tether. The idea behind this new controller is to take advantage of the same approximation that is used for the upgrade of the first version. The image feature vector is therefore $\mathbf{s} = [x_B\ y_B]^T$, where the coordinates $(x_0\ y_0)$ are approximated with $(x_B\ y_B)$. The new interaction matrix $\mathbf{L}_P$ that links the variation of the feature vector $\mathbf{s} = [x_B\ y_B]^T$ to the camera velocity is calculated starting with the catenary model given by Eq. (1).

Let ${}^c\mathbf{P}_0 = ({}^cX_0, {}^cY_0, {}^cZ_0)$, ${}^c\mathbf{P}_1 = ({}^cX_1, {}^cY_1, {}^cZ_1)$ and ${}^c\mathbf{P}_2 = ({}^cX_2, {}^cY_2, {}^cZ_2)$ be respectively the catenary frame center and the two attachment points in the follower camera frame Σ_c. Differentiating the projection equation [17] at ${}^c\mathbf{P}_0$ leads to:

$$\Sigma_c : \begin{cases} \dot{x_0} = \dfrac{{}^c\dot{X}_0 - x_0{}^c\dot{Z}_0}{{}^cZ_0} & (9a) \\ \dot{y_0} = \dfrac{{}^c\dot{Y}_0 - y_0{}^c\dot{Z}_0}{{}^cZ_0} & (9b) \end{cases}$$

Then, differentiating Eq. (1) at ${}^c\mathbf{P}_0$ and ${}^c\mathbf{P}_1$ yields

$${}^c\dot{Y}_0 = K_H\dot{D} \quad (10)$$

$$\Sigma_2 : \mathbf{L}_C = \begin{bmatrix} -\frac{K_H\sqrt{1-b^2}}{2H_{max}} & -\frac{K_H b}{2H_{max}} & 0 & 0 & 0 & \frac{K_H\left({}^2Y_1\sqrt{1-b^2} - {}^2X_1 b\right)}{2H_{max}} \\ \frac{b\sqrt{1-b^2}}{2D} & \frac{-1+b^2}{2D} & 0 & 0 & 0 & -\frac{{}^2Y_1 b\sqrt{1-b^2} + {}^2X_1(1-b^2)}{2D} \end{bmatrix} \quad (8)$$

The relative motion of ${}^c\mathbf{P}_1$ in the reference frame Σ_c, due to the camera velocity $\mathbf{v}_c = (v_c, w_c)$ expressed in Σ_c is given by

$${}^c\dot{\mathbf{P}}_1 = -v_c - \omega_c \times {}^c\mathbf{P}_1 \quad (11)$$

Geometrically,

$$\begin{bmatrix} {}^c\dot{X}_0 & {}^c\dot{Z}_0 \end{bmatrix}^T = \frac{1}{2}\begin{bmatrix} {}^c\dot{X}_1 & {}^c\dot{Z}_1 \end{bmatrix}^T \quad (12)$$

and using Eq. (11) leads to

$$\begin{bmatrix} {}^c\dot{X}_0 \\ {}^c\dot{Z}_0 \end{bmatrix} = \frac{1}{2}\begin{bmatrix} -1 & 0 & 0 & 0 & -{}^cZ_1 & {}^cY_1 \\ 0 & 0 & -1 & -{}^cY_1 & {}^cX_1 & 0 \end{bmatrix} \mathbf{v}_c \quad (13)$$

From $D^2 = ({}^cX_0 - {}^cX_2)^2 + ({}^cZ_0 - {}^cZ_2)^2$,

$$\dot{\hat{D}} = -\sin\theta\, {}^c\dot{X}_0 + \cos\theta\, {}^c\dot{Z}_0 \quad (14)$$

Using Eqs. (11) and (12) in Eq. (14),

$$\dot{\hat{D}} = \frac{1}{2}\begin{bmatrix} \sin\hat{\theta} \\ 0 \\ -\cos\hat{\theta} \\ -{}^cY_1\cos\hat{\theta} \\ {}^cZ_1\sin\hat{\theta} - {}^cX_1\cos\hat{\theta} \\ -{}^cY_1\sin\hat{\theta} \end{bmatrix}^T \mathbf{v}_c \quad (15)$$

Finally, replacing Eqs. (13), (15) and (10) in (9) and considering the approximation $x_0 \approx x_B$ and $y_0 \approx y_B$ leads to the interaction matrix $\mathbf{L}_P$ given by Eq. (17), where

$$\begin{aligned} {}^cZ_0 &= {}^cZ_2 + D\cos\theta \\ {}^cX_0 &= {}^cX_2 - D\sin\theta \\ {}^cZ_1 &= {}^cZ_2 + 2D\cos\theta \end{aligned}$$

Here it is assumed that the camera frame axes are parallel with the axes of the tether attachment point frame. The matrix is presented in frame Σ_2 in order to be compared to $\mathbf{L}_C$. The feature vector $\mathbf{s} = \begin{bmatrix} x_B & y_B \end{bmatrix}^T$ is then regulated through the following control law:

$$\mathbf{v}_{r2} = -\lambda^{r2}\mathbf{V}_2\mathbf{L}_P^{+}\mathbf{e} \quad (16)$$

IV. Comparisons and Analyzes

Let us compare the catenary-based controllers with traditional image-based visual servoing techniques. We can consider some salient points such as the highest and lowest tether points in the image or approximate the first portion of the tether with a line segment (see Fig. 2). Such features were successfully used to control a robot motion by regulating the shape of rigid objects in the image [18], [19]. The tether shape evolution, velocity commands and condition number of interaction matrices are studied in normal and limit cases for two image-based controllers and the catenary-based controllers:

- **image points**: coordinates of the tether highest and lowest point ($\mathbf{p}_A$ and $\mathbf{p}_B$) detected in the image without taking into account that the points belong to a catenary projection [20]
- **image line segment** : center, length and orientation of the segment $\overline{\mathbf{p}_A\mathbf{p}_B}$ linking the highest and the lowest points of the tether detected in the image, without taking into account that the points belong to a catenary projection [21]
- **catenary parameters**: two features $\mathbf{s} = [a\ b]^T$, related to the tether slackness and orientation that parameterize the catenary curve fitting the tether 3D shape [14].
- **catenary-projection lowest point**: coordinates of the tether lowest point ($\mathbf{p}_B$) detected in the image, making the approximation that this point is the projection of the 3D catenary lowest point

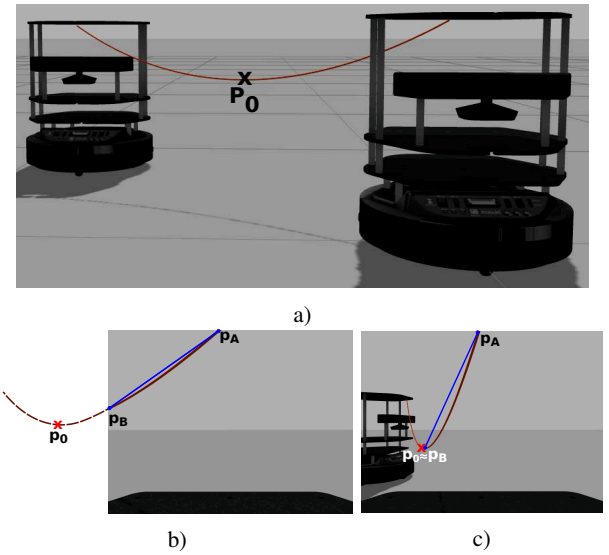


Fig. 2: Image features used to manage the tether shape: its highest and lowest point in the image ($\mathbf{p}_A$ and $\mathbf{p}_B$) and the line segment $\overline{\mathbf{p}_A\mathbf{p}_B}$ in blue. (a) 3D simulated scene where the tether 3D lowest point $\mathbf{P}_0$ is out of the follower's camera field of view. (b) corresponding embedded view with $\mathbf{p}_0$ being the perspective projection of $\mathbf{P}_0$. (c) Another situation where $\mathbf{P}_0$ is inside the follower's camera field of view

A. Simulation Setup

Simulation is performed using Gazebo [22] to test the whole control loop including computer vision algorithms. The setup is composed of two Turtlebots [23] linked by a 1 meter long and 3 mm thick orange rope that simulates the tether. The robots have 2 DOF (one translation and one

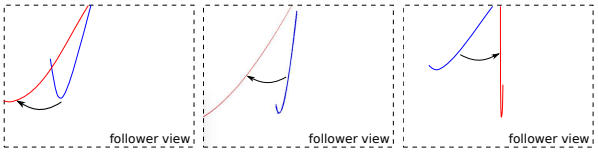
$$\Sigma_2 : \mathbf{L}_P = \frac{K_H}{2\ {}^cZ_0} \begin{bmatrix} \frac{x_B}{K_H} & \frac{1}{K_H} & 0 & 0 & 0 & \frac{({}^cZ_1 - {}^cZ_2) + x_B({}^cX_1 - {}^cX_2)}{K_H} \\ \frac{y_B}{K_H} - \cos\theta & -\sin\theta & 0 & 0 & 0 & -({}^cZ_1 - {}^cZ_2)\sin\theta - ({}^cX_1 - {}^cX_2)\left(\cos\theta - \frac{y_B}{K_H}\right) \end{bmatrix} \quad (17)$$

rotation) and the follower is equipped with a camera that monitors an extremity of the rope. The maximum authorized rope sag is fixed to $H_{max} = 0.40m$, corresponding to its attachment points height.

The rope is detected in the follower's camera by a color segmentation algorithm, that is refined by morphological skeletonization. The segmented points are then used to compute the visual servoing features.

Three simulations are carried out in order to compare the performances of vision-based tether shape controllers (see Fig. 3). The objective is to study the controllers behavior in normal conditions, when the tether shape is clearly visible, as well as in limit cases, when the tether is not completely visible or when it degenerates into a vertical line in the image.

The first simulation (Fig. 3a) is a typical example where the tether is completely visible and not degenerated in the image. The angle θ is required to vary from $20°$ to $45°$ while the tether slackness must remain the same (initial $H_0 = 0.20m$ and desired $H^* = 0.20m$). In the second simulation (Fig. 3b), the angle θ is required to vary from $10°$ to $60°$ to check controllers performance when the tether is partially visible (θ close to $60°$). The initial and desired slackness are the same ($H_0 = H^* = 0.25m$). In the third simulation (Fig. 3c), the angle θ is required to vary from $20°$ to $0°$ to check the controller performance when the tether plane is near to the camera optical axis (θ close to 0). The initial and desired slackness are $H_0 = 0.10m$ and $H^* = 0.24m$, respectively.



a) normal case b) limit case 1 c) limit case 2

Fig. 3: Follower robot camera view for (a) first, (b) second and (c) third simulation. Initial and desired tether shape in the image are drawn in blue and red lines, respectively

B. Results Analysis

Figure 4 shows, for the three studied cases, the tether shape parameters error and the velocity commands of the follower robot as a function of time. Figure 5 presents the trajectory of the tether lowest point in the image. Figure 6 presents the condition number evolution for the controllers interaction matrices and the Gauss-Newton Jacobian used for curve fitting (Eq. (3)). The condition number of a matrix $\mathbf{A}$ is defined as follows [24]:

$$cond(\mathbf{A}) = |\mathbf{A}||\mathbf{A}^+| \quad (18)$$

where $|.|$ is the L_2 norm.

In the normal case, the tether shape control is achieved by all the controllers. They also allow to reach the desired position in the image plane (see Figs. 4a to 4d and 5a). The tether parameters (slackness H and orientation θ) converge to their desired values even using image-based features. Image-based controllers are very sensitive to the detection of the lowest part of the tether, and work well when this part is well visible in the image. Compared with pure image-based controllers, the catenary lowest-point controller appears to execute a more balanced regulation between slackness (Fig. 4a) and orientation (Fig. 4b). In the image plane, the resulting trajectory of the lowest point is more regular than pure image-based trajectories (Fig. 5a). In addition, the catenary parameter controller is less sensitive to noisy tether detection whereas the other controllers are very dependent on the accurate detection of the lowest point in the image.

In the first limit case, the tether desired shape is only achieved by the catenary parameter controller (Figs. 4e to 4h). The other controllers cannot reach the desired tether shape leading to a residual offset of slackness (Fig. 4e) and angle (Fig. 4f). However, these controllers converge as expected in the image plane, while we observe a discontinuity of the lowest point trajectory in the case of catenary-parameter controller (Fig. 5b). This is not surprising since the catenary-parameter controller directly regulates the tether 3D shape, and converges to the real lowest point that is outside the image.

In the second limit case (see Figs. 4i to 4l), image-based and catenary lowest point controllers achieve the tether desired shape. The image features can be well estimated since at least half of the tether is visible. However, the catenary parameters controller cannot reach the target shape because of fitting failure when the tether gets close to the camera sagittal plane. This is confirmed by the condition number of the curve fitting Gauss-Newton Jacobian (Fig. 6d) that becomes ill-conditioned at the end of the simulation. Nevertheless, all the interaction matrices are well conditioned (Figs. 6a to 6c), which means that they are robust to input noise.

The catenary lowest-point controller makes use of an initial guess of the tether angle parameter. In the proximity of $\theta = 0$, the slackness parameter is not well estimated, since it is subjected to numerical drift resulting from small denominator value in Eq. (4b). In order to improve the catenary parameter controller, a solution will be to directly carry out an initial guess on the slackness parameter.

All four controllers need at least 50% of tether detected points in the image when the tether gets closer to the camera sagittal plane. In real situations it will be difficult to detect half the tether the longer it is. In order to deal with this issue,

a solution could be to equip the robot with an additional camera outside its sagittal plane. Another solution would consist of placing fiducial markers along the tether at known length, which could be easily detected in the image and overcome the noisy detection of the lowest point.

This work is a first step to the regulation of the tether shape inside a chain of robots. In order to navigate in narrow or cluttered environments, the relative position of two successive robots will have to be fully constrained. In this case, the distance between the two attachment points as well as the two relative angles between the robots orientation and the tether plane will have to be regulated. In the considered scenario, the leader robot will be able to move freely while the follower robot will be in charge of maintaining a desired tether slackness. Catenary-based visual servoing will be extended to fulfill this kind of scenario.

V. Conclusions and perspectives

This paper studied vision-based control strategies that can be used to manage the shape of a sagging tether linking two terrestrial robots. Four different visual controllers are tested to regulate the tether shape. Two of them are pure image-based while the other two are catenary-based controllers, one that uses catenary parameters as features and another one that uses image features taking advantage of the catenary model. The study shows that the additional knowledge about the tether 3D shape allows to obtain satisfactory tether handling. Both catenary-based controllers could be combined to design a more robust controller.

Future work will extend tether visual servoing control to 3D holonomic vehicles like underwater robots through the management of vertical translation and pitch rotation. One target application will be the displacement of a chain of tethered robots inside a cluttered environment.

Acknowledgments

This work was achieved thanks to specific funding from the France PACA (Provence-Alpes-Côte-d'Azur) region, and the French SUBSEATECH company, located in Marseille, South-East of France (http://www.subsea-tech.com, pierre.marty@subsea-tech.com).

References

[1] D. Tsai, I. A. D. Nesnas, and D. Zarzhitsky, "Autonomous vision-based tethered-assisted rover docking," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Nov 2013, pp. 2834–2841.

[2] J. Iqbal, S. Heikkila, and A. Halme, "Tether tracking and control of rosa robotic rover," in *Control, Automation, Robotics and Vision, 2008. ICARCV 2008. 10th International Conference on*, Dec 2008, pp. 689–693.

[3] S. Prabhakar and B. Buckham, "Dynamics modeling and control of a variable length remotely operated vehicle tether," in *Proceedings of OCEANS 2005 MTS/IEEE*, Sept 2005, pp. 1255–1262 Vol. 2.

[4] O. Khatib, X. Yeh, G. Brantner, B. Soe, B. Kim, S. Ganguly, H. Stuart, S. Wang, M. Cutkosky, A. Edsinger, P. Mullins, M. Barham, C. R. Voolstra, K. N. Salama, M. L'Hour, and V. Creuze, "Ocean one: A robotic avatar for oceanic discovery," *IEEE Robotics Automation Magazine*, vol. 23, no. 4, pp. 20–29, Dec 2016.

[5] D. P. Perrin, A. Kwon, and R. D. Howe, "A novel actuated tether design for rescue robots using hydraulic transients," in *Robotics and Automation, 2004. Proceedings. ICRA '04. 2004 IEEE International Conference on*, vol. 4, April 2004, pp. 3482–3487 Vol.4.

[6] P. McGarey, F. Pomerleau, and T. D. Barfoot, *System Design of a Tethered Robotic Explorer (TReX) for 3D Mapping of Steep Terrain and Harsh Environments*. Cham: Springer International Publishing, 2016, pp. 267–281.

[7] M. Krishna, J. Bares, and E. Mutschler, "Tethering system design for dante ii," in *Proceedings of International Conference on Robotics and Automation*, vol. 2, Apr 1997, pp. 1100–1105 vol.2.

[8] V. A. K. T. Rajan, A. Nagendran, A. Dehghani-Sanij, and R. C. Richardson, "Tether monitoring for entanglement detection, disentanglement and localisation of autonomous robots," *Robotica*, vol. 34, no. 3, p. 527548, 2016.

[9] Z. Echegoyen, I. Villaverde, R. Moreno, M. Graa, and A. dAnjou, "Linked multi-component mobile robots: Modeling, simulation and control," *Robotics and Autonomous Systems*, vol. 58, no. 12, pp. 1292 – 1305, 2010, intelligent Robotics and Neuroscience. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0921889010001533

[10] D. Navarro-Alarcon, Y.-h. Liu, J. G. Romero, and P. Li, "On the visual deformation servoing of compliant objects: Uncalibrated control methods and experiments," *The International Journal of Robotics Research*, vol. 33, no. 11, pp. 1462–1480, 2014.

[11] T. Dallej, M. Gouttefarde, N. Andreff, R. Dahmouche, and P. Martinet, "Vision-based modeling and control of large-dimension cable-driven parallel robots," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Oct 2012, pp. 1581–1586.

[12] J. Estevez and M. Graña, *Robust Control Tuning by PSO of Aerial Robots Hose Transportation*. Cham: Springer International Publishing, 2015, ch. Bioinspired Computation in Artificial Systems: International Work-Conference on the Interplay Between Natural and Artificial Computation, IWINAC 2015, Elche, Spain, June 1-5, 2015, Proceedings, Part II, pp. 291–300.

[13] T. Lee, "Geometric controls for a tethered quadrotor uav," in *2015 54th IEEE Conference on Decision and Control (CDC)*, Dec 2015, pp. 2749–2754.

[14] M. Laranjeira, C. Dune, and V. Hugel, "Catenary-based visual servoing for tethered robots," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, May 2017, pp. 732–738.

[15] F. Chaumette and S. Hutchinson, "Visual servo control. i. basic approaches," *IEEE Robotics Automation Magazine*, vol. 13, no. 4, pp. 82–90, Dec 2006.

[16] ——, "Visual servo control. ii. advanced approaches [tutorial]," *IEEE Robotics Automation Magazine*, vol. 14, no. 1, pp. 109–118, March 2007.

[17] R. I. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, 2nd ed. Cambridge University Press, 2004, ch. 6, p. 155.

[18] C. Collewet and F. Chaumette, "Positioning a camera with respect to planar objects of unknown shape by coupling 2-d visual servoing and 3-d estimations," *IEEE Transactions on Robotics and Automation*, vol. 18, no. 3, pp. 322–333, Jun 2002.

[19] A. Cherubini, F. Chaumette, and G. Oriolo, "An image-based visual servoing scheme for following paths with nonholonomic mobile robots," in *2008 10th International Conference on Control, Automation, Robotics and Vision*, Dec 2008, pp. 108–113.

[20] B. Espiau, F. Chaumette, and P. Rives, "A new approach to visual servoing in robotics," *IEEE Transactions on Robotics and Automation*, vol. 8, no. 3, pp. 313–326, Jun 1992.

[21] F. Chaumette, P. Rives, and B. Espiau, "Classification and realization of the different vision-based tasks," *Visual servoing*, vol. 7, pp. 199–228, 1993.

[22] N. Koenig and A. Howard, "Design and use paradigms for gazebo, an open-source multi-robot simulator," in *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (IEEE Cat. No.04CH37566)*, vol. 3, Sept 2004, pp. 2149–2154 vol.3.

[23] B. Gerkey and K. Conley, "Robot developer kits [ros topics]," *IEEE Robotics Automation Magazine*, vol. 18, no. 3, pp. 16–16, Sept 2011.

[24] J. T. Feddema, C. S. G. Lee, and O. R. Mitchell, "Automatic selection of image features for visual servoing of a robot manipulator," in *Proceedings, 1989 International Conference on Robotics and Automation*, May 1989, pp. 832–837 vol.2.

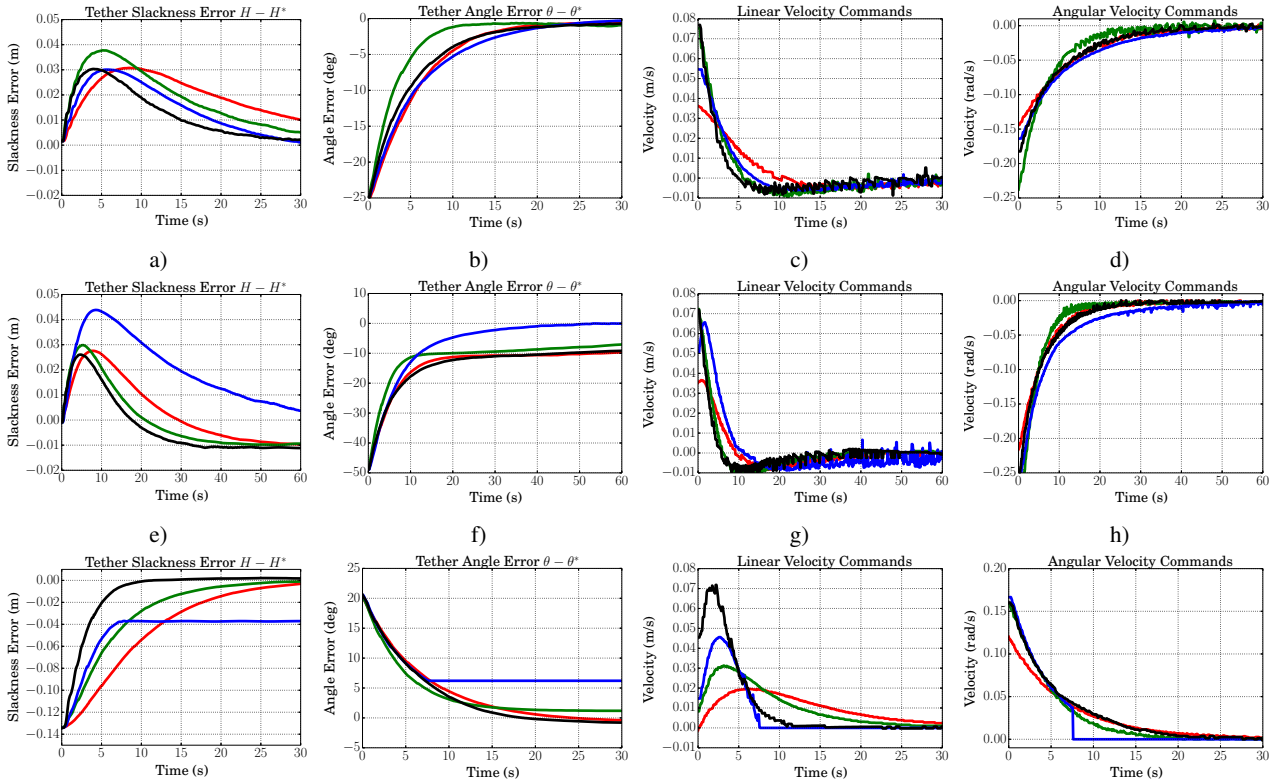


Fig. 4: Error evolution for tether slackness and orientation as well as linear and angular command velocities for normal case (Figs. a, b, c and d), limit case 1 (Figs. e, f, g and h) and limit case 2 (Figs. i, j, k and l). Tether initial and desired shape are respectively: $(H_0 = 0.20m, \theta_0 = 20°)$ and $(H^* = 0.20m, \theta^* = 45°)$; $(H_0 = 0.25m, \theta_0 = 10°)$ and $(H^* = 0.25m, \theta^* = 60°)$; $(H_0 = 0.10m, \theta_0 = 20°)$ and $(H^* = 0.24m, \theta^* = 0°)$. Legend in Fig. 5d. All controllers use a gain $\lambda = 1.0$.

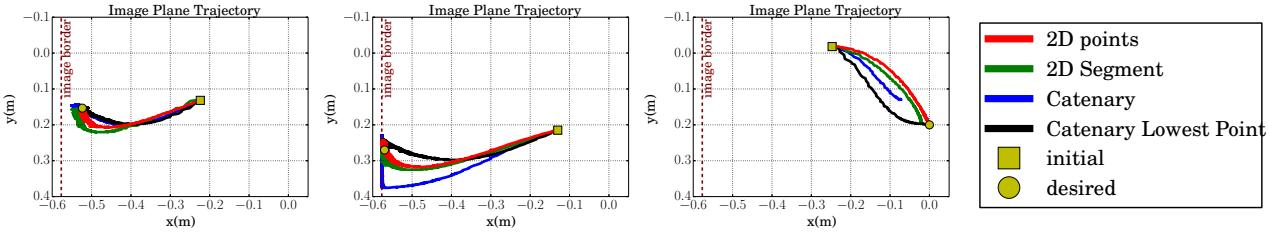


Fig. 5: Tether lowest point trajectory in the image plane for (a) normal case, (b) limit case 1 and (c) limit case 2. (d) legend

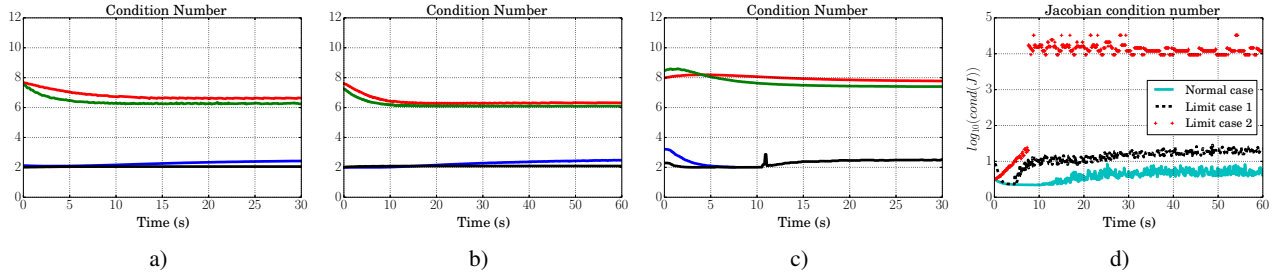


Fig. 6: Interaction matrices condition number evolution for (Fig. a) normal and (Figs. b and c). Legend in Fig. 5d. (d) condition number evolution for the curve fitting Gauss-Newton Jacobian during normal and limit cases

Appendix F

Interaction Matrix Test Protocol

Introduction

This appendix depicts a protocol to validate the interaction matrices in the context of visual servoing of a pair of underwater robots linked with a cable. The robots are assumed to remain horizontal. 4 degrees of freedom (DOF) are used: three translational (ν_x, ν_y, ν_z) and one rotational (ω_z).

Methodology

Definition

Let $\mathbf{v}$ be the Cartesian velocity of an embedded sensor and $\mathbf{s}$ a set of visual features. The interaction matrix $\mathbf{L}$ links the features time variations $\dot{\mathbf{s}}$ with the sensor motion $\mathbf{v} = [\nu_x, \nu_y, \nu_z, \omega z]$. It is defined as:

$$\dot{\mathbf{s}}(t) = \mathbf{L}(s,t)\mathbf{v}(t) \tag{F.1}$$

Feature trajectories estimation

Let us consider a discrete version $\mathbf{s}_k$ of $\mathbf{s}$ at time samples kdt, where $k \in \mathbb{N}$ and dt the time step duration, in seconds. Linearizing equation (F.5) leads to a recursive expression of $\mathbf{s}_k$:

$$\begin{cases} \mathbf{s}_0 = \mathbf{s}(t=0) \\ \mathbf{s}_{k+1} = \mathbf{s}_k + dt\mathbf{L}_k(\mathbf{s}_k)\mathbf{v}_k \end{cases} \tag{F.2}$$

Interaction matrices validation

In order to evaluate the interaction matrix $\mathbf{L}$, the measured features $\mathbf{s}(kdt)$ can be compared with the estimated features s_k for small displacements of the sensor. One dimension will be tested after the other. It allows to evaluate the matrix components independently.

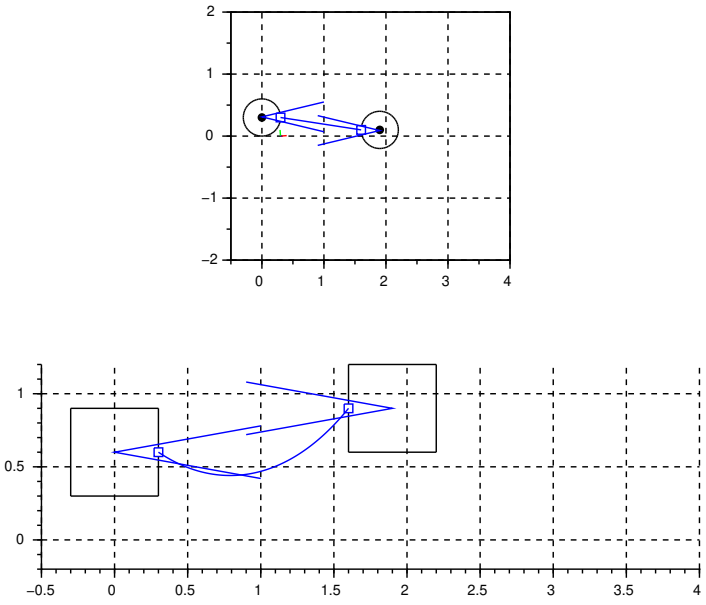


Figure F.1: Top view and lateral view of the robot set up.

For example, if $\mathbf{s}$ is a 3 dimensional, evaluating the first features $\mathbf{s}[0]$ for a forward motion in the X-direction corresponds to the component $\mathbf{L}_{1,1}$ of the interaction matrix. It allows to check the sign of the component and its rough evolution w.r.t features.

Matrices evaluation

Simulation set up

The matrices are tested on Scilab.

Let us introduce two robots that are symmetrically set and have the following positions in a global world reference frame (see figure F.1).

$$\begin{cases} {}^{W}\mathbf{P}_{r_2} = [0, 0, 0.2, 0, 0, 0] \\ {}^{W}\mathbf{P}_{r_1} = [1.9, 0.1, 0.5, 0, 0, \pi] \end{cases} \tag{F.3}$$

They are linked with a catenary shaped 1.5m length cable. The maximal sag is $H_{max} = 0.7m$ and the maximal difference between the attachment points is $\Delta H_{max} = 0.5m$.

Leader and follower interaction matrix evaluation

Let us consider the $\mathbf{L}_{r_2,2}$ and $\mathbf{L}_{r_1,2}$ interaction matrices, and the follower robot velocity ${}^{\mathbf{r_2}}\mathbf{v}$. In our case, the interaction matrix is directly computed in the robot frame. Then,

classical sensor velocity is changed for robot velocity. The follower $\mathbf{s}_2$ and leader features $\mathbf{s}_1$ can be expressed as

$$\dot{\mathbf{s}}_1 = \left[a_1, b_1, d_1\right]^T = \mathbf{L}_{r_1,2}{}^{r_2}\mathbf{v} \tag{F.4}$$

$$\dot{\mathbf{s}}_2 = \left[a_2, b_2, d_2\right]^T = \mathbf{L}_{r_2,2}{}^{r_2}\mathbf{v} \tag{F.5}$$

We will test the degrees of freedom one by one and compare. The estimated features are drawn in dashed line and the real features are in solid line.

Follower velocity in the X direction: test of first column

$$^{r_2}\mathbf{v} = [\pm 0.1, 0.0, 0.0, 0.0] \tag{F.6}$$

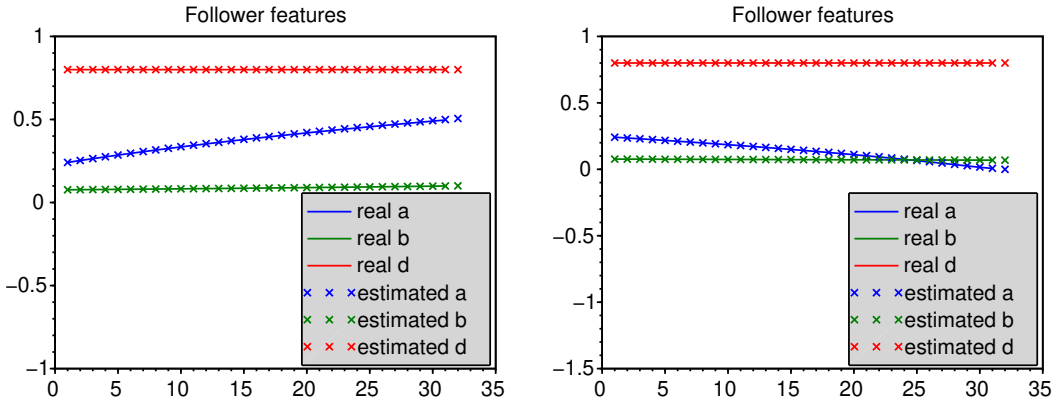


Figure F.2: The follower moves along the X-axis. Left: follower features when $\nu_x > 0$, Right: follower features when $\nu_x < 0$ (blue a_i, red b_i, green d_i)

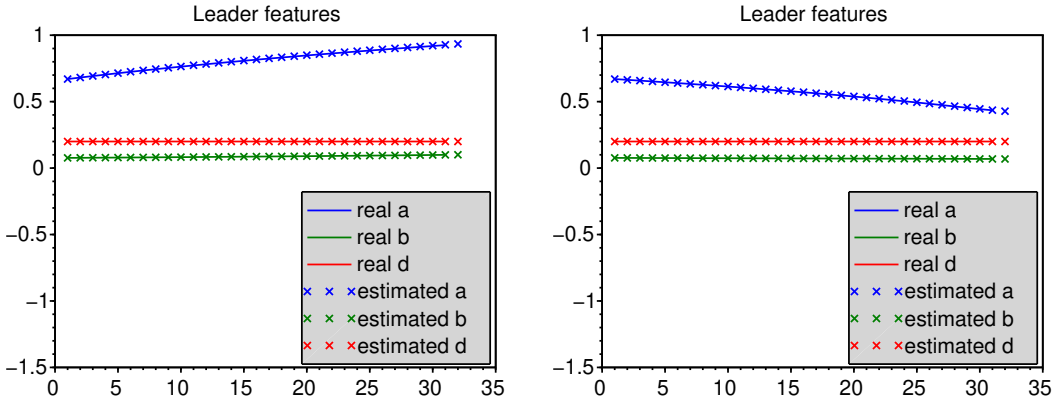


Figure F.3: The follower moves along the X-axis. Left: leader features when $\nu_x > 0$, Right: leader features when $\nu_x < 0$ (blue a_i, red b_i, green d_i)

Follower velocity in the Y direction: test of second column

$$^{r_2}\mathbf{v} = [0.0, \pm 0.1, 0.0, 0.0] \tag{F.7}$$

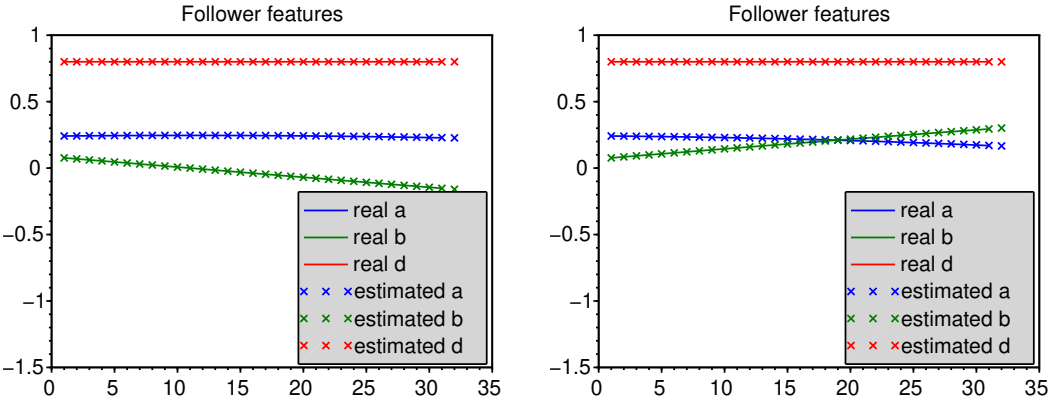


Figure F.4: The follower moves along the Y-axis. Left: follower features when $\nu_y > 0$, Right: follower features when $\nu_y < 0$ (blue a_i, red b_i, green d_i)

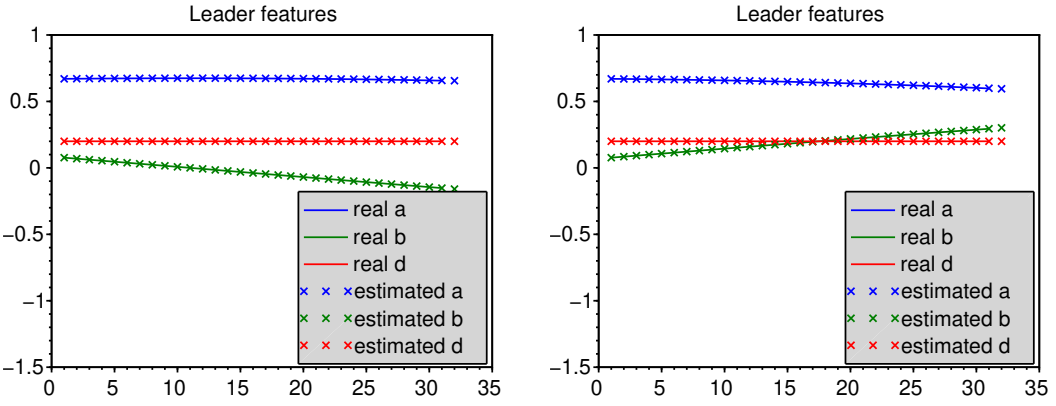


Figure F.5: The follower moves along the Y-axis. Left: leader features when $\nu_y > 0$, Right: leader features when $\nu_y < 0$ (blue a_i, red b_i, green d_i)

Follower velocity along the Z-axis: test of third column

$$^{r_2}\mathbf{v} = [0.0, 0.0, \pm 0.1, 0.0] \tag{F.8}$$

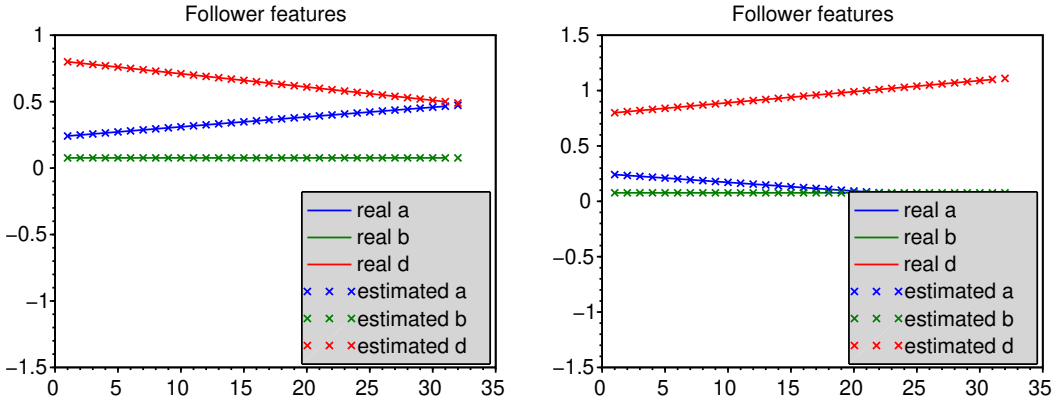


Figure F.6: The follower moves along the Z-axis. Left: follower features when $\nu_z > 0$, Right: follower features when $\nu_z < 0$ (blue a_i, red b_i, green d_i)

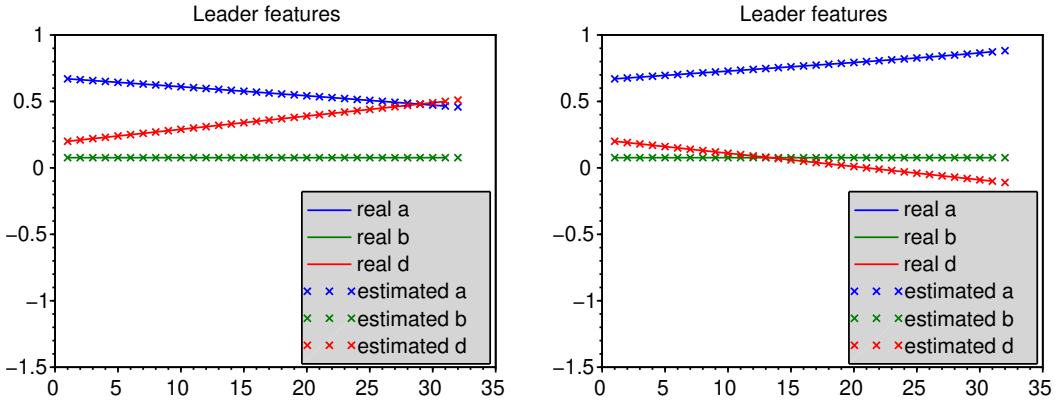


Figure F.7: The follower moves along the Z-axis. Left: leader features when $\nu_z > 0$, Right: leader features when $\nu_z < 0$ (blue a_i, red b_i, green d_i)

Follower velocity around the Z-axis: test of fourth column

$$^{r_2}\mathbf{v} = [0.0, 0.0, 0.0, \pm 0.1] \tag{F.9}$$

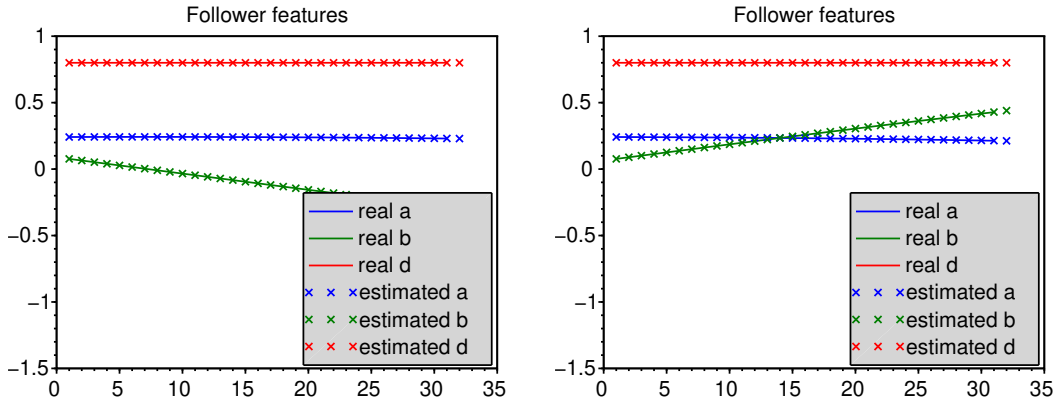


Figure F.8: The follower turns around the Z-axis. Left: follower features when $\omega_z > 0$. Right: follower features when $\omega_z < 0$. (blue a_i, red b_i, green d_i)

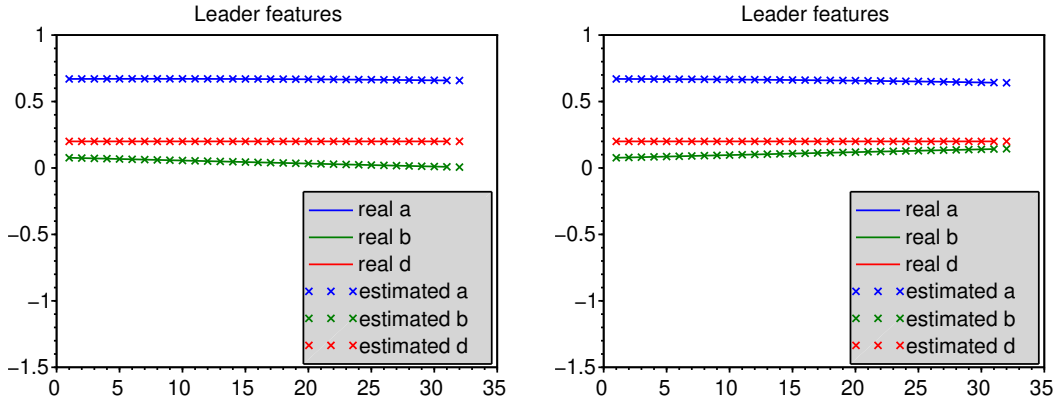


Figure F.9: The follower turns around the Z-axis. Left: leader features when $\omega_z > 0$. Right: leader features when $\omega_z < 0$. (blue a_i, red b_i, green d_i)

Conclusion

The estimated and real features are superimposed, which validates both matrices $\mathbf{L}_{r_1,2}$ and $\mathbf{L}_{r_2,2}$.

Appendix G

Scientific publications, Workshop Participations and Scientific Popularization Activities

International Conferences

1. M. Laranjeira, C. Dune and V. Hugel, **Catenary-Based Visual Servoing for Tethered Robots**, In *IEEE Int. Conf. on Robotics and Automation, ICRA'17*, Pages. 732-738, Singapore, 2017.

2. M. Laranjeira, C. Dune and V. Hugel, **Local Vision-Based Tether Control for a Line of Underwater Robots**, In *Workshop on "New Horizons for Underwater Intervention Missions: from Current Technologies to Future Applications" at IROS 2018*, Madrid, Spain, 2018.

3. M. Laranjeira, C. Dune and V. Hugel, **Embedded Visual Detection and Shape Identification of Underwater Umbilical for Vehicle Positioning**, In *MTS/IEEE OCEANS 2019*. Accepted.

Others

1. M. Laranjeira, "Contrôle référencé vision pour la robotique sous-marine", *8es Rencontres CARTT'16*, Toulon, France, 2016.

2. M. Laranjeira, "La commande de robots interconnectés", *Fête de la science 2016 à l'Université de Toulon*, Toulon, France, 2016.

3. M. Laranjeira, "Asservissement visuel pour objets déformables paramétriques", *9es Rencontres CARTT'17*, Toulon, France, 2017.

4. M. Laranjeira, "Étude comparative de contrôleurs référencés vision pour la commande coordonnée d'un ombilical", *Fête de la science 2017 à l'Université de Toulon*, Toulon, France, 2017.

5. M. Laranjeira, "Asservissement visuel appliqué au contrôle de forme d'ombilicaux pour la robotique sous-marine", *Journée GT2 Robotique marine et sous-marine*, Paris, France, 2017.

Appendix H

Résumé en français

Introduction

En robotique sous-marine, deux types de robots sont différenciés par la présence ou l'absence d'un ombilical les reliant à un navire de surface. On distingue ainsi les robots autonomes, dits AUV (autonomous underwater vehicles) et les robots téléopérés, dits ROVs (Remotely Operated Vehicles). Les robots autonomes ont une grande liberté de déplacement mais sont limités par leur autonomie et par l'intelligence embarquée. L'eau absorbant fortement les ondes, ils n'ont pas ou peu de moyen de communication avec la surface durant leur immersion. Ils sont généralement dédiés à des tâches de cartographie où leur mission est d'effectuer un parcours prédéfini pour couvrir une zone. Les ROVs, quant à eux, sont privilégiés dans des tâches d'exploration ou de maintenance pour lesquelles un opérateur en surface prend les décisions. Le câble sert de support au transfert des données, à haut débit. Il peut également transmettre de l'énergie et servir de support mécanique pour ramener le véhicule au navire de surface. Cependant, il limite la zone de déploiement et peut s'emmêler au fond ou s'accrocher à des objets dérivants, comme des filets.

La majorité des ROVs actuellement en service sont des engins de grande dimension qui nécessitent des systèmes de mise à l'eau semblables à des grues (LARS, launch and recovery systems) et des navires de surfaces de grand tonnage, dont le coût de fonctionnement est prohibitif. Le développement d'engins de petite taille, de moindre coût et pouvant être déployés par une ou deux personnes depuis une petite embarcation a un intérêt économique évident. Cependant, plus l'engin est petit, plus sa puissance et son inertie sont faibles et plus les effets hydrodynamiques s'exerçant sur le câble impactent sa manoeuvrabilité. Le laboratoire COSMER et la société Subseatech ont proposé d'aligner plusieurs petits ROVs afin de résoudre ce problème. L'union des forces de propulsion des ROVs leur permet de lutter efficacement contre les forces hydrodynamiques et libère le ROV de tête, le leader, pour des tâches d'exploration.

Nous appelons ce concept la **cordée de robots** (Fig. H.1). Les robots qui composent la cordée sont compacts, légers et avec une charge utile sensorielle limitée. Chaque robot joue le rôle d'actionneur et modifie la forme du câble en fonction de la situation. Notre stratégie est d'éviter tout contact avec les obstacles pour limiter les

emmêlements. Si la profondeur est trop faible, il est préférable de maintenir l'ombilical plus tendu afin d'éviter qu'il ne traîne sur le fond marin. Sinon, si l'environnement est plus large, l'ombilical peut être plus lâche afin de donner plus de liberté de mouvement aux robots.

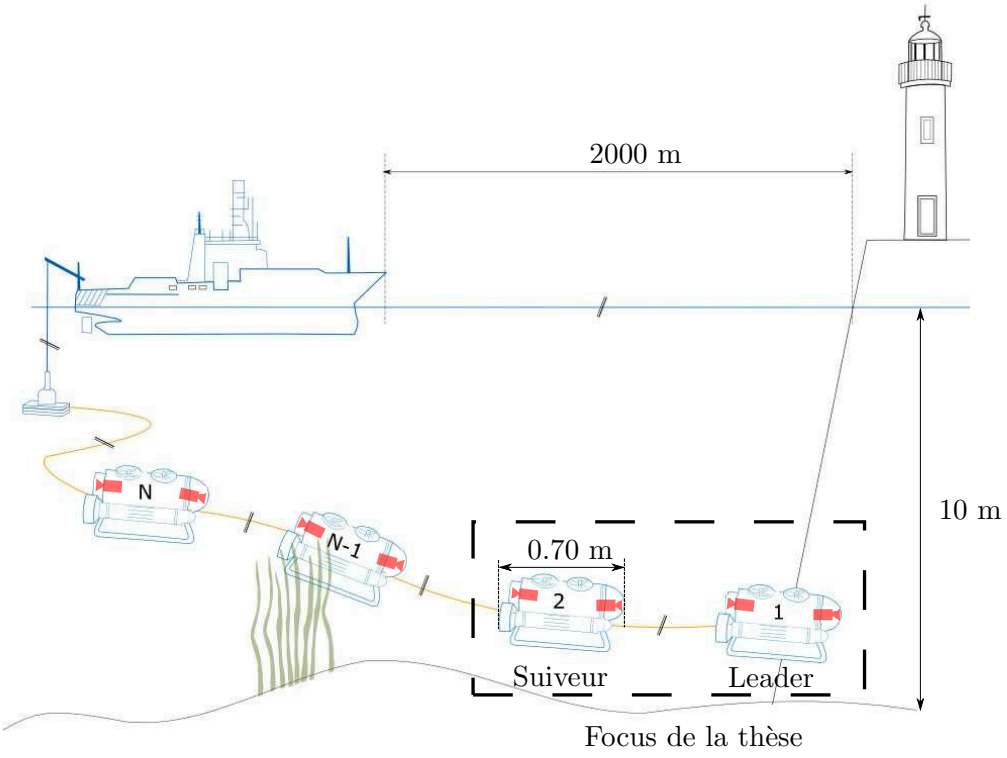


Figure H.1: Une chaîne de N robots reliés compacts utilisés pour explorer les eaux peu profondes. La thèse porte sur le contrôle de la forme du câble entre les deux premiers robots, à savoir le leader et son suiveur immédiat. Les robots qui composent la chaîne sont des mini-ROV légers équipés d'une caméra frontale, d'une caméra arrière, d'une centrale inertielle et d'un profondimètre.

Dans le cadre de cette thèse, seuls les capteurs embarqués sur les mini-ROVs sont utilisés, soit principalement les caméras embarquées, et de manière secondaire les profondimètres et les centrales inertielles. Le câble n'est pas instrumenté. Nous nous sommes intéressés plus particulièrement à la première paire de robots, supposant que le comportement obtenu pourrait ensuite servir de contrôleur bas niveau entre deux éléments de la cordée et être répliqué sur les autres paires de robots.

Les développements réalisés dans le cadre de cette thèse visent à définir des stratégies de contrôle d'objets déformables par asservissement visuel, appliquées aux objets de type câble reliant deux robots sous-marins, un leader et un suiveur, en utilisant le retour des caméras embarquées. Les deux robots ont les mêmes capacités de mouvement et sont supposés rester à l'horizontale, les angles de roulis et de tangage étant réglés de façon autonome par un contrôleur bas niveau haute fréquence.

L'objectif principal est de prouver la faisabilité d'un schéma de commande d'asservissement visuel qui permette au robot suiveur de réguler la forme de l'ombilical en libérant le robot leader pour explorer son environnement, en autonomie ou par téléopération.

Les contributions principales de cette thèse sont les suivantes :

- L'estimation en temps réel de la forme du câble par une procédure de mise en correspondance de sa projection dans le plan l'image avec le modéle géométrique de type chaînette.
- La conception d'une stratégie de régulation de la forme du câble grâce à une technique novatrice d'asservissement visuel basée sur les paramètres du modèle de chaînette.
- Le positionnement relatif de deux robots sous-marins reliés par un câble grâce à la perception locale du câble lui-même à partir de leurs caméras et grâce aux capteurs de profondeur intégrés.

Modélisation

Le fonctionnement de la chaîne proposée de mini-ROVs suppose les hypothèses suivantes :

1. la distance maximale entre robots est d'environ 10 mètres ;
2. les robots peuvent naviguer à des profondeurs légèrement différentes (différence inférieure à 5 mètres, en fonction de la longueur de l'ombilical) ;
3. le mouvement de roulis et de tangage des robots est mécaniquement stabilisé ou régulé à bas niveau pour maintenir le véhicule horizontal ;
4. les robots sont équipés d'une caméra frontale et/ou arrière qui filment le câble ;
5. chaque robot dans la chaîne doit gérer le câble qui le précède ;
6. le robot leader ne gère aucune partie de l'ombilical, et reste libre pour explorer son environnement et pour exécuter d'autres tâches ;
7. le robot leader peut se trouver en dehors du champ de vision de la caméra du robot suiveur, mais une partie du câble est toujours visible.
8. le câble est détectable dans le flux d'images de la caméra ;
9. l'ombilical reliant les deux robots est pesant et le plan du câble reste dans le plan vertical. Le point le plus bas est toujours situé entre les deux robots ;

Le câble reliant les robots est modélisé par une chaînette, dont les paramètres géométriques servent d'entrées aux algorithmes de contrôle. Le paramétrage est rendu symétrique par rapport aux points d'attache. Ces paramètres sont définis par rapport à chaque robot, à savoir la différence de hauteur entre les points d'attache ΔH_i, la hauteur H_i de câble et l'orientation α_i du plan du câble (Fig. H.2).

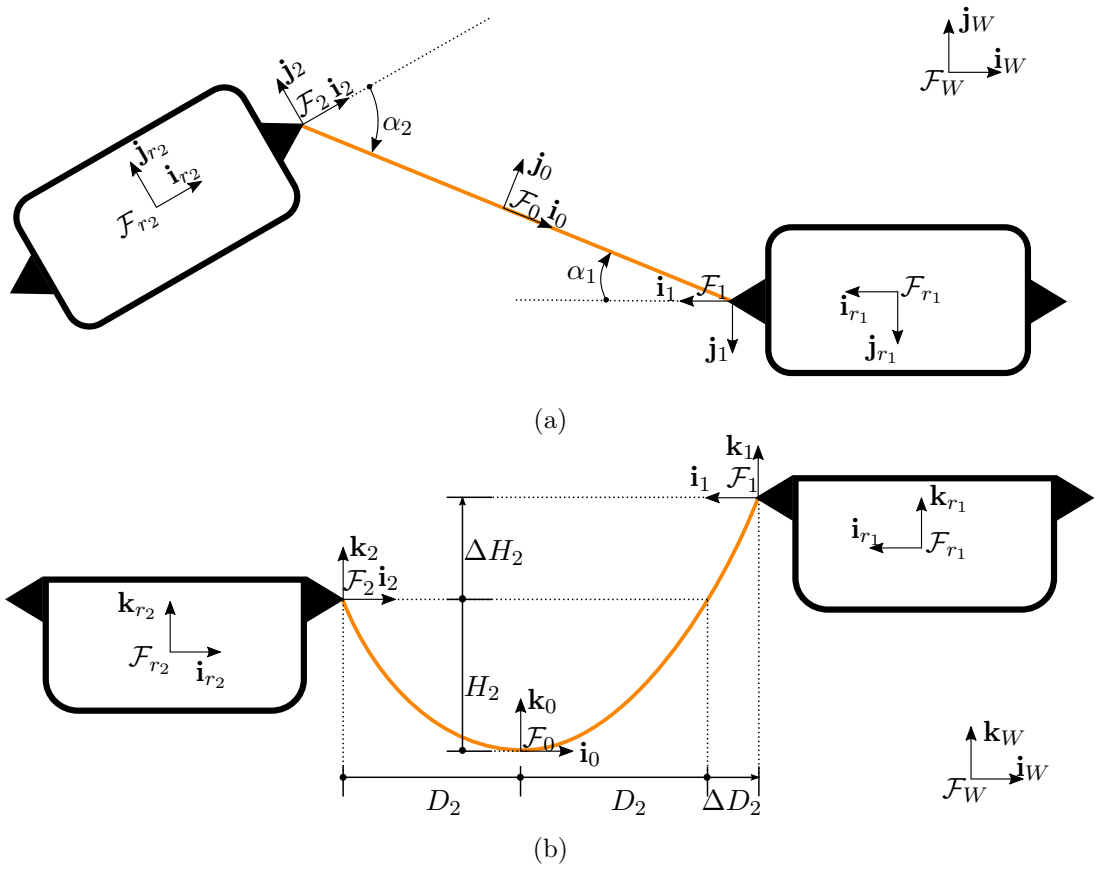


Figure H.2: (a) vue de dessus et (b) vue de côté de la première paire de robots composant la chaîne des robots. L'ombilical reliant les robots est modélisé par une chaînette dont la hauteur est notée H_2. La différence de hauteur entre les points d'attache du côté suiveur est ΔH_2. La portée horizontale de chaînette correspondant à la hauteur H_2 est notée D_2 et ΔD_2 est la portée supplémentaire due à la différence de hauteur. Les angles d'orientation du câble par rapport à chaque repère de point d'attache sont notés α_1 et α_2, respectivement pour les points d'attache du câble au robot de tête et au robot suiveur.

Perception proprioceptive du câble reliant deux robots sous-marins

Les robots sont équipés de capteurs de profondeur, et de caméras placées à proximité des points d'attache du câble. La différence de hauteur entre les points de fixation est calculée à partir des mesures prises par les capteurs de profondeur, tandis que la hauteur et l'orientation du câble sont estimées par une procédure de mise en correspondance dans le plan image basée sur l'algorithme de Gauss-Newton. Les points de la projection du câble dans l'image sont détectés par segmentation couleur dans l'espace HSV puis par l'application d'opérateurs morphologiques : ouverture, fermeture et squelettisation. La projection du modèle de chaînette dans le plan image est ainsi ajustée aux

points détectés, ce qui permet d'estimer les paramètres correspondant à la hauteur et à l'orientation du câble.

Le schéma général de l'algorithme de Gauss-Newton amélioré proposé pour l'estimation des paramètres de chaînette est présentée dans la figure H.3. Après l'acquisition d'une nouvelle image et la détection des points projetés dans l'image, l'estimation de la forme du câble passe par le calcul des résidus qui représentent la différence dans l'image entre les points détectés et la courbe à ajuster. Des fonctions de traitement des résidus ont été ajoutées à la procédure d'ajustement afin d'éviter les problèmes de calcul numérique dans la matrice Jacobienne de Gauss-Newton, afin d'accélérer son calcul et d'améliorer la robustesse de l'algorithme au bruit de détection. Elles sont définies suivant quatre étapes. L'étape 1 consiste à filtrer les résidus en éliminant les points correspondants à des écarts trop importants entre la courbe paramétrée et la courbe formée par les points détectés. L'étape 2 effectue le sous-échantillonnage des points 2-D de l'image afin de retenir un seul point par abscisse. L'étape 3 fait appel à la méthode des M-estimateurs afin de pondérer les écarts et de favoriser les plus faibles. L'étape 4 permet d'assurer que les paramètres restent bien dans leur domaine de définition.

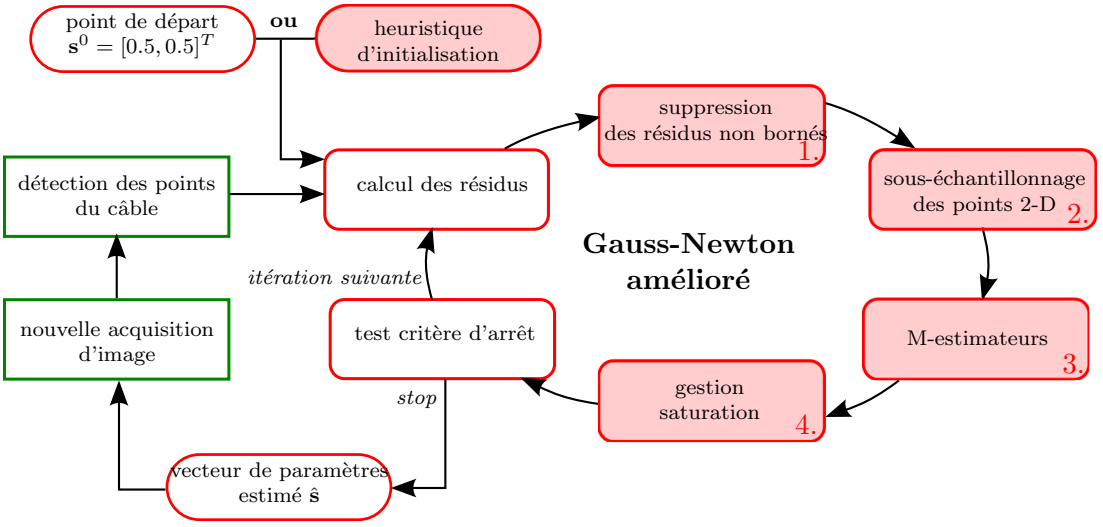


Figure H.3: Schéma de l'algorithme de Gauss-Newton amélioré avec les quatre étapes numérotées et l'heuristique d'initialisation.

De plus, une heuristique d'initialisation a été introduite afin d'obtenir un point de départ plus proche de la solution et ainsi éviter les minima locaux. L'heuristique d'initialisation est définie en considérant que la projection dans l'image du point le plus bas de la chaînette 3D est approximée avec le point le plus bas détecté dans l'image.

L'ajustement des points détectés sur une courbe de type chaînette a été validé en simulation et en situation réelle en bassin ; les résultats obtenus ont montré que l'estimation de la forme de l'ombilical était suffisante pour notre application lorsqu'au moins 30% des points de la projection du câble étaient détectés dans l'image.

Contrôle de la forme du câble par asservissement visuel

Le schéma global de commande du robot suiveur est décrit dans la figure H.4. La partie encadrée en bleu désigne le cœur des développements effectués en terme d'asservissement visuel sur les paramètres du câble qui se déforme en fonction des mouvements des points d'attache sur le robot de tête et sur le robot suiveur (Fig. H.5).

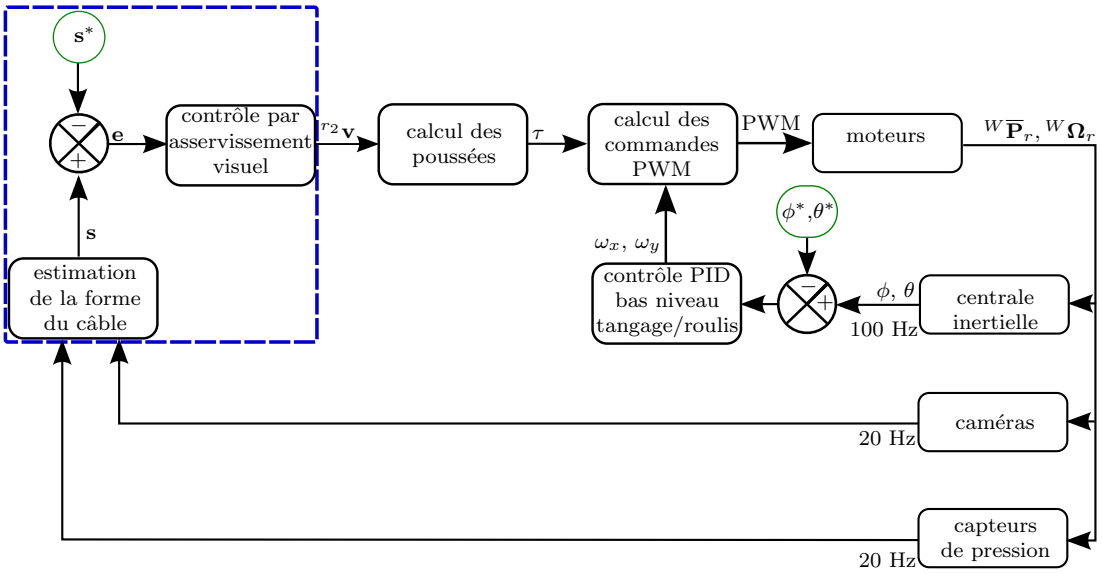


Figure H.4: Schéma global de contrôle par asservissement visuel basé chaînette du robot sous-marin suiveur

.

La forme de l'ombilical est régulée par le mouvement du robot suiveur qui génère un déplacement adéquat du point d'attache de l'ombilical. La figure H.5 décrit l'algorithme de contrôle par asservissement visuel qui consiste à calculer la vitesse que le robot suiveur doit produire afin de conserver la forme désirée du câble dans l'image de sa caméra embarquée. Les paramètres estimés du modèle de type chaînette sont regroupés dans un vecteur de paramètres, appelé **s**, qui doit être régulé afin de maintenir le câble à la forme désirée. Il doit être suffisamment lâche pour ne pas gêner les mouvements du robot de tête. De plus, une orientation adéquate entre chaque robot et le plan du câble doit être maintenue afin de conserver une vue du câble à l'intérieur du champ de vision de la caméra. Cela permet d'obtenir une estimation précise de la forme à partir du retour d'information de la caméra. Cet algorithme passe par le calcul d'une matrice d'interaction 3D, qui traduit la relation entre le mouvement du robot suiveur et les dérivées temporelles des paramètres de la chaînette. Un formalisme général pour le calcul de cette matrice d'interaction est proposé, permettant la régulation des paramètres du modèle de type chaînette par rapport aux robots leader et suiveur. Ce formalisme décompose la matrice d'interaction en deux parties, une partie décrivant la cinématique du système, et une deuxième partie décrivant le lien entre le mouvement

du point d'attache et la variation des paramètres du câble.

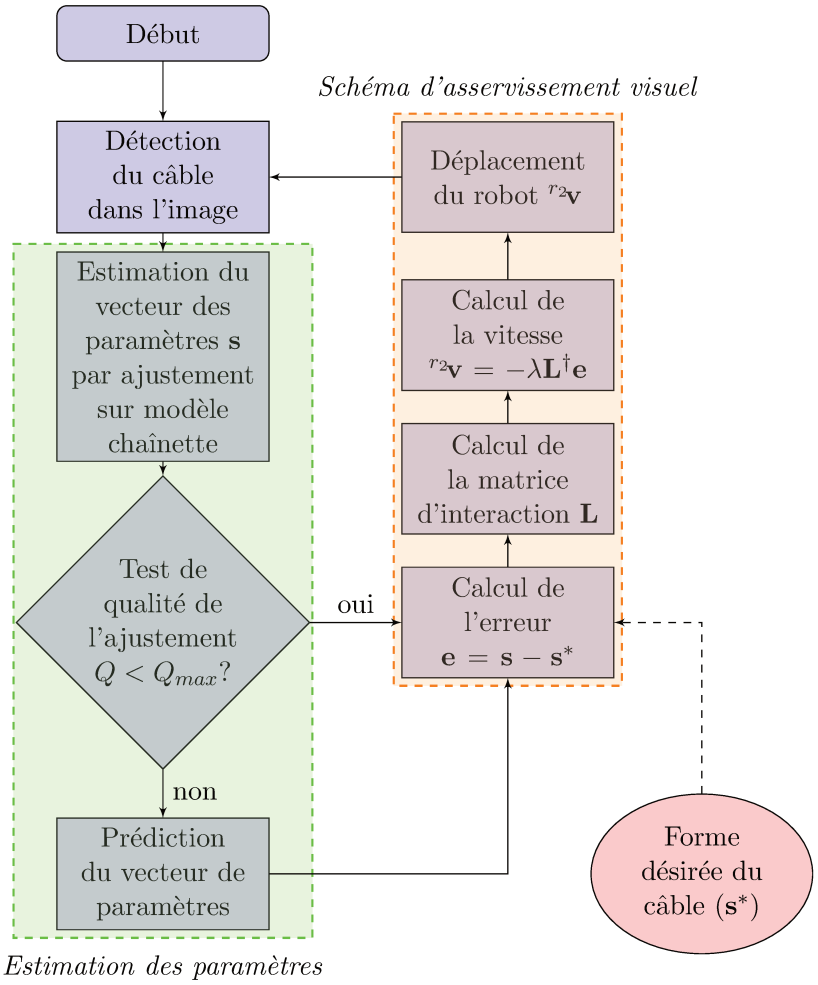


Figure H.5: Algorithme d'asservissement visuel basé chaînette.

Le schéma de commande d'asservissement visuel basé sur le modèle de chaînette a été validé en simulation. Deux variantes de loi de commande proportionnelle ont été testées, basées respectivement sur une utilisation simultanée ou hiérarchique des paramétres estimés du câble. Le schéma de commande proposé est capable de gérer correctement la forme du câble, même dans des conditions d'estimation grossières de la forme, en cas de visibilité partielle du câble.

Tout comme les schémas de commande de régulation simultanée des paramètres, les schémas de commande hiérarchiques ont également donné des résultats satisfaisants en termes de rapidité, d'efficacité et de robustesse aux perturbations externes. De fait, la matrice d'interaction générique introduite dans ces travaux s'est révélée efficace, puisque toutes les caractéristiques ont été bien suivies par le robot. Les schémas de contrôle hiérarchique peuvent être utiles dans la mise en œuvre de stratégies de nav-

igation de haut niveau au sein d'une cordèe de robots sous-marins afin de donner la priorité au placement d'un des robots par rapport au câble.

Perspectives

Dans la perspective du contrôle de la forme des câbles, une extension intéressante de ces travaux de thèse est l'utilisation d'informations sensorielles supplémentaires afin de fournir une estimation fiable de la forme pour des câbles longs (de plusieurs dizaines ou centaines de mètres de long). Les caméras ont une portée limitée et le câble reste visible jusqu'à quelques mètres seulement. De plus, les opérations en eau trouble peuvent rendre difficile la détection des points du câble dans l'image, ce qui compromet l'estimation de sa forme à partir du retour visuel de la caméra.

Des capteurs supplémentaires, tels que des accéléromètres mono ou tridimensionnels, peuvent être ajoutés aux extrémités du câble afin de fournir des informations supplémentaires sur sa forme. Ces capteurs pourraient être utiles pour mesurer les angles de départ du câble, et ainsi calculer la hauteur dans le cas d'un modèle de type chaînette. De plus, les accéléromètres peuvent fournir des informations sur le mouvement de balancement du câble, ce qui n'a pas été pris en compte dans les schémas de commande proposés.

Même dans le cas de la régulation de la forme de gros câbles, les caméras pourraient fournir des informations exploitables sur l'orientation des câbles par rapport aux robots. L'utilisation des caméras sur le robot de tête et le robot suiveur peut également fournir des informations redondantes sur le relâchement du câble, qui peuvent être utilisées avec les données des accéléromètres afin d'améliorer la fiabilité de l'estimation de la forme du câble.

Le concept de contrôle de la forme du câble par modèle de chaînette entre deux robots peut également être étendu à une chaîne composée de plusieurs robots sous-marins déployés par un navire de surface. Le schéma de commande d'asservissement visuel que nous avons introduit pourrait être adapté à la régulation de l'ombilical reliant deux robots consécutifs. Le mouvement d'un robot ayant un effet sur les autres, un contrôle de haut niveau doit être mis en place pour éviter l'effet de platooning.

La navigation de toute la chaîne des robots est un thème de recherche prometteur à développer. Le contrôleur de haut niveau doit gérer la position relative entre les robots en fonction de la situation. Si de petites cavités doivent être traversées, les robots doivent être le plus souvent alignés et l'ombilical pas trop lâche pour ne pas traîner sur le sol. Sinon, lors de l'exploration de sites ouverts, il serait préférable que les robots soient plus espacés latéralement afin d'avoir une meilleure perception de l'environnement. Les schémas de contrôle hiérarchiques peuvent être utilisés pour que la priorité soit donnée à l'orientation ou au relâchement, selon la situation.

Des configurations géométriques autres qu'une chaîne de robots peuvent être utilisées. Des réseaux maillés de formation triangulaire ou carrée peuvent être utiles pour explorer des sites ouverts et obtenir une perception plus distribuée de l'environnement.

Les applications possibles de ces travaux comprennent l'exploration des zones côtières

par de petits robots sous-marins reliés, appelés mini-ROV. De petits robots sous-marins sont nécessaires pour effectuer des missions d'exploration de ces zones en raison de leur faible profondeur. Seuls, ces robots ne peuvent pas contrebalancer les perturbations causées par les marées et les courants agissant sur le câble. Ensemble, cependant, ils peuvent contrôler plus efficacement la forme de la laisse. De plus, l'utilisation de plusieurs robots reliés par un câble permet une perception distribuée de l'environnement sous-marin pour l'opérateur humain en surface, ce qui est utile pour la cartographie et la collecte synchronisée des données de l'environnement.

Le contrôle de la forme du câble basé chaînette peut également être employé dans le contexte de l'exploration d'environnements encombrés, tels que les grottes sous-marines et les sites miniers submergés. Dans de telles situations, la gestion correcte de la forme du câble est cruciale pour le succès de la mission, car elle permet d'éviter la perte des robots suite à des accrochages avec les obstacles.

D'un point de vue général, les travaux présentés dans cette thèse ont prouvé que le câble peut être utilisé pour le positionnement relatif entre les robots, ou entre un robot et une base fixe. Le concept peut être mis à profit, par exemple, par un robot qui a pour mission de s'ancrer au relief du fond marin ou à une structure sous-marine afin de stabiliser sa position pour l'exécution de sa mission. Un autre exemple d'application serait le positionnement relatif entre un navire de surface et un robot éclaireur.

Bibliography

Abel, B. A. (1994). Underwater Vehicle tether management systems. In *Proceedings of OCEANS'94*, volume 2, pages II/495–II/500 vol.2.

Aili, A. and Ekelund, E. (2016). *Model-Based Design, Development and Control of an Underwater Vehicle*. Masters thesis, Linköping University.

Allen, P. K., Timcenko, A., Yoshimi, B., and Michelman, P. (1993). Automated tracking and grasping of a moving object with a robotic hand-eye system. *IEEE Transactions on Robotics and Automation*, 9(2):152–165.

Antonelli, G., Fossen, T. I., and Yoerger, D. R. (2008). Underwater robotics. In *Springer Handbook of Robotics*, pages 987–1008. Springer.

Bell, M. P. (2010). *Flexible Object Manipulation*. PhD thesis.

Bessa, W. M., Dutra, M. S., and Kreuzer, E. (2006). Thruster dynamics compensation for the positioning of underwater robotic vehicles through a fuzzy sliding mode based approach. In *ABCM Symposium Series in Mechatronics*, volume 2, pages 605–612.

BlueRobotics (2019). T200 Thruster Characteristics. [Online; accessed 01-march-2019].

Bowen, A. D., Jakuba, M. V., Yoerger, D. R., Whitcomb, L. L., Kinsey, J. C., Mayer, L., and German, C. R. (2012). Nereid UI: A Light-Tethered Remotely Operated Vehicle for Under-Ice Telepresence. In *OTC Arctic Technology Conference*. Offshore Technology Conference.

Bretl, T. and McCarthy, Z. (2014). Quasi-static manipulation of a Kirchhoff elastic rod based on a geometric analysis of equilibrium configurations. *The International Journal of Robotics Research*, 33(1):48–68.

Brignone, L., Raugel, E., Opderbecke, J., Rigaud, V., Piasco, R., and Ragot, S. (2015). First sea trials of HROV the new hybrid vehicle developed by IFREMER. In *OCEANS 2015 - Genova*, pages 1–7. ISSN:.

Brown, T., Stefanini, A., Georgiev, N., Sawoniewicz, J., and Nesnas, I. (2018). Series Elastic Tether Management for Rappelling Rovers. In *IROS: International Conference on Intelligent Robots and Systems*, pages 2893–2900, Madrid, Spain.

Buckham, B. J. (2003). *Dynamics Modelling of Low-Tension Tethers for Submerged Remotely Operated Vehicles*. PhD thesis.

Buckham, B. J., Driscoll, F. R., Radanovic, B., Nahon, M., et al. (2003). Three dimensional dynamics simulation of slack tether motion in an ROV system. In *The Thirteenth International Offshore and Polar Engineering Conference*. International Society of Offshore and Polar Engineers.

Carreras, M., Carrera, A., Palomeras, N., Ribas, D., Hurtós, N., Salvi, Q., and Ridao, P. (2015). Intervention payload for valve turning with an AUV. In *International Conference on Computer Aided Systems Theory*, pages 877–884. Springer.

Chaumette, F. (1994). Visual servoing using image features defined upon geometrical primitives. In *Proceedings of 1994 33rd IEEE Conference on Decision and Control*, volume 4, pages 3782–3787 vol.4. ISSN:.

Chaumette, F. (2004). Image moments: A general and useful set of features for visual servoing. *IEEE Transactions on Robotics*, 20(4):713–723.

Chaumette, F. and Hutchinson, S. (2006). Visual servo control. I. Basic approaches. *IEEE Robotics Automation Magazine*, 13(4):82–90.

Chaumette, F. and Hutchinson, S. (2007). Visual servo control. II. Advanced approaches. *IEEE Robotics Automation Magazine*, 14(1):109–118.

Chaumette, F., Rives, P., and Espiau, B. (1993). Classification and realization of the different vision-based tasks. In *World Scientific Series in Robotics and Intelligent Systems*, volume 7, pages 199–228. WORLD SCIENTIFIC.

Cherubini, A., Chaumette, F., and Oriolo, G. (2008). An image-based visual servoing scheme for following paths with nonholonomic mobile robots. In *2008 10th International Conference on Control, Automation, Robotics and Vision*, pages 108–113.

Childers, B. A., Gifford, D. K., Duncan, R. G., Raum, M. T., Vercellino, M. E., and Froggatt, M. E. (2010). Fiber optic position and shape sensing device and method relating thereto. US Patent 7,781,724.

Christ, R. D. and Wernli Sr, R. L. (2013). *The ROV Manual: A User Guide for Remotely Operated Vehicles*. Butterworth-Heinemann.

Collewet, C. and Chaumette, F. (2002). Positioning a camera with respect to planar objects of unknown shape by coupling 2-D visual servoing and 3-D estimations. *IEEE Transactions on Robotics and Automation*, 18(3):322–333.

Comport, A., Marchand, E., and Chaumette, F. (2004). Robust model-based tracking for robot vision. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, IROS'04*, volume 1, pages 692–697, Sendai, Japan.

Comport, A., Marchand, E., and Chaumette, F. (2007). Kinematic sets for real-time robust articulated object tracking. *Image and Vision Computing, IVC*, 25(3):374–391.

Conte, G., Scaradozzi, D., Mannocchi, D., Raspa, P., Panebianco, L., and Screpanti, L. (2018). Development and Experimental Tests of a ROS Multi-agent Structure for Autonomous Surface Vehicles. *Journal of Intelligent & Robotic Systems*, 92(3-4):705–718.

Creuze, V. (2014). Robots marins et sous-marins. Perception, modélisation, commande. *Techniques de l'Ingenieur*, TIP661WEB:S7783.

Dallej, T., Gouttefarde, M., Andreff, N., Dahmouche, R., and Martinet, P. (2012). Vision-based modeling and control of large-dimension cable-driven parallel robots. In *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference On*, pages 1581–1586. IEEE.

Dallej, T., Gouttefarde, M., Andreff, N., Michelin, M., and Martinet, P. (2011). Towards vision-based control of cable-driven parallel robots. In *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference On*, pages 2855–2860. IEEE.

Driscoll, F. R., Nahon, M., and Lueck, R. G. (2000). A comparison of ship-mounted and cage-mounted passive heave compensation systems. *Journal of Offshore Mechanics and Arctic Engineering*, 122(3):214–221.

Dune, C. (2009). *Localisation et Caractérisation d'objets Inconnus à Partir d'informations Visuelles*. PhD thesis, PhD thesis, Université de Rennes 1, France, 2009. 20.

Echegoyen, Z., Villaverde, I., Moreno, R., Graña, M., and d'Anjou, A. (2010). Linked multi-component mobile robots: Modeling, simulation and control. *Robotics and Autonomous Systems*, 58(12):1292–1305.

Eidsvik, O. A. and Schjølberg, I. (2016). Time domain modeling of rov umbilical using beam equations. *IFAC-PapersOnLine*, 49(23):452–457.

Espiau, B., Chaumette, F., and Rives, P. (1992). A new approach to visual servoing in robotics. *IEEE Transactions on Robotics and Automation*, 8(3):313–326.

Estevez, J. and Graña, M. (2015). Robust control tuning by PSO of aerial robots hose transportation. In *International Work-Conference on the Interplay Between Natural and Artificial Computation*, pages 291–300. Springer.

Estevez, J., Lopez-Guede, J. M., and Graña, M. (2015). Quasi-stationary state transportation of a hose with quadrotors. *Robotics and Autonomous Systems*, 63:187–194.

Feddema, J. T., Lee, C. G., and Mitchell, O. R. (1991). Weighted selection of image features for resolved rate visual feedback control. *IEEE Transactions on Robotics and Automation*, 7(1):31–47.

Feddema, J. T., Lee, C. S. G., and Mitchell, O. R. (1989). Automatic selection of image features for visual servoing of a robot manipulator. In *1989 International Conference on Robotics and Automation Proceedings*, pages 832–837 vol.2.

Feddema, J. T. and Mitchell, O. R. (1989). Vision-guided servoing with feature-based trajectory generation (for robots). *IEEE Transactions on Robotics and Automation*, 5(5):691–700.

Fossen, T. I. (2011). *Handbook of Marine Craft Hydrodynamics and Motion Control*. John Wiley & Sons.

Fossen, T. I., Johansen, T. A., and Perez, T. (2009). A survey of control allocation methods for underwater vehicles. In *Underwater Vehicles*. InTech.

Frank, J. E., Geiger, R., Kraige, D. R., and Murali, A. (2013). Smart tether system for underwater navigation and cable shape measurement. US Patent 8,437,979.

Fukushima, E. F., Kitamura, N., and Hirose, S. (2000). A new flexible component for field robotic system. In *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No.00CH37065)*, volume 3, pages 2583–2588 vol.3.

Gennery, D. B. (1992). Visual tracking of known three-dimensional objects. *International Journal of Computer Vision*, 7(3):243–270.

Gerkey, B. and Conley, K. (2011). Robot Developer Kits [ROS Topics]. *IEEE Robotics Automation Magazine*, 18(3):16–16.

Hashimoto, K. (1993). *Visual Servoing*, volume 7. World scientific.

Hawkes, G. S. and Jeffrey, D. C. (1987). Tether cable management apparatus and method for a remotely-operated underwater vehicle. US Patent 4,686,927.

Howell, C. T. (1992). *Investigation of the Dynamics of Low-Tension Cables*. PhD thesis, Woods Hole Oceanographic Institution MA.

Huber, P. (1981). *Robust Statistics*. Wiley Series in Probability and Statistics. Wiley.

Huntsberger, T. L., Trebi-Ollennu, A., Aghazarian, H., Schenker, P. S., Pirjanian, P., and Nayar, H. D. (2004). Distributed control of multi-robot systems engaged in tightly coupled tasks. *Autonomous Robots*, 17(1):79–92.

Huster, A., Bergstrom, H., Gosior, J., and White, D. (2009). Design and operational performance of a standalone passive heave compensation system for a work class ROV. In *OCEANS 2009, MTS/IEEE Biloxi-Marine Technology for Our Future: Global and Local Challenges*, pages 1–8. IEEE.

Inzartsev, A. and Alexander Pavin (2009). *AUV Application for Inspection of Underwater Communications.* INTECH Open Access Publisher. OCLC: 884025486.

Iqbal, J., Heikkila, S., and Halme, A. (2008). Tether tracking and control of ROSA robotic rover. In *Control, Automation, Robotics and Vision, 2008. ICARCV 2008. 10th International Conference On*, pages 689–693. IEEE.

Irvine, H. M. (1981). *Cable Structures*, volume 17. MIT press Cambridge, MA.

Jaulin, L. (2015). *Mobile Robotics.* Mobile Robotics.

Johnston, E. R., Beer, F., and Eisenberg, E. (2009). *Vector Mechanics for Engineers: Statics and Dynamics.* McGraw-Hill.

Katlein, C., Schiller, M., Belter, H. J., Coppolaro, V., Wenslandt, D., and Nicolaus, M. (2017). A New Remotely Operated Sensor Platform for Interdisciplinary Observations under Sea Ice. *Frontiers in Marine Science*, 4:281.

Kervrann, C. and Heitz, F. (1998). A hierarchical Markov modeling approach for the segmentation and tracking of deformable shapes. *Graphical Models and Image Processing*, 60(3):173–195.

Khatib, O., Yeh, X., Brantner, G., Soe, B., Kim, B., Ganguly, S., Stuart, H., Wang, S., Cutkosky, M., Edsinger, A., et al. (2016). Ocean one: A robotic avatar for oceanic discovery. *IEEE Robotics & Automation Magazine*, 23(4):20–29.

Kiribayashi, S., Yakushigawa, K., and Nagatani, K. (2017). Position estimation of tethered micro unmanned aerial vehicle by observing the slack tether. In *2017 IEEE International Symposium on Safety, Security and Rescue Robotics (SSRR)*, pages 159–165.

Klemas, V. V. (2015). Coastal and environmental remote sensing from unmanned aerial vehicles: An overview. *Journal of Coastal Research*, 31(5):1260–1267.

Koenig, N. and Howard, A. (2004). Design and use paradigms for Gazebo, an open-source multi-robot simulator. In *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (IEEE Cat. No.04CH37566)*, volume 3, pages 2149–2154 vol.3.

Krishna, M., Bares, J., and Mutschler, E. (1997). Tethering system design for Dante II. In *Robotics and Automation, 1997. Proceedings., 1997 IEEE International Conference On*, volume 2, pages 1100–1105. IEEE.

Krupinski, S., Desouche, R., Palomeras, N., Allibert, G., and Hua, M.-D. (2015). Pool testing of AUV visual servoing for autonomous inspection. *IFAC-PapersOnLine*, 48(2):274–280.

Laranjeira, M., Dune, C., and Hugel, V. (2017). Catenary-based visual servoing for tethered robots. In *Robotics and Automation (ICRA), 2017 IEEE International Conference On*, pages 732–738. IEEE.

Lee, T. (2015). Geometric controls for a tethered quadrotor UAV. In *Decision and Control (CDC), 2015 IEEE 54th Annual Conference On*, pages 2749–2754. IEEE.

L'Hour, M. and Creuze, V. (2016). French Archaeology's Long March to the Deep - The Lune Project: Building the Underwater Archaeology of the Future. In *Experimental Robotics*, pages 911–927. Springer.

Lowe, D. G. (1992). Robust model-based motion tracking through the integration of search and estimation. *International Journal of Computer Vision*, 8(2):113–122.

Ma, Y., Soatto, S., Kosecka, J., and Sastry, S. S. (2012). *An Invitation to 3-d Vision: From Images to Geometric Models*, volume 26. Springer Science & Business Media.

Mallapragada, V., Sarkar, N., and Podder, T. K. (2011). Toward a robot-assisted breast intervention system. *IEEE/ASME Transactions on Mechatronics*, 16(6):1011–1020.

Manhães, M. M. M., Scherer, S. A., Voss, M., Douat, L. R., and Rauschenbach, T. (2016). UUV Simulator: A Gazebo-based package for underwater intervention and multi-robot simulation. In *OCEANS 2016 MTS/IEEE Monterey*. IEEE.

Mansard, N. and Chaumette, F. (2007). Task Sequencing for High-Level Sensor-Based Control. *IEEE Transactions on Robotics*, 23(1):60–72.

Marani, G., Choi, S. K., and Yuh, J. (2009). Underwater autonomous manipulation for intervention missions AUVs. *Ocean Engineering*, 36(1):15–23.

Marchand, E., Bouthemy, P., Chaumette, F., and Moreau, V. (1999). Robust real-time visual tracking using a 2D-3D model-based approach. In *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference On*, volume 1, pages 262–268. IEEE.

Marchand, E. and Chaumette, F. (2005). Feature tracking for visual servoing purposes. *Robotics and Autonomous Systems*, 52(1):53–70.

Marchand, E., Spindler, F., and Chaumette, F. (2005). ViSP for visual servoing: A generic software platform with a wide class of robot control skills. *IEEE Robotics and Automation Magazine*, 12(4):40–52.

McGarey, P., MacTavish, K., Pomerleau, F., and Barfoot, T. D. (2016a). The line leading the blind: Towards nonvisual localization and mapping for tethered mobile robots. In *Robotics and Automation (ICRA), 2016 IEEE International Conference On*, pages 4799–4806. IEEE.

McGarey, P., MacTavish, K., Pomerleau, F., and Barfoot, T. D. (2017). TSLAM: Tethered simultaneous localization and mapping for mobile robots. *The International Journal of Robotics Research*, 36(12):1363–1386.

McGarey, P., Pomerleau, F., and Barfoot, T. D. (2016b). System design of a tethered robotic explorer (TReX) for 3D mapping of steep terrain and harsh environments. In *Field and Service Robotics*, pages 267–281. Springer.

McKerrow, P. J. and Ratner, D. (2007). The design of a tethered aerial robot. In *Proceedings 2007 IEEE International Conference on Robotics and Automation*, pages 355–360.

Merlet, J.-P. (2018a). Computing Cross-Sections of the Workspace of a Cable-Driven Parallel Robot with 6 Sagging Cables Having Limited Lengths. In *International Symposium on Advances in Robot Kinematics*, pages 392–400. Springer.

Merlet, J.-P. (2018b). An Experimental Investigation of Extra Measurements for Solving the Direct Kinematics of Cable-Driven Parallel Robots. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6947–6952.

Merlet, J.-P. (2018c). Some properties of the Irvine cable model and their use for the kinematic analysis of cable-driven parallel robots. In *European Conference on Mechanism Science*, pages 409–416. Springer.

Milutinović, M., Kranǰcević, N., and Deur, J. (2014). Multi-mass dynamic model of a variable-length tether used in a high altitude wind energy system. *Energy conversion and management*, 87:1141–1150.

Molchan, M. (2005). The role of micro-rovs in maritime safety and security. *Molchan Marine Sciences, USA*.

Murtra, A. C. and Tur, J. M. M. (2013). IMU and cable encoder data fusion for in-pipe mobile robot localization. In *TePRA*, pages 1–6.

Navarro-Alarcon, D. and Liu, Y. (2018). Fourier-Based Shape Servoing: A New Feedback Method to Actively Deform Soft Objects into Desired 2-D Image Contours. *IEEE Transactions on Robotics*, 34(1):272–279.

Navarro-Alarcon, D., Liu, Y., Romero, J. G., and Li, P. (2013a). Visually servoed deformation control by robot manipulators. In *2013 IEEE International Conference on Robotics and Automation*, pages 5259–5264.

Navarro-Alarcon, D., Liu, Y.-H., Romero, J. G., and Li, P. (2013b). Model-free visually servoed deformation control of elastic objects by robot manipulators. *IEEE Transactions on Robotics*, 29(6):1457–1468.

Navarro-Alarcon, D., Liu, Y.-h., Romero, J. G., and Li, P. (2014). On the visual deformation servoing of compliant objects: Uncalibrated control methods and experiments , On the visual deformation servoing of compliant objects: Uncalibrated

control methods and experiments. *The International Journal of Robotics Research*, 33(11):1462–1480.

Nayar, K. G., Sharqawy, M. H., Banchik, L. D., and V, J. H. L. (2016). Thermophysical properties of seawater: A review and new correlations that include pressure dependence. *Desalination*, 390:1–24.

Nickels, K. and Hutchinson, S. (2001). Model-based tracking of complex articulated objects. *IEEE Transactions on Robotics and Automation*, 17(1):28–36.

Nicotra, M. M., Naldi, R., and Garone, E. (2017). Nonlinear control of a tethered UAV: The taut cable case. *Automatica*, 78:174–184.

Papanikolopoulos, N. P., Khosla, P. K., and Kanade, T. (1993). Visual tracking of a moving target by a camera mounted on a robot: A combination of control and vision. *IEEE transactions on robotics and automation*, 9(1):14–35.

Park, J.-Y., Jun, B.-h., Lee, P.-m., and Oh, J. (2009). Experiments on vision guided docking of an autonomous underwater vehicle using one camera. *Ocean Engineering*, 36(1):48–61.

Perrin, D. P., Kwon, A., and Howe, R. D. (2004). A novel actuated tether design for rescue robots using hydraulic transients. In *Robotics and Automation, 2004. Proceedings. ICRA'04. 2004 IEEE International Conference On*, volume 4, pages 3482–3487. IEEE.

Pettersson, A., Ohlsson, T., Davis, S., Gray, J., and Dodd, T. (2011). A hygienically designed force gripper for flexible handling of variable and easily damaged natural food products. *Innovative Food Science & Emerging Technologies*, 12(3):344–351.

Prabhakar, S. and Buckham, B. (2005). Dynamics modeling and control of a variable length remotely operated vehicle tether. In *OCEANS, 2005. Proceedings of MTS/IEEE*, pages 1255–1262. IEEE.

Prats, M., Palomeras, N., Ridao, P., and Sanz, P. J. (2012). Template Tracking and Visual Servoing for Alignment Tasks with Autonomous Underwater Vehicles. *IFAC Proceedings Volumes*, 45(27):256–261.

Pratt, K. S., Murphy, R. R., Burke, J. L., Craighead, J., Griffin, C., and Stover, S. (2008). Use of Tethered Small Unmanned Aerial System at Berkman Plaza II Collapse. In *2008 IEEE International Workshop on Safety, Security and Rescue Robotics*, pages 134–139.

Pressigout, M. (2006). *Approches Hybrides Pour Le Suivi Temps-Réel d'objets Complexes Dans Des Séquences Vidéos*. PhD thesis, Rennes 1.

Rajan, V. A. K. T., Nagendran, A., Dehghani-Sanij, A., and Richardson, R. C. (2016). Tether monitoring for entanglement detection, disentanglement and localisation of autonomous robots. *Robotica*, 34(3):527–548.

Reverte, C. F., Thayer, S. M., Whittaker, W., Close, E. C., Slifko, A., Hudson, E., and Vallapuzha, S. (2011). Autonomous inspector mobile platform. US Patent 8,024,066.

Ropars, B., Lasbouygues, A., Lapierre, L., and Andreu, D. (2015). Thruster's dead-zones compensation for the actuation system of an underwater vehicle. In *2015 European Control Conference (ECC)*, pages 741–746, Linz, Austria. IEEE.

Sahin, T. and Unel, M. (2004). Globally stabilized 3L curve fitting. In *International Conference Image Analysis and Recognition*, pages 495–502. Springer.

Salgado-Jimenez, T., Gonzalez-Lopez, J. L., Martinez-Soto, L. F., Olguin-Lopez, E., Resendiz-Gonzalez, P. A., and Bandala-Sanchez, M. (2010). Deep water ROV design for the Mexican oil industry. In *OCEANS'10 IEEE SYDNEY*, pages 1–6, Sydney, Australia. IEEE.

Shimono, S., Matsubara, O., Toyama, S., Nishizawa, U., Kato, S., and Arisumi, H. (2015). Development of underwater inspection system for dam inspection. In *OCEANS 2015 - MTS/IEEE Washington*, pages 1–6, Washington, DC. IEEE.

Shirai, Y. and Inoue, H. (1973). Guiding a robot by visual feedback in assembling tasks. *Pattern recognition*, 5(2):99–106.

Short, A. D. and Woodroffe, C. D. (2009). *The Coast of Australia*. Cambridge University Press.

Siciliano, B. and Khatib, O. (2016). *Springer Handbook of Robotics*. Springer.

Smolowitz, R. J., Patel, S. H., Haas, H. L., and Miller, S. A. (2015). Using a remotely operated vehicle (ROV) to observe loggerhead sea turtle (Caretta caretta) behavior on foraging grounds off the mid-Atlantic United States. *Journal of Experimental Marine Biology and Ecology*, 471:84–91.

Tahri, O. and Chaumette, F. (2003). Application of moment invariants to visual servoing. In *International Conference on Robotics and Automation*, volume 3, pages 4276–4281.

Talke, K. A., de Oliveira, M., and Bewley, T. (2018). Catenary Tether Shape Analysis for a UAV - USV Team. In *IROS: International Conference on Intelligent Robots and Systems*, pages 7803–7809, Madrid, Spain.

Tognon, M. and Franchi, A. (2015). Control of motion and internal stresses for a chain of two underactuated aerial robots. In *Control Conference (ECC), 2015 European*, pages 1620–1625. IEEE.

Torabi, M., Hauser, K., Alterovitz, R., Duindam, V., and Goldberg, K. (2009). Guiding medical needles using single-point tissue manipulation. In *Robotics and Automation, 2009. ICRA'09. IEEE International Conference On*, pages 2705–2710. IEEE.

Triantafyllou, M. S. and Grosenbaugh, M. A. (1991). Robust control for underwater vehicle systems with time delays. *IEEE Journal of Oceanic Engineering*, 16(1):146–151.

Tsai, D., Nesnas, I. A., and Zarzhitsky, D. (2013). Autonomous vision-based tethered-assisted rover docking. In *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference On*, pages 2834–2841. IEEE.

Tsai, R. (1987). A versatile camera calibration technique for high-accuracy 3D machine vision metrology using off-the-shelf TV cameras and lenses. *IEEE Journal on Robotics and Automation*, 3(4):323–344.

van der Zwaan, S., Bernardino, A., and Santos-Victor, J. (2002). Visual station keeping for floating robots in unstructured environments. *Robotics and Autonomous Systems*, 39(3–4):145–155. Intelligent Robotic Systems - SIRS'2000.

Vishnu, T., Kumar, A., and Richardson, R. (2008). Tether monitoring techniques for environment monitoring, tether following and localization of autonomous mobile robots. In *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference On*, pages 2109–2114.

Wakamatsu, H., Arai, E., and Hirai, S. (2006). Knotting/unknotting manipulation of deformable linear objects. *The International Journal of Robotics Research*, 25(4):371–395.

Wang, X. and Bhattacharya, S. (2018). A Topological Approach to Workspace and Motion Planning for a Cable-Controlled Robot in Cluttered Environments. *IEEE Robotics and Automation Letters*, 3(3):2600–2607.

Wang, Z. and Hirai, S. (2011). Modeling and estimation of rheological properties of food products for manufacturing simulations. *Journal of food engineering*, 102(2):136–144.

Weiss, L., Sanderson, A., and Neuman, C. (1987). Dynamic sensor-based control of robots with visual feedback. *IEEE Journal on Robotics and Automation*, 3(5):404–417.

Wernli, R. L. and Christ, R. D. (2009). Observation Class ROVs Come of Age. In *Sixth International Symposium on Underwater Technology Wuxi, China, April*.

Wynn, R. B., Huvenne, V. A., Le Bas, T. P., Murton, B. J., Connelly, D. P., Bett, B. J., Ruhl, H. A., Morris, K. J., Peakall, J., Parsons, D. R., Sumner, E. J., Darby, S. E., Dorrell, R. M., and Hunt, J. E. (2014). Autonomous Underwater Vehicles (AUVs): Their past, present and future contributions to the advancement of marine geoscience. *Marine Geology*, 352:451–468.

Yang, W., Zhang, Z., and Zhang, A. (2008). Research on an active heave compensation system for remotely operated vehicle. In *Intelligent Computation Technology and Automation (ICICTA), 2008 International Conference On*, volume 2, pages 407–410. IEEE.

Yazicioglu, A. Y., Calli, B., and Unel, M. (2009). Image based visual servoing using algebraic curves applied to shape alignment. In *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5444–5449.

Zhu, J., Navarro, B., Fraisse, P., Crosnier, A., and Cherubini, A. I. (2018). Dual-arm robotic manipulation of flexible cables. In *IROS: International Conference on Intelligent Robots and Systems*, Madrid, Spain.

Zikou, L., Papachristos, C., and Tzes, A. (2015). The power-over-tether system for powering small UAVs: Tethering-line tension control synthesis. In *Control and Automation (MED), 2015 23th Mediterranean Conference On*, pages 681–687. IEEE.

List of Figures

Résumé

Cette thèse porte sur le problème du contrôle de la forme d'ombilicaux pour des robots sous-marins légers téléopérés (mini-ROVs), qui conviennent, grâce à leur petite taille et grande manoeuvrabilité, à l'exploration des eaux peu profondes et des espaces encombrés. La régulation de la forme de l'ombilical est cependant une tâche difficile, car ces robots n'ont pas une puissance de propulsion suffisante pour contrebalancer les forces de traînée du câble. Pour faire face à ce problème, nous avons introduit le concept de Cordée de mini-ROVs, dans lequel plusieurs robots sont reliés à l'ombilical et peuvent, ensemble, contrebalancer les perturbations extérieures et contrôler la forme du câble.

Nous avons étudié l'utilisation des caméras embarquées pour réguler la forme d'une portion de l'ombilical reliant deux robots successifs, un leader et un suiveur. Seul le robot suiveur se chargera de la tâche de régulation de la forme du câble. Le leader est libéré pour explorer ses alentours. L'ombilical est supposé être légèrement pesant et donc modélisé par une chaînette. Les paramètres de forme du câble sont estimés en temps réel par une procédure d'optimisation non-linéaire qui adapte le modèle de chaînette aux points détectés dans les images des caméras. La régulation des paramètres de forme est obtenue grâce à une commande reliant le mouvement du robot à la variation de la forme de l'ombilical. L'asservissement visuel proposé s'est avéré capable de contrôler correctement la forme du câble en simulations et expériences réalisées en basin.

Abstract

This thesis addresses the problem of tether shape control for small remotely operated underwater vehicles (mini-ROVs), which are suitable, thanks to their small size and high maneuverability, for the exploration of shallow waters and cluttered spaces. The management of the tether is, however, a hard task, since these robots do not have enough propulsion power to counterbalance the drag forces acting on the tether cable. In order to cope with this problem, we introduced the concept of a Chain of mini-ROVs, where several robots are linked to the tether cable and can, together, manage the external perturbations and control the shape of the cable.

We investigated the use of the embedded cameras to regulate the shape of a portion of tether linking two successive robots, a leader and a follower. Only the follower robot deals with the tether shape regulation task. The leader is released to explore its surroundings. The tether linking both robots is assumed to be negatively buoyant and is modeled by a catenary. The tether shape parameters are estimated in real-time by a nonlinear optimization procedure that fits the catenary model to the tether detected points in the image. The shape parameter regulation is thus achieved through a catenary-based control scheme relating the robot motion with the tether shape variation. The proposed visual servoing control scheme has proved to properly manage the tether shape in simulations and real experiments in pool.