Shape visual servoing of a suspended cable

Lev Smolentsev

COLLEGE MATHS, TELECOMS
DOCTORAL INFORMATIQUE, SIGNAL
BRETAGNE SYSTEMES, ELECTRONIQUE

Université de Rennes

THÈSE DE DOCTORAT DE

L'UNIVERSITÉ DE RENNES

ÉCOLE DOCTORALE N° 601
Mathématiques, Télécommunications, Informatique, Signal, Systèmes, Électronique
Spécialité : *Automatique, Productique et Robotique*

Par

Lev SMOLENTSEV

Shape visual servoing of a suspended cable

Thèse présentée et soutenue à Rennes, le 18 mars 2024
Unité de recherche : Centre Inria de l'Université de Rennes

Rapporteurs avant soutenance :

Antonio FRANCHI — Full Professor, University of Twente
Youcef MEZOUAR — Professeur des Universités, Université Clermont Auvergne

Composition du Jury :

Président :	Jacques GANGLOFF	Professeur des Universités, Université de Strasbourg
Examinateurs :	Antonio FRANCHI	Full Professor, University of Twente
	Youcef MEZOUAR	Professeur des Universités, Université Clermont Auvergne
	Isabelle FANTONI	Directrice de recherche CNRS, Laboratoire des Sciences du Numérique de Nantes
Dir. de thèse :	Alexandre KRUPA	Directeur de recherche Inria, Centre Inria de l'Université de Rennes
Co-encad. de thèse :	François CHAUMETTE	Directeur de recherche Inria, Centre Inria de l'Université de Rennes

Acknowledgement

To Palina, Svetlana and dear parents. No words can express how grateful I am for your support.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

With the progress of science and technology, the human being has advanced in the creation of machines capable of significantly enhancing or replacing his labour and expertise in various applications such as the production of goods, space exploration, medical diagnosis, and agriculture, to name a few. All such applications can be divided into a sequence of smaller procedures, also called tasks. Performing complex tasks in uncontrolled environments requires machines to behave in a similar way to humans. For this reason, the field of robotics has emerged to provide machines with autonomy, enabling them to adapt to dynamically changing environments. An example of a complex task is to take a soft ball (see Figure 1.1). For a human to perform this task, he has to perceive the ball with its eyes in order to localise its position on the table, then make the decision or plan how his hand will move and grasp it, and finally execute a sequence of body motions (or actions) to accomplish the task. It is worth noting that this task can be completed without making any decision by reasoning only on the perception of the current position of the hand with respect to the ball. Note also that in Figure 1.1 the target ball is made of foam and can be considered as a soft body. This means that it can change its shape under external stress, can be easily lifted with one arm, and will not be damaged by a firm hand grip. A human knows how to grasp this ball even without any knowledge about its material, size, weight and many other properties of this object. Unfortunately, this skill is not available to the robot for a large number of reasons.

As already mentioned, the task illustrated in Figure 1.1 is very primitive for a human but relatively complex for a robot due to a number of its limitations, such as:

— the lack of human eyes and his sense of touch to efficiently perceive the ball and the environment

— the lack of human awareness to reason about the actual accomplishment of the task

— the lack of human motor skills to perform an efficient action

That is why researchers in the field of robotics try to create tools and algorithms that

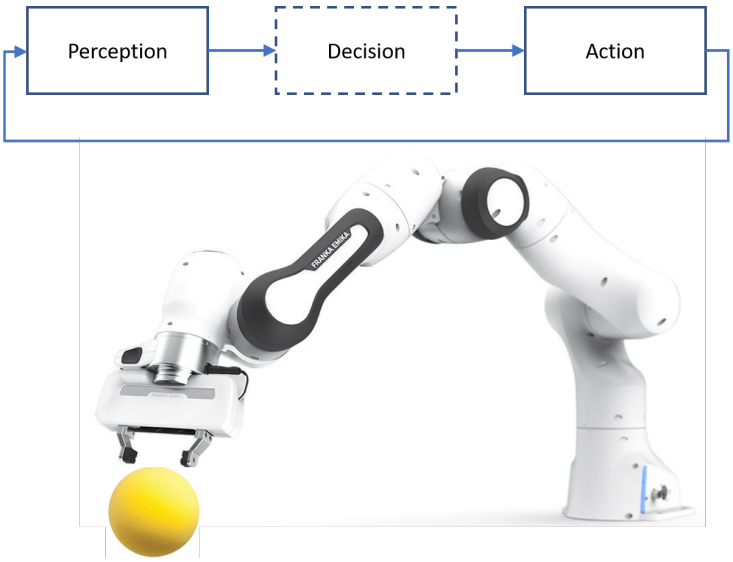


Figure 1.1 – On the top: a decomposition of any robotic task, the Decision block is not mandatory for a large number of tasks involving vision as a main perception tool. On the bottom: an illustration of a soft ball grasping task that is simple for an ordinary human but still complex to be performed by an advanced robot.

will help robots to approach the human-level performance. An RGB-D camera, a force sensor, a rapidly exploring random tree planning algorithm, and a visual servoing control algorithm are very few illustrative examples of such tools. Despite all mentioned limitations, robots can be designed to perform a specific task in the best possible way, even surpassing human capabilities. Examples of such specific task include lifting heavy objects, making long-range transportation, or rapidly exploring an unknown environment, as shown in Figure 1.2. Moreover, robots have greater endurance and power for performing fast repetitive tasks, and advances in robotics have made it possible to solve a large number of industrial tasks, although there is still a lot of research to be done to make them even more autonomous.

1.1 Context

1.1.1 Manipulation of deformable objects

Robotic interaction with soft objects is of particular interest to our research, as it involves an immense number of objects in industry and everyday life that could be efficiently manipulated with robots. In particular, industrial needs pose new challenges for

Figure 1.2 – Left image: example of a fully autonomous aerial robot equipped only with on-board sensors and navigating in an unknown environment. From [Loquercio et al., 2021]. Right image: self-driving truck navigating on the road to deliver freight.

the robotic manipulation of fabrics, ropes, clothes, cables, plants, and many other types of deformable objects. Illustrative examples of cloth and cable shaping tasks performed by robots are presented in Figure 1.3. However, the robotic manipulation of soft objects remains an open problem, as existing shape control strategies are only valid for a given type of soft object, which restricts them to an intended application. Consequently, providing a shape control solution that can generalise to all types of objects such as those mentioned above remains a major challenge in the robotics community. In this thesis, our first objective is to propose an efficient methodology for autonomously shaping a tether cable attached to different types of robot. In the past, this task has been studied for serial manipulators, and our aim is to contribute to these studies by proposing a new method that overcomes some of the limitations of previous works on this topic. Our second objective is to use this cable for aerial manipulation by controlling its shape under the action of drones that move its extremities. The envisaged application is to pick up an object with the cable and transport it to a new location. In the next section, we will introduce the reader to the aerial manipulation problem in robotics.

1.1.2 Aerial manipulation

A special area of research is aerial robotics. Flying robot can execute motion in the 3D environment surpassing gravity. There are various types of aerial robots or UAVs (Unmanned Aerial Vehicles) and especially VTOL (Vertical Take-Off and Landing) drones

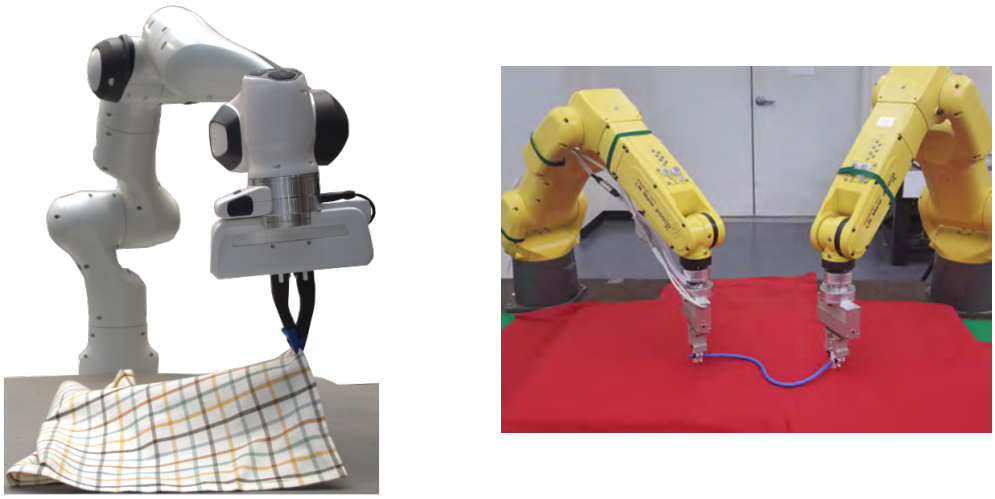
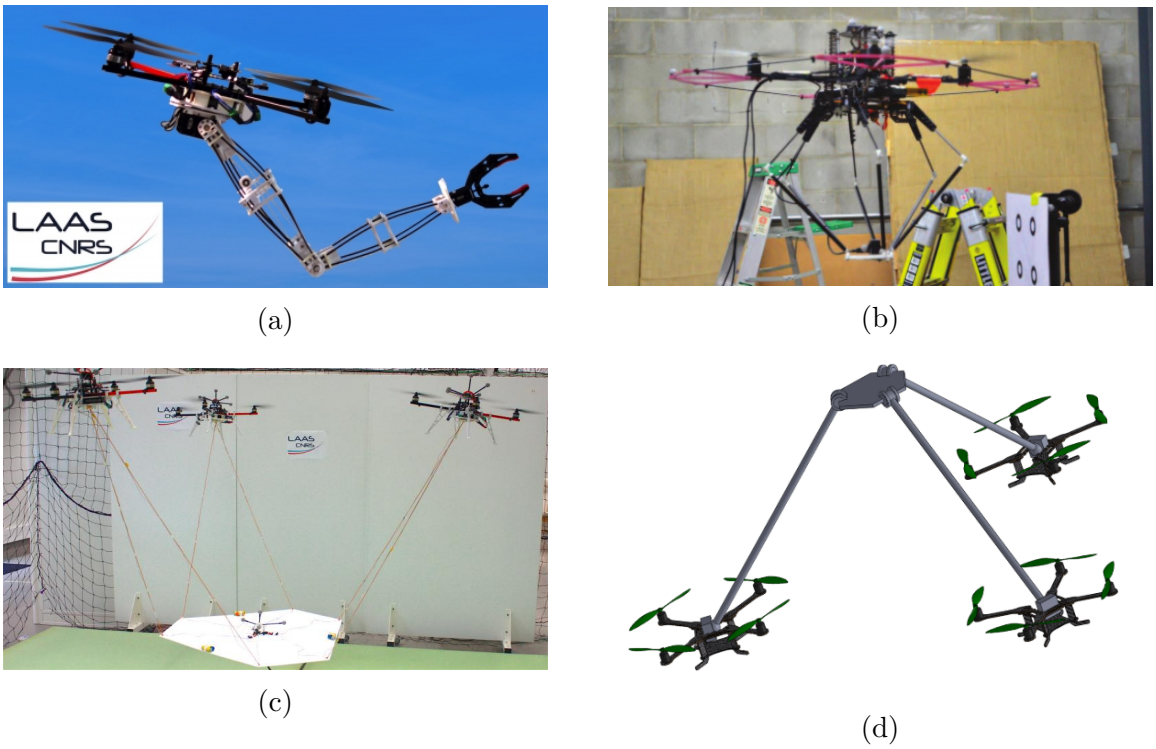


Figure 1.3 – (a) A serial manipulator performing the task of cloth unfolding. From [Blanco-Mulero et al., 2023]. (b) Cable shaping by two serial manipulators. From [Jin et al., 2019].

such as multirotor drones that attracted our attention. A particular interest is to use them for aerial manipulation (which is very similar to our example of lifting a ball) and transportation tasks, as the target destination can be achieved faster and there are generally less environmental constraints in the air at high altitude than on the ground. However, the gripping capacity of one drone remains limited to a metal clip that is not adaptable to different shapes of objects and can damage the drone during faulty grip due to rigid contact with the drone. To overcome this issue, several researchers have proposed to design a drone combined with other types of robotic grippers, a serial manipulator arm, or a parallel delta robot, as can be seen in Figure 1.4(a),(b). However, aerial robots present a very limited payload capacity due to its power consumption limited by an onboard source of power. That is why the multi-drone systems such as those shown in Figure 1.4(c),(d) have rapidly emerged. These types of cooperative drones are attached to a common object through additional flexible or rigid bodies. This object can be a magnetic platform for transporting magnetised construction pieces as proposed in [Jiménez-Cano et al., 2022]. In this thesis we are interested in the use of an additional body which is flexible, and more specifically in a cable for gripping an object in order to transport it. As shown in Figure 1.4(c), flexible bodies have a significant advantage over rigid bodies in terms of weight and compliance with external disturbances such as contact with the environment. In this thesis work, we contribute to the area of aerial manipulation by proposing to use a cable as a common object attached to a pair of drones and show that such a system can efficiently perform an aerial grasping and transportation task. In addition, we show that



(a) (b)

(c) (d)

Figure 1.4 – (a) A system composed of a quadrotor with a lightweight two-joint arm attached. (b) A delta robot attached to a quadrotor drone from [Danko et al., 2015]. (c) A flying Crane from [Sanalitro et al., 2022]. (d) A flying robot composed of three quadrotors linked by a rigid articulated architecture composed of three rigid legs and a platform from [Six et al., 2018].

a cable attached to drones can be efficiently shaped by their motion using an on-board visual sensor.

1.2 Objectives of the thesis

The main objective of this thesis is to elaborate a new control framework for shaping a suspended tether cable using visual servoing. To achieve this goal the following steps are proposed:

— Derivation of a model to represent a tether cable attached to robots.
— Determination of a relevant set of visual features and its related interaction matrix for performing an automatic cable shaping control task that consists in manipulating the extremities of the cable in such a way it reaches a desired shape.
— Elaboration of a method for tracking in real-time the cable features from RGB-D vision and other available sensors.
— Design of a visual control law that is applied to the robots to perform the cable shaping task.
— Validation of the proposed methodology from numerous simulations and experiments involving either a manipulator or multirotor drones.
— Discussion and analysis of the achieved results.

1.3 Contributions of the thesis

This thesis contributes to the state-of-the-art in the manipulation of deformable linear objects using serial manipulators or quadrotor drones. In particular, it proposes:

— A cable shape control method adaptable for a large number of existing robots presented in **Chapter 3** and which was published in [Smolentsev et al., 2023].
— A method for controlling the shape of a cable suspended between two drones using on-board sensing of the cable that is presented in **Chapter 4**.
— A method for manipulating a cable using two quadrotor drones for an efficient aerial grasping and transportation of rigid objects presented in **Chapter 4**.
— Numerous simulation and experimental results that validate the cable shape control and object manipulation tasks presented in **Chapter 4**.

1.4 Outline

The thesis document is organised as follows:

Chapter 2 introduces the basics of visual servoing, reviews deformable linear object modelling and shape control techniques together with aerial manipulation techniques.

Chapter 3 presents the methodological contribution of this thesis that concerns the shape visual servoing of a tether cable from parabolic features that is then validated from simulation and experimental results obtained with a robotic arm.

Chapter 4 presents our work on visual servoing of the shape of a tether cable using drones with simulations and experimental results.

Chapter 5 concludes the thesis and suggests future research perspectives.

Chapter 2

State of the art

This thesis deals with the manipulation of deformable linear objects. In this chapter, we begin by recalling the basics of visual servoing, which have been defined for interacting with rigid objects. In this first section, a number of important notions will be recalled to help the reader understand how perception and action can be linked to perform simple robotic tasks. Afterwards, we will briefly introduce the theory about the representation of deformable objects, which is a key component in understanding the physical behaviour of such objects. Subsequently, an overview of the methods available in the literature for controlling the shape of deformable objects, such as cables and planar objects, will be introduced. In that section, the point is made up of how the robot interacts with the soft object in order to apply an intentional and measurable deformation on it. Finally, a review of cable manipulation approaches using flying drones will be presented. In addition, a brief overview of aerial applications for deformable objects will be provided and discussed. The conclusion of the chapter will present the position of this thesis with regard to the state-of-the-art.

2.1 Visual servoing

Visual servoing refers to the use of vision data to control the motion of a robot. Vision data may be acquired from a camera mounted directly on a robot manipulator or a mobile robot, or remaining in a stationary configuration observing the robot but not moving with it. The first case is called *eye-in-hand*, and the latter case *eye-to-hand*. This section focuses on the case of *eye-in-hand* if the other configuration is not explicitly mentioned.

2.1.1 Basic concepts

As explained in [Chaumette, 2007], a visual servoing control scheme aims to minimise some error $\mathbf{e}(\mathbf{t})$ defined by:

$$\mathbf{e}(\mathbf{t}) = \mathbf{s}(\mathbf{m}(t), \mathbf{a}) - \mathbf{s}^* \tag{2.1}$$

The vector $\mathbf{s} \in \mathbb{R}^n$ is a vector of n visual features, $\mathbf{m}(t)$ being a set of image measurements such as interest points coordinates on the image or the coefficients of a conic section passing through that points, and $\mathbf{a}$ is a set of parameters of additional knowledge about the system such as camera intrinsic parameters or the 3D model of the observed object. The vector $\mathbf{s}^*$ represents the desired values (or targets) of the visual features to be achieved automatically by the robotic system.

The relation between the variation of the visual features $\dot{\mathbf{s}}$ and the velocity of the camera $\mathbf{v}_c = (\boldsymbol{v}_c, \boldsymbol{\omega}_c) \in \mathbb{R}^6$ is as follows:

$$\dot{\mathbf{s}} = \mathbf{L}_s \mathbf{v}_c \tag{2.2}$$

with $\mathbf{L_s} \in \mathbb{R}^{n \times 6}$ being the interaction matrix (or feature Jacobian) related to $\mathbf{s}$. To try to ensure an exponential decoupled decrease, the dynamics of $\mathbf{e}$ is chosen as follows: $\dot{\mathbf{e}} = -\lambda \mathbf{e}$, where λ is a positive gain. This choice leads to the control law:

$$\mathbf{v}_c = -\lambda \widehat{\mathbf{L_s}}^{+} (\mathbf{s} - \mathbf{s}^*) = -\lambda \widehat{\mathbf{L_e}}^{+} \mathbf{e} \tag{2.3}$$

where $\widehat{\mathbf{L_s}}^{+}$ is the approximate Moore-Penrose pseudo-inverse of the true value of $\mathbf{L_s}$. Note that in practice an approximation of $\mathbf{L_s}$ is used, denoted $\widehat{\mathbf{L_s}}$. Finally, by closing the visual servoing loop and assuming that the robot is capable of perfect execution of the velocity $\mathbf{v}_c$, we obtain:

$$\dot{\mathbf{e}} = -\lambda \mathbf{L_s} \widehat{\mathbf{L_s}}^{+} \mathbf{e} \tag{2.4}$$

which is different from the ideal error dynamics (*i.e.* $\dot{\mathbf{e}} = -\lambda \mathbf{e}$) if the approximations involved into $\widehat{\mathbf{L_s}}$ are coarse (*e.g.* $\mathbf{L_s} \widehat{\mathbf{L_s}}^{+} \neq \mathbf{I}_n$). Equation (2.4) is the basis for the Lyapunov stability analysis of the control law (2.3). Note that (2.3) should be transformed to express the camera velocity in the robot end effector frame:

$$\mathbf{v}_e = -\lambda {}^e\mathbf{V}_c \widehat{\mathbf{L_s}}^{+} (\mathbf{s} - \mathbf{s}^*) \tag{2.5}$$

where ${}^e\mathbf{V}_c$ is the velocity twist transformation matrix from camera frame into end-effector

frame and is given for in *eye-to-hand* case by:

$$ {}^{e}\mathbf{V}_c = \begin{bmatrix} {}^{e}\mathbf{R}_c & [{}^{e}\mathbf{t}_c]_\times \, {}^{e}\mathbf{R}_c \\ \mathbf{0}_{3\times 3} & {}^{e}\mathbf{R}_c \end{bmatrix} \tag{2.6} $$

with ${}^{e}\mathbf{R}_c$ being the rotation matrix from the camera frame to the end-effector frame and $[{}^{e}\mathbf{t}_c]_\times$ being the skew-symmetric matrix of the translation vector ${}^{e}\mathbf{t}_c$ from the camera frame to the end-effector frame. For a 6 DOF robot manipulator, the resulting joint control velocity is then given by:

$$ \dot{\mathbf{q}} = -\lambda \mathbf{J}_{\mathbf{q}}^{-1} {}^{e}\mathbf{V}_c \widehat{\mathbf{L}_{\mathbf{s}}}^{+} (\mathbf{s} - \mathbf{s}^*) \tag{2.7} $$

where $\dot{\mathbf{q}} = \mathbf{J}_{\mathbf{q}}^{-1}\mathbf{v}_e$ and $\mathbf{J}_{\mathbf{q}}^{-1}$ is the inverse of the robot Jacobian expressed in the robot end-effector frame. Note that for the eye-to-hand configuration, (2.5) and (2.7) change sign according to the relative motion of the visual features in the image, as illustrated in Figure 2.1. Moreover, ${}^{e}\mathbf{V}_c$ needs to be computed at every iteration of the control loop in the eye-to-hand case. Robot Jacobian and classic hand-eye calibration techniques can be used for that (*e.g.* [L.-W. Tsai, 1999]). It is important to mention that (2.5) and (2.7)

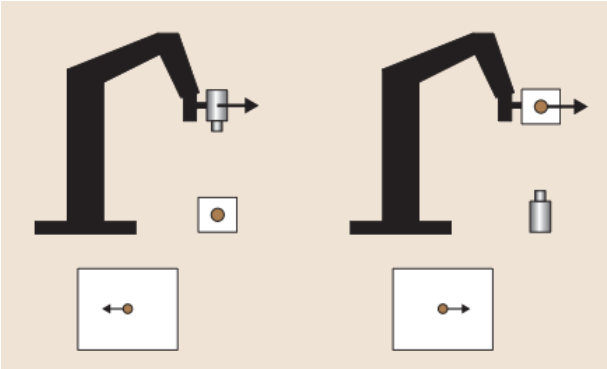


Figure 2.1 – Left image: eye-in-hand case, right image: eye-to-hand case. Relative motion of the visual features in the camera image produced by the same robot motion (bottom). From [Chaumette, 2007].

result in identical robot trajectories given that the same pseudo-inverse of $\widehat{\mathbf{L}_{\mathbf{s}}}^{+}$ is used and by assuming that the robot Jacobian is never singular during the visual servoing. In particular, control schemes similar to (2.3),(2.5) will be used in our work to apply either the control velocity to the centre of gravity (COG) of a quadrotor drone or to the end-effector of a robotic manipulator.

The interaction matrix $\mathbf{L_s}$ and visual features $\mathbf{s}$ can be defined following two main schemes of visual servoing:

— Image-Based Visual Servoing (IBVS), where $\mathbf{s}$ is defined from the camera intrinsic parameters *i.e.* $\mathbf{a}$ and image measurements $\mathbf{m}$ (*e.g.* pixel coordinates, moments, etc.)

— Pose-Based Visual Servoing (PBVS), where $\mathbf{s}$ is defined from the intrinsic parameters $\mathbf{a}$ of the camera, the 3-D model of the observed object, and image measurements $\mathbf{m}$, all being used to estimate the pose of the object with respect to the camera frame.

Image-Based Visual Servoing

An example of the interaction matrix derivation is detailed hereafter for the case where the visual feature corresponds to one point. To project a point with its coordinates expressed in the camera frame ${}^c\mathbf{p} = (X, Y, Z)$ into the image plane point ${}^i\mathbf{p} = (x, y)$ a simple pinhole camera model can be used:

$$\begin{aligned} x &= X/Z = \frac{u - c_u}{p_u} \\ y &= Y/Z = \frac{v - c_v}{p_v} \end{aligned} \tag{2.8}$$

where $\mathbf{m} = (u, v)$ represent horizontal and vertical pixel coordinates and $\mathbf{a} = (c_u, c_v, p_u, p_v)$ correspond to the camera intrinsic parameters: c_u, c_v are the coordinates of the principal point, p_u, p_v are the ratio between the focal length and the size of a pixel. Taking the time derivative of (2.8) gives:

$$\begin{aligned} \dot{x} &= (\dot{X} - x\dot{Z})/Z \\ \dot{y} &= (\dot{Y} - y\dot{Z})/Z \end{aligned} \tag{2.9}$$

Then, the motion of the point ${}^c\mathbf{p}$ in the camera frame is given by ${}^c\dot{\mathbf{p}} = -\boldsymbol{v}_c - \boldsymbol{\omega}_c \times {}^{\mathbf{c}}\mathbf{p}$. By combining it with (2.9) the expression of the interaction matrix for ${}^i\dot{\mathbf{p}} = \mathbf{L}_{{}^i\mathbf{p}}\mathbf{v}_c$ is obtained as:

$$\mathbf{L}_{{}^i\mathbf{p}} = \begin{bmatrix} -1/Z & 0 & x/Z & xy & -(1+x^2) & y \\ 0 & -1/Z & y/Z & 1+y^2 & -xy & -x \end{bmatrix} \tag{2.10}$$

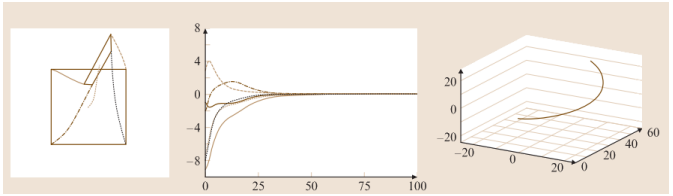
In the matrix $\mathbf{L}_{{}^i\mathbf{p}}$ since the depth coordinate of the observed point is usually unknown, an estimate or approximation $\widehat{\mathbf{L}_{{}^i\mathbf{p}}}$ of $\mathbf{L}_{{}^i\mathbf{p}}$ has to be used in the control scheme. An observer

that satisfies persistent excitation conditions can, for example, be used for $\widehat{Z}$ estimation during the visual servo, as proposed in [De Luca et al., 2007]. Moreover, there are several ways to construct the estimate of $\widehat{\mathbf{L}_\mathbf{e}}^+$, for example, choosing $\widehat{\mathbf{L}_\mathbf{e}}^+ = \mathbf{L}_{\mathbf{e}^*}^+$ for the desired pose of the camera when $\mathbf{e} = \mathbf{0}$. In this case, $\widehat{\mathbf{L}_\mathbf{e}}^+$ is constant and only the desired depth Z^* of the point must be set, avoiding the estimation of 3-D parameters during the visual servoing task.

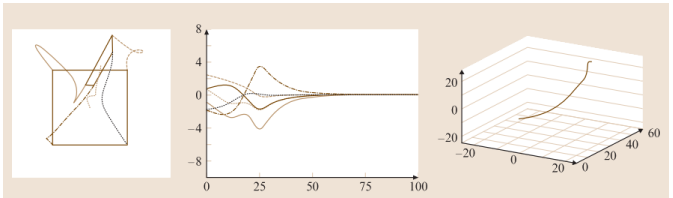
To control all degrees of freedom (DOFs) of the camera, at least three image points must be set as visual features $\mathbf{s} = ({}^i\mathbf{p}_1, {}^i\mathbf{p}_2, {}^i\mathbf{p}_3)$. For this case the interaction matrix related to $\mathbf{s}$ is given by:

$$\mathbf{L}_\mathbf{s} = \begin{bmatrix} \mathbf{L}_{{}^i\mathbf{p}_1} \\ \mathbf{L}_{{}^i\mathbf{p}_2} \\ \mathbf{L}_{{}^i\mathbf{p}_3} \end{bmatrix} \tag{2.11}$$

There are certain configurations for which $\mathbf{L}_\mathbf{s}$ is singular [Michel & Rives, 1993]. Additionally, there are four distinct camera poses for which $\mathbf{e}$ is zero, which means that there are four global minima for the error function $||\mathbf{e}||$ and it is not possible to differentiate them [Fischler & Bolles, 1981]. This is why more than three points are usually taken into account in IBVS (see Figure 2.2).



(a)



(b)

Figure 2.2 – Trajectories during IBVS for the same initial and desired camera poses. The considered visual feature vector represents four square corners image points. (a) Perfect estimation when $\widehat{\mathbf{L}_\mathbf{e}}^+ = \mathbf{L}_\mathbf{e}^+$. (b) Coarse approximation when $\widehat{\mathbf{L}_\mathbf{e}}^+ = \mathbf{L}_{\mathbf{e}^*}^+$. From the left column to the right: image points trajectories, $\mathbf{v}_c$ at each iteration of the control scheme, 3-D trajectories of the camera optical centre in the desired frame. From [Chaumette, 2007].

2.1.2 Target self-motion compensation

It is common practice to compensate for the self-motion of a moving target in visual servoing schemes. When the desired visual feature vector $\mathbf{s}^*$ is constant (though the same method is applied for non-constant $\mathbf{s}^*(t)$) the control law is given by:

$$\mathbf{v}_c = -\lambda \widehat{\mathbf{L}_\mathbf{e}}^+ \mathbf{e} - \widehat{\mathbf{L}_\mathbf{e}}^+ \frac{\widehat{\partial \mathbf{e}}}{\partial t} \tag{2.12}$$

where $\frac{\widehat{\partial \mathbf{e}}}{\partial t}$ is an estimation or an approximation of the time variation of the error $\frac{\partial \mathbf{e}}{\partial t}$ due to the target self-motion. This term enables the compensation of the target motion during visual servoing. By closing the control loop, the error dynamics becomes as follows:

$$\dot{\mathbf{e}} = -\lambda \mathbf{L}_\mathbf{e} \widehat{\mathbf{L}_\mathbf{e}}^+ \mathbf{e} - \mathbf{L}_\mathbf{e} \widehat{\mathbf{L}_\mathbf{e}}^+ \frac{\widehat{\partial \mathbf{e}}}{\partial t} + \frac{\partial \mathbf{e}}{\partial t} \tag{2.13}$$

which will not converge to zero even if $\mathbf{L}_\mathbf{e} \widehat{\mathbf{L}_\mathbf{e}}^+ > 0$ due to the tracking error $\mathbf{e}_{tr=} - \mathbf{L}_\mathbf{e} \widehat{\mathbf{L}_\mathbf{e}}^+ \frac{\widehat{\partial \mathbf{e}}}{\partial t} + \frac{\partial \mathbf{e}}{\partial t}$. There are various techniques to compensate for $\mathbf{e}_{tr}$:

- Setting a high gain λ to reduce the tracking error $\mathbf{e}_{tr}$ (although this may induce a risk of unstable control)
- Directly compensating for $\frac{\partial \mathbf{e}}{\partial t}$ if the target motion is exactly known (similar to what we use in our experiments with drones attached to the cable)
- Using of an integral action to compensate for constant velocity target motion $\frac{\widehat{\partial \mathbf{e}}}{\partial t} = \mu \sum_j \mathbf{e}_j$ with an integral gain μ to be tuned and $\sum_j \mathbf{e}_j$ being the cumulative error over previous control iterations.
- Using an estimation of $\frac{\partial \mathbf{e}}{\partial t}$ through image measurements $\mathbf{m}$ and camera odometry, such as the methods proposed in [Corke & Good, 1993] or [Bensalah & Chaumette, 1995].

In the next section, we will focus on the representation of Deformable Linear Objects (DLO) and how to manipulate their shape using visual servoing.

2.2 Modelling of Deformable Linear Objects

Deformable Linear Object (DLO) is a concept that represents a soft body whose length L is significantly larger than other dimensions [M. Yu et al., 2022], as illustrated in Figure 2.3. Deformable linear objects can be described using continuum mechanics.

Continuum mechanics is a branch of mechanics that deals with the behaviour of materials as if they were continuous, rather than composed of discrete particles. It provides a framework for analysing and predicting the response of materials to various mechanical loads and deformations.

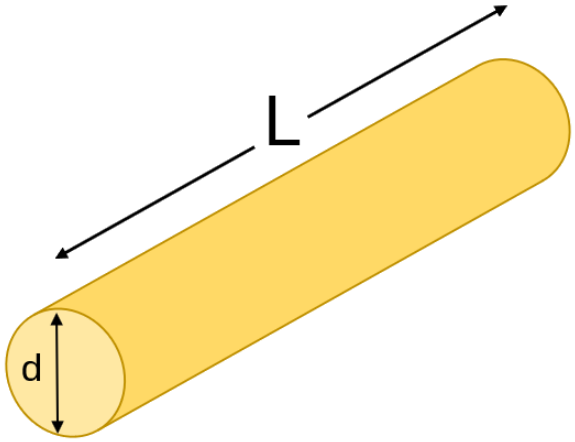


Figure 2.3 – Cable segment of length L with circular cross section diameter denoted d. It can be considered as a DLO if $L/d \geq 10$.

2.2.1 Relation between strain and stress in the DLO

In the context of deformable linear objects, we will focus on linear elastic cables made from materials which obey the Hooke's law. Consider $\mathbf{x_0}(\mathbf{m}) \in \mathbb{R}^3$ to be the position in the world frame $\mathcal{F}^w$ of a point $\mathbf{m}$ that belongs to a DLO in its initial rest configuration and let $\mathbf{x}(\mathbf{m})$ be its new position that results from the application of a deformation on the DLO. A displacement vector field can then be defined as $\mathbf{u} = \mathbf{x} - \mathbf{x_0}$ as illustrated in Figure 2.4 where the object coordinates of $\mathbf{m}$ are omitted for simplicity. Taking into account only small deformations, we consider the following relation between the strain tensor $\bar{\boldsymbol{\epsilon}}$ and $\mathbf{u}$, well-known as the Cauchy strain tensor:

$$\bar{\boldsymbol{\epsilon}} = \frac{1}{2}(\nabla \mathbf{u} + [\nabla \mathbf{u}]^T), \bar{\boldsymbol{\epsilon}} \in \mathbb{R}^{3,3} \tag{2.14}$$

The Hook's law gives the relation between the deformation of the material (*i.e.* strain $\bar{\boldsymbol{\epsilon}}$) and the internal forces resulting from such deformation (*i.e.* stress $\bar{\boldsymbol{\sigma}}$ in $\frac{N}{m^2}$ or Pa) for

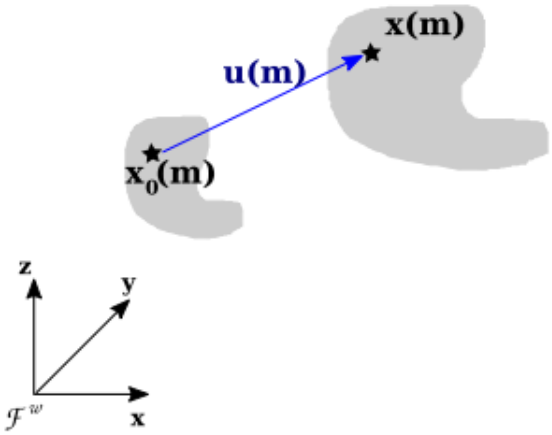


Figure 2.4 – Illustration of the displacement vector field **u**. From [Lagneau, 2020].

linear elastic object for which the stress is linearly dependent of the strain as follows:

$$\bar{\boldsymbol{\sigma}} = \bar{\mathbf{C}}\bar{\boldsymbol{\epsilon}} \tag{2.15}$$

where $\bar{\mathbf{C}}$ represents the material tensor being a rank four tensor. It linearly relates the stress tensor to the strain tensor. The deformation is considered small if it is less than or equal to 5% of the size of the object. An isotropic material is a material that maintains the same characteristics when it is deformed in different directions. In contrast, anisotropic materials are materials that demonstrate different properties when deformed in various directions. For an isotropic material, Hook's law can be expressed in the following form:

$$\bar{\boldsymbol{\epsilon}} = -\frac{\nu}{E} tr(\bar{\boldsymbol{\sigma}})\mathbf{I} + \frac{1+\nu}{E}\bar{\boldsymbol{\sigma}} \tag{2.16}$$

with $tr(\bar{\boldsymbol{\sigma}})$ representing the trace of the stress tensor $\bar{\boldsymbol{\sigma}}$, $\mathbf{I}$ the identity matrix, E represents the Young's modulus and ν is the Poisson's ratio. The linear part of the stress-strain curve for a material under tension or compression is described by the Young's modulus, which is the slope of the graph in Figure 2.5 [1]. If a cable segment with cross section diameter d and length L (see Figure 2.3) is subject to tension so that its length changes by ΔL, then its diameter d will change proportionally to Poisson's ratio for small deformations as:

$$\frac{\Delta d}{d} = -\nu\frac{\Delta L}{L} \tag{2.17}$$

1. Nicoguaro, CC BY 4.0 <https://creativecommons.org/licenses/by/4.0>, via Wikimedia Commons

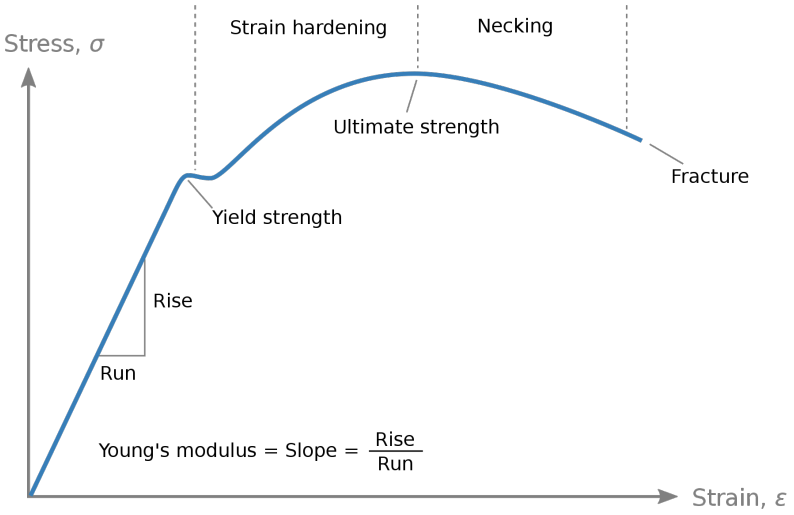


Figure 2.5 – Typical stress vs strain diagram for a ductile material (*e.g.* steel). From Wikipedia.

2.2.2 Analytic models of sagging cable

In this thesis, we are particularly interested in manipulating a DLO that corresponds to a cable. We consider that it can be modelled by a planar curve that only takes into account the centerline of the cable. Furthermore, the cable is subject to strain only in the tangent direction τ to the curve. Moreover, we assume that the cable is deformed by an external point load applied to its end and by gravity, and that it has reached a static equilibrium similar to [Yuan et al., 2015] (see Figure 2.6). Therefore, (2.16) becomes as follows:

$$\epsilon = \frac{\sigma}{E} \tag{2.18}$$

where ϵ, σ are now scalars. In Figure 2.6 $Oxyz$ stands for a reference Cartesian frame that is centred on the cable connection point B. The cable sags in the xOz plane that is parallel to the gravity g. The other end of the cable is attached at the point C. The point C can move in $Oxyz$, while the point B is fixed. The cable is clamped with rotational joints at both attachment points B, C. An imaginary straight line connecting points B and C by its length l_c and slope α is called the cord line. The maximum sag of the cable d is defined as the distance between the furthest point Q belonging to the cable and the

cord where the tangent vector τ is parallel to the cord. Let us consider an arbitrary point $P(x, z)$ on the cable. Its Cartesian coordinates $x(s), z(s)$ are functions of the Lagrangian coordinate s of the unstrained cable (curvilinear coordinate system with its origin in B). As mentioned before, the cable is deformed subject to external forces, *e.g.* weight (with

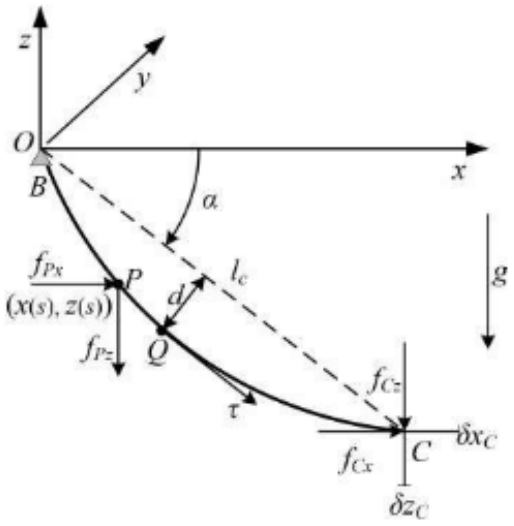


Figure 2.6 – Extensible catenary model of the cable (that is, the model used as a ground truth model by our simulator introduced in Chapter 3). From [Yuan et al., 2015].

its mass per unit length ρ $[kg/m]$), and external point load $\mathbf{f}_C = (f_{C_x}, f_{C_z})$. Under these forces, it is stretched into a sagging line. The resulting tension force at the point P is $\mathbf{f}_P = (f_{P_x}, f_{P_z})$. The static equilibrium equations are projected in the x-z plane as follows:

$$\begin{aligned} f_P \frac{dx}{dp} = f_{P_x} &= f_{C_x} \\ f_P \frac{dz}{dp} = f_{P_z} &= f_{C_z} - \rho g(l_{us} - s) \end{aligned} \tag{2.19}$$

where p stands for the stretched (or strained) cable Lagrangian coordinate, l_{us}, l_s are unstrained and strained cable lengths respectively, $\frac{dx}{dp}, \frac{dz}{dp}$ are cosine and sine of the angle of inclination respectively. If we consider a flat-sag cable profile with small elongation, the following geometric relation holds:

$$dx^2 + dz^2 = dp^2 \tag{2.20}$$

Finally, the Hook's law from (2.18) becomes as follows:

$$\left(\frac{dp}{ds} - 1\right) = \frac{f_P}{AE} \tag{2.21}$$

with A being the cable unstrained cross-sectional area. It could be interesting to know the cable displacement vector field from (2.14): $\mathbf{u} = (x(s), 0, z(s))$ by solving (2.19) to (2.21). The solution is given by [Yuan et al., 2015] as follows:

$$\begin{aligned} x(s) &= \frac{f_{C_x} s}{EA} + \frac{|f_{C_x}|}{\rho g}\left(\operatorname{arcsinh}\left(\frac{f_{C_z} - \rho g(l_{us} - s)}{f_{C_x}}\right) - \operatorname{arcsinh}\left(\frac{f_{C_z} - \rho g l_{us}}{f_{C_x}}\right)\right) \\ z(s) &= \frac{f_{C_z} s}{EA} + \frac{\rho g}{EA}\left(\frac{s^2}{2} - l_{us} s\right) \\ &+ \frac{1}{\rho g}\left(\sqrt{f_{C_x}^2 + (f_{C_z} - \rho g(l_{us} - s))^2} - \sqrt{f_{C_x}^2 + (f_{C_z} - \rho g l_{us})^2}\right) \end{aligned} \tag{2.22}$$

Note that in (2.22) both x and z depend on the external force applied to the cable distant endpoint C. In particular, the coordinates of this point are given by:

$$\begin{aligned} x_C &= x(l_{us}) = \frac{f_{C_x} l_{us}}{EA} + \frac{|f_{C_x}|}{\rho g}\left(\operatorname{arcsinh}\left(\frac{f_{C_z}}{f_{C_x}}\right) - \operatorname{arcsinh}\left(\frac{f_{C_z} - \rho g l_{us}}{f_{C_x}}\right)\right) \\ z_C &= z(l_{us}) = \frac{f_{C_z} l_{us}}{EA} - \frac{\rho g l_{us}^2}{2EA} + \frac{1}{\rho g}\left(\sqrt{f_{C_x}^2 + f_{C_z}} - \sqrt{f_{C_x}^2 + (f_{C_z} - \rho g l_{us})^2}\right) \end{aligned} \tag{2.23}$$

It is thereafter straightforward to obtain the expressions of the cord length l_c and slope α:

$$l_c = \sqrt{z_C^2 + x_C^2} \tag{2.24}$$

$$\alpha = \arctan\left|\frac{z_C}{x_C}\right| \tag{2.25}$$

Finally, the cable strained length l_s and the sag d are given by:

$$l_s = \int_0^{l_{us}} \sqrt{\left(\frac{dx}{ds}\right)^2 + \left(\frac{dz}{ds}\right)^2}\, ds \tag{2.26}$$

$$d = \frac{z_C x_Q - x_C z_Q}{l_c} \tag{2.27}$$

where the Lagrangian coordinate of the point Q is $s_Q = \frac{f_{C_x} z_C - f_{C_z} x_C}{\rho g x_C} + l_{us}$. The integral (2.26) can be computed by numerical integration.

The above-mentioned nonlinear analytic model of a static sagging cable is known as the extensible (or elastic) catenary model and provides ground-truth simulations for this thesis work. It allows us to find the forces applied to the cable ends by observing its end-point motion and to reconstruct the shape of the cable. However, we make the assumption that the elongation of the sagging cable can be considered negligible $\epsilon \simeq 0$ (which is reasonable when we consider stiff to stretch materials). The following mechanical model [Irvine, 1992] can be applied for such cable shape analysis (see Figure 2.7). The tension in some small element of the cable is denoted T, w is the weight of the cable per unit length. Vertical and horizontal equilibrium of a small element of the cable of the

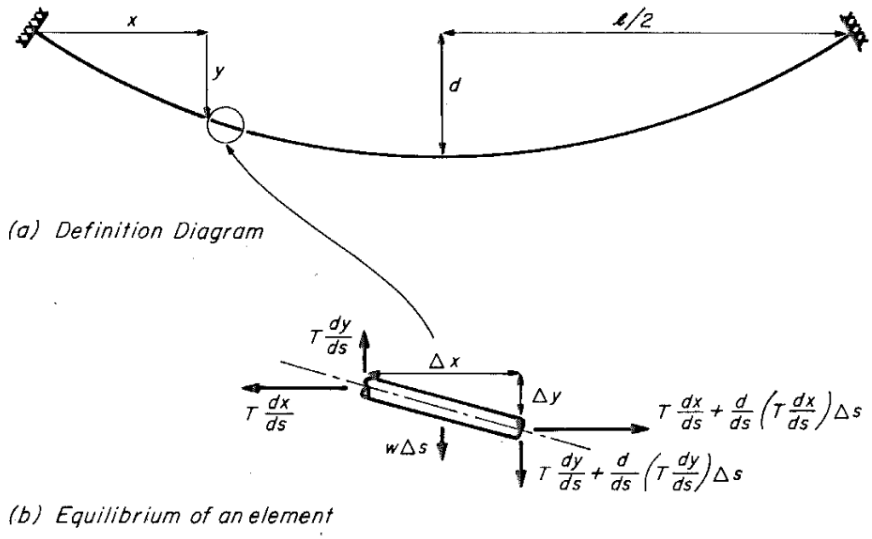


Figure 2.7 – The parabolic model that will be considered in this thesis. From [Irvine, 1992].

length ds are given by:

$$\begin{aligned} \mathrm{w} &= -\frac{d}{ds}\left(T\frac{dy}{ds}\right) \\ H &= T\frac{dx}{ds} \end{aligned} \tag{2.28}$$

The solution of this system of differential equations, for the coordinate system in Figure 2.7

is given by:

$$y = -\frac{\mathrm{w}}{2H}x^2 + \left(\frac{\mathrm{w}l}{2H} - \frac{s_v}{l}\right)x \tag{2.29}$$

where l stands for the horizontal span of the cable (and not its length) and H is the horizontal tension force component. Note that s_v in (2.29) represents the vertical distance between the cable extremities with respect to the y axis and is set to zero in Figure 2.7. It is straightforward to rewrite (2.29) in the form of a parabolic curve: $y = ax^2 + bx$. Note that if we can observe the cable profile with a vision sensor, it is possible to estimate the model parameters a and b given enough visual measures.

If we proceed further and consider the cable profile under a point load P acting at a distance x_1 from the origin (see Figure 2.8) the following equilibrium holds:

$$\begin{cases} (H+h)\frac{\mathrm{dv}}{dx} = P\left(1 - \frac{x_1}{l}\right) - h\frac{dy}{dx}, 0 \leq x \leq x_1 \\ (H+h)\frac{\mathrm{dv}}{dx} = -\frac{Px_1}{l} - h\frac{dy}{dx}, x_1 < x \leq l \end{cases} \tag{2.30}$$

where v represents vertical cable deflection and h being the increment in the horizontal component of cable tension due to the point load. The right-hand side of (2.30) represents a shear force in a supported beam of uniform weight under point load action. There is

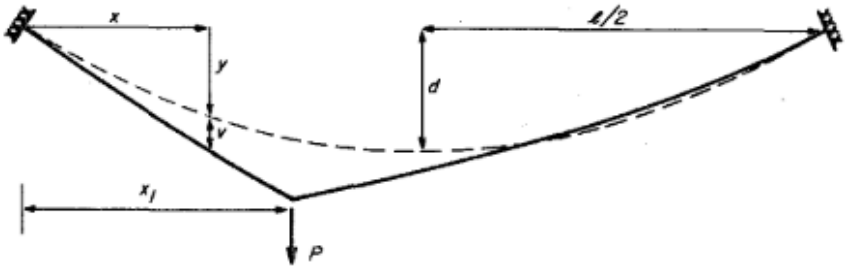


Figure 2.8 – Model of a cable under a point load. It is demonstrated that it is composed of two parabola. From [Irvine, 1992].

an analytic solution to find v by integrating (2.30) and rearranging terms. We obtain the

following system:

$$\begin{cases} \mathrm{v} = \overbrace{\left(\frac{2P^*m}{h^*}(1-\frac{x_1}{l})-m\right)}^{e+c} x + \overbrace{\frac{m}{l}}^{d} x^2, 0 \leq x \leq x_1 \\ \mathrm{v} = \underbrace{\frac{2P^*mx_1}{h^*}}_{f} + \underbrace{\left(-\frac{2P^*mx_1}{h^*l}-m\right)}_{c} x + \frac{m}{l}x^2, x_1 \leq x \leq l \end{cases} \tag{2.31}$$

where $m = \frac{Pl}{H}\frac{h^*}{2(1+h^*)P^*l}$, $h^* = \frac{h}{H}$ and $P^* = \frac{P}{\mathrm{wl}}$. It can be seen in (2.31) the parabola form of the cable deflection v even though it is discontinuous in x_1. Note also that h from (2.31) can be found using the equation of the length of the cable segment and according to [Irvine, 1992] the solution is a cubic equation with only one positive real root. The discontinuity in (2.31) is in x_1 due to shear with the same second-order coefficients for both equations. Now, let us gather all terms to simplify the system (2.31) as follows:

$$\begin{cases} \mathrm{v} = (e+c)x + dx^2 \\ \mathrm{v} = f + cx + dx^2 \end{cases} \tag{2.32}$$

and recalling the inextensible parabola profile as $y = ax^2 + bx$ we obtain the final solution of the sagging cable subject to the point load y_{pl} as follows:

$$y_{pl} = y + v = \begin{cases} (d+a)x^2 + (b+e+c)x, 0 \leq x \leq x_1 \\ (d+a)x^2 + (b+c)x + f, x_1 < x \leq l \end{cases} \tag{2.33}$$

It is straightforward that the obtained new cable profile represents two parabola, the second one shifted by f in y when $x = x_1$. We can use the Heaviside step function $\xi = \begin{cases} 1, 0 \leq x \leq x_1 \\ 0, x_1 < x \leq l \end{cases}$ centred on x_1 to write (2.33) in the following form:

$$y_{pl} = \underbrace{(a+d)}_{a_{new}} x^2 + \underbrace{(b+c+e\xi)}_{b_{new}} x + f(1-\xi) \tag{2.34}$$

Thanks to the developments by [Irvine, 1992] it is clear that $f = f(H, x_1, l_{us}, E, A)$ remains constant if the load does not slide on the cable (static equilibrium of the load) and small for a stiff enough cable that does not stretch under the load. Note also that (2.34) provides us with a continuous differentiable solution for the distal end coordinates as a function of

the parabolic parameters a_{new}, b_{new}.

Further, the physical meaning of the parabola coefficients are explained. For example, a represents the ratio between the cable weight per unit length and the horizontal cable tension force component H as follows: $a = \frac{-\mathrm{w}}{2H}$, and hence a is measured in $[\frac{1}{m}]$. Analogously, $d = \frac{h\mathrm{w}}{2H(H+h)}$ represents the increment to a due to the influence of the point load force. The dimensionless coefficient given by $b = \frac{\mathrm{w}l}{2H} - \frac{s_v}{l}$ accounts for the horizontal shift of the parabola vertex and dimensionless c, e for its increment due to point load. Finally, f in $[m]$ represents the shear value at x_1.

This analysis inspired by [Irvine, 1992] and [Megson, 1996] demonstrates that the shape of the sagging cable can be modelled by one segment of a parabolic curve without a load and two segments of a parabolic curve if a point load is applied, and shows the physical meaning of the parabolic coefficients (further parabolic features) that play a central role in this thesis work.

In the next section, some recent works on controlling the shape of a DLO to a desired shape will be presented. All of the further-mentioned works use unimodal sensing of the DLO *i.e.* vision with a camera to track DLO shape during shape control similar to visual servoing discussed in previous sections.

2.3 Manipulation of Deformable Linear Objects

The manipulation of Deformable Linear Objects (DLO) by a robotic manipulator is a challenging task due to the high number of DLO degrees of freedom (DOF) and the low number of robot DOF. Therefore, a robot acting on some local region of the DLO can cause its deformation in another region, which should be taken into account in the robot controller to handle undesirable deformations. Cable routing and cable inserting are common examples of DLO manipulation tasks in industry. Since these tasks are still mainly performed manually, their automation by robots has a great potential as mentioned by [Fresnillo et al., 2022; Galassi & Palli, 2021; Jin et al., 2022; Zürn et al., 2022].

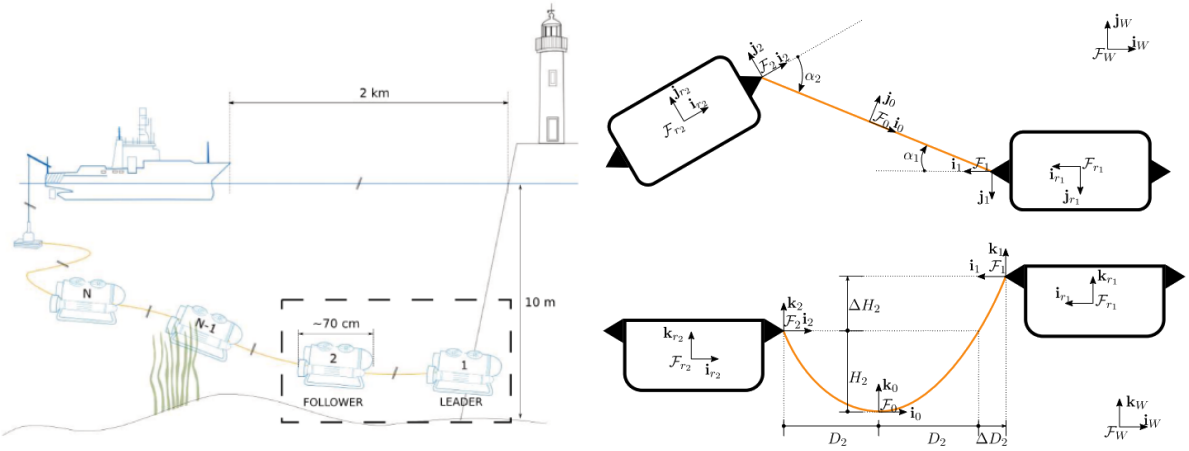
In the field of aerial manipulation using cables, a potential application would be to observe the site of an earthquake using a cable towed drone [Kiribayashi et al., 2017]. Another application is the installation of power lines by drones [Malveiro & Cordeiro, 2017; Pawlak & Serek, 2017]. There exist also various load transport applications [Gabellieri et al., 2023; Sanalitro et al., 2022; Tognon et al., 2018; G. Yu et al., 2023]. Drones can also provide significant potential for last-mile delivery [Moshref-Javadi et al., 2020], which

corresponds to the final step of the delivery process, when a package is transported from a transport hub to its final destination, typically a home or retail store.

In Section 2.3.1 we review methods that use a DLO model to control its shape. Afterwards, Section 2.3.2 gives an overview of shaping methods that do not need any DLO model. Finally, Section 2.3.3 deals with the manipulation and control of DLO shape using aerial robots.

2.3.1 Model-based deformable body manipulation

Controlling the shape of power cables that connect a chain of underwater robots was proposed in [Laranjeira et al., 2020]. The authors consider the catenary model with an RGB camera in order to establish a visual servoing control law for the chain of underwater robots linked by the power cables to maintain a desired shape of that cable (see Figure 2.9(a)). A visual feature vector is proposed to track the cable shape by an RGB



(a) (b)

Figure 2.9 – (a) Illustration of a chain of underwater vehicles linked by power cables during mission. (b) Top view of the cable-linked leader-follower formation. The bottom is the side view of the formation. From [Laranjeira et al., 2020].

camera and to perform visual servoing to maintain some desired shape. The set of visual features (see Figure 2.9(b)) corresponds to $\mathbf{s} = (a, b, d)$ where $a = \frac{H_2}{H_{max}}$ with H_2 being the sag between the robot r_2 cable attachment point and the lowest point of the catenary and H_{max} is the maximum acceptable sag. $b = \sin\alpha_2$ with α_2 representing the yaw angle of the follower robot with respect to the cable plane. The third feature $d = \frac{\Delta H_2 + \Delta H_{max}}{2\Delta H_{max}}$ (with ΔH_{max} being the maximum acceptable altitude between the cable

attachment points) is estimated by two robot external pressure sensors. The authors used non-linear least-squares fitting of the catenary curve to estimate the features $\mathbf{s}$ from the camera image. The proposed visual servoing scheme controls four components of the twist of the follower robot r_2, which are its translational velocity and its yaw angular velocity. The authors show that singularities of the corresponding interaction matrix occur when the cable attachment points are on the same vertical or when the cable plane is perpendicular to the robot longitudinal plane and when the cable represents a straight line. The main drawback of the method is the presence of a singularity when the cable is a straight line. Moreover, the authors used vision to estimate only two features, while the third is estimated with a pressure sensor that may be unavailable or less accurate when using aerial robots.

The authors of [Koessler et al., 2021] introduce a method for controlling the shape of a sheet of rubber based on a finite element formulation. They consider the control node velocities as features such that the coordinates of target nodes converge towards their goal values using a Finite Element Model (FEM) that formulates locally the static equilibrium equations for these points to avoid real-time model simulations and directly derive control outputs. The authors used 1D rod elements for FEM, taking into account the deformations that result mainly from the bending of the rod under its own weight. RGB-D vision is used to track the displacement of nodes of an object that can be partially occluded. The experimental setup is shown in Figure 2.10. The main drawback of this

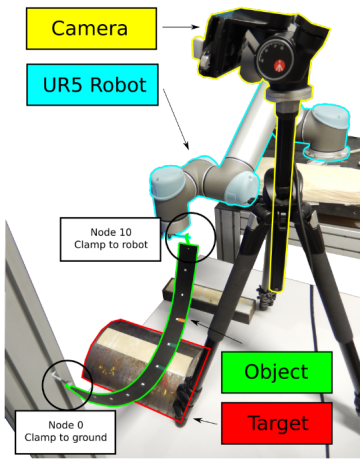
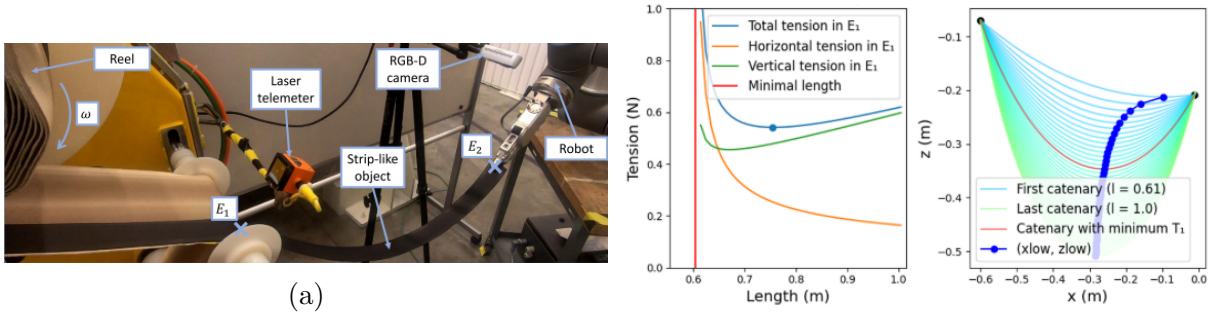


Figure 2.10 – Experimental setup for sagging rubber sheet manipulation from [Koessler et al., 2021].

method is that it only considers 1D beam elements, as the parameter estimation for nodes

is too difficult in more general cases.

Another example of industrial tasks is the extraction of elasto-plastic objects in strips from an industrial reel. In order to manage the mechanical stresses inside the object, [Filella et al., 2022] propose a vision-based control system to reduce tension by adjusting the angular speed of a motorised reel around which the strips are wound (see Figure 2.11(a)). The main idea is to control the tension in the sagging profile of the strip by varying the length of the segment. An inextensible catenary model was used to represent the tension T in the strip profile. The authors propose to control the angular



(a) (b)

Figure 2.11 – (a) A strip attached to the robot gripper in point E_2. The point E_1 is the beginning of the sagging strip (b) Tension distribution as a function of the length l of the catenary (left). Evolution of the shape of the catenary profile while l is increasing (right). From [Filella et al., 2022].

velocity ω of the reel of radius R by the following control law:

$$\omega = \frac{1}{R}\frac{dl}{dT}K(T - T_d) \tag{2.35}$$

where T_d is the desired tension in the strip and $K > 0$ is the control gain. The authors used simulation to find the lowest feasible tension T_d while modifying the length of the sagging part of the strip l from the length of a straight line E_1E_2 (see Figure 2.11(b)) to some maximum length ($l \leq 3$). The term $\frac{dl}{dT}$ is estimated using RGB-D vision and the inextensible catenary model. The authors reported many scenarios of experiments in which the method performs efficiently. The main drawback of the method is that, when the length increases, the object shape is no longer a catenary *i.e.* the fluctuation of the control output makes the shape of the object more sensitive to oscillations. The thickness of the stripe also plays an important role.

The work of [Liu et al., 2023] considers a geometric model of a DLO where its shape

is composed of nodes connected by 3D-oriented segments (or ropes). The method is based on the XPBD simulator of deformable objects from [Macklin et al., 2016] that can represent the coupling of shear/stretch and bend/twist effects of a DLO. The main advantage is that unlike force-based methods such as the Euler-Bernoulli beam or Cosserat rod theory [Soler et al., 2018], full strain and torsion deformations can be updated using a position-based dynamics solver. A set of geometric constraints (*e.g.* describing shear and stretch, bend and twist) for nodes displacement is formulated with corresponding gradients, making the method very easy to integrate into hardware optimised solvers. Given the above constraints, the dynamics of a rope-like object can be represented by a set of discrete particles with position and orientation evolution (*i.e.*, the 3-D positions of nodes and 3-D orientations of the rods connecting a couple of consecutive nodes). However, this is only an approximation of the real model for the rope objects. Additional weight parameters are added to the constraints, taking into account the values of real object parameters that must be identified. The DLO shaping and the adjustment of the Real-to-Sim parameters of the object is performed by an optimal control framework. The authors used 3-D point clouds of the observed DLO projected onto a hyperplane defined by the gravity vector. To match the reality and model parameters, the authors proposed several criterion functions and tested them in various combinations for different static shapes of the DLO. The authors have obtained a full discretised model of the DLO and used it in shape control experiments to predict the shape and, consequently, derive the positions of the nodes rigidly attached to the robot for position control (see Figure 2.12). The main drawbacks are complex perception of the shape (need for markers), the need to identify model parameters for a particular DLO material and to run the model simulation during shaping task. The experimental results show a high generalisation capability of the proposed method, but for some target shapes the convergence is not obtained.

The authors of [Boyer et al., 2023] used simulations to model the behaviour of the system composed of two DLO being modelled as two Cosserat rods [Soler et al., 2018], each attached to a hexarotor from one extremity and to a cubic payload from another (see Figure 2.13). Both extremities represent fixed joints. The authors neglected gravity in their simulation to demonstrate a wide range of possible shapes of the DLO in interaction with drones. Drones are simulated to follow a circular trajectory. The authors demonstrate the efficiency of their improved Newton-Euler recursive algorithm to compute the deformation of the DLO subject to the motions of the drones and then to compute the resulting wrenches from those deformations acting back on the dynamics of the drones. The

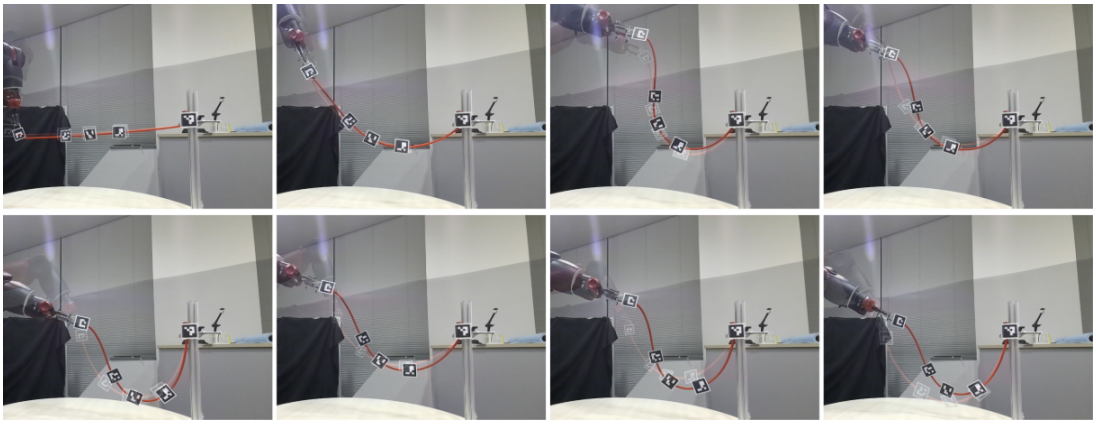


Figure 2.12 – Experimental setup for sagging cable manipulation. The cable image with law opacity represents the target shapes, while the solid opacity represents the result of the control. Markers are used to estimate the ground truth shape. The three markers in the middle represent three visual features (the origin of each marker) used to compute a position of the left most marker, which is then used to control the robot end-effector (3 translational DOFs of the robot used for the control task). From [Liu et al., 2023].

dynamics of each drone is assumed to be able to react instantly to the external wrench from any direction (which is not possible in the case of a quadrotor UAV). The main drawback of the method is the lack of experimental results using real hexarotors. It is also computationally expensive and requires powerful hardware, making it difficult to use for embedded applications on small drones. Finally, the proposed DLO model requires many parameters to be estimated from its shape, which makes it difficult to use with a limited number of onboard sensors (markers or bending sensors need to be attached to the cables).

2.3.2 Model-free deformable body manipulation

In contrast to the previous section, we will now discuss methods that do not require any knowledge of the model of a deformable object.

In connection with the two methods that we will mention further, a model-free approach is proposed by [Zhu et al., 2018] applied to 2-D wire manipulation with two robotic arms. The shape is tracked by a monocular camera. The visual feature vector $\mathbf{s}$ represents the coefficients of the Fourier series decomposition of the cable profile. A deformation Jacobian is numerically estimated from past measurements of the robot motions and visual feature variations. This deformation Jacobian relates the variation of the visual features

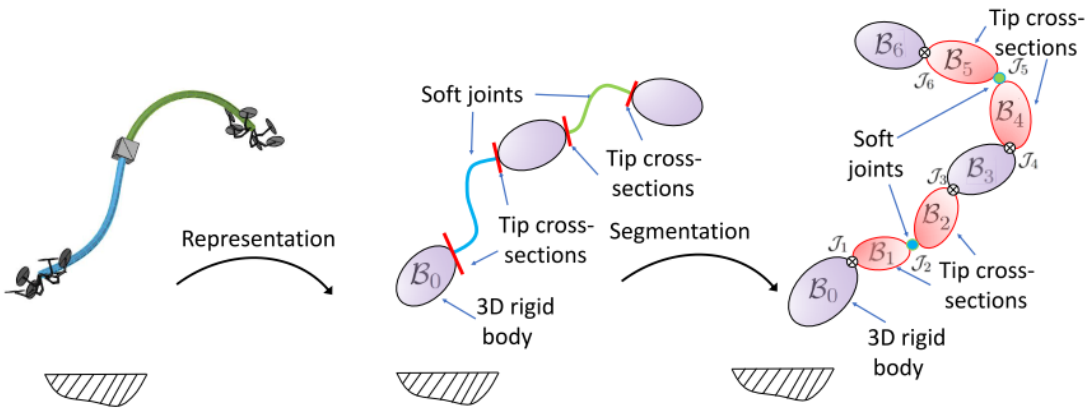
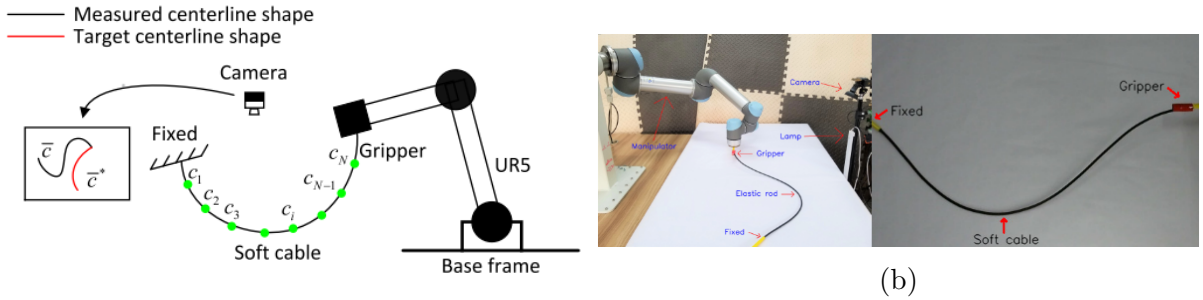


Figure 2.13 – Left image: the overall 3D view of the simulation. Middle image: representation of the system composed of rigid bodies, soft joints and tip cross-sections. Right image: final representation of the system used in the algorithm, tip cross-sections are represented as rigid bodies. From [Boyer et al., 2023].

to the robot motion and its inverse is used in the control law to compute the velocity of the robot that minimises the shape visual error. However, the numerical estimation of the deformation Jacobian from the measured data does not guarantee the global stability of the system. Moreover, unknown singularities of the shape Jacobian and inability to converge precisely to zero shape error remain common drawbacks of all model-free methods.

The authors of [Qi et al., 2020] propose a vision-based shape servoing method for various possible shapes of a cable, *i.e.* the shape of a cable is no longer a catenary, but some continuous curve, as can be seen in Figure 2.14(b). The authors propose to



(a) (b)

Figure 2.14 – (a) Illustration of experimental setup of cable manipulation by a robotic arm. (b) Experimental setup with a cable manipulated by shape visual servoing. From [Qi et al., 2020].

estimate the shape Jacobian by using Kalman filters and consider as visual features the

coefficients of Bezier or NURBS curves to efficiently represent elastic rods. An adaptive gain controller is proposed and experimental results are presented. The cable is modeled as a discrete curve with N points $\mathbf{c}_i = \mathbf{f}(\rho_i)$ where ρ_i represents the arc length between the start point $\mathbf{c}_1$ and $\mathbf{c}_i$. Pixel coordinates of these points are measured with a monocular camera. The feature vector $\mathbf{s}_k$ at the current time step k is calculated from $\mathbf{c}$ as follows:

$$\mathbf{s}_k = \mathbf{G}^+(\rho)\mathbf{c} \tag{2.36}$$

with $\mathbf{G}$ being a regression matrix which corresponds to different linear parameterization with respect to $\mathbf{c}$ *e.g.* polynomial, Bézier, NURBS, Fourier. The authors assume that a small robot motion Δr_k produces a small variation of $\Delta\mathbf{s}_k$ which can be described as a local deformation model of DLO (as demonstrated in [Navarro-Alarcon et al., 2016]) as follows:

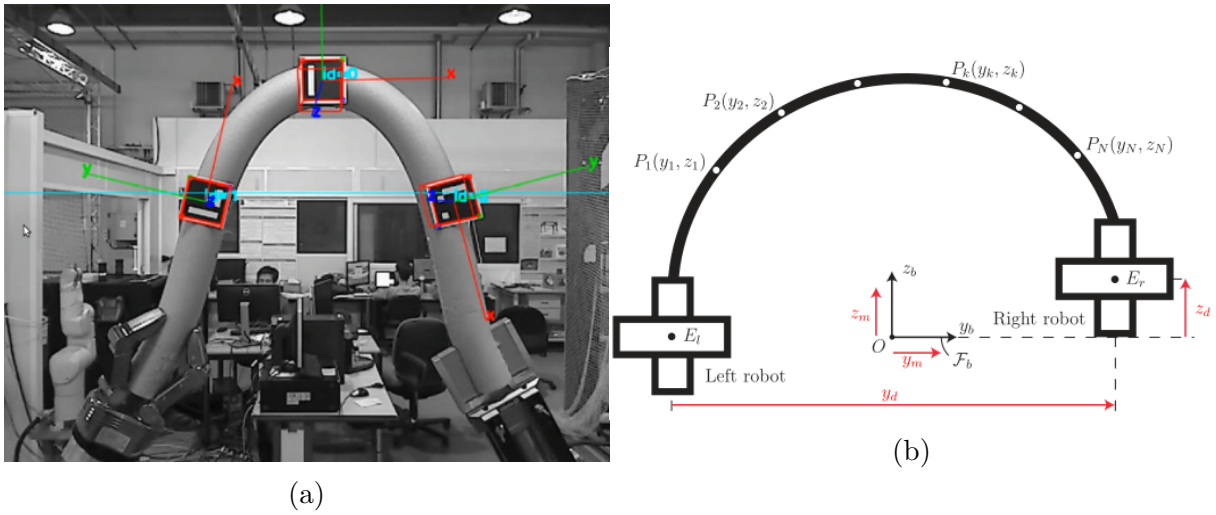
$$\Delta\mathbf{s}_k = \mathbf{G}^+\mathbf{D}_k\Delta\mathbf{r}_k \tag{2.37}$$

where the shape Jacobian is $\mathbf{J}_k = \mathbf{G}^+\mathbf{D}_k$ with $\mathbf{D}_k$ representing the unknown model of the cable. This Jacobian transforms the robot motions into shape changes, however, it cannot be calculated analytically as the deformation characteristics of the object are not known. Instead of identifying the full mechanical model $\mathbf{D}_k$, the authors propose a filter to approximate $\mathbf{J}_k$ during visual servoing. The main drawbacks of the method are its limited local stability, *i.e.* convergence of the shape error norm $||\mathbf{s} - \mathbf{s}^*||$ to a local region around the origin (which is common for all model-free approaches) and that the proposed visual features are only valid for a planar manipulation of the cable (*e.g.* on a table).

3-D shape manipulation of wires by two robotic arms is proposed in [Lagneau et al., 2020]. The authors define a new visual feature $\mathbf{s}$ based on a B-spline curve to track and control the wire deformations with an RGB-D camera. A particle filter is used to estimate the best spline that fits the observed point cloud of the wire. Afterwards, a number of equidistant 3D points are sampled from the best spline. These points represent the visual feature vector $\mathbf{s}$ used in the local deformation model of the DLO. A new approach to online update the shape Jacobian from previous measures of variation of $\mathbf{s}$ and robot odometry is proposed. It is based on least-squares minimisation with sliding window and was successfully evaluated during experiments. The main drawback remains the local stability of the approach, but the authors propose to generate a set of intermediate target shapes to achieve a desired deformation that is far from the initial wire state.

Two robot arms are used in [Caro et al., 2021] to manipulate a flexible beam to adapt

its shape before an assembly operation. In this work, the main idea is to represent a flexible beam as a planar curve composed of a finite number of keypoints (and expressed in the robot body frame $\mathcal{F}_b$) that can be tracked by a monocular camera using markers (see Figure 2.15(a)). The relation between small displacements of these keypoints coordi-



(a) (b)

Figure 2.15 – (a) Two KUKA LWR 4+ 7-dof robotic arms are used to grasp a flexible beam, which has three ArUco Markers attached to it in order to monitor its shape. (b) The diagram shows N keypoints of the beam, denoted as P_i, and the shape control signals, which are represented in red. The centres of the left and right robot end-effectors are labelled as E_l and E_r respectively. From [Caro et al., 2021].

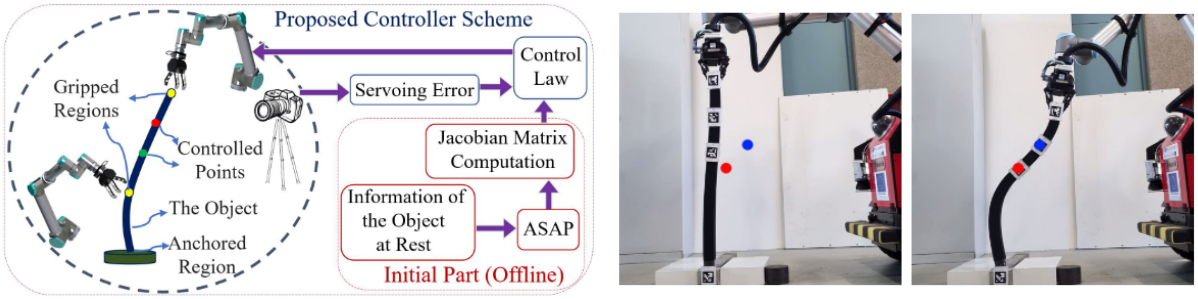
nates (y_k, z_k) and small changes in the relative position of the arms (y_d, z_d) and common translation of the arms (y_m, z_m) (red signals on the Figure 2.15(b)), all expressed in the robot body frame $\mathcal{F}_b$, is called sensitivity Jacobian matrix of size $(2N \times 4)$, where N is the number of keypoints, of the following form:

$$\mathbf{J} = \begin{bmatrix} \mathbf{j}_{yd} & \mathbf{j}_{zd} & \mathbf{j}_{ym} & \mathbf{j}_{zm} \end{bmatrix} \tag{2.38}$$

The last two columns of $\mathbf{J}$ are constant vectors and correspond to the translation of the entire beam, which means that all keypoints exhibit a common translation. The two first columns represent a small change in the keypoint positions (y_k, z_k) relative to a small change in the control signals (y_d, z_d) given by $\frac{\partial y_k}{\partial y_d}, \frac{\partial z_k}{\partial z_d}$ and must be identified either in simulation or statistically from series of experiments. The control law is then similar to the one of classic visual servoing $\dot{\mathbf{u}} = -\lambda \widehat{\mathbf{J}}^{+}(\mathbf{x} - \mathbf{x}_{des})$ where $\dot{\mathbf{u}}$ is a vector of control signals related to (2.38) and $\mathbf{x}$ is a vector stacking the keypoint coordinates. The drawback of

the method is that it is unable to converge to a zero error at the end of the task due to the absence of a deformation model of the flexible beam in the control law.

The authors of [Aghajanzadeh et al., 2022] proposed a control scheme for fixed-length elastic linear objects lying on a 2D workspace. The main idea is to encode the object deformation behaviour in an offline constant Jacobian matrix (see Figure 2.16(a)). It is done based on ASAP (As-Similar-As-Possible) modelling of a deformable object from [Chen & Gotsman, 2016]. The shape of a deformable object is represented by a set of adjacent nodes sampled uniformly along the object middle line, grouped by three, hence called a triad. The object rest shape is needed for the ASAP model. ASAP employs a two-dimensional formulation of the deformation energy, which is a sparse symmetrical matrix based on the triad structure and the shape of the object in its resting shape. The authors express the nodal force equilibrium equations similar to FEM with the help of ASAP giving the relation between the gripped and servoed nodes of the manipulated object. To control the position of the servoed nodes, a classic visual servoing scheme is used. The shape Jacobian is composed of a constant part due to the gripped nodes rigidly attached to the grippers and a constant part from ASAP. This method does not require to estimate



(a) (b)

Figure 2.16 – (a) A schematic representation of the ASAP-based control scheme. (b) Precise positioning of the two points represented in blue and red dots. From [Aghajanzadeh et al., 2022].

the shape Jacobian during visual servoing. The main drawback is the need for markers to track the positions of the controlled points in the DLO. In addition, an elastic behaviour is assumed with the hypothesis that the DLO resists to the gravity and returns to its rest shape (a vertical straight line) when no external deformation forces (exerted by a robot) are applied. This is generally not the case for suspended cables used in aerial manipulation of DLO that we discuss in the next section.

2.3.3 Aerial manipulation of DLO

Accurately and automatically controlling the shape of a suspended cable can also be useful for Unmanned Aerial Vehicle (UAV), such as managing the shape of the power cable of tethered UAVs. A robotic platform composed of multiple drones to effectively collect waste from rivers and canals represents another example of the application of tethered drones. An elastically suspended drone can be used as an end effector of a cable-driven parallel robot, as proposed by [Perozo et al., 2022]. This design allows one to compensate for the limitations of cable-driven parallel robots (complexity of control for handling cable tensions) and aerial manipulators (limited flight time).

For instance, in [Xiao et al., 2018] and [Kiribayashi et al., 2017] a cable attached to a drone was used to locate the UAV when GPS is not supported. The authors used a catenary model and measured the tension or the length of the cable to locate a tethered UAV. For managing the cable of a tethered UAV, [Zikou et al., 2015] proposed a device composed of a winch mounted on the ground and estimated the cable model parameters using a second order polynomial (a parabola) decomposition of the mapping between the winch drum angle and the cable length. The authors proposed a cascaded control scheme to maintain a desired length of the free part of the cable by winch rotation and an estimator (based on the catenary model) of the cable tension in order to generate an appropriate targeted length, which is controlled to maintain the power cable in a slack configuration to avoid pulling the drone with a taut cable and touching the ground with a very slack cable. The proposed mechanism is capable of controlling the shape of the power cable of the attached UAV while performing other tasks.

In [Micotra et al., 2014] the authors present a geometric controller for an inextensible and massless power cable for a tethered UAV in order to maintain the cable taut. They formulate the control stability conditions for the UAV under cable tension.

Another possible application is aerial transportation performed by a flying gripper made of UAVs connected to a point load by a cable [G. Yu et al., 2023] or by cable-suspended parallel robots actuated by flying robots [Erskine et al., 2018], [Sanalitro et al., 2022], [Tognon et al., 2018], [Gabellieri et al., 2023]. In these works, the cables are assumed to be taut under the weight of an attached platform or a rigid beam. Cable sag detection and control strategies could increase the number of possible applications of these robots allowing for contacts with the environment.

In [D'Antonio et al., 2021], the authors propose a flying gripper made of two UAVs linked by a cable (see Figure 2.17). The cable is modelled with a catenary model with the

following parameters: the yaw angle of the vertical plane containing the cable, the span of the cable between the two drones, and the position of the lowest point of the catenary. The authors estimate these parameters with the UAVs odometry under the assumption that the two drones remain on the same altitude with the same yaw. It is worth noting that the method compensates for cable tension. The tension in the cable increases with its mass, necessitating the quadrotors to tilt at a steeper angle to compensate for it. This work show experiments where the catenary robot is used to grasp and transport an object. Nonetheless, it was not explained how the catenary model is affected by the point load. The main drawbacks of the proposed approach are the non tracking of the cable which

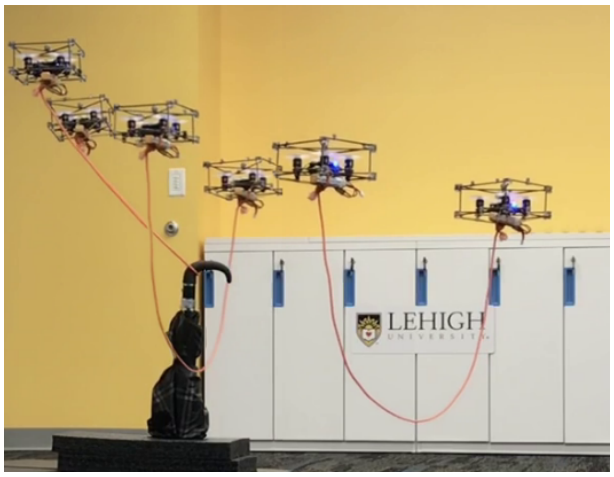


Figure 2.17 – A robot made up of two quadrotors connected to either end by a cable. By controlling the lowest point, span, and orientation of the cable curve, this robot can be used to manipulate objects, such as an umbrella. From [D'Antonio et al., 2021].

leads to an open loop control of the cable shape and hard constraints on the motion of the drones which are forced to follow the same yaw and same altitude.

In their next work the authors of [D'Antonio & Saldana, 2022] propose multi-catenary robot to fold knots forming a kind of net potentially convenient for objects transportation. The same catenary model is used to form a continuous piecewise function where knots are defined by crossing conditions between a pair of catenaries in the predefined grid resulting from the knot topology. Authors demonstrate fold knotting in simulations (see Figure 2.18). The trajectory, a time parameterised piece-wise fifth-degree polynomial, is constructed based on the catenaries crossing points or target points found from knot topology. A leader-follower approach is employed in which the initial robot (leader) follows a trajectory and the subsequent robots follow the leader with some delay. The

main drawback might be the lack of experimental results for the proposed knot folding.

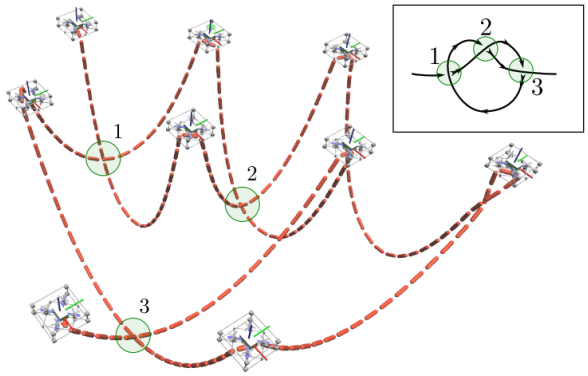


Figure 2.18 – A group of aerial robots are folding a knot, with the green disks indicating the crossings of the knot. From [D'Antonio & Saldana, 2022].

The authors of [Gabellieri & Franchi, 2023] proposed a system composed of two quadrotors connected with a cable. A dynamic model was developed which is based on a discrete representation of a deformable and extensible cable (see Figure 2.19). This model is composed of lumped masses connected by linear springs through passive spherical joints. The system produces a set of flat outputs, which is used to design a method to manipulate the cable. This method is tested through numerical simulations. Mass-link and mass-quadrotor connections are modelled as passive spherical joints. Given the desired trajectory for the outputs of the system (*i.e.* positions of two consecutive points of the discretised cable and yaw of each drone and their derivatives respectively) the spring force acting on the cable segment between these points is calculated and then propagated in both directions to find desired positions of each of drones and their derivatives. An integral term was also considered to track for the error between desired point of the cable and the actual one. The main drawback of the proposed approach is the increasing complexity when modelling for more points: as the number of segments used to divide the cable increases, the order of derivatives needed also increases, making the closed-form solutions for the derivatives more and more complex as the cable is divided into smaller parts. The authors demonstrate the efficiency of the proposed approach through simulations.

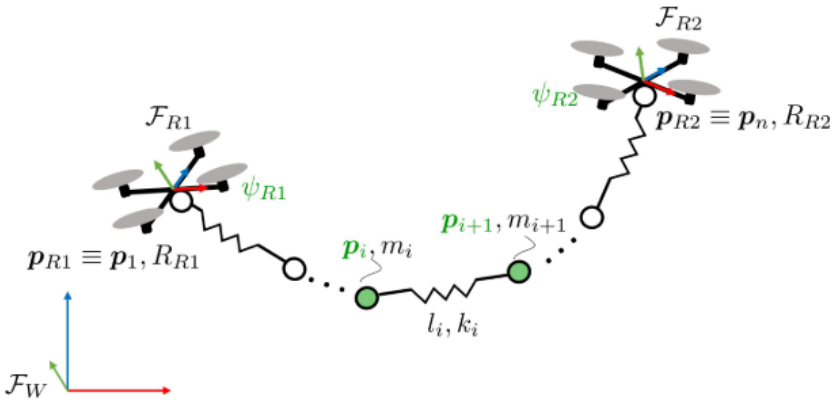


Figure 2.19 – A diagram of the system made up of two quadrotors manipulating a flexible and elastic cable. The main components of the system are displayed and the flat outputs are marked in green. From [Gabellieri & Franchi, 2023].

2.4 Conclusion

DLO manipulation by robotic manipulator or drones is an area of growing interest. There is an increasing industrial need for aerial manipulation with cables. However, manipulating deformable objects is more complicated, as it is not only required to estimate some translations and rotations of the object frame, but also the new position of each object particle within that frame (unlike a rigid body for which these positions are constant). This can be achieved using continuum mechanics equations. Unfortunately, these equations become intractable for more general configurations of deformable objects for real-time shape control/manipulation. However, by considering some assumptions in those equations, a simple solution can still be applied in a high variety of industrial tasks. That is why finite element methods are studied for DLO manipulation purposes. In recent years, research has been very productive in proposing new methods for manipulating soft objects. Two branches of methods can be distinguished: model-based and model-free data-driven methods. Both have strong sides and weaknesses. However, there exists a tendency to generalise approaches in order to manipulate a greater number of deformable object features. We can also distinguish some particular application scenarios that require manipulating a DLO, especially for aerial manipulation, when the DLO can be considered as a line/spring or some geometric curve, *e.g.* a catenary. Serial manipulators allow us to work with various deformable objects with ease, being sustainable to the force reactions of these objects, whereas small quadrotor drones are vulnerable to these

reactions, requiring the compensation of these reactions. Moreover, most industrial tasks involving serial manipulators are produced in well-constrained environments wherein the aerial manipulation tends to go into the "wild", which imposes different levels of sensor uncertainty and disturbance needed to be taken into account during the task. Drones can be fast, but most deformable object manipulation methods rely on static assumptions. Drones have a larger working area than standard manipulators, providing an opportunity to use them for aerial transport scenarios. In such scenarios, deformable objects are used as compliant grippers, allowing the drone to compensate for the lack of precision and disturbance. Deformation of these grippers can produce the grasping action or drive the overall system to some configuration in space, thus avoiding potential collisions between drones/objects in space.

2.5 Positioning of this thesis with regards to the literature

The approach proposed in this thesis is based on the model-based branch of methods, but requires knowledge of only one simply identifiable physical parameter of the manipulated object, its length. Moreover, it uses an analytic formulation of the shape Jacobian (thanks to the model), allowing us to guarantee the stability and to point out all possible singular configurations of the control law, which is particularly important when using drones or manipulating in the particular shape of DLO. In contrast to more general model-based methods, the proposed features of DLO are extracted from a markerless tracking of the object. Moreover, our method can be executed at camera frame rate using the embedded hardware of a small quadrotor drone and profit from the embedded sensors in the drone, yet being robust to its noise. Compared to the model-free branch of methods, ours does not require any initialisation phase for the shape Jacobian, reveals potential singular configurations when manipulating DLO and is proven to be robust to the RGB-D sensor noise while tracking the object features and calculating the shape Jacobian. Our method does not require a particular pose of the camera with respect to the manipulated object in order to extract its features. Our method is tested on a robotic manipulator as well as on quadrotor drones in different manipulation scenarios, from conventional slack cable manipulation to load grasping and transportation, demonstrating the generalisation and efficiency of the method for different tasks.

Chapter 3

Shape visual servoing of a tether cable from parabolic features

This chapter presents our first contribution, which is a new visual servoing approach that controls the deformation of a suspended tether cable subject to gravity from visual data provided by a RGB-D camera. The cable shape is modelled with a parabolic curve together with the orientation of the plane containing the cable. The visual features considered are the parabolic coefficients and the yaw angle of that plane. The analytical formulation of the interaction matrix that relates the variation of the visual features to the velocities of the cable extremities is derived. Singularities are demonstrated to occur if and only if the cable is taut horizontally or vertically. An image processing algorithm is proposed to extract in real-time the current features by fitting the parabola to the observed point cloud of the cable. Simulations and experimental results demonstrate the efficiency of the proposed visual servoing approach to deform the tether cable toward a desired shape configuration.

The chapter is organised as follows: Section 3.1 describes the modelling to control the shape of the tether cable. Section 3.2 gives a stability analysis of the proposed control scheme. Section 3.3 details the image processing developed to extract the visual features from the RGB-D data. Simulation and experimental results obtained with a robotic manipulator are presented in Section 3.4 to validate our cable shape servoing approach. Section 3.5 concludes this chapter by summarising its contributions.

3.1 Polynomial visual servoing

The cable can be slack or taut, so we propose to generalise this possible set of configurations in one model: a 2^{nd} order polynomial curve that is a parabola whose general form allows representing also a straight line. Simulations and experimental results will demonstrate that such an approximation is sufficient. The shape control approach elab-

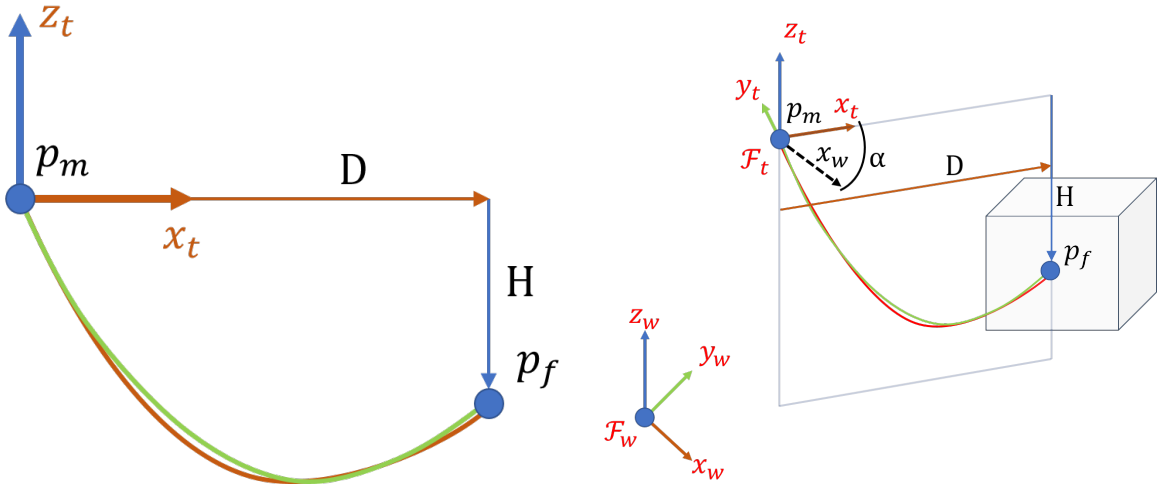
orated in this chapter only uses the length of the cable as *a priori* knowledge and the visual information provided by a RGB-D camera that observes the cable. Moreover, it is capable of operating with slack and taut configurations and does not need to observe the cable extremities. In particular, the chain model considered in [Laranjeira et al., 2020] and in [D'Antonio et al., 2021] does not allow to consider any taut configurations due to a singularity of the corresponding shape Jacobian, while our parabola model is not singular in this configuration apart when the cable is taut horizontally or vertically. An analytic expression of the shape Jacobian is derived for this model, thus avoiding the limitations of the model-free methods mentioned in Chapter 2. Another advantage of our solution is that it does not require any mechanical model parameters of the DLO and thus does not need any knowledge about its material mechanical properties. In contrast, our method relies only on the model described in Section 2.2.2 that is extracted and tracked from RGB-D data.

3.1.1 Tether cable Modelling

Let us start by defining some Cartesian frames and 3D points necessary to model the tether shape. The coordinate frames are shown in Figure 3.1. Note that both the tether frame $\mathcal{F}_t$ and the world frame $\mathcal{F}_w$ are defined so that their z axis is in the opposite direction of gravity. The points p_m and p_f represent the tether attachment points and p_m is selected as the origin of $\mathcal{F}_t$. To be as general as possible, in this section, it is considered that both the 3D positions of p_m and p_f can be controlled.

Since the system has 3 degrees of freedom (that are the 3 components of the relative position between p_m and p_f), 3 independent parameters are sufficient to control it. Note that a trivial solution would be to select as visual features the relative position between p_m and p_f. However, these two points may not be visible. That is why the proposed method does not necessitate measuring their 3D coordinates. As already said, a parabola is used to model the shape of the cable. Its equation in the plane (x_t, z_t) to which it belongs is given by $z = ax^2 + bx$. Thus, as first two visual features, we use the parameters a and b that characterise the shape of this parabola. In order to control the orientation of the cable with respect to the world frame, similarly to [Laranjeira et al., 2020], we consider as a third feature the yaw angle of the vertical plane containing the cable, *i.e.*, the angle α between x_t and x_w (note that $\sin\alpha$ was used in [Laranjeira et al., 2020], which induces a singularity when $\alpha = \pm\pi/2$).

In frame $\mathcal{F}_t$ the coordinates $\mathbf{p}_m$ and $\mathbf{p}_f$ are respectively given by $(0, 0, 0)$ and $(D, 0, H)$



(a) Side view (b) Isometric view

Figure 3.1 – Tether cable (green curve) and its parabola model (red curve): $D > 0$, $H < 0$ in this configuration.

and any point belonging to the parabola that represents the tether has as coordinates:

$$ {}^t\mathbf{p} = \begin{pmatrix} {}^t x \\ 0 \\ a\ {}^t x^2 + b\ {}^t x \end{pmatrix} \tag{3.1} $$

where $H = aD^2 + bD$. Since $\mathcal{F}_t$ and $\mathcal{F}_w$ have been defined with parallel $\mathbf{z}$ axis, their relative pose is given by the homogeneous transformation matrix (c stands for cos, s for sin):

$$ {}^w\mathbf{H}_t = \begin{bmatrix} c\alpha & -s\alpha & 0 & {}^w p_{m_x} \\ s\alpha & c\alpha & 0 & {}^w p_{m_y} \\ 0 & 0 & 1 & {}^w p_{m_z} \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.2} $$

This allows expressing any tether point in $\mathcal{F}_w$ by:

$$ {}^w\mathbf{p} = {}^w\mathbf{p}_m + \begin{pmatrix} {}^t x\ c\alpha \\ {}^t x\ s\alpha \\ a\ {}^t x^2 + b\ {}^t x \end{pmatrix} \tag{3.3} $$

In particular, the coordinates of the point p_f in $\mathcal{F}_w$ is given by:

$$ {}^w\mathbf{p}_f = {}^w\mathbf{p}_m + \begin{pmatrix} D\ c\alpha \\ D\ s\alpha \\ aD^2 + bD \end{pmatrix} \tag{3.4} $$

3.1.2 Interaction matrix

As already said, the set $\mathbf{s}$ of considered visual features is:

$$ \mathbf{s} = \begin{pmatrix} a \\ b \\ \alpha \end{pmatrix} \tag{3.5} $$

The aim of the following development is to determine the variation of $\mathbf{s}$ in function of the velocities of ${}^w\mathbf{p}_m$ and ${}^w\mathbf{p}_f$ in $\mathcal{F}_w$ for obtaining the analytical form of the interaction matrix of $\mathbf{s}$. By differentiating (3.4) we obtain:

$$ {}^w\dot{\mathbf{p}}_f = {}^w\dot{\mathbf{p}}_m + \begin{pmatrix} \dot{D}c\alpha - s\alpha D\dot{\alpha} \\ \dot{D}s\alpha + c\alpha D\dot{\alpha} \\ (2aD + b)\dot{D} + D^2\dot{a} + D\dot{b} \end{pmatrix} \tag{3.6} $$

This equation can be rewritten as:

$$ {}^w\dot{\mathbf{p}}_f - {}^w\dot{\mathbf{p}}_m = \begin{bmatrix} 0 & 0 & c\alpha & -Ds\alpha \\ 0 & 0 & s\alpha & Dc\alpha \\ D^2 & D & 2aD + b & 0 \end{bmatrix} \begin{pmatrix} \dot{a} \\ \dot{b} \\ \dot{D} \\ \dot{\alpha} \end{pmatrix} \tag{3.7} $$

As D depends on a and b, it is necessary to remove $\dot{D}$ from (3.7) to obtain a one-to-one relationship between $\dot{\mathbf{s}}$ and ${}^w\dot{\mathbf{p}}_f - {}^w\dot{\mathbf{p}}_m$. In frame $\mathcal{F}_t$, D is related to a, b and the length of the tether L by the following equation:

$$ L = \int_0^D \sqrt{1 + (2ax + b)^2}dx \tag{3.8} $$

To solve this integral analytically, let us choose $t = 2ax + b$, from which $dt = 2adx$ and we obtain:

$$L = \frac{1}{2a}\int_{t_1}^{t_2}\sqrt{1+t^2}dt \tag{3.9}$$

where $t_1 = b$ and $t_2 = 2aD + b$. Notice that this equation is not valid when $a = 0$ (that is, when the parabola degenerates to a straight line, in which case the tether is taut) for which:

$$L = D\sqrt{1+b^2} = \sqrt{D^2+H^2} \tag{3.10}$$

Since it is well known that:

$$\int\sqrt{1+t^2}dt = \frac{1}{2}t\sqrt{1+t^2} + \frac{1}{2}\ln\left|t+\sqrt{1+t^2}\right| \tag{3.11}$$

and

$$\ln\left|t+\sqrt{1+t^2}\right| = \operatorname{arcsinh} t, t \in \mathbb{R} \tag{3.12}$$

the expression of the tether length becomes as follows:

$$\begin{aligned} L &= f(D,a,b) \\ &= \frac{(2aD+b)\sqrt{1+(2aD+b)^2} - b\sqrt{1+b^2} + \operatorname{arcsinh}(2aD+b) - \operatorname{arcsinh} b}{4a} \end{aligned} \tag{3.13}$$

Since $\dot{L} = 0$, by differentiating (3.13) we get:

$$\frac{\partial f}{\partial D}\dot{D} = -\frac{\partial f}{\partial a}\dot{a} - \frac{\partial f}{\partial b}\dot{b} \tag{3.14}$$

from which we obtain:

$$\dot{D} = k_1\dot{a} + k_2\dot{b} \tag{3.15}$$

where

$$k_1 = 4(L - D\sqrt{1+t_2^2})/k \tag{3.16}$$

$$k_2 = 2(\sqrt{1+b^2} - \sqrt{1+t_2^2})/k \tag{3.17}$$

with

$$k = 4a\sqrt{1+t_2^2} \tag{3.18}$$

Note that $k = 0$ when $a = 0$ (*i.e.,* the tether is taut). Fortunately, after applying a second-order Taylor expansion to the term $f(a) = \sqrt{1+t_2^2} = \sqrt{1+(2Da+b)^2}$, we obtain

$$\lim_{a\to 0} k_1 = \frac{-2D^2 b}{1+b^2} \tag{3.19}$$

$$\lim_{a\to 0} k_2 = \frac{-Db}{1+b^2} \tag{3.20}$$

It is thus always possible to compute the terms k_1 and k_2 whatever the tether configuration. Then by injecting (3.15) into (3.7) the following relation is obtained:

$${}^w\dot{\mathbf{p}}_f - {}^w\dot{\mathbf{p}}_m = \mathbf{M}\dot{\mathbf{s}} \tag{3.21}$$

with

$$\mathbf{M} = \begin{bmatrix} k_1 c\alpha & k_2 c\alpha & -Ds\alpha \\ k_1 s\alpha & k_2 s\alpha & Dc\alpha \\ n_1 & n_2 & 0 \end{bmatrix} \tag{3.22}$$

where $n_1 = D^2 + t_2 k_1$ and $n_2 = D + t_2 k_2$. The inverse of $\mathbf{M}$ can be analytically calculated and is given by:

$$\mathbf{M}^{-1} = \frac{1}{Dq}\begin{bmatrix} -n_2 c\alpha & -n_2 s\alpha & k_2 \\ n_1 c\alpha & n_1 s\alpha & -k_1 \\ -qs\alpha & qc\alpha & 0 \end{bmatrix} \tag{3.23}$$

with $q = Dk_2 - k_1$. From (3.21), we finally obtain the relation:

$$\dot{\mathbf{s}} = \mathbf{M}^{-1}({}^w\dot{\mathbf{p}}_f - {}^w\dot{\mathbf{p}}_m) = \mathbf{L_s}\mathbf{v} \tag{3.24}$$

where $\mathbf{L_s} = \begin{bmatrix} -\mathbf{M}^{-1} & \mathbf{M}^{-1} \end{bmatrix}$ is the 3×6 shape interaction matrix that relates the variation of the visual features $\mathbf{s}$ to the translational velocity of p_m and p_f expressed in $\mathcal{F}_w$: $\mathbf{v} = (\mathbf{v}_m, \mathbf{v}_f) = ({}^w\dot{\mathbf{p}}_m, {}^w\dot{\mathbf{p}}_f)$

3.1.3 Control law

From the modelling presented in the previous section, we can immediately use the classical control law:

$$\mathbf{v} = -\lambda \mathbf{L}_\mathbf{s}^+(\mathbf{s} - \mathbf{s}^*) \tag{3.25}$$

for ensuring an exponential decrease of the visual feature error $\mathbf{e} = \mathbf{s} - \mathbf{s}^*$ toward zero [Chaumette, 2007]. Here $\mathbf{s}^*$ denotes the desired value of the visual features, $\lambda(> 0)$ is the control gain, and $\mathbf{L}_\mathbf{s}^+$ is the Moore-Penrose pseudoinverse of $\mathbf{L_s}$. In case point p_f is static (which is the case for our first experiment involving a robotic arm), the controller reduces to:

$$\mathbf{v}_m = \lambda \mathbf{M}(\mathbf{s} - \mathbf{s}^*) \tag{3.26}$$

Note that apart the singular configurations exhibited in Section 3.1.4, $\mathbf{M}$ is always of full rank 3, which ensures that the system is globally asymptotically stable in case $\mathbf{M}$ is correctly evaluated in (3.25) [Chaumette, 2007].

As can be seen from (3.22), matrix $\mathbf{M}$ depends on the visual features a, b, α that are measured at each iteration of the control scheme, the tether length L that is constant and supposed to be known, and D that is varying. Therefore, we need to estimate D. There are two ways for that: the first would be to track the attachment points p_f and p_m and measure their 3D position. As already said, this would not be convenient because the camera would need to have both points in its field of view, which would reduce the range of possible applications. The second method relies on estimating D from (3.13). Since there is unfortunately no explicit form for the solution of this equation (except when $H = 0$, then $D = \frac{b}{a}$ for a sagging cable), search for the value of D that minimises the

square error ε between the known length L and its expression (3.13):

$$\varepsilon = \left(4aL - t_2\sqrt{1+t_2^2} + b\sqrt{1+b^2} - \operatorname{arcsinh} t_2 + \operatorname{arcsinh} b\right)^2 \tag{3.27}$$

We determine D by numerically solving the following objective function:

$$\underset{D\in\mathbb{R}}{\operatorname{argmin}}\, \varepsilon \tag{3.28}$$

with a Gauss-Newton method detailed in Algorithm 1 using the derivatives of (3.27), (see Appendix B). In practice, only a couple of iterations are needed to estimate the new value of D at each iteration of the control loop by using its previous value as an initial guess. We used $D_{prev} = L/2$ for the very first image. The simulation will show in Section 3.4 that our method is also robust in the case of $D = \widehat{D}$ with $\widehat{D}$ being an approximation of D defined from the stability conditions of the control law derived in the following sections. The value $D = \frac{L}{1+b^2}$ is also considered in Algorithm 1 when a approaches zero in order

$D = D_{prev}$
$\varepsilon = 1e6$
$\epsilon = 1e-6$
if $a > 0.05$ **then**
  while $\varepsilon > \epsilon$ *and* $i < 10$ **do**
    $D = D - \frac{\mathbf{J}_{\mathbf{s},D}}{\mathbf{H}_{\mathbf{s},D}}$ (see Appendix B)
    $\varepsilon = \left(4aL - t_2\sqrt{1+t_2^2} + b\sqrt{1+b^2} - \operatorname{arcsinh} t_2 + \operatorname{arcsinh} b\right)^2$
    $++\, i$
  end
  return D
else
  return $D = \frac{L}{1+b^2}$

Algorithm 1: A procedure used for the estimation of parameter D

to avoid local minima of ε. Figure 3.2 illustrates the evolution of the square error ε in function of D for different configurations of the parameters a, b and L of the cable. We can note that the ε function shows a unique zero that can be easily obtained when a is not too low.

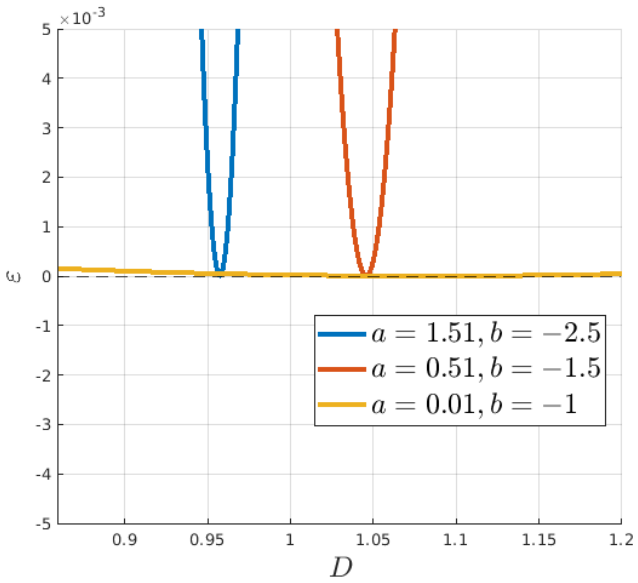


Figure 3.2 – Evolution of the square error ε (3.27) in function of D for different values of a, b. The length of the cable is fixed to $L = 1.52m$. The determination of D is obtained when ε reaches the zero value. Note the increasing span of the curves when a tends to zero.

3.1.4 Study of the singularities of $\mathbf{L_s}$

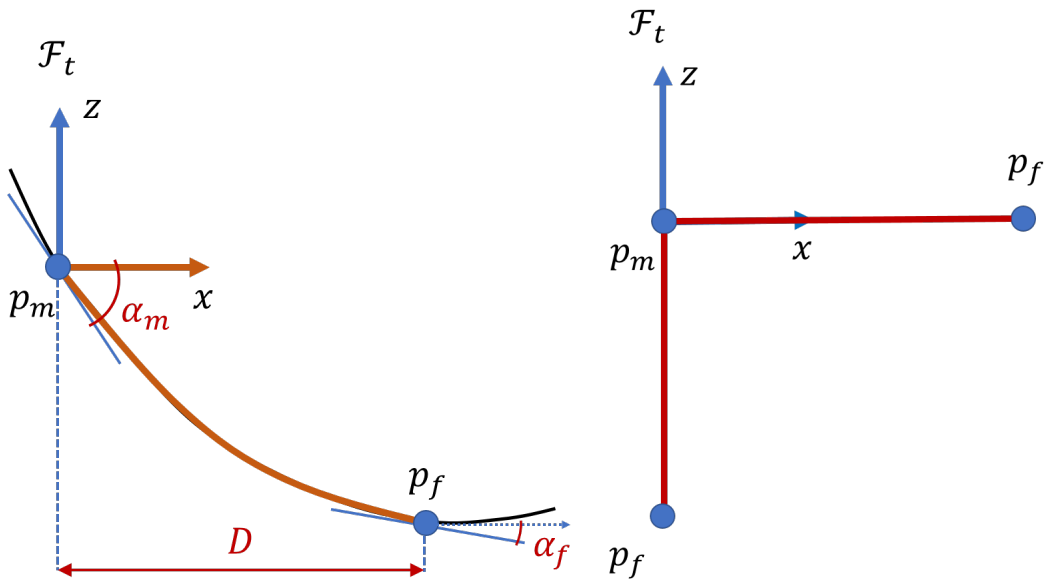
In the following, a study on the possible singularities that the interaction matrix $\mathbf{L_s}$ may encounter is developed. The determinant of $\mathbf{M}$ is given by:

$$\det \mathbf{M} = D^2 q = D^2(Dk_2 - k_1) \tag{3.29}$$

$\mathbf{M}^{-1}$ and thus $\mathbf{L_s}$ are singular when D is null, meaning that points p_f and p_m coincide or lay on the same vertical. This is logical since the angle α is no longer defined in that degenerate case. Another singularity is reached for the condition $Dk_2 - k_1 = 0$. Using (3.16) and (3.17) this condition can be rewritten as:

$$D(\sqrt{1 + \tan^2 \alpha_f} + \sqrt{1 + \tan^2 \alpha_m}) = 2L \tag{3.30}$$

with $\tan \alpha_f = t_2 = 2Da + b$ and $\tan \alpha_m = b$, where α_f and α_m are respectively the angles of the tangent of the parabola at point p_m and p_f (see Figure 3.3(a)).



(a) Tangent of the parabola at points p_m and p_f.

(b) Singular configurations of the model.

Figure 3.3 – Planar view of the cable and its two singular configurations (b) that generate singularities of the interaction matrix $\mathbf{L_s}$.

Since $1 + \tan^2 \alpha = \frac{1}{\cos^2 \alpha}$, (3.30) is equivalent to

$$\frac{D}{|\cos \alpha_f|} + \frac{D}{|\cos \alpha_m|} = 2L \tag{3.31}$$

then, by combining (3.31) and (3.13), and using $\tan \alpha_f - \tan \alpha_m = 2Da$, we obtain:

$$\frac{\tan \alpha_f}{|\cos \alpha_m|} - \frac{\tan \alpha_m}{|\cos \alpha_f|} = \operatorname{arcsinh}\left(\frac{\tan \alpha_f}{|\cos \alpha_m|} - \frac{\tan \alpha_m}{|\cos \alpha_f|}\right) \tag{3.32}$$

The singularity condition is given by the solution of (3.32), which exists in $\mathbb{R}$ if and only if

$$\frac{\tan \alpha_f}{|\cos \alpha_m|} = \frac{\tan \alpha_m}{|\cos \alpha_f|} \tag{3.33}$$

Thus, the singularity occurs when $\alpha_m = \alpha_f$, meaning that the cable is taut. However, when the cable is taut, it is directly obtained from (3.19) and (3.20) that $q = Dk_2 - k_1 = D^2 b/(1 + b^2)$. Apart from the configuration exhibited previously $D = 0$, q vanishes only when $b = 0$, that is, when the tether is horizontally taut.

This analysis demonstrates only two singular configurations: when p_f and p_m coincide, and when the parabola degenerates into a vertical or horizontal straight line, as can be seen in Figure 3.3(b). The former configuration corresponds to a pure degenerate case that cannot occur in practice. The latter can be avoided by never specifying such a desired configuration.

Note that in the case where only one cable extremity is manipulated (the other being fixed), the interaction matrix is $\mathbf{L_s} = \pm\mathbf{M}^{-1}$ and $\mathbf{M}$ gives directly the analytical form (3.22) of its inverse.

3.2 Stability Analysis

Let us consider the case where only one cable extremity is manipulated. We know that the closed-loop dynamics of the visual error is globally asymptotically stable in the case of a not too coarse estimation of the features and parameters involved in $\mathbf{M}$ thanks to [Chaumette, 2007]. It would be interesting to study the stability condition for an arbitrary value of D in order to keep it constant during the control task and avoid its estimation. Let us start by introducing the notation $\widehat{D}$ as the estimate of the true values of D. All values related to $\widehat{D}$ are marked in a similar way, $\widehat{k_1}, \widehat{k_2}, \widehat{n_1}, \widehat{n_2}$. Consider the following Lyapunov function:

$$\mathcal{L} = \frac{1}{2}||\mathbf{e}(t)||^2 \tag{3.34}$$

Its derivative is given by

$$\begin{aligned} \dot{\mathcal{L}} &= \mathbf{e}^T\dot{\mathbf{e}} \\ &= -\lambda\mathbf{e}^T\mathbf{L}_s\widehat{\mathbf{L}_s}^{-1}\mathbf{e} \\ &= -\lambda\mathbf{e}^T\mathbf{M}^{-1}\widehat{\mathbf{M}}\mathbf{e} \end{aligned} \tag{3.35}$$

where $\widehat{\mathbf{L}_s}^{-1} = -\widehat{\mathbf{M}}$ is the inverse of the estimated interaction matrix. To guarantee global asymptotic stability, we must ensure $\dot{\mathcal{L}} < 0$, which is equivalent to ensure the following condition:

$$\mathbf{S} = \mathbf{L}_s\widehat{\mathbf{L}_s}^{+} = \mathbf{M}^{-1}\widehat{\mathbf{M}} > 0 \tag{3.36}$$

To conduct the study on the conditions of $\widehat{D}$ that guarantee the global asymptotic stability, we consider that the parameters a, b, α of the system are perfectly estimated. In this case, the product $\mathbf{S} = \mathbf{M}^{-1}\widehat{\mathbf{M}}$ can be written in a compact form and its symmetric part is given by:

$$\frac{1}{2}\left(\mathbf{S}+\mathbf{S}^T\right) = \begin{bmatrix} \frac{k_2\widehat{n_1}-\widehat{k_1}n_2}{Dq} & -\frac{k_1\widehat{n_1}}{2Dq}+\frac{k_2\widehat{n_2}}{2Dq}+\frac{\widehat{k_1}n_1}{2Dq}-\frac{\widehat{k_2}n_2}{2Dq} & 0 \\ -\frac{k_1\widehat{n_1}}{2Dq}+\frac{k_2\widehat{n_2}}{2Dq}+\frac{\widehat{k_1}n_1}{2Dq}-\frac{\widehat{k_2}n_2}{2Dq} & \frac{-k_1\widehat{n_2}+\widehat{k_2}n_1}{Dq} & 0 \\ 0 & 0 & \frac{\widehat{D}}{D} \end{bmatrix} \tag{3.37}$$

The eigenvalues of (3.37) are given by:

$$\begin{aligned} \lambda_1 &= \frac{-k_1\widehat{n_2}+k_2\widehat{n_1}+\widehat{k_2}n_1-\widehat{k_1}n_2+\sqrt{\delta}}{2Dq} \\ \lambda_2 &= \frac{-k_1\widehat{n_2}+k_2\widehat{n_1}+\widehat{k_2}n_1-\widehat{k_1}n_2-\sqrt{\delta}}{2Dq} \\ \lambda_3 &= \frac{\widehat{D}}{D} \end{aligned} \tag{3.38}$$

where

$$\begin{aligned} \delta &= \left(n_1^2+n_2^2\right)\widehat{k_1}^2+\left(\left(-2k_1n_1-2k_2n_2\right)\widehat{n_1}-2\widehat{n_2}\left(k_1n_2-k_2n_1\right)\right)\widehat{k_1} \\ &+ \left(n_1^2+n_2^2\right)\widehat{k_2}^2+\left(\left(2k_1n_2-2k_2n_1\right)\widehat{n_1}-2\widehat{n_2}\left(k_1n_1+k_2n_2\right)\right)\widehat{k_2} \\ &+ \left(\widehat{n_1}^2+\widehat{n_2}^2\right)\left(k_1^2+k_2^2\right) \end{aligned} \tag{3.39}$$

Since the matrix (3.37) is positive if all its eigenvalues are positive, our analysis will now focus on the conditions that guarantee their positivity. Firstly, λ_3 is positive under the simple condition that $\widehat{D}$ remains positive regardless of the value of $\widehat{L}$, which means that $\widehat{D}$ and $\widehat{a}, \widehat{b}, \widehat{L}$ do not disturb the convergence of the angle α. Secondly, after developing the analytical form of $\lambda_i = f(e_{\widehat{D}}, a, b, L)$ without considering the error on L and by posing $e_{\widehat{D}} = D - \widehat{D}$ we can study the evolution of each eigenvalue during the control task as a function of the parameter estimation error $e_{\widehat{D}}$.

The stability of the control law will be studied in Section 3.4.1 through simulations for desired cable shapes with a constant value $\widehat{D}$ chosen as $\widehat{D} = L$ or $\widehat{D} = \frac{L}{\sqrt{1+b^{*2}}}$.

3.3 Features extraction

In practice, the cable shape is estimated from visual data provided by a RGB-D camera. In particular, its RGB image together with the depth are used to fit a parabola into the observed point cloud of the cable. The image processing pipeline consists in first applying an HSV threshold on the RGB image to segment the cable in the image from its color (yellow in our case). A thinning algorithm [Zhang & Suen, 1984] is then applied for determining the middle line of the cable in the image. The 3D coordinates of the N points belonging to this middle line are first expressed in the camera frame $\mathcal{F}_c$ thanks to the depth data (see Figure 3.4) and then in the world frame $\mathcal{F}_w$ thanks to the knowledge of the transformation matrix ${}^w\mathbf{H}_c$ (determined offline using a classical hand-eye calibration technique [R. Tsai, 1987]). Using these 3D points ${}^w\mathbf{p}_1 \dots {}^w\mathbf{p}_N$, the parameters ${}^w\mathbf{n}$ and ${}^wd$ of the cable plane equation ${}^w\mathbf{n}^T\ {}^w\mathbf{p} + {}^wd = 0$ in $\mathcal{F}_w$ are estimated by using a robust least square method [Comport et al., 2004] for eliminating potential outliers due to depth measurement errors. Notice that the z component of normal ${}^w\mathbf{n}$ is imposed to be ${}^wn_z = 0$ during the estimation process, as the plane is always vertical in $\mathcal{F}_w$ due to gravity. The angle α is then directly given by:

$$\alpha = \arctan(-{}^wn_x/{}^wn_y) \tag{3.40}$$

Afterwards, we use the cable plane equation expressed in the camera frame $\mathcal{F}_c$: ${}^c\mathbf{n}^T{}^c\mathbf{p} + {}^cd = 0$ to reconstruct the 3D coordinates ${}^c\mathbf{p}_i$ of each point belonging to the middle line of the cable from its observed image coordinates $\mathbf{p}_{im} = (\tilde{x}_i, \tilde{y}_i, 1)$. Indeed, since $\tilde{x}_i = x_i/z_i$ and $\tilde{y}_i = y_i/z_i$ due to perspective projection, we have the following:

$${}^c\mathbf{p}_i = (\tilde{x}_i z_i, \tilde{y}_i z_i, z_i) \tag{3.41}$$

with

$$z_i = -\frac{{}^cd}{{}^c\mathbf{n}^T\mathbf{p}_{im}} \tag{3.42}$$

This process allows considering a new set of N 3D points that are less tainted by the sensor depth measurement noise. Note that ${}^c\mathbf{n}^T\mathbf{p}_{im} = 0$ in (3.42) if and only if the camera optical center belongs to the cable plane, which is a degenerate case we do not consider for now, but we will consider in Chapter 4 when manipulating cable with drones, one of which

being equipped with an on-board camera using a different processing pipeline. Finally, the parabolic features a and b are estimated using a classical least squares method from this set of N points expressed in $\mathcal{F}_t$. It is important to mention that the first phase of the algorithm based on the pointcloud rejects the outliers and, therefore, a standard least squares method is sufficient to obtain a, b. For estimating a and b, the points are expressed in $\mathcal{F}_t$ using the transformation ${}^t\mathbf{H}_c = {}^t\mathbf{H}_w {}^w\mathbf{H}_c$ where ${}^t\mathbf{H}_w$ is the inverse of the homogeneous transformation (3.2) that is computed using the estimated α and the 3D position of p_m in $\mathcal{F}_w$ provided by the robot odometry. Furthermore, a and b are obtained by solving $a\,{}^t x_i^2 + b\,{}^t x_i = {}^t z_i, i = 1, \ldots, N$ whose solution is given by:

$$\begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} {}^t x_1^2 & {}^t x_1 \\ \ldots & \ldots \\ {}^t x_N^2 & {}^t x_N \end{bmatrix}^+ \begin{bmatrix} {}^t z_1 \\ \ldots \\ {}^t z_N \end{bmatrix} \tag{3.43}$$

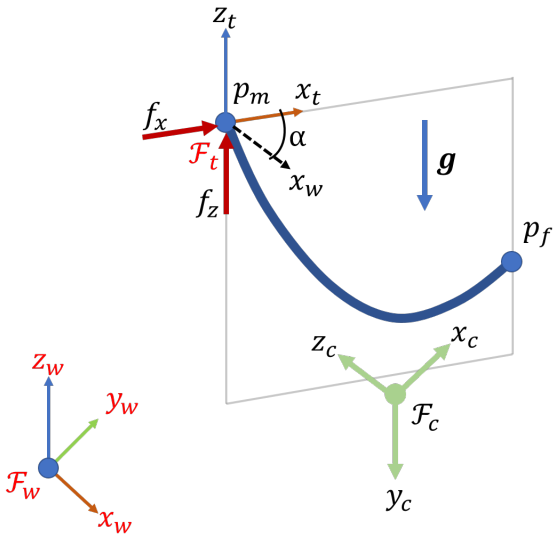


Figure 3.4 – The tether cable setup used for the simulations and the experiments. The external forces used in the simulation are depicted in red.

Note that at least two points are necessary to obtain a and b, but the more the better. Also, the regression matrix appearing in (3.43) is not invertible if and only if the cable is vertical.

3.4 Results

In a first part, we test the proposed control approach in a simulation framework by using a continuous mechanical model of the tether cable to simulate its deformation. In a second part, we present experimental results obtained with a real system involving a robotic arm and show that our method is robust to the approximation of the cable by a parabolic curve. Note that there is no difference between eye-in-hand and eye-to-hand setups since the pose of the camera frame $\mathcal{F}_c$ with respect to the world frame $\mathcal{F}_w$ is constant and known from hand-eye calibration for the eye-to-hand setup while varying pose is obtained from hand-eye calibration and robot odometry for the eye-in-hand setup.

3.4.1 Simulation framework

A simulation framework has been implemented to validate the control approach developed in Section 3.1. The overall pipeline of the simulation is presented by Algorithm 2 where dk is the simulation time step. To simulate realistic cable deformation, the shape of the tether is expressed following the extensible catenary model presented in [Yuan et al., 2015]. To use this model, arbitrary values for the Young's modulus E, the tether cross-section A and the tether mass per unit length ρ given in Table 3.1 have been set. These parameters correspond to the mechanical parameters of the cable that are required for the physics-based model [Yuan et al., 2015].

E, Pa	$0.05e9$
A, m^2	$1.26e-5$
$\rho, kg/m$	0.067
l_{us}, m	1.0

Table 3.1 – Mechanical parameters values used in the simulation

From the known position of the two tether extremities, the corresponding external force $\mathbf{f}$ with components $(f_x, 0, f_z)$ in $\mathcal{F}_t$ to be applied to the extremity point p_m is first estimated (see Figure 3.4). Then, the solution at static equilibrium of the forces applied to the tether, that is, the external force $\mathbf{f}$ and the gravity $\mathbf{g}$, allows determining the shape of the cable, from which a Point Cloud (PC) is generated. The coefficients a, b of the parabola are then determined using the least squares method described in Section 3.3 and a next iteration of the control loop can be achieved. ViSP [Marchand et al., 2005] is

used for the implementation of the overall simulation.

while $||\mathbf{e}_k|| > \epsilon$ **do**
  $\mathbf{e}_k = \mathbf{s}_k - \mathbf{s}^*$
  compute D from (3.28)
  compute $\mathbf{M}(\mathbf{s}_k, D, L)$ from (3.22)
  compute $\mathbf{v}$ from (3.25)
  compute ${}^w\mathbf{p}_{m(k+dk)}$,${}^w\mathbf{p}_{f(k+dk)}$ with the Euler step size fixed to $40ms$
  compute α from ${}^w\mathbf{p}_{m(k+dk)}$ and ${}^w\mathbf{p}_{f(k+dk)}$
  compute ${}^w\mathbf{H}_t$ from (3.2)
  $\mathbf{f}$ = simulator.estimateF(${}^t\mathbf{p}_m, {}^t\mathbf{p}_f, E, A, \rho$)
  cable PC = simulator.sampleCatenary($\mathbf{f}, E, A, \rho$)
  compute a, b from cable PC
  $\mathbf{s}_k = (a, b, \alpha)$
end

Algorithm 2: Simulation of the proposed visual servoing

The results of two shape servoing tasks are illustrated in Figure 3.5. This first simulation demonstrates a control law in which the position of one cable extremity is controlled. The first target shape is $\mathbf{s}^* = [0, -0.7, 0]$, which means that we want to reach a taut configuration, while the second one is such that $\mathbf{s}^* = [3.6, -1, -30^o]$, which corresponds to a general slack configuration. The initial configuration of the tether is set to $\mathbf{s}_0 = [1.2, -1.4, 75^o]$, see Figure 3.5(a),(b). It can be seen a perfect exponential decrease of the error in Figure 3.5(c),(d). The three translational velocities of p_m depicted in Figures 3.5(e)(f) also show a nice behaviour without any instabilities even when the tether becomes taut. These simulation results demonstrate that our control solution based on an approximation of the tether shape with a parabola can accurately perform the task, whatever the desired configuration of the cable.

The second simulation demonstrates the results of the same control law as in the previous scenario. In contrast, this time, the inverse of the interaction matrix, that is the matrix $\mathbf{M}$, is maintained with a constant value of the parameter $D = \widehat{D}$ at each iteration of the control law. The value of $\widehat{D}$ is chosen to be different depending on the type of task: bringing the cable to the taut shape with $\widehat{D} = \frac{L}{\sqrt{1+b^{*2}}}$ or to the slack shape with $\widehat{D} = L$ respectively (see Figure 3.6(b),(a)). Both values are computed from the length of the cable L or its desired shape, which is known beforehand. After several simulations we concluded that the value of $\widehat{D}$ needs to be close to the value of D obtained at the end of the task when the desired cable shape is a taut cable. In contrast, for the

considered slack cable target shapes the value of $\widehat{D}$ can vary in a larger interval whose size depends on the initial and the target cable shape. It can be seen in Figure 3.6 the convergence of both control laws. Although we cannot provide theoretical guarantees for all possible initial and target shapes, it is clear from Figure 3.6(a),(b) that the system converges to the two arbitrary chosen target shapes from an arbitrary chosen initial shape. The target velocity profiles in Figure 3.6(c),(d) are also rather smooth compared to the perfect case. The evolution of the eigenvalues (3.39) during the control task is illustrated in Figure 3.7(a),(b) for taut and slack target shapes, respectively. It can be seen that all the eigenvalues remain positive during both tasks, which confirms the stability of the control law. As expected, the third eigenvalue is hardly influenced by $\widehat{D}$, which is not the case for the first two. This demonstrates that in some region around the initial shape of the cable, the control law converges even with a wrong estimate of the parameter D to some desired shape that respects the stability condition given by (3.37) throughout the control task. Obviously, the convergence rate is not exponential, but in the case of the second scenario (bringing the cable to the slack configuration), the rate can be even faster than the exponential, as can be seen in Figure 3.6(b). In Figure 3.7 it can be seen the influence of the error on each eigenvalue during each control task. For the taut task, λ_1 is the most sensitive to the estimation error $e_{\widehat{D}} = D - \widehat{D}$, while it is λ_2 for the slack task.

The last simulation demonstrates a cable shaping task by controlling both attachment points (see Figure 3.8). The initial cable shape is chosen to be the same as in the previous simulations, as well as the target shape. This time, the system is redundant with respect to the cable shaping task, using the six available DOFs to control the three visual features of the cable. It can be seen in Figure 3.8(c),(d) a perfect exponential decrease of the shape error as it was the case in the first simulation. The velocity profiles shown in Figure 3.8(e),(f) represent smooth symmetric curves for $\mathbf{v}_m$ and $\mathbf{v}_f$ which reveals a well-conditioned Moore-Penrose pseudoinverse of the interaction matrix $\mathbf{L_s}^+ = \begin{bmatrix} -\mathbf{M}^{-1} & \mathbf{M}^{-1} \end{bmatrix}^+$ for both attachment points when the cable is taut and slack.

3.4.2 Experimental validation

The experimental setup is composed by an Omron© 6 DOF articulated robot Viper650 (see Figure 3.9(a)). The left extremity of a 86 centimeters length cable is attached to the robot end-effector and the right one is fixed to a tripod. The value of $\widehat{D}$ is estimated using Algorithm 1 at each iteration of the control law (3.26). A remote Intel© D435 RGB-D camera with a resolution of 1280x720 pixels for both data streams is mounted on a second

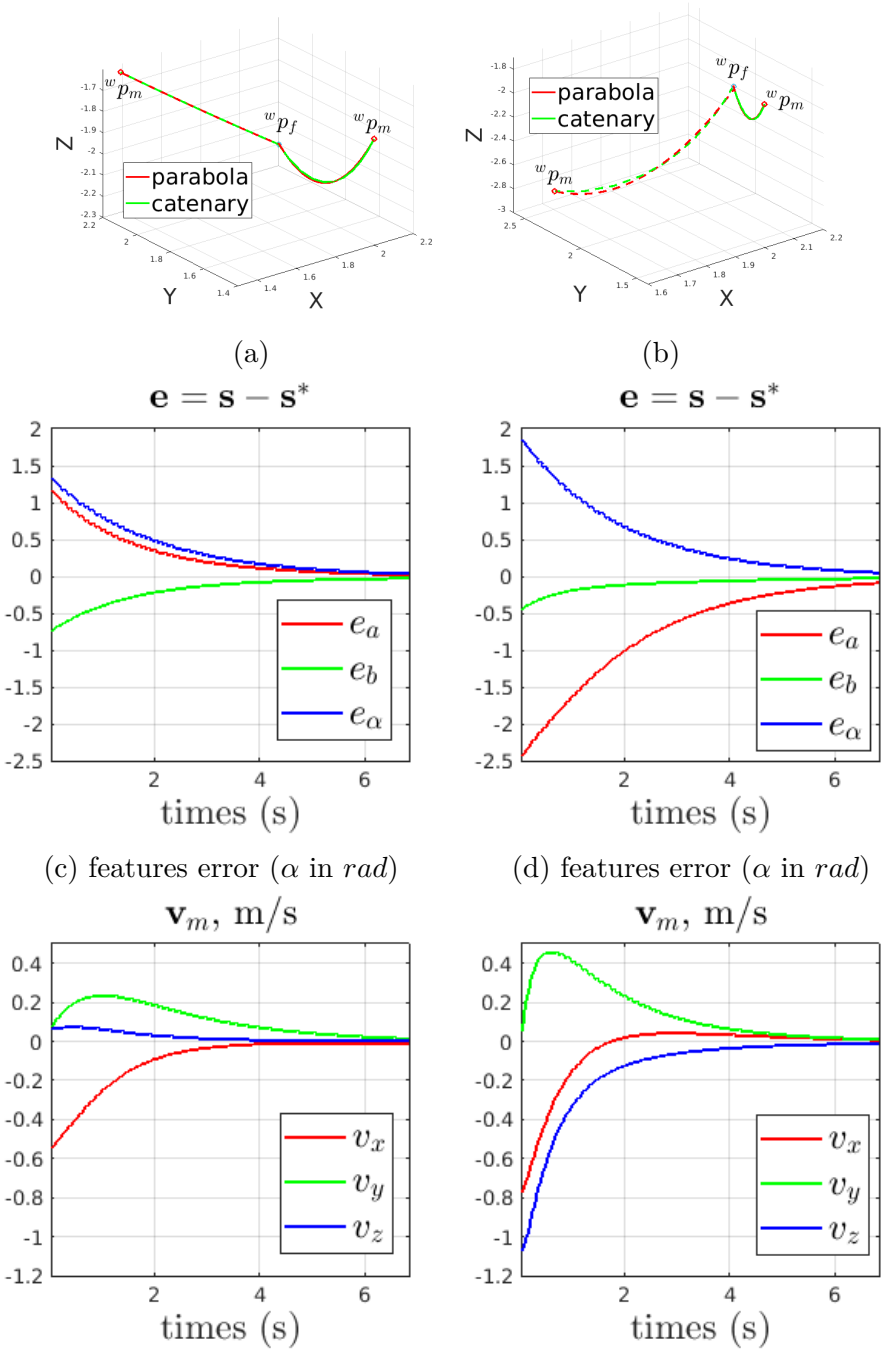
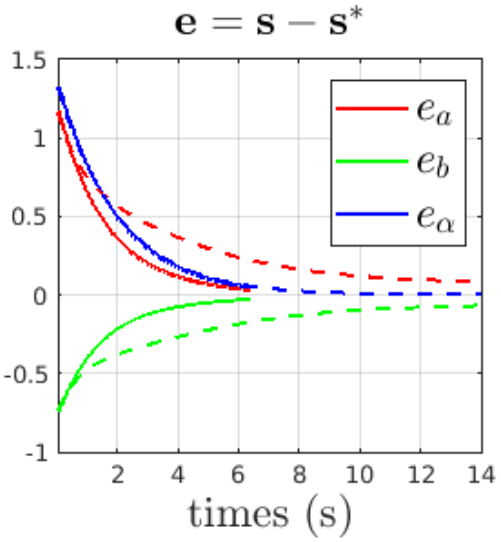
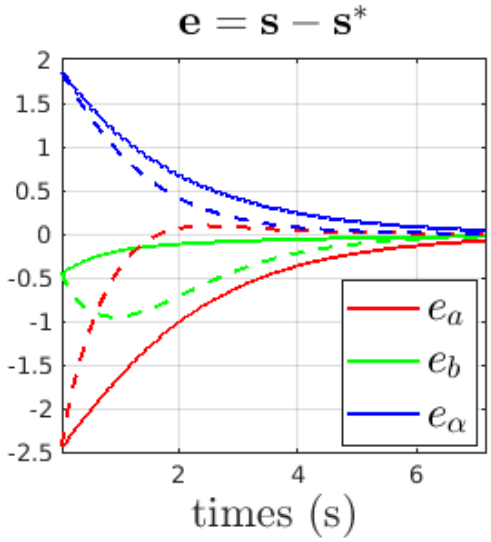


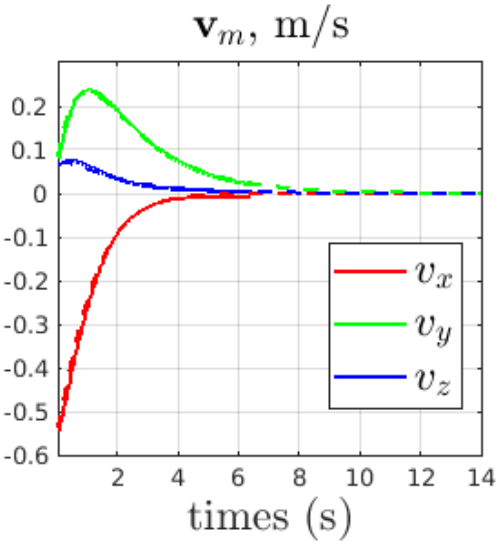
Figure 3.5 – Simulation results when the position of one cable extremity is controlled. Cable configuration at the beginning and at the end for the first task (a) and for the second task (b) (parabola in red and catenary in green, target parabola in dotted red). Evolution of the visual features error (c),(d) and velocity applied to point $\mathbf{p}_m$ (e),(f) during the first and the second tasks respectively. a in m^{-1} , b no units.



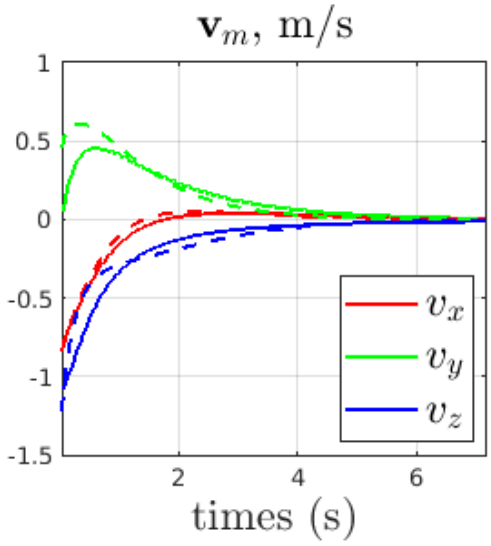
(a) features error (α in rad)



(b) features error (α in rad)



(c) velocity components in m/s



(d) velocity components in m/s

Figure 3.6 – Comparison of control law convergence when $\widehat{D} = D$ (solid line) vs $\widehat{D} = L$ (dotted line) (a),(c) for the first task and $\widehat{D} = \frac{L}{\sqrt{1+b^{*2}}}$ (dotted line) (b),(d) for the second task depicted on Figure 3.5(a),(b). Note that the convergence rate is faster in (b) and much slower in (a) compared to the same task.

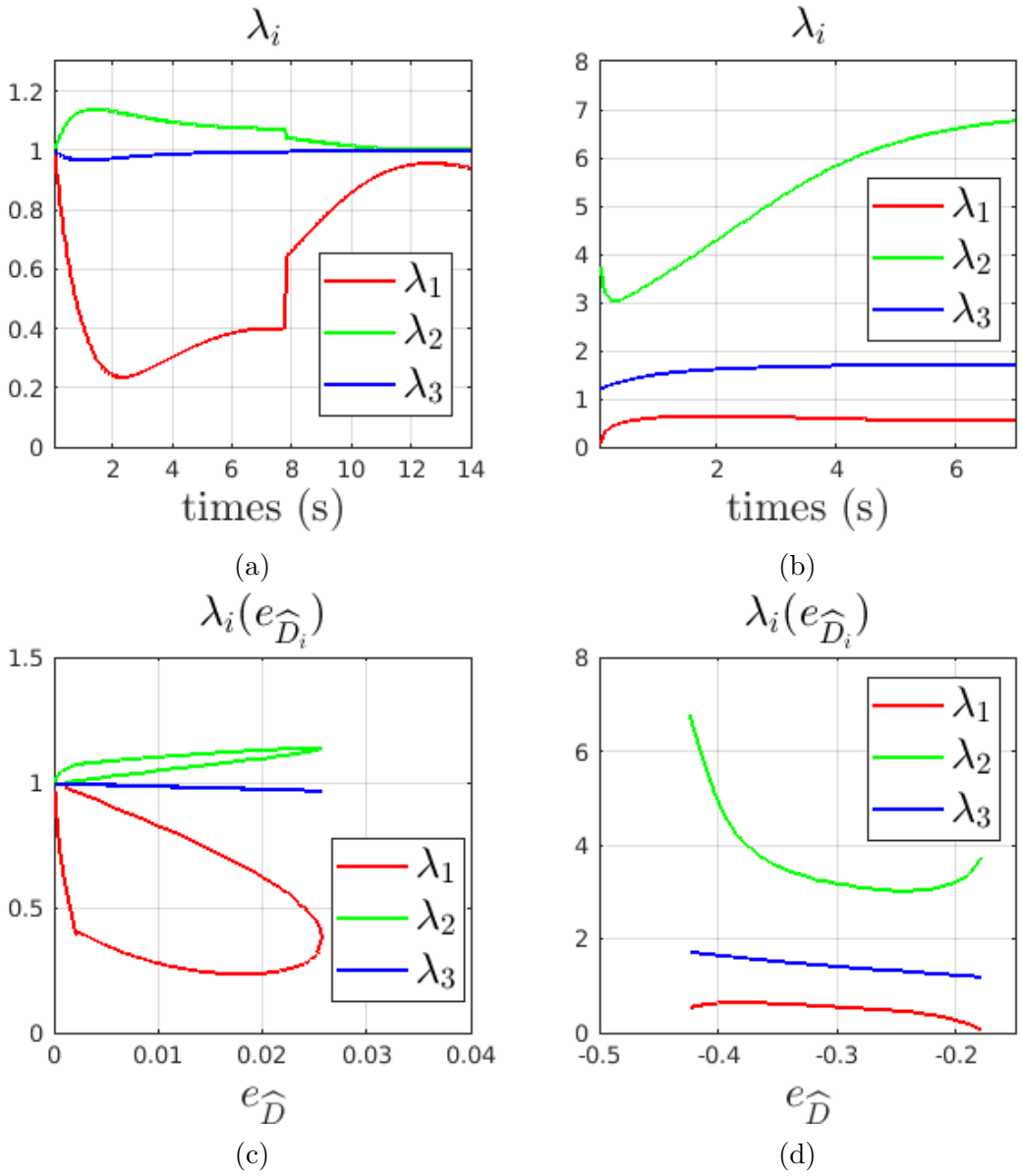


Figure 3.7 – Eigenvalues (3.38) during (a),(c) the taut cable target task with $\widehat{D} = \frac{L}{\sqrt{1+b^{*2}}}$ and (b),(d) the slack cable target task with $\widehat{D} = L$. Curves (a),(b) reveal the evolution of λ_i over time, while the last two curves (c),(d) are the evolution of $\lambda_i(e_{\widehat{D}})$ as a function of the estimation error of $\widehat{D}$ for the tasks shown in Figure 3.6. Note the sharp change in the eigenvalues (a) around time $8s$, (c) when $e_{\widehat{D}}$ is near zero for the taut cable task, which is explained by (3.19), (3.20) of the terms k_1, k_2 of $\mathbf{M}$ when the shape of the cable approaches a straight line. This change is beneficial for λ_1 as its value significantly increases.

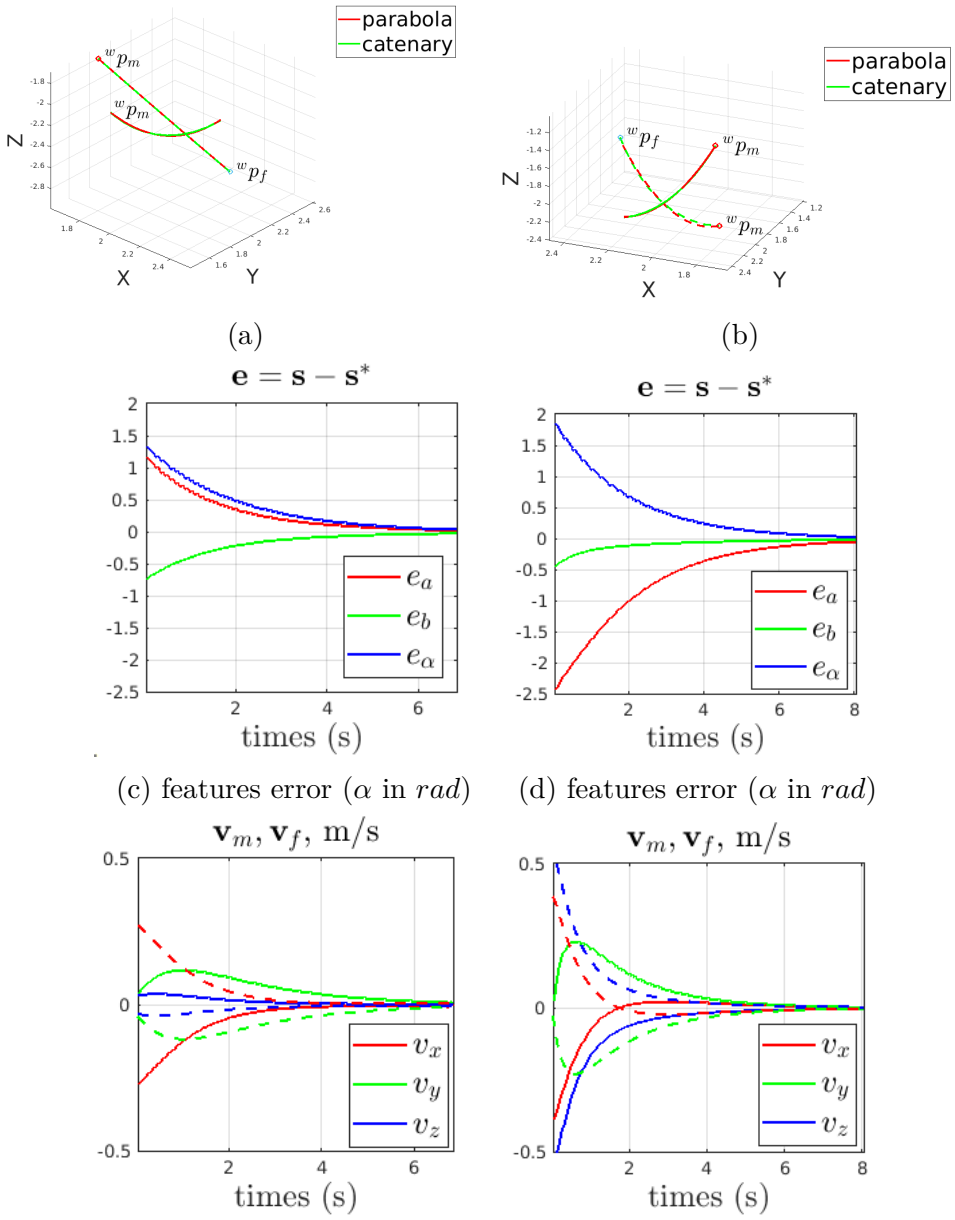


Figure 3.8 – Simulation results when controlling both extremities of the cable. Cable configuration at the beginning and at the end for the first task (a) and for the second task (b) (parabola in red and catenary in green, target parabola in dotted red). Evolution of the visual features error (c),(d) and velocity applied to points $\mathbf{p}_m$ (solid line) and $\mathbf{p}_f$ (dotted line) (e),(f) during the first and the second tasks respectively.

fixed tripod and observes the scene. An off-line calibration step was performed in order to estimate the transformation matrix between the robot base frame $\mathcal{F}_w$ and the RGB-D camera frame $\mathcal{F}_c$ ${}^{w}\mathbf{H}_{c}$. Note that the depth and RGB streams of the camera are expressed in the camera frame.

A video showing the simulation and experimental results is available at https://youtu.be/81n8LBIahXE. To validate our method two different tasks were tested: the first is to stretch the cable by setting $\mathbf{s}^* = [0, -0.4, -30^o]$ and the second is to move it back to a slack configuration with $\mathbf{s}^* = [1.5, -1.4, -15^o]$. The initial state is such that $\mathbf{s}_0 = [1, -1.1, -17^o]$. The robot control sampling frequency is set to the camera frame rate (30 Hz) and the control gain to $\lambda = 0.5$. Typical RGB and depth images acquired by the camera are presented in Figure 3.9. The temporal evolution of the feature error together with the control velocity applied at the point p_m are shown in Figure 3.10. One can see in Figure 3.10(a) and 3.10(b) the exponential decoupled decrease of the features error to zero, which demonstrates the efficiency of the proposed approach. Note in Figures 3.10(c) and 3.10(d) that the noise on the estimated value of α, which is due to the low accuracy of the depth measurements, induces some noise on the v_x and v_y components of the end-effector velocity, but without destabilising the system. These results show that even if our approach is based on coarse modelling of the tether cable by a parabolic curve, it can efficiently perform the shaping task thanks to the robustness of the shape servoing closed-loop control scheme. Both extremities can be covered in a random fashion with a sheet of paper to show that there is no need to track the entire cable during the shaping task (see Figure 3.9(c)).

3.5 Conclusion

In this chapter, we addressed the automatic manipulation of a suspended tether cable. We proposed three visual features from a simple parabolic geometric model to control the translational velocities of its extremities. The analytical form of the related interaction matrix has been determined, allowing to demonstrate the absence of singularities apart when the tether is taut vertically or horizontally. Robustness to the coarse estimate of the value of the parameter D was demonstrated for an arbitrary initial and two different target shapes but was not proved for all possible initial and target cable shapes. Perfect decoupling was demonstrated between the error dynamics for the error components e_a, e_b of the visual features and for the cable plane orientation error component e_α using Lya-

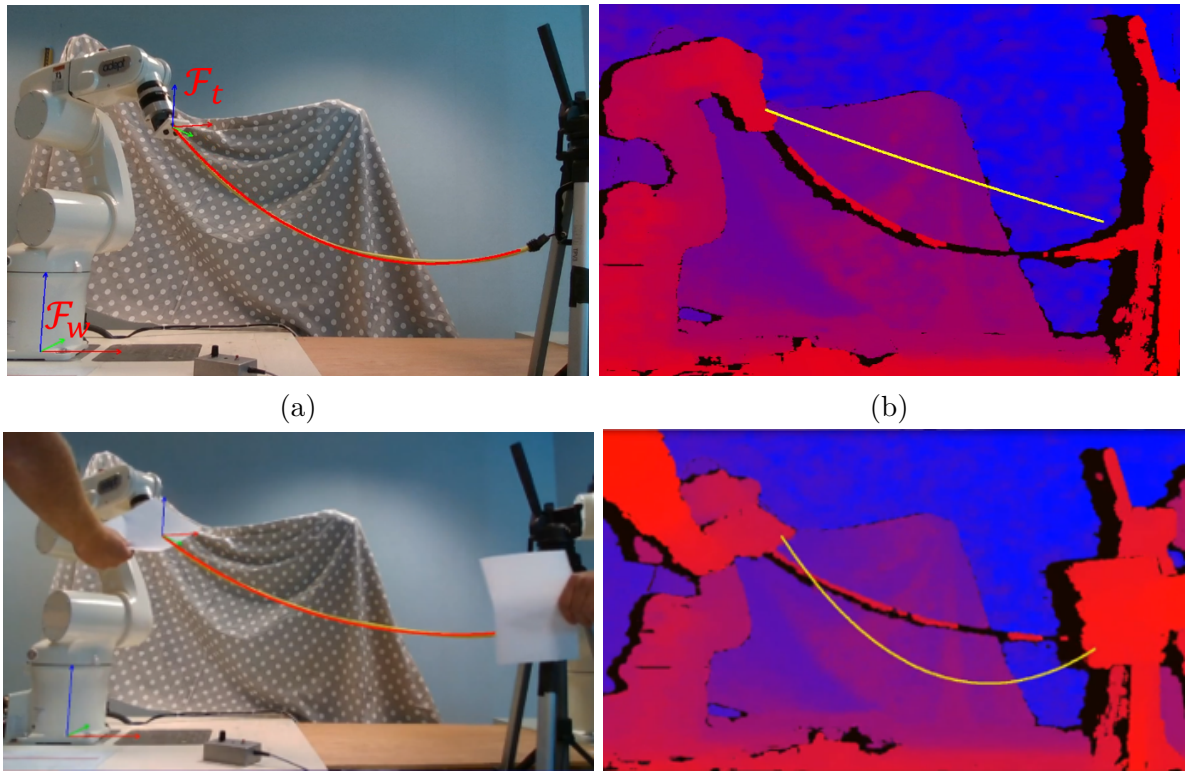


Figure 3.9 – (a) RGB image of the experiment during the first task. The fitted parabola, $\mathcal{F}_w$ and $\mathcal{F}_t$ frames are depicted in red. (b) Depth image with the first task target shape overlaid in yellow. (c) RGB image of the experiment during the task with occlusions of the extremities of the cable. (d) Depth image with target shape in yellow.

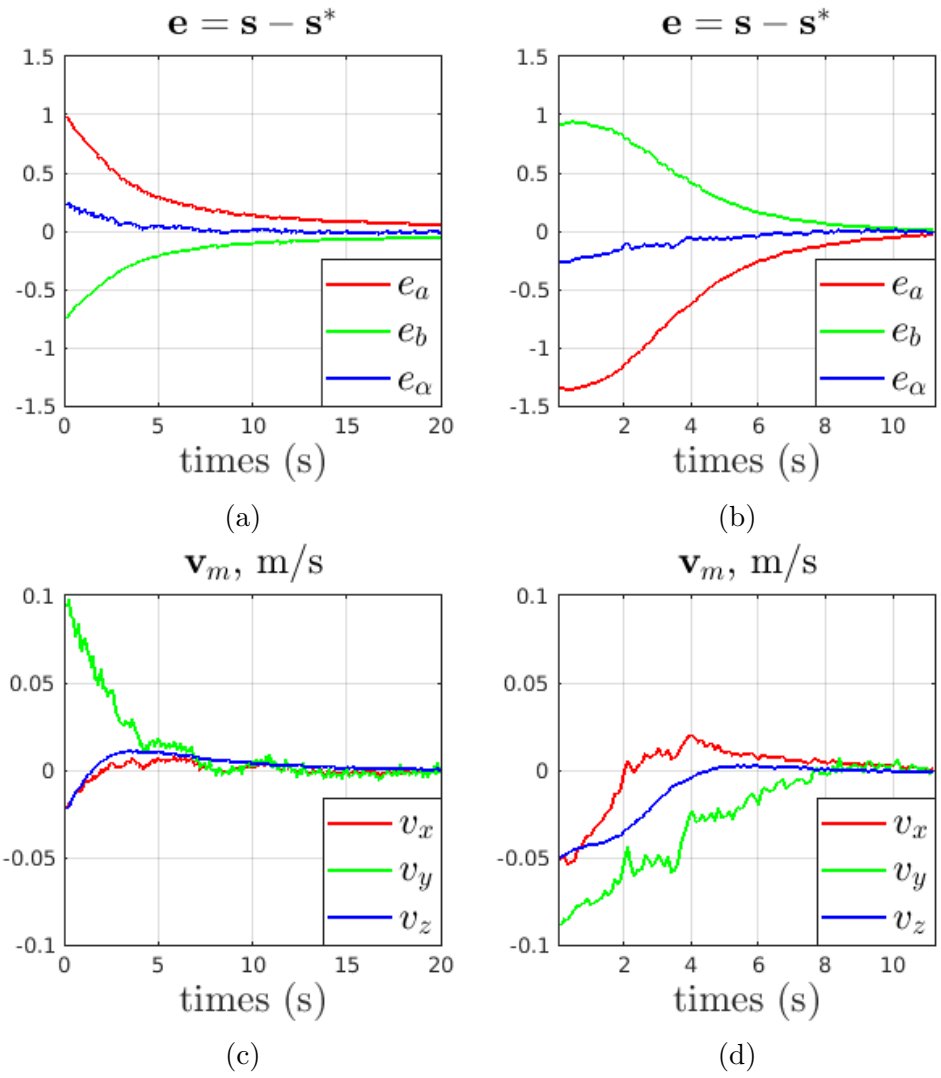


Figure 3.10 – Evolution of the visual features error during Task 1 (a) and Task 2 (b). Applied control velocity $\mathbf{v}_m$ during Task 1 (c) and Task 2 (d). For features units, α in rad, a in $\frac{1}{m}$, b no units.

punov theory for stability analysis. Similarly, we demonstrated that the error dynamics of e_α does not depend on D whatever its value (except that it must be positive). We also proposed exploiting the estimated cable plane normal to be more robust to noise during feature extraction.

Simulations together with real world tests have shown the efficiency of this method that does not require any knowledge of material mechanical properties.

In comparison to model-free methods, the proposed approach has the advantage to provide a complete study of potential singular configurations and it gets rid of an online estimation of the shape Jacobian. Moreover, it does not require any path planning for local minima avoidance and allows the system to bring the cable to taut configurations, which was not studied before. Even if the proposed interaction matrix was derived from a simple geometric parabola model, it is a continuum mechanics solution for cables that are in static equilibrium under its own weight and in the presence of some point load hanging on the cable, as was pointed out in Chapter 2.

In the next chapter, among others, an interesting property of the parabola model representing the cable profile subject to some point load hanging on it will be demonstrated in a series of experiments with quadrotor drones attached to the cable while transporting a load.

Chapter 4

Shape visual servoing of a tether cable using drones

In this chapter, we show an example of a practical application of the shape visual servoing framework derived in the previous chapter. Two application scenarios are demonstrated for the visual servoing of a slack cable. We present our modelling for control, simulation, and real-world experimental results that show the effectiveness of the proposed visual-servoing controller and image-processing pipeline to shape a slack cable into a desired shape and even to manipulate other objects.

The chapter is organised as follows: Section 4.1 introduces the possible applications of deformable object manipulation using drones, Section 4.2 describes our proposed image processing to track the cable deformation. Section 4.3 introduces the low-level drone control method used in our experiments. Section 4.4 deals with cable deformation using only one drone while the other end of the cable is static. Section 4.5 explains and experimentally demonstrates the control of the shape of the flexible cable by a pair of drones, and finally Section 4.6 concludes this chapter.

4.1 Application context

First, according to [Malveiro & Cordeiro, 2017], [Pawlak & Serek, 2017] drones can be used for high voltage transmission line stringing operations. The main task is to lay power cables from one high-voltage tower to another. Performing such an operation on a tower at high altitude presents a major risk to human operators. It is therefore safer and more efficient to replace human operators with a drone to carry out this type of task. In another context, drones can be used to search for and rescue people trapped in debris or collapsed buildings. In these situations, it may be necessary to manipulate deformable objects, such as ropes and ladders. In [Kiribayashi et al., 2017] the use of drones for emergency restoration work has been studied for tasks such as obstacle removal and embankment

construction for preventing damage spread. Japan has particularly been researching and developing unmanned construction methods using teleoperated construction machines and the use of drones with flexible power cables attached to the roof of these machines for emergency restoration work. In this context, the drone can be localised by observing the shape of a tether power cable and controlled to maintain a desired shape.

Drones can be used to manipulate deformable objects such as crops, fruits, and vegetables for agricultural purposes [Mayer et al., 2023]. Flying drones with cables can be used to manipulate objects, such as bricks, concrete, and steel, on construction sites, as demonstrated by [Tognon et al., 2018], [Sanalitro et al., 2022], [Jiménez-Cano et al., 2022]. Drones can also be used to deliver goods. In the context of e-Commerce, there is an increasing need for a more efficient logistics and transportation industry. One of the main challenges for logistics providers is the last mile delivery problem (delivery of packages to the final customer from the e-Retailer hub) due to its high cost, environmental damage and inefficiency. It is the least desirable part of the supply chain for providers, and the use of drones can significantly reduce the negative effects of last mile delivery [Madani & Ndiaye, 2022]. Soft objects (e.g. cables) are energy efficient, easier to recycle, light, and capable of carrying large loads and colliding with the environment without breaking the drone or damaging the environment, which are advantageous properties compared to conventional approaches such as the combination of a drone with a serial manipulator [Tognon et al., 2019] or the combination of a drone with a delta robot [Bodie et al., 2021] and [Danko et al., 2015]. Furthermore, manipulating slack cables involves planning and control of multi-drone system formation to avoid knotting of the power cables [Cao et al., 2023] or, in contrast, using similar formations to fold knots with cables to form nets [D'Antonio & Saldana, 2022], which can also serve for load transportation. Despite all the benefits of using cables and other soft bodies, controlling its shape remains an open problem. In our work, an efficient solution for manipulating a flexible slack cable to control its shape is proposed and validated by experiments.

4.2 Visual features estimation

In this section, we present an image processing pipeline for estimating in real time the parameters of the parabola that best fits the shape of the cable from the data provided by a RGB-D camera. We consider two cases for the camera location. In the first case, it is fixed on a static tripod while observing the drone together with the cable (see Figure 4.3).

In the second case, the camera is mounted on the drone and observes only the cable (see Figure 4.4). The main difference with respect to the image processing presented in Chapter 2 is that the cable is no more supposed to be in the vertical plane and its segmentation is performed from the entire point cloud and does not rely on a skeleton computation of the cable. In fact, the previous method based on the skeleton process has the limitation of discarding many measurement points from the parabola estimation, and as a result the obtained parabola parameters were less accurate and also more sensitive to measurement noise. The new image processing we propose consists first in capturing the current point cloud provided by the RGB-D camera. Then the PCL library [Rusu & Cousins, 2011] is used to downsample the point cloud using a voxelized grid approach. For this, the VoxelGrid [1] algorithm from the PCL is employed to create a 3D voxel grid over the input point cloud data (see Figure 4.1). The centroid of each voxel is then computed, and only its coordinates are used further. The point cloud is then expressed in the world frame

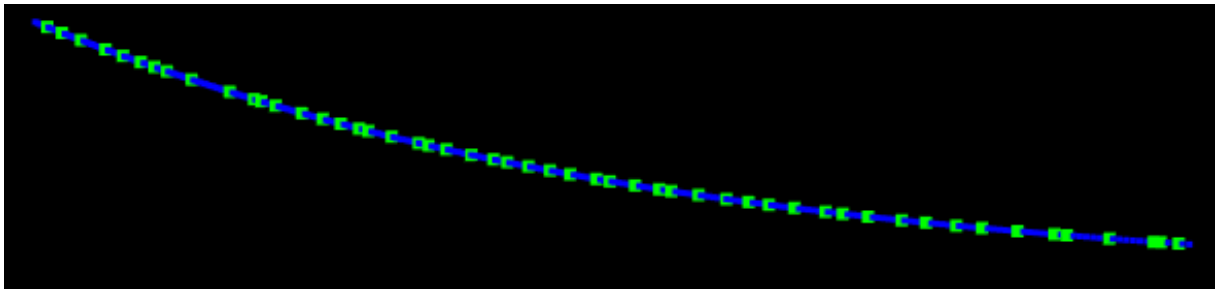


Figure 4.1 – An illustration of the VoxelGrid downsampling of the cable pointcloud. The original pointcloud is depicted in blue.

$\mathcal{F}_w$ thanks to the transformation between the camera frame $\mathcal{F}_c$ and $\mathcal{F}_w$ that is obtained by the classical calibration method [R. Tsai, 1987] for the first case where the camera is static. In case of the embedded camera on the drone, we first find the transformation matrix between $\mathcal{F}_c$ and the drone body frame $\mathcal{F}_{b_m}$ with the same method as described in the previous case. Afterwards, the drone odometry is used to express the reconstructed point cloud of the cable in $\mathcal{F}_w$ at each iteration of the cable shape controller. Then a RANSAC method is used to estimate the cable plane that best contains all voxel centroids. This results in an estimate of its normal ${}^w\mathbf{n} = \left(\frac{A}{\sqrt{A^2+B^2+C^2}}, \frac{B}{\sqrt{A^2+B^2+C^2}}, \frac{C}{\sqrt{A^2+B^2+C^2}}\right)$ and the set of inliers in the point cloud corresponding to the best-fit plane whose parameters

1. https://pointclouds.org/documentation/tutorials/voxel_grid.html

A, B, C, D are obtained by minimising the following objective:

$$\begin{aligned} \min_{A,B,C,D} \quad & \textstyle\sum_i Ax_i + By_i + Cz_i + D \\ \text{s.t} \quad & |\arctan \tfrac{C\sqrt{2}}{\sqrt{A^2+B^2}}| \leq 15^o \end{aligned} \tag{4.1}$$

where each centroid coordinate is given by (x_i, y_i, z_i) and D being the distance to the

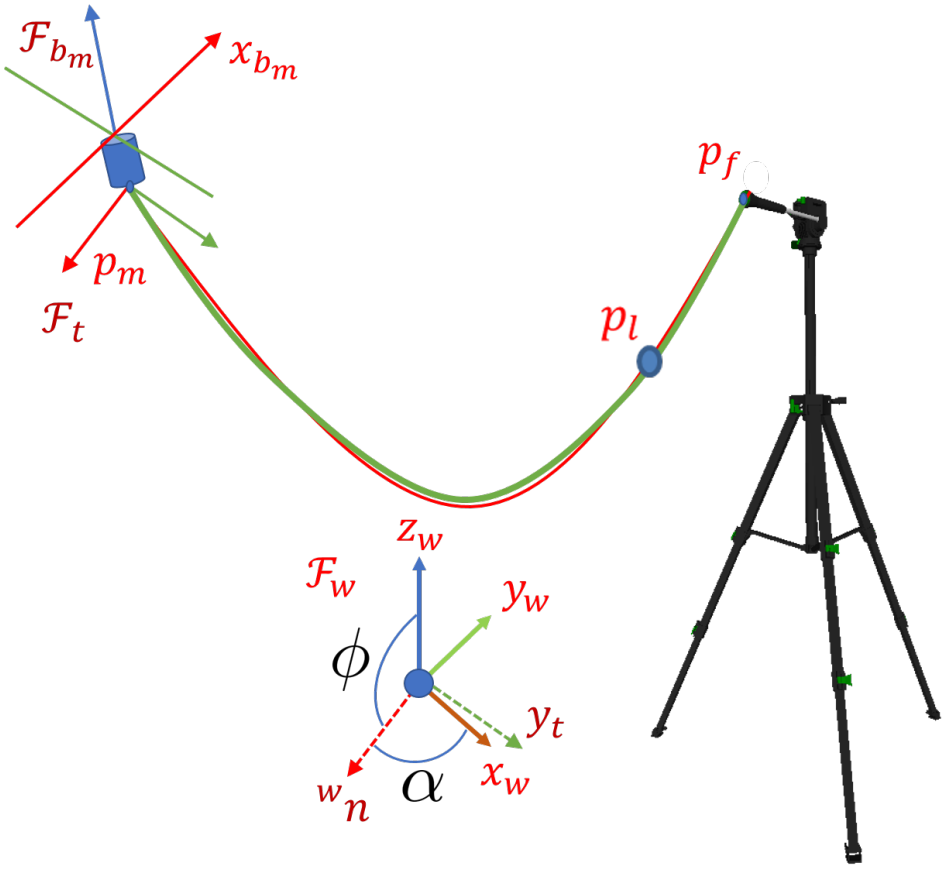


Figure 4.2 – Illustration of quantities used for features extraction. Drone with a cable in green. The parabola is depicted in red. $\mathcal{F}_t$ stands for the cable frame, $\mathcal{F}_{b_m}$ for the drone body frame. The unit vectors ${}^w\mathbf{n}, \mathbf{y}_t$ depicted in dotted red and green respectively represent axis of $\mathcal{F}_t$ projected in $\mathcal{F}_w$ to illustrate the plane orientation angles α and ϕ.

plane from the origin. In the case where the cable shape becomes a straight line, the minimisation of (4.1) by the RANSAC algorithm results in a vertical plane. Since the resulted normal can have different signs pointing towards the cable plane or in the opposite direction, one could use a Kalman filter to reject such fluctuations, but it is highly dependent on the initialisation condition for such normal. Instead, we propose to exploit

the fact that the point cloud is sorted in a list by its Z coordinates and that the last point ${}^{w}\mathbf{p}_l$ is the farthest point observed by the camera from the attachment point on the drone $\mathbf{b}_m$. The following relation gives a direction $\mathbf{d}_n$ for the cable in $\mathcal{F}_w$ starting from its attachment point in $\mathbf{p}_m$ (see Figure 4.2):

$$\mathbf{d}_n = \frac{{}^{w}\mathbf{p}_l - {}^{w}\mathbf{p}_m}{||{}^{w}\mathbf{p}_l - {}^{w}\mathbf{p}_m||} \tag{4.2}$$

To guarantee that the normal always points toward the cable a simple cross-product $\mathbf{c}$ is required:

$$\mathbf{c} = {}^{w}\mathbf{n} \times \mathbf{d}_n \tag{4.3}$$

It is important to check the sign of $\mathbf{c}_z$: positive sign means no normal flip is required and negative requires that the direction of the normal has to be reversed in the opposite direction. In the next step, a noise reduction of the estimated normal of the plane is performed thanks to the use of a Kalman filter that is based on a constant velocity model. After that, similar to the previous chapter, it is straightforward to find the yaw angle of the plane that corresponds to the feature α:

$$\alpha \quad = \quad \arctan \frac{{}^{w}\mathbf{n}_y}{{}^{w}\mathbf{n}_x} \tag{4.4}$$

From the estimated normal vector of the plane, we can also obtain the roll angle ϕ of the plane that describes its deviation with respect to the gravity vector:

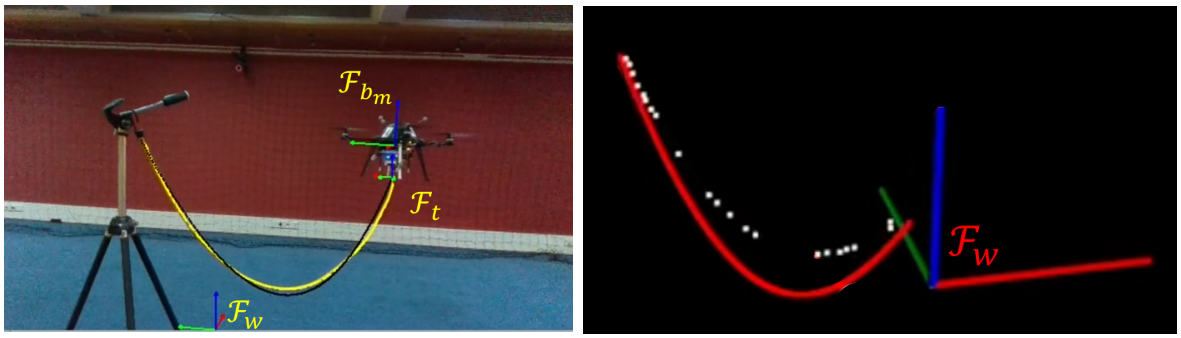
$$\phi = \arctan \frac{{}^{w}\mathbf{n}_z}{\sqrt{{}^{w}\mathbf{n}_x^2 + {}^{w}\mathbf{n}_y^2}} \tag{4.5}$$

Note that in our modelling, we have assumed that the plane containing the cable is vertical due to the gravity. However, in practice, when manipulating the cable with one or several drones, this plane may also exhibit a slight rolling motion due to the dynamics of the drones (the drone vibrates and slightly tilts to change its pose). Unfortunately, there is no way to control that tilt using underactuated quadrotor drones. From experiments, we observed that ϕ varies in an interval of $\pm 15^o$. The estimate ϕ given by (4.5) can be used to compensate for this tilt when estimating parabolic features a, b with the least-squares fitting method presented in Chapter 3. We now describe how to proceed to take into account that the plane containing the cable may deviate in practice by the roll angle ϕ. The homogeneous transformation ${}^{w}\mathbf{H}_t$ that was introduced in (3.2) for the case of a

vertical plane becomes:

$$ {}^{w}\mathbf{H}_{t} = \begin{bmatrix} {}^{w}\mathbf{n} & {}^{w}\mathbf{n} \times (0,0,-1) & {}^{w}\mathbf{n} \times ({}^{w}\mathbf{n} \times (0,0,-1)) & {}^{w}\mathbf{p}_{m} \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{4.6} $$

As mentioned before, ϕ can not be controlled but its online estimation allows to consider that the parabola plane is not vertical which may be the case when slight cable oscillations are induced by the drone motion. Moreover, to increase the robustness of the parabolic feature tracking, another Kalman filter is applied to the parabola parameters a, b to avoid abrupt changes of their values that may occur due to the limited field of view of the camera. In practice, a high camera frame rate and fast image processing is

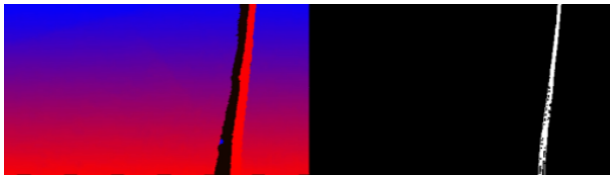


(a) (b)

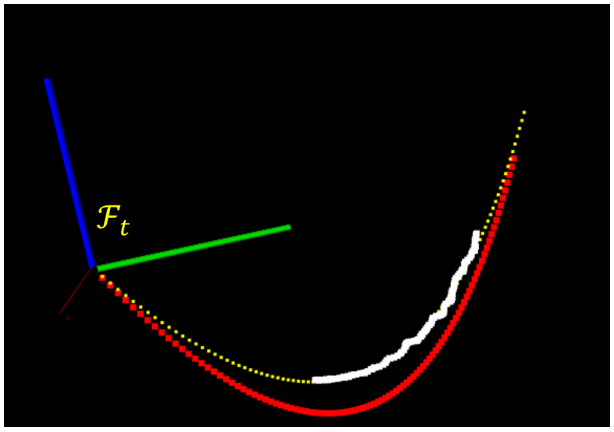
Figure 4.3 – Demonstration of the proposed visual features estimation with eye-to-hand camera setup. (a) Camera RGB image: extracted cable pointcloud (projected in RGB image) depicted in yellow, fitted parabola depicted in black after processing of cable pointcloud in the world frame composed of: downsampling → RANSAC → ${}^{w}\mathbf{n}$ sign check and filtering → α, ϕ → (4.6) → resampling of tether frame points using (3.42) → least squares fitting and a, b filtering.(b) 3D view of the process: white points correspond to the best fit normal inliers and red points represent some desired shape a^*, b^*, α^*. Note that (3.42) is used instead of inliers to ensure uniform distribution of tether points, thus reducing noise.

preferable for the visual servoing to be responsive. Therefore, processing the point cloud to extract the visual features and the plane angle ϕ becomes a challenging task for the embedded hardware on the small drone due to the processing delay between camera shot and tracking. To achieve real-time capability, a hardware acceleration of image processing was implemented on the GPU using the CUDA framework [2]. It allows us to track the visual features at a rate of $60Hz$ (17 ms). It is important to mention that the constant

2. https://docs.nvidia.com/cuda/doc/index.html



(a)



(b)

Figure 4.4 – Demonstration of the proposed visual features estimation with eye-in-hand camera setup. (a) Depth image on the left and segmented cable on the RGB image on the right. Depth image is aligned with the RGB image. (b) 3D view: the white points correspond to the cable points which are projected to the tether frame (4.6) and then used to estimate a, b. Fitted parabola in $\mathcal{F}_t$ depicted in yellow after the processing of the cable pointcloud in the world frame composed of: downsampling $\rightarrow$ RANSAC $\rightarrow$ ${}^{w}\mathbf{n}$ sign check and filtering $\rightarrow \alpha, \phi \rightarrow$ (4.6) $\rightarrow$ least squares fitting and a, b filtering. Red points correspond to some desired shape a^*, b^*. Note that the cable inliers distribution is more dense compared to the eye-to-hand case, but less of the cable is visible in the RGB camera FOV that is limited by the lens and orientation of the camera.

position ${}^{b_m}\mathbf{p}_m$ of the cable attachment point with respect to the drone body frame must be calibrated carefully, since it is required in the estimation process of visual features.

4.3 Modelling and control of the quadrotor

In this section, we explain the modelling of the quadrotors used for low-level control of its Centre of Gravity (COG) position and body yaw angle. For both scenarios, the input of this controller is the velocity $\mathbf{v}_m^*$ of the cable attachment point whose position is assumed to be known in the drone frame together with the drone yaw angular velocity $\dot{\psi}_{b_m}^*$ (which is set to an arbitrary value in the first scenario). These inputs are provided at a rate of $20Hz$ to the low-level drone controller.

4.3.1 Modelling of the quadrotor dynamics

The system is described by the body frame $\mathcal{F}_{b_m}$ of the drone centered on its COG ${}^w\mathbf{p}_m$ and the world frame $\mathcal{F}_w$ where are expressed the control velocities provided by our visual control law (3.26) (see Figure 4.6). The drone physical model is given by:

$$\begin{aligned} {}^w\dot{\mathbf{p}}_m &= \mathbf{v}_m \\ \dot{\mathbf{v}}_m &= \mathbf{a}_m = \frac{f}{m_{b_m}} \begin{bmatrix} 2(q_w q_y + q_x q_z) \\ 2(q_y q_z - q_w q_x) \\ 1 - 2(q_x^2 + q_y^2) \end{bmatrix} - g\mathbf{e}_3 \\ \dot{\mathbf{q}} &= \frac{1}{2} \begin{bmatrix} 0 \\ \boldsymbol{\omega} \end{bmatrix} \otimes \mathbf{q} \\ \dot{\boldsymbol{\omega}} &= \mathbf{I}_{b_m}^{-1}(\boldsymbol{\tau} - \boldsymbol{\omega} \times \mathbf{I}_{b_m}\boldsymbol{\omega}) \end{aligned} \tag{4.7}$$

where $f, \boldsymbol{\tau}$ are total thrust and body moments expressed in $\mathcal{F}_{b_m}$, $g\mathbf{e}_3$ is the gravity vector with $\mathbf{e}_3 = (0, 0, 1)$, m_{b_m} stands for the mass of the quadrotor and $\mathbf{I}_{b_m}$ represents its inertia matrix. In (4.7) the unit quaternion $\mathbf{q} = (q_w, q_x, q_y, q_z)$ is used to represent the rotation of the body frame $\mathcal{F}_{b_m}$ with respect to the world frame $\mathcal{F}_w$ and it is important to introduce the rotation matrix $\mathbf{R}(\phi_{b_m}, \theta_{b_m}, \psi_{b_m}) = \mathbf{f}(\mathbf{q})$, with $(\phi_{b_m}, \theta_{b_m}, \psi_{b_m})$ the drone body orientation (roll, pitch, and yaw angles in the world frame $\mathcal{F}_w$, respectively), used in the equations of the drone tracking controller described further. Finally, $\boldsymbol{\omega}$ denotes the angular velocity of the body frame of the quadrotor. To map total thrust f and body

moments $\boldsymbol{\tau}$ to propeller velocities $\boldsymbol{\omega}_{prop}$, the inverse drone allocation matrix $\mathbf{A}^{-1}$ is used with $\mathbf{A}$ given by (under the assumption that the barycenter of the drone coincides with its COG):

$$\begin{pmatrix} f \\ \boldsymbol{\tau} \end{pmatrix} = k \begin{bmatrix} 1 & 1 & 1 & 1 \\ 0 & l & 0 & -l \\ -l & 0 & l & 0 \\ c & -c & c & -c \end{bmatrix} \boldsymbol{\omega}_{prop} \tag{4.8}$$

where k and c are aerodynamic coefficients of the propellers that can be identified following the procedure described in [Spica et al., 2013] for a real quadrotor, l being the distance between the COG of the quadrotor and the propellers. In the next section, the control law of the drone will be introduced.

4.3.2 Control of the quadrotor drone

The drone is controlled by designing a thrust vector $f\mathbf{R}^*\mathbf{e}_3$ to track the desired translation $\mathbf{p}_m^*$ of the drone and its first derivative, denoted $\mathbf{v}_m^*$, and to compensate for gravity. At the same time, the quadrotor torques $\boldsymbol{\tau}$ are controlled by aligning its body orientation with the desired one denoted $\mathbf{R}^*$ to apply the thrust in the right direction and follow the yaw angle ψ_{b_m} to the desired value $\psi_{b_m}^*$. Note that the quadrotor in (4.7) is underactuated and has only four controllable DOFs, which are ${}^w\mathbf{p}_m, \psi_{b_m}$ also called outputs of the quadrotor. One common solution to this control problem is to apply a *Lee* controller [Lee et al., 2010] that achieves an almost global exponential stability of the tracking errors. The general quadrotor controller equations are given by:

$$
\begin{aligned}
\mathbf{e}_{p_m} &= {}^w\mathbf{p}_m^* - {}^w\mathbf{p}_m \\
\mathbf{e}_{v_m} &= \mathbf{v}_m^* - \mathbf{v}_m \\
\mathbf{e}_R &= \frac{1}{2}\left(\mathbf{R}^{*T}\mathbf{R} - \mathbf{R}^T\mathbf{R}^*\right)^\vee \\
\mathbf{e}_\omega &= \boldsymbol{\omega} - \mathbf{R}^T\mathbf{R}^*\boldsymbol{\omega}^* \\
\mathbf{z}_{b_m}^* &= \frac{\mathbf{K}_{p_m}\mathbf{e}_{p_m} + \mathbf{K}_{v_m}\mathbf{e}_{v_m} + \mathbf{K}_i\int \mathbf{e}_{p_m}dt + m_{b_m}g\mathbf{e}_3 + m_{b_m}\mathbf{a}_m^*}{||\mathbf{K}_{p_m}\mathbf{e}_{p_m} + \mathbf{K}_{v_m}\mathbf{e}_{v_m} + \mathbf{K}_i\int \mathbf{e}_{p_m}dt + m_{b_m}g\mathbf{e}_3 + m_{b_m}\mathbf{a}_m^*||} \\
\mathbf{x}_{b_m}^* &= (\cos\psi_{b_m}^*, \sin\psi_{b_m}^*, 0) \\
\mathbf{y}_{b_m}^* &= \frac{\mathbf{z}_{b_m}^* \times \mathbf{x}_{b_m}^*}{||\mathbf{z}_{b_m}^* \times \mathbf{x}_{b_m}^*||} \\
\mathbf{R}^* &= \begin{bmatrix}\mathbf{y}_{b_m}^* \times \mathbf{z}_{b_m}^* & \mathbf{y}_{b_m}^* & \mathbf{z}_{b_m}^*\end{bmatrix} \\
f &= \left(\mathbf{K}_{p_m}\mathbf{e}_{p_m} + \mathbf{K}_{v_m}\mathbf{e}_{v_m} + \mathbf{K}_i\int \mathbf{e}_{p_m}dt + m_{b_m}g\mathbf{e}_3 + m_{b_m}\mathbf{a}_m^*\right)^T \mathbf{R}\mathbf{e}_3 \\
\boldsymbol{\tau} &= -\mathbf{K}_R\mathbf{e}_R - \mathbf{K}_\omega\mathbf{e}_\omega
\end{aligned}
\tag{4.9}
$$

where $()^\vee$ is the "vee" map and all $\mathbf{K}$ represent diagonal 3×3 control gain matrix that gives us fifteen parameters that need be tuned, which is the main drawback of the controller [Lee et al., 2010]. The integral terms in (4.9) compensate for the unmodeled disturbances and parameter estimation errors. As can be seen in (4.9) it is necessary to provide some reference trajectory $(\mathbf{p}_m^*(t), \psi_{b_m}^*(t))$ and its derivatives (which are $\dot{\mathbf{v}}_m^* = \mathbf{a}_m^*$,$\mathbf{v}_m^*$ and $\dot{\psi}_{b_m}^* = \omega_\psi^*$ with $\boldsymbol{\omega}^* = (0, 0, \omega_\psi^*)$) to the controller to generate the total thrust f and the torques $\boldsymbol{\tau}$. Then, it is straightforward to compute the motor reference signals:

$$
\boldsymbol{\omega}_{prop}^* = \mathbf{A}^{-1}\begin{pmatrix} f \\ \boldsymbol{\tau} \end{pmatrix} \tag{4.10}
$$

These motor reference signals $\boldsymbol{\omega}^*_{prop}$ must be generated at a frequency of approximately $1kHz$ for our drone to be able to fly. That is a real challenge for embedded drone controller software. Furthermore, as can be seen again in (4.9) to follow the trajectory, the drone needs to know its current pose ${}^w\mathbf{p}_m, \mathbf{R}$ and velocity $\mathbf{v}_m, \boldsymbol{\omega}$ which are estimated using an unscented Kalman filter at a frequency of $1kHz$. The structural diagram of the control law (4.9) is shown in Figure 4.5. In the next section we will explain how (4.9) is

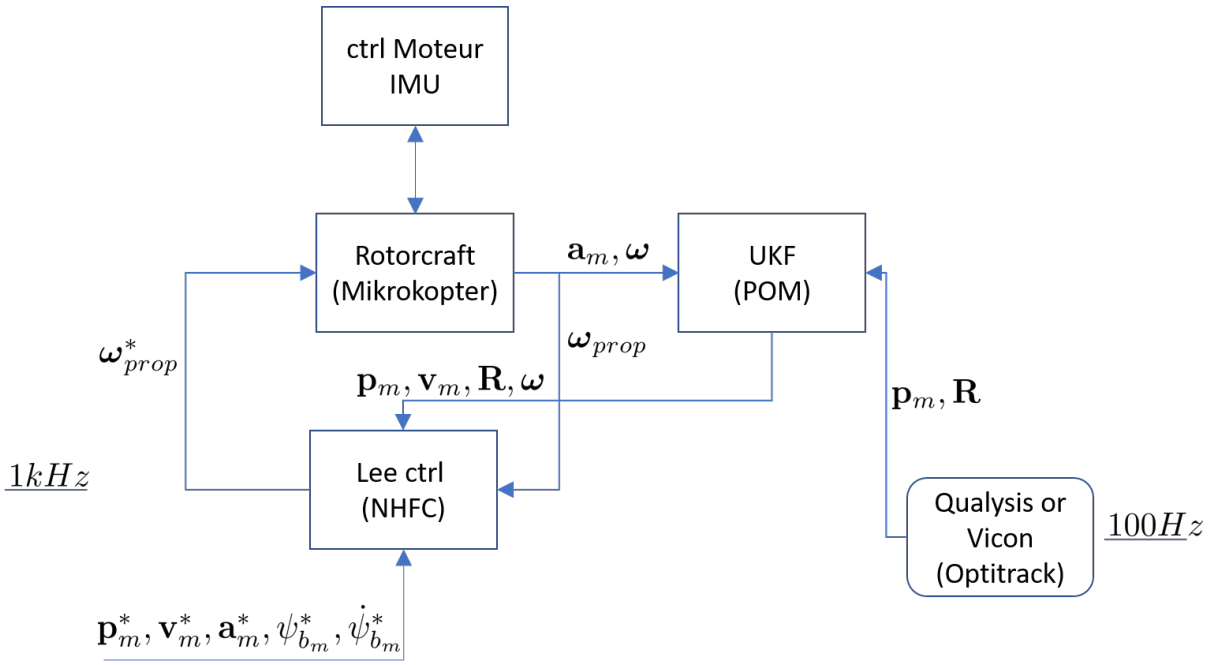


Figure 4.5 – Scheme of (4.9) associated to the middleware control layer depicted in Figure 4.10.

implemented to deal with the cable dynamics and low rate of the camera (compared to the rate of motor reference signals).

4.4 First scenario: Cable manipulation by one drone using eye-to-hand visual servoing

In this section, we develop a strategy for controlling a quadrotor drone manipulating a cable. In this first scenario, one cable extremity is attached to the quadrotor drone, while the other extremity is fixed on a static tripod. An eye-to-hand RGB-D camera is used to observe the quadrotor and the cable within its workspace. The set of visual features

considered is the same as defined in the previous chapter. A specific mechanical design of the cable attachment platform creates a passive ball joint that allows ignoring the drone orientation and thus decoupling the rotational degrees of freedom from translational ones. Therefore, only the translational degrees of freedom of the drone are used for the visual servoing task. It is demonstrated and validated by experiments that the proposed visual features allow to generate the desired thrust input of each rotor in order to deform the cable to a desired shape. We present simulation and real-world experimental results showing the effectiveness of the proposed visual-servoing controller and image-processing pipeline to shape a slack cable.

4.4.1 Modelling of the system

The overall scheme for the first scenario is presented in Figure 4.6. To reduce the coupling between the orientation change of the drone and the translational motion of the cable extremity held by the drone, the passive ball joint is positioned as much as possible in the center of gravity (COG) of the drone. The system is described by the body frame $\mathcal{F}_{b_m}$ of the drone, with origin in the COG and the world frame $\mathcal{F}_w$ where the control velocities are expressed. From the relation (3.21) that gives the variation of the visual features $\mathbf{s} = (a, b, \alpha)$ as a function of the velocities ${}^w\dot{\mathbf{p}}_m$, ${}^w\dot{\mathbf{p}}_f$ of the cable end points, we directly have ${}^w\dot{\mathbf{p}}_f = 0$ for this scenario where one extremity is motionless. In the following, we make the assumption that the origin $\mathbf{p}_m$ of the tether frame $\mathcal{F}_t$ coincides with the origin of the drone body frame $\mathcal{F}_{b_m}$. The dynamics of the drone $\mathbf{b}_m$ holding a cable is described by the following system of equations in compact form given by:

$$
\begin{aligned}
{}^w\dot{\mathbf{p}}_m &= \mathbf{v}_m \\
\mathbf{v}_m &= -\mathbf{M}\dot{\mathbf{s}} \\
\dot{\mathbf{v}}_m &= \frac{f}{m_{b_m}} \begin{bmatrix} 2(q_w q_y + q_x q_z) \\ 2(q_y q_z - q_w q_x) \\ 1 - 2(q_x^2 + q_y^2) \end{bmatrix} - g\mathbf{e}_3 \\
\dot{\mathbf{q}} &= \frac{1}{2} \begin{bmatrix} 0 \\ \boldsymbol{\omega} \end{bmatrix} \otimes \mathbf{q} \\
\dot{\boldsymbol{\omega}} &= \mathbf{I}_{b_m}^{-1}(\boldsymbol{\tau} - \boldsymbol{\omega} \times \mathbf{I}_{b_m}\boldsymbol{\omega})
\end{aligned}
\tag{4.11}
$$

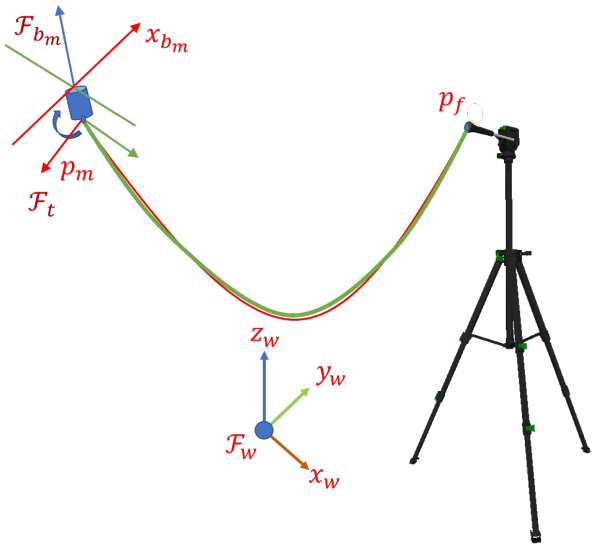


Figure 4.6 – Drone with a cable in green. The parabola is depicted in red. The blue cylinder with a circle on its end represent a passive ball joint. $\mathcal{F}_t$ stands for tether frame, $\mathcal{F}_{b_m}$ for drone body frame.

where the second equation is the control output of the visual control law (3.21) introduced in Chapter 3. The other equations come from the dynamics of the drone introduced in (4.7). It is important to mention that this model does not take into account the tension of the cable, since we made the assumption that its weight is negligible. In the next section, we present the system control which will then be validated from real experiments.

4.4.2 Control of the system

The main idea behind the control of the system (4.11) is to drive the quadrotor COG $\mathbf{p}_m$ to a desired pose in $\mathcal{F}_w$ to achieve the desired cable shape $\mathbf{s}^*$. It can be done by designing a thrust vector $f\mathbf{R}^*\mathbf{e}_3$ to track the desired translation $\mathbf{p}_m^*$ of the drone and its derivative, denoted $\mathbf{v}_m^*$. At the same time, the quadrotor torques $\boldsymbol{\tau}$ are controlled by aligning its body orientation with the desired one denoted $\mathbf{R}^*$ to apply the thrust in the right direction and drive the yaw angle ψ_{b_m} to the desired value $\psi_{b_m}^*$ (which is set arbitrarily in the first scenario but allows us to analyse how well the passive ball joint behaves in practice). Note that the quadrotor in (4.11) is underactuated and has only four controllable DOFs, which are ${}^w\mathbf{p}_m, \psi_{b_m}$ (although we do not control the yaw by the proposed shape visual servoing in this scenario). Furthermore, the model (4.11) shows

that, given some desired trajectory of the cable shape $\dot{\mathbf{s}}^*(t)$, it can be directly mapped to $\mathbf{v}_m(t)$. To map $\mathbf{p}_m(t)$ to $\mathbf{s}(t)$ (to track the trajectory of the drone COG in case it is available from some task), the position of the distal end of the cable should be extracted according to the cable model given by (3.4). The derivatives of (3.21) allow us to easily reconstruct the remaining output derivatives of $\mathbf{p}_m(t)$ up to the desired order, which is not trivial for the cable model [Gabellieri & Franchi, 2023]. Therefore, given some trajectory $\mathbf{s}^*(t)$ and the trajectory of one end of the cable *e.g.* $\mathbf{p}_m$ it would be possible to reconstruct the output trajectory of the other end of the cable (in case it would not be static) or vice versa, given $\mathbf{p}_m(t)$ and $\mathbf{p}_f(t)$, it would be possible to reconstruct $\mathbf{s}(t)$. Nevertheless, in this work, we consider that only the desired shape $\mathbf{s}^*$ of the cable is known. One common solution to this control problem is to apply the controller (4.9) which presents no other singularity conditions of the system apart from the singularities of the cable model considered in Chapter 3. Therefore, we can choose the reference velocity tracking error $\mathbf{e}_{v_m}$ for (4.9) to be as follows:

$$\begin{aligned} \mathbf{e}_{v_m} &= \mathbf{v}_m^* - \mathbf{v}_m \\ \mathbf{e} &= \mathbf{s} - \mathbf{s}^* \\ \mathbf{v}_m^* &= \lambda \mathbf{M} \mathbf{e} \end{aligned} \tag{4.12}$$

where the shape visual servoing gain λ gives us the sixteenth parameter (together with all $\mathbf{K}$ from (4.9)) that needs to be tuned for the controller. The challenge is to generate the trajectory for the quadrotor outputs $(\mathbf{p}_m^*(t), \psi_{b_m}^*(t))$ by observing the shape of the cable with a camera, whose rate is approximately $60Hz$ taking into account the time of tracking of the visual feature $\mathbf{s}$ and the calculation of the desired velocity $\mathbf{v}_m^*$. Note that there is no need for $\psi_{b_m}^*(t)$ in this first scenario, thus any arbitrary value can be used. A straightforward solution would be to interpolate the outputs each time a new visual servoing command $\mathbf{v}_m^*$ is computed. It can be done by using some differentiable curves, such as splines or polynomials, that respect the boundary conditions on the outputs and its derivatives to achieve $\mathbf{v}_m^*$ as fast as possible. The latter is important since the stability conditions of the visual servoing controller provided in the previous chapter hold under the assumption that the control velocity $\mathbf{v}_m$ can be executed by the robot. Fortunately, this solution already exists, as proposed by [Boeuf et al., 2015] with the quadrotor steering method to reconstruct the outputs trajectories (with their derivatives). The generated trajectory is feasible by the drone (is flyable) if the outputs derivatives are correctly

bounded.

In practice, the control law (4.9),(4.12) is implemented using an inner and an outer control loop strategy to deal with the slow rate of $\mathbf{v}_m^*$ and the fast rate needed for $\boldsymbol{\omega}_{prop}^*$. The structural diagram of the control law (4.9),(4.12) is shown in Figure 4.7. Since the cable model is derived using the static equilibrium assumption, the quadrotor can only move in near hovering mode in order to not destabilise the cable dynamics, and therefore its outputs derivatives remain correctly bounded during all the shape visual servoing task.

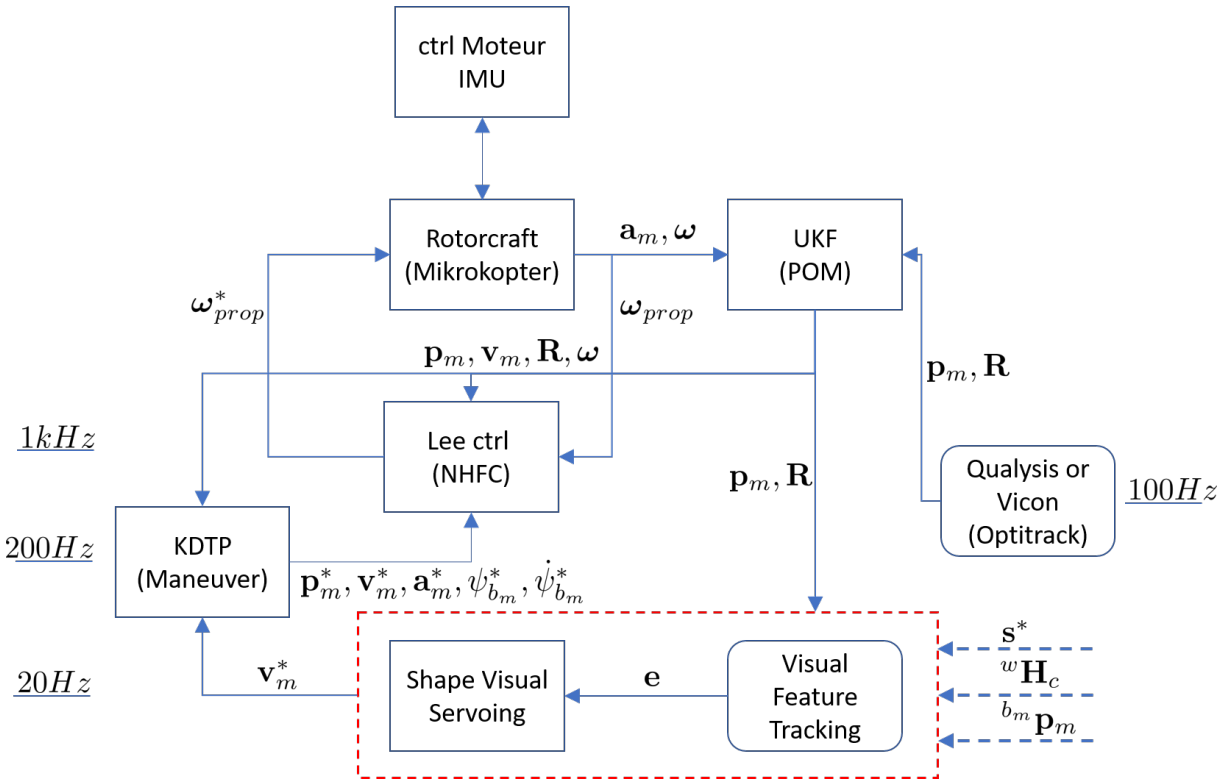


Figure 4.7 – Scheme of the quadrotor associated to the middleware control layer depicted in Figure 4.10. The terms ${}^w\mathbf{H}_c$ and ${}^{b_m}\mathbf{p}_m$ (together with $\mathbf{p}_m, \mathbf{R}$) are used to reconstruct the origin of the tether frame $\mathcal{F}_t$ in order to estimate the features a, b.

4.4.3 Simulation

This section aims to demonstrate the validity of our shape visual servoing to deform a cable with a drone. To this end, a software simulation environment was implemented. It consists of three submodules that simulate in a virtual 3D scene, a deformable cable,

an RGB-D camera that observes the cable, and a drone attached to one end of the cable. These 3 submodules are presented in the next subsections. We use Coppeliasim [Rohmer et al., 2013] and the bullet [3] physics engine integrated into Coppeliasim to simulate the dynamics of the cable. This is a practical choice because there is a well-documented and efficient programming interface (via scripts) provided by Coppeliasim for physical modelling of a cable (bullet dynamics), realistic rendering of the overall scene (3D visualisation), vision sensor handling (to simulate image processing needed for the extraction of the cable pointcloud) and finally the communication via the middleware tools (*i.e.* with drone simulator and visual servoing controller). The drone is simulated with Genom3 software [4]. It should be noted that the recent version of Coppeliasim (from version 4.4.0) implements the interface to the MuJoCo [Todorov et al., 2012] physics engine, which is oriented to physical simulation of soft objects. To our knowledge, until now the models of stretching cables are still not supported in the current version of MuJoCo, that is why our cable is simulated as a mass-spring-damper (MSD) system (series of mass interconnected by springs and dampers), which can be easily simulated in stretched form (see [Gabellieri & Franchi, 2023] for a detailed example of the mass-spring system).

Cable submodule

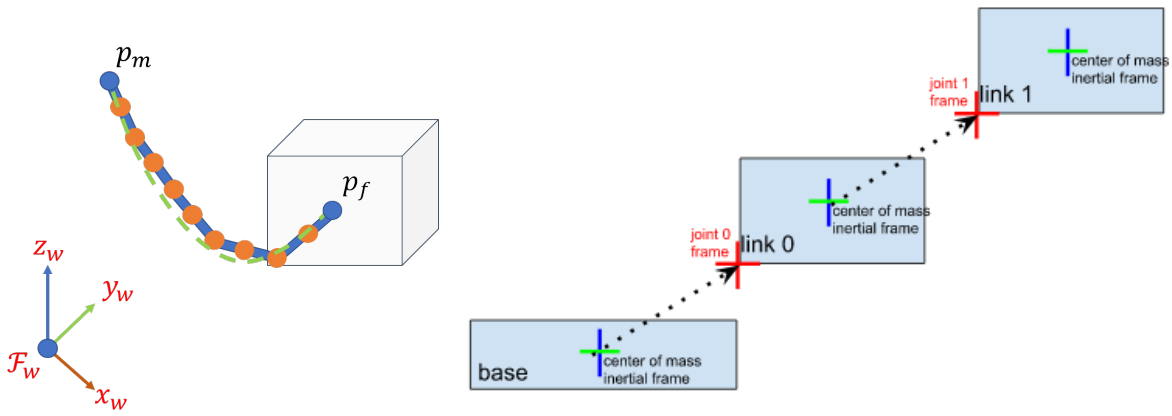
To simulate the deformable cable, we implement the MSD model [Chang & Padır, 2020] that was considered for soft robot modelling and control in [Della Santina et al., 2018]. Unlike the extensible catenary model that we used as ground truth in the simulation results presented in Chapter 3, we now choose to model the dynamics of the cable using an MSD. The main reason for this choice is that it allows simulation of cable deformations that may deviate from the vertical plane. The cable is represented as a chain made up of a series of rigid links connected by passive ball and socket joints (see Figure 4.8). The total length of the cable is $L = 1.6m$, which results in 11 links. Its dynamics can be described in a similar way as a serial robot manipulator by the following Lagrange equation (expressed in the joint space):

$$\mathbf{B}\ddot{\mathbf{q}} + \mathbf{C}\dot{\mathbf{q}} + \mathbf{g} + \mathbf{J}^T\mathbf{f}_{ext} + \mathbf{K}\mathbf{q} + \mathbf{D}\dot{\mathbf{q}} = \boldsymbol{\tau} \tag{4.13}$$

where $\mathbf{B}$ is the manipulator inertia matrix, $\mathbf{q}$ is the joint position, $\mathbf{C}$ is the centrifugal and Coriolis forces matrix, $\mathbf{g}$ is the gravitational force vector, $\mathbf{J}$ is the system Jacobian matrix, $\mathbf{f}_{ext}$ is the six-component vector of resultant force and moment exerted at the end

3. https://pybullet.org
4. https://git.openrobots.org/projects/mrsim-genom3

effector, **K** and **D** are the stiffness and damping matrices of the system, and finally $\boldsymbol{\tau}$ is the vector of joint torques applied by the actuators which is **0** for passive joints. The reader can refer to [L.-W. Tsai, 1999] for a complete overview of the dynamics of (4.13). For the simulated cable, we set **K** and **D** with the arbitrary parameters values given in Table 4.1. Equation (4.13) is solved recursively, including two phases: forward recursion to compute **q** and its derivatives followed by backward recursion to compute forces and moments exerted by the links on each other. Note that for simplification, we consider a flexible and mass-less cable and therefore do not propagate the torque that results at the extremity of the cable into the drone dynamics. For our simulation, the cable base link is connected to the drone and its last link is connected to a virtual 3D tripod through spherical joints, which are not simulated by the MSD (see Figure 4.9). That is why we impose a friction constraint on the base link to simulate damping in the lateral direction to the plane of the cable. The simulation time step is chosen to be 50 ms with 10 calculation passes for bullet dynamics.



(a) (b)

Figure 4.8 – (a) Cable (green curve) and its MSD model: orange circles represent 2 DOF revolute joints, blue rectangles represent rigid links and blue circles are spherical joints. (b) Relation between the joints and frames.

Note that not only are the cable dynamics being processed during simulation time but also the RGB-D data of the simulated camera. In the following, we briefly describe how to simulate an RGB-D camera in Coppeliasim.

m_i, kg	0.025
$k, Nm/deg$	0.01
$d, Nsm/deg$	0.03

Table 4.1 – Mechanical parameters values used in the MSD simulation

RGB-D camera submodule

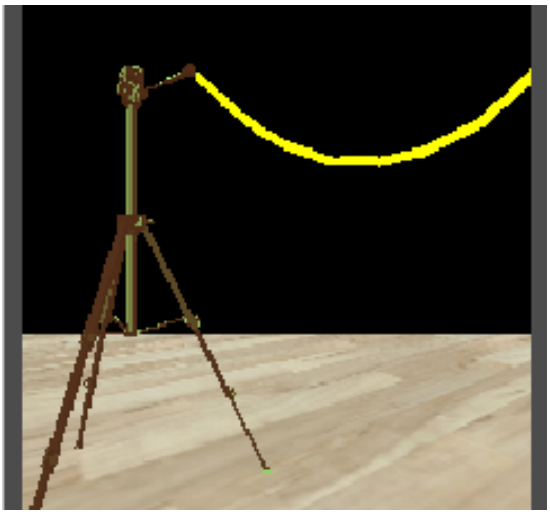
Coppeliasim is a 3D simulator with realistic rendering and light conditions. It provides a vast choice of image-processing tools, which makes it very practical for visual servoing simulations. Therefore, we can simulate the colour segmentation of the cable from the image and recover 3D coordinates for each point of the cable. The resolution of the camera is chosen to be $res_x = res_y = 200$ pixels with a field of view of $\alpha_x = \alpha_y = 57^o$ around the X and Y axes of the camera frame, providing a calculation rate of about 30 fps, which is a very convenient setup for real-time simulation. There is no image distortion. To convert between 3D coordinates of the point ${}^c\mathbf{p} = (X, Y, Z)$ and pixels $\mathbf{p}_{im} = (x, y)$, the following projection model is used:

$$\begin{aligned} x &= (0.5 - 0.5\frac{X}{Z\tan\alpha_x/2})(res_x - 1) \\ y &= (0.5 - 0.5\frac{Y}{Z\tan\alpha_y/2})(res_y - 1) \end{aligned} \tag{4.14}$$

Figure 4.9 shows an example of the visual and depth data provided by the simulated RGB-D camera in CoppeliaSim.

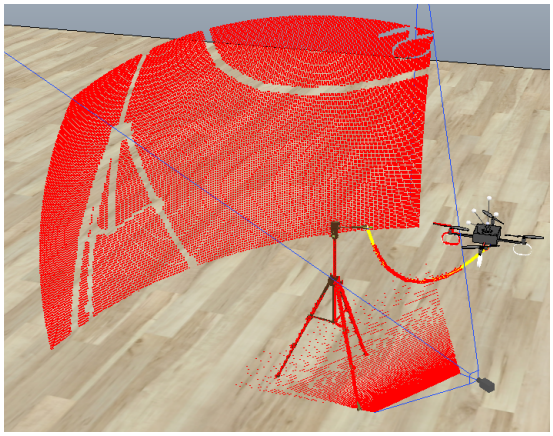
Drone submodule

Simulating the dynamics of a drone that is coupled with a flexible object remains a difficult and open problem. Recently, the authors of [Tempel et al., 2022] developed an interaction model that takes into account the partially constant strain model of a flexible object that is in physical interaction with drones. However, their model, which is based on Cosserat theory, is quite complex and cannot be used in a real-time robotic application. Therefore, in our simulation, we make the assumption that the drone dynamics is not affected by the cable and that only the dynamics of the cable is influenced by the drone motion according to the MSD presented before. Therefore, we simulate the drone behaviour independently. We directly apply the total thrust and body moments



(a)

(b)



(c)

Figure 4.9 – (a) RGB image from the static camera. (b) Binary image of the segmented cable from the static camera. (c) Overall view on the scene where the total of 200x200 points taken by the static camera are in red.

given by the control law (4.9),(4.12). Note that we use the SITL (Software-In-The-Loop) approach, allowing one to launch the drone controller either in simulation or in a real hardware experiment. It is implemented using the Genom3 platform [Foughali et al., 2017]. The reader can refer to [Foughali et al., 2017] for details of each block in Figure 4.10. Also note that control gain coefficients from (4.12) are already tuned thanks to

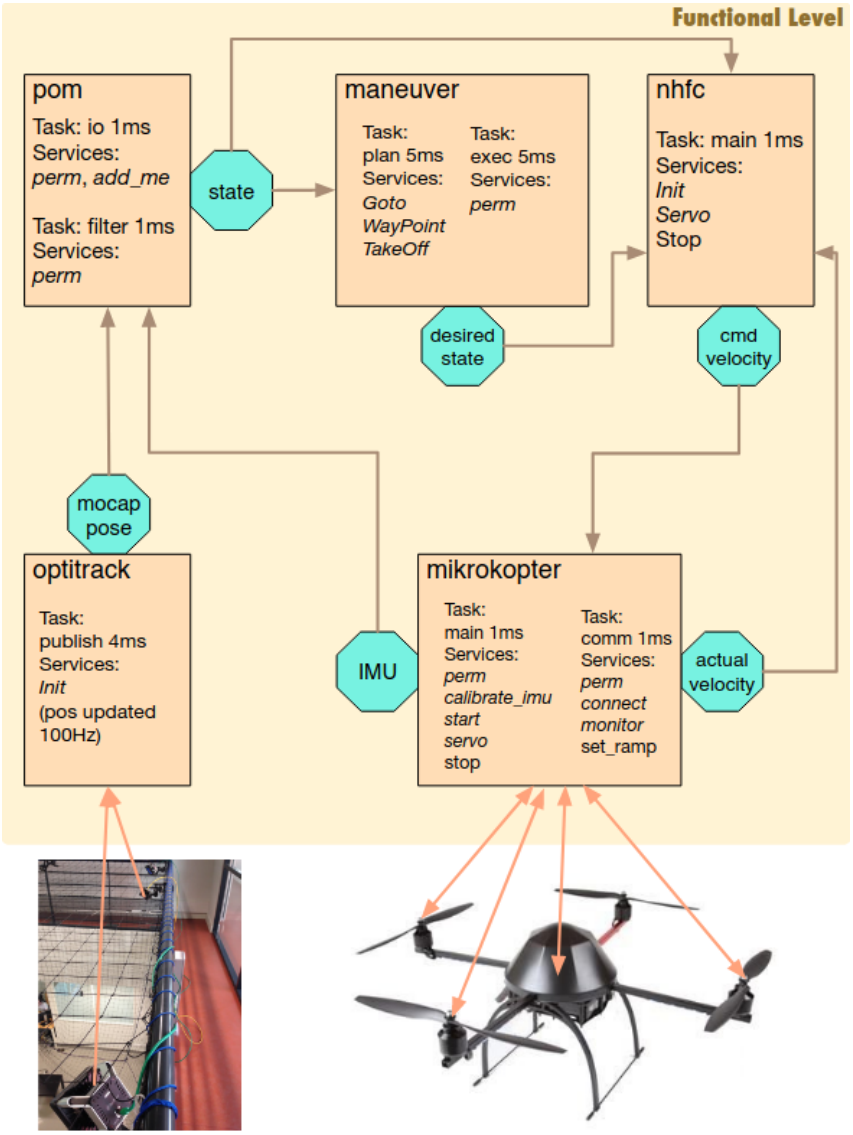


Figure 4.10 – Software blocks used in SITL simulations and real experiments from [Foughali et al., 2017]

.

Genom3 maintainers. Only the value of the visual servoing gain λ remains to be tuned.

Middleware

In previous sections, we described the cable, camera, and quadrotor drone submodules. All these components exchange data during simulation using a middleware interface. For this purpose, ROS Noetic [5] and Matlab [6] are used. To understand the basic concepts of ROS, the reader can refer to [Quigley et al., 2009]. Figure C.1 in Appendix C presents the ROS graph showing the dependencies implemented between the different nodes of the system. We provide brief explanations of how this graph works further. The RGB-D sensor publishes a binary image of the segmented cable on the "imageRGB" topic, and all coordinates of the 3D points on the "depthCoppelia" topic. The drone state messages are recovered from the "vrep" topic. The drone state together with the visual data provided by the RGB-D camera are synchronised in the "converter node" using approximate time-step synchronisation of the messages of the topics "vrep" (contains drone state), "imageRGB" (contains segmented binary image of the cable), and "depthCoppelia" (contains the 3D coordinates of all points in the camera field of view). The topic "cableCloud"(contains 3D coordinates of the points belonging to the cable) receives the drone state together with the point cloud. The node "controller node" subscribes to the "cableCloud" and "tf" topics (this one contains poses of the camera recovered from Coppeliasim, allowing it to move freely during simulation without disturbing the control of the system, which is very useful when choosing another view point of the camera). Here, the control velocity $\mathbf{v}_m^*$ and $\dot{\psi}_{b_m}$(which can be given an arbitrary value as it is unnecessary in the eye-to-hand case to control the drone yaw) are calculated and sent to the "quad1" topic. Afterwards, this velocity is recovered by the Genom3 "Maneuver" module via the Simulink interface, which are the nodes on the left of the rosgraph: the node "sim ros interface" corresponds to Coppeliasim, and the two remaining are for Simulink.

Simulation results

The proposed control approach is tested in a simulation framework using the MSD model of the tether cable to simulate its deformation. The desired drone yaw rate $\dot{\psi}_{b_m}^*$ is generated using $\dot{\psi}_{b_m}^* = -\lambda(\psi_{b_m} - \psi_{b_m}^*)$ to demonstrate that the control law of the cable plane yaw (4.12) based on $(\alpha - \alpha^*)$ is decoupled from the drone yaw control (4.9). A video of the simulation results is available at https://youtu.be/ADnjtzIcp70.

5. http://wiki.ros.org/noetic
6. https://www.mathworks.com

The initial shape of the cable is $(\mathbf{s}_0, \psi_{b_m 0}) = (1.65, -1.45, 203^o, 325^o)$, where the last component is ψ_{b_m}, which is further given an arbitrary target value. We apply a sequence of target shapes starting from $(\mathbf{s}_1^*, \psi_{b_m 1}^*) = (1.3, -1.5, 60^o, 15^o)$ to $(\mathbf{s}_2^*, \psi_{b_m 2}^*) = (3.7, 0, 60^o, 15^o)$, then to $(\mathbf{s}_3^*, \psi_{b_m 3}^*) = (2.5, -1.8, 60^o, 15^o)$,
then to $(\mathbf{s}_4^*, \psi_{b_m 4}^*) = (0, -0.5, 60^o, 15^o)$, and finally to $(\mathbf{s}_5^*, \psi_{b_m 5}^*) = (0, 0.5, 60^o, 15^o)$, as illustrated in Figure 4.12. Note that $\mathbf{s}_4^*$ and $\mathbf{s}_5^*$ represent straight line target shapes of the cable, while the others represent slack target shapes with the corresponding poinclouds shown in Figure (4.11). The control law (4.9),(4.12) is applied with $\lambda = 0.1$.

First, as we examine the outer control loop, the simulation results demonstrate the successful convergence of the proposed control law (4.9),(4.12) during all tasks, as shown in Figure 4.13 for slack target shapes and in Figure 4.14 for taut target shapes. However, the error dynamic does not follow a perfectly exponential decay for slack targets due to noise from the quadrotor state coupled with the cable dynamics (4.13). This is due to the cable links slight oscillation subject to drone noise. It is interesting to note that this effect becomes negligible when the cable is stretched. Even in the presence of a noisy estimate of the current shape error $\mathbf{e}$ (see Figure 4.13(c),(e)) the desired velocity profiles (see Figure 4.13(d),(f)) are rather smooth due to the right choice of control gain λ of the proposed shape controller. It should be noted again that this noise of $\mathbf{e}$ is due to the drone state noise which is Gaussian.

Then, looking at the inner loop behaviour, it is clear from Figures 4.15 and 4.16 that the generated trajectory for the quadrotor outputs is feasible.

The demonstrated results show that we can successfully control the shape of the dynamical model of the cable (4.13) while the quadrotor remains stable. From now on, we test our controller in real experiment.

4.4.4 Experiment with one drone and a static camera

In this section we present the experimental results obtained for the first scenario in which only a single drone is used to move one extremity of the cable, the other being attached to a static tripod.

Experimental setup

For this scenario, the camera is fixed in the room in an eye-to-hand configuration. The quadrotor is made of a Mikrokopter autopilot board with IMU (Inertial Measurement

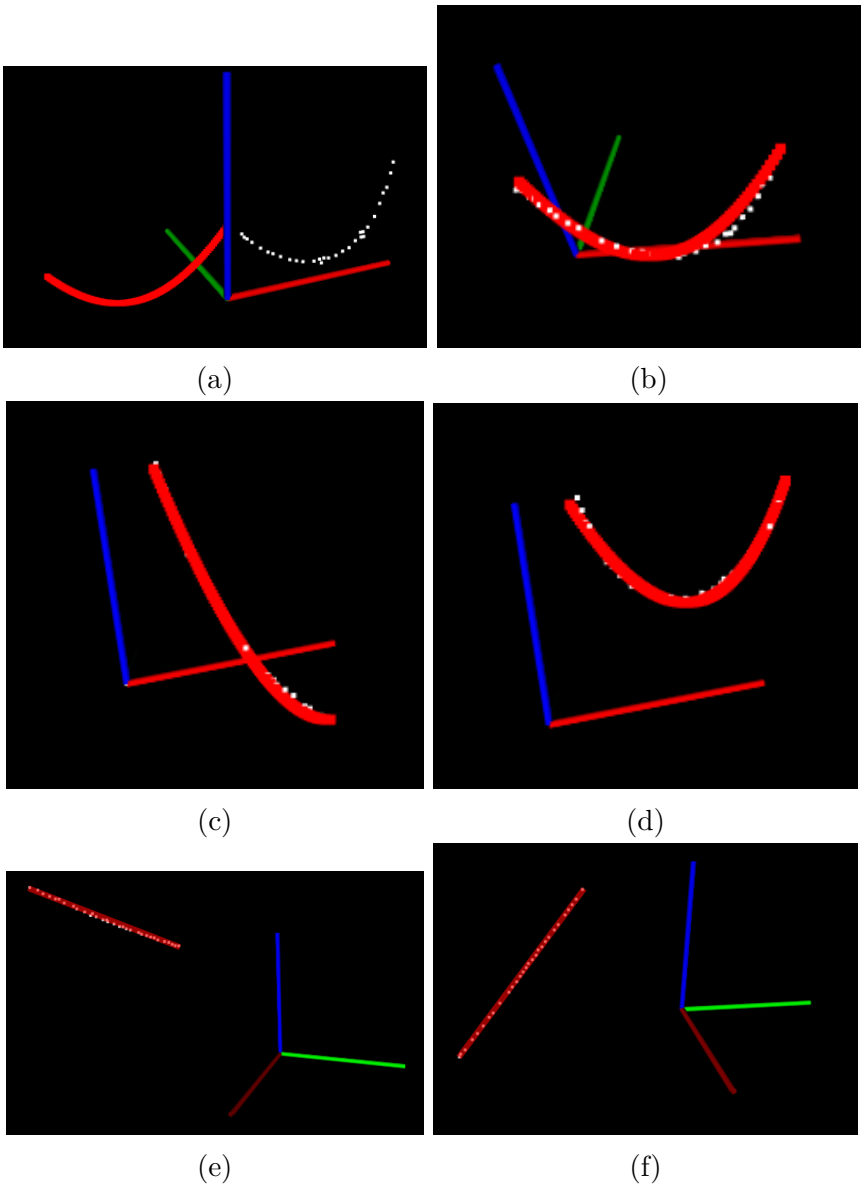


Figure 4.11 – Pointcloud view of the simulation in $\mathcal{F}_w$. The cable target shape is coloured red and the point cloud is coloured white at the beginning and end of each control task: (a) initial configuration $\mathbf{s}_0^*$, (b) convergence to $\mathbf{s}_1^*$, (c) convergence to $\mathbf{s}_2^*$ (from initial configuration $\mathbf{s}_1^*$), (d) convergence to $\mathbf{s}_3^*$ (from initial configuration $\mathbf{s}_2^*$), (e) convergence to $\mathbf{s}_4^*$ (from initial configuration $\mathbf{s}_3^*$), (f) convergence to $\mathbf{s}_5^*$ (from initial configuration $\mathbf{s}_4^*$).

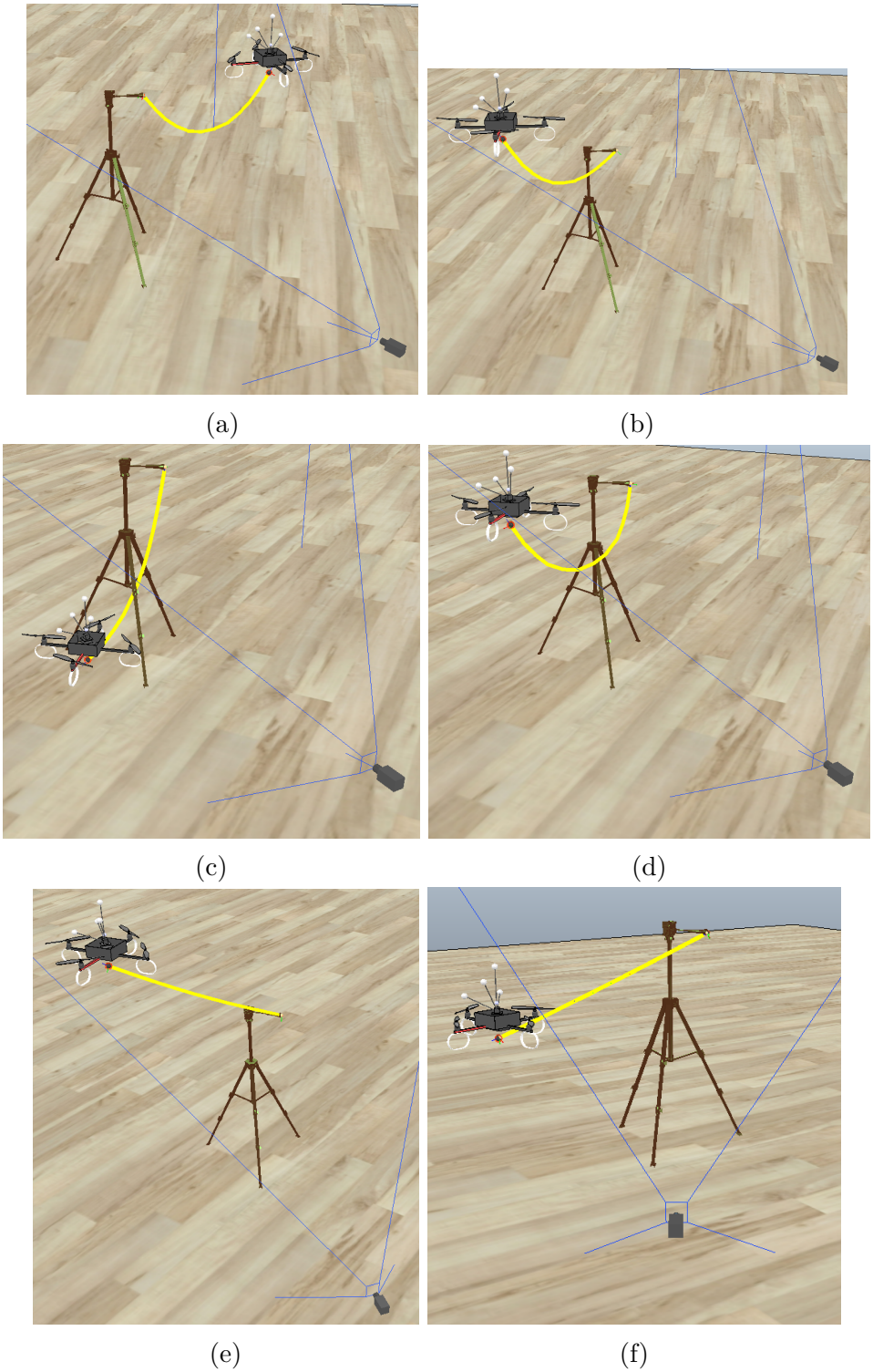


Figure 4.12 – Cable configurations at the beginning and at the end for each control task: (a) initial configuration $\mathbf{s}_0^*$ (b) convergence to $\mathbf{s}_1^*$, (c) convergence to $\mathbf{s}_2^*$, (d) convergence to $\mathbf{s}_3^*$, (e) convergence to $\mathbf{s}_4^*$, (f) convergence to $\mathbf{s}_5^*$.

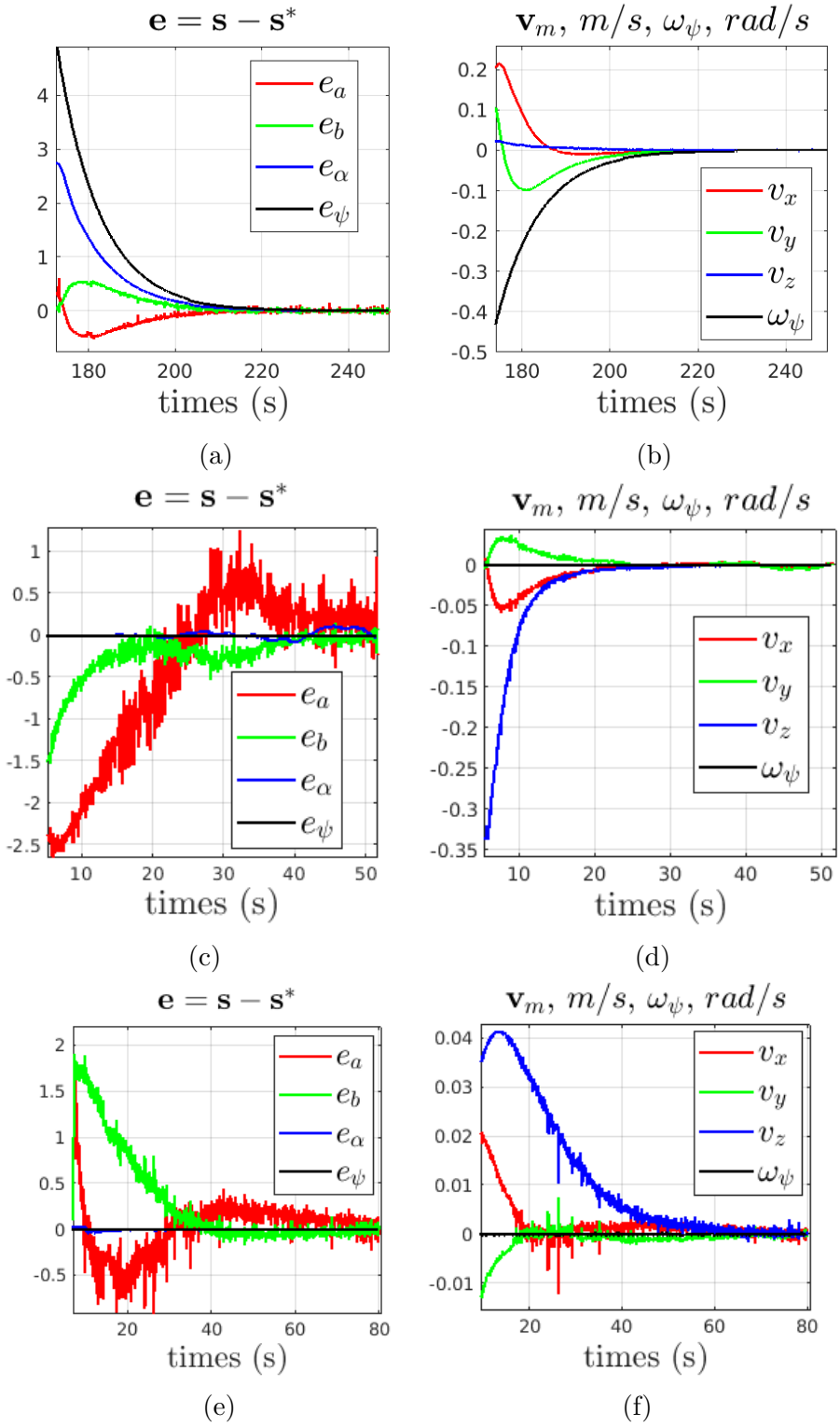


Figure 4.13 – Simulation results in Coppeliasim for slack cable shapes. Alphabetic order preserves the order of the slack cable target configurations.

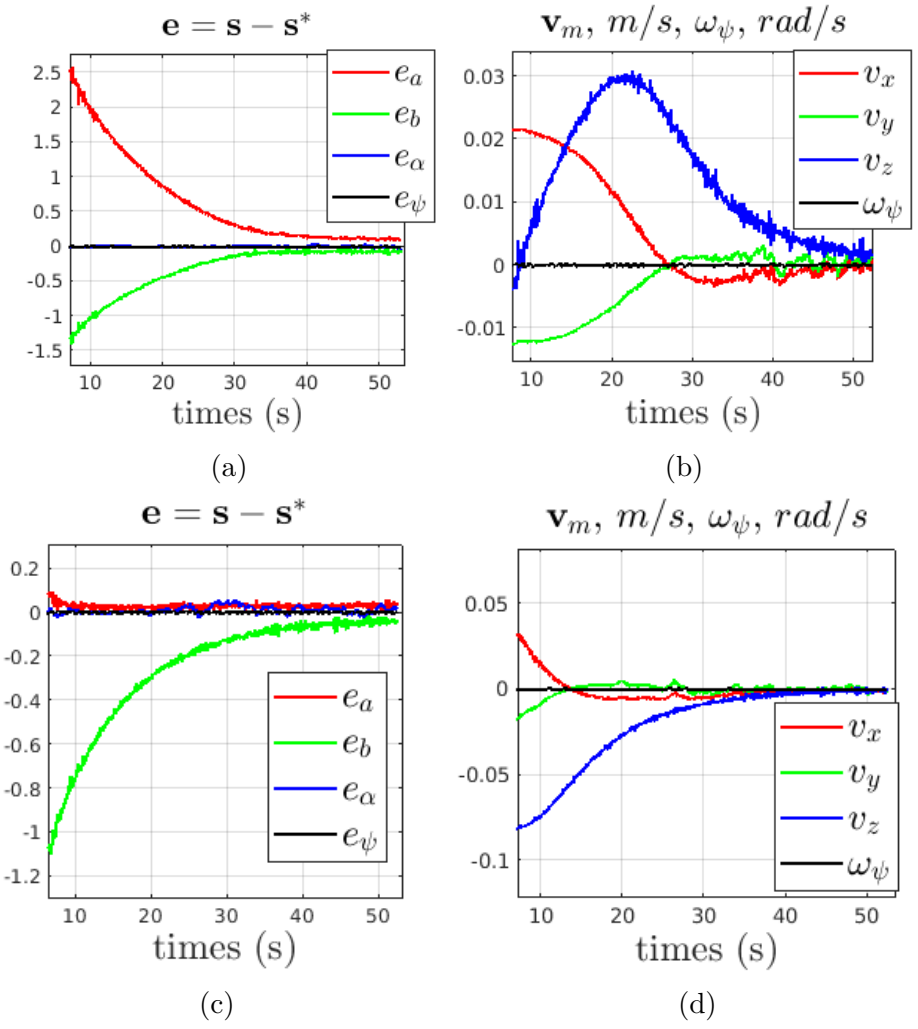


Figure 4.14 – Simulation results in Coppeliasim for taut cable shapes. Alphabetic order preserves the order of taut cable target configurations.

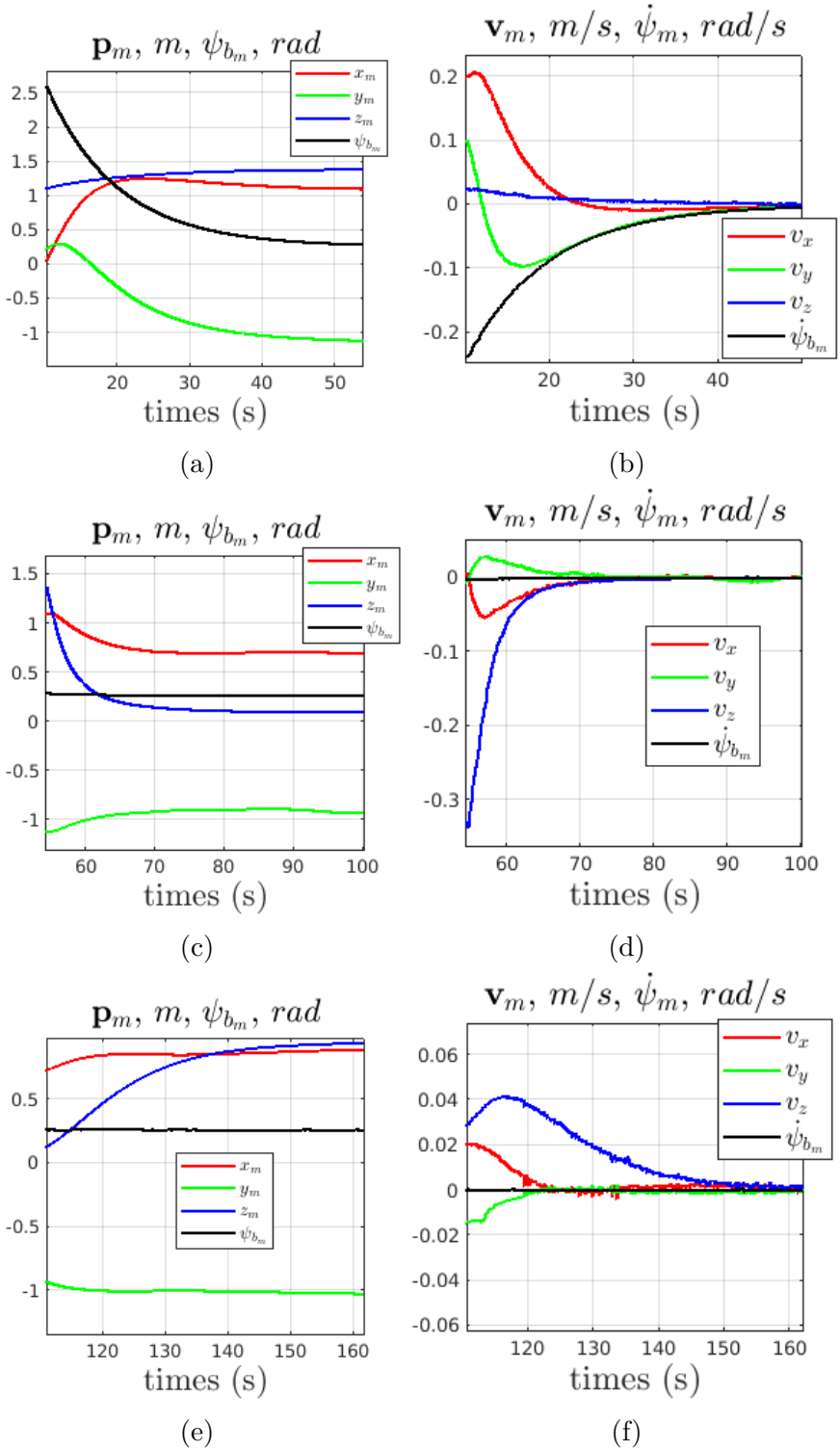


Figure 4.15 – Simulation results in Coppeliasim for slack cable shapes. Generated outputs at $200Hz$ by the KDTP (Maneuver) module: (a) $\mathbf{p}_m, \psi_{b_m}$ and (b) $\mathbf{v}_m, \dot{\psi}_{b_m}$.

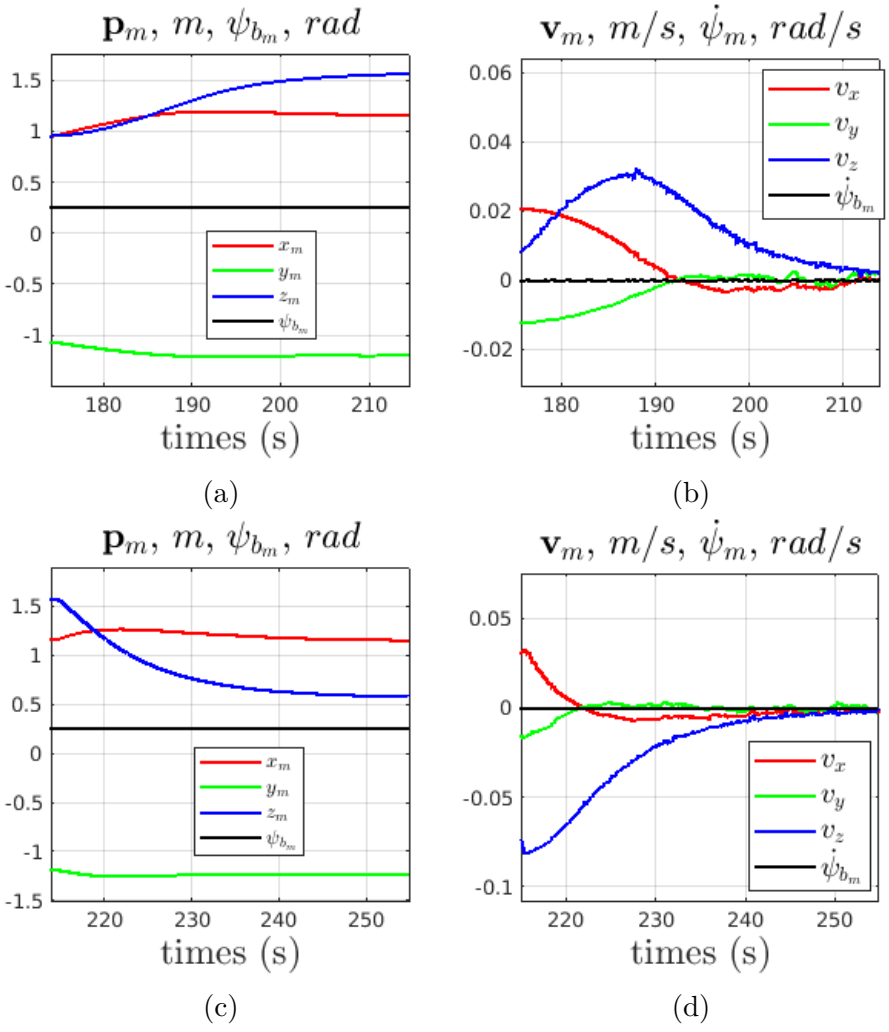


Figure 4.16 – Simulation results in Coppeliasim for taut cable shapes. Generated outputs at $200Hz$ by the KDTP (Maneuver) module: (a) $\mathbf{p}_m, \psi_{b_m}$ and (b) $\mathbf{v}_m, \dot{\psi}_{b_m}$ by the KDTP (Maneuver) module.

Unit) and is connected to the electric motor speed controllers. This board is connected via a serial bus to a NVIDIA Jetson TX2 onboard computer, where the inner-loop (or low-level) control layer is running. The component of the same name ("mikrokopter", see Figure 4.10) communicates with the autopilot board, the velocities of the propellers and the sensor readings. The state of the quadrotor is estimated using its IMU and an optical tracking system (see Figure 4.10 and Figure 4.7). Therefore, the tracking system frame is chosen as the world frame $\mathcal{F}_w$, but it could also be the frame related to the static tripod (in case we do not want to use the optitrack system to measure the pose of the drone but another camera mounted on the drone for visual odometry). Similarly to the experiments in Chapter 3, the RGB-D camera used is the Intel RealSense D435 with the same resolution. Its extrinsic parameters ${}^{w}\mathbf{H}_{c}$ are calibrated using the method proposed in [R. Tsai, 1987] similar to what was done in Chapter 3. This time an AprilTag [Olson, 2011] was attached to a rigid planar object that was also localised by the optitrack system in contrast to the robot end-effector pose given by its joints that was used in Chapter 3. The planar object to which the AprilTag is attached was then manually moved to different positions in front of the camera to collect the measures required for the calibration. The coordinate of the cable attachment point in the quadrotor body frame ${}^{b_m}\mathbf{p}_m = (0, 0, -0.13)$ was measured manually and was supposed to be constant during all experiments. Note that the attachment point is not directly in the drone COG due to the presence of the battery and the Jetson TX2 that balance the position of the COG closer to the barycenter of the drone. The cable length is chosen to be $L = 1.6m$.

Starting from the initial configuration $(\mathbf{s}_0, \psi_{b_m 0}) = (4.8, -4.5, 11^o, 15^o)$, a sequence of target shapes is defined as follows: $(\mathbf{s}_1^*, \psi_{b_m 1}^*) = (3.8, -3.6, 0^o, 0^o)$, $\mathbf{s}_2^* = (1.5, -1.8, 60^o)$, $\mathbf{s}_3^* = (1.4, -1.4, 40^o)$ and finally $\mathbf{s}_4^* = (0.25, 0.12, 38^o)$, see Figure 4.17(b). The control gain is chosen the same as in the simulation $\lambda = 0.1$.

Experimental results

The obtained results (see Figure 4.17(a),(b)) show even better error convergence with the real cable than with its digital twin as it is clear that the shape error dynamics better approaches an exponential decrease rate. However, there is some noise in the estimate of the shape error (see Figure 4.17(a)). We think this is due to the unmodeled behaviour of the cable such as local and fast oscillations of the cable inside and outside its plane and the drone dynamics. However, these perturbations do not destabilise the shape controller.

Now, it is certainly clear from the demonstrated results that we can efficiently deform

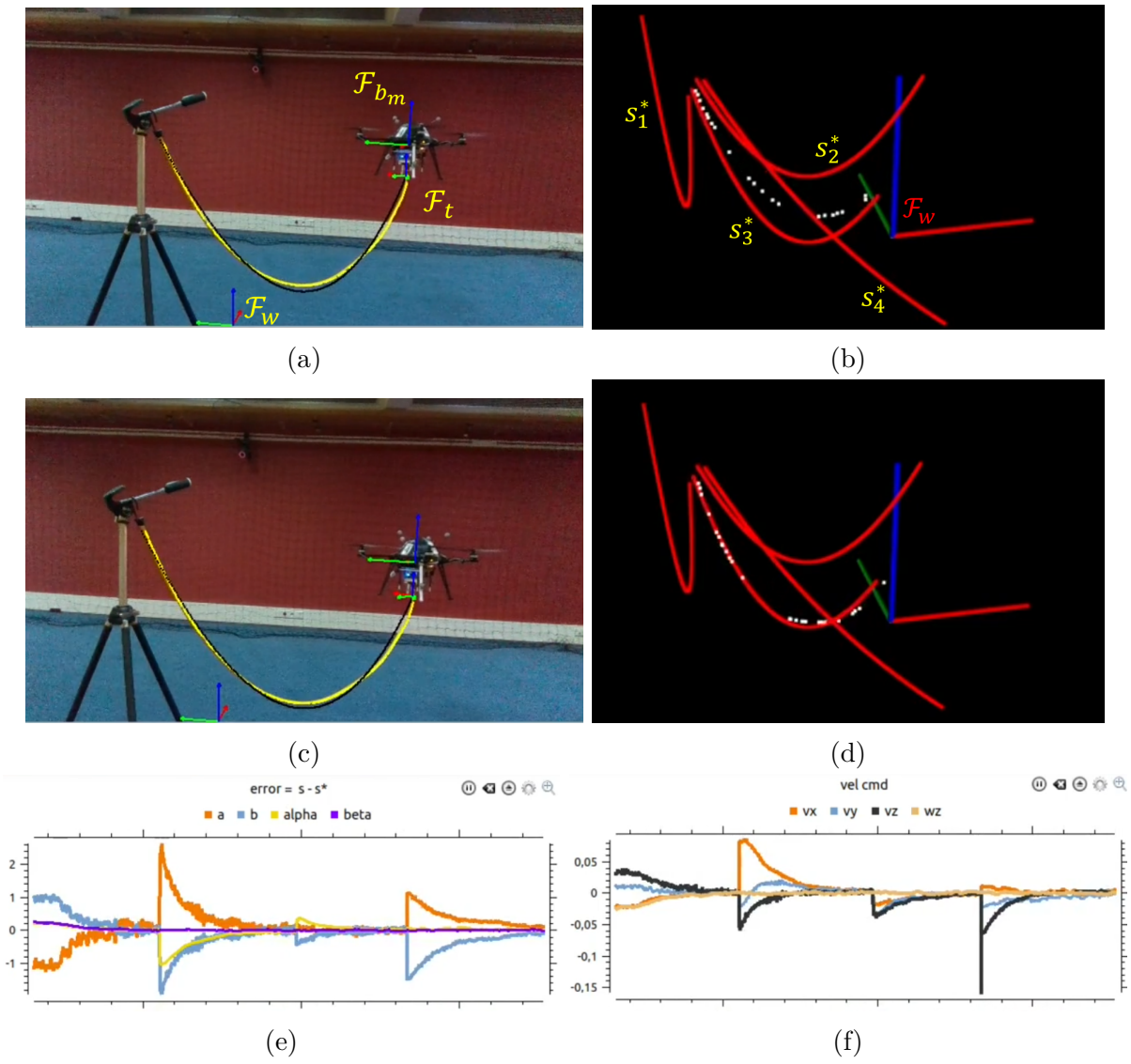


Figure 4.17 – Experiment of the one drone shaping cable task. The video is available at https://youtu.be/OMs7ycb6rzQ. (a),(c) RGB view of the static camera ((c) at the end of the task), segmented cable points in yellow, fitted parabola model in black. (b),(d) Point cloud view with target shapes $\mathbf{s}_i^*$ in red, white points represent 3D cable points that remain after processing the pointcloud. (c) Shape error during servoing, "beta" stands for drone yaw ψ_{b_m} which is given some arbitrary value. (f) Control velocity $\mathbf{v}_m^*, \dot{\psi}_{b_m} = \omega_\psi$ (denoted as "wz") during shape servoing.

a cable acting on its end point. Moreover, our proposed model demonstrates close to perfect convergence of the shape error in the experiments, which means that such an approximation of complex cable dynamics is a valuable choice and is not a coincidence, as can be seen in Chapter 2 of this work, where it was shown that a simple parabola is a suitable model for a slack heavy cable with small deflection and large span. We proceed further by demonstrating the manipulation of the cable by two drones with an interesting application to load transportation.

4.5 2nd scenario: cable manipulation by two drones using eye-in-hand visual servoing

In this section, we model the cable shaping task by two drones, one being teleoperated with a joystick by a human operator. Similarly to [Laranjeira et al., 2020], a leader-follower control strategy of the two drones is proposed. The master drone (the leader) $\mathbf{b}_f$ (see Figure 4.18) is teleoperated using a joystick that sends a linear velocity command and a yaw rate using the same inner-loop control (4.9). The idea is to allow a human operator to control the rigid motion of the cable, while the second drone (the follower) performs a shape visual servoing task to autonomously control the deformation of the cable. The cable is attached between the two drones. One extremity of the cable is held by a first drone that we call the master drone, which is fully teleoperated with a joystick. The other extremity is held by a second drone, named follower drone, which is fully autonomously controlled by our shape visual servoing using an RGB-D camera embedded on this drone. For this second scenario, which relies on an eye-in-hand configuration, an additional feature is considered with the previous set of visual features in the control law. It corresponds to the yaw angle of the follower drone that we propose to also control to maintain the cable in the field of view of the camera during the cable shaping task. We demonstrate by experiments that such system allows to teleoperate the master drone toward a box object located on the floor in order to grasp it by passing the cable underneath a hook attached to the object while the follower drone maintains the desired shapes of cable. Finally, we demonstrate by experiments that the proposed approach is efficient for grasping and transporting an object.

4.5.1 Modelling and control of the system

The shape of the cable changes when the position of its end points changes, as demonstrated by (3.24). This model only accounts for the relative position of the cable end points for the change in the shape of the cable and does not consider the rigid motion of the cable. To deal with the rigid motion of the cable, we use the following model simply derived from (3.21):

$$\mathbf{v}_m = \mathbf{v}_f - \mathbf{M}\dot{\mathbf{s}} \tag{4.15}$$

where $\mathbf{v}_f$ is the linear velocity of the COG of the master quadrotor $\mathbf{b}_f$. The drone $\mathbf{b}_m$ is controlled by shape visual servoing with an embedded camera mounted on the drone $\mathbf{b}_m$ (see Figure 4.18). It is assumed that both drones communicate their velocity expressed in $\mathcal{F}_w$. The choice of $\mathcal{F}_w$ depends on the available measures of the drone pose. It could also be the frame of a tag attached on the box to be grasped with the cable. In that case, α^* could be defined such that the cable passes underneath the hook attached to the box. This case is of particular interest for autonomous grasping without operator command and optitrack system while relying only on the onboard visual odometry (*e.g.* tracking the tag by the leader in order to approach the box and performing visual-inertial SLAM after grasping is finished). In our experiments we considered the optitrack frame as $\mathcal{F}_w$ without loss of generality. The velocity $\mathbf{v}_f$ of the leader drone $\mathbf{b}_f$ is used as a feed-forward

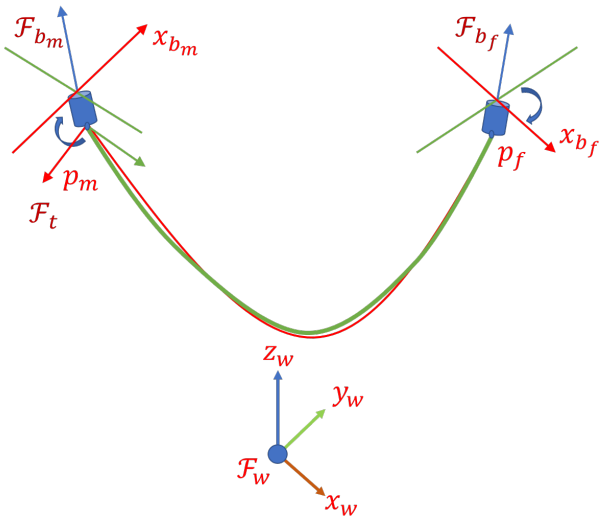


Figure 4.18 – Two drones holding a cable. The parabola is depicted in red. The blue cylinders with a circle represent passive ball joints. $\mathcal{F}_t$ stands for the cable frame, $\mathcal{F}_{b_m}$ for the follower drone body frame, $\mathcal{F}_{b_f}$ for the leader drone body frame.

term (see Figure 4.20) in the control loop of the drone $\mathbf{b}_m$, allowing it to compensate for this motion. Meanwhile, because of non-modelled perturbations, it is possible to have a steady-state error, especially at the end of the shaping task, when the control velocity is small. To compensate for this, an integral term is used. Finally, we propose the following outer loop control law for the follower drone $\mathbf{b}_m$, which easily generalises the system model for the case where only one drone is used, given by:

$$\begin{aligned} \mathbf{v}_m^* &= \mathbf{v}_f + \lambda \mathbf{M}\mathbf{e} + \mu \mathbf{M} \sum_i^N \mathbf{e}_i \\ \dot{\psi}_{b_m}^* &= -\lambda(\psi_{b_m} - \psi_{b_m}^*) - \mu \sum_i^N (\psi_{b_m i} - \psi_{b_m}^*) \end{aligned} \tag{4.16}$$

The first equation in (4.16) provides the target velocity of the follower drone for the inner loop controller (see Figure 4.20). The second equation in (4.16) provides the control law for the yaw angle ψ_{b_m} of the drone $\mathbf{b}_m$, such that the cable remains in the center of the image when the camera is mounted on the drone $\mathbf{b}_m$ (see Figure 4.21). Note that the yaw motion of the drone is decoupled from the yaw motion of the cable α due to the passive ball joint that attaches the cable extremity to the drone. In order to maintain the best visibility of the cable in the center of the camera RGB image we align the projection of the cable in such a way that the optical axis $\mathbf{z}_c$ of the camera is coplanar to the cable frame axis $\mathbf{y}_t$ as illustrated in Figure 4.19. Note that when the axes $\mathbf{z}_c$ and $\mathbf{y}_t$ are aligned, the cable profile seen on the RGB image represents a straight line and not a parabola which is not problematic for tracking the features a, b since we use the depth data provided by the depth camera. In practice we set $\psi_{b_m}^* = \alpha - \frac{\pi}{4}$ for the experimental setup with a real drone.

The last term in (4.16) is composed of the integral gain μ that needs to be tuned. A sliding window approach is used to compute the cumulative error $\sum_i^N \mathbf{e}_i$ and $\sum_i^N (\psi_{b_m i} - \psi_{b_m}^*)$ from N previous control iterations. As the sliding window forgets old errors, it is therefore well suited to robotic tasks involving reaching a succession of different desired cable shapes $\mathbf{s}^*$.

4.5.2 Experimental setup

We used the same quadrotor drones as in the previous scenario (note that for this scenario, the Jetson TX2 also runs the shape visual servoing module). This time, the

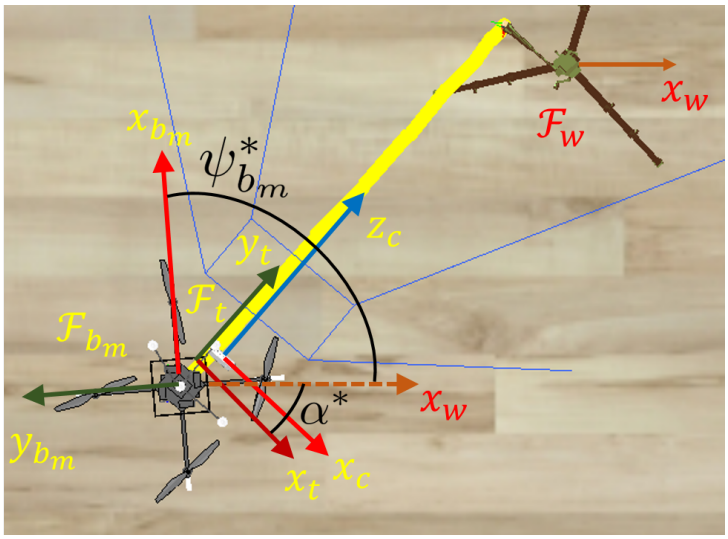


Figure 4.19 – Illustration of the top view of the eye-to-hand camera setup on the drone in the simulator. Note that it is important to compute the desired angle $\psi^*_{b_m}$ so that the rotation follows the shortest path.

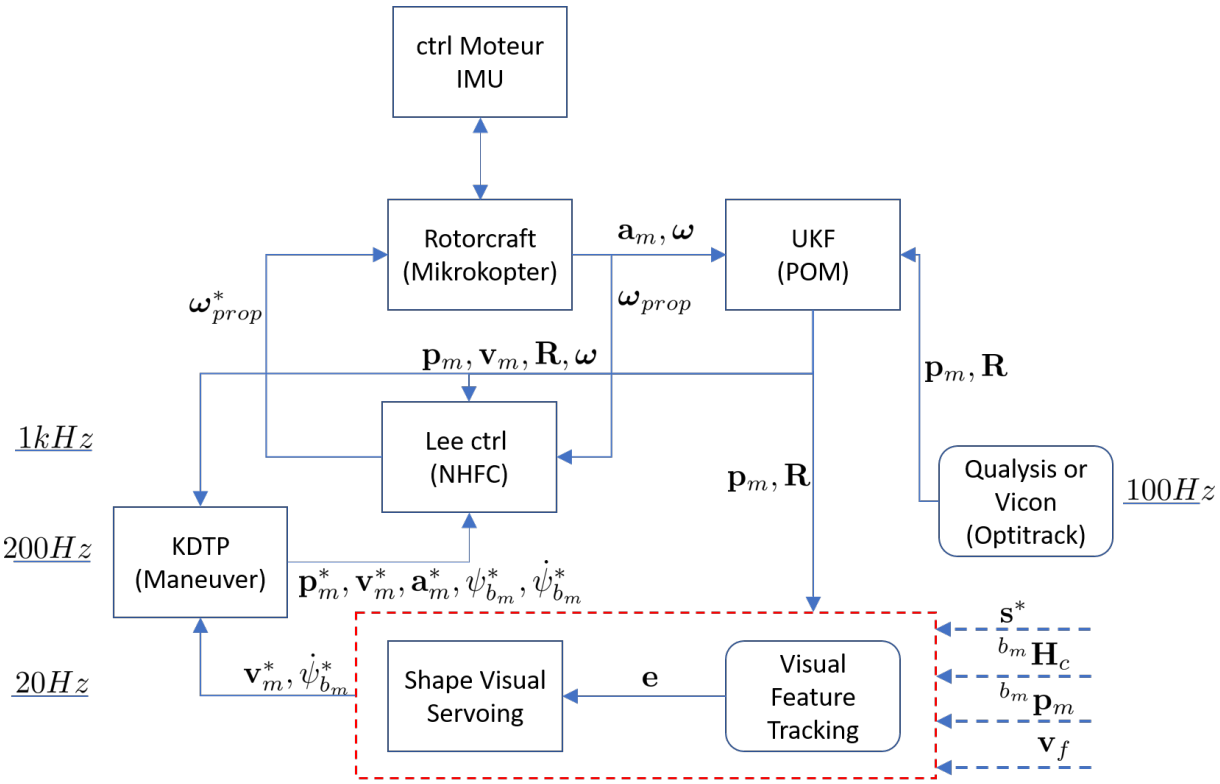
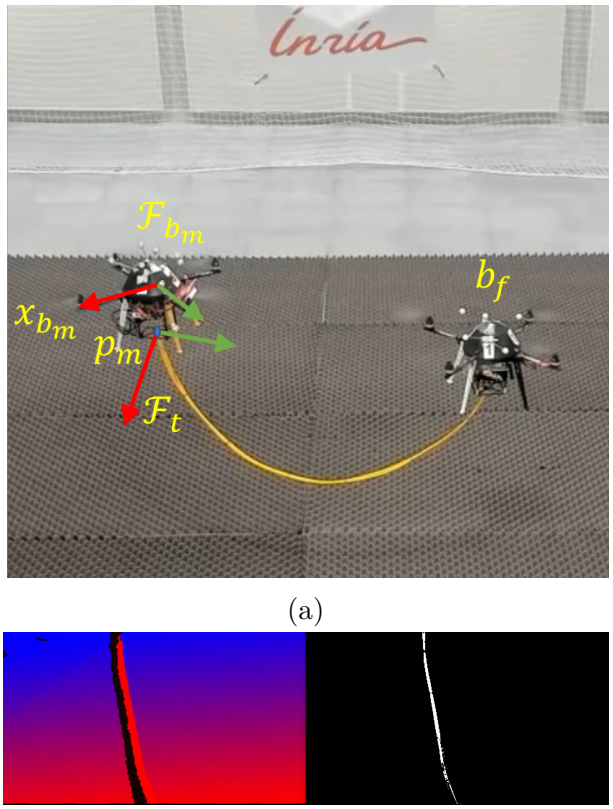


Figure 4.20 – Scheme of the quadrotor associated to the middleware control layer depicted in Figure 4.10. The terms ${}^{b_m}\mathbf{H}_c$ and ${}^{b_m}\mathbf{p}_m$ (together with $\mathbf{p}_m, \mathbf{R}$) are used to reconstruct the origin of the tether frame $\mathcal{F}_t$ to estimate the features a, b.



(a)

(b)

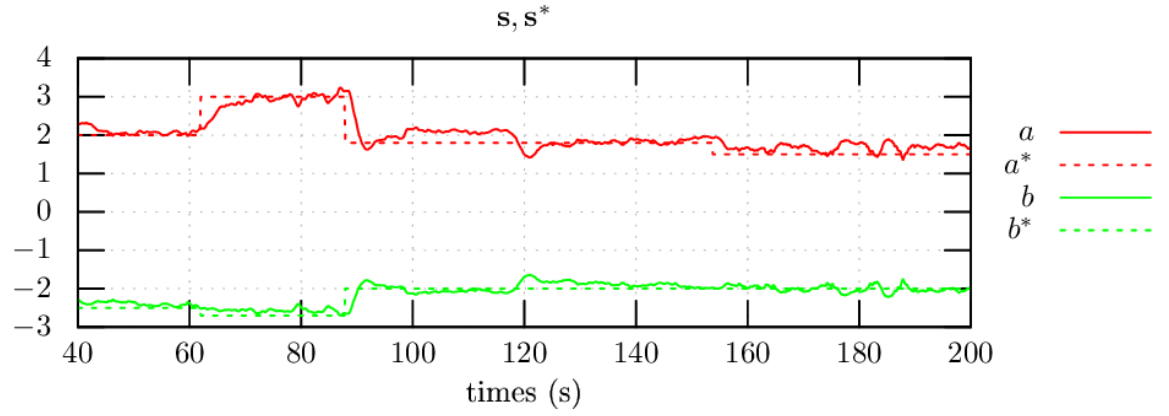
Figure 4.21 – (a) Two drones holding a yellow cable. $\mathcal{F}_t$ stands for the cable frame, $\mathcal{F}_{b_m}$ for the body frame of the follower drone which is controlled by visual servoing, $\mathbf{p}_m$ is the cable attachment point and origin of $\mathcal{F}_t$, b_f is the leader drone that is teleoperated. (b) View from the onboard RGB-D camera (left drone), colour depth image on the left and segmented RGB image on the right. Note that the image view of the cable is almost a straight line making impossible to track its shape using only the RGB image.

Intel RealSence D435 camera of resolution 848×480 pixels is mounted on the quadrotor $\mathbf{b}_m$ with a 3D printed support. The camera is calibrated once using the same method as in the previous scenario for obtaining ${}^{b_m}\mathbf{H}_c$, which is constant (see Figure 4.20). The first two experiments will show hereafter the influence of the integral term in (4.16) on the system behaviour, therefore allowing to compare the results with and without the integral term.

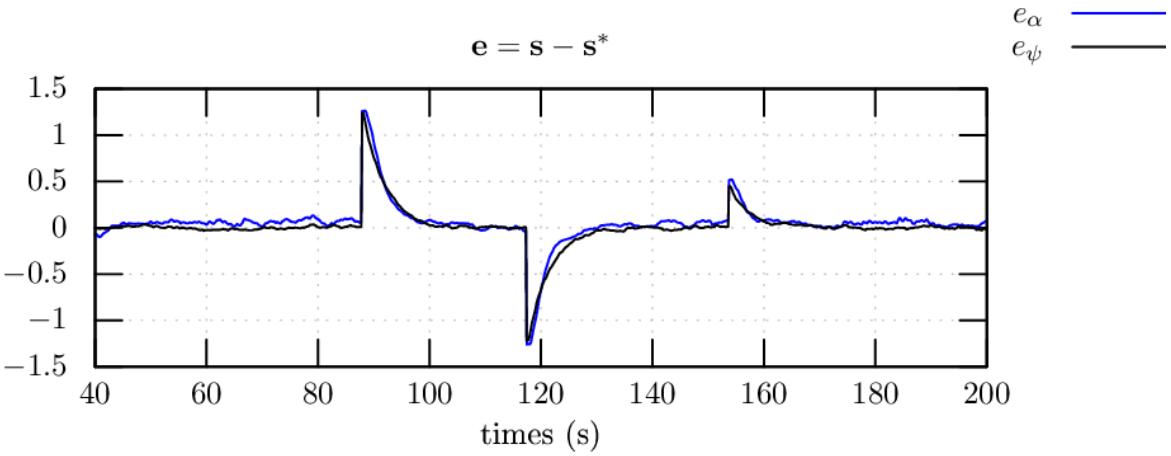
4.5.3 Experimental results

In this section, we demonstrate slack cable manipulation results while teleoperating the drone $\mathbf{b}_f$ (see Figure 4.21). Experiment videos are available at https://youtu.be/kZWMwDrdINs for the first experiment and at https://youtu.be/dAdMxZFH5fk for the second experiment. In the first experiment, the sequence of target shapes is as follows without using the integral gain $\mu = 0, \lambda = 0.25$: $\mathbf{s}_0 = (1.1, -1.3, 186^o, 142^o)$, $\mathbf{s}_1^* = (2, -2, 170^o, 125^o)$, $\mathbf{s}_2^* = (2, -2.5, 170^o, 125^o)$, $\mathbf{s}_3^* = (3, -2.7, 170^o, 125^o)$, $\mathbf{s}_4^* = (1.8, -2.0, 100^o, 55^o)$, $\mathbf{s}_5^* = (1.8, -2.0, 170^o, 125^o)$, $\mathbf{s}_6^* = (1.5, -2.0, 145^o, 140^o)$. Figure 4.22(a) shows the presence of some steady-state error slowly converging to the reference in the time interval $[120, 160]s$ for b and $[100, 120]s$ for a. The quadrotor yaw and α angular errors converge exponentially, as can be seen in Figure 4.22(b), which demonstrates a perfect decoupling between them. Figures 4.23 and 4.24 show the comparison of the control signals in the inner and outer loop provided by the "Maneuver" and "Shape Visual Servoing" (SVS) blocks. It is hard to notice any difference between the curves meaning that interpolation by splines works well (see Figure 4.23). On Figure 4.24(a) one can notice a slight difference in $v_{z_m}^*$, meaning that the actual vertical velocity measure contains more noise. It is clear that the $\mathbf{v}_m^*$ and $\psi_{b_m}^*$ signals are accurately approximated by the "Maneuver" block, and the quadrotor executes exactly the shape visual servoing control output. The last Figure 4.25(a) shows the reference trajectories of the quadrotor outputs generated by the "Maneuver" block while the drone is performing the cable shaping task and compensating for the velocity of the leader drone $\mathbf{b}_f$ received from the joystick. The analysed curves demonstrate the efficiency of the proposed approach to deform a slack cable into a desired shape using drones though we need to deal with a small steady-state error.

The second experiment is then carried out with the integral term to reject the steady-state error. The integral gain is $\mu = 0.025$ and window size $N = 10$, $\lambda = 0.25$. The sequence of target shapes is now: $\mathbf{s}_0 = (1.86, -1.87, 177^o, 136^o)$, $\mathbf{s}_1^* = (2, -2, 170^o, 125^o)$,

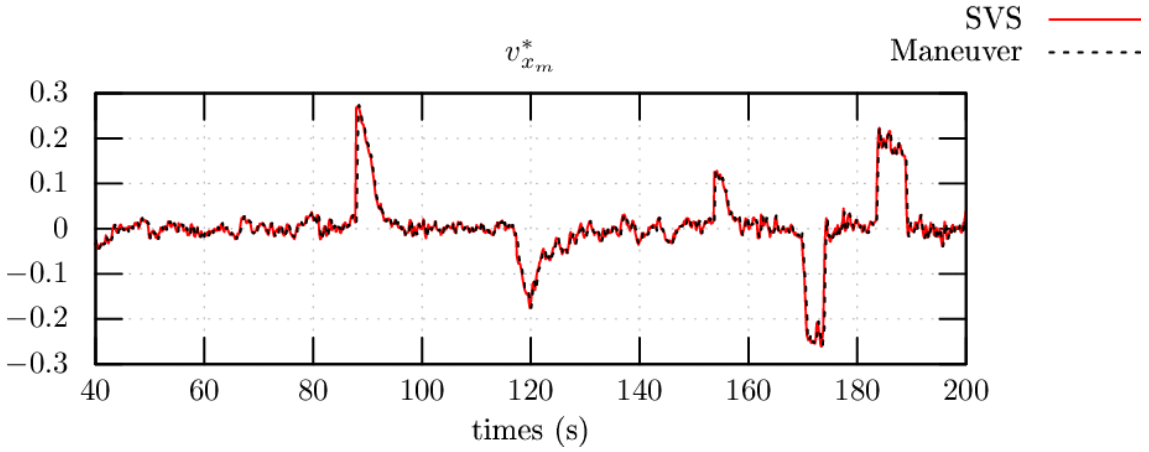


(a)

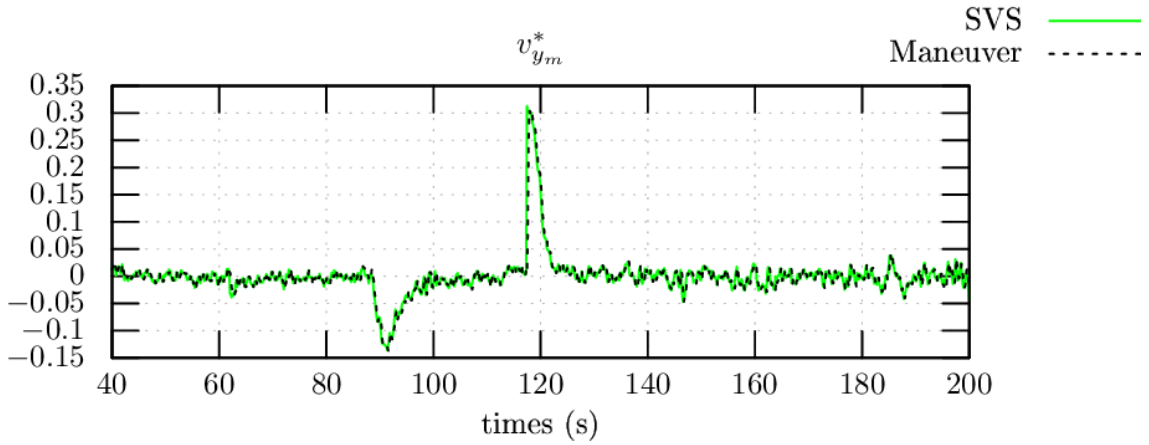


(b)

Figure 4.22 – Experimental results for visual features and yaw angle of the drone without the integral term (video of the experiment available at https://youtu.be/kZWMwDrdINs). (a) Evolution of the desired and current parabola coefficients. (b) Evolution of the error for α and ψ_{b_m} in radians.

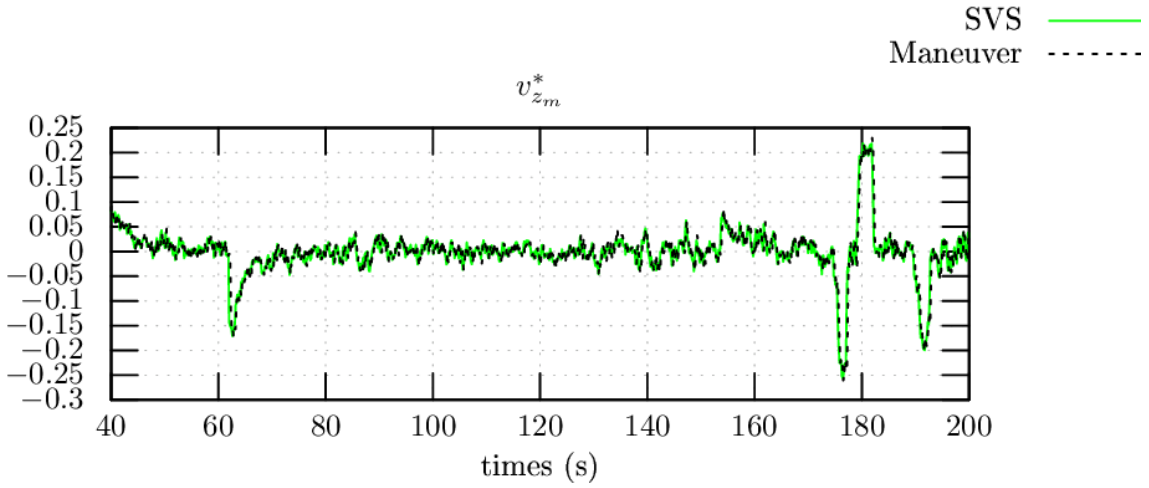


(a)

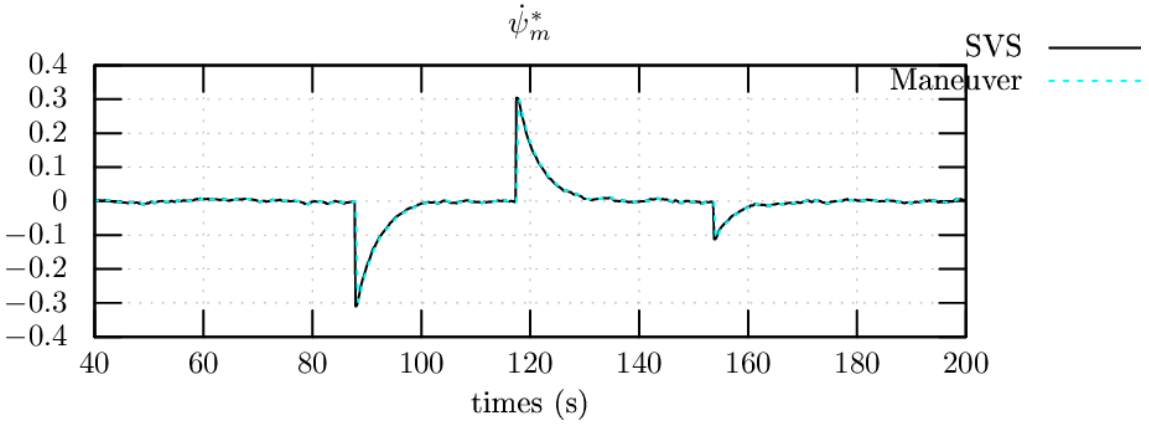


(b)

Figure 4.23 – Trajectories generated by the Maneuver module approximating SVS.

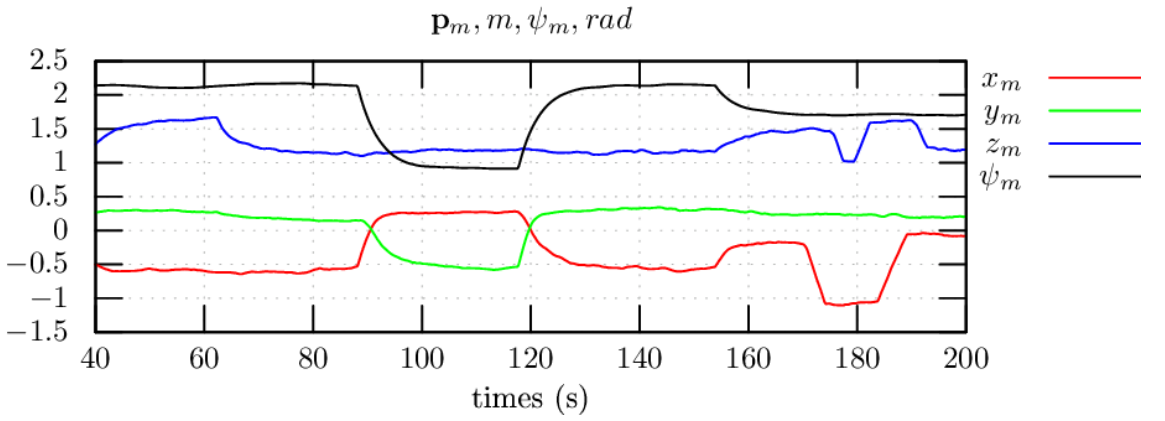


(a)

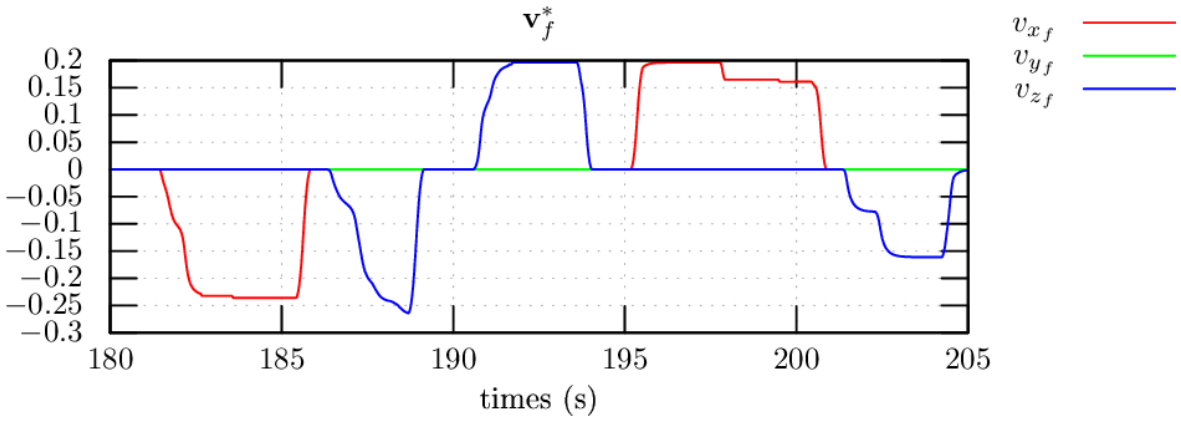


(b)

Figure 4.24 – Trajectories generated by the Maneuver module.



(a)



(b)

Figure 4.25 – Reference trajectories generated by the Maneuver module. (a) For drone $\mathbf{b}_m$. (b) Feedforward term used during the control of the follower drone $\mathbf{b}_m$ corresponding to the velocity of the leader drone $\mathbf{b}_f$.

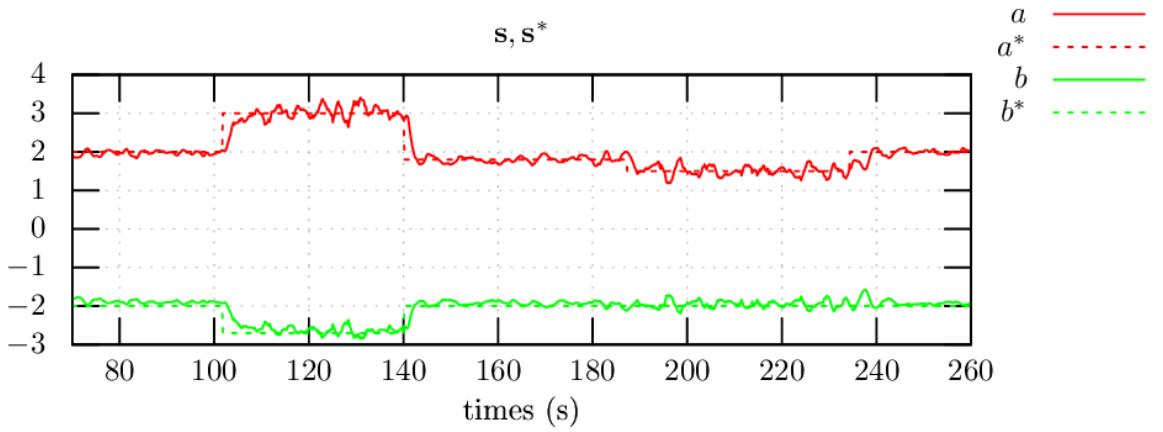
$\mathbf{s}_2^* = (3, -2.7, 145^o, 100^o)$, $\mathbf{s}_3^* = (1.8, -2, 145^o, 100^o)$, $\mathbf{s}_4^* = (1.5, -2, 145^o, 100^o)$, $\mathbf{s}_5^* = (2, -2, 170^o, 125^o)$. One can see the complete rejection of the steady-state error (Figure 4.26(a)) but also the appearance of damped oscillations for all features shown in Figure 4.26(a),(b), which influences the reference trajectory of the drone outputs with slight oscillations, as can be seen in Figures 4.27, 4.28 and 4.29. The leader drone $\mathbf{b}_f$ does not move during this cable shaping task. These plots demonstrate the efficiency of the approach. One should pay attention to the integral gain tuning process (μ, N), which is not trivial to find the best. The observability of the cable by the camera also plays an important role in the efficiency of the shaping task.

We demonstrated that two flying drones attached to a cable can efficiently deform it into some desired shape during flight in the presence of various disturbances and uncertainties. Our proposed approach is efficient for deforming the cable, even if only the cable is only partially observed by the camera (as shown before in Figure 4.4). In the following, we demonstrate a load manipulation task with our system.

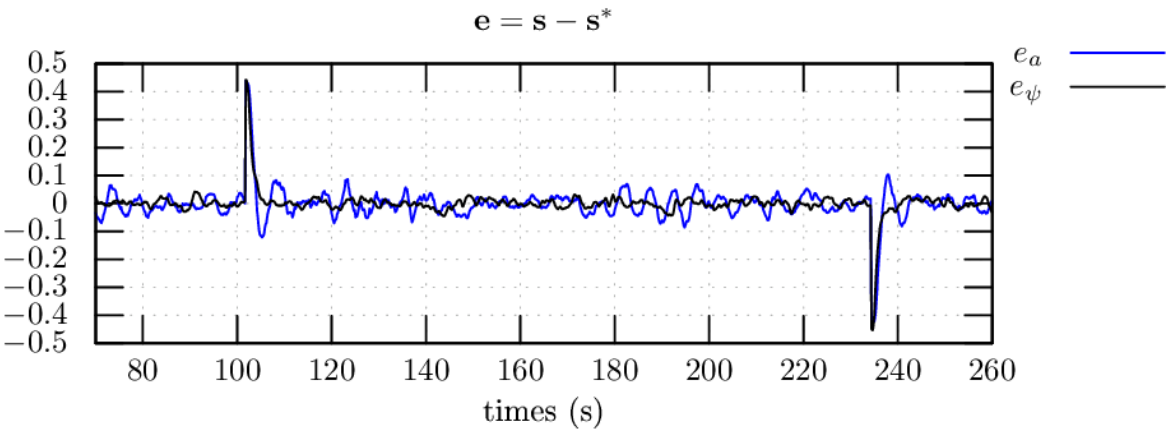
Box grasping and manipulation with cable

During this last experiment, the control law parameters are chosen as follows: $\lambda = 0.25$, $\mu = 0.05$, $N = 5$. The goal is to lift and transport a box of mass $m_{box} = 0.3kg$. At the beginning of the experiment, the box is attached to the floor with duct tape to prevent the box from blowing away by the air flow of the propellers. The cable is then used to grasp and manipulate this box (see Figure 4.30). A video of the experiment is available at https://youtu.be/849KGUYceoo. The drone $\mathbf{b}_m$ autonomously adjusts the shape of the cable to maintain a desired shape $\mathbf{s}^*$. We show the robustness of the proposed control scheme to the tension in the cable due to the unknown mass of the box, as well as when grasping and lifting the box. The task consists of several actions, the first of which is the launch of the two drones, and the second the activation of the shape visual servoing controller when $\mathbf{b}_m$ becomes autonomous. Then the leader drone $\mathbf{b}_f$ is moved with a joystick towards the box with a hook attached. When the cable passes under the hook, a vertical velocity is sent to the drone $\mathbf{b}_f$ to fly up, therefore lifting the box. Then a variation of the desired features values is done to see how the system maintains the cable shape, and finally the release of the box is performed in another place of the room.

As can be seen in Figure 4.31 the shape of the cable is well maintained with dumped oscillations at the moments of lifting and releasing the box. This is due to the significant and fast deformations of the cable during the lift and release phases. The cable plane yaw

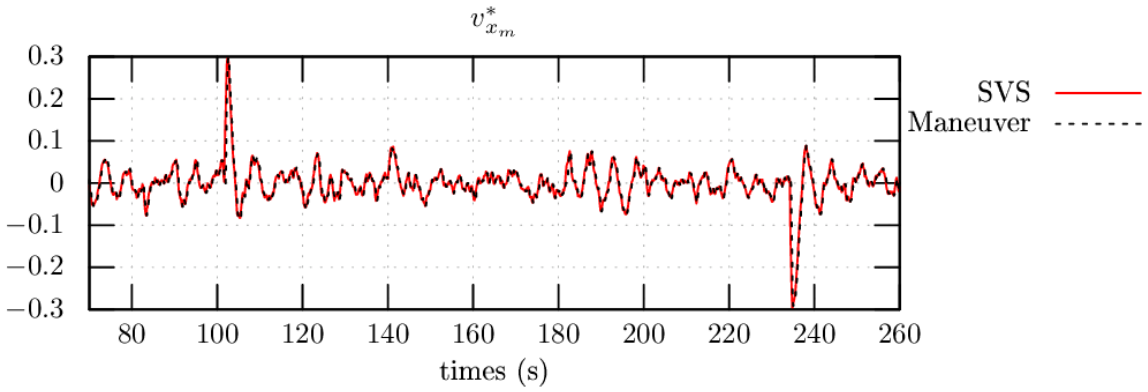


(a)

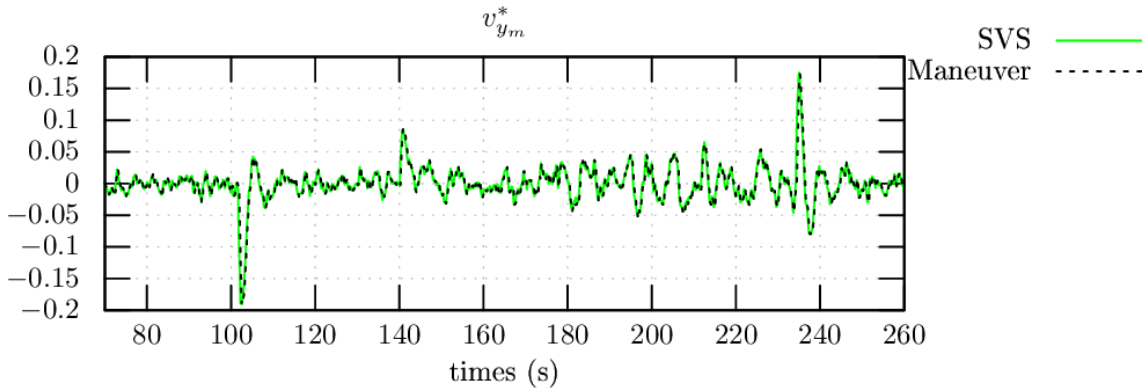


(b)

Figure 4.26 – Experimental results for visual features and yaw angle of the follower drone using an integral term (video of the experiment available at https://youtu.be/dAdMxZFH5fk). (a) Desired and estimated parabola coefficients. (b) Error for α and ψ_{b_m} in radians.

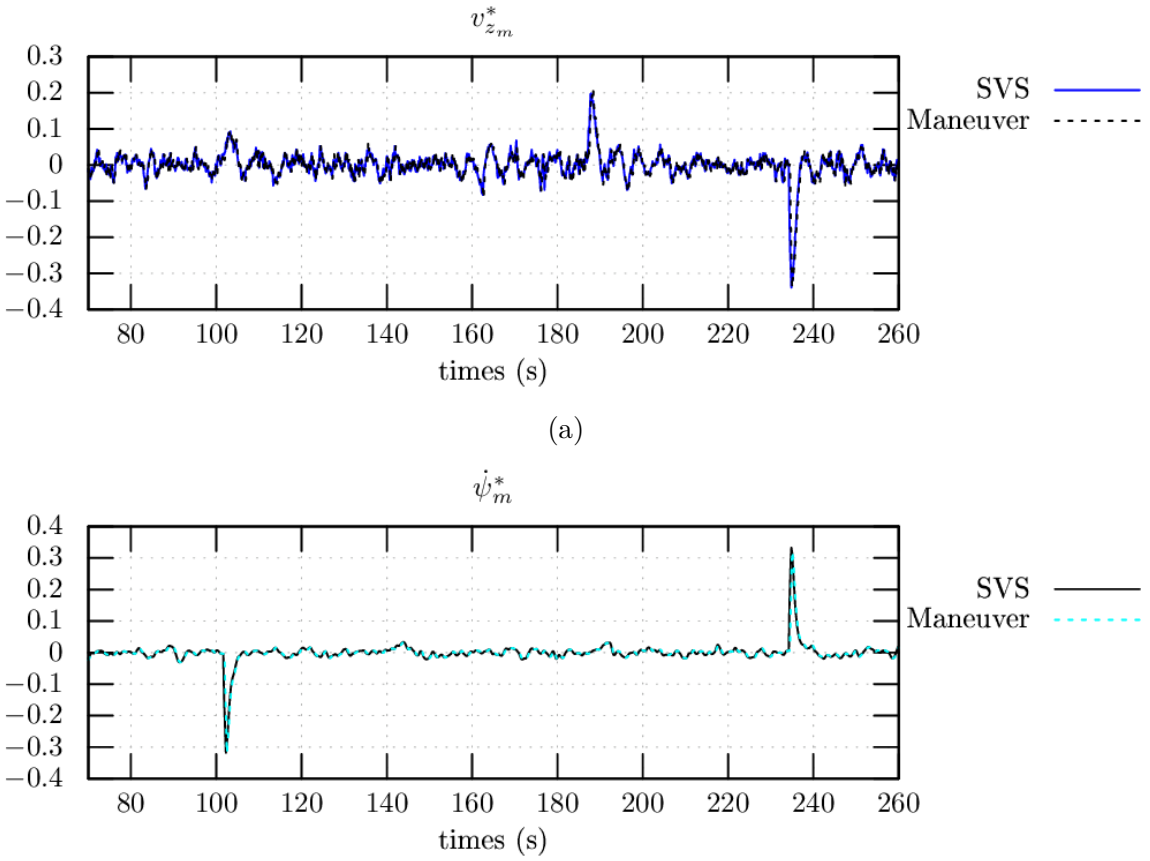


(a)



(b)

Figure 4.27 – Trajectories generated by the Maneuver module approximating SVS. It is hard to notice any difference between the curves meaning that interpolation by splines works well.



(a)

(b)

Figure 4.28 – Trajectories generated by the Maneuver module. (a) One can notice a slight difference in $v^*_{z_m}$, meaning that the actual vertical velocity measure contains more noise.

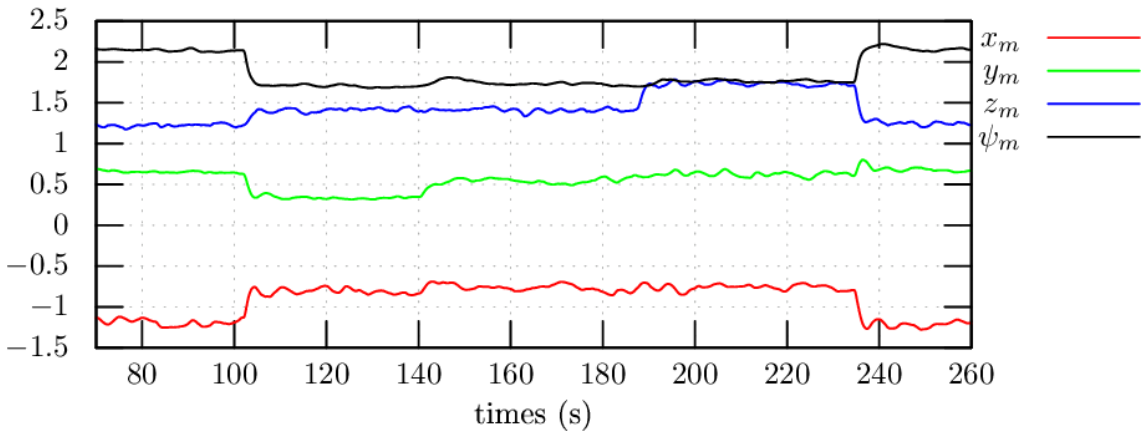


Figure 4.29 – Reference trajectories generated by the Maneuver module.

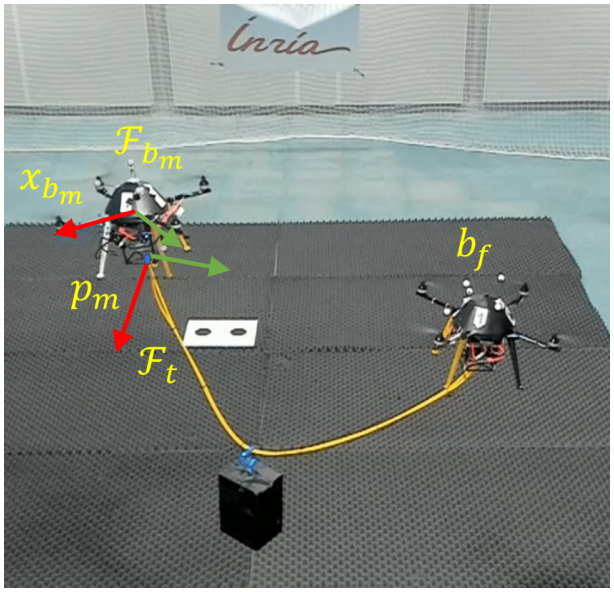
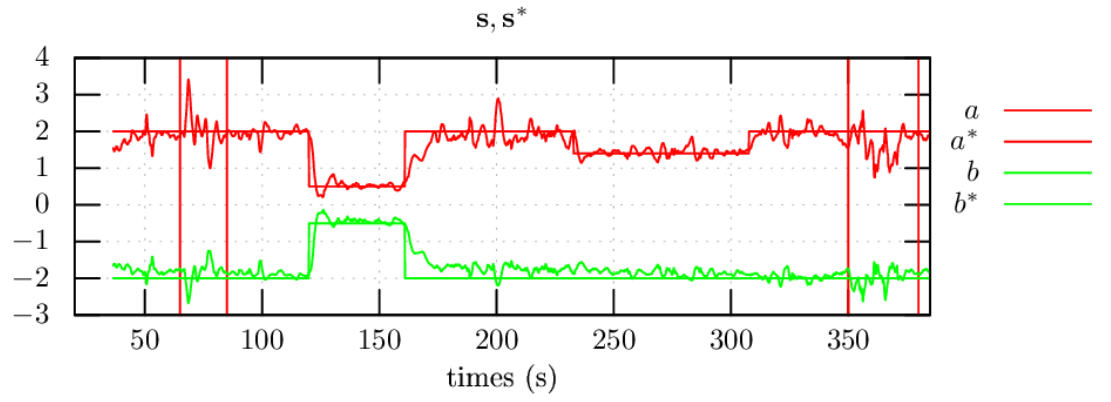


Figure 4.30 – Drones with a cable in yellow transporting a box. Video of the experiment is available at https://youtu.be/849KGUYceoo. $\mathcal{F}_t$ stands for the cable frame, $\mathcal{F}_{b_m}$ for the body frame of the follower drone which is controlled by visual servoing, $\mathbf{p}_m$ is the cable attachment point and the origin of $\mathcal{F}_t$, b_f is the leader drone that is teleoperated.

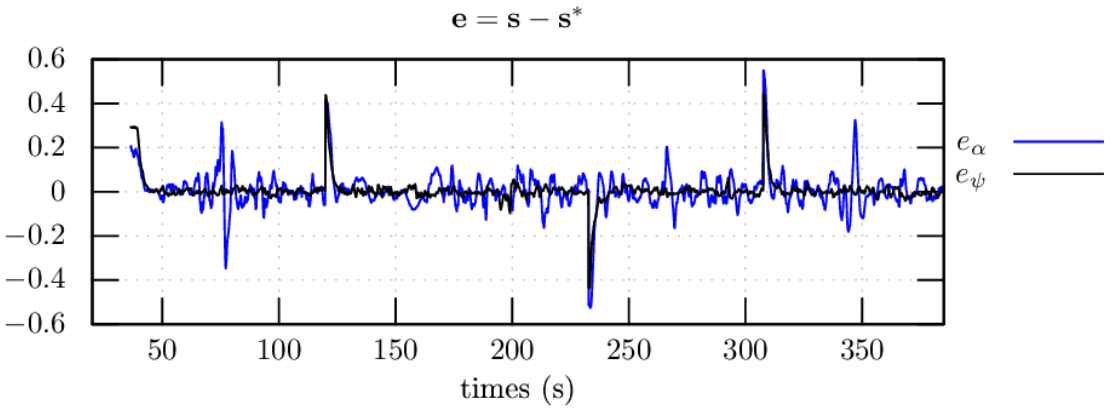
angle error e_α is subject to oscillations around $\pm 15^o$ with rare pics due to the rotation of the box swinging the cable. The cable plane with the box attached is subject to increased but still bounded tilt because the hook can slide and rotate on the cable, meaning that one should carefully choose a desired shape to avoid very fast slipping of the load on the cable. Anyway, the shape controller deals efficiently with the manipulation task as demonstrated by the convergence of the visual features even in the presence of a point load on the cable.

4.6 Conclusion

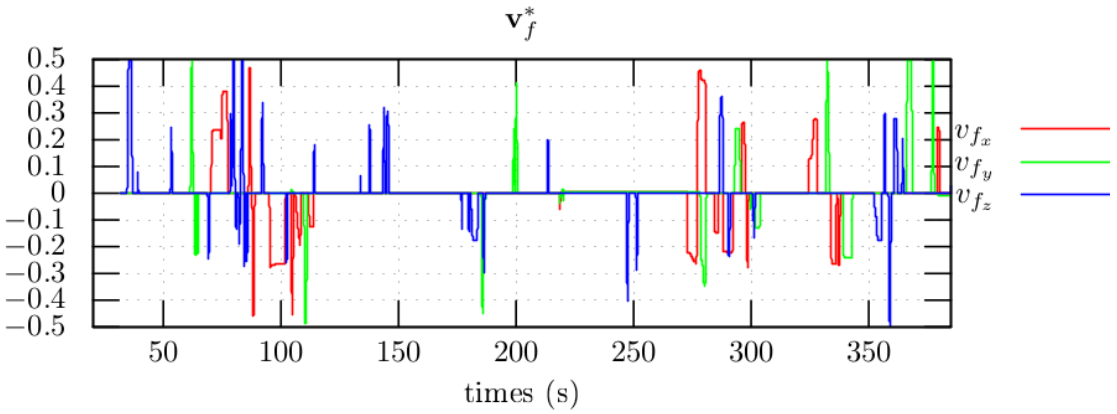
In this chapter, we showed that it is possible to manipulate a cable using a pair of drones holding it. The simulation and experimental results demonstrate the validity of our approach to deform a flexible cable to some desired shapes and even to use its deformation to manipulate rigid objects. The proposed simulation framework allows to simulate physical interaction together with the 3-D perception and control of the drone acting on a cable. The proposed tracking shape visual controller used visual features and integral actions to deform a cable to reach a desired configuration while compensating



(a)



(b)



(c)

Figure 4.31 – Experimental results for visual features and yaw angle of the drone. (a) Desired and estimated parabola coefficients with the grasp and release of an object that occur within the time periods indicated by the red boxes. (b) Error for α and ψ_{b_m} in radians. (c) Feedforward term used during the control of drone b_m that corresponds to the velocity of the leader drone b_f.

for its tension, which is shown by the experimental results to be sufficient to manipulate light objects. To the best of our knowledge, the proposed approach is among the first that provides an onboard shape visual servo control for manipulating and deforming a cable with quadrotor drones. The simulations demonstrated with an MSD cable model and the results of real experiments show that we can efficiently deform soft objects with variable bending stiffness with our method. Furthermore, experiments with different camera setups under different light conditions reveal the robustness of the method to uncertainties from depth estimation and drone odometry. Finally, we experimentally validated that our method remains efficient when grasping and transporting an object by a cable handled by two UAVs.

Chapter 5

Conclusion and Perspectives

5.1 Conclusion

This thesis dealt with the automatic manipulation of soft deformable linear objects. In particular, in the context of robotic applications in industry, the representation of the shape of a DLO and the control of this shape using vision-based methods have been studied.

Although DLOs are very attractive due to their properties such as light weight, resistance to forces, non-invasive contact with the environment, and their many possible shapes, the robotic manipulation of DLOs nevertheless poses various challenges:

— How to represent the shape of a DLO?
— How to model the grasp of a DLO by a robot gripper?
— How to establish the change in the DLO shape with respect to the motion of the robot manipulating it?
— How to estimate the physical parameters of a DLO?
— How to track in real-time the shape of the DLO during robotic manipulation?
— How to manipulate a DLO near singular configurations of its model or when it loses compliance?

In this thesis we contributed to answer to some of these questions by proposing in Chapter 3 a shape visual servoing control framework based on parabolic features for automatically manipulating a suspended cable toward a desired slack or taut configuration. The compliance of the soft DLO was exploited to model its deformation induced by its own weight. A simple but accurate and efficient parabolic model was derived and validated in a series of simulations and experiments involving a robotic manipulator. A real-time markerless shape visual tracking method was also proposed and validated to be effective in the context of the shape control task even in the presence of partial occlusions of the cable shape. We also demonstrated the ability of our method for the manipulation of the DLO in the non-compliant shape of a straight line. Thanks to the derivation of an analytical

model of the interaction matrix, we identified the singular configurations and provided a basic stability analysis of the proposed shape control law. Furthermore, we studied and demonstrated by simulations the robustness of the proposed control law against a coarse estimation of the span of the DLO. Finally, most of the results obtained were presented in an article at the 2023 IEEE International Conference on Robotics and Automation.

Another attractive area for the industrial application of soft DLO manipulation is aerial robotics. In particular, we showed that quadrotor drones have been increasingly studied in the academy in combination with other types of robot. However, the use of quadrotor drones for manipulation tasks presents a number of difficulties such that:

— trade-off between the flight time, distance and payload on the small drones due to the power consumption limited by the battery
— precise navigation, flight and perception of the environment in the presence of the limited payload with limited number of sensors subject to uncertainties
— limited grasping capabilities due to the two previous arguments
— less-studied manipulation capabilities for the cooperative solutions
— less-studied shape control of the DLO for grasping due to the under-actuation of drones and the complexity of representing and controlling its shape

To contribute the aforementioned problems we adapted our shape visual servoing for cooperative manipulation of slack cables by a pair of drones in **Chapter 4** and demonstrated its efficiency by simulations and experiments. We designed a new image processing method for tracking the shape of the soft DLO that can work in real-time on board a small drone equipped with an RGB-D camera. In addition, we proposed a decentralised visual servoing approach that compensates for the rigid translations of the DLO while performing the shape control task and keeping the DLO in the field of view of the camera. We demonstrated the efficiency of our method through simulations and real experiments using a pair of quadrotor drones. Finally, we experimentally validated the suitability of our method for using a DLO as a soft gripper to grasp and transport a rigid object. The results obtained demonstrate that cooperative drones can efficiently manipulate a soft DLO to maintain it to a desired shape even in the presence of disturbances such that the drone state measurement noise or the point load attached to the DLO.

5.2 Future work

In this work, we used the RGB image to detect the cable, and then the depth image aligned with the RGB image to reconstruct the cable pointcloud. Currently, the alignment of two images is not done in the sensor therefore requiring substantial computational effort of the on-board computer of the drone. Furthermore, the RGB camera FOV is smaller than that of the depth cameras FOV meaning that the depth image should be cropped during the alignment procedure. To solve this a perspective work could exploit only the depth image to detect the cable directly from it. Moreover, the depth image can be processed three times faster than the colour image, providing higher frame-rates and therefore increasing responsiveness of the visual servoing control laws.

Another perspective work would be to measure the tension forces in the cable attachment points to compare the data with the tension estimated using a camera. It could be useful to include this cable tension as a feedforward term into the low-level controller of the drone to compensate for it during the flight. In particular, before grasping some point load P with a cable in its form described by (2.29) we could identify the tension $H = \frac{-\mathrm{w}}{2a}$ of the cable at rest (undeformed shape). Afterwards, at the moment after grasping the point load, the form of the cable can be (2.34). Indeed, estimating the new quadratic coefficient $a_{new} = a + d$ resulting from the cable deformation due to the load lifting could give us the increase in horizontal tension of $h = \frac{a_{new}H}{a - a_{new}}$ even without knowing the value of P. This is an illustrative example of using the deformation of the DLO that is tracked from vision to identify the unknown mass of a manipulated rigid object which would not be possible when lifting an object with a rigid gripper.

Subsequent future perspectives could be a study of the influence of the above-mentioned feedforward terms on the orientation of the drones. The eventual tilt of the drone can deteriorate the observability of the cable by a camera. Consequently, shape tracking of the DLO will not be possible and this arises the following challenges:

— Using an actuated pan/tilt camera unit to compensate for the orientation of the drones.
— Using a fully-actuated drones to control also its orientation for better observability of the cable.
— Using another drone tracking the cable shape from the best view.

Furthermore, to make the overall system more autonomous, it would be preferable to avoid the need of the human operator who teleoperates the leader drone and add the

following capabilities:

— The system could follow a predefined trajectory.
— Move away from that trajectory when the object of interest is detected.
— Perform the grasping of that object of interest.
— Return on the trajectory and autonomously avoid eventual collisions of the held object with the environment.

Finally, an interesting future work would be to study how well the proposed deformation model describes real cable when reducing its span. Therefore, the main questions that will arise for the perspective work would be:

— What are the limitations of the proposed model with respect to all possible shapes of a slack cable?
— How to extend the proposed approach to incorporate these limitations?
— Can the proposed model be applied to the shape control of a stiffer DLO such as an elastic beam?
— Is it possible to extend our approach to use with stretchable cables or even when the length of the cable can vary greatly with the use of passive or actuated winches?
— What would be a more efficient model to take into account more complex shapes and allow to consider contacts with the environment?
— What would be a new grasping strategy that involves a more complex model of a DLO and how to control its shape in that case?
— How to choose the most efficient material for a DLO as a compromise between stiff and soft behaviour for grasping?

Appendix A

Résumé en Français

Avec le progrès de la science et de la technologie, l'être humain a progressé dans la création de machines capables d'améliorer ou de remplacer de manière significative son travail et son expertise dans diverses applications telles que la production de biens, l'exploration spatiale, le diagnostic médical et l'agriculture, pour n'en citer que quelques-unes. Toutes ces applications peuvent être réalisées par une séquence de procédures intermédiaires, également appelées tâches. L'exécution de tâches complexes dans des environnements non contrôlés nécessite que les machines se comportent de la même manière que les humains. C'est pour cette raison que le domaine de la robotique a émergé pour fournir aux machines une autonomie leur permettant de s'adapter à des environnements en évolution dynamique.

A.1 Contexte

A.1.1 Manipulation d'objets deformables

La manipulation robotique d'objets deformables reste un problème ouvert car les stratégies de contrôle de forme existantes ne sont valables que pour un type donné d'objet souple, ce qui les restreint à une application définie à l'avance. Par conséquent, fournir une solution de contrôle de forme pouvant se généraliser à tous types d'objets reste un défi majeur dans la communauté robotique. Dans cette thèse, notre premier objectif est de proposer une méthodologie efficace pour mettre en forme de manière autonome un câble dont les extrémités sont actionnées par des robots. Dans le passé, cette tâche a été étudiée pour des robots manipulateurs de type série, et notre objectif est de contribuer à cette étude en proposant une nouvelle méthode qui surmonte certaines limites des travaux antérieurs sur ce sujet. Notre deuxième objectif est de fournir une solution pour manipuler ce câble par des robots aériens en contrôlant sa forme sous l'action de drones qui déplacent ses extrémités. L'application envisagée est de saisir un objet avec le câble et de

le transporter vers un nouvel emplacement. Dans la section suivante, nous présenterons au lecteur la problématique de la manipulation aérienne en robotique.

A.1.2 Manipulation aérienne

Un domaine de recherche particulier est la robotique aérienne. Le robot volant peut exécuter des mouvements dans un environnement 3D dépassant la gravité. Il existe différents types de robots aériens ou UAV (Unmanned Aerial Vehicles) et notamment les drones VTOL (Vertical Take-Off and Landing) comme les drones multirotor qui ont retenu notre attention. Un intérêt particulier est de les utiliser pour des tâches de manipulation aérienne et de transport, car la destination cible peut être atteinte plus rapidement et il y a généralement moins de contraintes environnementales dans l'air à haute altitude. Cependant, la capacité de préhension d'un drone reste limitée à un clip métallique qui n'est pas adaptable aux différentes formes d'objets et peut endommager le drone lors d'une mauvaise prise en raison d'un contact rigide avec le drone. Pour pallier ce problème, plusieurs chercheurs ont proposé de concevoir un drone combiné à d'autres types de préhenseurs robotiques, un bras manipulateur en série ou un robot delta parallèle.

Cependant, les robots aériens présentent une capacité de charge utile très limitée en raison de leur consommation électrique limitée par une source d'énergie embarquée. C'est pourquoi les systèmes multi-drones ont rapidement émergé. Ces drones coopératifs sont attachés à un objet commun via des corps supplémentaires flexibles ou rigides. Cet objet peut être une plateforme magnétique pour transporter des pièces de construction magnétisées comme proposé dans [Jiménez-Cano et al., 2022]. Dans cette thèse nous nous intéressons à l'utilisation d'un corps supplémentaire souple, et plus précisément à un câble permettant de saisir un objet afin de le transporter. Les corps souples ont un avantage significatif sur les corps rigides en termes de poids et de résistance aux perturbations externes telles que le contact avec l'environnement. Dans ce travail de thèse, nous contribuons au domaine de la manipulation aérienne en proposant d'utiliser un câble comme objet commun attaché à une paire de drones et montrons qu'un tel système peut effectuer efficacement une tâche de préhension et de transport aérien. De plus, nous montrons que la forme d'un câble attaché à des drones peut être efficacement contrôlée par leur mouvement à l'aide d'un capteur visuel embarqué.

A.2 Objectifs de la thèse

L'objectif principal de cette thèse est d'élaborer un nouvelle méthode de de commande par asservissement visuel pour contrôler la forme d'un câble d'attache suspendu. Pour atteindre cet objectif, les étapes suivantes sont proposées :

— Dérivation d'un modèle pour représenter un câble suspendu attaché entre deux robots.
— Détermination d'un ensemble pertinent de caractéristiques visuelles et de sa matrice d'interaction associée pour effectuer une tâche de contrôle automatique de la forme du câble qui consiste à manipuler les extrémités du câble de manière à ce qu'il atteigne une forme souhaitée.
— Elaboration d'une méthode de suivi en temps réel des caractéristiques du câble à partir de la vision RGB-D et d'autres capteurs disponibles.
— Conception d'une loi de commande visuelle appliquée aux robots pour effectuer la tâche de mise en forme du câble.
— Validation de la méthodologie proposée à partir de nombreuses simulations et expérimentations impliquant soit un robot manipulateur, soit des drones multirotor.
— Discussion et analyse des résultats obtenus.

A.3 Contributions de la thèse

Cette thèse contribue à l'état de l'art dans la manipulation d'objets linéaires déformables(DLO) à l'aide de manipulateurs série ou de drones quadrirotor. Elle propose notamment :

— Une méthode de contrôle de forme de câble adaptable pour un grand nombre de robots existants présentée dans le **Chapitre 3** et qui a été publiée dans [Smolentsev et al., 2023].
— Une méthode de contrôle de la forme d'un câble suspendu entre deux drones utilisant une détection visuelle embarquée du câble qui est présentée au **Chapitre 4**.
— Une méthode de manipulation d'un câble à l'aide de deux drones quadrirotor pour une saisie aérienne et un transport efficace d'objets présentée dans le **Chapitre 4**.
— De nombreux résultats de simulation et expérimentaux qui valident les tâches de contrôle de forme de câble et de manipulation d'objets présentées dans le

Chapitre 4.

A.4 Conclusion et perspectives

Cette thèse a porté sur la manipulation automatique d'objets linéaires déformables souples. En particulier, dans le cadre d'applications robotiques en industrie, la représentation de la forme d'un DLO et le contrôle de cette forme à l'aide de méthodes basées sur la vision ont été étudiés.

Dans cette thèse, nous avons proposé au chapitre 3 une méthode de commande par asservissement visuel de forme qui se base sur des caractéristiques visuelles paraboliques pour manipuler automatiquement un câble suspendu vers une configuration tendue ou non tendue souhaitée. La souplesse du DLO a été exploitée pour modéliser sa déformation induite par son propre poids. Un modèle parabolique simple mais précis et efficace a été dérivé et validé dans une série de simulations et d'expériences impliquant un manipulateur robotique. Une méthode d'asservisement visuel de forme en temps réel et sans marqueur a également été proposée et validée pour être efficace dans le contexte de la tâche de contrôle de forme même en présence d'occlusions partielles de la forme du câble. Nous avons également démontré la capacité de notre méthode à manipuler le DLO pour qu'il atteigne une ligne droite. Grâce à la dérivation d'un modèle analytique de la matrice d'interaction, nous avons identifié les configurations singulières et fourni une analyse de stabilité de la loi de commande de forme proposée. De plus, nous avons étudié et démontré par simulations la robustesse de la loi de commande proposée par rapport à une estimation grossière de l'étendue du DLO.

Nous avons adapté notre asservissement visuel de forme pour la manipulation coopérative de câbles par une paire de drones dans le **Chapitre 4** et avons démontré son efficacité par des simulations et des expériences. Nous avons conçu une nouvelle méthode de traitement d'images pour suivre la forme du DLO qui peut fonctionner en temps réel à bord d'un petit drone équipé d'une caméra RGB-D. De plus, nous avons proposé une approche d'asservissement visuel décentralisée qui compense les translations rigides du DLO tout en effectuant la tâche de contrôle de forme et en gardant le DLO dans le champ de vision de la caméra. Nous avons démontré l'efficacité de notre méthode à travers des simulations et des expériences réelles utilisant une paire de drones quadrirotor. Enfin, nous avons validé expérimentalement l'adéquation de notre méthode pour utiliser un DLO comme pince souple pour saisir et transporter un objet rigide. Les résultats obtenus démontrent que

les drones coopératifs peuvent manipuler efficacement un DLO pour le maintenir dans une forme souhaitée même en présence de perturbations telles que le bruit de mesure de l'état du drone ou une charge ponctuelle attachée au DLO.

Un travail futur serait de mesurer les forces de tension dans les points d'attache des câbles pour comparer les données avec la tension estimée à l'aide d'une caméra. Il pourrait être utile d'inclure cette tension de câble comme terme de rétroaction dans le contrôleur de bas niveau du drone pour la compenser pendant le vol.

Les perspectives ultérieures pourraient être une étude de l'influence des termes de rétroaction mentionnés ci-dessus sur l'attitude des drones. L'éventuelle inclinaison du drone peut détériorer l'observabilité du câble par une caméra.

De plus, pour rendre l'ensemble du système plus autonome, il serait préférable d'éviter le recours à l'opérateur humain qui téléopère le drone leader.

Enfin, un futur travail intéressant consisterait à étudier dans quelle mesure le modèle de déformation proposé décrit le câble réel pour des DLO de faible étendue.

A.5 Plan

Le document de thèse est organisé comme suit :

Le **Chapitre 2** présente les bases de l'asservissement visuel, passe en revue les techniques de modélisation d'objets linéaires déformables et de contrôle de forme ainsi que les techniques de manipulation aérienne.

Le **Chapitre 3** présente l'apport méthodologique de cette thèse qui concerne l'asservissement visuel de forme d'un câble d'attache à partir d'éléments paraboliques qui est ensuite validé à partir de résultats de simulation et expérimentaux obtenus avec un bras robotique.

Le **Chapitre 4** présente nos travaux sur l'asservissement visuel de la forme d'un câble d'attache à l'aide de drones qui sont validés par des simulations et des résultats expérimentaux.

Le **Chapitre 5** conclut la thèse et suggère des perspectives de recherche futures.

Appendix B

Cost function and its derivatives for D estimation

An important parameter of our proposed cable model is its span, denoted D. In Chapter 3 we show that it can be estimated through the equation (3.13) involving the known length of the cable. A simple solution would be to use some common function zero search method as a bisection method. These methods are slower and harder to initialise than the gradient-based methods such as Levenberg-Marquardt or Gauss-Newton. To be able to use these powerful tools, one must provide derivatives of the function. Below expressions are provided for the function f and its derivatives $\mathbf{J}_{\mathbf{s},D}, \mathbf{H}_{\mathbf{s},D}$ of the first and the second order respectively given by:

$$
\begin{aligned}
f &= (4aL - (2aD+b)\sqrt{(2aD+b)^2+1} + \operatorname{arcsinh}(2aD+b) - b\sqrt{b^2+1} - \operatorname{arcsinh} b)^2 \\
\mathbf{J}_{\mathbf{s},D} &= 2\left(4aL - (2aD+b)\sqrt{(2aD+b)^2+1} + \operatorname{arcsinh}(2aD+b) - b\sqrt{b^2+1} - \operatorname{arcsinh}(b)\right) \\
&* \left(-2a\sqrt{(2aD+b)^2+1} - \frac{2(2aD+b)^2 a}{\sqrt{(2aD+b)^2+1}} + \frac{2a}{\sqrt{(2aD+b)^2+1}}\right) \\
\mathbf{H}_{\mathbf{s},D} &= 2\left(-2a\sqrt{(2aD+b)^2+1} - \frac{2(2aD+b)^2 a}{\sqrt{(2aD+b)^2+1}} + \frac{2a}{\sqrt{(2aD+b)^2+1}}\right)^2 \\
&+ 2\left(4aL - (2aD+b)\sqrt{(2aD+b)^2+1} + \operatorname{arcsinh}(2aD+b) - b\sqrt{b^2+1} - \operatorname{arcsinh}(b)\right) \\
&* \left(-\frac{12a^2(2aD+b)}{\sqrt{(2aD+b)^2+1}} + \frac{4(2aD+b)^3 a^2}{\left((2aD+b)^2+1\right)^{\frac{3}{2}}} - \frac{4a^2(2aD+b)}{\left((2aD+b)^2+1\right)^{\frac{3}{2}}}\right)
\end{aligned}
$$

Appendix C

ROS GRAPH

Here, the ROS computation graph of the simulator is illustrated to provide the reader with some details about the data flow in-between nodes and topics of the simulator presented in Chapter 4.

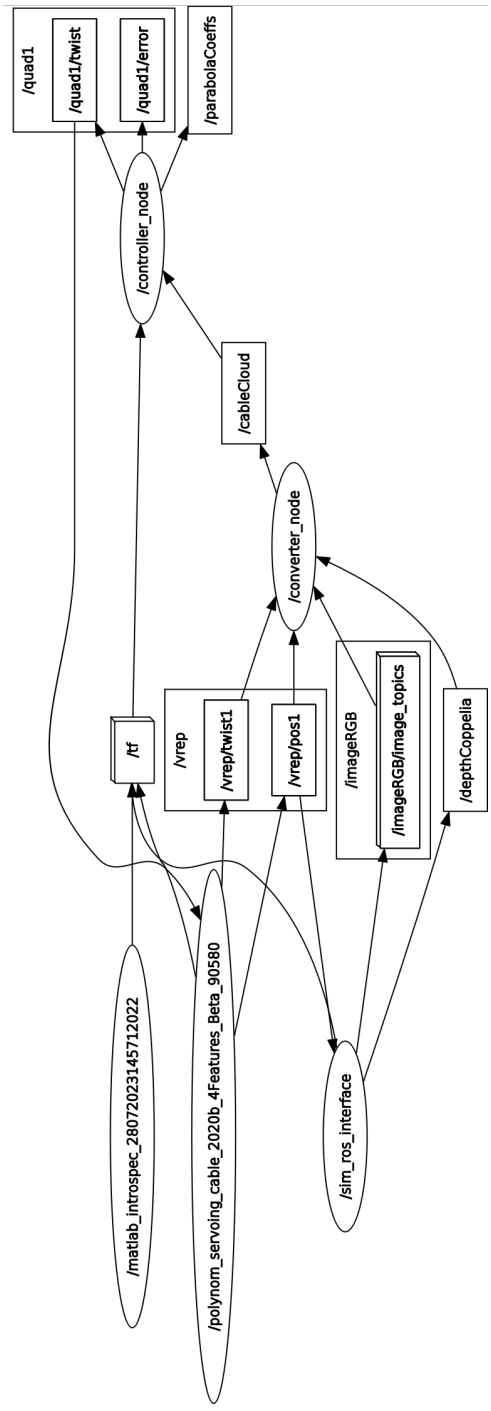


Figure C.1 – Ros graph of the simulation.

BIBLIOGRAPHY

Aghajanzadeh, O., Aranda, M., López-Nicolás, G., Lenain, R., & Mezouar, Y., (2022), An offline geometric model for controlling the shape of elastic linear objects, *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2175–2181.

Bensalah, F., & Chaumette, F., (1995), Compensation of abrupt motion changes in target tracking by visual servoing, *IEEE/RSJ International Conference on Intelligent Robots and Systems. Human Robot Interaction and Cooperative Robots*, *1*, 181–187 vol.1.

Blanco-Mulero, D., Alcan, G., Abu-Dakka, F. J., & Kyrki, V., (2023), Qdp: learning to sequentially optimise quasi-static and dynamic manipulation primitives for robotic cloth manipulation, *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*.

Bodie, K., Tognon, M., & Siegwart, R., (2021), Dynamic end effector tracking with an omnidirectional parallel aerial manipulator, *IEEE Robotics and Automation Letters*, *6*(4), 8165–8172.

Boeuf, A., Cortés, J., Alami, R., & Siméon, T., (2015), Enhancing sampling-based kinodynamic motion planning for quadrotors, *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2447–2452.

Boyer, F., Gotelli, A., Tempel, P., Lebastard, V., Renda, F., & Briot, S., (2023), Implicit time integration simulation of robots with rigid bodies and Cosserat rods based on a Newton-Euler recursive algorithm, *IEEE Transactions on Robotics*, InPress.

Bradski, G., (2000), The opencv library, *Dr. Dobb's Journal of Software Tools*, *25*, 120–125.

Cao, M., Cao, K., Yuan, S., Nguyen, T.-M., & Xie, L., (2023), Neptune: nonentangling trajectory planning for multiple tethered unmanned vehicles, *IEEE Transactions on Robotics*, *39*(4), 2786–2804.

Caro, S., Chevallereau, C., & Remus, A., (2021), Manipulating deformable objects with a dual-arm robot, *2nd International Conference on Robotics, Computer Vision and Intelligent Systems (ROBOVIS)*, 48–56.

Chang, P., & Padır, T., (2020), Model-based manipulation of linear flexible objects: task automation in simulation and real world, *Machines*, *8*(3), 46 (article number: 46).

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., (2007), Visual servoing, *In* E. Dombre & W. Khalil (Eds.), *Robot manipulators: modeling, performance analysis and control* (pp. 279–336), ISTE.

Chaumette, F., & Malis, E., (2000), 2 1/2 d visual servoing: a possible solution to improve image-based and position-based visual servoings, *IEEE International Conference on Robotics and Automation*, 630–635 vol.1.

Chen, R., & Gotsman, C., (2016), Generalized as-similar-as-possible warping with applications in digital photography, *Computer Graphics Forum*, *35*(2), 81–92.

Comport, A., Marchand, E., & Chaumette, F., (2004), Robust model-based tracking for robot vision, *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 692–697.

Corke, P., & Good, M., (1993), Controller design for high-performance visual servoing [12th Triennal Wold Congress of the International Federation of Automatic control. Volume 3 Applications I, Sydney, Australia, 18-23 July], *IFAC Proceedings Volumes*, *26*(2, Part 3), 629–632.

Danko, T. W., Chaney, K. P., & Oh, P. Y., (2015), A parallel manipulator for mobile manipulating uavs, *IEEE International Conference on Technologies for Practical Robot Applications (TePRA)*, 1–6.

D'Antonio, D. S., Cardona, G. A., & Saldana, D., (2021), The catenary robot: design and control of a cable propelled by two quadrotors, *IEEE Robotics and Automation Letters*, *6*(2), 3857–3863.

D'Antonio, D. S., & Saldana, D., (2022), Folding knots using a team of aerial robots, *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 3372–3377.

De Luca, A., Oriolo, G., & Giordano, P. R., (2007), On-line estimation of feature depth for image-based visual servoing schemes, *IEEE International Conference on Robotics and Automation*, 2823–2828.

Della Santina, C., Katzschmann, R. K., Biechi, A., & Rus, D., (2018), Dynamic control of soft robots interacting with the environment, *IEEE International Conference on Soft Robotics (RoboSoft)*, 46–53.

Erskine, J., Chriette, A., & Caro, S., (2018), Wrench capability analysis of aerial cable towed systems, *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, *Volume 5A: 42nd Mechanisms and Robotics Conference*, V05AT07A032.

Filella, N. R., Koessler, A., Bouzgarrou, B., & Corrales Ramon, J.-A., (2022), 3d visual-based tension control in strip-like deformable objects using a catenary model, *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 3210–3217.

Fischler, M. A., & Bolles, R. C., (1981), Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography, *Commun. ACM*, *24*(6), 381–395.

Foughali, M., Ingrand, F., & Mallet, A., (2017), *GenoM3 Templates: from Middleware Independence to Formal Models Synthesis* [working paper or preprint].

Fresnillo, P. M., Vasudevan, S., & Mohammed, W. M., (2022), An approach for the bimanual manipulation of a deformable linear object using a dual-arm industrial robot: cable routing use case, *IEEE 5th International Conference on Industrial Cyber-Physical Systems (ICPS)*, 1–8.

Gabellieri, C., & Franchi, A., (2023), Differential flatness and manipulation of elasto-flexible cables carried by aerial robots in a possibly viscous environment, *2023 International Conference on Unmanned Aircraft Systems (ICUAS)*, 963–968.

Gabellieri, C., Tognon, M., Sanalitro, D., & Franchi, A., (2023), Equilibria, stability, and sensitivity for the aerial suspended beam robotic system subject to parameter uncertainty, *IEEE Transactions on Robotics*, *39*(5), 3977–3993.

Galassi, K., & Palli, G., (2021), Robotic wires manipulation for switchgear cabling and wiring harness manufacturing, *IEEE 4th International Conference on Industrial Cyber-Physical Systems (ICPS)*, 531–536.

Guenard, N., Hamel, T., & Mahony, R., (2008), A Practical Visual Servo Control for an Unmanned Aerial Vehicle, *IEEE Transactions on Robotics*, *24*(2), 331–340.

Huber, P., & Ronchetti, E., (2011), *Robust statistics*, Wiley.

Irvine, H., (1992), *Cable structures*, Dover Publications.

Jiménez-Cano, A. E., Sanalitro, D., Tognon, M., Franchi, A., & Cortés, J., (2022), Precise Cable-Suspended Pick-and-Place with an Aerial Multi-robot System, *Journal of Intelligent and Robotic Systems*, *105*(3), 68.

Jin, S., Lian, W., Wang, C., Tomizuka, M., & Schaal, S., (2022), Robotic cable routing with spatial representation, *IEEE Robotics and Automation Letters*, *7*(2), 5687–5694.

Jin, S., Wang, C., & Tomizuka, M., (2019), Robust deformation model approximation for robotic cable manipulation, *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 6586–6593.

Kiribayashi, S., Yakushigawa, K., & Nagatani, K., (2017), Position estimation of tethered micro unmanned aerial vehicle by observing the slack tether, *IEEE Int. Symp. on Safety, Security, and Rescue Robotics (SSRR)*, 159–165.

Koessler, A., Filella, N. R., Bouzgarrou, B., Lequièvre, L., & Ramon, J.-A. C., (2021), An efficient approach to closed-loop shape control of deformable objects using finite element models, *IEEE Int. Conf. on Robotics and Automation (ICRA)*, 1637–1643.

Lagneau, R., Krupa, A., & Marchal, M., (2020), Automatic shape control of deformable wires based on model-free visual servoing, *IEEE Robotics and Automation Letters*, *5*(4), 5252–5259.

Lagneau, R., (2020), *Shape control of deformable objects by adaptive visual servoing* (Doctoral dissertation), INSA Rennes.

Laranjeira, M., Dune, C., & Hugel, V., (2020), Catenary-based visual servoing for tether shape control between underwater vehicles, *Ocean Engineering*, *200*, 107018.

Lee, T., Leok, M., & McClamroch, N. H., (2010), Geometric tracking control of a quadrotor uav on se(3), *49th IEEE Conference on Decision and Control (CDC)*, 5420–5425.

Li, X., Wang, Z., & Liu, Y.-H., (2019), Sequential robotic manipulation for active shape control of deformable linear objects, *IEEE International Conference on Real-time Computing and Robotics (RCAR)*, 840–845.

Liu, F., Su, E., Lu, J., Li, M., & Yip, M. C., (2023), Robotic manipulation of deformable rope-like objects using differentiable compliant position-based dynamics, *IEEE Robotics and Automation Letters*, *8*(7), 3964–3971.

Loquercio, A., Kaufmann, E., Ranftl, R., Müller, M., Koltun, V., & Scaramuzza, D., (2021), Learning high-speed flight in the wild, *Science Robotics*, *6*(59).

Macklin, M., Müller, M., & Chentanez, N., (2016), Xpbd: position-based simulation of compliant constrained dynamics, *9th International Conference on Motion in Games*, 49–54.

Madani, B., & Ndiaye, M., (2022), Hybrid truck-drone delivery systems: a systematic literature review, *IEEE Access*, *10*, 92854–92878.

Malveiro, M., & Cordeiro, J., (2017), Overhead power line stringing with uav, *12th International Conference on Live Maintenance (ICOLIM)*, 1–2.

Marchand, E., Spindler, F., & 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.

MathWorks, (2020), *Matlab version: 9.13.0 (r2020b)*, Natick, Massachusetts, United States, https://www.mathworks.com

Mayer, N. M., Liu, J.-H., Chao, Y.-Y., Lin, B.-H., & Xu, J.-W., (2023), Fruit harvesting drone based on yolov4-tiny algorithm, *International Conference on Advanced Robotics and Intelligent Systems (ARIS)*, 1–5.

Megson, T., (1996), *Structural and stress analysis*, Arnold.

Michel, H., & Rives, P., (1993), *Singularities in the determination of the situation of a robot effector from the perspective view of 3 points* (Doctoral dissertation), INRIA.

Micotra, M. M., Naldi, R., & Garone, E., (2014), Taut cable control of a tethered uav, *19th World Congress The Int. Federation of Automatic Control*, *47*, 3190–3195.

Moshref-Javadi, M., Hemmati, A., & Winkenbach, M., (2020), A truck and drones model for last-mile delivery: a mathematical model and heuristic approach, *Applied Mathematical Modelling*, *80*, 290–318.

Navarro-Alarcon, D., & Liu, Y., (2013), Uncalibrated vision-based deformation control of compliant objects with online estimation of the jacobian matrix, *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 4977–4982.

Navarro-Alarcon, D., Yip, H. M., Wang, Z., Liu, Y.-H., Zhong, F., Zhang, T., & Li, P., (2016), Automatic 3-d manipulation of soft objects by robotic arms with an adaptive deformation model, *IEEE Transactions on Robotics*, *32*(2), 429–441.

Olson, E., (2011), Apriltag: a robust and flexible visual fiducial system, *IEEE International Conference on Robotics and Automation (ICRA)*, 3400–3407.

Pawlak, K., & Serek, D., (2017), High voltage transmission line stringing operation. usage of unmanned aerial vehicles for installation of conductor and grounding wires with optical fibers, *15th International Conference on Electrical Machines, Drives and Power Systems (ELMA)*, 32–37.

Perozo, M. A., Dussine, J., Yiğit, A., Cuvillon, L., Durand, S., & Gangloff, J., (2022), Optimal design and control of an aerial manipulator with elastic suspension us-

ing unidirectional thrusters, *International Conference on Robotics and Automation (ICRA)*, 1976–1982.

Qi, J., Ma, W., & Navarro-Alarcon, D., (2020), Adaptive shape servoing of elastic rods using parameterized regression features and auto-tuning motion controls, *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 353–354.

Quigley, M., Gerkey, B., Conley, K., Faust, J., Foote, T., Leibs, J., Berger, E., Wheeler, R., & Ng, A., (2009), Ros: an open-source robot operating system, *IEEE International Conference on Robotics and Automation (ICRA) Workshop on Open Source Robotics*.

Rohmer, E., Singh, S. P. N., & Freese, M., (2013), Coppeliasim (formerly v-rep): a versatile and scalable robot simulation framework [www.coppeliarobotics.com], *IEEE International Conference on Intelligent Robots and Systems (IROS)*.

Rusu, R. B., & Cousins, S., (2011), 3D is here: Point Cloud Library (PCL), *IEEE International Conference on Robotics and Automation (ICRA)*.

Sanalitro, D., Tognon, M., Cano, A. J., Cortés, J., & Franchi, A., (2022), Indirect force control of a cable-suspended aerial multi-robot manipulator, *IEEE Robotics and Automation Letters*, *7*(3), 6726–6733.

Shetab-Bushehri, M., Aranda, M., Mezouar, Y., & Özgür, E., (2022), As-rigid-as-possible shape servoing, *IEEE Robotics and Automation Letters*, *7*(2), 3898–3905.

Six, D., Briot, S., Chriette, A., & Martinet, P., (2018), The kinematics, dynamics and control of a flying parallel robot with three quadrotors, *IEEE Robotics and Automation Letters*, *3*(1), 559–566.

Smolentsev, L., Krupa, A., & Chaumette, F., (2023), Shape visual servoing of a tether cable from parabolic features, *IEEE International Conference on Robotics and Automation (ICRA)*, 734–740.

Soler, C., Martin, T., & Sorkine-Hornung, O., (2018), Cosserat rods with projective dynamics, *Computer Graphics Forum*, *37*(8), 137–147.

Sorkine, O., & Alexa, M., (2007), As-Rigid-As-Possible Surface Modeling, *Eurographics Symp. on Geometry Processing*, 109–116.

Spica, R., Giordano, P. R., Ryll, M., Bülthoff, H. H., & Franchi, A., (2013), An open-source hardware/software architecture for quadrotor uavs [2nd IFAC Workshop on Research, Education and Development of Unmanned Aerial Systems], *IFAC Proceedings Volumes*, *46*(30), 198–205.

Srour, A., Franchi, A., & Robuffo Giordano, P., (2023), Controller and Trajectory Optimization for a Quadrotor UAV with Parametric Uncertainty, *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 1–7.

Tahri, O., & Chaumette, F., (2005), Point-based and region-based image moments for visual servoing of planar objects, *IEEE Transactions on Robotics*, *21*(6), 1116–1127.

Tempel, P., Boyer, F., Briot, S., Robuffo Giordano, P., & Chriette, A., (2022), Simulating the manipulation of flexible bodies through drones (S. Briot, J. Burgner-Kahrs, M. Carricato, G. Loianno, & S. Lilge, Eds.), *ICRA 2022 Workshops: New Frontiers of Parallel Robotics (2nd edition)*.

Todorov, E., Erez, T., & Tassa, Y., (2012), Mujoco: a physics engine for model-based control, *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 5026–5033.

Tognon, M., Chávez, H. A. T., Gasparin, E., Sablé, Q., Bicego, D., Mallet, A., Lany, M., Santi, G., Revaz, B., Cortés, J., & Franchi, A., (2019), A truly-redundant aerial manipulator system with application to push-and-slide inspection in industrial plants, *IEEE Robotics and Automation Letters*, *4*(2), 1846–1851.

Tognon, M., Gabellieri, C., Pallottino, L., & Franchi, A., (2018), Aerial Co-Manipulation with Cables: The Role of Internal Force for Equilibria, Stability, and Passivity, *IEEE Robotics and Automation Letters*, *3*(3), 2577–2583.

Tsai, L.-W., (1999), *Robot analysis and design: the mechanics of serial and parallel manipulators* (1st), John Wiley & Sons, Inc.

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.

Walker, M. W., & Orin, D. E., (1982), Efficient Dynamic Computer Simulation of Robotic Mechanisms, *Journal of Dynamic Systems, Measurement, and Control*, *104*(3), 205–211.

Xiao, X., Fan, Y., Dufek, J., & Murphy, R., (2018), Indoor uav localization using a tether, *2018 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*, 1–6.

Yu, G., Xie, W., Cabecinhas, D., Cunha, R., & Silvestre, C., (2023), Adaptive control with unknown mass estimation for a quadrotor-slung-load system, *The International Society of Automation (ISA) Transactions*, *133*, 412–423.

Yu, M., Zhong, H., & Li, X., (2022), Shape control of deformable linear objects with offline and online learning of local linear deformation models, *IEEE International Conference on Robotics and Automation (ICRA)*, 1337–1343.

Yuan, H., Courteille, E., & Deblaise, D., (2015), Static and dynamic stiffness analyses of cable-driven parallel robots with non-negligible cable mass and elasticity, *Mechanism and Machine Theory*, *85*, 64–81.

Zhang, T. Y., & Suen, C. Y., (1984), A fast parallel algorithm for thinning digital patterns, *Communications of the Association for Computing Machinery (ACM)*, *27*(3), 236–239.

Zhu, J., Navarro, B., Fraisse, P., Crosnier, A., & Cherubini, A., (2018), Dual-arm robotic manipulation of flexible cables, *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 479–484.

Zikou, L., Papachristos, C., & Tzes, A., (2015), The power-over-tether system for powering small uavs: tethering-line tension control synthesis, *23rd Mediterranean Conf. on Control and Automation (MED)*, 681–687.

Zürn, M., Wnuk, M., Lechler, A., & Verl, A., (2022), Software architecture for deformable linear object manipulation: a shape manipulation case study, *IEEE/ACM 4th Int. Workshop on Robotics Software Engineering (RoSE)*, 9–16.

COLLEGE DOCTORAL BRETAGNE
MATHS, TELECOMS
INFORMATIQUE, SIGNAL
SYSTEMES, ELECTRONIQUE

Université de Rennes

Titre : Asservissement visuel de la forme d'un câble suspendu

Mot clés : asservisement visuel, manipulation de câbles, objets linéaires déformables, manipulation aérienne

Résumé : Cette thèse se situe dans le domaine de l'interaction robotique avec des objets déformables. Elle présente une approche de commande robotique pour la manipulation autonome d'un câble déformable attaché entre 2 robots et soumis à la gravité. Le travail de recherche a porté sur l'élaboration d'une approche d'asservissement visuel qui utilise une caméra RGB-D pour extraire la forme du câble et l'angle de lacet du plan vertical qui le contient. Pour concevoir la commande du système, nous avons proposé d'utiliser, en tant qu'informations visuelles, les coefficients d'une courbe parabolique représentant une approximation de la forme du câble et l'angle de lacet de son plan. Le modèle d'interaction qui relie les variations de ces informations visuelles aux vitesses des extrémités du câble a été dérivé analytiquement. Des résultats expérimentaux ont dans un premier temps été obtenus avec un bras robotique manipulant une extrémité du câble et ont démontré l'efficacité de cette approche d'asservissement visuel pour déformer le câble vers une configuration de forme désirée. Cette approche a ensuite été adaptée à la manipulation robotique aérienne et validée expérimentalement sur un scénario robotique impliquant la saisie et le transport d'un objet par un câble manipulé par deux drones quadrotors dont l'un, qui est équipé d'une caméra RGB-D, est contrôlé par la méthode d'asservissement visuel proposée.

Title: Shape visual servoing of a suspended cable

Keywords: visual servoing, cable manipulation, deformable linear objects, aerial manipulation

Abstract: This PhD thesis deals with robotic interaction with deformable objects. It presents a robotic control approach for the autonomous manipulation of a deformable cable attached between 2 robots and subjected to gravity. The research work focused on developing a visual servoing approach that uses an RGB-D camera to extract the shape of the cable and the yaw angle of the vertical plane containing it. To design the system control, we proposed to use, as visual features, the coefficients of a parabolic curve representing an approximation of the cable shape and the yaw angle of its plane. The interaction model that relates the variations of these visual features to the velocities of the cable extremities was analytically derived. Experimental results were first obtained with a robotic arm manipulating one end of the cable, demonstrating the effectiveness of this visual servoing approach in deforming the cable to a desired shape configuration. This approach was then adapted to aerial robotic manipulation and experimentally validated on a robotic scenario that involves the grasping and transport of an object by a tether cable manipulated by two quadrotor UAVs with one being equipped with an RGB-D camera and controlled by the proposed visual servoing method.