Flight control and position estimation for a tethered micro aerial robot

Ricardo Miguel Parreiral Pinheiro de Matos Martins

Thesis to obtain the Master of Science Degree in

Electrical and Computer Engineering

Supervisors: Ph.D. Meysam Basiri
Professor Pedro Manuel Urbano de Almeida Lima

Examination Committee

Chairperson: Professor João Fernando Cardoso Silva Sequeira
Supervisor: Ph.D. Meysam Basiri
Member of the Committee: Professor Bruno Guerreiro

December 2019

Declaration

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

Contents

Nomenclature . . . vi

1 Introduction **1**

1.1 Motivation . . . 1

1.2 Problem Definition . . . 1

1.3 State of Art . . . 2

1.4 Objectives . . . 4

1.5 Dissertation Outline . . . 5

2 Quadcopter Modelling and Control Structure **7**

2.1 Basic Flight Concepts . . . 7

2.2 Quadcopter Modelling . . . 8

2.3 Control structure . . . 10

2.4 Matlab simulation . . . 11

2.4.1 Simulink scheme and results . . . 11

2.4.2 PID tuning . . . 12

3 Characterization of the tether's shape **14**

3.1 Introduction to catenary curve . . . 14

3.2 Deduction of the catenary parameters . . . 16

3.2.1 Knowing two points and matching x_0 to the origin . . . 17

3.2.2 Knowing two points and the total length of the tether . . . 20

3.3 Simulink representation . . . 21

4 Tension Estimation **24**

4.1 Calculate tension . . . 24

4.2 Filtering - the Kalman filter . . . 25

4.2.1 General information . . . 25

4.2.2 Filtering process using a constant model . . . 26

4.2.3 Filtering process using a non-constant model . . . 27

4.3 Simulation results - Tension estimation . . . 28

4.3.1 Kalman filter using a constant model . . . 28

4.3.2 Comparison between the constant and the non-constant model in Kalman filter . . 30
4.4 Experimental results 32
4.4.1 Estimation of vertical tension 32
4.4.2 Estimation of horizontal tension 34

5 Tension-based flight control **37**
5.1 Tension following 37
5.1.1 Simulation results 37
5.1.2 Experimental results 40
5.2 Tension-based position control 40
5.2.1 Simulink overview 42

6 Position Estimation **43**
6.1 Position estimation based on knowing the origin point of the tether and matching its lowest point to the origin 44
6.2 Position estimation based on knowing the origin point of the tether and its length 45
6.3 Matlab simulation 46
6.3.1 Simulation results based on knowing the origin point of the tether and matching its lowest point to the origin 47
6.3.2 Simulation results based on knowing the origin point of the tether and its length . . 48
6.4 Experimental results - knowing the tether full length 49

7 Conclusion **54**
7.1 Future Work 55

Bibliography **56**

A Quadcopter's firmware and hardware **59**
A.1 Firmware 59
A.2 Implemented Hardware 60
A.2.1 Location 60
A.2.2 Power System 61

B Tension Variance deduction **63**

C Pseudo-code to compute the parameters of the catenary **65**

Acknowledgments

In the first place, I would like to thank Dr. Meysam Basiri and Professor Pedro Lima for proposing me this challenge regarding a topic that has a wide range of application and it is increasingly important in several engineering daily tasks. Furthermore I would like to thank Dr. Meysam Basiri for the help and availability over this thesis.

Secondly, I would like to thank my family for the support and patience during these last six months. Thank you to my parents, Paulo and Isabel, and sister, Catarina, for the help and guidance, providing me everything that I needed and sometimes even more. I would also like to thank my grandparents - Victória, António, Cecília and Alcino - that were the most comprehensive people in the hardest moments. I would like to highlight the importance of my grandfather Alcino on my academic record and choices, making me follow the path that I'm currently on.

Last but not least, thank you to my friends António Silva and Luís Castro that together with me went through a lot of challenges in these last 5 years. I would also like to thank to my oldest friend Duarte Belo that was present in the hardest times during my academic path, and to the help of Inês Ventura and my father Paulo in the final manuscript review.

Resumo

Esta tese foca-se no estudo dos voos de quadrirrotores pequenos ligados por um cabo a uma plataforma-terrestre, o que pode ser utilizado para fornecer energia de forma contínua ao quadrirrotor e aumentar significativamente o seu tempo de vôo. Estudam-se e implementam-se algumas características que facilitam a interação entre um quadrirrotor e uma plataforma-terrestre. Este trabalho começa por desenvolver um método capaz de estimar a tensão aplicada ao quadrirrotor através da informação proveniente dos sensores UMI (unidades de medição inercial) e da força aplicada pelos motores. Para filtrar o ruído indesejado das medições implementa-se um filtro de Kalman. Depois do processo de filtragem, a estimativa da tensão é utilizada para desenvolver dois métodos alternativos para controlar o quadrirrotor, baseando-se para isso na tensão aplicada. O primeiro método permite mudar a posição do quadrirrotor quando aplicada uma força. Esta solução extende-se à implementação de um processo de aterragem de um quadrirrotor através da recolha do cabo ligado. O segundo método permite desenvolver uma estrutura de controlo da posição do UAV inovadora, baseando-se na tensão que o cabo aplica no quadrirrotor, e na curvatura que a catenária assume. Por último, implementa-se uma metodologia para estimar a posição do quadrirrotor, considerando a tensão estimada e considerando um cabo com uma forma que encaixe no modelo da catenária. Concluindo, desenvolve-se também uma simulação do quadrirrotor e da sua estrutura de controlo, que é útil para testar as soluções implementadas em cada fase do seu desenvolvimento em cenários reais.

Palavras-chave:

Voos com cabo, Quadrirrotor, Curva catenária, Filtro de Kalman.

Abstract

This thesis focuses on the study of tethered ights of a small quadcopter from a ground-base, which can be used to supply continuous power to the aerial robot and signi cantly extend its ight time. Some features that facilitate the interaction between the quadcopter and the ground-base are studied and implemented. This work starts by developing a method capable of estimating the tension applied to a quadcopter by using the inertial information from the IMU sensors and the quadcopter's thrust. To lter the undesired noise presented in the IMU measurements a Kalman lter is implemented. After the lter-ing process, the tension estimate is used to present two alternative methods to control the quadcopter, based on the tension applied to it. The rst method allows changing the quadcopter's position by applying a force to it. This solution is extended to implement a process of landing a quadcopter by pulling the tether attached to it. Moreover, the second method allows developing a novel position control structure, based on the tension that the tether applies to the quadcopter, and based on the shape of a tether that outlines a catenary curve. At last, a methodology to estimate the position of the quadcopter is implemented, taking into account the tension estimates and the shape of a tether that ts into the catenary model. Concluding, it is also developed a Matlab simulation of the quadcopter and its control structure, useful to test the solutions implemented on each stage in a safe environment before its development in a real-life scenario.

Keywords:

Tethered ight, Quadcopter, Catenary curve, Kalman Filter.

List of Tables

3.1 Number of necessary iterations for the Newton-Raphson method to converge 19

6.1 Average Relative Error (ARE), Average Absolute Error (AAE), and Maximum Absolute Error (MAE) of the different experiments 53

List of Figures

1.1 UAV for teleoperated machine and for providing awareness to a ground-robot [4] [5], respectively . . . 3

2.1 Crazy ie top view and body frame . . . 7

2.2 World frame f W g, body frame f B g, radial vector $\vec{r}$, its horizontal projection $\vec{r}$, and Euler angles . . . 8

2.3 Body frame B, world frame W, motor axes, and torque vectors ($_1$ and $_2$) . . . 9

2.4 Quadratic regression mapping the input command and the propeller's thrust. Image from [33] . . . 10

2.5 Control structure - cascade loop . . . 11

2.6 Simulink scheme of a control structure in cascade . . . 12

2.7 Goal and current positions of the quadcopter . . . 12

2.8 Matlab tool box for PID tuning . . . 13

3.1 Catenary curve and its related parameters; horizontal tension H and vertical tension T_v; f W g represents the world frame, f C g represents the catenary frame . . . 14

3.2 Horizontal tension decomposition . . . 16

3.3 Working principle of the ground-controller to adjust parameter x_0 . . . 16

3.4 Quadcopter attached to the ground-controller, maintaining x_0 at the origin ($x_0 = 0$) . . . 17

3.5 Comparison between the second and fourth order approximations of f (a) with f (a), for $x_2 = 1$ (left) and $x_2 = 0:1$ (right) . . . 19

3.6 Catenary curve shape with the second point at $x_2 = 0:1$ and $y_2 = 2$ . . . 19

3.7 Catenary curve shape with rst point at $x_1 = 0; y_1 = 0:5$, second point at $x_2 = 0:8; y_2 = 1:1$, and length of 1.55m . . . 21

3.8 Simulink diagram considering the tether's impact - Tether block . . . 22

3.9 Goal and current positions of the quadcopter for tethered ight, using the scenario where the tether has a variable length . . . 22

4.1 Error histogram and Gaussian t of tension along x, y and z . . . 27

4.2 Simulink overview to estimate the tension applied to the UAV . . . 28

4.3 tension estimate without noisy measurements neither drag effect . . . 29

4.4 tension estimate without noisy measurements and drag effect . . . 29

4.5 Simulink scheme inside the Tension estimation block 30
4.6 tension estimate with noisy measurements and drag effect 30
4.7 Comparison of the tension estimate by ltering with the constant and non-constant model 30
4.8 Zoom of gure 4.7 for steady values 31
4.9 State observer error (<^{3} vector) 31
4.10 Variance of the error's measurement for the vertical tension 32
4.11 Estimation of the vertical tension for different case scenarios 33
4.12 Test-bench for computing the horizontal tension 34
4.13 Variance of the error's measurement for the horizontal tensions 34
4.14 Horizontal tension estimates for hovering ight at x ' 0:4m, y ' 0m and z ' 1m, using the UAV powered by battery 35
4.15 Horizontal tension estimate for hovering ight at x ' 0:3m, y ' 0m and z ' 1m, with the power supply tether 35
4.16 Horizontal tension estimate for hovering ight at x ' 0m, y ' 0m and z ' 1m, with the power supply tether 36

5.1 Simulink overview having into account the position update structure 38
5.2 Quadcopter's position with external pulling on x 38
5.3 Tension along x direction of the powering tether (catenary) and global tension estimate - without tension following 39
5.4 Quadcopter's position with external pulling at the x (blue) direction 39
5.5 Tension along x direction of the powering tether (catenary) and global tension estimate - with tension following 39
5.6 Screenshots of the quadcopter following the tension's direction 40
5.7 Screenshots of the landing of the quadcopter using the tension following feature 40
5.8 Tension-based control structure 41
5.9 Current and goal positions of the simulation results along x, y and z 41
5.10 Simulink overview having into account the control structure presented in gure 5.8 42

6.1 Quadcopter attached to the catenary curve in <^{3} space; World f Wg and catenary f Cg frames 43
6.2 Altitude of the UAV for a vertical takeoff 46
6.3 Simulink overview with position estimation block 46
6.4 Position estimation of a hovering ight at x = 0:5, y = 0:2 and z = 1, using a variable length (case1) 47
6.5 Position estimation of a vertical takeoff and hovering at x = 0, y = 0 and z = 1, using a variable length (case1) 47
6.6 Position estimation of a hovering ight at x = 0:5, y = 0:2 and z = 1, using a xed length (case2) 48

6.7 Position estimation of a vertical takeoff and hovering at $x = 0$, $y = 0$ and $z = 1$, using a xed length (case2) 48
6.8 Test-bench of the position estimation 49
6.9 Hovering at same height z and similar radial distance (x, y) for small angle (< 20°) - test 1 and test 2 50
6.10 Hovering at different heights z and similar radial distance (x, y) for small angle (< 10°) - test 3 and test 4 51
6.11 Hovering at same height z and different radial distance (x, y) for non-small angle (> 30°)- test 5 and test 6 52

A.1 System architecture overview 59
A.2 Diagram of inner controller 60
A.3 Attitude and Rate controller attached to the quadcopter rmaware 60
A.4 Crazy ie UAV and structure for position tracking 61
A.5 Scheme of the crazy ie powering system with and without the step-down converter 61
A.6 Step-down converter and silicon power supply tether 62

Acronyms and Symbols

P : Proportional controller.

PD : Proportional-Derivative controller.

SMC : Sliding mode control.

PID-A : Proportional-Integral-Derivative controller with an acceleration term.

PID - Proportional-Integral-Derivative controller.

x_0 - Catenary lowest point regarding the abscissas coordinates, in $<^2$.

r_0 - Catenary lowest point regarding the radial coordinate, in $<^3$.

a and C - Catenary curve parameters.

T_H - Horizontal tension along the catenary plane.

T_V - Vertical tension along the catenary plane.

s_{tot} - Tether full length.

s_1 - Tether length from the origin to the lowest point.

s_2 - Tether length from the lowest point to the quadcopter.

$\vec{a}$ - acceleration vector with respect to the world frame

$\vec{a}_B$ - acceleration vector with respect to the body frame

$\vec{g}$ - gravity vector

F_p - Thrust of the quadcopter

F_{ext} - External force applied to the quadcopter

- roll angle

- pitch angle

- yaw angle

- attitude $<^3$ vector containing the roll, pitch and yaw angles.

R( ) - rotation matrix between the world and the body frames

MOCAP - Motion Capture system.

! - Tether weight per length unit.

Chapter 1

Introduction

1.1 Motivation

The small Unmanned Aerial Vehicles(UAV), such as the Crazy ie drone, are increasingly being used for indoor applications. These micro aerial robots are known for their versatility and ability to get information from higher altitudes, whereas ground-robots are not able to do so. However, their small size does not allow them to carry much load, which also means that their batteries must have a small size. Due to that, their ight time can be very short under certain conditions, disabling the completion of a given task. By adding a tether that could supply the quadcopter's power, the ight time of the UAV could signi cantly increase allowing the completion of more complex tasks such as traf c counting, rescue missions, photography or operations supervision, which could not be done before, at least in the most ef cient way. Given that ground-robots have a big payload capacity, higher computational power, and extended battery life-time, the small quadcopter can be attached to a ground-robot, increasing the quadcopter's bene ts. For simpli cation purposes, throughout this thesis, the implemented solutions consider a non-movable ground-base, but they can be transposed to a scenario with a mobile ground-base. Nevertheless, the quadcopter can also provide to the ground-robot the awareness and versatility that they usually do not have. This way, the UAVs would have longer time ights and the ground-robots would have access to information that could not capture before.

1.2 Problem De nition

How to power the quadcopter through the tether? - Despite the bene ts mentioned above, several challenges must be considered when conducting tethered ights, such as the voltage drop across the wire and the consequent extra weight and torque that the wire generates, which undermines the ight's performance and can cause instability problems during the ight. The thinner wires represent a more signi cant resistance, which means that the voltage drop across them is also more considerable. Contrarily, using larger wires represents a heavier weight added to the UAV. Moreover, increasing the supplied voltage is not a viable solution since the resistance of the

UAV is dependent on the motor's thrust, i.e., the supplied voltage required to keep the UAV stable would not be xed.

How to model the shape of the tether? - While ying, the tether that is attached to the quadcopter has a speci c shape, depending on its malleability and density. The modeling of the tether is useful to study the impact that it has on the ight's performance, and it can also be used to estimate the position of the quadcopter. In case of a non-rigid tether connecting the quadcopter to a ground-base, the shape of the tether approximately outlines a catenary curve. Its mathematical expression is given by equation 1.1 - where $\langle a \rangle$, $\langle x_0 \rangle$ and $\langle C \rangle$ are tether's parameter, while x and y correspond to the coordinates in a $<^2$ space.

$$y = a \cdot \cosh \frac{x \quad x_0}{a} \quad C \tag{1.1}$$

However, when trying to solve equation 1.1 in order to compute parameter $\langle a \rangle$, it produces a transcendental equation, which implies that equation 1.1 does not have a closed-form solution.

How to control the UAV based on the tension applied to it? - To control the position of the UAV using the tension applied to it, it is necessary to have an accurate estimate of that tension. This can be done through the Newton-Euler formulation (which describes the quadcopter dynamics), and using the measurements of the IMU sensors along with the quadcopter's thrust. However, the IMU measurements have a high level of noise that do not allow to achieve the desired accuracy for the tension's estimates. Such inaccuracy, if not reduced, has a signi cant undesired impact on the ight control of the quadcopter and, moreover, on the position estimate.

How to exploit the tether to improve the position estimate? - The tension on the end-points of a catenary curve is related to its shape, which means that the relation between the model parameters ($\langle a \rangle$ and $\langle x_0 \rangle$) and the tension applied to the quadcopter can be expressed mathematically. Thus, assuming that the tether outlines a catenary curve, and knowing the tension that the tether applies to the UAV, it is possible to estimate the tether parameters and the quadcopter's position, without using any external system (such as the Mocap system).

1.3 State of Art

In recent years many studies related to autonomous ights of micro aerial robots have been carried out. These researches have different purposes and have been performed in different environments.

Tethered Flight

In [1], a quadcopter was used to paint murals, large walls and hard-to-reach surfaces using a small ink-soaked sponge attached to the UAV, producing a stippled print. However, without using a tethered ight strategy to provide the UAV's power, only small ights could be performed, which means that big walls or big canvas would take longer to be painted. Concerning the amount of transport payload and the versatility of some desired tasks, the use of a single quadcopter may not be enough. A multi-robot

scenario is studied in [2], with two quadcopters tethered between them and to a ground xed point. This aims to solve problems of transportation using multiple quadcopters, whilst obtaining a desired formation and maintaining the tethers stretched. Moreover, [3] presents a more embracing multirobot scenario, with an arbitrary number of tethered UAVs, where only the rst one is attached to a ground xed point.

In several situations, the UAVs can be used to provide additional information to a ground-robot, such as described in [4] and [5]. In [4] a tethered UAV is used to assist a teleoperated construction machine to obtain perspective images in scenarios of natural disasters, where the conventional xed cameras are not indicated due to its lack of mobility. The construction machine provides the UAV energy through a tether and has a winch to control the tether tension, providing additional safety in case of malfunction. In [5] it is studied whether the inclusion of a tethered UAV with a camera can improve the awareness of the ground-robot regarding its surroundings. It is also analyzed the impact of the camera's characteristics: higher resolutions always increase the performance of the system while wider camera lenses provide more awareness, and narrow lenses provide more detail on a target.

Figure 1.1: UAV for teleoperated machine and for providing awareness to a ground-robot [4] [5], respectively

A mechanical and electrical design of a platform for tethered control of the UAV, which includes the ground-station, the tether, and the UAV is presented in [6]. Furthermore, it is also analyzed the tether's shape, considering that it outlines a catenary curve. In [7] it is introduced the concept of an Auxiliary Power Unit (APU), which must ensure the basic functionality for a critical system - one that should never go of ine - in the absence of the cable power supply.

Untethered Flights

In [8], embracing scenarios such as ”pesticide-spraying” or ”landscape survey” are taken into account. For that, a ”teleoperation control framework for multiple UAVs over the internet” is implemented, since most of real-life applications occur in uncertain environments where it is not possible to have fully autonomous ights.

Additionally, vision-based systems are often used in quadcopters for position tracking. However, visual algorithms lose visual tracking when performing quick maneuvers or in scenarios where signi cant variations of illuminance occur, which implies a manual recovery process. In [9] an approach to allow a monocular-vision-based quadcopter to recover from unexpected conditions is presented. The recovery

process is executed by rst stabilizing the attitude and the altitude of the quadcopter and then, to automatically re-initialize its vision pipeline during the ight. In [10] it is introduced an alternative way of controlling a quadcopter using non-invasive scalp electroencephalogram (EEG) in humans.

The quadcopters drones are widely used in several recurrent applications where sometimes the current solutions represent a human-safe risk or are just not enough to reach speci c objectives. In [11] an alternative way of performing the inspection of transmission lines, using a quadcopter with a monitoring system and features such as a stabilization mode and an altitude hold is described. In [12] is presented another example of a repetitive task - gas detection on hard-to-reach areas - that can be improved using a quadcopter based solution. Moreover, quadcopters can also be used in rescue missions and on objects tracking. An example of usage regarding a rescue mission is presented in [13], where it is developed an automatic search algorithm to discover missing people after avalanches, which aims to be implemented on a quadcopter. In [14], [15] and [16] different approaches to object tracking are displayed using a vision-based system. Furthermore, it is presented in [17] a multi-object tracking procedure, adopting a Convolution Neural Network (CNN) together with a vision-based system.

All the situations described before need to use a control system. This system can be built taking into account the inclusion of a tether, as in [18], [19] and [20] or, in untethered ight situations, without inclusion, as in [21] and [22]. Although the usual development of the quadcopter's model uses Newton-Euler equations, the estimation of the parameters can be done in different ways. In [23] the parameters are calculated on the base of physical experiments made for each UAV. Still, an alternative way is to use a system identi cation method [22], which tries to predict in real-time the system parameters, based on the input and output of the plant, as in [19].

The control systems usually adopt a cascade structure with an inner-control loop to track the attitude and an outer-control loop to track the position [2], [18] and [22]. Controllers as PID and simple PD can be implemented ([2], [18]) or more complex ones, as PID-A (PID with an accelerator term) [19] and sliding control methods [24]. A more detailed comparison between the SMC, P, PD, and PID controllers performance can be analyzed in [25]. Another scenario of tethered control can be restraining the tether to be taut, avoiding oscillations and reducing the drag effects ([26], [27] and [28]). In this case a winch is used to control the length of the tether, releasing or retracting as needed. The detection of a taut tether can be made using the inertial sensor or using sensors in the controller of the winch.

The resistance and weight of the cable cannot be negligible, and because of that, different cables have different performances. In [29] a method to make a cable selection according to the relation between ”power loss in the cable and payload” is discussed, in order to increase the system performance.

1.4 Objectives

The use of tethered ights has several practical applications but also many challenges. The main objective of this research is to study and improve the performance of indoor tethered ights attached to a ground-base. In order to facilitate the attainment of the main objective, the following speci c objectives

are de ned.

Modelling and simulation of the quadcopter with a tether - To test the methodologies before implementing them in a real-life scenario. Firstly, the quadcopter and the control structure that allows to perform untethered ights are modeled; and secondly, the tether is modeled and connected to the simulated quadcopter's plant, in order to execute tethered ights.

Tension estimation and following - To estimate the tension applied to the UAV and, if necessary, to follow that tension. Moreover, the tension estimate is useful to estimate the position of the quadcopter.

Tension-based ight control - To execute the UAV ight control using the tension estimate and the shape of the tether, i.e., without using the quadcopter's position.

Position estimation - To predict the position of the UAV based on the tension that the tether applies to it and the properties of the tether. The position estimate of the quadcopter is implemented for a tether with a xed length and for a tether with variable length.

1.5 Dissertation Outline

In Chapter 1, the arguments that support the importance of the theme are presented, along with the de nition of the main problems related to this work. Additionally, a global overview of other related works is done, and the main objectives of this thesis are introduced.

Chapter 2 starts by describing the basic ight concepts of the quadcopter drone. It presents the quadcopter modelling process and its control structure that allows to perform stable ights. These are developed in a Matlab simulation, which is also presented, as well as the strategy used to tune the PID controllers of the simulated structure.

In Chapter 3, the shape of the tether is studied, assuming that the tether outlines a catenary curve. Such shape allows to deduce the mathematical expression that describes the tether in the $<^3$ space. These expression are obtained taking into account two different assumptions: the full length of the tether is known and xed; the length of the tether is variable but the lowest point of the tether must be at the origin. The parameters of the tether are related to the tensions at its end-points, which gives an essential relation to develop the method of position estimation of the UAV.

Chapter 4 presents the development of a procedure to estimate the tension applied to the quadcopter by making use of the IMU measurements and the quadcopter's thrust. Due to the noise that the sensors' measurements have, two ltering methods using the Kalman lter are presented, and their performances are studied and compared in simulation.

Chapter 5 uses the tension estimate formulation to develop two strategies to control the position of the UAV: the rst one updates the position of the quadcopter based on the tension applied to it, in which it is necessary to have an external motion system; the second one uses the tension estimates, and the catenary shape, to control the position of the quadcopter without needing to know the position of the UAV, i.e., without using an external motion system.

Chapter 6 uses, once more, the tension estimate formulation - computed in chapter 4 - and the properties of the catenary curve - presented in chapter 3 - to develop a method able to estimate the position of the quadcopter, considering that the origin is at the wire's end-point that is not attached to the UAV.

In Chapter 7, a summary of the research work developed throughout this thesis is presented. This chapter also introduces some limitations and different approaches that could have been taken into account. Moreover, based on the conducted study, a set of recommendations for future work is discussed.

Chapter 2

Quadcopter Modelling and Control Structure

Summary

This chapter starts by presenting the basic ight concepts of the quadcopter, explaining how different combinations of the motor's inputs allow obtaining different responses from the quadcopter. Moreover, it presents both the process of modelling the quadcopter system - through the Newton-Euler formulation - and also the control structure that is responsible for the UAV ight's performance. Using the system modelling and the control structure, a Matlab simulation is developed in Simulink, which simulates the ight of the quadcopter. Lastly, the system dynamics obtained through the Newton-Euler formulation will be used to estimate the tension that the tether applies to the UAV in chapter 4.

The work developed throughout this chapter is based in [2], [30] and [31].

2.1 Basic Flight Concepts

The Crazy ie drone is a small quadcopter with four independent motors. Two of them have a clock-wise rotation while the other two have a counter clock-wise rotation (see gure 2.1 [1]).

Figure 2.1: Crazy ie top view and body frame

[1] Figure from Getting started with the crazy ie

Due to the layout displayed in gure 2.1, different combinations of the motor's inputs produce different quadcopter's movements. Assume that the body frame is the one from gure 2.1, there are no external disturbances and the motor operations are theoretically perfect. In this case, to rotate about the x axis, the thrust of M4 and M3 must be the same, as well as the thrust of M1 and M2 but, at the same time, the thrust of each pair must be different from the other one. In resemblance, to rotate about the y axis, the thrust of M4 and M1, as well as the thrust of M3 and M2 must be the same between them, but the thrust of each pair must be different from the other. Different inputs for each clock-wise and counter clock-wise rotation pairs rotate the quadcopter about the z.

In order to perform a vertical movement, the four propellers must rotate at the same speed and, depending on the thrust of the motors, the UAV can move to higher or lower altitudes. Furthermore, there are three possible scenarios: 1) if the motor's thrust is not enough to overcome the gravity force, then it lowers the altitude; 2) if it can overcome the gravity, then the altitude increases; 3) if the thrust of the motors is equal to the gravity force, the UAV hovers in a speci c position. If the intended movement is no longer the vertical one but a movement about the x axis or the y axis, then the UAV has to do a rotation along the y and the x axis, respectively. The velocity of the movement is then related to the rotation and the thrust of each propeller.

In general cases of more complex trajectories, a different combination of rotations about the axis allows to obtain different directions and movements.

2.2 Quadcopter Modelling

Start by de ning the world frame W, where $\{w_1; w_2; w_3\} \in W$, and the body frame B, where $\{b_1; b_2; b_3\} \in B$. The relation between them is given by the position vector $\vec{r}$ - with its horizontal projection de ned as being the radial distance vector $\vec{\rho}$ - and by the rotation matrix $R(\ )$. Figure 2.2 illustrates both frames ($\{W\}$ and $\{B\}$) as well as the vector position $\vec{r}$, and the Euler angles (, , and).

Figure 2.2: World frame $\{W\}$, body frame $\{B\}$, radial vector $\vec{r}$, its horizontal projection $\vec{\rho}$, and Euler angles

Equation 2.1 presents the rotation matrix between the world frame and the body frame, where = [; ;] is the attitude vector containing the roll, pitch and yaw angles, respectively.

$$R( ) = \begin{bmatrix} \cos( )\cos( ) & \sin( )\cos( ) + \cos( )\sin( )\sin( ) & \sin( )\sin( ) + \cos( )\sin( )\cos( ) \\ \sin( )\cos( ) & \cos( )\cos( ) + \sin( )\sin( )\sin( ) & \cos( )\sin( ) + \sin( )\sin( )\cos( ) \\ \sin( ) & \cos( )\sin( ) & \cos( )\cos( ) \end{bmatrix} \tag{2.1}$$

The quadcopter can be modelled from Newton-Euler equations - using the classical physics equation for motion - or through Euler-Lagrange equations - using the relation between kinetic and potential energies (see [32]). Both methodologies produce the same equations, whereby it does not matter which method is chosen. Consider $\mathbf{a}$ as the acceleration of the quadcopter in the world frame $\{W\}$, F_p as the thrust of the quadcopter, m as the mass of the quadcopter, $\mathbf{g}$ as the gravity vector, $\dot{\boldsymbol{\eta}}$ as the angular velocity vector of the quadcopter regarding the body frame, I as the inertia matrix of the quadcopter, and as the torque made by the propellers. Therefore, the system is obtained from the Newton-Euler equation as follows:

$$m\mathbf{a} = R( )F_p + m\mathbf{g} + F_{ext} \tag{2.2}$$

$$\dot{\mathbf{H}} = \quad \dot{} \quad (\dot{\mathbf{H}}) + \quad _{ext} \tag{2.3}$$

Equation 2.4 displays the relation between the torques of the quadcopter and the thrust produced by each motor, assuming that the axes of the body frame are not aligned with axes of the motors.

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} \frac{p}{2}l(\quad f1 \quad f2 + f3 + f4) \\ \frac{p}{2}l(\quad f1 + f2 + f3 \quad f4) \\ C(\quad f1 + f2 \quad f3 + f4) \end{bmatrix} \tag{2.4}$$

Figure 2.3 displays the relation between the body frame and the axes of the motors.

Figure 2.3: Body frame B, world frame W, motor axes, and torque vectors ($_1$ and $_2$)

The thrust of the quadcopter corresponds to the sum of the thrust produced by each propeller.

$$F_p = f_{p1} + f_{p2} + f_{p3} + f_{p4} \tag{2.5}$$

It is possible to map the torques and thrust of the quadcopter to the thrust of each motor, according to equation 2.6, where l is the length of the quadcopter's arms.

$$\begin{bmatrix} f_1 \\ f_2 \\ f_3 \\ f_4 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ \frac{\sqrt{2}}{2}l & \frac{\sqrt{2}}{2}l & \frac{\sqrt{2}}{2}l & \frac{\sqrt{2}}{2}l \\ \frac{\sqrt{2}}{2}l & \frac{\sqrt{2}}{2}l & \frac{\sqrt{2}}{2}l & \frac{\sqrt{2}}{2}l \\ C & C & C & C \end{bmatrix}^{-1} \begin{bmatrix} F_x \\ F_y \\ F_z \end{bmatrix} \tag{2.6}$$

The input of the motors is a PWM signal, whereby it is important to obtain the mapping between the thrust and the input command. According to [33], this relation is experimentally obtained by providing to the motors speci c input commands and logging the values of the matching thrust. Figure 2.4 illustrates the quadratic regression of the data points, as well as the pair values.

Figure 2.4: Quadratic regression mapping the input command and the propeller's thrust. Image from [33]

Based on gure 2.4, equation 2.7 presents the quadratic regression expression.

$$f_i = 2.13 \cdot 10^{-11} cmd_i^2 + 1.03 \cdot 10^{-6} cmd_i + 5.48 \cdot 10^{-4} \tag{2.7}$$

2.3 Control structure

The quadcopter corresponds to a system that has 4 inputs and 6 degrees of freedom, which makes its control not trivial. To solve this issue, it is used a cascade control strategy, as shown in gure 2.5. This type of approach is only possible because the inner loop controller runs at a higher frequency than the outer loop controller, making the inner loop look static from the outer loop's perspective.

Figure 2.5: Control structure - cascade loop

The outer loop receives the position of the quadcopter (in general by using an external motion system), and produces a desired thrust, and attitude. It consists of 4 PID controllers, one for x, y and z, and another one for the yaw. This way, it follows that:

the error between the current altitude and the goal altitude produces the thrust;

the error between the current and the goal x coordinate produces the desired pitch;

the error between the current and the goal y coordinate computes the desired roll;

the error between the current and the goal yaw computes the desired yaw.

The inner controller is divided into two parts: the attitude controller - composed of 3 PID controllers, which by receiving the desired attitude, and the current values of the Euler angles, compute the desired torques - and the motor controller - which receives the desired torques and thrust (in PWM), and calculates the PWM values that are directly used as inputs of each motor, according to equation 2.6. These values are afterwards fed into the motor drives, which induce a thrust in the quadcopter, enabling its ight.

2.4 Matlab simulation

2.4.1 Simulink scheme and results

This section presents a simpli ed model of the quadcopter, which was built in Simulink, and based on the results produced in sections 2.2 and 2.3. The simulation considers that the quadcopter weighs 30 gram, and there is a damping coef cient of 0.01 N=m:s ^{1}, which has the purpose of representing the drag effect. Similar to what is presented in image 2.5, gure 2.6 illustrates the global overview of the control structure in Simulink.

Figure 2.6: Simulink scheme of a control structure in cascade

Notice that the motors were not represented in simulink. In order to include its functionality in the scheme, its stationary response is incorporated in the motor controller block through equation 2.7. However, this equation, which maps the input command to the thrust, does not model the transition state of the motors.

Figure 2.7 presents the result of running the simulation model, using an initial goal position of $x = 0{:}8$, $y = 0{:}4$ and $z = 1$. Due to the symmetry of the quadcopter, the PID controllers for x and y directions were tuned with symmetrical values, and so it is also expected that the responses to the step inputs are identical.

Figure 2.7: Goal and current positions of the quadcopter

2.4.2 PID tuning

The PID tuning process is not trivial since the system to tune consists of a control structure in cascade. The PID auto-tuning tool from the Matlab toolbox, which is presented in gure 2.8, is not by itself enough to tune the outer loop controller. To overcome this situation, it is rst tuned the inner loop controller while the outer controller is in open loop, using for this the auto-tuning tool from Matlab. Afterwards, the outer loop controller is manually tuned, and so it is fundamental to introduce the PID basic operating principles.

Figure 2.8: Matlab tool box for PID tuning

A PID controller is divided into 3 parts (gains), and each of them is related to the performance of the system in a different way. These are:

the proportional gain - it is responsible to the responsiveness of the system - faster or slower; however increasing this value may lead to a bigger overshoot and a higher frequency of the oscillations. Additionally, the proportional gain does not remove the steady-state error.

the derivative gain - it improves the response of the system by reducing the overshoot and the oscillation frequency. This gain is very useful and powerful, especially when combined with the proportional gain. However, it is very sensitive to noise, particularly high-frequency noise. To prevent that, it is normally used along with a low pass lter.

the integrator gain - its main advantage consists of removing the steady-state error, which cannot be done using only the proportional gain. Without eliminating the steady-state error, the quadcopter would never achieve its goal position due to the gravity force. In a hypothetical scenario where only a PD controller is used, there would be a time instant in which the thrust provided by the propellers would match the weight of the quadcopter, making it impossible to move vertically. This impossibility would cause an undesired steady-state error.

By making use of the previously mentioned concepts, the PID is tuned empirically. The proportional and the derivative gains are rst adjusted, and then, the integral term is tuned. In the end, small adjustments are made, and the PID gains are set to their nal values.

Chapter 3

Characterization of the tether's shape

Summary

This chapter studies the case of a non-rigid tether connecting the quadcopter to a point in space, the shape of the tether approximately outlines a catenary curve . This chapter aims to introduce the main concepts of such curve and to develop different procedures to compute its parameters, which allows the mathematical representation of the tether's shape. The parameters of the curve are calculated through the resolution of a system of equations, assuming that the two end-points of the tether are known. However, this information may not be enough to produce a solution, and so it necessary to gather information from additional sources, in order to have a possible and determined system of equations. Given that, two different approaches are evaluated allowing to collect that information or knowing the tether's lowest point; knowing the full length of the tether.

3.1 Introduction to catenary curve

The catenary model consists of a hanging cable, with no stiffness, sagging under its own weight, and supported only by its ends (see gure 3.1). The point $(x_1; y_1)$ corresponds to the origin and the point $(x_2; y_2)$ to the position of the quadcopter. The following results are based on references [34] and [35].

Figure 3.1: Catenary curve and its related parameters; horizontal tension H and vertical tension T_v; f W g represents the world frame, f C g represents the catenary frame

The catenary shape can be written according to a mathematical model, in which expression 3.1 presents the equation of the catenary .

$$y = a \cdot \cosh \frac{x - x_0}{a} + C \tag{3.1}$$

Parameter $\langle x_0 \rangle$ is the abscissa of the lowest point. Parameter $\langle a \rangle$ corresponds to the y coordinate of the lowest point of the curve ($x = x_0$) regarding the catenary frame $\{C\}$, and it must always be positive.[1]. Parameter $\langle C \rangle$ is an offset between the world frame $\{W\}$ and the catenary frame $\{C\}$, which depends on the tether's parameter $\langle a \rangle$ and the y coordinate of the lowest point - y_0 - regarding the world frame.

$$C = y_0 - a \tag{3.2}$$

In practice, the parameters $\langle a \rangle$ and $\langle C \rangle$ are not possible to observe directly, since the altitude of the lowest point y_0 is the sum of $\langle a \rangle$ and $\langle C \rangle$. In opposition, the parameter $\langle x_0 \rangle$ is directly observable[2].

Expressions 3.3 and 3.4 introduces the tether parameters $\langle s_1 \rangle$ and $\langle s_2 \rangle$, which represent the arc-length from the curve lowest point to the origin and to the UAV, respectively.

$$s_1 = a \cdot \sinh \frac{|x_1 - x_0|}{a} \tag{3.3}$$

$$s_2 = a \cdot \sinh \frac{|x_2 - x_0|}{a} \tag{3.4}$$

Equations 3.5 and 3.6 respectively present the horizontal and vertical tensions for a generic point of the catenary tether. Both the horizontal and vertical tension depend on the tether's parameters - $\langle a \rangle$ or $\langle s \rangle$ - and on the weight of the tether, where ρ is the tether weight per length unit [34].

$$T_H = \rho \cdot a \tag{3.5}$$

$$T_V = \rho \cdot s \tag{3.6}$$

In expression 3.6, the variable $\langle s \rangle$ corresponds to a generic arc-length - an arc-length from the lowest point $\langle x_0 \rangle$ to another point of the tether - and its expression is similar to the one presented in equations 3.3 and 3.4.

$$s = a \cdot \sinh \frac{|x - x_0|}{a} \tag{3.7}$$

The absolute value of the tension results of the euclidean norm between the horizontal and vertical tensions, according to equation 3.8.

$$|T| = \sqrt{T_V^2 + T_H^2} \tag{3.8}$$

The quadcopter flying in a $\mathbb{R}^3$ space allows to define the horizontal tension (T_H) in terms of a component along the x direction and another along the y direction, as shown in figure 3.2.

[1]A negative value of parameter $\langle a \rangle$ would only have a physical meaning if the shape of the catenary was concave instead of convex

[2]By a parameters that is directly observable it is meant that it can be directly measured, i.e., without using any calculations

Figure 3.2: Horizontal tension decomposition

The previous circumstance relates the tension along the x and y direction according to equations 3.9 and 3.10, respectively.

$$T_x = \cos(\ ) \cdot |T_H| \tag{3.9}$$

$$T_y = \sin(\ ) \cdot |T_H| \tag{3.10}$$

3.2 Deduction of the catenary parameters

The parameters of the catenary cannot be mathematically computed by knowing only the two end-points of the cable. Thus, the parameters of the catenary - $\langle a \rangle$, $\langle x_0 \rangle$, $\langle C \rangle$, $\langle s_1 \rangle$, $\langle s_2 \rangle$, and $\langle s_{tot} \rangle$ - can be produced gathering additional information from two different sources: knowing the lowest point of the catenary $\langle x_0 \rangle$ (case 1), or the total length of the tether $\langle s_{tot} \rangle$ (case 2). In both cases, $\langle x_0 \rangle$ and $\langle s_{tot} \rangle$ can be measured directly, i.e., without using calculations. The discussion on which way the parameter $\langle x_0 \rangle$ and $\langle s_{tot} \rangle$ can be computed is described below.

Case 1 - A possible procedure to measure parameter $\langle x_0 \rangle$ is by adding a marker that slides to the lowest point of the tether. Nevertheless, the weight of the marker may change the shape of the curve to a non-catenary shape. An alternative approach is <u>matching the lowest point of the tether $\langle x_0 \rangle$ to the origin</u> through a ground-controller, by knowing that the vertical tension at $\langle x_0 \rangle$ is null. This ground-controller would be responsible for pulling or releasing the tether in order to maintain the vertical tension at the origin null. To ensure the proper operation of the ground-controller, it is necessary that the quadcopter does not y below the height of the ground-controller. Otherwise, it is not possible to achieve a desired $\langle x_0 \rangle$ by only pulling or releasing wire.

Figure 3.3: Working principle of the ground-controller to adjust parameter x_0

Case 2 - The measurement of the full length of the tether can be done during the ight or before the ight. Measuring the length of the tether during the ight needs a ground-controller that has a built-in odometry system able to measure the amount of cable released. The approach of measuring in advance the length of the cable limits the action range of the UAV, but does not present odometry errors, neither needs the implementation of a ground-controller.

The strategies underlined above were the ones that this thesis focuses on. In summary, each one of them has its pros and cons:

Matching the $\langle x_0 \rangle$ point to the origin - Con: A ground-controller is necessary in order to maintain the vertical tension null at the origin, i.e, to maintain the lowest point $\langle x_0 \rangle$ at the origin.
Pro: The quadcopter is not limited to a speci c range, since the length of the tether is not xed.

Measuring in advance the length of the cable - Con: The ight of the quadcopter is limited to the tether's length, since this one is xed.
Pro: The tether's lowest point does not need to match to the origin and hence it is not necessary to implement a ground-controller.

3.2.1 Knowing two points and matching x_0 to the origin

Start by considering that equation 3.11 and 3.12 describes the two points with the rst point $(x_1; y_1)$ matching the origin (see equation 3.13).

$$y_1 = a \cdot \cosh\left(\frac{x_1 - x_0}{a}\right) + C \tag{3.11}$$

$$y_2 = a \cdot \cosh\left(\frac{x_2 - x_0}{a}\right) + C \tag{3.12}$$

$$x_0 = x_1 \tag{3.13}$$

Figure 3.4 illustrates the case

Figure 3.4: Quadcopter attached to the ground-controller, maintaining x_0 at the origin ($x_0 = 0$)

When replacing $\langle x_0 \rangle$ in equation 3.11 by equation 3.13 it follows expression 3.14, and subtracting equation 3.12 by equation 3.11 results the expression in equation 3.15, with $y = y_2 - y_1$.

$$C = y_1 - a \cdot \tag{3.14}$$

$$f(a) = \quad y \qquad a{:}\cosh\frac{x_2}{a} \qquad a \quad = 0 \tag{3.15}$$

Parameter $\langle a \rangle$ is computed by using the iterative method of Newton-Raphson in equation 3.15. The method calculates the zeros of the original function using the function itself, and its derivative (see equation 3.16). This method stops when the absolute value of $f(a_n)$ is below a previously speci ed threshold.

$$a_{n+1} = a_n + \frac{f(a_n)}{f^0(a_n)} \tag{3.16}$$

The Newton-Raphson method does not guarantee the convergence of the parameter to be estimated. To improve the convergence of the method, it is necessary to have an accurate initial estimation of the parameter a_n, which not only helps the method to converge but also reduces the necessary steps to do it, turning it faster. The expansion of the hyperbolic cosine using a Taylor series (see equation 3.17), and using the Cardano's formula on the resultant cubic equation, allows to estimate an initial approximation for parameter $\langle a \rangle$.

$$\cosh(x) = \frac{e^x + e^{\ x}}{2} = \sum_{n=0}^{inf} \frac{x^{2n}}{(2n)!} = 1 + \frac{x^2}{2!} + \frac{x^4}{4!} + \frac{x^6}{6!} + :::: \tag{3.17}$$

The Cardano's formula produces the roots of a cubic polynomial having the general expression of equation 3.18 with a, b and c 2 < .

$$x^3 + \ x^{\ 2} + \ x \ + \quad = 0 \tag{3.18}$$

Any cubic expression having the form of expression 3.18 can be re-written in a reduced form, as shown in equation 3.19.

$$y^3 + p{:}y + q = 0 \tag{3.19}$$

where $y = x + \frac{}{3}$; $p = \frac{2}{2}$ and $q = \ + \frac{2\ 3}{27} \ \frac{:}{3}$. After re-writing expression 3.18 into expression 3.19, the Cardano method allows to compute the roots of expression 3.19 (see equation 3.20). Notice that these roots correspond to the roots of expression 3.19, so it is necessary to apply the change of variable $x = y \ \frac{}{3}$ in order to obtain the roots of expression 3.18 instead.

$$\text{roots} = \sqrt[3]{\frac{q}{2} + \sqrt{\frac{q^2}{4} + \frac{p^3}{27}}} + \sqrt[3]{\frac{q}{2} \ \sqrt{\frac{q^2}{4} + \frac{p^3}{27}}} \tag{3.20}$$

Given that, equation 3.21 results from the fourth-order approximation of equation 3.15 according to the Taylor expansion in equation 3.17.

$$y \ \frac{x_2^2}{2!a} \ \frac{x_2^4}{4!a^3} = 0 \tag{3.21}$$

Using equation 3.21 it results $= \frac{x_2^2}{2!\ y}$; $= 0$ and $= \frac{x_2^4}{4!\ y}$. Using the Cardano's formula on equation 3.18, replacing , and by these new values, the initial estimation for the parameter $\langle a \rangle$ is computed.

Equation 3.22 presents the result for the initial estimation of parameter $\langle a \rangle$ taking into account only the Taylor's second-order approximation.

$$a = \frac{x_2^2}{2!y} \tag{3.22}$$

Figure 3.5 shows the difference between the second-order approximation and the fourth-order approxi-

mation comparing to the original tether expression for different values of x_2 variable.

Figure 3.5: Comparison between the second and fourth order approximations of f (a) with f (a), for $x_2 = 1$ (left) and $x_2 = 0:1$ (right)

When crossing the zero, the function that results from the Taylor's fourth-order approximation is closer to the original function than the one that outcomes from the second-order approximation.

The Newton-Raphson method uses the initial estimation of the parameter h_{ai}. Thus, to evaluate the performance of these two approximations it is also essential to analyze its impact on the convergence time and robustness of that method. Table 3.1 illustrates those results, where Nan means that the Newton-Raphson method cannot converge (the used threshold is 0.001)

y	x_2	4	2
2	0.1	6	39
2	0.01	13	396
2	0.001	30	Nan

Table 3.1: Number of necessary iterations for the Newton-Raphson method to converge

The Matlab script implemented assumes that the rst point is at the origin ($x_1 = y_1 = 0$) and calculates the parameters of the tether given any second point. The script plots the tether and the correspondent points that generated it (see algorithm 1 in appendix C). The parameters of the tether are well calculated if the given points match the plotted curve, such as shown in gure 3.6.

Figure 3.6: Catenary curve shape with the second point at $x_2 = 0:1$ and $y_2 = 2$

3.2.2 Knowing two points and the total length of the tether

Another way of solving the problem is assuming that x_0 is unknown, but the length of the tether is constant and known. This assumption leads to a possible and determined system of equations, similar to the one given by equations 3.11-3.13. Given that, solving the system of equations 3.23-3.27 generates the curve parameters.

$$y_1 = a \cdot \cosh\left(\frac{x_1 - x_0}{a}\right) + C \tag{3.23}$$

$$y_2 = a \cdot \cosh\left(\frac{x_2 - x_0}{a}\right) + C \tag{3.24}$$

$$s_{tot} = s_2 + s_1 \tag{3.25}$$

$$s_1 = a \cdot \sinh\left(\frac{|x_1 - x_0|}{a}\right) \tag{3.26}$$

$$s_2 = a \cdot \sinh\left(\frac{|x_2 - x_0|}{a}\right) \tag{3.27}$$

Subtracting equation 3.23 from equation 3.24, and making use of hyperbolic cosine properties, it follows:

$$Y = 2a \cdot \sinh\left(\frac{x}{a}\right) \cdot \sinh\left(\frac{x_{average} - x_0}{a}\right); \tag{3.28}$$

where $x = \frac{x_2 - x_1}{2}$, $x_{average} = \frac{x_2 + x_1}{2}$ and $y = y_2 - y_1$. The expression of the length of the tether $\langle s_{total} \rangle$ (see equation 3.29) is re-written by replacing equation 3.26 and 3.27 into expression 3.25, and using once more the hyperbolic sine properties.

$$s_{tot} = 2a \cdot \sinh\left(\frac{x}{a}\right) \cdot \cosh\left(\frac{x_{average} - x_0}{a}\right); \tag{3.29}$$

Equation 3.30 presents a useful relation between $\langle x_0 \rangle$ and $\langle a \rangle$, which results from the division of $\langle Y \rangle$ by $\langle s_{tot} \rangle$

$$x_0 = x_{average} - a \cdot \tanh^{-1}\left(\frac{Y}{s_{total}}\right) \tag{3.30}$$

The insertion of equation 3.30 in equation 3.28 allows to obtain equation 3.31. Afterwards, the appliance of the Newton-Raphson method to the equation 3.31 produces the value of parameter $\langle a \rangle$, similar to what is done in sub-section 3.2.1. However when $Y = 0$ the parameter $\langle a \rangle$ is impossible to compute since equation 3.31 does not depend on $\langle a \rangle$ to be valid. Nevertheless, from the knowledge that $Y = 0$ it comes that x_0 is known and corresponds to $x_{average}$, which means that equation 3.29 can compute the parameter $\langle a \rangle$. An alternative approach is to force the value of Y to be not null by adding a small offset.

$$Y - 2 \cdot a \cdot \sinh\left(\frac{x}{a}\right) \cdot \sinh\left(\tanh^{-1}\left(\frac{Y}{s_{total}}\right)\right) = 0 \tag{3.31}$$

The expansion of the Taylor series derives the initial estimation of parameter $\langle a \rangle$.

$$\sinh(x) = \frac{e^x - e^{-x}}{2} = \sum_{n=0}^{inf} \frac{x^{2n+1}}{(2n+1)!} = x + \frac{x^3}{3!} + \frac{x^5}{5!} + \cdots \tag{3.32}$$

By applying this approximation to expression 3.31, one can re-write this last equation as a result of a 5th sinh order approximation for the hyperbolic sine, according to equation 3.33.

$$\frac{Y}{2:\sinh \quad \tanh \quad 1(\frac{Y}{S_{tot}})} \qquad x \quad a^4 \quad \frac{x^3}{3!}a^2 \quad \frac{x^5}{5!} = 0 \tag{3.33}$$

The assumption that $= a^2$ reduces the equation 3.33 to a 2nd order. Furthermore, the quadratic formula, presented in equation 3.34, produces the solution for this 2nd order expression

$$= \frac{b + = \quad \sqrt{b^2 \quad 4:}}{2:} \tag{3.34}$$

where

$$= \frac{Y}{2:\sinh(\tanh \quad ^1(\frac{Y}{S_{total}}))} \qquad x; b = \frac{X^3}{3!}; c = \frac{x}{5!}: \tag{3.35}$$

Inverting the variable substitution produces the desired value for $\langle a \rangle$, according to equation 3.33.

$$a = \sqrt{\bar{(\quad)}}; \tag{3.36}$$

Since equation 3.33 is a 4_{th} order equation it produces 4 roots - in the relevant domain, two of them are complex roots and two of them real roots, one positive and other negative. Only the positive real root has a physical meaning and so it is the only one to be taken into account.

The replacement of $\langle x_0 \rangle$ and $\langle a \rangle$ into equation 3.23 or into equation 3.24 allows to compute the $\langle C \rangle$ parameter. Averaging both equations (3.23 and 3.24) increases the accuracy of the estimation of parameter C. This way, for a given pair of points, and a speci c tether's length, it is possible to outline the catenary curve that matches these points (see algorithm 2 in appendix C), as shown in gure 3.7.

Figure 3.7: Catenary curve shape with rst point at $x_1 = 0; y_1 = 0:5$, second point at $x_2 = 0:8; y_2 = 1:1$, and length of 1.55m

3.3 Simulink representation

In chapter 2, section 2.4 introduces the Simulink model of the control structure of the quadcopter without taking into account the tether's in uence. Once the tether parameters are computed, the tension

that the tether applies to the quadcopter is added to the Simulink model, as illustrated in gure 3.8. The Tether block is a Matlab function that receives the current position of the quadcopter and other necessary parameters, such as the gravity force, the length and the mass of the tether. The block outputs the tension along x, y and z directions that the tether applies, and the quadcopter model (represented by the Plant block) receives them as inputs.

Figure 3.8: Simulink diagram considering the tether's impact - Tether block

Due to the weight and malleability of the tether, its in uence cannot be neglected when tuning the PID controllers. Following the strategy presented in section 2.4.2, the PID controllers are re-tuned considering the tension that the tether applies to the quadcopter. Figure 3.9 presents the trajectory of the quadcopter for the same goal positions as the ones in chapter 2, assuming that the tether has a variable length. A similar result can also be obtained using the ight situation with a xed length of the tether. However, since the tensions applied to the UAV are different between situations, the PID gains are also different.

Figure 3.9: Goal and current positions of the quadcopter for tethered ight, using the scenario where the tether has a variable length

Notice that, due to the catenary properties, when the altitude and the y coordinate change abruptly, the tension applied along x is also going to change. This reality causes a slight oscillation on the x

coordinate of the quadcopter. A solution to that would pass by improving the robustness of the PID controllers to decrease the oscillation, though, it would be a trade-off between a faster response and a more stable response.

In summary, one can conclude that the PID controllers present a stable solution if the position does not have wide changes. However, the simulation ights do not take into account the oscillations of the tether, but the ights in real-life do it. Given that, the PID controllers must also be robust enough to mitigate the oscillation's impact on the ight's performance of the quadcopter.

Chapter 4

Tension Estimation

Summary

The tension applied to the quadcopter can be computed using its attitude, thrust and acceleration. However, the measurements of the sensors are not accurate enough to present the results with the desired accuracy. Two different Kalman filters are introduced to filter the noise from the sensors: one with a model that assumes the tension is constant over time, which is useful for hovering flights; another one with a model that does not assume a constant tension over time, which initially seems to be not ideal for the hovering flights, but has a wider range of applications.

Afterwards, the work developed under this chapter is used in chapter 5 - to do the UAV flight control based on the tension applied to it - and in chapter 6 - to estimate the UAV position, since the catenary parameters that allows to estimate the quadcopter's position are related with the tension at its end-point.

4.1 Calculate tension

As described in chapter 2, the dynamic model of the quadcopter is given according to equations 2.2 and 2.3. The tether adds an extra force to the quadcopter, so equation 2.2 must be updated to equation 4.1. This procedure assumes that the tether is attached to the quadcopter's center of mass, and so the applied tension has no impact on the torques or angular velocities of the quadcopter, whereby equation 2.3 does not change. Knowing the mass m of the quadcopter, its acceleration $\bar{a}$ with respect to the world frame $\{W\}$, its attitude, and its thrust F_p, equation 4.1 computes the tension T that the tether applies to the UAV. The parameters $\bar{g}$ and $R(\ )$ correspond to the gravity vector and the rotation matrix presented in 2.1.

$$T = m(\bar{a} + \bar{g}) \quad R(\ )F_p + F_{ext} \tag{4.1}$$

The direction of the tension has the same referential as the fixed frame, meaning that pulling the wire in the same direction as one of the axes produces a positive tension regarding that axis. Pulling in the opposite direction, by the same logic, produces a negative tension.

The acceleration from the IMU sensors have their values with respect to the quadcopter's frame and not the world frame. Thus, equation 4.1 is updated into equation 4.2, where the acceleration a_B regards the body frame and already includes the gravitational acceleration.

$$T = mR(\ )a_B \quad R(\ )Fp + F_{ext} \tag{4.2}$$

4.2 Filtering - the Kalman Filter

The sensor readings concerning the attitude, accelerations, and thrust of the quadcopter present a high level of noise, which means that it is impossible to determine the tension of the quadcopter with the desired accuracy. To filter the undesired noise, and assuming that the noise is white Gaussian, a Kalman filter is implemented.

Equations 4.6 and 4.7 describe a linear system in which w_k and v_k are the process and the observation noise, respectively.

$$x_k = Ax_{k\ 1} + Bu_k + w_k \tag{4.3}$$

$$y_k = Cx_k + D + v_k \tag{4.4}$$

Additionally, denote Q and R as the process and observation noise covariance matrices.

$$Q = \begin{bmatrix} w_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & w_n \end{bmatrix}; R = \begin{bmatrix} v_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & v_n \end{bmatrix} \tag{4.5}$$

The Kalman filter presents an optimal solution for linear systems affected by white-Gaussian noise.

4.2.1 General information

The Kalman filter has two main steps, which are the prediction and correction. The prediction step estimates the system state having only into account the model (a priori estimation - see equation 4.6), and it also estimates the a priori error covariance, which represents the a priori state uncertainty (see equation 4.7). The a posteriori estimate of the state is then computed, having into account the a priori estimate, and also the current measurements (see equation 4.9), where the weight of each component is managed through the Kalman gain (see equation 4.8). On the one hand, if the measurements are very noisy the values in the R matrix are very large, and the Kalman gain goes to zero, meaning that the a posteriori estimate matches the a priori estimation ($\lim_{R\to 1} K_k = 0$). On the other hand, if the a priori estimation is too noisy, the values in the covariance error matrix (P_k) are very large, and the Kalman gain goes to $C^{\ 1}$. The a posteriori estimate has now only into account the sensor measurements ($\lim_{P_k\to 1} K_k = C^{\ 1}$). At the end, the error covariance matrix is also going to be updated to be used on the next iteration

Prediction:

$$\hat{x}_k = A\hat{x}_{k-1} + Bu_k \tag{4.6}$$

$$P_k = AP_kA^T + Q \tag{4.7}$$

Correction:

$$K_k = P_k C^T (CP_k C^T + R)^{-1} \tag{4.8}$$

$$\hat{x}_k = \hat{x}_k + K_k(y_k - C\hat{x}_k) \tag{4.9}$$

$$P_k = (I - K_kC)P_k \tag{4.10}$$

4.2.2 Filtering process using a constant model

The implemented solution considers a model where the tension remains the same, which is valid for hovering flights, and can be extended to situations where the wire is not abruptly pulled neither have considerable oscillations. The system's dynamic model is given by

$$\hat{x}_{k+1} = \hat{x}_k \tag{4.11}$$

The state estimate $\hat{x}_k$ is a $\mathbb{R}^3$ vector - including the tension along x, y, and z direction -, and y_k is the observation vector - including the measurements of the tensions. The observation vector is computed through indirect measurements, using the inertial information from the quadcopter. The state estimate and the observation vector, as well as A, B, C and D matrices, are represented in equation 4.12.

$$\hat{x}_k = \begin{bmatrix} T\hat{x}_k \\ T\hat{y}_k \\ T\hat{z}_k \end{bmatrix}; y_k = \begin{bmatrix} Tx_k^{obs} \\ Ty_k^{obs} \\ Tz_k^{obs} \end{bmatrix}; A = C = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}; B = D = 0 \tag{4.12}$$

The tension measurements T^{obs} are obtained through expression 4.1, by replacing its parameters with the sensor readings. Since the observation vector y_k is not obtained through direct observations, the measurement noise matrix R can not be built so straight forward as if the observations were directly taken from the sensors. This situation implies the need to establish the relation between the variance of each sensor with the variance of the measured tension. It is considered that the sensor readings have their results affected by white Gaussian noise with variance $_i^2$ and $= 0$. Making use of the variance properties and considering the observations as independent events, $Var[T_j^{obs}]$ is given by 4.13 - the complete deduction is presented in appendix B. R_{ij} is the element of the rotation matrix R( ) in the ith row and j^{th} column.

$$\begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix} = \begin{bmatrix} Var(T_x^{obs}) \\ Var(T_y^{obs}) \\ Var(T_z^{obs}) \end{bmatrix} = \begin{bmatrix} m^2\ Var(R_{11}:a_x) + Var(R_{12}:a_y) + Var(R_{13}:a_z) + Var(R_{13}:F_p) \\ m^2\ Var(R_{21}:a_x) + Var(R_{22}:a_y) + Var(R_{23}:a_z) + Var(R_{23}:F_p) \\ m^2\ Var(R_{31}:a_x) + Var(R_{32}:a_y) + Var(R_{33}:a_z) + Var(R_{33}:F_p) \end{bmatrix} \tag{4.13}$$

The Kalman filter is only optimal if the error measurements follow a Gaussian distribution. Figure 4.1 presents the empirical results of the error's probability distribution for the observed tensions along x, y, and z (equation 4.1), considering that the attitude, acceleration, and thrust error measurements have a Gaussian distribution.

(a) Error histogram for T_x^{obs}

(b) Error histogram for T_y^{obs}

(c) Error histogram for T_z^{obs}

Figure 4.1: Error histogram and Gaussian fit of tension along x, y and z

4.2.3 Filtering process using a non-constant model

In section 4.2.2, the Kalman filter models the tension of the tether as being constant. This section introduces a model that takes into account the first and the second derivatives, in order to predict the pulling movement that may be applied to the UAV. To simplify the development of the model, without loss of generality, consider only the tension along the x direction.

Assume that equation 4.14 defines the Kalman model in discrete time.

$$T_{x_{k+1}} = T_{x_k} + a(T_{x_k} - T_{x_{k-1}}) + b(T_{x_k} - T_{x_{k-1}} - (T_{x_{k-1}} - T_{x_{k-2}})) \tag{4.14}$$

Equation 4.15 represents the state space derived from equation 4.14, where parameters a and b represent the weight given to each difference term.

$$\begin{bmatrix} x_{1[k+1]} \\ x_{2[k+1]} \\ x_{3[k+1]} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ b & (a+2b) & 1+a+b \end{bmatrix} \begin{bmatrix} x_{1[k]} \\ x_{2[k]} \\ x_{3[k]} \end{bmatrix} \tag{4.15}$$

Equation 4.16 presents the matrices A, B, C, D, as well as the state and the observation vector of the

Kalman lter - $\hat{x}_k$ and y_k.

$$\hat{x}_k = \begin{bmatrix} Tx_k \\ Tx_k \\ Tx_k \end{bmatrix}^{2}; y_k = Tx_k^{obs}; A = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ b & (a+2b) & 1+a+b \end{bmatrix}; C = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}; B = D = 0 \tag{4.16}$$

To extend the solution to take into account the tension along y and z, instead of using the matrix A presented in equation 4.16, it would be necessary a new matrix A with dimensions 9 9. This new matrix A would have the matrix A presented in equation 4.16 in its diagonal. Furthermore, it would also be necessary to extend the dimension of matrix C to a 3 9 matrix, the dimension of the state estimate vector to 9 1 vector, and the observations vector to a 3 1 vector.

4.3 Simulation results - Tension estimation

4.3.1 Kalman lter using a constant model

To estimate the tension applied to the quadcopter, the Tension Estimation block is added to the Simulink model (see gure 4.2). This block outputs the value of the estimated tension, relying on the accelerometers' values, on the quadcopter's attitude, and the motors' thrust.

Figure 4.2: Simulink overview to estimate the tension applied to the UAV

To evaluate the performance of the Tension Estimation block, a force is applied to the quadcopter's center of mass. The most extreme case of simulating an abrupt pull consists of using a step signal to model it. In real-life examples, it is not recommended this kind of extreme tests, since it may turn the quadcopter ight unstable. In the simulations that were performed, the inclusion of those ”step-signals” during the ight did not make the ight simulation unstable, but the obtained responses presented a large overshoot, taking a long time to converge to its goal position. Nevertheless, this chapter aims to estimate the value of the applied tension correctly, and so the ight performance is neglected, once both

situations are independent of each other.

As a rst test, consider the most straightforward situation, where the sensors present noiseless measurements and there is no drag effect ($F_{ext} = 0$). In this case, it is not necessary to implement the Kalman lter since the value of the tension can be directly acquired from equation 4.1. Figure 4.3 illustrates the results obtained.

Figure 4.3: tension estimate without noisy measurements neither drag effect

The simulation includes the drag effect through a damping coef cient (Units - N=m:s ^{1}). The parameter F_{ext} is no longer zero and, due to that, some disparities emerge. Assuming that the measurements are still noiseless, gure 4.4 illustrate the obtained result.

Figure 4.4: tension estimate without noisy measurements and drag effect

As expected, the tension estimate no longer matches its real value. Since the drag effect is proportional to the ight speed, only when it approaches zero - hovering ight - it is possible to estimate the tension correctly.

To simulate the lack of accuracy of the sensors, their logged signals are passed through an Additive White Gaussian Noise channel (AWGN). The tension is estimated using the Matlab function block, and passes through the Kalman lter - section 2.4 - to suppress the noise added by the sensors. Figure 4.5 shows the scheme of the Tension estimation block where the different components are observable.

Figure 4.5: Simulink scheme inside the Tension estimation block

Figure 4.6 presents the estimate of the tensions after and before the Kalman lter.

Figure 4.6: tension estimate with noisy measurements and drag effect

The tensions estimated along y and z have values around zero, while the tension along x converges to -0.02N after the time instant t = 150s. On the one hand, the Kalman lter has a signi cant impact on reducing the tension's noise, since its variance is quite low when compared to the values obtained before the Kalman ltering. On the other hand, the constant model introduces a delay on the system's response when abrupt changes are applied.

4.3.2 Comparison between the constant and the non-constant model in Kalman lter

The performance of these two models is tested in Matlab for the case where the applied tension is a step signal, but without considering the drag effect, as shown in gure 4.7.

Figure 4.7: Comparison of the tension estimate by ltering with the constant and non-constant model

As it can be observed, the new model presents slight improvements, since it can converge faster to the real value and it also presents less noise for the steady values, as illustrated in gure 4.8.

Figure 4.8: Zoom of gure 4.7 for steady values

These results may look contradictory since the initial model is ideally “perfect” for modeling constant values. However, this new model allows increasing the con dence in the model (by decreasing the process noise variance), making it less susceptible to the observation noise without losing the ability to follow fast changes.

The parameters a and b are computed by trial and error. Nevertheless, in this process, it is necessary to take into account the stability of the state observer. The state observer error $e_{[k]}$ corresponds to the difference between the estimated state and the real state.

$$e_{[k]} = \hat{x}_{[k]} \quad x_{[k]} \tag{4.17}$$

Equation 4.18 presents the evolution of the state observer error, which is asymptotically stable if the eigenvalues of matrix $M = A \quad K_k C$ are all inside of the unit circle.

$$e_{[k+1]} = (A \quad K_k C) e_{[k]} \tag{4.18}$$

Making use of the previous Matlab simulation, one can compute the magnitude of the eigenvalues for the matrix M in each Kalman iteration, and as seen in gure 4.9, they are bounded between 0 and 1.

Figure 4.9: State observer error ($<^3$ vector)

4.4 Experimental results

The results obtained through the Matlab simulations have their advantages, but it is also essential to evaluate the accuracy of the procedures using real-life examples. To achieve that, the horizontal and vertical tensions are estimated, based on the theoretical results from section 4.1, and are compared to their ground-truth values. These ground-truth values are determined using the weight of the tether and/or the weight of a small mass (coin) attached to the UAV - see appendix A to see the tether's and quadcopter's speci cations.

Due to the limitations of the used test-bench, computing the ground-truth value of the horizontal tension whilst the quadcopter is ying may signi cantly decrease its accuracy. Given that, the estimation is performed assuming that the quadcopter is not moving, i.e., the accelerations of the UAV are null. Nevertheless, estimating the ground-truth of the vertical tension does not require a test-bench as for estimating the ground-truth of the horizontal tension, and so the accelerations are taken into account.

4.4.1 Estimation of vertical tension

The validation of the vertical tension estimate is done by performing a vertical tethered takeoff, which represents the simplest experiment to compute its ground-truth value Tv_{gt}. This last one is calculated through the height of the quadcopter z, and the weight per length unit of the the tether !

$$Tv_{gt} = !:z \tag{4.19}$$

The distribution of the measurement's error of the observed vertical tension T_z^{obs}, taking now into account the acceleration from the IMU sensors (equation 4.2), no longer matches a perfect Gaussian as before (see gure 4.1). Nevertheless, a good approximation can be done, as shown in gure 4.10

Figure 4.10: Variance of the error's measurement for the vertical tension

Once the gravitational acceleration is considered on the IMU accelerometers, the mean of the error can no longer be considered null. This causes the variation to slightly move to the value of the gravitational force causing also some deformation on the Gaussian.

Figure 4.11 illustrates the estimation of the vertical tension computed in three different scenarios: for a vertical takeoff of height 0,3m, 1m, and 1,3m. As it can be observed, the estimated tension converge

to the ground-truth value, and the implementation of a Kalman lter is essential to suppress the sensor's noise. Due to the force that the ground exerts on the quadcopter while this is not ying, the estimation of the vertical tension in the initial instants is not so accurate and does not converge to the ground-truth value. In the initial moments, the force that the ground exerts is undesirably taken into account on the tension estimate, and it approximately corresponds to the weight of the quadcopter. After the propellers reach enough speed, the thrust produced by the motors can compensate the quadcopter's weight, and the tension estimate converges to its ground-truth value.

(a) Vertical tension for 0,3m

(b) Vertical tension for 1m

(c) Vertical tension for 1,3m

Figure 4.11: Estimation of the vertical tension for different case scenarios