Università degli Studi di Padova

Dipartimento di Ingegneria
dell'Informazione

Corso di Laurea in Ingegneria
dell'Automazione

Attitude and Tension Control of a Tethered Formation of Aerial Vehicles

Laureando:
Marco Tognon

Relatore:
Ch.mo Prof. Ruggero Carli

Advisor:
Dot. Antonio Franchi

Anno accademico 2013/2014

Tognon, Marco:
Attitude and Tension Control of a Tethered Formation of Aerial Vehicles
Master Thesis Automation Engineer
University of Padua
Thesis period: Set 2013 - Mar 2014

Contents

Abstract

The Unmanned Aerial Vehicles (UAVs) are flying robots which can fly completely autonomously or be piloted remotely. These types of vehicles are very promising robotic systems for their flexibility, low cost and safety. These characteristics allow them to be applied in a lot of fields and for a lot of purposes, reducing the cost and danger associated with the absence of human pilots on board. Moreover, this is improved by the use of Miniature UAVs (MAVs), i.e. autonomous flying vehicles of narrow size, typically small enough to be man-portable. A popular vehicle that belongs to the family of UAVs is the quadrotor. It is a particular kind of rotor-craft propelled by four fixed rotors placed on the vertices of a square and independently actuated. By controlling the rotor speeds in different ways, the quadrotor can move in any direction of the space.

Nowadays, a very hot research topic regarding the UAV is the formation control in a multirobot scenario. The goal is to coordinate the behavior of a set of robots in order to achieve a task. In this thesis we deal with the problem of formation control exploiting external constraints. In particular, we want to tether two quadrotors to each other and to a fixed point by ropes. Then, we want to control the quadrotors in order to drive the orientation of the formation, keeping the cables taut.

At first, we analyzed the system deriving the dynamic equation. Then, using this latter, we designed a controller using the feedback linearization technique, in order to attain a desired orientation of the formation, trying to keep all the cables taut. Afterwards, we verified the performances by simulations, tuning the gains for the tracking of a desired trajectory of the orientation. Then, for the purpose to test the system also with real experiments, we implemented a testbed based on ROS and a new nano quadrotor: *Crazyflie*. The performances of the proposed controller were finally tested using realistic physical experiments.

Sommario

Gli Unmanned Aerial Vehicles (UAVs) sono robot in grado di volare in modo completamente autonomo o pilotati a distanza. Questi tipi di veicoli sono sistemi robotici molto promettenti per la loro flessibilità, basso costo e sicurezza. Quste caratteristiche permettono loro di essere impiegati in numerosi campi e per molti scopi, riducendo il costo e il pericolo associato all'assenza del pilota umano a bordo. Inoltre, questo è accentuato dall'uso di Miniature UAVs (MAVs), ovvero veicoli volanti autonomi di piccole dimensioni, in genere talmente piccole da essere trasportabili da un uomo. Un popolare velivolo che appartiene alla famiglia degli UAVs è il quadrotor. Esso è un particolare tipo di aeromobile ad ala rotante mosso da quattro rotori fissati ai vertici di un quadrato ed attuati indipendentemente. Controllando la velocità dei rotori in modo differente, il quadrotor può muoversi in qualsiasi direzione dello spazio.

Oggigiorno, un argomento di ricerca attuale riguardante gli UAVs è il controllo di formazione in uno scenario multirobot. Lo scopo è quello di coordinare il comportamento di un insieme di robots al fine di svolgere un compito. In questa tesi abbiamo affrontato il problema del controllo di formazione sfruttando vincoli esterni. In particolare vogliamo legare due quadrotor tra loro e ad un punto fisso tramite corde. Dopodiché vogliamo controllare i quadrotor al fine di muovere l'orientamento della formazione, mantenendo le corde tese.

All'inizio, abbiamo analizzato il sistema derivando l'equazione della dinamica. Poi, partendo da quest'ultima, abbiamo progettato un controllore utilizzando la tecnica della feedback linearizzazione, al fine di ottenere un orientazione desiderata della formazione, cercando di mantenere le corde tese. Dopodiché, abbiamo verificato le prestazioni attraverso simulazioni, regolato i guadagni per l'inseguimento di una traiettoria desiderata dell'orientazione. Successivamente, con lo scopo di testare il sistema anche con esperimenti reali, abbiamo implementato una piattaforma basata su ROS (Robot Operating System) e su un nuovo nano quadrotor: il *Crazyflie*. Le prestazioni del controllore proposto sono state alla fine testate con esperimenti realistici.

Acknowledgements

First of all, I'd like to thank *Dr. Antonio Franchi*, the group leader of the *Autonomous Robotics and Human-Machine Systems group of the Max Planck Institute*, as well as my advisor, for having allowed me to develop my thesis in this group. Here I had the opportunity to learn a lot, to grow up my knowledge and maturity, and to have this experience in a different state. He supported and helped me during all the period spent at the Max Planck Institute, giving me hints and suggestions for the success of the thesis.

Then, I would like to thank my professor and supervisor from the university of Padua, *Prof. Ruggero Carli*, for the great opportunity to work in an innovative research institute, and also for the support during all the period abroad.

Furthermore, I'm grateful to the other members of the group of the Max Planck Institute who helped me in many occasions, answering my frequent questions, in particular to become familiar with the tool of the laboratory.

A special thanks goes to *Giorgia Fiocco* who supported me during my stay in Germany. Although the distance, she gave me the strength to accomplish my work in many difficult periods, even ignoring her personal tasks only for me. The success of this thesis is also thanks to her.

Then, I want to thanks to my colleagues and friends *Carlo Gerboni*, *Matteo Gagliardi*, *Guido Gioioso* and *Massimo Basile*, for the nice time spent together, at work, in the laboratory, and outside the office. I will miss the dinners together at the weekends to forget a long week of work.

Lastly, I want to use this occasion to thank my family. They supported me during all my studies and specially for this period abroad, understanding my often closed and introvert character.

List of Abbreviations

API	**A**pplication **P**rogramming **I**nterface
CPU	**C**entral **P**rocessing **U**nit
DoF	**D**egrees **o**f **F**reedom
ECF	**E**xplicit **C**omplementary **F**ilter
e.g.	for example (from Latin: **e**xempli **g**ratia)
GPS	**G**lobal **P**osition **S**ystem
HO	**H**armonic **O**scillator
i.e.	that is (from Latin: **i**d **e**st)
IMU	**I**nertial **M**easurement **U**nit
MAV	**M**iniature UAV
PC	**P**ersonal **C**omputer
PID	**P**roportional **I**ntegral **D**erivative
PWM	**P**ulse **W**idth **M**odulation
ROS	**R**obot **O**perating **S**ystem
SLS	**S**tandard **L**inear **S**olid model
SUAV	**S**mall UAV
TK	**T**ele**K**yb
UAV	**U**nmanned **A**erial **V**ehicle
VTOL	**V**ertical **T**ake-**O**ff and **L**eanding

Chapter 1

Introduction

1.1 Introduction

The Unmanned Aerial Vehicle (UAVs), and in particular the quadrotors, are the focus of many research in the robotics area, with a enormous increasing interest over the last decade. This can be explained by the improvement of the technology, making possible to use miniaturized, slight and cheap sensors, as well as microcontroller with good, although limited, computational capability. Thanks to this, the quadrotors became cheap and accessible by whoever, making them really spread and studied for many applications, ranging from environmental monitoring over search and rescue to military tasks. Many of these tasks take place in hostile environments and therefore, using UAVs, humans are prevented from getting harmed in numerous cases.

Although in this fields we can already find some real and commercial solutions, there are still many research open problems and projects involving specially the multiagent control and the environmental interaction. These topics are particularly challenging for several reasons. The first issue in a cooperative scenario is that the quadrotors must to safely manage themselves in the whole group in order to achieve a task, taking care about their neighbors. Moreover, this is made more complicated due to the limited sensors capability of the quadrotors, that reduces the knowledge of the surrounding environment and the position of its neighbor. Often, the information needed

in these scenarios, for instance in the formation control, is the relative position and orientation between the robots. This kind of information is not easy to be gathered if we think at the limitations of cameras and other range sensors based on infrared or sound.

Then, when we want interact with the environment with robots, many unknown forces, torques and disturbances in general come out, and their effects can destabilize the system. A typically scenario where we can find all these problems, is represented by the cooperative manipulation and transport of payloads suspended by cables by multiple quadrotor robots. In this case, the robots have to cooperate with each others to move a payload avoiding collision and unwanted behavior of the load, as swings, dealing with its reactions and undesired effects.

This thesis aims to investigate these types of problems. In particular, we want to start studying a new strategy of formation control which uses external constrains to maintain a desired formation, avoiding the whole knowledge about the quadrotors position. Our method is based on tether the quadrotors by cables between them and at some anchored points, in order to attain a desired formation. Then we want to design a controller to keep the cables taut and to control the entire formation, as an actuated tensegrity[1] structure (Fig. 1.1).

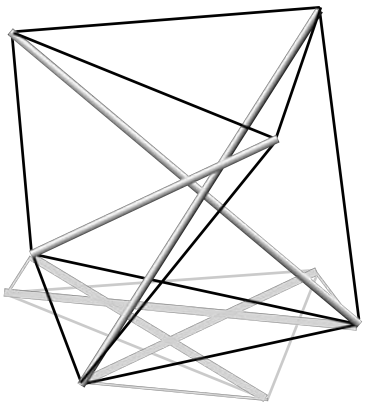


Figure 1.1: Example of a tensegrity structure.

[1]**Definition (Hanaor, 1994)** internally prestressed, free-standing pin-jointed networks, in which the cables or tendons are tensioned against a system of bars or struts [1].

Moreover, if you think to connect this system to an object, instead of some anchors, it will be possible to manipulate it controlling the formation, such as position and orientation. The idea is really similar to a parallel cable robot where the actuators are quadrotors instead of fixed motors [2].

Since this is a novel strategy, the thesis will examine the analysis and the control design for a preliminary and restricted problem which consists of a limited formation of two quadrotors tethered to each other and to one anchor fixed on the floor. In this way the formation is basically represented by a triangle having one of the corners fixed. Controlling the attitude of the two quadrotors, we want to control the orientation of the formation, in particular its rotation around the anchor point, keeping each cables taut all the time.

1.2 State of Art

Extensive research related to the transporting and manipulating objects with quadrotors and helicopters have been done in the last years. Indeed, many of the major research institutes working with quadrotors have developed systems and algorithms for the problem of transporting cable suspended payloads. Using autonomous aerial vehicles to transport any types of material is more safe and suitable from an economical point of view. Indeed we can find an real example of application at Amazon, which is developing a faster delivery service based on quadrotors. Although this should take a while to be implemented, there are many types and strategies of control to stabilize the flight of a quadrotor carrying a payload [3] [4] [5] (Fig. 1.2) . Additionally, in [6], the authors have developed a learning method to eliminate undesired swings on the object during the movement, and in [7] the authors also have taken account of the possibility of an flexible cable.

However, since the quadrotors have usually a poor payload, you can not carry and transport big and heavy object. In order to manage bulky loads, we can improve the power of the single quadrotor building it bigger for mounting powerful motors, or increasing the number of propellers making an hexacopter or a octocopter [8] [9] [10]. Another way is to use more than one quadrotor in order to increase the total power of the system making possible

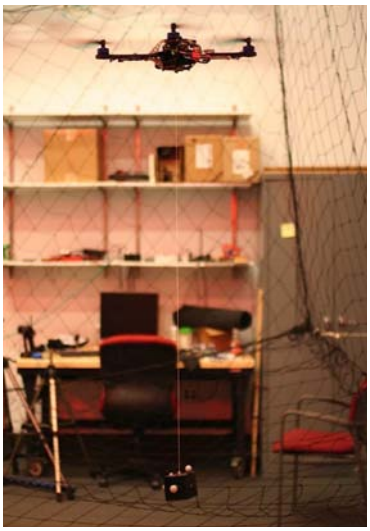


Figure 1.2: Quadrotor flying while carries a cable suspended load.

to manipulate bigger and heavier loads. In [11] the authors have developed a control and planning framework for cooperative manipulation of payloads suspended by cables. In particular, given a desired trajectory of the load, the controller dynamically compute a feasible trajectory for the quadrotors. Another example is in [12] where the authors have designed a controller to carry a flexible payload directly attached to the quadrotors. The control framework sends the commands in order to control the payload position and, at the same time, to minimize the deformation of the object.

The main drawback of all these solutions implemented for the single and multi quadrotor transport problem, is the need of a complete knowledge of the state of whole system, i.e., robots and load. Basically, only indoor applications are possible, where you can mount an tacking system able to give the position of all the objects of the system. Whereas, an outside application, where is not possible to have a tracking system, are not feasible.

One related work in this field, which doesn't require any external motion capture system, was done by Lupashin in [13]. He designed a controller and estimator for hover-capable flying vehicle attached to a fixed point by a taut tether able to attain a desired position in the vertical plane, everything without a tracking system. Indeed, to recover the vehicle's relative position and

absolute orientation, he used an estimation method which requires only onboard inertial sensors, and indirectly measures the string force, enabling the additional use of the tether as a physical user interaction medium (Fig. 1.3).

Figure 1.3: *Fotokite* [14] is a flying camera based on the system designed by Lupashin in [13].

Also from the point of view of the formation control a lot is already done. In this field of research, the goal is to design a global or distributed controller for each agents, in order to attain a desired formation of the quadrotors, and than drive it as an unique system to complete some tasks. Applications that rely on the multi-agent solutions vary from interferometry in deep space, distributed sensing and data collection, surveillance, construction and transportation, and search and rescue operations.

There are many techniques and approaches to the problem of the multirobot control, and in particular to the formation control. In [15], to control the swarm, the authors propose an algorithm to generate dynamically feasible trajectories for aerial robots navigating through cluttered known environments. Other methods use a leader-follow approach to the problem as in [16] [17] [18], where one robots is labelled as leader and its motion defines the bulk motion of the group. Then, the motion of individual members within the formation is described with the respect to the lead robot. Another very popular techniques used to control swarms of robot is based on artifi-

cial potential field. In [19] [20] [21] an artificial potential is used to define interaction control forces between neighboring vehicles and are designed to enforce a desired formation. Moreover, particular kinds of artificial potential are designed to attain specifics task, for example to maintain the connectivity or the rigidity of the formation [22] [23] [24]. At the end, using this kinds of formation control, we can abstract the swarm as a unique object that could be manipulated by some parameters, such as position and orientation [25] [26].

The majority of the previous controllers need at least that each agent knows the relative position of its neighbours. Thus, also in these cases, although the onboard sensors are becoming smaller, slighter and with better performances, actually they are often not sufficient to implement the cited control's strategies. So external tracking system are used which, as already said, are not feasible for outdoor applications.

1.3 Thesis Objectives

The purpose of this thesis is to design a formation control system which uses external physical constrains to keep a desired formation. Instead of using the techniques reported in the previous section, we want to use external constrains allowing to avoid the estimation of the relative position from the agents. If we model the swarm as a graph, where each agent corresponds to a node of the graphs, we want to connect the robots with physical links, i.e. cables, in order to create a rigid graph [27] [28] (Fig. 1.4). Then, keeping the

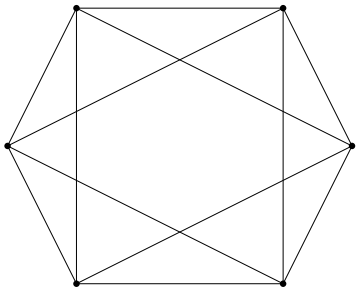


Figure 1.4: Example of rigid graph with six agents. The dots represent the robots, whereas the edges represent the cables.

cables taut every time, the formation will be fixed. Moreover, tethering the system to some anchor points, leaving some degrees of freedom, we want to control the formation's orientation with respect to these points.

In this thesis we will investigate a restricted system composed by two quadrotors and one anchor connected to each others by three cables, in order to form a triangle (Fig. 1.5). Our goals is to derive a controller to attain a desired orientation with respect to the anchor, keeping the cables taut. We want also implement this system and test it with real experiments based on a new nano quadrotor: Crazyflie.

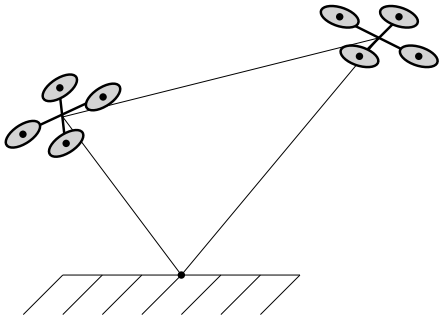


Figure 1.5: Tethered system composed by two quadrotors and one anchor, bound together by cables. The resulting formation is expressed by a triangle.

1.4 Motivations

As we noticed in Sec. 1.2 there are already many methods and solutions for the formation control problem. However, these need a set of informations that includes the relative positions between each agent that, by now, are only available from external tracking systems. Moreover, the complete autonomous flight, using only on board sensor, is only at the first steps. So, as we saw in the literature, the existing solution are feasible only for indoor application, in areas equipped with motion capture systems.

The reason why we want to use physical constraints to fix the relative position, is to avoid the need of the information not directly available from onboard sensors. Indeed, as showed in [13], using external physical constraints it is possible to infer the position information from only onboard

inertial sensors. Although this thesis focus on the control aspect, we want to extends the principles in [13] to a multiagent case, expanding the results for a system composed by two agents.

The use of multirobot system bring many advantages. For examples, as we mentioned before, it lets to transport and manipulate heavier load which can not be done by a single quadrotor. Using our system, connecting the anchor to a load, it could be possible, using other cables attached to the load, to directly transport the object, without any trajectory planner. This will let the user to manage the load as an air balloon. Other direct applications are the scanning and mapping of environments, rescue, surveillance and so on. Indeed, with more agents equipped with appropriate sensors, as cameras or scanner lasers, we can inspect a wide area in a shorter time and more accurately.

1.5 Thesis Outline

In the Chap. 2 we will present a summary about the quadrotor: why they are so important today, the history of the design of the quadrotor and its first flight, the principles of flight, the dynamic model, and finally the common control techniques and estimation methods.

Afterwards we will start to analyse our problem deriving, in the Chap. 3, the dynamic model of the system. In next chapter we will present the design of the controller based on an advanced non linear control technique (Feedback Linearizzation). In the Chap. 5 we will show the simulation validation of the controlled system showing the results of the tuned controller for the tracking of a desired trajectory of the formation's orientation.

After the theoretical end simulation validation, we started to design a testbed to conduct some real experiments. Thus, in the Chap. 6 is presented the designed and implemented platform based on ROS (Robot Operating System) and on a new type of nano quadrotor. At the end we will show the result obtained from the experiments highlight the major problems. In last chapter we will summarize the work of this thesis suggesting some future works to improve the controller and to extend the project.

Chapter 2

Quadrotor

2.1 Introduction

The quadrotor belongs to the big family of the Unmanned Aerial Vehicles (UAVs), also commonly called Drones. They are flying robots which can fly completely autonomously or be piloted remotely, more in general they are aerial vehicles that don't need a human operator on-board. These types of systems are becoming very widespread and famous for their flexibility, low cost and safety. These characteristics allow to apply them in a lot of fields and for a lot of purposes. At first, UAVs were mainly used on Military field, such as for spy missions, autonomous weapons, reconnaissance, survey and other types of missions.

Then, technological advances have made sensor payloads smaller, lighter, and more capable and have greatly increased the bandwidth connectivity of the data links used for vehicle command and control, payload command and control, and data transfer. Advances in microprocessor technology and software development have enabled on-board processing of sensor data, while advances in inertial and GPS navigation have enabled robust autonomous flight control systems. The development of the above technology made these vehicles more accessible and widespread, and therefore they started to be used also on civil field, mainly for remote sensing [29] [30]. The UAV Remote Sensing consists on an UAV which has on-board remote sensing technology

such as measuring sensors, GPS (Global Position System) and a communication system which allow to acquire data quicker, cheaper, and safer than with piloted aircraft (Fig. 2.1).

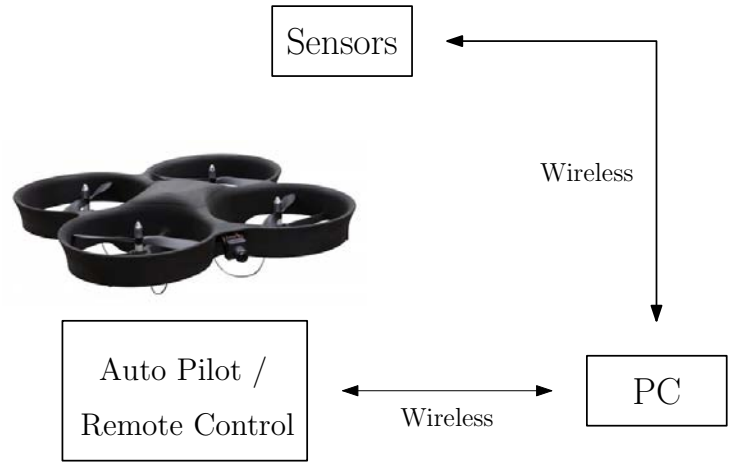


Figure 2.1: Remote Sensing

For this reasons they are used on agricultural activities, for example paddy monitoring and spraying, but also for rescue missions after environmental disasters such as storms, earthquakes, forest fires and so on. Other examples of application of the UAVs are commercial and motion picture film-making, domestic policing, scientific research, maritime patrol and so on. Indeed, an UAV equipped with sensors like cameras, thermometers, sensors for emissions, can be utilized to scan an area of interest, that could be inaccessible for humans, returning all the necessary data. Moreover, thank to the the miniaturizing of the technology, it has been possible to develop Miniature UAVs (MAVs) and Small UAVs (SUAV), i.e. autonomous flying vehicle of narrow size. Although they have low payloads and, consequently, low on-board computational capabilities and poor sensor equipment, actually, they can be easily carried and launched by a man also in small and narrow area such as indoor environments.

Nowadays there are really many types of UAVs available. We can classify them by the weight, the endurance and range, the maximum altitude, the engine type, the type of mission and so on. In any case, we can define two main families of UAV: the first consists on autonomously vehicles with fixed

(a) (b)

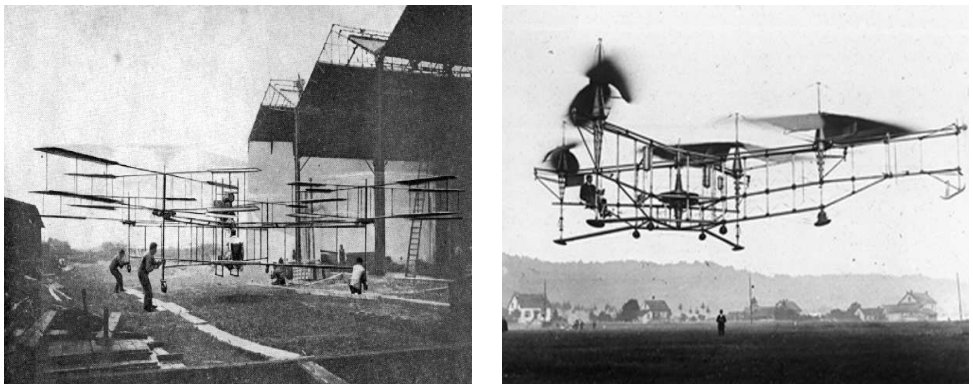
Figure 2.2: Two different types of UAVs. (a): MQ-9 Reaper is a fixed wing hunter-killer surveillance UAV. (b): MQ-8 Fire Scout is an unmanned autonomous helicopter.

wings (Fig. 2.2a), the second, instead, consists on the Vertical Take-Off and Landing (VTOL) vehicles (Fig. 2.2b).

Fixed wing UAVs are essentially similar to the normal passenger aircraft, which use wings to provide lift, and a propeller or a jet to provide forward thrust. On the other hand, the family of VTOL vehicles, includes all the aerial vehicles which are able to hover, take-off, and land vertically. Helicopters, quadrotors, ducted-fan and tail-sitter belong to this last family. The particularity of this types of vehicles, differently from the fixed wing UAV, is that they can hover, i.e. they can stay on a fixed point. This allows them to be utilized in a lot of applications where is necessary to flight over a certain object such as surveillance, inspection, or interaction with the environments [31] [32].

2.2 History

The quadrotor (Fig. 2.3), also called quadcopter or quadrocopter, is a particular rotorcraft, belonging to the family of VTOL vehicles. The first prototype of quadrotor that we can find on the history is the *Gyroplane No.1* (Fig. 2.3a), designed on the 1907 by Louis and Jacques Breguet in association with Professor Charles Richet [33] [34]. Clearly the Breguet brothers approached the problem of the helicopter more scientifically than others at



(a) *Gyroplane No.1.* (b) *Oehmichen No.2.*

Figure 2.3: Examples of quadrotors.

the time, and thought hard about the design of the vehicle, in order to solve also the stability problem, although the first requirement for the machine was simply to lift itself and a pilot off the ground under its own power. The Breguet-Richet quadrotor consisted of four long girders made of welded steel tubes forming a horizontal cross. The rotors were placed at each of the four corners of the cross, and the pilot sat in the center of the structure below an internal combustion engine. The first experiments with their *Gyroplane No.1* were carried out at Douai, France, where the machine with its pilot reached an heigh of 1.2 [m].

After the first experiments on the design of a quadrotors, others researchers started to deal with this vehicle. In the 1920s, Etienne Oehmichen made its own quadrotor, *Oehmichen No.2* (Fig. 2.3b) which exhibited a considerable degree of stability and controllability being able to hover for several minutes.

Then the researches on this topic kept on, but without important success until the late 1990s when, thanks to the rapid growth of MEMS (Micro Electro Mechanical Systems), which make possible the design of micro controller, the design and control of quadrotor became an interesting topic for many researchers [35]. Initial attempt to build micro quadrotor was launched by hobby production but many research groups opted for designing their own quadrotor instead of modifying commercial RC quadrotors, like in [36], start-

ing to design, implement and test many types of control to stabilize the flight and to track a desired trajectory.

In the middle of the first decade in the 21th century, most pioneers in the quadrotor research field had finished their prototypes and tried many kinds of existing control algorithms on their separate testbeds. From that time on, research on the quadrotor diversified in a lot of fields, applications and topics such as vision based control, autonomous flight, multiagent control, environmental interaction and so on. For instance, GRASP lab, from the University Of Pennsylvania, in [37] [38] developed a formation control algorithm, and in [11], it considered the planning and control of multiple aerial robots manipulating and transporting a payload in three dimensions via cables. Another promising direction consists on the control for aggressive maneuvers. In [39] is designed a learning control algorithm for perching maneuver. Moreover, Lupashin from ETH Zurich, developed a simple and intuitive policy gradient method for improving quadrotor multi-flips in [40]. Although extensive research on quadrotors has been performed, this field is still full of open problems end topics for research.

2.3 Principles of flight

Basically it consists of a rigid body with four rotors propelled by brushless DC electric motors mounted at the edges of a rigid cross frame. Controlling the rotary speeds of the propellers and in particular making unequal the rotational speed of the two opposing rotors, we can control the global thrust and the torques acting on the body, and so we can move the vehicle in any direction. Since it can move in a three dimensional space, it has six degree of freedom, three for translational motion and three for rotation, through, it has only four control input, i.e. the four propellers, therefore it follows that the quadrotor is an underactuated system.

To have hovering, the quadrotor has to be horizontal and the four propellers have to turn at the same speed, producing a global vertical thrust that compensates the gravity. Then, increasing or decreasing the total thrust, we obtain the vertical motion. Instead, for the horizontal motion, the quadrotor

has to tilt in the desired direction in order to obtain an acceleration component on the $x - y$ plane, then increasing the thrust to compensate the gravity. More precisely, the quadrotor has to do a rotation along the x-axes of a certain angle ϕ (roll) to move in the y direction, or along the y-axes of a certain angle ϑ (pitch) to move in the x direction. These rotations of the body frame are achieved changing the power of the motors along the same axes (Fig. 2.4).

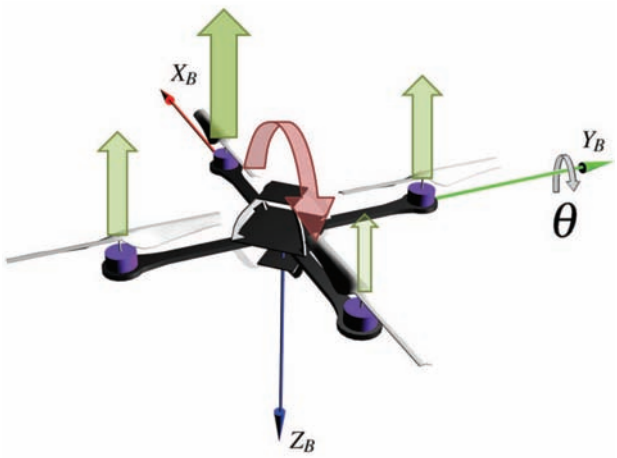
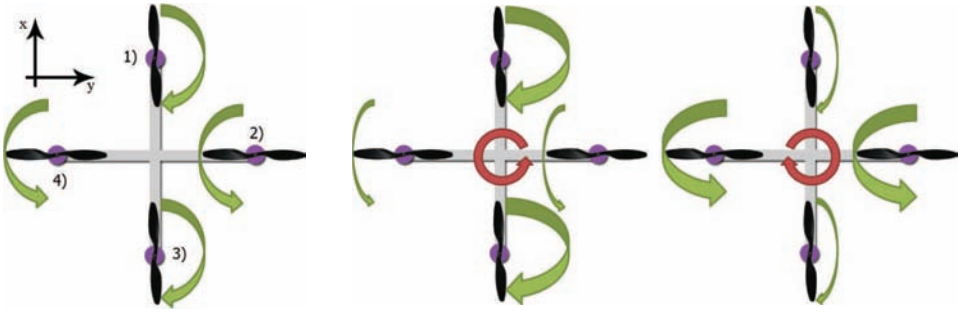


Figure 2.4: Example of a quadcopter pitching.



(a) Propellers rotation. (b) Yaw rotation.

Figure 2.5: Rotation along the z-axes. (a): motors one and three turn clockwise; motors two and four turn counterclockwise. (b): on the left, increasing the velocity of the motors one and three gives a counterclokwise rotation of the vehicle along the z axis; on the right the opposite.

The rotation of the propellers, due to the air resistance, generates a react-

ing force opposite to the rotation of the rotor. For this reason, the propellers on one axes turn in the opposite direction than those on the other axes. For example, in Fig. 2.5a, the propellers (1,3), along the x-axes, rotate clockwise, instead, the opposite propellers (2,4), along the y-direction, rotate counter-clockwise. In this way, when all rotors turn at the same speed, as in the hovering state, all of them produce the same reacting torque and, due the different rotation, the sum is zero and there is no rotation along the z-axes. On the other hand, to have a rotation along the z-axes of a desired angle ψ (yaw), it is sufficient to increase the power of the motors along the same axes (Fig. 2.5b). With the inequality of the reacting forces we obtain a certain global torque along the z-axes, and thus the rotation.

2.4 Applications

As we have just understood, the physic's principle of flight of a quadrotor, and its control, is pretty simple. For this reason, in the last years, they were abundantly studied by the research community, and now, they start to be used in a lot of fields and for various purposes, replacing the commons one rotor helicopters. Indeed, the traditional helicopters control is a function of the orientation of the main rotor. Thus, for moving in the space, you must change the its orientation in order to change the direction of the craft. This makes the mechanical linkages very complex and it complicates the dynamics and thus the helicopter control too. Also three-copters are more difficult to drive because their dynamics includes an imbalance of the moments induced by the spinning of the rotors, moreover they require a servo to tilt one or more rotors which is more mechanically complicated.

On the other hand, with four rotors or more (normally even number) you get improved stability which allows to flight very easily. Moreover, using more and more sophisticated controllers, which use also haptic feedback devices and on-board cameras, you can remotely drive it in a very intuitive way [41] [42] [43] [44]. Then, using four rotors instead of one, allows to reduce the diameter of the propellers making the quadrotor more agile. Indeed, with it, it is possible to do a lot of stunts, such as flips and rolls [40] [45] and also to

execute quick flight maneuvers and particular and complex trajectories [46].

Moreover, in case of crash with obstacles or walls, since quadrotors are often smaller end more compact than helicopters, the damages are usually less and the reparations are easier and cheaper.

(a) Pars Rescue Robot.

(b) Aeryon Scout.

(c) Amazon Prime Air.

Figure 2.6: Examples of quadrotors applications.

All these characteristics see that the quadrotors are used in a lot of application such as search & rescue, inspection in hazardous environments, surveillance, archeology and renovation, delivery and so on. For instance, since quadrotor is able to hover and, using obstacle avoidance systems, to fly in narrow space, it cold be used for rescue missions after earthquakes or tsunami. Indeed, may with the help of an human operator, the quadrotor could search for trapped persons giving them medical aim. Moreover, the

rescue team, could use the images to design the best path to reach the objective and pull to safety the person. The Fig. 2.6a shows an example: the Pars, designed and made by the RTS lab, it is used in ships and off shore relief for saving human lives [47].

The quadrotors could be very useful also in other dangerous situations, such as chemical or nuclear incidents (for example *Fukushima, 2011*), or more in general in places damaging and inaccessible for the human being. Using a quadrotor we could inspect the area, doing some measurements on the air or on the plot, such as the pollution or the radioactivity, in a totally safety way, without jeopardizing the health of a person.

Another natural field of application is the surveillance, indeed, the ability of the quadrotor to hover over an objective, makes it perfectly suitable for patrolling an area or to track a person or a vehicle. An example is the Aeryon Scout (Fig. 2.6b) created by Canadian company Aeryon Labs. It is equipped with a powerful camera to point people and objects on the ground. The company claims that the machine has played a key role in a drug bust in Central America by providing visual surveillance of a narco-trafficker's compound deep in the jungle [48]. Moreover it is used as support to the soldiers in some delicate military missions, by the police in the traffic and accident management, and by the fire fighters in the fire investigation and damage assessment.

But, due to its facility of control and for its accessibility, the quadrotors are also starting to find employments in civil applications. One of the most actual example is in Amazon, a big delivery company. It started to develop a new delivery method, *Amazon Prime Air*, based on quadrotors (Fig. 2.6c). Using these vehicles they want to accomplish the delivery of a package in 30 minutes or less.

2.5 Dynamic Model

Notations	
$\{A\}$	inertial frame
$\{B\}$	body-fixed frame
$m \in \mathbb{R}$	total mass
$I \in \mathbb{R}^{3\times3}$	inertia matrix with respect to the body-fixed fame
$R \in \mathbb{R}^3$	rotation matrix from $\{B\}$ to $\{A\}$
$\eta \in \mathbb{R}^3$	vector of the RPY-Euler angles, $\eta := \begin{bmatrix}\phi, & \vartheta, & \psi\end{bmatrix}^T \in \mathbb{R}^3$
M_i	i-th motor
$r \in \mathbb{R}^3$	position of the center of mass in the inertial frame
$v \in \mathbb{R}^3$	linear velocity of the center of mass in the inertial frame
$d \in \mathbb{R}$	distance of each rotor from the center of mass
$f_i \in \mathbb{R}$	thrust of the i-th rotor directed along the $-b_3$ axis
$\omega_i \in \mathbb{R}$	angular velocity of the i-th rotor
$f \in \mathbb{R}$	total thrust
$F \in \mathbb{R}^3$	total thrust in the inertial frame
$\tau \in \mathbb{R}^3$	total moment in the body-fixed frame
$X \in \mathbb{R}^{12}$	state of the system, $X := (r, v, \eta, \Omega)$
$u \in \mathbb{R}^4$	control input, $u := \begin{bmatrix}f, & \tau\end{bmatrix}^T$

To derive the dynamic model of the quadrotor we used the usual Newton-Euler approach as in [49] [50] [51]. At the first step we have to fix the frames to describe the quadrotor's position and orientation. Let $\{A\}$ a right-hand inertial frame with axes $\{\vec{a}_1, \vec{a}_2, \vec{a}_3\}$ expressed in $\{A\}$. Then, $\{B\}$ is the right hand body fixed frame whose center coincides with the center of mass of the quadrotor and whose axes are $\{\vec{b}_1, \vec{b}_2, \vec{b}_3\}$ respect to the frame $\{A\}$. The first and second axis of the body fixed frame, $\vec{b}_1, \vec{b}_2$ lie on the quadrotor's plane, defined by the center of the four rotors. In particular $\vec{b}_1$ stays on the line joining motor one, $M1$, and three, $M3$; Whereas, $\vec{b}_2$ stays on the line joining motor two, $M2$, and four, $M4$. On the other hand, the third axis $\vec{b}_3$ is normal to this plane, and directed downward, opposite to the total thrust (Fig. 2.7).

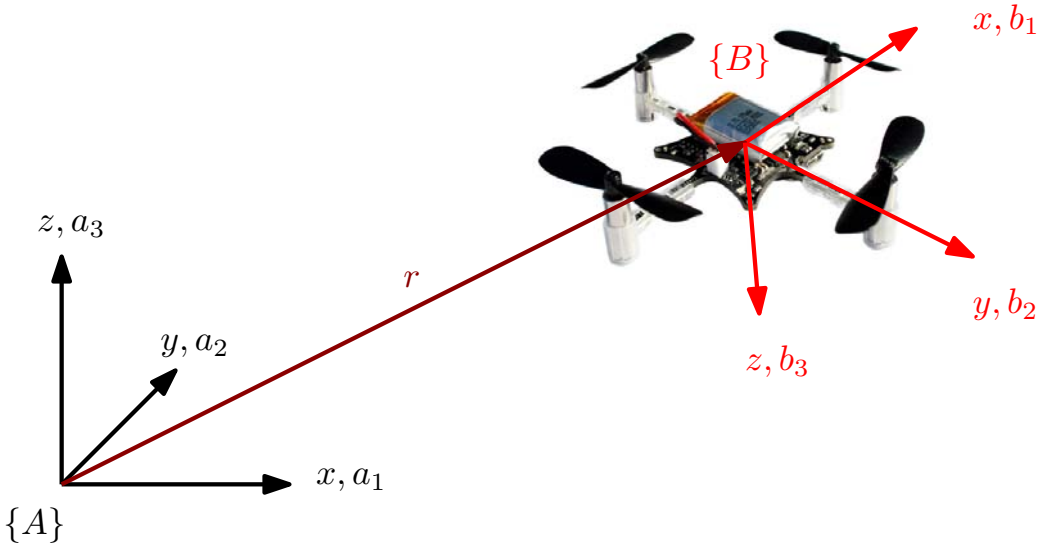


Figure 2.7: Inertial and body fixed frame.

Now the configuration of the quadrotor can be expressed by the position and the orientation of the frame $\{B\}$ with respect to the inertial frame $\{A\}$. We denote $r = (x, y, z) \in \{A\}$ the position of the center of mass of the quadrotor and $R = [\vec{b}_1, \vec{b}_z, \vec{b}_3] \in SO(3)$ the rotation matrix of $\{B\}$ with respect to $\{A\}$. We can also represent the orientation of the robot using the Z-X-Y Euler angles [52], commonly called yaw, roll and pitch respectively. In this representation, the rotation in given by three consecutive elementary rotation along each axes. We first rotate about a_3 by the yaw angle ψ, then about the x axis of the rotated frame by the roll angle ϕ, followed by a rotation about the new y axes through the pitch angle ϑ. Let $\eta := \begin{bmatrix}\phi, & \vartheta, & \psi\end{bmatrix}^T \in \mathbb{R}^3$ the RPY Euler angles, then, using this consequently rotation method, the rotation matrix is given by:

$$R(\eta) = \begin{bmatrix} c\psi c\vartheta - s\phi s\psi s\vartheta & -c\phi s\psi & c\psi s\vartheta + c\vartheta s\phi s\psi \\ c\vartheta c\psi + c\psi s\phi s\vartheta & c\phi c\psi & s\psi s\vartheta - c\psi c\vartheta s\phi \\ -c\phi c\vartheta & c\phi & c\phi c\vartheta \end{bmatrix} \tag{2.1}$$

where c stands for $\cos(\cdot)$ and s stands for $\sin(\cdot)$.

Now that we defined how to describe the state of the quadrotor, we start the derive the dynamic behavior. The thrust generated by each rotor, in

steady state, may be modelled as:

$$f_i = c_T \omega_i^2 \tag{2.2}$$

where the constant $c_T \in \mathbb{R}$ depends on the aerodynamic properties of the propeller and can be easily estimated from static thrust tests. Though, since the dynamics of the rotors and propeller is faster than those of the quadrotor, we can neglect it and assume that the thrust of each propeller is directly controlled [46] [53]. Then, assuming all the rotors fixed exactly according to the $\vec{b}_3$ axis, the direction of all the thrusts is normal to the quadrotor's plane. From this, it comes out that the total thrust acts along the direction of $-\vec{b}_3$, and it is:

$$f = \sum_{i=1}^{4} f_1$$

Moreover, according to the definition of the rotation matrix R, the body fixed frame axis b_i correspond at the i-th column of R, i.e. $b_i = Re_i$, where $e_1 = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}', e_2 = \begin{bmatrix} 0 & 1 & 0 \end{bmatrix}', e_3 = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}' \in \mathbb{R}^3$. Therefor, the total thrust in the inertial frame is

$$F = -fR(\eta)e_3 \in \mathbb{R}^3$$

If we assume that the rotors one and three rotate clockwise, whereas the others rotate counterclockwise, the reacting torque (due to the rotor drag [49] [54]), acting on the airframe may be modelled as $Q_i = (-1)^i c_Q f_i$, where $c_Q \in \mathbb{R}$ depends on the aerodynamic properties of the quadrotor. Then, if we assume that the center of mass is in the intersection of the two lines connecting the two opposite motors, and that the distance of each rotor from the center is constant $d \in \mathbb{R}$, the general torques acting on the center of gravity are:

$$\tau = \begin{bmatrix} \tau_x \\ \tau_y \\ \tau_z \end{bmatrix} = \begin{bmatrix} d(f_4 - f_2) \\ d(f_1 - f_3) \\ c_Q(f_2 + f_4 - f_1 - f_3) \end{bmatrix}$$

Summarizing we can write the total thrust f and the total moment τ as

a matrix relation:

$$\begin{bmatrix} f \\ \tau_1 \\ \tau_2 \\ \tau_3 \end{bmatrix} = \underbrace{\begin{bmatrix} 1 & 1 & 1 & 1 \\ 0 & -d & 0 & d \\ d & 0 & -d & 0 \\ -c_Q & c_Q & -c_Q & c_Q \end{bmatrix}}_{\Gamma} \begin{bmatrix} f_1 \\ f_2 \\ f_3 \\ f_4 \end{bmatrix} \tag{2.3}$$

From the previous equation, given a desired thrust and moments, we can determine the thrust of each motors, f_i, inverting the equation (2.3) and afterward, using the relation (2.2), we can find the command velocity for the motors. The inverse of the relation (2.3) is always possible because the matrix $\Gamma \in \mathbb{R}^{4\times4}$ is full rank, indeed, the determinant is $8c_Qd^2$ that is different from zero since $d \neq 0$ and $c_Q \neq 0$. For instance, in order for the vehicle to hover, we must set $\tau = 0$ and $f = mg$, and then to calculate the suitable ω_i inverting the matrix Γ. From this equations we can consider the total thrust f and the total moment τ as control input of the quadrotor $u := [f, \tau]^T$.

The configuration of the quadrotor can be described by its position, r, and its orientation, $R(\eta)$. To describe its state, we can also use the derivatives of these quantities. Let $v \in \{A\}$ denote the linear velocity of the center of mass expressed in the inertial frame $\{A\}$. Let $\Omega \in \{B\}$ denote the angular velocity of $\{B\}$ with respect to $\{A\}$, expressed in $\{B\}$. Let $m \in \mathbb{R}$ denote the mass of the rigid body and $I \in \mathbb{R}^{3\times3}$ denote the inertia matrix around the center of mass, expressed in the body-frame $\{B\}$. To derive the dynamic model of the quadrotor, defining the state of the system $X := (r, v, \eta, \Omega)$ we can simply use the Newton-Euler formalism that yields:

$$\dot{r} = v \tag{2.4a}$$
$$m\dot{v} = mg\vec{a}_3 + R(\eta)F + F_{ext} \tag{2.4b}$$
$$\dot{R}(\eta) = R(\eta)\Omega_\times \tag{2.4c}$$
$$I\dot{\Omega} = -\Omega \times I\Omega + \tau + \tau_{ext} \tag{2.4d}$$

where $\Omega_\times$ denotes the skew-symmetric matrix, such that $\Omega_\times v = \Omega \times v$ for

the vector cross product $\times$ and any vector $v \in \mathbb{R}^3$. $F_{ext} \in \{A\}$ and $\tau_{ext} \in \mathbb{R}^3$ correspond to the aerodynamic perturbations and external forces and moments acting on the quadrotor. In the free flight, these quantities are zero because there are any external agents acting on the quadrotor. On the other hand, the external force and moment become different from zero usually when the quadrotor interacts whit the environment. For example, when the quadrotor is carrying a load or is pushing an object, the external agents could produce forces or moments acting on the vehicle.

2.6 Control

As said in the previous section, we can assume that the control inputs of the quadrotor are the total thrust f and the total moment τ, $u := (f, \tau_x, \tau_y, \tau_z) \in \mathbb{R}^4$, and the state is given by $(r, v, R(\eta), \Omega)$. In this assumption, given a smooth trajectory $(R^*(t), r^*(t))$, the problem of control consists to find a smooth static state feedback u, depending on the measurable state, in order to track the desired trajectory, in other words, such that the tracking error $(r(t) - r^*(t), R(t) - R^*(t))$ is asymptotically stable.

This problem is not so trivial for several reasons. The first is that the quadrotor is an underactuated vehicle, indeed it has four inputs but it moves in SE(3) space that is six-dimensional. Then the derived model is only an approximation of the real dynamics of the quadrotor, for example it doesn't take amount of the drag effect and the effects produced by the propellers, such as turbulences and ground effects. Finally, the inputs are themselves idealized. In practical we should consider the dynamics of the motors and their interaction with the drag forces on the propellers.

The general approach to control the vehicle consists in a hierarchical structure ([44] [46] [50] [49] [54]) composed by three inner-outer loops.

- The higher level, that is the slowest outer-loop controller, is the *position tracking controller* for control the translational dynamics (2.4a) (2.4b). It allows to drive the position $r(t)$ to track the desired trajectory $r^*(t)$, giving the commands of thrust and attitude to the following level;

- The next level is the *attitude controller*. This is a faster inner-loop controller designed for the attitude dynamics (2.4c) (2.4d), in order to attain the desired attitude given from the outer-loop. This controller computes the desired torques to send to the next control level;

- Finally, the lowest level, with the highest bandwidth, is the control of the speed of the motors, in order to perform the desired torques and the total thrust given by the previous controller.

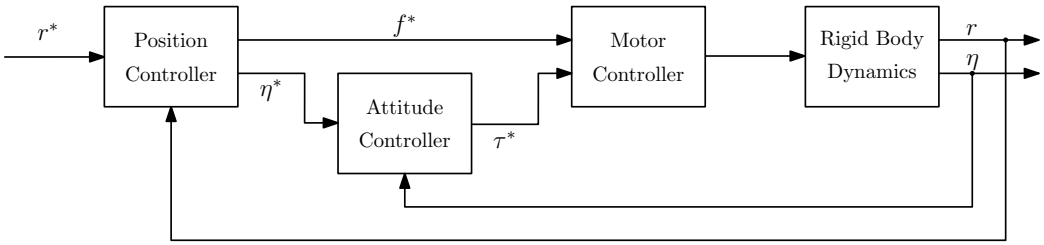


Figure 2.8: Hierarchical control structure: **a** - the innermost *motor control loop*, **b** -the intermediate *attitude control loop*, **c** - the outer *position control loop*.

This inner-outer loop structure, specially for the first two loops, is possible thanks to the facts that the attitude dynamics is independent to the translational dynamics, as we can see in the equations (2.4), whereas the contrary doesn't hold. Indeed, the equations (2.4a) (2.4b) depend to (2.4c) (2.4d) due to the term $R(\eta)$. In particular, the translational dynamics depends only on the angles ϕ, ϑ and the thrust amount f. As we mentioned in Sec. 2.3, the quadrotor is an underactuated vehicle that must tilt to have translational motions. Since the planar motion depends only to the roll and pitch angles, the yaw of the airframe is independent and arbitrary, so it must be controlled separately.

2.6.1 Motors Control

This hight frequency controller is responsible to attain the desired total thrust f, denoted by u_1, and the torques (τ_x, τ_y, τ_z), denoted by u_2, controlling the rotor speed of the motors according to the equations (2.2) (2.3). Since the

rotor speed drives the dynamic model of the vehicle, an high-quality control of the motor speed is fundamentally important for overall control of the vehicle.

Generally, the majority of the quadrotors mount brushless dc motors with an high-frequency pulsewidth modulation (PWM) to control motor voltage [55] [56]. Since the goal of this thesis is not to fully investigate the controller of the quadrotor, we don't go in the details of this part of the controller, moreover, the quadrotor is often equipped with standard brushless drivers. We simply mention the common problem where the rotor speed, for a given PWM command setting, will decrease as the battery voltage reduces during flight. Furthermore, the performance of the motor controller is limited by the current that can be supplied by the battery. This may be a significant limiting factor for smaller vehicles. Overly aggressive tuning and extreme maneuvers may cause the voltage bus to drop excessively, reducing the thrust for other motors and, in extreme cases, causing the onboard electronics to brownout. For this reason, it is common to introduce a saturation, although we loose the linearity of motor/rotor response during aggressive maneuvers.

2.6.2 Attitude Control

As we saw in Sec. 2.3, in order to obtain the lateral motion, the quadrotors has to tilt in the desired direction. For this purpose an attitude control is necessary. It is the heart of the overall control system, attaining the desired orientation, expressible by a rotation matrix or, equivalent, by the desired angles roll, pitch and yaw. So, given the desired airframe attitude R^*, we first denote the rotation error by the measure:

$$e_{R_x} = \frac{1}{2}((R^*)^T R^* - R^T R) \tag{2.5}$$

which is a skew-symmetric matrix representing the axis of rotation required to go from R to R^*, and whose magnitude is equal to the sine of the angle of rotation.

Now, in order to obtain a linear controller, we linearize the rotational dynamics about the nominal hover position, i.e. in a neighbourhood of the

angles $(\phi, \vartheta) = (0, 0)$ with angular velocity close to zero. If we write the rotation matrix $R = {}^{A}R_{B}$ as a product of the yaw rotation ${}^{A}R_{E}(\psi)$ and ${}^{E}R_{B}(\phi, \vartheta)$, we can linearize the rotation about $(\psi, \phi, \vartheta) = (\psi_0, 0, 0)$, where $\{E\}$ is the frame after the yaw rotation.

$$\begin{aligned} {}^{A}R_{B} &= {}^{A}R_{E}(\psi_0 + \Delta\psi)\,{}^{E}R_{B}(\Delta\phi, \Delta\vartheta) \\ &= \begin{bmatrix} c\psi & -s\psi & \Delta\vartheta c\psi + \Delta\phi s\psi \\ s\psi & c\psi & \Delta\vartheta s\psi - \Delta\phi c\psi \\ -\Delta\vartheta & \Delta\phi & 1 \end{bmatrix} \end{aligned}$$

where $\psi = \psi_0 + \Delta\psi$. If $R^* = {}^{A}R_{B}(\psi_0 + \Delta\psi, \Delta\phi, \Delta\vartheta)$ and $R = {}^{A}R_{B}(\psi_0, 0, 0)$, the equation (2.5) gives

$$e_{R_x} = \begin{bmatrix} 0 & \Delta\psi & -\Delta\vartheta \\ -\Delta\psi & 0 & \Delta\phi \\ \Delta\vartheta & -\Delta\phi & 0 \end{bmatrix} \tag{2.6}$$

which corresponds to the error vector

$$e_R = (\Delta\phi, \Delta\vartheta, \Delta\psi)^T$$

which simply corresponds to the errors between the desired roll, pitch, yaw angles and the actual. In order to attain the desired angles and to let the attitude error zero, if the desired angular velocity is zero, we can compute the desired torques with a PD controller:

$$u_2 = -k_P e_R - k_D e_\Omega \tag{2.7}$$

where the gains $k_P, k_D \in \mathbb{R}^{3\times3}$ are positive definite matrices. This controller guarantees stability for small deviations from the hover position. Looking at the (2.7), we can implement it with a series of two P controller (Fig. 2.9). The first, from the attitude error compute the desired angular velocity which

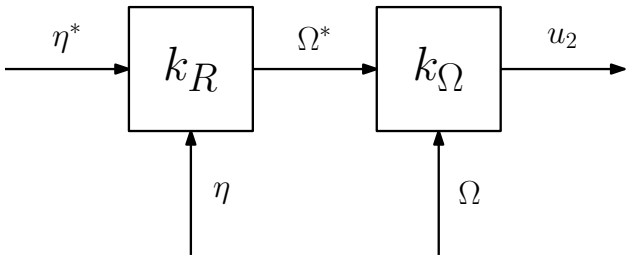


Figure 2.9: Attitude control structure: the first P control compute the desired angular velocity in order to attain null attitude error, then, the second, compute the desired torques to attain null angular velocity error.

is used by the second controller to compute the desired torques:

$$\begin{aligned} \Omega^* &= -k_R e_R \\ u_2 &= -k_\Omega e_\Omega \end{aligned}$$

2.6.3 Position Control

The outer controller deals to track a desired trajectory $r^*(t)$. To derive the equations of the controller, we can first write the rotation matrix as sequence of rotation along the body-frame axis as $R(\eta) = R_z(\psi)R_x(\phi)R_y(\vartheta)$ ((2.1)). Now, assuming any aerodynamic perturbation and external forces, $F_{ext} = 0$, the equation (2.4b) becomes

$$\begin{aligned} m\dot{v} &= mg\vec{a}_3 - fR_z(\psi)R_x(\phi)R_y(\vartheta)e_3 \\ R_z(\psi)m\dot{v} &= mg\vec{a}_3 - fR_x(\phi)R_y(\vartheta)e_3 \qquad (2.8a) \\ \begin{bmatrix} c\psi & -s\psi & 0 \\ s\psi & c\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} m\dot{v} &= \begin{bmatrix} 0 \\ 0 \\ mg \end{bmatrix} - f \begin{bmatrix} c\vartheta & 0 & s\vartheta \\ s\phi s\vartheta & c\phi & -s\phi c\vartheta \\ -c\phi s\vartheta & s\phi & c\phi c\vartheta \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \qquad (2.8b) \end{aligned}$$

whose last row is

$$m\dot{v}_3 = mg - f\cos(\phi)\cos(\vartheta)$$

From the above equation we can derive the control low for the total thrust $f = u_1$

$$u_1 = -\frac{m}{\cos(\phi)\cos\vartheta}\left[-g + \ddot{r}_3^* + k_{d,z}(\dot{r}_3^* - \dot{r}_3) + k_{p,z}(r_3^* - r_3)\right] \tag{2.9}$$

where $k_{d,z}, k_{p,z} \in \mathbb{R}$ are the positive gains of a standard PD controller, which ensures local exponential stability of $(r_3^* - r_3)$, i.e. steady state error equal to zero, as long as the system is away from the singularity $\cos(\phi)\cos(\vartheta) = 0$, which means $\phi \neq \pm\pi/2$ and $\vartheta \neq \pm\pi/2$. Physically this implies that the axis $\vec{b_3}$ of the body frame can't be horizontal.

The first two row of (2.4b) are given by

$$\begin{aligned} m\begin{bmatrix}\dot{v}_x\\ \dot{v}_y\end{bmatrix} &= -f\begin{bmatrix}c\psi s\vartheta + c\vartheta s\phi s\psi\\ s\psi s\vartheta - c\psi c\vartheta s\phi\\ c\phi c\vartheta\end{bmatrix} = -f\begin{bmatrix}c\psi & c\vartheta s\psi\\ s\psi & -c\psi c\vartheta\end{bmatrix}\begin{bmatrix}s\vartheta\\ s\psi\end{bmatrix} \\ &= -fQ(\phi,\vartheta)\begin{bmatrix}s\vartheta\\ s\psi\end{bmatrix} \end{aligned} \tag{2.10}$$

where we defined $Q(\phi,\vartheta) := \begin{bmatrix}c\psi & c\vartheta s\psi\\ s\psi & -c\psi c\vartheta\end{bmatrix}$, which is a square matrix with determinant $\det(Q) = \cos(\vartheta)\left(\cos^2(\psi) + \sin^2(\psi)\right) = \cos(\vartheta)$. So it is analytically invertible as long as $\cos(\vartheta) \neq 0$, that means $\vartheta \neq \pi/2$, condition that we already found for the (2.9).

This shows that the planar position error $(r_x^* - r_x, r_y^* - r_y)$ will be locally exponentially stable, if the attitude controller can attain the desired attitude, or in other world the desired pitch and roll commands (ϑ^*, ϕ^*), given by inverting the (2.10) and using a PD control

$$\begin{bmatrix}\sin\vartheta^*\\ \sin\phi^*\end{bmatrix} = \frac{mQ^{-1}}{-f}\begin{bmatrix}\ddot{r}_x^* + k_{d,x}(\dot{r}_x^* - \dot{r}_x) + k_{p,x}(r_x^* - r_x)\\ \ddot{r}_y^* + k_{d,y}(\dot{r}_y^* - \dot{r}_y) + k_{p,y}(r_y^* - r_y)\end{bmatrix} \tag{2.11}$$

Summarizing, the position controller, represented in the Fig. 2.10, given a desired trajectory, computes the desired total thrust, f^*, and the desired attitude angles $\eta^* := [\phi^*, \vartheta^*, \psi^*]$, where (ϕ^*, ϑ^*) are given above from the

position controller (2.11), while ψ^* can be set arbitrarily (e.g., $\psi^* = 0$).

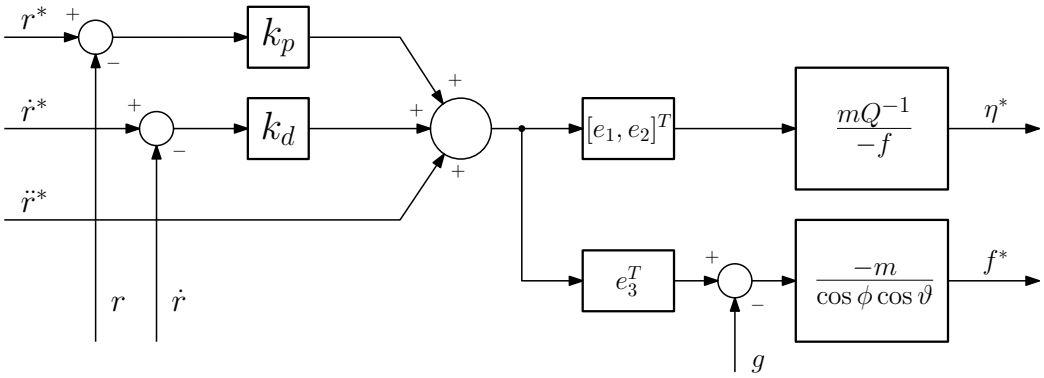


Figure 2.10: Position control structure: the first part compute the proportional and derivative corrective action, the second part represents the equations (2.9) (2.11).

2.7 Estimating the Vehicle State

In order to implement the control strategy showed before, the estimation of the vehicle state is necessary. In particular, the key state estimates are the position, attitude, angular velocity, and linear velocity. Since the inner control loop is the attitude control, ignoring the motor control, the primary and most important variables used to control the vehicle are the attitude and the angular velocity, that are also needed at an higher frequency. Then, for the position control, the other state variables, position and linear velocity, are needed, but at slower frequency.

For the estimation of the attitude and angular velocity, the basic instrument carried by any quadrotor is the inertial measurement unit (IMU), which includes an accelerometer and a gyroscope. Often, this basic sensors are combined with a magnetometer and some form of heigh measurement, either acoustic, infrared, barometric, or laser based. For a complete state estimation of the vehicle this is not sufficient, and more sophisticated sensors are necessary to perform the estimation of the position and linear velocity. Therefore, for many robotics applications other sensors are used, such as motion capture system (VICON), global positioning system (GPS) or cameras.

2.7.1 Attitude Estimation

As we just noticed in the Sec. 2.6, the estimation of the attitude and the angular velocity of the airframe, is a crucial problem in controlling a quadrotor since it is used for the inner control loop which stabilize the attitude. Moreover, since with the improving of the technology the quadrotors are becoming smaller, their small size and maneuverability can also create problems. Indeed, their fast dynamics requires accurate and fast orientation and velocity state estimation in order to drive controllers to stabilize the flight. Thus, precise and timely estimation of attitude and angular velocity are the key ingredients in the quadrotor control.

In practice, a single sensor is not sufficient to measure the attitude of the vehicle with respect to the inertial frame. Often it is necessary a set of sensors which produce a set of data that is then fused to obtain a robust estimate of the orientation of the vehicle. Although there are a variety of sensors which are able to measure the orientation, actually, inertial based sensory system, commonly called IMU (*Inertial Measurement Unit*), are the most used for quadrotors and UAVs in general. This is due to the fact that they are low cost, tiny, slight and completely self contained in a single unit in the electronics module. An IMU is in general composed of 3-axis rate gyroscopes and 3-axis accelerometers which are able to track the rotational and translational movements. In particular, the rate gyro measures the rotational velocity, whereas, the accelerometers, measures the non-gravitational linear acceleration of the system. The gyro measures are modelled by

$$\bar{\Omega} = \Omega + b + \mu \tag{2.12}$$

where Ω is the true value, $b := b(t)$ denotes a deterministic gyro bias vector that is slowly varying with time, and μ is a Gaussian noise process. The accelerometers measures the non-gravitational linear acceleration of the system in the body fixed frame $\{B\}$.

$$a = -R^T(g_0 - \dot{v}) \tag{2.13}$$

where $\dot{v}$ is the acceleration of the body fixed frame with respect the inertial frame, and $g_0 = |g_0|\, e_3$ denotes the gravity force vector, where $|g_0| \approx 9.81$ is the gravitational constant.

Since the gyroscopes measures directly the rotational velocity, it could be possible to compute the angles of rotation through integration of the sensor signal. Although this integration process is really simple, actually, the presence of the bias error b, even small, will lead to divergence of the angle. For this reason, many estimation techniques are implemented to fuse the two sensors obtaining a fast an stable measure of the attitude. Some methods that are commonly employed are based on the traditional linear Kalman filter [57], then implemented in various way, such as the *Extended Kalman Filter* (EKF) [58] [59]. Then there are more sophisticated stochastic filtering techniques such as particle filter and unscented filters, though they require too much computational load on the low cost microcontroller used in the majority of commercial vehicles. Another popular filtering techniques are based on *linear complementary filter* with frequency domain analysis [60]. At the end, there are also some implementation of the *nonlinear complementary filter* that combines accelerometer output for low frequency attitude estimation with integrated gyroscope output for high frequency estimation [61] [62]. In particular, the *explicit complementary filter* (ECF) uses the measured angular velocities $\bar{\Omega}$ and a measurement of an inertial direction denoted by $\bar{v}$. Using the data from the accelerometer we can derive the inertial direction $\bar{v}$ from the best estimation of the gravitation direction g obtained from the system, $\hat{g}^B$ in the body fixed frame

$$\bar{v} = \frac{\hat{g}^B}{|\hat{g}^B|} \tag{2.14}$$

Normally, in the (2.13), the term g_0 dominates the value of a for a sufficiently low frequency response ($\dot{v} \approx 0$) . So $a \approx -R^T g_0$ is a reasonable low frequency estimate of the gravity in the body fixed frame ($\hat{g}^B = a$). Thus the (2.14) become

$$\bar{v} = \frac{a}{|a|} \approx -R^T e_3 \tag{2.15}$$

The ECF is simply an integration of the angular velocity, correcting the drifting with the low frequency estimation of the gravity direction. In the quaternion form [63], it can be expressed as an observer

$$\dot{\hat{q}} = \frac{1}{2}\hat{q} \otimes \mathbf{p}(\bar{\Omega} + \delta) \tag{2.16a}$$

$$\delta = k_P e + k_I \int e \tag{2.16b}$$

$$e = \bar{v} \times \hat{v} \tag{2.16c}$$

where $\hat{q}$ is an estimate of the system attitude expressed as a unit quaternion; δ is an innovation of the filter equation generated by a proportional-integral feedback, and the error e is the relative rotation between the measured inertial direction $\bar{v}$ and the predicted direction $\hat{v}$. The gains $k_P, k_I \in \mathbb{R}$ are positive gains, while $\mathbf{p}(\cdot)$ is the pure quaternion operator, $\mathbf{p}(\Omega) = (0, \Omega)$.

The estimate $\hat{v}$ is the ECF's best estimate of the gravitation direction, which is the Z-axis of the inertial frame. So, given the quaternion estimate $\hat{q}$, we have

$$\hat{v} = \begin{bmatrix} 2(\hat{q}_1\hat{q}_3 + \hat{q}_0\hat{q}_2) \\ 2(\hat{q}_2\hat{q}_3 + \hat{q}_0\hat{q}_1) \\ \hat{q}_0^2 - \hat{q}_1^2 - \hat{q}_2^2 + \hat{q}_3^2 \end{bmatrix} \tag{2.17}$$

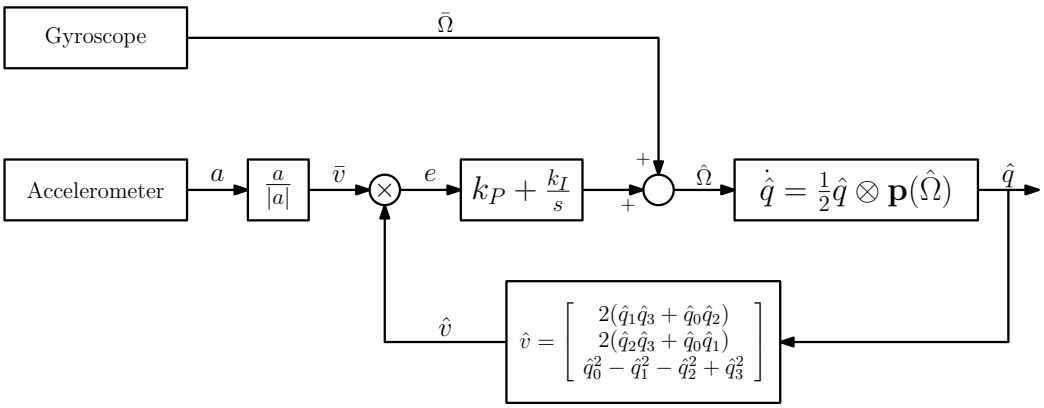


Figure 2.11: Attitude estimation: Explicit Complementary Filter based on accelerometer and gyroscope signals.

In Fig. 2.11 is represented the block diagram of the filter. In the PI com-

pensation, the proportional term governs the frequency cross-over between the accelerometer based attitude estimates and integrated gyro estimates. While, the integral term corrects for gyro bias.

2.7.2 Position Estimation

In this part we want to briefly describe the major techniques utilized to solve the central problem of estimating the position of the vehicle. As we just said, the IMU measures are not sufficient to this purpose, and other more technological sensors are needed.

At first we can split the problem in two: estimation of the planar position and the height. The later is characterized by two different measure of height, the absolute height of the vehicle and the relative height over the terrain at a given time. Often quadrotors are equipped whit a barometer which gives the absolute height with a precision of a few centimeters. Moreover, absolute height can also be measured using GPS, VICON, or a full SLAM (Simultaneous Localization and Mapping) system. The relative height, important to avoid obstacles, can be estimated using acoustic sensor such as ultrasonic, infrared or laser-ranging.

Also for the planar position we can measure an absolute position or a relative position. Absolute position could be determined using GPS system (few-centimeter accuracy at up to 10 Hz), but it doesn't work indoors. Another solution is to use an external motion capture systems such as VICON (50μm accuracy at 375Hz), thought it is expansive and their sensor array has a limited spatial extent that is impractical for large indoor environments.

On the other hand, relative position can be estimated measuring the distance to the objects, using onboard sensors, typically small onboard laser range finders (LRFs) or RGBD camera system such as the Kinect. The vision techniques are becoming very popular since the sensors are small, lightweight, and low power. Moreover, the vision algorithms can provide essential navigational competences such as odometry, attitude estimation, mapping, place and object recognition, and collision detection. However, since the algorithms are computationally intense and on board the computational power

is limited, is difficult to process the data completely onboard. Thus, often the system transmits the images to a ground station by wireless, which increases system complexity, control latency with the consequent risk of instability. Moreover, the underactuated quadrotor has to tilt to point the thrust vector in the direction of the desired translational motion. For a camera rigidly fixed on the vehicle, this attitude control motion induces a large apparent motion in the image. To solve the problem and eliminate this effect it is necessary to estimate the attitude at the instant the images was captured.

Chapter 3

Dynamic Model

Introduction

In this chapter we will derive the equation of the system composed of two quadrotors tethered together by one rope, and to one fixed point, called anchor, by two more ropes. At first we will examine the rope, finding two types of model. The first one very easy used for the model of the whole system. While the second more sophisticate, which describes better the elastic behavior, was used for the simulation part. Afterwards we will find the equation of the whole system with respect to a single generalized coordinate, represented by the formation orientation, using the Lagrange formalism.

3.1 Rope Model

Notations	
$(q_1, q_2) \in \mathbb{R}^3$	edges position
$l \in \mathbb{R}$	length $l := \|q_2 - q_1\|$
$l_0 \in \mathbb{R}$	unstressed length
$L \in \mathbb{R}^3$	the normalized axis $L := (q_2 - q_1)/l$
$T_1, T_2 \in \mathbb{R}^3$	internal tensions $T_1 := -tL$ and $T_2 := -T1$
$t \in \mathbb{R}$	intensity of the tension
$F_i \in \mathbb{R}^3$	external force applied at the edge q_i
$k \in \mathbb{R}$	stiffness coefficient for a spring
$b \in \mathbb{R}$	damping coefficient for a damper

A rope consist of one or more threads which are twisted together in order to combine them into a larger and stronger form. Its basic characteristics are tensile strength, but not compressive strength. Thus it is used for dragging and lifting but not for pushing or similar compressive application.

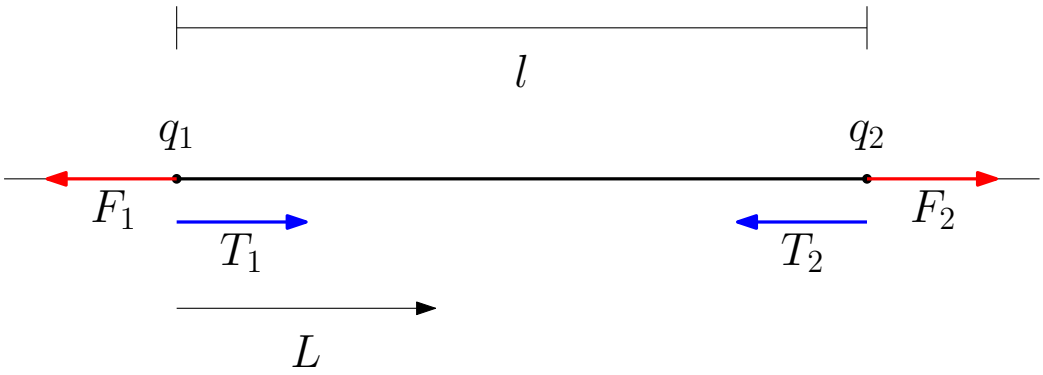


Figure 3.1: Schematic of a taut rope.

With respect to Fig. 3.1, the main describing variables of a rope are: the position of the edges $q_1, q_2 \in \mathbb{R}^3$, the length of the rope $l := \|q_2 - q_1\| \in \mathbb{R}$, the unstressed length $l_0 \in \mathbb{R}$, the normalized axis of the cable $L := (q_2 - q_1)/l \in \mathbb{R}^3$ and the internal tensions $T_1 := tL \in \mathbb{R}^3$ and $T_2 := -T_1$ where $t \in \mathbb{R}$ is the intensity of the tension. Basically the cable could be described as a

hybrid system described by two states: taut and not taut. The rope is taut when its length is equal o grater than the unstressed length and the internal tension is greater than zero, otherwise it is slack. The rules which describe the behavior of a taut rope are:

- The tension strength is always directed along the rope's axis L;
- The tension strengths on the ends of the tether are opposed, $T_2 := -T_1$;
- The tension strength is equal to the sum of the forces applied at the end and projected on the rope axis $T + \sum F' = 0$, where $'$ indicate the projection of the vector along the cable's axis. If there is an object of mass m on that end, the tension strength results $T + \sum F = ma$.

3.1.1 Basic Model

The basic model that we can use for a rope simply says that, if the length is less than the unstressed, then the cable is slack, the tension is zero and the two edges are independent. Otherwise, when the length is equal or grater than the unstressed and the external forces are stretching the cable, the latter is taut and it behaves as a rigid stick. The hybrid model representing the two states Σ_t, Σ_s, respectively cable taut and slack, is explicitly described by the following equations, where the operator $^{\perp} : \mathbb{R}^3 \times \mathbb{R}^3 \to \mathbb{R} : c \times L \to c^{\perp}$, for any vector $c \in \mathbb{R}^3$ corresponds to the projection of c into the cable's axis L, $c^{\perp} := c^T \cdot L$, and $F_1, F_2 \in \mathbb{R}^3$ are the sum of the external forces acting on the edges of the cable.

$$\Sigma_t \quad : \quad \begin{cases} F_1 + T_1 = m_1 \ddot{q}_1 \\ F_2 + T_2 = m_2 \ddot{q}_2 \\ T_1 = -T_2 \\ \ddot{q}_1^{\perp} = \ddot{q}_2^{\perp} \end{cases} \qquad \text{if } l \geq l_0 \text{ and } t > 0 \tag{3.1a}$$

$$\Sigma_s \quad : \quad \begin{cases} F_1 = m_1 \ddot{q}_1 \\ F_2 = m_2 \ddot{q}_2 \\ T_1 = -T_2 = 0 \end{cases} \qquad \text{if } l < l_0 \tag{3.1b}$$

Moreover, when the cable goes from being slack to taut, i.e. from Σ_s to Σ_t, there is a discrete change in the velocity of the system, and this is modeled based on a perfectly inelastic collision, that ensures $\dot{q}_1 = \dot{q}_2$.

An equivalent model for the rope, which avoids the assumption of inelastic collision, is a spring with an infinite stiffness $k \in \mathbb{R}$, which produces a force if and only if its length is grater than the unstressed one (Fig. 3.2). The results is a simple undamped harmonic oscillator (HO).

$$t = \begin{cases} -k(l - l_0), & \text{if } |l| > l_0 \\ 0 & \text{otherwise} \end{cases}$$

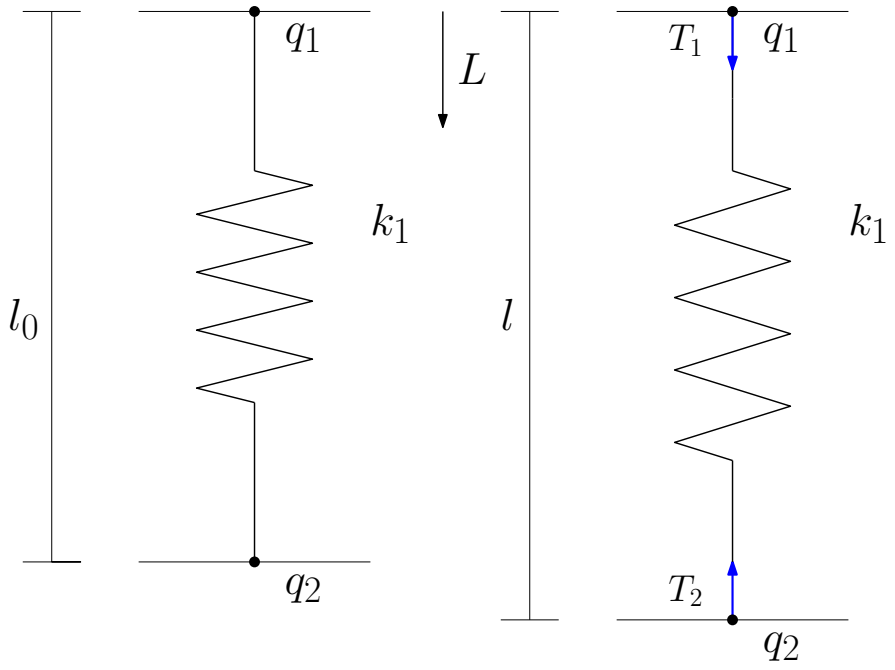


Figure 3.2: Rope's model as a simple spring: On the left unstressed rope, on the right stretched rope.

3.1.2 Standard Linear Solid Model

Although the most popular model is the previous, actually, it doesn't describe the elongation property of a rope. The HO model doesn't reproduce the

response delay due to the relatively slow microscopic deformation process acting in the rope when some external forces are stretching it. Indeed, during the stretch, there is an energy dissipation due to the transformation of the mechanical energy into thermal energy. This is the typical characteristic of a viscoelastic material such as the rope.

On the other hand, a model that describes better this viscoelastic behavior is the *Standard Linear Solid model* (SLS) [64] [65] [66]. The cable dynamic behavior during the stretching, when the rope goes from being slack to taut, could be divided in two phases. In the initial stretch the rope behaves like an undamped oscillator, i.e. as a single spring; while, at the end, before the first maximum elongation is reached, a strong dumping occurs. The SLS model, which describes very well this dynamics, consist of a spring which model the first stretch behavior , with in series a damped harmonic oscillator composed of a damper and a spring in parallel which describes the inner friction (Fig. 3.3).

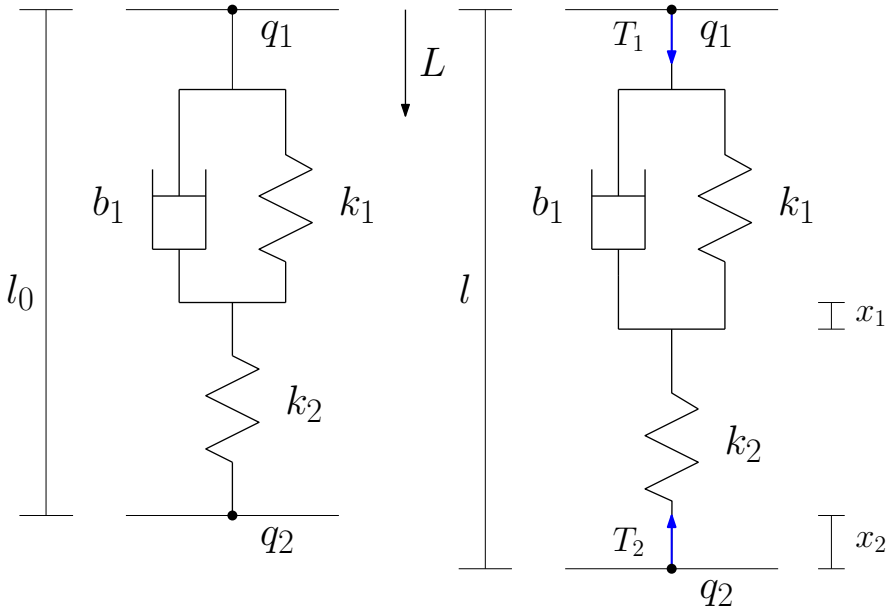


Figure 3.3: Rope's model as a SLS model: On the left unstressed rope, on the right stretched rope.

We defined $x_1 \in \mathbb{R}$ the elongation of damper harmonic oscillator, that is an unobservable variable, and $x_2 \in \mathbb{R}$ the elongation of the whole rope.

Therefore, the instantaneous length $l(t)$ of the rope is given by:

$$l = \|q_2 - q_1\| = l_0 + x_2 \tag{3.2}$$

Then, using the same definitions in the previous sections, and defining $b_1 \in \mathbb{R}$ the viscosity coefficient, $k_1, k_2 \in \mathbb{R}$ the elastic coefficients, when the rope is taut ($x_2 \geq 0$), the forces acting on the system are:

$$\begin{aligned} F^{b_1} &= -b_1 \dot{x}_1 \cdot L \\ F^{k_1} &= -k_1 x_1 \cdot L \\ F^{k_2} &= k_2(x_2 - x_1) \cdot L \\ T_2 &= -F^{k_2} \end{aligned} \tag{3.3}$$

The equilibrium of the inner forces ($F^{b_1} + F^{k_1} + F^{k_2} = 0$) and the derivative of (3.2) yields the following differential equations:

$$\begin{aligned} b_1 \dot{x}_1 &= -k_1 x_1 + k_2(x_2 - x_1) \\ \dot{x}_2 &= \dot{l} \end{aligned} \tag{3.4}$$

Then we can derive the dynamic equations of the cable in the state space formalism, considering as state the two variables (x_1, x_2), as output the tension on the cable t and as input the difference of the two ends velocity ($\dot{l} = \dot{q_2} - \dot{q_1}$).

$$\begin{cases} \begin{bmatrix} \dot{x_1} \\ \dot{x_2} \end{bmatrix} = \begin{bmatrix} -\frac{k_1+k_2}{b_1} & \frac{k_2}{b_1} \\ 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} \dot{l} \\ t = \begin{bmatrix} k_2 & -k_2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \qquad \text{if } x_2 \geq 0 \\ t = 0 \qquad \text{if } x_2 < 0 \end{cases} \tag{3.5}$$

Looking at the eigenvalues of the system, $\lambda_1 = 0$ and $\lambda_2 = -\frac{k_1+k_2}{b_1}$, we can notice that the system is always stable and the tether tension behaves

as an integrator of the tether elongation velocity.

Regarding the model parameters b_1, k_1, k_2, they are usually expressed as:

$$\begin{aligned} q &= \pi \left(\frac{d_c}{2}\right)^2 \\ k_1 &= \frac{qE_1}{l_0} \\ k_2 &= \frac{qE_2}{l_0} \\ b_1 &= \frac{q\eta}{l_0} \end{aligned} \tag{3.6}$$

where $d_c \in \mathbb{R}$ is the diameter of the rope, $E_1, E_2 \in \mathbb{R}$ are the elastic modulus and $\eta \in \mathbb{R}$ is the viscosity, which depend on the types of the rope.

3.2 System model

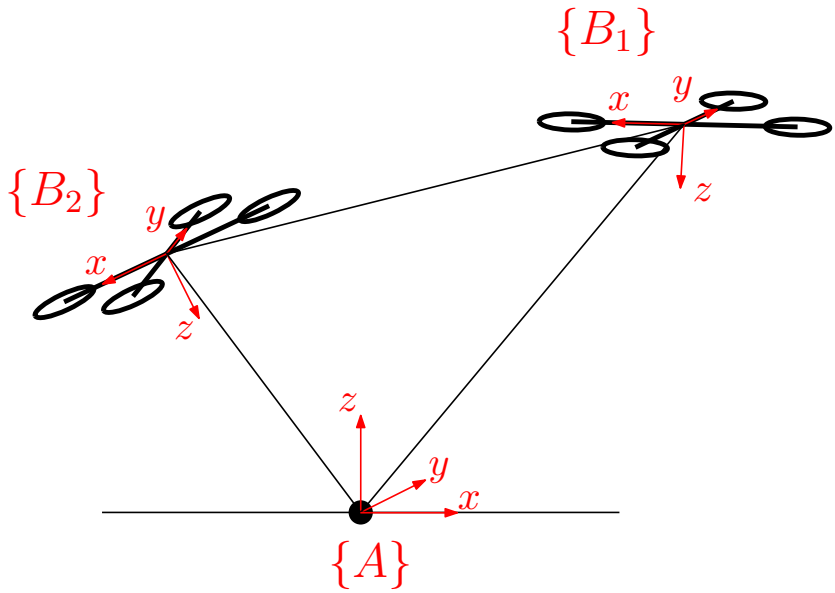


Figure 3.4: Reference frames of the system. $\{A\}$ is the inertial global frame, centered in the anchor point. $\{B_i\}$ is the fixed body frame of the i-th quadrotor.

The system consists of two quadrotors, Q_1, Q_2, attached to an anchored

point $O \in \mathbb{R}^3$ by two ropes, and to each others by a third cable. At first we need to define the reference frames of the system. We define the inertial frame $\{A\}$ centered in the anchor point and with the z-axis directed upwards. Then we define the body frame of the two quadrotors $\{B_1\}, \{B_2\}$, centered in the center of gravity, with the z-axis directed downwards and the x-axis directed to the left (Fig. 3.4).

For simplicity, we consider the system restricted in a two dimensional space. In practice we constraint the robots to move only in a vertical plane passing for the anchor point. Defining $q_i \in \mathbb{R}^3$ the position of the i-quadrotor, we impose $q_{iy}(t) = 0 \; \forall t \in \mathbb{R}, i = 1, 2$. The system is clearly described by the Fig. 3.5, where the forces acting on the system and its variables are showed. We define $q_1, q_2 \in \mathbb{R}^2$ the position of the two quadrotors in the planar space, $L_i \in \mathbb{R}^2$ is the axis of the i-th cable, in particular we have $L_1 = q_1 - O$, $L_2 = q_2 - O$, $L_3 = q_1 - q_2$, $T_i = -t_i L_i \in \mathbb{R}^2$ is the internal tension in the i-th cable with intensity $t_i \in \mathbb{R}$ and $F_i \in \mathbb{R}^2$ is the thrust of the i-th quadrotor.

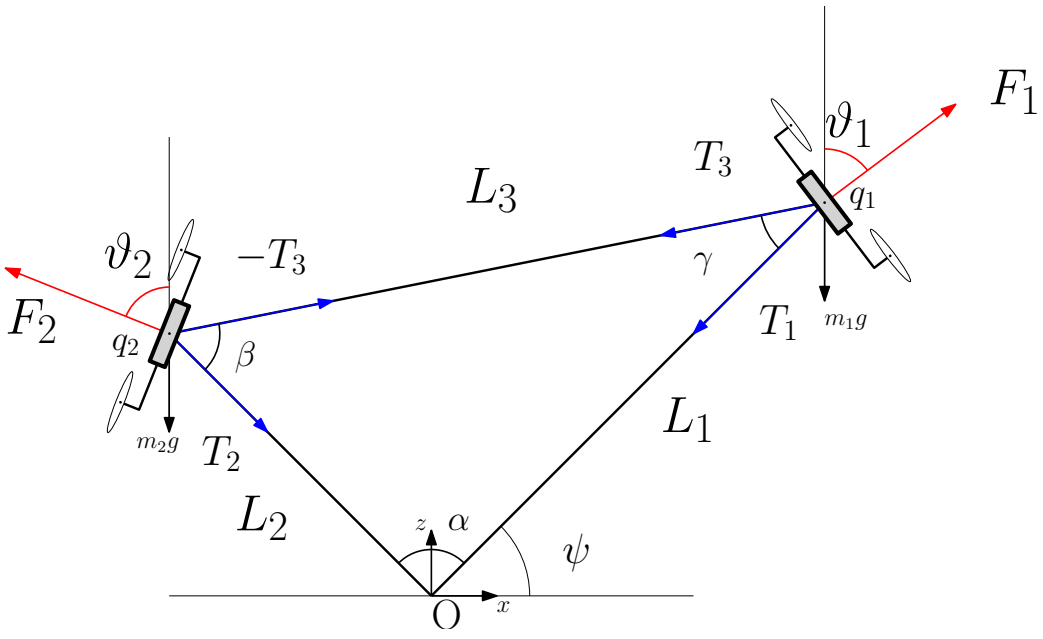


Figure 3.5: Schematic diagram of the system. The black lines represent the ropes, the fat black lines represent the quadrotors craft, the red arrows represent the total thrust generated by the vehicles and the blue arrows represent the internal tensions.

We assume that the vehicles have mass $m_1, m_2 \in \mathbb{R}$, whereas the tethers have a negligible mass, are straight, always taut and of fixed length $l_1, l_2, l_3 \in \mathbb{R}$. Moreover the ropes are attached to the center of mass of the vehicles. In this way there aren't external torques applied to the quadrotor. In this assumption we can model the quadrotors as a mass and a orientable force on the end of the tethers, described by the two variables: $f_i \in \mathbb{R}$ that is the intensity of the total thrust, and $\vartheta_i \in \mathbb{R}$ that is the orientation of thrust, defined positive clockwise. In other words, considering the body fixed frame $\{B_i\}$, ϑ_i is the pitch angle of the i-th quadrotor. Additionally, since we assume taut cables whit a fixed length, the triangle is a rigid figure where the internal angles are constant and defined by the length of the ropes. In particular, given l_1, l_2 and α, the other variables are directly determined by the triangle's laws:

$$\begin{aligned} l_3 &= \sqrt{l_2^2 + l_1^2 - 2l_1 l_2 \cos\alpha} \\ \beta &= \sin^{-1}\left(\frac{l_1}{l_3}\sin\alpha\right) \\ \gamma &= \pi - (\alpha + \beta) \end{aligned}$$

With this assumption, the orientation of the formation is described by a single variable defined as the angle, $\psi \in \mathbb{R}$, between the first cable and the x-axis of the inertial frame, measured positive counter-clockwise. Moreover we can notice that, if the tethers are taut, the two quadrotor can only turn around the anchor O, with the same angular velocity $\dot{\psi}$. Now we can derive the equation of the dynamic model of the system, in particular of orientation angle ψ as function of the thrust forces applied by the two quadrotors. To do this we used the Lagrangian formalism [67] [68] [69].

Kinematic Energy

Since the agents are attached to each other, they must have the same velocity. Moreover they are attached to the same anchor so they can move only of circular motion. This means that they have the same angular velocity $\dot{\psi}$.

Thus, their kinematic energies are, respectively:

$$\begin{aligned} K_1 &= \frac{1}{2} m_1 \dot{v}_1^2 = \frac{1}{2} m_1 l_1^2 \dot{\psi}^2 \\ K_2 &= \frac{1}{2} m_1 \dot{v}_2^2 = \frac{1}{2} m_2 l_2^2 \dot{\psi}^2 \end{aligned} \tag{3.7}$$

Potential Energy

The only conservative force is the gravity, so the potential energy of the vehicles are:

$$\begin{aligned} V_1 &= m_1 g h_1 = m_1 g l_1 \sin\psi \\ V_2 &= m_2 g h_2 = m_2 g l_2 \sin(\psi + \alpha) \end{aligned} \tag{3.8}$$

Generalized Forces

The thrusts produced by the quadrotors are not conservative forces, and they are defined by:

$$F_1 = f_1 \begin{bmatrix} \sin\vartheta_1 \\ \cos\vartheta_1 \end{bmatrix} \qquad\qquad F_2 = f_2 \begin{bmatrix} \sin\vartheta_2 \\ \cos\vartheta_2 \end{bmatrix}$$

where f_i is the intensity of the thrust and ϑ_i is the tilt of the quadrotor that identify the direction of the force. Their application points, q_i, are also described by the vectors which represent the tethers:

$$q_1 = l_1 L_1 = l_1 \begin{bmatrix} \cos\psi \\ \sin\psi \end{bmatrix} \qquad\qquad q_2 = l_2 L_2 = l_2 \begin{bmatrix} \cos(\psi+\alpha) \\ \sin(\psi+\alpha) \end{bmatrix}$$

Now the quantities that replace the forces in the Lagrangian formalism, called *generalized forces*, are given by the relation $Q_\psi = \sum_{i=1}^{2} F_i \cdot \frac{\partial q_i}{\partial \psi}$, thus:

$$\frac{\partial q_1}{\partial \psi} = l_1 \begin{bmatrix} -\sin\psi \\ \cos\psi \end{bmatrix} \qquad\qquad \frac{\partial q_2}{\partial \psi} = l_2 \begin{bmatrix} -\sin(\psi+\alpha) \\ \cos(\psi+\alpha) \end{bmatrix}$$

$$Q_\psi = f_1 l_1 \cos(\psi + \vartheta_1) + f_2 l_2 \cos(\psi + \alpha + \vartheta_2) \tag{3.9}$$

In this case we can notice that Q_ψ is the sum of the external torques acting on the system. Indeed, the two terms in (3.9) are the vectorial product between the rope and the thrust, $f_1 l_1 \cos(\psi + \vartheta_1) = q_1 \times F_1$ and $f_2 l_2 \cos(\psi + \alpha + \vartheta_2) = q_2 \times F_2$.

Lagrangian Function

Now we can determine the *Lagrangian function*:

$$L = (K_1 + K_2) - (V_1 + V_2) = \frac{1}{2}(m_1 l_1^2 + m_2 l_2^2)\dot{\psi}^2 - g(m_1 l_1 \sin\psi + m_2 l_2 sin(\psi + \alpha)) \tag{3.10}$$

ad its derivate necessary to write the dynamic equations:

$$\begin{aligned} \frac{\partial L}{\partial \dot{\psi}} &= (m_1 l_1^2 + m_2 l_2^2)\dot{\psi} \\ \frac{d}{dt}\frac{\partial L}{\partial \dot{\psi}} &= (m_1 l_1^2 + m_2 l_2^2)\ddot{\psi} \\ \frac{\partial L}{\partial \psi} &= -g(m_1 l_1 \cos\psi + m_2 l_2 \cos(\psi + \alpha)) \end{aligned}$$

Lagrangian Equations

From the Lagrangian mechanics, the dynamic model is given by the common Lagrangian equation $\frac{d}{dt}\frac{\partial L}{\partial \dot{\psi}} - \frac{\partial L}{\partial \psi} = Q_\psi$:

$$(m_1 l_1^2 + m_2 l_2^2)\ddot{\psi} + g(m_1 l_1 \cos\psi + m_2 l_2 \cos(\psi + \alpha)) = f_1 l_1 \cos(\psi + \vartheta_1) + f_2 l_2 \cos(\psi + \alpha + \vartheta_2)$$

Then we can write the previous equation in the normal form, i.e. as a second order differential equation in the output variable ψ which describes it dynamics, depending on the thrust of the two quadrotors that are the inputs of the system.

$$\ddot{\psi} = \frac{-g(m_1 l_1 \cos\psi + m_2 l_2 \cos(\psi + \alpha)) + f_1 l_1 \cos(\psi + \vartheta_1) + f_2 l_2 \cos(\psi + \alpha + \vartheta_2)}{m_1 l_1^2 + m_2 l_2^2} \tag{3.11}$$

We recall another time that this model is true only when the tethers are taut, otherwise, the assumption that the two vehicle have the same angular

velocity is not valid. To simplify the writing we define:

$$\begin{aligned} I &:= m_1 l_1^2 + m_2 l_2^2 \\ g(\psi) &:= g(m_1 l_1 \cos\psi + m_2 l_2 \cos(\psi + \alpha)) \end{aligned}$$

Now the dynamic equation become

$$\ddot{\psi} = \frac{-g(\psi) + f_1 l_1 \cos(\psi + \vartheta_1) + f_2 l_2 \cos(\psi + \alpha + \vartheta_2)}{I} \tag{3.12}$$

Immediately we can notice that it is a strongly non linear model, and it is basically the second cardinal equation of the dynamics. If we think at $I = m_1 l_1^2 + m_2 l_2^2$ as the inertia of the system with respect to the point O, the (3.12) is the balance of torques.

3.3 Tethers Internal Tension

As we said in the Sec. 3.1, we can describe the tether as a hybrid model, characterized by two behaviors: *taut tether* and *slack tether*. We already said that the variable which describe the state of the tether is its tension force $T = -t \cdot L$. Now we want to obtain the geometric conditions for the external forces which ensure to keep the tethers taut. We start from the simplest case, one robot tethered with one cable, then we analyze the case of one robot tethered with two cables, and, at the end, the case of this project: two robots tethered with three cables. We can simplify the problem assuming the absence of the gravity (in other way we can take $\tilde{F} = F - mg \cdot e_2$).

3.3.1 One robot, one cable

Suppose to have a single cable fixed to a point and at the other edge is acting a force representing the thrust produced by a quadrotor ($f \geq 0$). In steady state condition ($\dot{q}_1 = \dot{q}_1 = 0$) and recalling the equation (3.1), the rope is

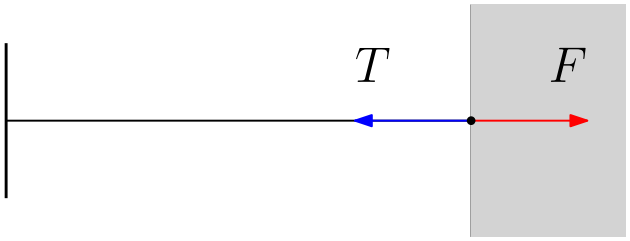


Figure 3.6: Tension on a taut tether with a fixed edge and on the other acting a force.

taut if:

$$t = F \cdot L = f \sin(\vartheta + \psi) \geq 0 \quad \Leftrightarrow \quad -\psi < \vartheta < \pi - \psi \quad \forall \quad f > \varepsilon$$

This shows that the cable is taut if the force (plus the gravity) is inside the half plane opposite and perpendicular to the cable's axis.

3.3.2 One robot, two cables

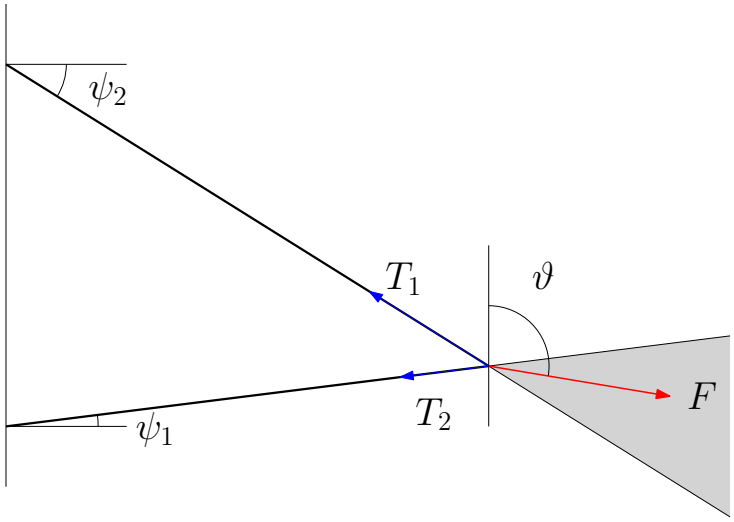


Figure 3.7: Tension on two taut tethers.

The case that we are going to analyze consist of one force (quadrotor) applied at the end of two fixed cables. Using the same definitions of before and the first equation of Newton on static condition, we derive the tensions

t_1, t_2 as function of the external force.

$$T_1 + T_2 + F = 0$$
$$\underbrace{\begin{bmatrix} L_1 & L_2 \end{bmatrix}}_{A} \begin{bmatrix} t_1 \\ t_2 \end{bmatrix} = F$$

where we defined $A = \begin{bmatrix} \cos(\psi_1) & \cos(\psi_2) \\ \sin(\psi_1) & \sin(\psi_2) \end{bmatrix}$ that is an invertible square matrix. The determinant is $\Delta = \sin(\psi_2 - \psi_1) \neq 0$, because $\psi_1 \neq \psi_2 + k\pi \ \forall \ k = 1, 2, \ldots,$ otherwise the two ropes would be parallel. Thus we can determine the two internal tensions t_1, t_2

$$\begin{bmatrix} t_1 \\ t_2 \end{bmatrix} = A^{-1}F = \frac{-f}{\sin(\psi_1 - \psi_2)} \begin{bmatrix} -\cos(\vartheta + \psi_2) \\ \cos(\vartheta + \psi_1) \end{bmatrix} \tag{3.13}$$

Since in our case $\psi_1 - \psi_2 \in (\pi/2, \pi)$ and $f > 0$, it is clear that:

$$\begin{bmatrix} t_1 \\ t_2 \end{bmatrix} > 0 \quad \Longleftrightarrow \quad \begin{cases} -\pi/2 - \psi_2 < \vartheta < \pi/2 - \psi_2 \\ +\pi/2 - \psi_1 < \vartheta < 3\pi/2 - \psi_1 \end{cases}$$

This shows that, to ensure taut ropes, the external force has to stay between the two tether's axis.

3.3.3 Two robots, three cables

Now we want to determine the equations of the tethers internal tension in the complete system with the two quadrotors tethered with the three cables. Recalling the previous definitions, we can use the first equation of Newton, in steady state condition condition ($\dot{\psi} = \ddot{\psi} = 0$), to derive the tensions t_1, t_2, t_3 as function of the external force. The balance of the forces on the center of

gravity of each agent results:

$$\begin{cases} T_3 + F_1 + T_1 = 0 \\ -T_3 + F_2 + T_2 = 0 \end{cases} \tag{3.14}$$

Taking T_3 from the second equation and replacing on the first, we obtain a linear system of two equations on two variable t_1, t_2:

$$\begin{aligned} -T_1 - T_2 &= F_1 + F_2 + F_{c_1} + F_{c_2} = 0 \\ \underbrace{\begin{bmatrix} \cos\psi & \cos(\psi+\alpha) \\ \sin\psi & \sin(\psi+\alpha) \end{bmatrix}}_{A} \underbrace{\begin{bmatrix} t_1 \\ t_2 \end{bmatrix}}_{T} &= \underbrace{\begin{bmatrix} \sin\vartheta_1 & \sin\vartheta_2 \\ \cos\vartheta_1 & \cos\vartheta_2 \end{bmatrix}}_{B} \underbrace{\begin{bmatrix} f_1 \\ f_2 \end{bmatrix}}_{F} + \underbrace{\begin{bmatrix} \cos\psi & \cos(\psi+\alpha) \\ \sin\psi & \sin(\psi+\alpha) \end{bmatrix}}_{A} \underbrace{\begin{bmatrix} m_1 l_1 \\ m_2 l_2 \end{bmatrix}}_{M_l} \dot{\psi}^2 \end{aligned}$$

where $A \in \mathbb{R}^3$ is an invertible square matrix since its determinant is different from zero, $\Delta = \sin(\alpha) \neq 0$ because $\alpha \neq 0, \pi$, otherwise the the shape of the formation would be a degenerate triangle.

$$\begin{aligned} AT &= BF + AM_l\dot{\psi}^2 \\ T &= A^{-1}BF + M_l\dot{\psi}^2 \end{aligned}$$

Thus the tensions t_1, t_2 are:

$$t_1 = \frac{-1}{\sin\alpha}\left(\cos(\psi+\alpha+\vartheta_1)f_1 + \cos(\psi+\alpha+\vartheta_2)f_2\right) \tag{3.15}$$

$$t_2 = \frac{1}{\sin\alpha}\left(\cos(\psi+\vartheta_1)f_1 + \cos(\psi+\vartheta_2)f_2\right) \tag{3.16}$$

Now we can derive t_3 using the first or the second equation. We use both calculating t_3' from the firs equation and t_3'' from the second equation.

$$\begin{aligned}
t_3' &= L_3^T(F_1 + T_1) \\
&= f_1\left(\sin(\psi - \gamma + \vartheta_1) + \frac{\cos(\psi + \alpha + \vartheta_1)\cos\gamma}{\sin\alpha}\right) + f_2\left(\frac{\cos(\psi + \alpha + \vartheta_2)\cos\gamma}{\sin\alpha}\right) \\
&= f_1 c_{11} + f_2 c_{12} \\
t_3'' &= -L_3^T(F_2 + T_2) \\
&= f_1\left(\frac{\cos(\psi + \vartheta_1)\cos(\alpha + \gamma)}{\sin\alpha}\right) + f_2\left(-\sin(\psi - \gamma + \vartheta_2) + \frac{\cos(\psi + \vartheta_2)\cos(\alpha + \gamma)}{\sin\alpha}\right) \\
&= f_1 c_{21} + f_2 c_{22}
\end{aligned}$$

To make sure that $t_3' = t_3''$ we need to proof that $c_{11} = c_{21}$ and $c_{12} = c_{22}$. We start to verify that $c_{11} - c_{21} = 0$

$$\begin{aligned}
c_{11} - c_{21} &= \sin(\psi - \gamma + \vartheta_1) + \frac{\cos(\psi + \alpha + \vartheta_1)\cos\gamma}{\sin\alpha} - \frac{\cos(\psi + \vartheta_1)\cos(\alpha + \gamma)}{\sin\alpha} \\
&= \sin\alpha\sin(\psi - \gamma + \vartheta_1) + \cos(\psi + \alpha + \vartheta_1)\cos\gamma + \cos(\psi + \vartheta_1)\cos(\alpha + \gamma) \\
&= \sin(\psi + \vartheta_1)[\sin\alpha\cos\gamma - \sin\alpha\cos\gamma] + \\
&\quad + \cos(\psi + \vartheta_1)[\sin\alpha\sin\gamma - \sin\alpha\sin\gamma + \cos\alpha\cos\gamma - \cos\alpha\cos\gamma] \\
&= 0
\end{aligned}$$

Similarly, we can verify the other two terms, $c_{12} - c_{22} = 0$. Therefore, we can write t_3 in the easiest form, choosing the best factors.

$$\begin{aligned}
t_3 &= f_1 c_{21} + f_2 c_{12} \\
&= f_1\left(\frac{\cos(\psi + \vartheta_1)\cos(\alpha + \gamma)}{\sin\alpha}\right) + f_2\left(\frac{\cos(\psi + \alpha + \vartheta_2)\cos\gamma}{\sin\alpha}\right) \qquad (3.17)
\end{aligned}$$

Summarizing the three tensions in steady state are:

$$\begin{aligned} t_1 &= \frac{-1}{\sin\alpha}\left(\cos(\psi+\alpha+\vartheta_1)f_1+\cos(\psi+\alpha+\vartheta_2)f_2\right)+m_1l_1\dot{\psi}^2 \\ t_2 &= \frac{1}{\sin\alpha}\left(\cos(\psi+\vartheta_1)f_1+\cos(\psi+\vartheta_2)f_2\right)+m_2l_2\dot{\psi}^2 \\ t_3 &= f_1\left(\frac{\cos(\psi+\vartheta_1)\cos(\alpha+\gamma)}{\sin\alpha}\right)+f_2\left(\frac{\cos(\psi+\alpha+\vartheta_2)\cos\gamma}{\sin\alpha}\right) \end{aligned} \tag{3.18}$$

In this case is not so easy to derive a geometric constraint that ensure taut cables, why the tensions t_1, t_2 depend on both the external forces. This because the third cable takes the effects of both the thrusts on the other two cables. However, looking the results found in the previous section, trying to center the forces on the cone described by the cable axis is a good strategy to have taut cables. On a quality level, we can say that to ensure the tension on the anchored cable the quadrotors have to be perpendicular to the cable as much as possible, and, to ensure the tension on the third rope, both the vehicle have to tilt toward the axis cable direction. We need also to notice that the quadrotor, since has a limited thrust, can't tilt to much, otherwise it wouldn't be able to compensate the gravity.

Now we want to determinate the same equations but in a general case: reintroducing the gravity and supposing any type of acceleration in a rotational movement. At first we define the accelerations $A_i \in \mathbb{R}^2$ and the gravity G_i acting on the i-th quadrotor. We want to recall that, in a rotational movement, the acceleration is composed by a centripetal term, directed along the axis of the anchored rope, and a tangent term, perpendicular to the anchored cable's axis. Moreover, since we assumed taut cable, the angular velocity and acceleration are equal for each vehicles.

$$\begin{aligned} A_1 &= l_1\ddot{\psi}\begin{bmatrix}-\sin\psi\\ \cos\psi\end{bmatrix}+l_1\dot{\psi}^2\begin{bmatrix}-\cos\psi\\ -\sin\psi\end{bmatrix} \\ A_2 &= l_2\ddot{\psi}\begin{bmatrix}-\sin(\psi+\alpha)\\ \cos(\psi+\alpha)\end{bmatrix}+l_2\dot{\psi}^2\begin{bmatrix}-\cos(\psi+\alpha)\\ -\sin(\psi+\alpha)\end{bmatrix} \end{aligned} \tag{3.19}$$

$$
\begin{aligned}
G1 &= \begin{bmatrix} 0 \\ g_1 \end{bmatrix} = \begin{bmatrix} 0 \\ -m_1 g \end{bmatrix} \\
G2 &= \begin{bmatrix} 0 \\ g_2 \end{bmatrix} = \begin{bmatrix} 0 \\ -m_2 g \end{bmatrix}
\end{aligned}
\tag{3.20}
$$

The balance of the forces on the center of gravity of each agent results now:

$$
\begin{cases}
T_3 + T_1 + F_1 + G_1 = m_1 A_1 \\
-T_3 + T_2 + F_2 + G_2 = m_2 A_2
\end{cases}
\tag{3.21}
$$

Using the same approach of before we have:

$$
-T_1 - T_2 = F_1 + F_2 + G_1 + G_2 - m_1 A_1 - m_2 A_2
$$

and, resolving the system, we obtain:

$$
\begin{aligned}
t_1 &= \frac{1}{\sin\alpha}\left[-\left(\cos\left(\psi+\alpha+\vartheta_1\right) f_1 + \cos\left(\psi+\alpha+\vartheta_2\right) f_2\right) + \right. && \text{(3.22a)} \\
&\left. +g(m_1+m_2)\cos\left(\psi+\alpha\right) + \left(m_1 l_1 \cos\alpha + m_2 l_2\right)\ddot{\psi}\right] + m_1 l_1 \dot{\psi}^2 \\
t_2 &= \frac{1}{\sin\alpha}\left[\left(\cos\left(\psi+\vartheta_1\right) f_1 + \cos\left(\psi+\vartheta_2\right) f_2\right) + \right. && \text{(3.22b)} \\
&\left. -g(m_1+m_2)\cos\psi - \left(m_1 l_1 + m_2 l_2 \cos\alpha\right)\ddot{\psi}\right] + m_2 l_2 \dot{\psi}^2 \\
t_3 &= f_1\left(\frac{\cos(\psi+\vartheta_1)\cos(\alpha+\gamma)}{\sin\alpha}\right) + f_2\left(\frac{\cos(\psi+\alpha+\vartheta_2)\cos\gamma}{\sin\alpha}\right) + \\
&+g\left(m_2\sin\left(\psi-\gamma\right) - \frac{(m_1+m_2)\cos\left(\psi\right)\cos\left(\alpha+\gamma\right)}{\sin\alpha}\right) + \\
&+\ddot{\psi}\left(m_1 l_1 \sin\gamma - \frac{\left(m_1 l_1\cos\alpha + m_2 l_2\right)\cos\gamma}{\sin\alpha}\right) && \text{(3.22c)}
\end{aligned}
$$

Like previously, we find two version for the tension t_3, but the coefficients that multiply the terms $f_1, f_2, g, \dot{\psi}, \ddot{\psi}$, are equal. Thus we can chose the easiest form.

Chapter 4

Control

Introduction

In this chapter we want to derive the equations that describe the controller in order to attain a desired orientation of the formation. The purpose is to implement a controller to track a desired ψ^*. Since the inputs of the system are the strength (thrust) produced by the two quadrotor, we determine the control inputs for the quadrotors, using the feedback linearization technique, in order to attain the desired ψ^*. Moreover, since the system turns out to be overactuated, we want to use the second degree of freedom to keep the cables taut. In particular we want to attain a desired tension on the cable that links the two quadrotors.

4.1 Orientation Control

Recalling the dynamic system (3.12) and the model of the quadrotor, we can assume that the control inputs of the generic i-quadrotor are the thrust f_i (lift force) and the pitch angle ϑ_i. Actually, the real input is the rate angle, but we can think to use another controller, in particular the which one described in the Sec. 2.6.2, to perform the attitude control. Moreover, if this controller is good and fast enough, we can assume that the real ϑ_i is always e instantaneously equal to the desired angle ϑ_i^*. This is also validate by the

fact that the dynamics of the quadrotor is faster then the dynamics of the system.

Furthermore, we recall the assumption that the ropes are tethered exactly on the center of mass of the quadrotor, so there aren't any external torques acting on the rotational behavior. These assumptions let us to approximate the quadrotor as an object, of the same mass, able to produce any desired force on the vertical plane, directed upwards.

Let a desired trajectory of the formation orientation, $\psi^*(t)$, we can compute the desired acceleration $\ddot{\psi}^*$ with a classical PD controller:

$$\ddot{\psi}_u = K_D(\dot{\psi}^* - \dot{\psi}) + K_P(\psi^* + \psi) + \ddot{\psi}^* \tag{4.1}$$

If we fix the modulus of the thrust to a certain value, $f_i = f_i^*$, now that we have the control variable $\ddot{\psi}_u$, using the model equation (3.12), we can determine the desired attitude ϑ_i^* for the two quadrotors feedback linearizing the model [70]. In particular we want to determine the control laws $c(v_i)$, function of a new virtual inputs $v_i \in \mathbb{R}$, which erase the non linear part of the model (3.12) and allows to control directly $\ddot{\psi}$. Defining $w_1(f_1, \vartheta_1) := f_1 \cos(\psi + \vartheta_1)$ and $w_2(f_2, \vartheta_2) := f_2 \cos(\psi + \alpha + \vartheta_2)$, we can rewrite the model:

$$\ddot{\psi} = \frac{-g(\psi) + w_1 + w_2}{I}$$

Now since w_1, w_2 are functions of the inputs, we define them as new global input. Using the feedback linearization method, we have to choose the inputs functions, w_1^*, w_2^*, in order to erase the non linearity of the system and directly control the output $\ddot{\psi}$ with two other virtual input $v_1, v_2 \in \mathbb{R}$. Basically we can write the global inputs as

$$\begin{aligned} w_1^* &= \alpha_1(\psi) + \beta_1(\psi)v_1 \\ w_2^* &= \alpha_2(\psi) + \beta_2(\psi)v_2 \end{aligned} \tag{4.2}$$

Replacing it in the model we obtain

$$\ddot{\psi} = \frac{-g(\psi) + \alpha_1(\psi) + \beta_1(\psi)v_1 + \alpha_2(\psi) + \beta_2(\psi)v_2}{I}$$

Now we can choose $\alpha_1(\psi), \alpha_2(\psi), \beta_1(\psi), \beta_2(\psi)$ as

$$\begin{aligned} \alpha_1 = \alpha_2 &= \frac{g(\psi)}{2} \\ \beta_1 = \beta_2 &= \frac{I}{2} \end{aligned} \tag{4.3}$$

Thus the model results

$$\ddot{\psi} = \frac{v_1 + v_2}{2} \tag{4.4}$$

So, using the virtual inputs v_1, v_2 we can directly control the angular acceleration of the system. But we have found a control function only for the global inputs that we defined before. Now we have to derive the real control signal for the quadrotor, in particular the amount of thrust f_i an the pitch angle ϑ_i. If we fix the thrust to a constant value, $f_i = f_i^*$, the only control input for the quadrotor is the pitch angle ϑ_i, and it can be easily found from the definition of w_1, w_2 and the equations (4.2) (4.3).

$$\vartheta_1^* = -\psi + \cos^{-1}\left[\frac{Iv_1 + g(\psi)}{2f_1 l_1}\right] \tag{4.5a}$$

$$\vartheta_2^* = -(\psi + \alpha) + \cos^{-1}\left[\frac{Iv_2 + g(\psi)}{2f_2 l_2}\right] \tag{4.5b}$$

In the previous equations the $\cos^{-1}$ is always well defined. Indeed, if $\psi \in (0, \pi/2)$ then $\vartheta_1 \in (0, \pi/2)$ to ensure the tension in the third cable, so $\cos^{-1}(x) \in (0, \pi)$. Thus the contrary, if $\psi \in (\pi/2, \pi)$ then $\vartheta_1 \in (-\pi/2, 0)$ to ensure the tension in the third cable, thus $\cos^{-1}(x) \in (-\pi, 0)$. Similarly for the function of ϑ_2.

Assuming that $\vartheta_i(t) = \vartheta_i^*(t)\ \forall\ t$, the dynamic model becomes as (4.4). Looking at the equations (4.5), it is necessary to impose some conditions on v_1 and v_2, to avoid singularity calculating $\cos^{-1}$. In particular the argument has to stay between the values -1 and 1.

$$\begin{aligned} \left|\frac{Iv_1 + g(\psi)}{2f_1 l_1}\right| &\leq 1 \\ \left|\frac{Iv_2 + g(\psi)}{2f_2 l_2}\right| &\leq 1 \end{aligned}$$

From the above equations we can find two feasible regions for the two controls input v_1, v_2. In other words, we can find the limits of the saturation. This limits are due to the maximum torque the the quadrotor can apply to the system. Defining $d_1 := 2f_1l_1$ and $d_2 := 2f_2l_2$, we have:

$$\frac{-d_1 - g(\psi)}{I} \leq v_1 \leq \frac{d_1 - g(\psi)}{I} \tag{4.6}$$

$$\frac{-d_2 - g(\psi)}{I} \leq v_2 \leq \frac{d_2 - g(\psi)}{I} \tag{4.7}$$

It is interesting to note that the above limits depend on the thrust amount (f_1, f_2) and on the position of the robot, due to the orientation of the formation (ψ). This is reasonable because the maximum angular acceleration depends on the force that the quadrotor can apply and on the gravity effect, which in turn depends on the position.

Looking at the equation (4.4), we can use a change of input variables $\begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = \begin{bmatrix} \frac{1}{2} & \frac{1}{2} \\ \frac{1}{2} & -\frac{1}{2} \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \end{bmatrix}$ that implies:

$$\ddot{\psi} = u_1 \tag{4.8}$$

From the previous equation we can notice that if the cables are taut and $\vartheta_i(t) = \vartheta_i^*(t)\ \forall\ t$, with the control law (4.5), the system results a simply double integrator. So the system is controllable and, using the PID controller (4.1), we can allocate the poles as we wants.

Moreover (4.8) clearly shows that the system is redundant, i.e. we have two inputs, u_1, u_2 and one output $\ddot{\psi}$. Thus we can chose $u_1 = \ddot{\psi}_u^*$ in order to have the tracking of a desired trajectory $\psi(t)$ and u_2 for another purpose. For example to solve an optimization problem with a specific cost function or to attain a particular behavior. We will show the design of the control input u_2 in the following Sec. 4.2.

Calculated the virtual control variables u_1, u_2 we determine the desired attitudes of the two quadrotors by the equations (4.5), that in the new vari-

able become:

$$\vartheta_1^* = -\psi + \cos^{-1}\left[\frac{I(u_1+u_2)+g(\psi)}{2f_1 l_1}\right] \tag{4.9a}$$

$$\vartheta_2^* = -(\psi+\alpha) + \cos^{-1}\left[\frac{I(u_1-u_2)+g(\psi)}{2f_2 l_2}\right] \tag{4.9b}$$

As we have just said, practically, we can't control directly the attitude of the quadrotor, because the real control input is the angular rate $\dot{\vartheta}_i$ and the thrust amounts f_i. We set $f_i = f_i^*$ constant and we can use a P controller to determine the $\dot{\vartheta}_i$ from the desired attitude:

$$\dot{\vartheta}_i^* = K_P(\vartheta_i^* - \vartheta_i) \tag{4.10}$$

The whole controlled system is showed on fig. 4.1.

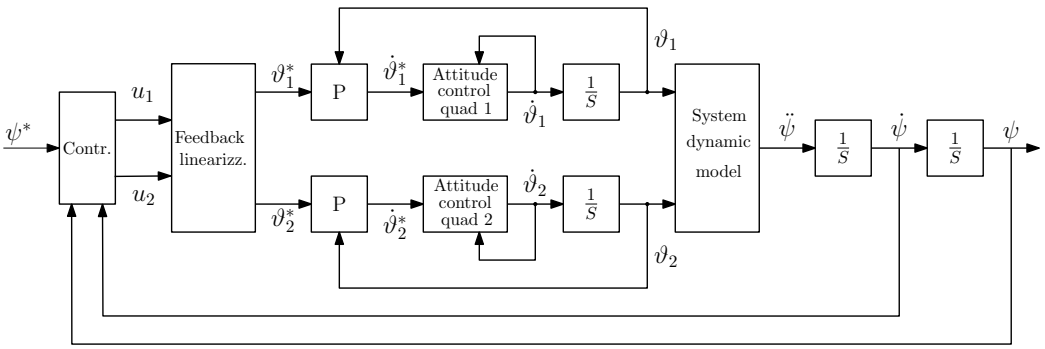


Figure 4.1: Control system: the first block, *Contr.* contain the PID controller and the equations (4.8) and (4.11), that, from the desired angle compute the virtual inputs u_1, u_2. Afterwards the block *Feedback lineariz.* implements the equations (4.9) which compute the desired attitude from the virtual inputs. Then there is the attitude controller of the quadrotor and its dynamics, and finally the whole system.

On [13] is showed that this control architecture works fine also with the assumption $\vartheta_i(t) = \vartheta_i^*(t)\ \forall\ t$ that is not theoretically correct because the quadrotor has its own dynamics and can't attain a desired attitude instantaneously. However, since the dynamic of the quadrotor is faster than the dynamic of the angle ψ, the delay between the commanded pitch and the actuated will be not big.

4.2 Tension Control

As we said before, since we have a redundant system, we can use one of the two DoF to find an optimum control input for u_2. Since the model which we found is valid only if all the tethers are taut, we need to keep them taut as much as possible. The trust of the quadrotors are directed mostly upwards to compensate the gravity, so the two anchored cables have a tiny probability to loose the tension, except for particular cases as for $\psi \approx 0$. The cable that has more problem is the third. This is due to the fact that the quadrotor has to tilt more to stretch the third rope.

For this purpose we can write u_2 as function of a desired tension t_3 and the first control input u_1 that is already fixed from (4.8). Thus, replacing the formulas (4.9) in the equation (3.22c), we derive u_2. For simplify the writing, we define some variables

$$\begin{aligned} e_1 &:= f_1 \frac{\cos(\alpha+\gamma)}{\sin\alpha} \\ e_2 &:= f_2 \frac{\cos\gamma}{\sin\alpha} \\ e_3 &:= g\left(m_2 \sin(\psi-\gamma) - \frac{(m_1+m_2)\cos(\psi)\cos(\alpha+\gamma)}{\sin\alpha}\right) \\ e_4 &:= m_1 l_1 \sin\gamma - \frac{(m_1 l_1 \cos\alpha + m_2 l_2)\cos\gamma}{\sin\alpha} \end{aligned}$$

With this notations, the equation (3.22c) becomes

$$t_3 = e_1 \frac{I(u_1+u_2)+g(\psi)}{2d_1} + e_2 \frac{I(u_1-u_2)+g(\psi)}{2d_2} + e_3 + e_4 \ddot{\psi}$$

Now, setting a desired tension in the third cable, t_3^*, we can derive the control input u_2 which ensure that value of tension.

$$u_2 = \frac{2d_1 d_2 t_3 - (d_2 e_1 + d_1 e_2) I u_1 - g(\psi)(d_2 e_1 + d_1 e_2) - 2d_1 d_2 e_3 - 2d_1 d_2 e_4 \ddot{\psi}}{I(d_2 e_1 - d_1 e_2)} \tag{4.11}$$

The last equation has not singularities because the denominator is always different from zero. Indeed $I \neq 0$, and

$$\begin{aligned} d_2e_1 - d_1e_2 &= f_2f_1l_2\frac{\cos(\alpha+\gamma)}{\sin\alpha} - f_1f_2l_1\frac{\cos\gamma}{\sin\alpha} = \frac{f_1f_2}{\sin\alpha}\left(l_2\cos(\alpha+\gamma) - l_1\cos\gamma\right) \\ &= \frac{f_1f_2}{\sin\alpha}\left(l_2\cos(\pi-\beta) - l_1\cos\gamma\right) = \frac{f_1f_2}{\sin\alpha}\left(-l_2\cos\beta - l_1\cos\gamma\right) \\ &= \frac{f_1f_2}{\sin\alpha}\left(-l_3\right) \neq 0 \end{aligned}$$

Since we don't have sensors to measure the internal tension in the rope, this control technique is a simpler feedforward term determined from the equations of the system. Although it is not a feedback controller and we can't practically have zero error, actually we want only to have a value of tension near to the desired one, not so precisely, in order to be sure to keep the cable taut.

4.3 Tuning of the PD Controller

In the Sec. 4.1 we found that using the control inputs (4.9), which come from the feedback linearization, the model becomes a simpler double integrator (4.8). Moreover, to attain a desired orientation, ψ^*, or to track a desired trajectory $\psi^*(t)$, we decided to use a common PD controller, (4.1).

Now we want to shortly proof that the system is controllable, and, obtaining the equation of the poles of the system, we want to find a constraint on the gains in order to have purely real negative poles. Considering the equation (4.8), we can rewrite the model in state space, defining the vector state as $X = \begin{bmatrix} \psi & \dot{\psi} \end{bmatrix}^T \in \mathbb{R}^2$:

$$\dot{X} = \underbrace{\begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}}_{A} X + \underbrace{\begin{bmatrix} 0 \\ 1 \end{bmatrix}}_{B} u_1 \tag{4.12}$$

looking a the controllability matrix, $\mathcal{R} = \begin{bmatrix} B & AB \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$, which is full rank, we can immediately say that the system is controllable. Then, recalling the equation (4.1) and taking the virtual control input $u_1 = \ddot{\psi}^*$, the equation (4.12) becomes:

$$\begin{aligned} \dot{X} &= \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} X + \begin{bmatrix} 0 \\ 1 \end{bmatrix} \left(- \underbrace{\begin{bmatrix} k_P & k_D \end{bmatrix}}_{K} X + \underbrace{\begin{bmatrix} k_P & k_D & 1 \end{bmatrix} \begin{bmatrix} \psi^* \\ \dot{\psi}^* \\ \ddot{\psi}^* \end{bmatrix}}_{D} \right) \\ &= (A - BK) X + BD \end{aligned}$$

Since the characteristic polynomial of $(A - BK)$ is $p_{(A-BK)}(\lambda) = \det(\lambda I - (A - BK)) = \lambda^2 + k_D\lambda + k_P$ and the PD's gins are $k_P, k_D > 0$, for the theorem of Cartesio, the eigenvalues have always negative real part, i.e. the system is asymptotically stable. In particular the eigenvalues of the system are:

$$\lambda_{1,2} = \frac{-k_D \pm \sqrt{k_D^2 - 4k_P}}{2} \tag{4.13}$$

In order to have the cables always taut, it is reasonable to set the controller such that the response of the system is as smooth as possible. Indeed, whit a slow movement, the internal tension would be almost constant. To have this behavior we can impose purely real poles. From the equation (4.13) it has to be

$$k_D^2 - 4k_P > 0 \tag{4.14}$$

In this way the response of the system will be without overshoots end oscillation. Then we can increase the values of the gains to improve the speed of the system. However they can't be too much height for not go in the saturation region. Moreover, near the saturation, the quadrotors has to tilt a lot and this can be a problem for the tension of the two anchored rope and for the attitude control and estimation because, as we saw in the Chap. 2,

they are valid for the near hovering condition. In addiction, if the gains are big, also the control inputs ϑ_i^* will be height and with fast variation that can invalidate the assumption $\vartheta_i(t) = \vartheta_i^*(t) \ \forall \ t$. On the other way, with low gains, especially in the experiments, it could be more difficult to keep constant the desired angle ψ^*. During the simulation we will do a first empirical tuning, trying to set the gains as hight as possible, considering the limitations reported above.

Chapter 5

Simulations

Introduction

In this chapter we will show what kind of models we used for simulate every parts of the whole system. In particular we will show the behavior of the two model of the rope found in the Sec. 3.1. Afterwards we setted the parameters to obtain a object that reproduces as much as possible the behavior of the cable used for the experiments. Then we will show how we created the simulator using *Simulink*. At the and we will present various simulations, at first to validate the model, and than to test the controller designed in the Chap. 4. In particular we will take attention to the error between the desired and the actual orientation and tension value on the cables. Moreover, we will do a first tuning of the PD controller, observing the resulting response of the system to different types of references, in particular to the step and to the ramp.

5.1 Cable Model

In the Sec. 3.1 we modeled the cable with two different system. At first we modelled it as a spring with an high elastic constant, whereas, in the second step we derived a more sophisticated model, the Standard Solid Model that use a combination of two spring and a dumper. In the following sections we

will show the behavior of the two types of model, in order to understand which model describe better the real dynamics of a rope. To test the models, we simulated an object with mass $m = 1$ [kg] tethered to a rope of length $l_0 = 1$ [m] fixed in the origin. Then we observed the trajectory of the object in a free fall from the start hight $z_0 = 0$ [m]. Moreover we added a damping effect to the object due to the aerodynamics, with damping coefficient $b = 0.5$ [N·s/m], in order to avoid constant oscillations with the spring cable's model.

In the Fig. 5.1 is represented the schematic of the simulated system. In this way we want to understand the behavior of the model during the stretching of the rope.

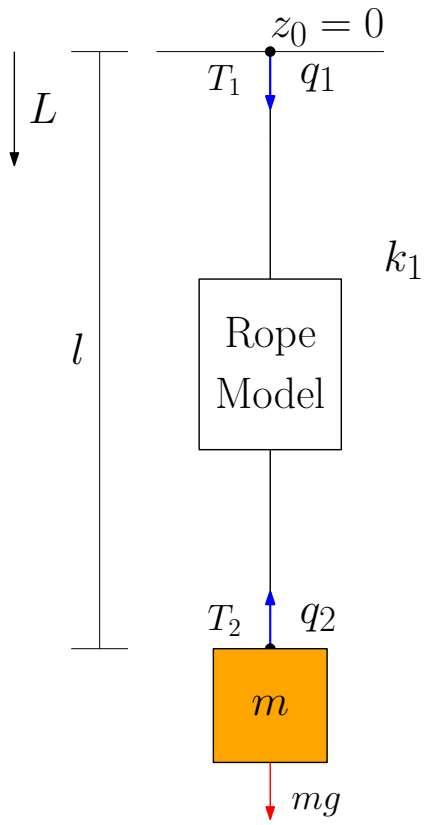


Figure 5.1: Schematic of the simulated system: An object of mass m attached to a cable fixed to the ceiling at height $z_0 = 0$ [m].

Cable as a Spring

In Fig. 5.2 is showed the trajectory of the object and the internal tension on the rope for three different elastic coefficients. As we can see, increasing the stiffness, the maximum tension increases, the maximum elongation decreases, but increases also the oscillation frequency. This behavior differs from the real for two reasons: we don't reach a constant tension equal to mg, then, we have a lot of "jumps" that decrease due to the damping effect on the object. On the contrary, what we expect should be few jumps and then a constant tension.

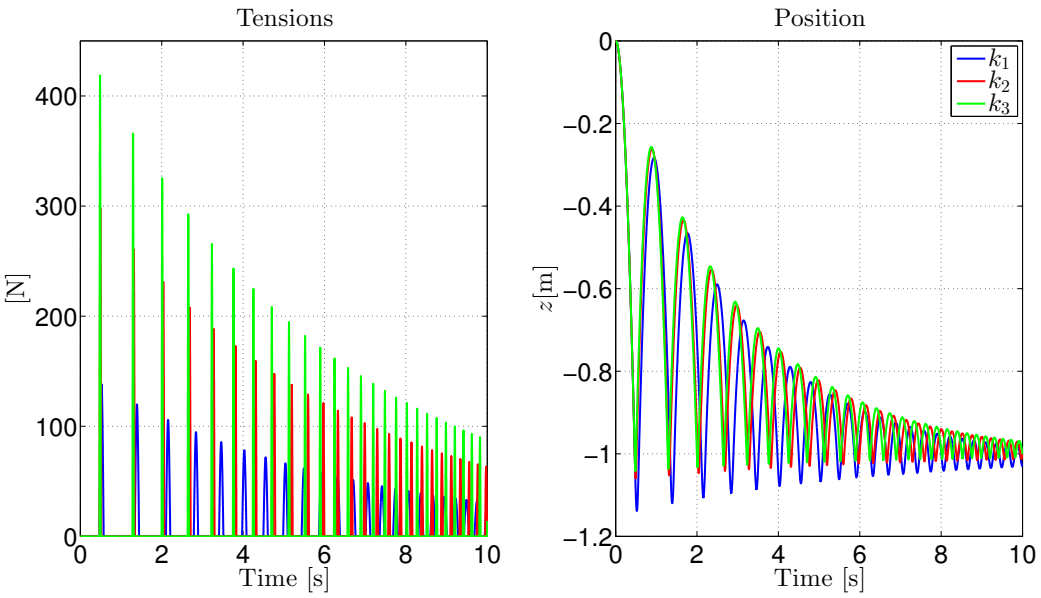


Figure 5.2: Free fall behavior for a object attached to a cable modeled by a spring. On the left the diagram of the internal tension, on the right the position of the object for three different elastic coefficients: $k1 = 500$ [N/m], $k1 = 1000$ [N/m], $k1 = 5000$ [N/m].

Cable as a Standard Linear Solid Model

To simulate the free fall with the SLS model for the cable, we need at first to set the system parameters, i.e. the elastic and viscosity coefficients k_1, k_2, b_1. To find some good initial values, we can use the relations (5.1), that express

the elastic and viscosity parameters in terms of the characteristic parameters of a rope, in particular: diameter, length, elastic modulus and viscosity. We took, as a first step, the constants on Tab. 5.1.

$$q = \pi \left(\frac{d_c}{2}\right)^2 \tag{5.1a}$$

$$k_1 = \frac{qE_1}{l} \tag{5.1b}$$

$$k_2 = \frac{qE_2}{l} \tag{5.1c}$$

$$b_1 = \frac{q\eta}{l} \tag{5.1d}$$

Name	Coefficient	Value	Unit
Diameter	d_c	1×10^{-3}	[m]
Elastic modulus	E_1	194×10^6	[N/m^2]
	E_2	543×10^6	
Viscosity	η	96×10^6	[N/m^2· s]
Cable length	l	1	[m]

Table 5.1: Coefficient values for a standard rope gotten from [64].

Since we don't know the parameters of our cable, from this starting values, we could change the coefficients in order to obtain the desired behavior, which is most similar to the real one. In Fig. 5.3 is showed the free-fall behavior, in the same previous conditions, with the three different configurations of the parameters showed in Tab. 5.2.

In the first raw there is the values obtained from the equations (5.1) using the coefficients in Tab. 5.1. As we can see in the Fig. 5.3, whit this

Configuration	b_1 [N/m· s]	k_1 [N/m]	k_2 [N/m]
1	75	152	426
2	50	100	600
3	10	100	600

Table 5.2: Coefficient values for three free-fall simulation.

configuration we obtain a lot of jumps before to reach the stable length. Moreover, looking at the graphics of the tensions, there are a lot of peak in the first part. So we adjusted the parameters, reducing k_1 and b_1 to reduce the frequency and the amplitude of the jumps during the fist part of the stretch. We also increased k_2 to improve the stiffness of the rope. As we can notice, the third configuration is probably the best choice. Indeed, after one "bounce" the object reaches the stationary position and also the tension becomes constant soon, in other words, the transient goes on for very few time. In this way, passing from slack rope to taut rope, we don't have to much oscillations. Since we showed that this latter model describe better the dynamics of a real cable, we used this to simulate the whole system.

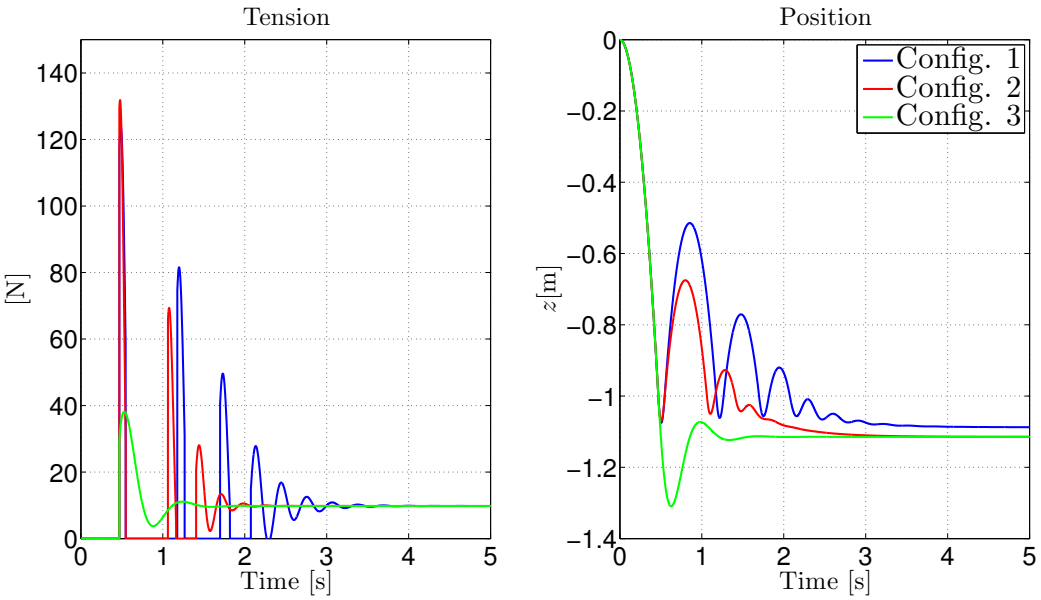


Figure 5.3: Free fall behavior for an object attached to a cable modeled by the SLS model. On the left the diagram of the internal tension, on the right the position of the object for the three different configuration reported in Tab. 5.2.

In Fig. 5.4 is reported the block simulink used for the simulation. It has as inputs the position of the two edges, and as output the two opposite tensions on the end of the cable. In the state there are all the informations, i.e. tension, cable's axis and length. Inside the block, the model (3.5) is devel-

oped. The tension outputs are used for the interaction with the quadrotors.

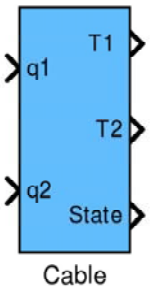


Figure 5.4: Simulink block for the cable model.

5.2 Quadrotor Model

To simulate the quadrotor, we implemented in simulink the equations (2.4) that describes its dynamics, neglecting the motors dynamics and the aerodynamics effects. Then we implemented the attitude controller described in Sec. 2.6.2. For the parameters of the quadrotors, as mass $m \in \mathbb{R}$ end inertia $I \in \mathbb{R}^{3\times3}$, we didn't do an accurate identification because our purpose is not to compare the simulation end experiments results, so we don't need precise values. Thus we used some reasonable values for the real quadrotor used for the experiments. The values are reported in Tab. 5.3.

Parameters		
Param	Value	Unit
m	0.021	[kg]
I	$5 \times 10^{-3} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$	[kg·m^2]

Table 5.3: Parameters of the quadrotor used in the simulation.

To tune the attitude controller gains in the equation (2.7), we did some simulations in order to study the response of the controlled system to a

step in the desired angle. To set the controller parameters we used the usual intuitive tuning rules for the design of PID controllers [71], in order to obtain a smooth response as fast as possible but without overshoots. After some simulations, the best gains that we found are reported below:

$$\begin{aligned} k_P &= 100 \\ k_D &= 20 \end{aligned}$$

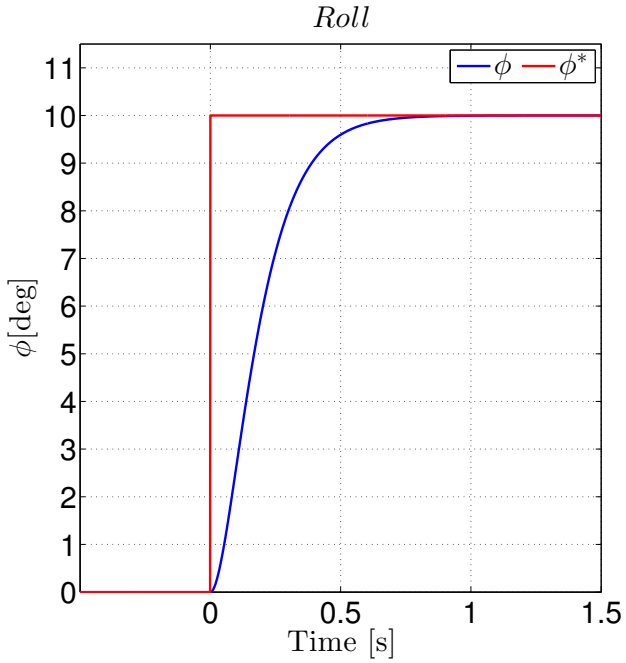


Figure 5.5: Response of the actuated roll to a step reference of ten degree.

For this gain values, in the Fig. 5.5 is reported the step response for the actuated roll. We set a desired thrust equal to the hovering value and we sent a step of ten degree in the desired roll, ϕ^*. As we can see, the response is sufficiently fast but without any overshoots and oscillations that can destabilize the quadrotor during the flight. We don't report also the step response of the pitch angle because it is perfectly the same since the dynamics is symmetric and the gains are equals.

In the figure Fig. 5.6 is showed the simulink block used for the simulations. Is contains the quadrotors dynamics and the attitude controller, indeed the inputs are the desired attitude and thrust, while, in the output, we have the whole state of the quadrotor, i.e. position, linear velocity and acceleration, Euler angles which describes the attitude, and the thrust in the world frame used for the plots. Moreover we add as input, the external force acting on the quadrotor to reproduce the interaction between it and the attached cables.

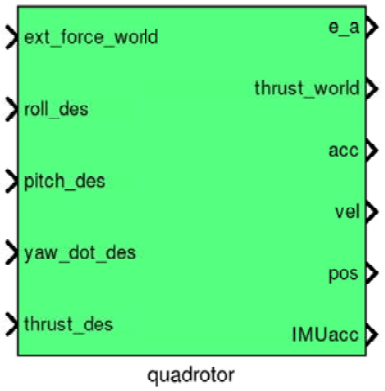


Figure 5.6: Simulink block for the quadrotor model with inside also the attitude controller.

5.3 Whole System Model

To create the simulator for the whole system described in the Sec. 3.2, we used the two Simulink blocks showed before to simulate the two quadrotors and the three cables, connecting every object in the proper way. Essentially, the cables ends are fixed on the center of mass of the quadrotor, and the last one interact with the cables by the forces produced by the tensions on them. We designed a Simulink system for the cables (Fig. 5.7) which contains all the cables having as input the position of the three corner of the triangle, or rather the position of the two quadrotors and of the anchor. As output it has the internal tension acting on the cables, i.e. $T_1, T_2, T_3, -T_3$ defined in the Sec. 3.2.

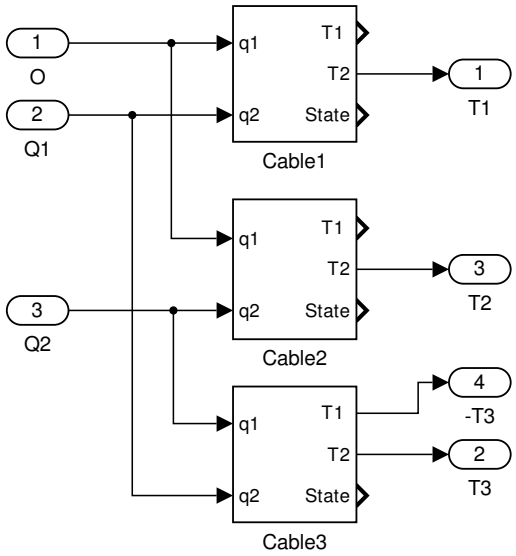


Figure 5.7: Simulink block for the system of cables.

Then, this system is connected to the Simulink blocks of the two vehicles like in Fig. 5.8, where the internal tensions of the ropes, coming out from the previous block, are connected to the external force input of the quadrotors model. Recalling the controller and the assumptions in the Chap. 4, the inputs of the quadrotors are the desired pitch angle, ϑ_i^*, and the desired total thrust, f_i. While, since we restricted the model in a vertical plane, we fixed the quadrotors on it setting the desired roll and the desired yaw rate to zero. In this way, since in simulation there aren't any disturbances, recalling the frames of the system (Fig. 3.4)), zero roll and zero yaw rotation correspond to zero movement along the y direction in the world frame. This means that, if we start the simulation with the quadrotors already positioned with $y = 0$, they will keep that position coordinate for all the simulation, staying in the vertical plane.

Since in the practice this is not possible because there are many disturbances in the measures and in the actuators, we will explain in the Chap. 7 how we fixed the quadrotor position on the desired vertical plane.

To verify if the model replicate the real behavior, at least in steady state condition, we simulated a symmetric formation, imposing equal length for all the cables. Then we fixed the desired total thrust at 0.25 [N], more than the

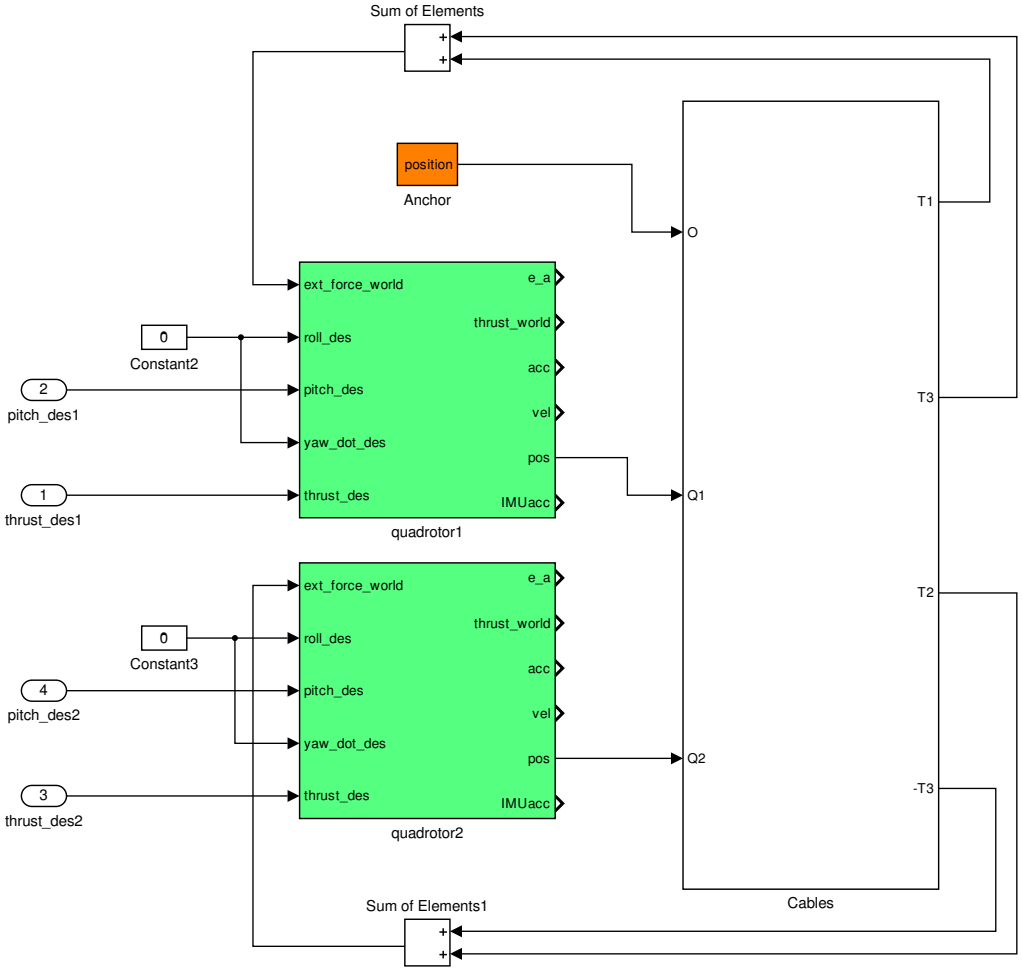


Figure 5.8: Simulink block for the entire system. The two quadrotor blocks are connected to the system of cables, represented in Fig. 5.7, and interact with them by their the internal tensions.

hover thrust, and we imposed a constant symmetric pitch for the quadrotors, $\vartheta_1 = 15°$ $\vartheta_2 = -15°$. Starting with the cables already taut we verified that the system doesn't become unstable if we use constant inputs. What we obtained, is a constant behavior described by the Fig. 5.10, confirming that the forces keep the cable taut without any oscillation.

To possible to understand graphically the configuration of the formation and the attitude of the two quadrotors, we implemented a stylized animation which represents the system during the simulations. the Fig. 5.9, that is a

picture of an animation. The blue lines correspond to the taut cables, the black segments represent the quadrotors and the red ones are their trust vector.

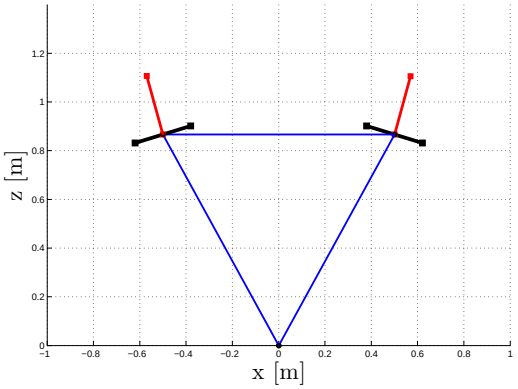


Figure 5.9: Representation of the system in a steady state behavior, imposing constant inputs without any control.

5.4 Simulations and Tuning of the Controller

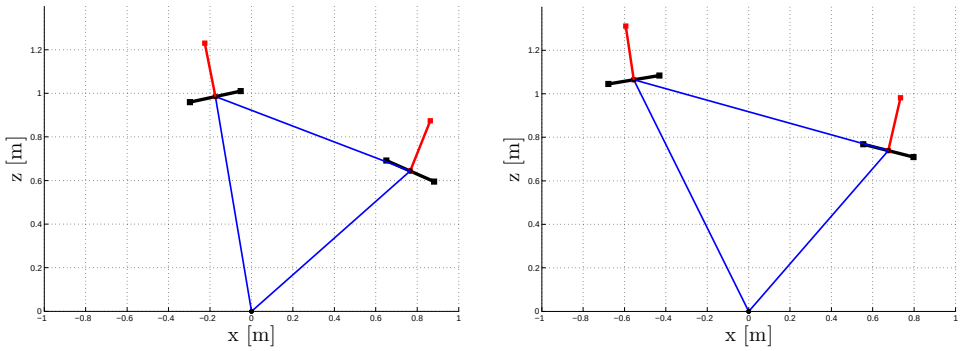
In the following paragraphs we will do various simulations and tests to validate the controller and then to tune the gains of the PD to attain some desired characteristics in the tracking of the desired orientation, $\psi^*(t)$. In particular we will first tune the controller analyzing the step response, and afterward we will improve it for the ramp response which is a more reasonable trajectory.

In all the simulations, if it is not specify differently, we imposed a formation equal to a equilateral triangle, using cable of the same measure, $l_i = 1$ [m] $\forall\ i = 1, 2, 3$. Moreover we fixed the total thrust to a value grater than the hover thrust but lower than the maximum thrust, $f_1 = 0.25$ [N]. This choice is due to the fact the we need sufficiently thrust to compensate the gravity and to apply tension to the ropes. Thought, the thrust can't be near the maximum value otherwise, since each motors can apply a limited lift force, the maximum torque obtainable to drive the quadrotor would be too low. More easily, in this case we have that the system would be in sat-

uration, thus the attitude controller could not perform the desired angles in the proper way.

5.4.1 Steady State Behavior

At first we want to verify if the controller is able to keep a desired constant angle ψ^* and tension t_3^* on the third cable, starting the system already in the desired situation, with the cables taut and $\psi = \psi^*$, $t_3 = t_3^*$. Although we are in the hypothesis of the Chap. 4, and so we know that the system is stable, actually we want simply to ensure that the designed controller is stable at least in steady state. We imposed $\psi^* = 40°$ and $t_3^* = 0.05$ [N] and, at first, equal cable's length, $l_i = 1$ [m] $\forall\, i = 1, 2, 3$.



(a) (b)

Figure 5.10: Representation of the system in a steady state behavior, imposing constant desired orientation, ψ^*.
(a): the formation is an equilateral triangle with $l_i = 1$ [m] and $\psi^* = 40°$.
(b) the formation is a scalene triangle with $l_1 = 1$ [m], $l_2 = 1.2$ [m], $\alpha = 70°$ and $\psi^* = 50°$.

The whole system is showed in Fig. 5.10a, whereas, in Fig. 5.11, starting from the left, are plotted the tension values of the ropes, the pitch angles of the quadrotors, and the global orientation of the formation. So we can say that, the controlled system, if starts from the desired situation (taut cable, desired orientation, and desired tension on the third cable), it is stable and the controller keep constant the orientation, ψ, keeping all the cables taut.

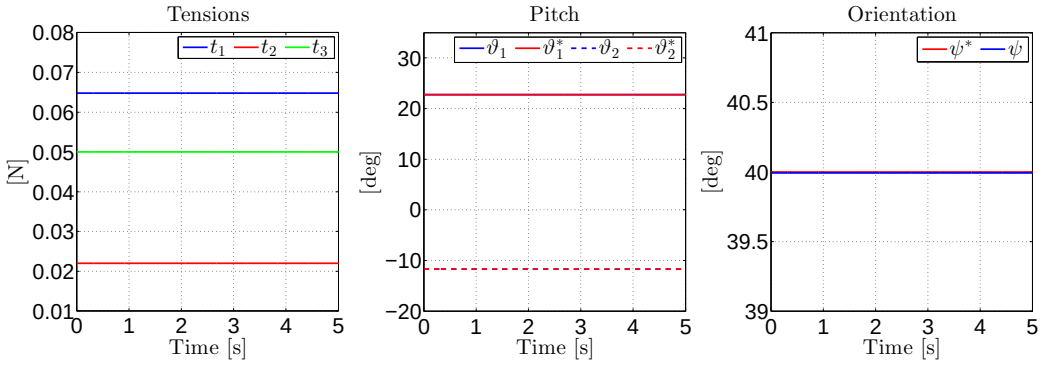


Figure 5.11: Main variables of the system, from the left the tensions of the cable, the pitch angles of the quadrotors, and the formation's orientation. Starting from the desired situation ($\psi^* = 40°$, $t_3^* = 0.05$ [N]), the controller is able to keep constant the orientation and the tension on the third cable at the desired values.

Moreover this is true also with a non symmetric formation represented by a scalene triangle (Fig. 5.10b).

5.4.2 Step Response and Tuning

Now that we verified that the controlled system is at least stable in steady state condition, we have to analyze its behavior during the transient. Moreover we need to tune the gains to attain some desired characteristics of the response of the system to the step and ramp references.

As we said previously in the Sec. 4.3, to ensure taut ropes, it is suitable to have the response of the system as smooth as possible. Indeed, our goal is not the speed of the system, but to have taut ropes all the time and the desired orientation at steady state. Thus, if the changes on the system state are low and gradual, we will avoid sudden variation on the tension that can make the cable slack. Moreover, aggressive and fast responses involve big control inputs which correspond big and fast variation on the desired pitch of the quadrotors. Since the latter has limits on the actuators, if the desired pitch is too much variable, there will be delays and errors between the desired and actual angles. This differences don't respect the assumption $\vartheta_i(t) = \vartheta_i^*(t)\ \forall\ t \in \mathbb{R}$, made in the section Sec. 4.1, and can destabilize the

system.

So we can not set the poles of the controlled system too fast, i.e. we can't place them too much in the left part of the plane. Moreover, to obtain a smooth response without any oscillation, we need to impose purely real poles. This means that the gains of the PD controller have to follow the constraint (4.14) which ensure imaginary part equal to zero.

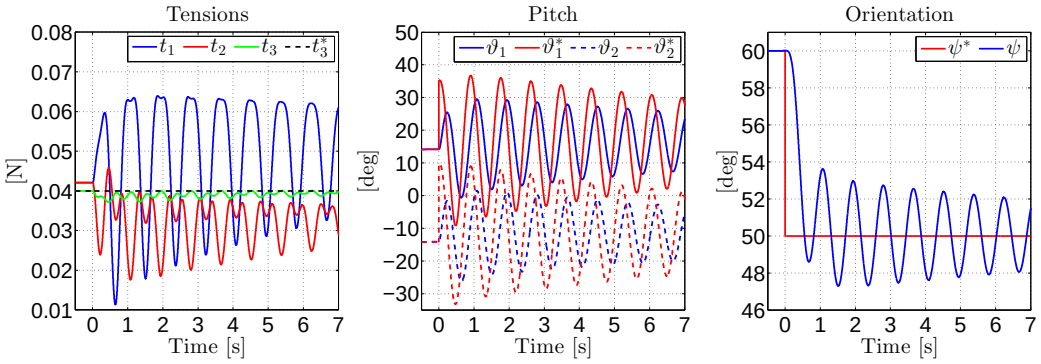
To tune the gains we followed an iteration procedure doing some simulations and changing the parameters at any steps. We started with hight gain values reducing them until we reached a good configuration, always respecting the constraints that ensures real poles. Some significant configurations of the parameters are reported in the Tab. 5.4, while, in the Fig. 6.10 are reported the correspondent responses to a step of ten degree. We started the system with taut cables and initial orientation $\psi = 60°$, then at time $t = 0$ we observed the response of the system to a step of the desired orientation, from $\psi^* = 60°$ to $\psi^* = 50°$. Moreover we imposed a desired tension in the third cable equal to $t_3^* = 0.04$ [N]

Config.	k_P	k_D
1	50	20
2	25	15
3	10	8

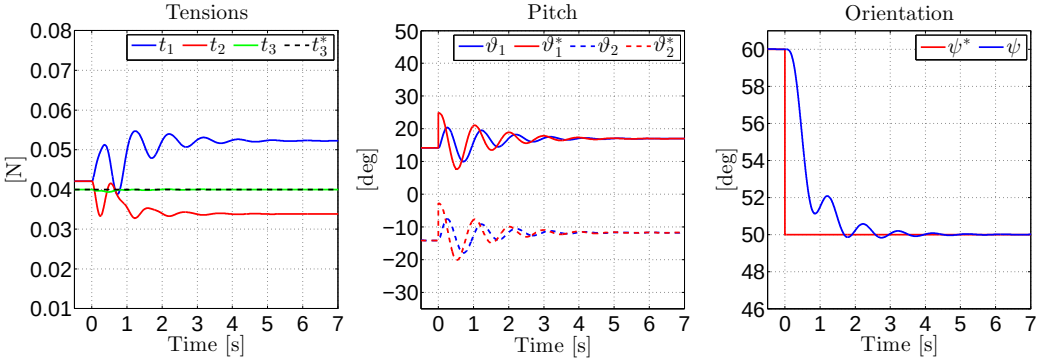
Table 5.4: Iteration for the manual tuning of the controller's parameters

In the first configuration (Fig. 5.12a), with $k_P = 50$, $k_D = 20$, we can observe that the gains are so much hight that the quadrotors are not able to attain the desired pitch angles. This error between the desired and the actuated pitch involves big oscillations in the orientation of the formation. Moreover, the desired tension on the third cable is almost obtained if we neglect a little oscillation around the desired value. Though, the tension on the other two cables is very variable with big oscillations.

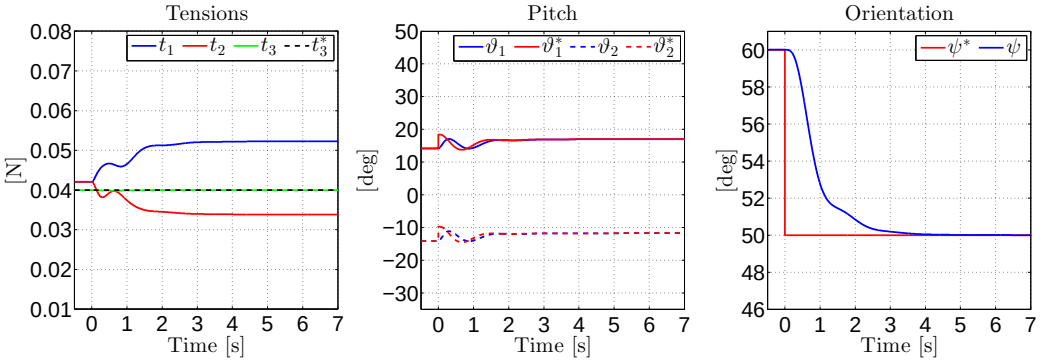
Reducing the gains to $k_P = 25$, $k_D = 15$, we observe in the Fig. 5.12b that the behavior is greatly improved. Indeed the desired pitch angles now are smaller than previously and are attained better than before with only



(a) Configuration 1: $k_P = 50$, $k_D = 20$.



(b) Configuration 2: $k_P = 25$, $k_D = 15$.



(c) Configuration 3: $k_P = 10$, $k_D = 8$.

Figure 5.12: Response of the system of a step of teen degree, from $\psi^* = 60°$ to $\psi^* = 50°$. (a) (b) (c) correspond to the response with three different gains of the PD controller. From the left there are the graphics of the internal tension of the ropes, the desired and actuated pitch angles for the qua quadrotors and on the right the comparison between the desired and actuated orientation of the formation.

a little delay. Now the dynamics of the orientation $\psi(t)$ is smoother then the configuration one. There are still some oscillations but that disappears after 6 [s]. Also the dynamics of the internal tension is smoother and the desired tension t_3^* is attain almost perfectly except for the initial part of the transient.

Finally, with the third configuration, $k_P = 10$, $k_D = 8$, we have the best behavior. In this situation, although the system is slower, the error between the desired and actuated pitch angles is very little. Only in the first instant, when the error $e_\psi = \psi^* - \psi$ is maximum, there is a step in the desired pitch that can not be immediately attained. Indeed, with this kind of reference, at the step time, it is required an impulsive acceleration of the system and in particular of $\ddot{\psi}$. This means that the quadrotors should rotate instantaneously, but due to their dynamics, is not possible. However, reducing the controller's gain, we reduce the error between the desired and actuated pitch angles and we obtain the dynamics of the system's orientation, $\psi(t)$, very smooth and without any overshoots. Also the evolution of internal tension on the cables change slowly. This mean that there aren't tear on the ropes or sudden stretches, and the cables are kept taut. Moreover the internal tension in the third cable is almost constant and equal to the desired value.

5.4.3 Ramp Response

Usually we never use as step as reference signal. Thus, we want now investigate the response of the controlled system to a more suitable reference such as a ramp. In this section we will use the gains of the third configuration determined in the previous section, and a reference signal equal to a limited ramp starting form $\psi^* = 60°$ to $\psi^* = 50°$ with a slope rate equal to $\frac{\Delta\psi^*}{T} = -2$ [deg/s]. This means the the desired orientation goes from the initial ψ to the final one in 5 [s].

In the Fig. 5.13 is reported the response of the system to the ramp. With the previous gains we can notice that the error between the desired and actuated pitch angles is very little and increase only at the start and at the end of the ramp. Indeed, although this reference trajectory is smoother than

before, its desired velocity is discontinued at the start and stop of the ramp. This involve, like for the step, impulses on the desired acceleration $\ddot{\psi}^*$, which are impossible to attain. However, the little error $e_{\vartheta_i} = \vartheta_i^* - \vartheta_i$ only imply a little delay in the actuated orientation $\psi(t)$.

We saw that using smoother and continuous desired trajectory we obtain better performance on the tracking. The best solution should be to use sinusoidal trajectories that are continuous in all the derivates, or cubic splines that have continuous acceleration. Although this method will take excellent performances, is not the purpose of this thesis to design a good trajectory for the formation.

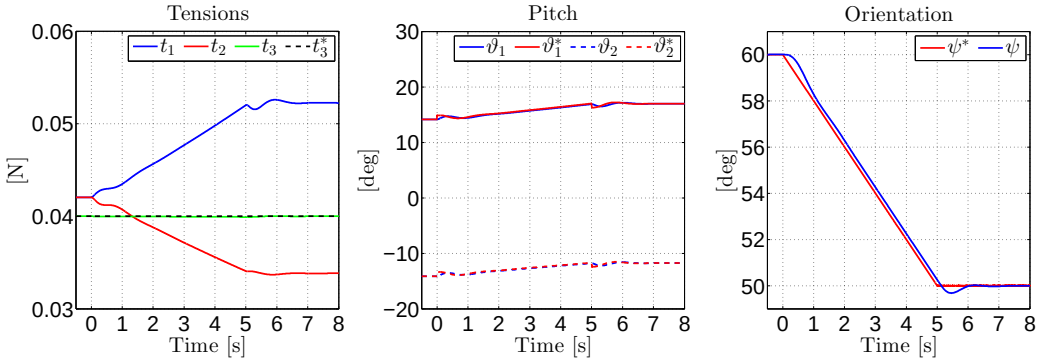


Figure 5.13: Response of the system to a ramp with PD's gains $k_P = 10$, $k_D = 8$.

Chapter 6

Experimental Testbed

Introduction

In this chapter we will describe the development of an experimental testbed to conduct validation experiments for the designed system. The purpose is to create an experimental system suitable for any kind of experiments which use a new type of nano quadrotor, *Crazyflie*.

In particular, in the laboratory of the *Autonomous Robotics and Human-Machine Systems* at the *Max Planck Institute for Biological Cybernetics*, there is already a testbed system, based on ROS, for conducting experiments with quadrotors, called *TeleKyb*. The *Robot Operating System* (ROS) is a flexible framework for writing robot software. It is a collection of tools, libraries, and conventions that aim to simplify the task of creating complex and robust robot behavior across a wide variety of robotic platforms. Then TeleKyb is on of this tools implemented in our lab to control and design applications for quadrotors. So we had only to create an interface that allows the control system TeleKyb to interact with the Crazyflie.

In this chapter we will describe briefly the Crazyflie, its hardware, software and control system. Then we will explain ROS and what is TeleKyb, how it works and what are its purposes and motivations. Finally we will show the implementation of the interface for the Crazyflie.

6.1 Crazyflie

Figure 6.1: Crazyflie Nano Quadrotor.

The Crazyflie is a recently nano quadrotor designed by Btcraze company [72]. The Crazyflie platform is a completely open development platform consisting of open hardware and open source firmware/software letting the research community to improve and develop application with it.

The quadcopter weighs about 20 [g] with a battery mounted, and it has a maximum payload of 10 [g]. It measures 3.8 [cm] along the side of the base and 9.8 [cm] between opposing engines. In flight, the power to the quadcopter is delivered from an 170 [mAh] lithium-ion polymer (LiPo) battery with 3.7 [V] nominal voltage which provides up to 7 minutes of flight time. The quadcopter is also powered on, and charged, when connected by USB to a computer.

The heart of the quadcopter is an 32-bit STM54F325CB Cortex-M5 microcontroller (MCU), running at 72 [Mz]. The quadcopter uses several tasks to control the motors, read sensor values and to communicate with the computer using a radio chip. The board contains the control circuitry for a 3-axis accelerometer and 3-axis high-performance MEMs gyros (Invensense MPU-6050), which means that it is capable of self-leveling. Moreover it has a 3-axis magnetometer (HMC5883L) and a barometer (MS5611). This theoretically let to stabilize the heading direction of the quadrotor and also it height. The frame is made of the circuit board itself, and the motors are attached using

small plastic adapters which slide onto the board.

The Crazyflie quadcopter is controlled wireless from a nearby computer, using a Universal Serial Bus (USB) dongle that works at 2.4 [GHz] and that can perform communications along 125 different channels with a 2Mbps, 1Mbps and 250Kbps communication data-rate. Then it can send and receive data packets of up to 32 bytes.

6.1.1 Firmaware

The firmware, implemented in C, is based on a real time operating system which is able to run different tasks such as radio communication, stabilization, power management, motor control etc. The stabilization algorithm, which runs at 250 [Hz] is the same presented in the Sec. 2.6.2, which is schematically represented in the Fig. 6.2

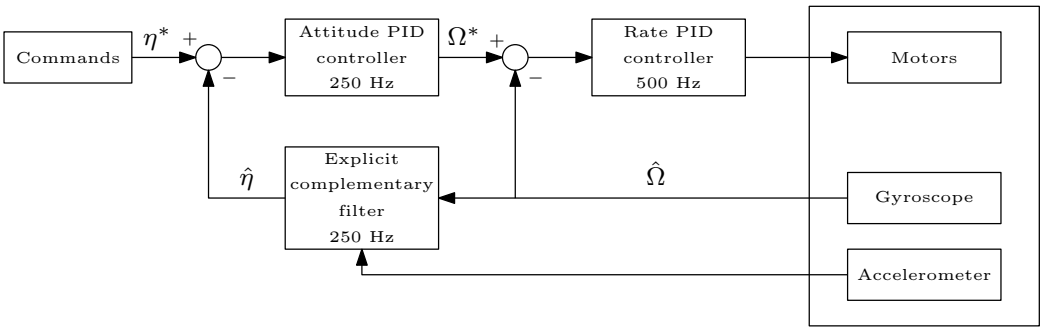


Figure 6.2: Block diagram of the control system implemented on the firmware.

Essentially the schematic is equivalent to that in the Fig. 2.9, where the first block, that computes the desired angular velocity from the attitude error, runs at 250 [Hz], while, the second block that computes the actuator commands from the angular rate error, runs faster, at 500 [Hz].

The control loop for the angular rate is closed directly with the data from the gyroscope at a frequency of 500 [Hz]. Whereas, for the attitude controller, the estimation of the Euler angles are provided by explicit complementary filter presented in Sec. 2.7.1.

6.1.2 CrazyClient

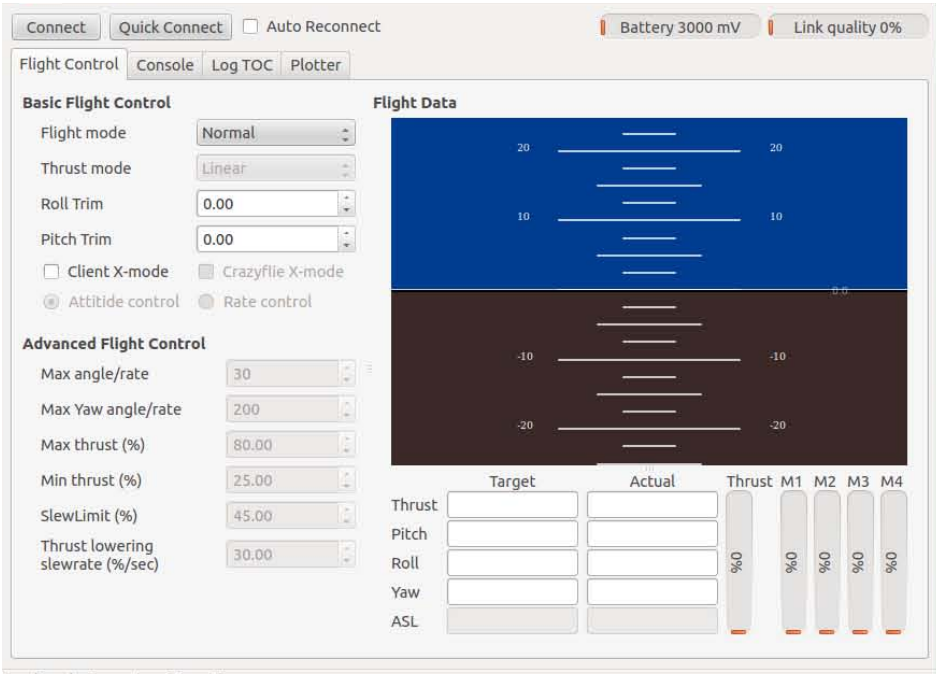


Figure 6.3: The Crazycilent is an user interface application to control the quadrotor and to read data from it.

A client application, called CrazyClient, which is written in Python, is used to control the quadcopter from the computer. It reads input data from a gamepad and sends commands to the quadcopter via the USB radio dongle. The client application shows gamepad input values and various data from the quadcopter, such as the attitude and the power motor, as can be seen in Fig. 6.3. This user interface application is based on a Python library that gives high-level functions and hides the details in order to easily use and control the Crazyflie. This graphical interface allows the user to immediately drive to quadrotor, simply connecting a gamepad and the radio dongle to the computer. There are also options for fine-tuning different flight parameters such as an offset on the commands to calibrate the vehicles, or to set some limits such as the maximum thrust, maximum angles rate and so on. Moreover it gives the possibility to directly plot pre-defined variables or defined by the user.

However, this interface can't be practically used for research purpose, or at least it should be changed a lot. Because using it, we can send commands only through a joystick. If we want to use an external controller, such as the position controller described in the Sec. 2.6.3, we should thoroughly study the client's code and change it to make possible to send data from outside. Moreover, also the data coming from the quadrotor can't be saved or read form other programs or applications. Also for this we should change the code. Since the changes on the existing code would be excessive, we preferred to develop our own interface from scratch that we will present later in the Sec. 6.4. Fortunately, the CrazyClient is based on a open source library (`cflib`) that provides methods to connect the quadrotor, send commands and define log packages that we want to receive from the robot at a specified frequency. Thus, using this library we can implement our interface based on ROS more easily and at an high level. Indeed this library made in Python hides the details and allows to neglect the low code used for example to communicate with the quadrotor through the wifi radio dongle.

6.2 ROS

The Robot Operating System (ROS) [73], developed by Robotics research center Willow Garage, is an open source collection of programs which allows the user to easily control the mobile operation of a robot or a group of robots. It can be compared with a normal operating system on a personal computer. It takes care of the hardware of the machine, choosing the applications and the programs that have to run an a certain moment. Moreover it provide a user-friendly environment for the execution of the application programs. In the same way, ROS, provide at the user an easy method to control the mobile operations of a robotic system letting possible more hardware abstraction, low-level device control, implementation of commonly-used functionality, message-passing between processes, and package management. It also provides tools and libraries for obtaining, building, writing, and running code across multiple computers. ROS basically provide an easy way to write code in a common and more abstract languages, like C++ or Python.

For this reasons the development environment ROS is usually used to create a system to control a robots in a more abstracted way. Then, using the same code, it is possible to create a framework to control a groups of robots neglecting the low level structure. Indeed this system is the base of various experimental platform. Because, when the robot is interfaced with ROS, is very easy to control it and develop other system that use the robot as actuator of a more global system. An example is TeleKyb that we will describe in the next section.

ROS is designed as a peer-to-peer network composed of processes that are processing data together. A process, called *node*, is the system that perform computation. Usually a robot control system comprises many nodes. For example, one node controls a laser range-finder, one node controls the wheel motors, one node performs localization, one node performs path planning, one node provides a graphical view of the system, and so on. This is the principle of a modular system, where the task is not performed by a single big program, but is carried out by a set of smaller process-specific programs working together shearing data and other variables.

The nodes exchange data together using *messages* which is simply a data structure, comprising typed fields. To communicate messages to each other they use *topics* which can be represented as the link where the message flows. Then there are nodes, called *publisher*, which send out a message by publishing it to a given topic; and nodes, called *subscriber*, which "listen" the data from a given topic. Although this two types of node can work on the same topic, actually they are completely independent. Moreover there may be multiple concurrent publishers and subscribers for a single topic, and a single node may publish and/or subscribe to multiple topics. Logically, we can think of a topic as a strongly typed message bus. Each bus has a name, and anyone can connect to the bus to send or receive messages as long as they are the right type. The Fig. 6.4 is represented the method of communication between nodes.

This method allows to neglect the synchronization of the nodes, because publishers and subscribers are independent. For this reasons ROS is used to control single robot, for instance having a node for each sensor and actuators,

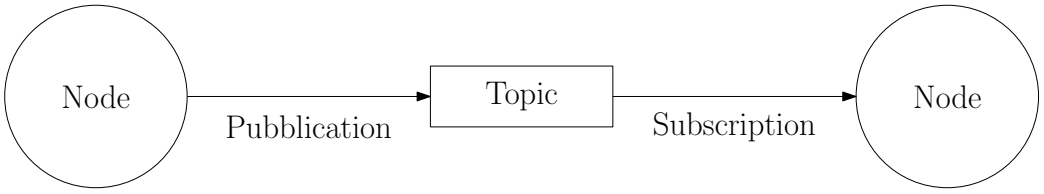


Figure 6.4: The schematic represents the method of communication between nodes.

but also for big project, such as the control of a system of robots. Indeed this method allows to abstract each robot with a single node, making easer the control of the entire set. This is for example the case of TeleKyb.

6.3 TeleKyb

The open source *Tele-Operation Platform of the MPI for Biological Cybernetics (TeleKyb)* [74] is an extensible high-level controller for mobile robots. Nowadays, the robotic platforms need to be as modular as possible in order to increase the ability to reuse existing code or to extend and improve the existing framework. Thus, the purpose of TeleKyb is to create a modular control framework for generic quadrotors UAVs based on ROS. This provides a standardized interface for developing and testing bilateral teleoperation systems between human interfaces (e.g., haptic devices or touch screens) and groups of mobile robots, such as UAVs.

TeleKyb includes three main layers: *human interface* for the interaction with the operator, *control* for the motion planning and the actuation of the quadrotor, and the *hardware interface* for interfacing with the particular quadrotor hardware. In order to use the Crazyflie to conduct the experiments with this framework, we had to extend the hardware interface layer for our specific hardware allowing TeleKyb to control directly the quadrotor. In the following, we describe the main components of TeleKyb.

A Human Interface

This layer is developed to insert the human input in the controller. In this way is possible to create semi autonomous application where the human operates as high level controller, while the machine deal with the low level control, such as in [44] [75] [76]. Moreover, *TeleKyb Haptics* provide a unique interface for the common haptic devices, i.e. human input devices which provide force feedback to the user. With this general interface is possible to design haptic control algorithm using standardize methods. Finally there is also an interface for the common joysticks and gamepads that don't provide haptic feedback.

B TeleKyb Core

The *TeleKyb Core* is the heart of the framework. It is a high-level closed loop robotic controller composed by three elements: the *State Estimator* estimates the position of the robot, *TeleKyb Behavior* compute the next position and velocity, and the *TeleKyb Tracker* compute the next commands which are then sent to the robot.

In particular the *State Estimator* generates a standard state message $(p, \dot{p}, \eta, \Omega)$ where $p, \dot{p}, \eta, \Omega$ are respectively the position, velocity, orientation and angular velocity of the robot. Since it is possible to change the state estimator, even runtime, estimators based on different sensors can be used. At the moment, the most used system to estimate the position of the vehicle is based on the measures of the VICON[1]. The latter is a motion capture system able to record the motion of objects. It is composed by a set of image sensor, usually infrared cameras, used to triangulate the 3-D position of a subject. To acquire the data is traditionally used some retroreflective markers attached to the object. So the system is able to provide the 3-D position of each marker. Then, using at least three markers attached to the object, using the relative distance from each others, is possible to compute also its orientation. Then, the knowledge of the linear and angular velocity is provided with a numerical derivation from the pose and orientation.

[1] http://vicon.com/

Then, at each instant that the *State Estimator* provides the new state, the active behavior generate the new trajectory. The default behavior, already included in TeleKyb are: *Take-Off*, *Hover*, *Fly-To*, *Trajectory Following*, *Human Control* and *Land*. A sequence of these behavior lets to design a complete experiment, such as: *Take-Off* $\rightarrow$ *Hover* $\rightarrow$ *Trajectory Following* $\rightarrow$ *Hover* $\rightarrow$ *Land*. After that the *Trajectory Behavior* computed the new trajectory, the *Trajectory Tracker* computes the needed robot commands from the current state and trajectory message. Recalling the quadrotor controller described in Sec. 2.6, based on the decoupling between the attitude and linear dynamics, the *Trajectory Tracker* runs the slower position controller (Sec. 2.6.3) sending the attitude set point $\eta^*(t)$ and the desired total thrust f^*, so as to attain the desired translational motion. Whereas, the faster attitude control is implemented directly on board.

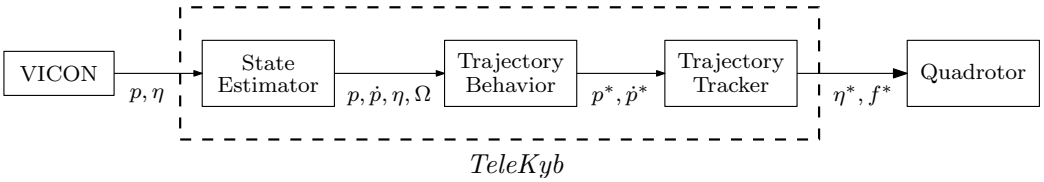


Figure 6.5: Schematic of the *TeleKyb* controller.

C Experimental Flow Manager

With the *Experimental Flow Manager* is possible to design an experiment project as a finite state machine. With it, we can control the execution of the experiment consisting of one ore more *TeleKyb Core*, one for each vehicle. With the *Experimental Flow Manager* it is possible to specify the various behavior and when they need to be activated.

D Hardware Interface

The *Hardware Interface* allows TeleKyb to interact directly with the quadrotor. In particular this interface has to send the commands, (η^*, f^*), and read data form it, such as the measurements of the IMU, or the level of battery.

Basically it is the connection between the software controller running on the personal computer and the real robot. Since the global system is based on ROS this become quite easy because the *Hardware Interface* has only to read the commands from the *TeleKyb Core* in a specific topic and than to manage the physical communications with the vehicle. For the data coming form the robot it is similar, the interface, when a new data value arrives, publishes it in a determinate topic. ROS allows to implement this interface completely independent from the rest of the code.

6.4 Crazyflie ROS Interface

Remembering the high lever controller TeleKyb, it basically run the position control described in the Sec. 2.6.3, taking the information of position and orientation from the VICON. Then it compute the quadrotor's commands, in terms of desired attitude and desired thrust (η^*, f^*), in order to track a desired trajectory, and then publishes them in to a specific topic. Thus, the main tasks of Crazyflie Interface (`tk_cfinterface`) are to read the commands from the *TeleKyb Core* and to forward them over a wireless link to the remote UAV, using the radio dongle. A simplified schematic of the system is represented in the Fig. 6.6.

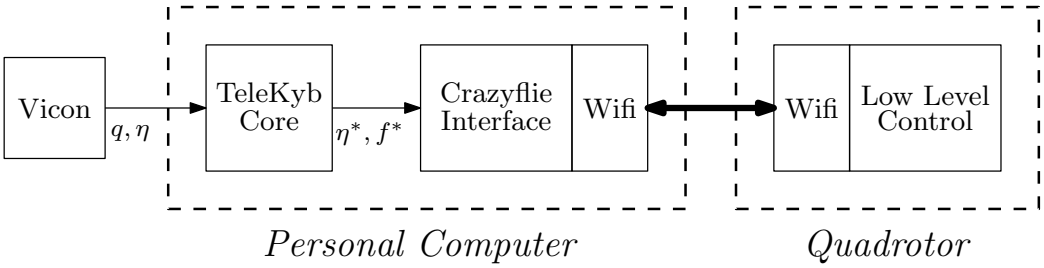


Figure 6.6: Configuration of the global working system.

Basically the Crazyflie Interface represents a transparent interface to the actual robot hardware. Moreover, in this way, an instance of the Core can't distinguish if it is running on a simulator backend or the real UAV. A fact that drastically simplifies the transition from simulation to the utilization of the real quadrotors.

Since the library for the Crazyflie is written in Python we did the same for the ROS interface. We designed the interface as three main classes: *CFInterface*, *CFROSInterface* and *Callbacks*.

- The *CFInterface* is the main of the interface, indeed it instantiates the other two. Its tasks are essentially to create and remove the wireless link between the computer and the quadrotor identified by the number of the channel and the frequency of transmission. Moreover, it provides a method, `send_commands`, to send the commands to the robot.

- The *CFROSInterface* deals with the publication and subscription to the various topics. At first, when it is initialized, it crates a ROS node, "*tk_ cfinterface*" which has to subscribe to the topic where to read the commands from the *TeleKyb Core*, and to publish the data coming from the quadrotor. In particular we implemented the subscriber in the way that when a new commands is read, the method `send_commands` of the class *CFInterface* is called to directly send the commands to the quadrotor. Whereas, about the publishing of the data, we designed different messages on different topics for the measures coming from the IMU, the barometer, the magnetometer and the battery. Then we implemented the methods to publish the data on the respective topic, for example `publish_IMU`.

- The *Callbacks* deals with the receiving of the log data packages from the quadrotor. In particular it provides methods for setting up logging configurations that are used for logging variables from the firmware. In other words we are able to set the quadrotor in order to receive some desired data at a desired frequency. Then, to publish immediately the incoming data, we used a system based on callbacks that allows, when a log package arrives, to automatically call the methods provided by the *CFROSInterface* class to forward the data on the corresponding topic.

The structure of the Crazyflie interface is schematically represented in the Fig. 6.7. By this image is easy to understand the connections between the

classes and their tasks.

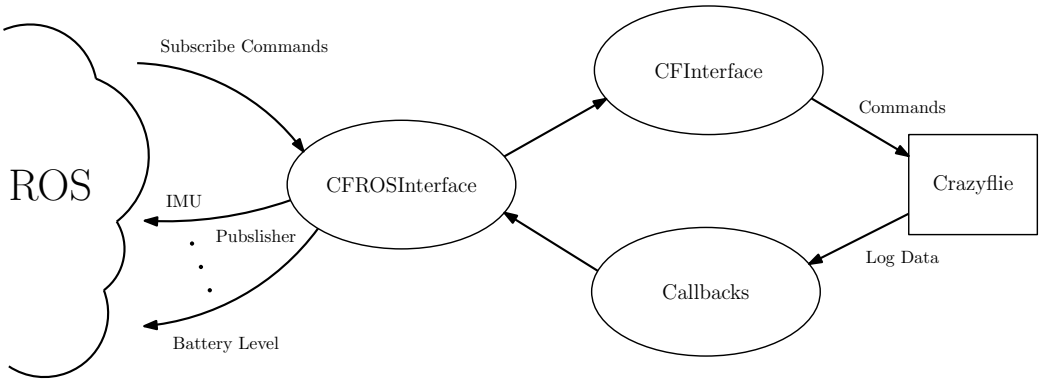


Figure 6.7: Schematic structure of the Crazyflie interface.

6.5 Experiment Setup

To do the experiments for validate the controller, some modifications of the quadrotor are necessary. In particular we have to make sure that the VICON is able to track the quadrotor, providing its position. Moreover, we need a base that works as an anchor, tethered on that the quadrotors.

6.5.1 Crazyflie Setpu

In order to drive the Crazyflie with the TeleKyb framework, it is necessary to make the Crazyflie recognisable by the motion capture system (VICON). As we said before, the VICON is able to provide the position of reflecting balls. So, to obtain the position and orientation of the robot we have to attach to it at least three markers. Therefore, since the Crazyflie is really tiny and has a payload of only 10 [g], the markers available on the market are not feasible. Indeed they are to big and to heavy for the dimension of the robot. To resolve the problem, obtaining little and slight markers, we did some hand made markers using little polystyrene balls covered by a reflecting material. In the Fig. 6.8, the materials and the final resulting markers are showed.

In this way we were able to produce markers with a weight less than 1 [g], that are feasible with the payload of the Crazyflie. However the little

Figure 6.8: Hand made markers. On the left a normal marker used for the MikroKopter quadrotor. On the right the hand made markers with a diameter about 7-8 [mm]. On the bottom of the figure there are the materials used for build the reflecting balls.

dimension of the robot implies another problem: the motion tracking system is not able to distinguish markers distant less than 3 [cm] to each other. Thus is difficult to pose the markers on the robots frames letting the VICON to recognize and to well track each ball. This is complicate also by the reduced dimension of the hand made markers. The solution was to add some tiny sticks, made on carbon fibre, attached above the motors. In this way we could glue three balls on these, increasing the distances between each marker, and letting the VICON to provide the position and orientation of the body frame associated to the robot. On the other way, this solution changes the the center of mass of the system and also the inertial matrix. Indeed, after the implementation of the solution, a calibration was necessary, trying to reposition the center of mass in the geometric center. But also the agility of the robot became worst. Thus we can control it with an high level controller but we are loosing performances.

Moreover, since we have to use two quadrotors on the same times, to distinguish them, the position of the balls must be different. For this reason we added a vertical stick characterized by different length and inclination,

with on top a marker. The Fig. 6.9 shows the modification made on one Crazyflie.

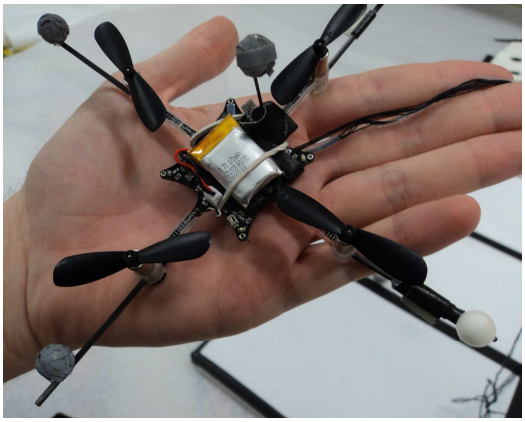
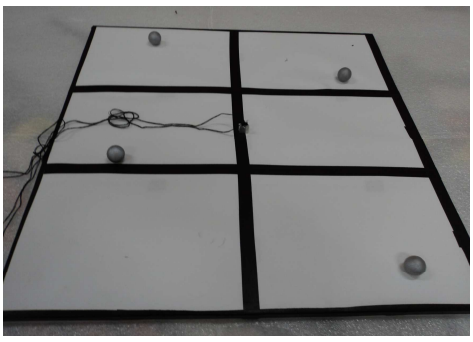


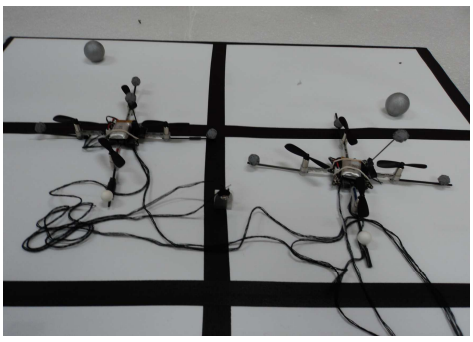
Figure 6.9: Modified Crazyflie with additional carbon fibre sticks in order to attach the markers with a proper distance.

For the implementation of the experiment, we need to tether two cables to the base of the quadrotor as much as possible near to the center of mass, in order to avoid external torques acting on the body. To do this, we designed a sort of harness, made by an iron cable, with a little ring on the center that allows to tie the cables.

6.5.2 Anchor



(a)



(b)

Figure 6.10: (a): Anchor base. (b): Two Crazyflies tethered to the anchor.

For make the experiments we need a fixed point where to tether the two quadrotor. For this purpose we used a plastic plate where we attached a kind of ring for tether the two cables connected to the quadrotors. In order to make it fixed, we made it with a weigh grater than 5 [kg]. In this way, it is too heavy for the two quadrotors and so it is practically fixed. Moreover we screwed four markers on it to obtain it exactly position. As we will explain later this is necessary to measure the orientation of the formation and to fix the quadrotors in the vertical plane passing to the anchor point.

Chapter 7

Experiments

Introduction

In this chapter we will present how we designed the experiments to validate the controller. In particular we will describe every phases that compose the experiment. Indeed, since we assumed taut cables, before to run the controller, it is necessary that all the ropes are taut. For this purpose we divided the initial part of the experiment into three phases with the goal of stretch every cables. Afterwards, the designed controller can be applied in order to track a desired trajectory, keeping the cables taut. As in the simulation chapter (Chap. 5), we will show the results obtained with the racking of a ramp in the desired orientation, keeping a desired constant tension on the third rope.

7.1 Design of the Experiment

For the following experiments we decided to impose a formation equal to a equilateral triangle, thus we took three cables with the same length $l_i = 1[m] \, \forall \, i = 1, 2, 3$, and we tethered the quadrotors.

Recalling the controller in the Chap. 4, and in particular the equations (4.1), (4.5) and (4.11), the measures of the orientation of the formation ψ, and its derivatives, angular velocity $\dot{\psi}$ and acceleration $\ddot{\psi}$ are necessary. In

particular the orientation serves to calculate the desired pitch, while the orientation and angular velocity need to determine the control inputs u_1, u_2. To measure the orientation we use the VICON obtaining the position of the two quadrotors and of the anchor. By this measures we compute the actual orientation of the formation, ψ. Then the angular velocity is simply obtained by a numerical derivation. Whereas, for the angular acceleration, needed to calculate the control input u_2, which determines the tension on the third rope, we use the desired one that comes form the PD controller.

In the Chap. 3 we assumed a system restricted on a two dimensional vertical plane passing to the anchored point. To ensure this condition we use the position control, provided by TeleKyb, in order to keep the y coordinate constant and equal to that of the anchor. In this way, recalling the body fixed frame of the two quadrotors with respect to the world frame, represented in the Fig. 3.4 at page 41, utilizing only the desired roll provided by TeleKyb, is possible to keep constant the y coordinate. Moreover, to avoid rotation of the quadrotor along its z axis, we use also the yaw control provided by TeleKyb. In this way, we ensure that the assumption of planar motion is respected and also the yaw orientation of the quadrotor is kept constant. Thus the yaw and roll angle are controlled by the position controller TeleKyb, while, the desired pitch angle and the thrust amount of the two quadrotors are provided by our controller. Summarizing, the global controller of the two quadrotors is a combination of the position control TeleKyb and the orientation and tension controller.

Another assumption that we did for deriving the model and afterwards the controller, is that every cables are taut. Thus, before to apply our controller, is necessary to reach a minimum tension in all the rope. To reach this state in a proper way, and to verify our controller tracking a ramp profile in the desired orientation, we designed the experiments as a finite state automata characterized by seven states. Each state can be activate only if the previous is active, except for the sixth and seventh state that can be activated at any time. Indeed they correspond to *safe position* and *land* that can be always activated for safe reasons. This automata and also the controller are implemented in Simulink that can read the pushed bottom from a joystick

reading the streaming from a ROS topic. In this way, the user can decide the transitions of state of the experiments pushing a specific bottom of the joystick. Then, depending on the actual state, the Simulink blocks merges the commands from TeleKyb and from our controller and publishes them into a specific ROS topic. The latter is then read by the *CFInterface*, and the commands are physically sent to the quadrotor. In the Fig. 7.1 is represented the relation between the two controller and the physical quadrotor.

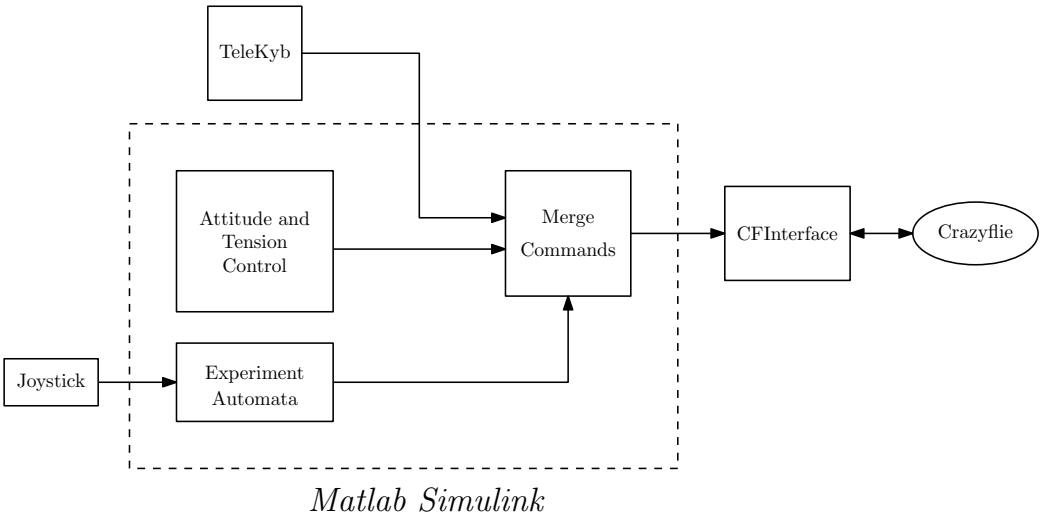


Figure 7.1: Schematic of the software setup for the experiment. In particular this graphic shows that the control commands form the quadrotor are obtained from the merge of the commands provided by TeleKyb and our controller, depending on the actual state decided by the user using a joystick.

The list of all the states that compose the phases of the experiment are:

- **State 0 -** *Ground:* This is the real first state when we start the experiment with the two quadrotor on the ground;
- **State 1 -** *Take Off:* From the starting position, the quadrotors take off vertically to the height of 30 [cm]. In this state, and also in the next one, the control is only provided by the position controller TeleKyb;
- **State 2 -** *Anchored cables tension:* In this sate the goal is to reach the tension of the two anchored cables. To do this we thought to track a desired trajectory equal to a radial segment with the last point over

the maximum length of the cables. In this way the tethered quadrotor, trying to reach that position, will stretch the cable. In order to don't have to much oscillation reaching the tension, we setted a low velocity of the trajectory. Moreover, since we want to stretch only the two anchored cable, the angle between the two trajectory segments has to be little such as the third cable would be slack. In the Fig. 7.2 is showed the movement of the two quadrotors.

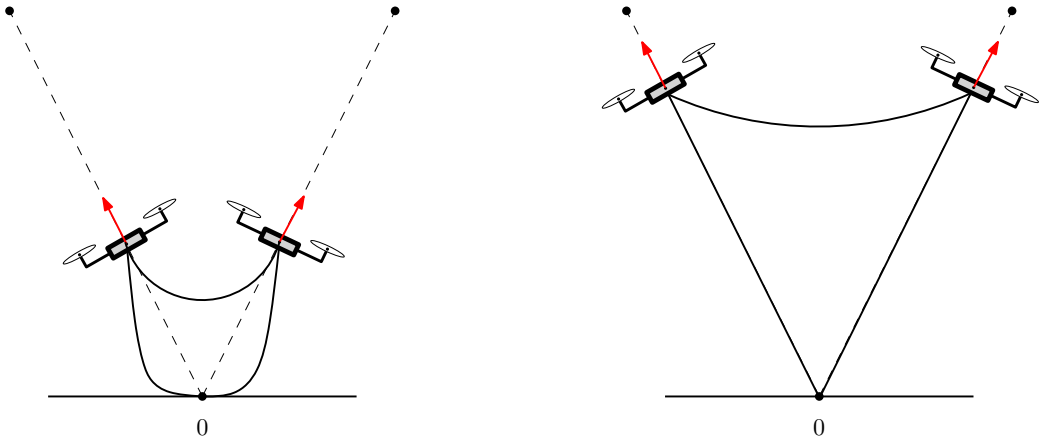


Figure 7.2: Schematic image of the motion of the two quadrotors to reach the tension of two anchored cables. The dotted lines are the desired trajectory. On the left, the initial part of the motion. Until the cables are slack, the vehicles are in free flight and can easily follow the trajectory. While, on the right of the figure, when the distances between the quadrotors and anchor points are equal of the length of the cables, trying to reach the final position (the black dot) the vehicles stretch the cables making them taut.

- **State 3 -** *Third cable tension:* When the two anchored cables are taut, remain the third cable to make it taut. For this purpose, we simply enlarge the angle between the two anchored cable as long as the two quadrotors stretch also the cable between them. To do this, we implemented the controller derived in the Chap. 4 but for a single quadrotor. Indeed, considering the quadrotor one, if we impose $m_2 = 0$ we obtain the model of the single tethered quadrotor and the relative controller to control the angle of the anchored cable with respect to the horizon. In this way, looking at the single quadrotors, forgetting that the two

vehicles are tethered to each other, we can control independently the angle ψ_1 and ψ_2, described in the Fig. 7.3. Thus, we can use this controller to pass to the actual angle α, to the nominal value that comes from the length of the cables. Increasing the angle alpha we are able to stretch also the third cable making it taut;

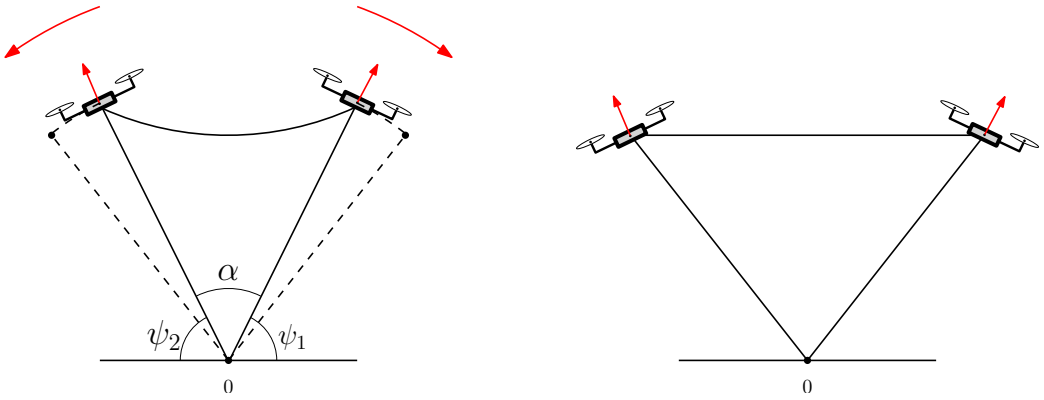


Figure 7.3: Schematic image of the motion of the two quadrotor to reach the tension of the third cable. The dotted lines are the desired trajectory and the black dot is the final quadrotors position. On the left the initial part of the motion. The vehicles have to follow an arc of circle, distancing from each other, until the middle rope becomes taut. The figure on the right represents the final situation.

- **State 4 -** *Attitude and tension control:* When all the cables are taut, we can apply our attitude and tension controller (Chap. 4) to keep constant the actual orientation of the formation applying the desired tension t_3^* on the third cable;

- **State 5 -** *Tracking of a trajectory:* When the formation is stable, with all the cable taut, we can verify our controller for the tracking of a desired trajectory of the formation's orientation, $\psi^*(t)$;

- **State 6 -** *Safe position:* In case of emergency or any unexpected events, we defined a safe position where the quadrotors have to fly in order to avoid crashes or undesired behaviour. This state can be activated in any time, and when it is active, the position controller TeleKyb drives

the quadrotor to the relative safe positions. This position is near the ground and ensures slack cables;

- **State 7 -** *Land:* The quadrotors simply land to the ground.

7.2 Results

As mentioned before, to verify the controller, we imposed a formation represented by an equilateral triangle. Thus we used cables of the same length $l_i = 1[m] \, \forall \, i = 1, 2, 3$. Then we setted the initial formation's orientation $\psi_0^* = 60°$. We decided this setting in order to start with a symmetric condition, that implies a equal configuration for both the quadrotors. Then we designed the desired trajectory as a ramp with initial orientation $\psi_0^* = 60°$ and final orientation $\psi_f^* = 52°$, whit a constant velocity equal to $\frac{\Delta\psi^*}{T} = \frac{8}{2} = 4$ [deg/s]. We decided to set a $\Delta\psi^*$ of only 8°, because, with a lower final desired angle ψ_f^*, there is the risk that the propeller of the quadrotor one touches the cable. Moreover we fixed a desired tension on the third cable equal to $t_3^* = 0.02$ [N].

Regarding the controller we setted a constant thrust of the two Crazyflie equal to $f^* = 0.25$ [N] and the same gains determined in the simulation tuning (Chap. 5). We also did a manual tuning of them, but the best configuration remained the initial one. In the Fig. 7.4 and Fig. 7.5 are reported the experimental results during the tracking of the desired ramp in the formation orientation.

Starting looking the the last figure on the left, we can observe the dynamics of the real orientation of the formation with respect the desired one. At first we can notice that the orientation doesn't star exactly from the desired position and also, at the end, in steady state, the angle ψ is not perfectly constant but shows some oscillation, whit maximum amplitude of 2°, around the desired value. While, during the motion, the tacking is quite good, with a little delay and with the same oscillation in the steady state. Also in the simulations we noticed a little delay between the desired and the actuated orientation angle. Thus, although in the experiment the delay is bigger, it is justify because the system can not apply instantaneous acceleration required

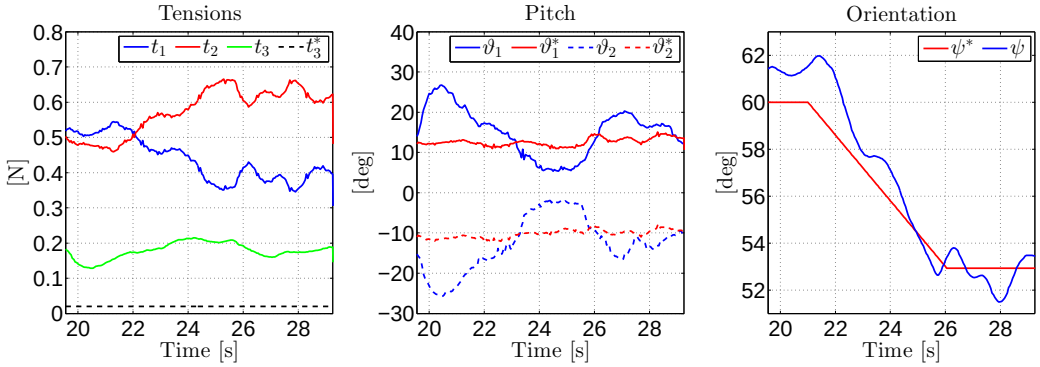
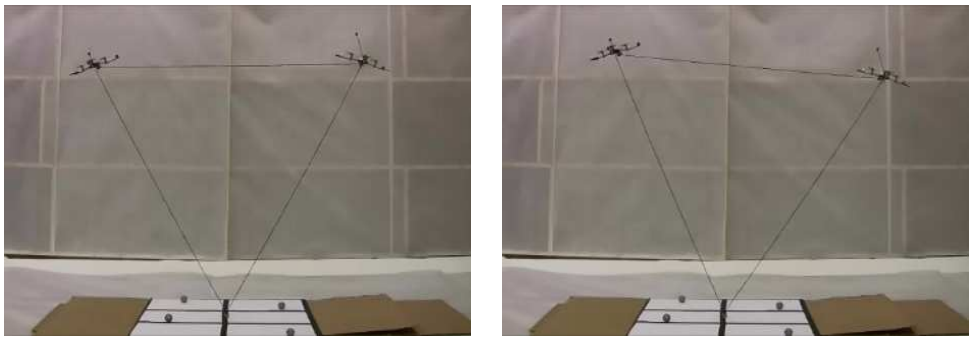


Figure 7.4: Experiment results of the tracking of the desired ramp in the formation orientation. Starting from the left there is the estimated tension on the cables, then the comparison between the desired (red line) and actuated (blue line) pitch angles, and final the comparison between the desired trajectory (red line) and the actual one (blue line).



(a) Initial orientation: $\psi = 60°$.

(b) Final orientation: $\psi = 52°$.

Figure 7.5: Experiment's images for the tracking of the ramp in the formation's orientation.

from the ramp. Indeed, to pass from zero angular velocity to a constant value, it needs an infinite impulse in the angular acceleration that is not feasible for the system.

Then, in the first graphics on the left, we plotted the estimated tension on the cables. Was impossible to add some sensors to measure directly the tension on the cable. Thus we implemented an easy estimator, used after the experiment, to have a qualitative measure of the internal tension of the ropes. For this purpose we can use the equations (3.22) at page 52. Indeed,

after the experiments, we have all the data from the VICON to compute the equations, moreover, we can apply a non-causal filter to obtain better values of the angular velocity and acceleration of the formation's angle, $\dot{\psi}, \ddot{\psi}$, than the standard numerical derivation, which can introduce a lot of noise. Looking at the comparison between the desired tension on the third cable and the estimated, we can notice that the estimated is almost constant but with a big offset. This is due to the errors between the desired pitch and the actuated. In the middle figure we can see that the actuated pitch angles, ϑ_i, have a symmetric error between the commanded one, ϑ_i^*. This is due to the fact that the assumption of cables attached in the center of mass of the vehicle is not true. Indeed, with the actual mechanic configuration of the Crazyflie, there is a little displacement between the center of gravity of the system and the point where the cables are tethered. This displacements implies that, when the cables are taut, the internal tension causes a torque on the quadrotor frame. This external torque, considered as an external disturbance, doesn't allow the attitude controller of the quadrotor to attain perfectly the desired pitch angle. Thus, since the quadrotors tilt more than the required, the tension on the third cable increases. Nevertheless, the formation orientation doesn't show big problems due to these differences between the actuated and desired pitch. This because, as we can see in the middle graphics of the Fig. 7.4, the errors are symmetric on the two quadrotors. This means that the additional torques acting on the formation, due to the over tilting, are balanced by themselves. Indeed since they are almost equal and opposite with respect the fixed point, the total torque applied to the whole system is almost zero. Thus the tracking of the desired trajectory of the formation's orientation is still good, and the over tilting cause only an additional offset in the desired tension of the third cable.

Another problem that can cause the oscillation in the formation's orientation angles, also in steady state, is the position control which is not very precise. We recall that our model and controller is restricted on a vertical plan, and we use the position controller TeleKyb to keep the quadrotors in this plane. Though, probably due to the little dimension of the quadrotor and of the hand made markers, the vehicles oscillate around the desired y

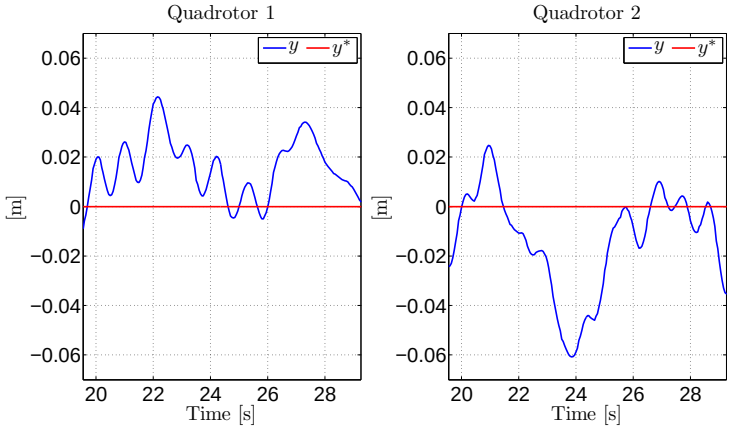


Figure 7.6: Comparison between the desired and the actual coordinate y for the two quadrotor. On the left the position of the first quadrotor, on the right the second. The red line is the desired y position and the blue one the real.

position with an amplitude of $3 - 6$ [cm] as we can see in the Fig. 7.6.

In the end, we setted the desired thrust to $f^* = 0.25$ [N], but in practice we don't know the exactly force that the quadrotor is providing, because is difficult to map the commanded thrust to the real one. Moreover, due to the drain of the battery, the real thrust decreases during the execution of the experiment. This not precise variable, such as also the mass of the vehicles and the length of the cables during the stretching, may introduce uncertainty on the controller that produce the errors between the desired and actuated variables, like formation orientation and internal tension in the third rope.

Chapter 8

Conclusions

8.1 Summary

Starting from the analysis of the system, in Chap. 3 we found the equation that describes the dynamics of the system. Considering the formation orientation as the state and the pitch angle of the two quadrotor as inputs, every quantities appear in the equation as arguments of trigonometric functions. This means that the system is non linear with respect the state and also the inputs. Using the feedback linearization technique, in Chap. 4, we wrote the control inputs as functions of the state and of two additional virtual control inputs. To apply this technique, we assumed that we are able to directly control the attitude of the quadrotor. This is not true because there is another controller to attain the desired orientation of the vehicle. But, as we verified in the simulation, and also in the experiments, the dynamics of the quadrotor is faster than the one of the system, thus the assumption is adequate. With the designed functions, we were able to linearize the system, and to directly control the angular acceleration of the formation's orientation with one of the two virtual inputs. Then, in order to control the formation's orientation and to track a desired trajectory, we introduced a classical PD controller to obtain the desired angular acceleration. The second control input was used to control the tension on the third cable. Since it was not possible to use some sensors to measure the internal tension on the cables, this last controller is

only a open loop control, that comes from the equation of the internal tension as function of the state and the control inputs. Although this method is not precise, actually, our main purpose is to ensure taut cables.

After the analysis of the system and the design of the controller, in the Chap. 5 we presented the simulation results. We showed that, at least in a ideal world, without disturbances, the controller works very well. It is able to keep constant the tension on the third cable to the desired value, while tracking a trajectory of the formation's orientation. Moreover, we tuned the controller's gains in order to have a smooth response, such that to ensure taut cables.

Then, to test our controller also with real experiments, we started to develop a testbed, based on a new type of nano quadrotor: Crazyflie. In particular we would to exploit the high level controller for quadrotors, that already exists in our laboratory (TeleKyb), but using the new mini quadrotor. To do this, we implemented a interface, written in Python, that links the framework TeleKyb with the hardware of the quadrotor. Using this system, based on ROS (Robot Operating System), we were able to control the quadrotor directly in position, using many type of controller to drive it, from the classical joystick to a Matlab program. Indeed, thanks to this interface, and using the system of messages of ROS, it is possible to send the command to the quadrotor directly from Matlab.

Thus we implemented our controller on Simulink and we designed the experiment to test the system. Looking at the results, we noticed that, thanks to the controller, the system is able to track quite well a desired trajectory of the formation's orientation, keeping all the cables taut. Although the tracking error is not so high, there are some oscillations around the desired angle, probably due to the position control that is not very precise. Indeed, to keep the quadrotor on a vertical plane we used the position controller TeleKyb that uses a motion tracking system to obtain the position feedback. But, due to the little size of the quadrotor and of the markers, there are some oscillations around the desired position that disturbs the rotation of the formation around the anchor point. Regarding the control of the tension on the third cable, we noticed that the obtained value is almost constant but with

an offset with respect the desired value. This in due to the fact that the taut cables, since they aren't exactly tethered on the center of mass of the quadrotors, they apply torques on the body of the vehicles. This external torques acting on the quadrotors cause errors between the desired and actuated pitch. Although this errors don't cause big problem in the orientation control, actually they increase the tension in the third cable. Finally, the last source of problems consists in the uncertainty of the model. Indeed the model is only an approximation of the real system, for instance we didn't considered the elasticity of the cables. Then there are some quantities, that we considered constant and know, but that in practice we don't know exactly. For example the thrust provided by the quadrotor is not perfectly equal to the desired values, moreover it decreases with the drain of the battery. Thus, although in the real experiments we found a lot of problems deriving from the non ideality of the system, actually the controller is quite robust to them and the formation is able to track quite well a desired trajectory of the orientation, keeping all the cables taut.

8.2 Future Works

The first objective after the thesis will be to refine the system. In particular we have at first to solve the hardware problem explained before. Indeed we want to design a better craft for the quadrotor in order to obtain a position control more precise. Moreover, always for this purpose we will do a better and precise tuning of all the gains of the controllers. While, regarding the problem of the external torques caused by the ropes attached not in the center of gravity we want at first to develop a mechanical solution to reduce this effects. Then the second step will be to consider this torques as external disturbances and trying to reject or compensate them. Finally, to have a good knowledge of the real thrust provided by the quadrotor, we will map the relation between the desired and actuated thrust, considering also the drain of the battery.

After solving these problems linked to the mechanical design of the system, we will try to increase the performances of the controller also from a

theoretical point of view. At first we want to control not the attitude of the quadrotor, but the angular rate of its attitude. In this way, the assumption to control directly the orientation of the vehicle is not necessary. This change will also solve some of the problems linked to the external torques. Moreover we will use also the thrust as input to improve the global performances.

Afterwards there are many open problem linked to this system. At first we assumed a system restricted to a vertical two dimensional plane. The next natural step is to analyze the same problem but in a full three dimensional space. Then, another interesting problem, related to real application such as the transportation of loads, is to consider not a formation tethered to a fixed point, but to an object that can move. At the end, the most challenging aspect is to extends the problem in a general multirobot formation. Where a set of robots, more than two, of different types, are tethered together an have to keep a constant formation. The problem is to find the rules to implement the formation and design a distributed controller to stabilize the whole system.

Bibliography

[1] C. Sultan, "Modeling, design and control of tensegrity structures with applications." Ph.D. dissertation, Purdue University, School of Aeronautics and Astronautics, West Lafayette, 1999.

[2] S. Oh, K. K. Mankala, S. K. Agrawal, and J. S. Albus, "Dynamic modeling and robust controller design of a two-stage parallel cable robot," *Multibody System Dynamics*, vol. 13, no. 4, pp. 385–399, 2005.

[3] I. Palunko, P. Cruz, and R. Fierro, "Agile load transportation: Safe and efficient load manipulation with aerial robots," *Robotics & Automation Magazine, IEEE*, vol. 19, no. 3, pp. 69–79, 2012.

[4] K. Sreenath, N. Michael, and V. Kumar, "Trajectory generation and control of a quadrotor with a cable-suspended load - a differentially-flat hybrid system," in *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, May 2013, pp. 4888–4895.

[5] D. Mellinger, Q. Lindsey, M. Shomin, and V. Kumar, "Design, modeling, estimation and control for aerial grasping and manipulation," in *2011 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, San Francisco, CA, Sep. 2011, pp. 2668–2673.

[6] A. Faust, I. Palunko, P. Cruz, R. Fierro, and L. Tapia, "Learning swing-free trajectories for uavs with a suspended load," in *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, May 2013, pp. 4902–4909.

[7] F. A. Goodarzi, D. Lee, and T. Lee, "Geometric stabilization of quadrotor uav with a payload connected by flexible cable," *arXiv preprint arXiv:1309.6717*, 2013.

[8] V. Artale, C. Milazzo, and A. Ricciardello, "Mathematical modeling of hexacopter," *Appl. Math. Sci*, vol. 7, no. 97, pp. 4805–4811, 2013.

[9] C. Eschmann, C. M. Kuo, C. H. Kuo, and C. Boller, "Unmanned aircraft systems for remote building inspection and monitoring," in *Proceedings of the sixth European workshop on structural health monitoring*, 2012.

[10] J. E. Meng, Y. Shenghai, and W. Ning, "Development control and navigation of octocopter," in *Control and Automation (ICCA), 2013 10th IEEE International Conference on*, June 2013, pp. 1639–1643.

[11] K. Sreenath and V. Kumar, "Dynamics, control and planning for cooperative manipulation of payloads suspended by cables for multiple quadrotor robot," *Robotics: Science and Systems*, Jun 2013.

[12] R. Ritz and R. D'Andrea., "Carrying a flexible payload with multiple flying vehicles," in *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*, Nov 2013, pp. 3465–3471.

[13] S. Lupashin and R. D'Andrea, "Stabilization of a flying vehicle on a taut tether using inertial sensing," in *2013 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, Tokyo, Japan, Nov 2013, pp. 2432–2438.

[14] Fotokite. (2014) Fotokite is a flying camera that enables immediate, safe, intuitive access to birds-eye perspectives of the world in front of you. [Online]. Available: http://www.fotokite.com/

[15] M. Turpin, K. Mohta, N. Michael, and V. Kumar, "Goal assignment and trajectory planning for large teams of aerial robots," in *Proc. Robotics: Science and Systems*, 2013.

[16] G. L. Mariottini, F. Morbidi, D. Prattichizzo, N. Vander Valk, N. Michael, G. Pappas, and K. Daniilidis, "Vision-based localization

for leader-follower formation control," *IEEE Trans. on Robotics*, vol. 25, no. 6, pp. 1431–1438, 2009.

[17] J. Chen, D. Sun, J. Yang, and H. Chen, "Leader-follower formation control of multiple non-holonomic mobile robots incorporating a receding-horizon scheme," *The International Journal of Robotics Research*, vol. 29, no. 6, pp. 727–747, 2010.

[18] R. Fierro, A. K. Das, V. Kumar, and J. P. Ostrowski, "Hybrid control of formations of robots," in *Robotics and Automation, 2001. Proceedings 2001 ICRA. IEEE International Conference on*, vol. 1, 2001, pp. 157–162.

[19] N. E. Leonard and E. Fiorelli, "Virtual leaders, artificial potentials and coordinated control of groups," in *40th IEEE Conf. on Decision and Control*, Orlando, FL, Dec. 2001, pp. 2968–2973.

[20] M. R. Pac, A. M. Erkmen, and I. Erkmen, "Control of robotic swarm behaviors based on smoothed particle hydrodynamics," in *2007 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, San Diego, CA, Nov. 2007, pp. 4194–4200.

[21] L. C. A. Pimenta, N. Michael, R. C. Mesquita, G. A. S. Pereira, and V. Kumar, "Control of swarms based on hydrodynamic models," in *2008 IEEE Int. Conf. on Robotics and Automation*, Pasadena, CA, May 2008, pp. 1948–1953.

[22] T. Nestmeyer, P. Robuffo Giordano, and A. Franchi, "Multi-target simultaneous exploration with continual connectivity," in *2th Int. Work. on Crossing the Reality Gap - From Single to Multi- to Many Robot Systems, at 2013 IEEE Int. Conf. on Robotics and Automation*, Karlsruhe, Germany, May 2013.

[23] P. Robuffo Giordano, A. Franchi, C. Secchi, and H. H. Bülthoff, "A passivity-based decentralized strategy for generalized connectivity maintenance," *The International Journal of Robotics Research*, vol. 32, no. 3, pp. 299–323, 2013.

[24] D. Zelazo, A. Franchi, F. Allgöwer, H. H. Bülthoff, and P. Robuffo Giordano, "Rigidity maintenance control for multi-robot systems," in *2012 Robotics: Science and Systems*, Sydney, Australia, Jul. 2012.

[25] C. Belta and V. Kumar, "Abstraction and control for groups of robots," *IEEE Trans. on Robotics*, vol. 20, no. 5, pp. 865–875, 2004.

[26] A. Franchi, C. Secchi, M. Ryll, H. H. Bülthoff, and P. Robuffo Giordano, "Shared control: Balancing autonomy and human assistance with a group of quadrotor UAVs," *IEEE Robotics & Automation Magazine, Special Issue on Aerial Robotics and the Quadrotor Platform*, vol. 19, no. 3, pp. 57–68, 2012.

[27] B. Jackson, "Notes on the rigidity of graphs," in *Levico Conference Notes*, Levico, Oct. 2007.

[28] B. D. O. Anderson, C. Yu, B. Fidan, and J. M. Hendrickx, "Rigid graph control architectures for autonomous formations," *IEEE Control Systems Magazine*, vol. 28, no. 6, pp. 48–63, 2008.

[29] X. Li and L. Yang, "Design and implementation of uav intelligent aerial photography system," in *Intelligent Human-Machine Systems and Cybernetics (IHMSC), 2012 4th International Conference on*, vol. 2. Nanchang, Jiangxi: IEEE, Aug 2012, pp. 200–203.

[30] A. M. Samad, N. Kamarulzaman, M. A. Hamdani, T. A. Mastor, and K. A. Hashim, "The potential of unmanned aerial vehicle (uav) for civilian and mapping application," in *System Engineering and Technology (ICSET), 2013 IEEE 3rd International Conference on*, Shah Alam, Aug 2013, pp. 313–318.

[31] G. Gioioso, A. Franchi, G. Salvietti, S. Scheggi, and D. Prattichizzo, "The Flying Hand: a formation of uavs for cooperative aerial telemanipulation," in *2014 IEEE Int. Conf. on Robotics and Automation*, Hong Kong, China, May. 2014.

[32] G. Gioioso, M. Ryll, D. Prattichizzo, H. H. Bülthoff, and A. Franchi, "Turning a near-hovering controlled quadrotor into a 3D force effector," in *2014 IEEE Int. Conf. on Robotics and Automation*, Hong Kong, China, May. 2014.

[33] J. G. Leishman, "The breguet-richet quad-rotor helicopter of 1907," *Vertiflite*, vol. 47, no. 3, pp. 58–60, 2002.

[34] S. A. Raza and W. Gueaieb, "Intelligent flight control of an autonomous quadrotor," in *Motion Control*. IN-TECH, 2010, pp. 245–264.

[35] J. Jiang, J. Qi, D. Song, and J. Han, "Control platform design and experiment of a quadrotor," in *Control Conference (CCC), 2013 32nd Chinese*. IEEE, July 2013, pp. 2974–2979.

[36] S. Lupashin, A. Schöllig, M. Hehn, and R. D'Andrea, "The flying machine arena as of 2010," in *2010 IEEE Int. Conf. on Robotics and Automation*, Anchorage, AK, May 2010, pp. 2970–2971.

[37] M. Turpin, N. Michael, and V. Kumar, "Decentralized formation control with variable shapes for aerial robots," in *2012 IEEE Int. Conf. on Robotics and Automation*, St. Paul, MN, May 2012, pp. 23–30.

[38] A. K. Das, R. Fierro, V. Kumar, P. Ostrowski, J. Spletzer, and C. J. Taylor, "A vision-based formation control framework," *IEEE Trans. on Robotics and Automation*, vol. 18, no. 5, pp. 813–825, 2002.

[39] D. Mellinger, M. Shomin, and V. Kumar, "Control of quadrotors for robust perching and landing," in *Proceedings of the International Powered Lift Conference*, Oct 2010.

[40] S. Lupashin, A. Schöllig, M. Sherback, and R. D'Andrea, "A simple learning strategy for high-speed quadrocopter multi-flips," in *2010 IEEE Int. Conf. on Robotics and Automation*, Anchorage, AK, May 2010, pp. 1642–1648.

[41] P. Stegagno, M. Basile, H. H. Bülthoff, and A. Franchi, "A semi-autonomous UAV platform for indoor remote operation with visual and haptic feedback," in *2014 IEEE Int. Conf. on Robotics and Automation*, Hong Kong, China, May. 2014.

[42] P. Stegagno, M. Basile, H. H. Bülthoff, and A. Franchi, "Vision-based autonomous control of a quadrotor UAV using an onboard RGB-D camera and its application to haptic teleoperation," in *2nd IFAC Work. on Research, Education and Development of Unmanned Aerial Systems*, Compiegne, France, Nov. 2013.

[43] C. Masone, P. Robuffo Giordano, H. H. Bülthoff, and A. Franchi, "Semi-autonomous trajectory generation for mobile robots with integral haptic shared control," in *2014 IEEE Int. Conf. on Robotics and Automation*, Hong Kong, China, May. 2014.

[44] D. J. Lee, A. Franchi, H. I. Son, H. H. Bülthoff, and P. Robuffo Giordano, "Semi-autonomous haptic teleoperation control architecture of multiple unmanned aerial vehicles," *IEEE/ASME Trans. on Mechatronics, Focused Section on Aerospace Mechatronics*, vol. 18, no. 4, pp. 1334–1345, 2013.

[45] D. Brescianini, M. Hehn, and R. D'Andrea, "Quadrocopter pole acrobatics," in *2013 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, Nov 2013, pp. 3472–3479.

[46] T. Lee, M. Leokyand, and N. H. McClamroch, "Geometric tracking control of a quadrotor UAV on SE(3)," in *49th IEEE Conf. on Decision and Control*, Atlanta, GA, Dec. 2010, pp. 5420–5425.

[47] R. lab. (2013, Nov) Pars aerial rescue robot. [Online]. Available: http://rtsideas.com/index.php?option=com_k2&view=itemlist&layout=category&task=category&id=1&Itemid=468&lang=en

[48] A. lab. (2014) Aeryon scouot. [Online]. Available: http://www.aeryon.com

[49] R. Mahony, V. Kumar, and P. Corke, "Multirotor Aerial Vehicles: Modeling, Estimation, and Control of Quadrotor," *IEEE Robotics & Automation Magazine*, vol. 19, no. 3, pp. 20–32, 2012.

[50] P. Corke, *Robotics, Vision and Control: Fundamental Algorithms in MATLAB*. Springer, 2011, vol. 73.

[51] T. Hamel, R. Mahony, R. Lozano, and J. Ostrowski, "Dynamic modelling and configuration stabilization for an x4-flyer." *International Federation of Automatic Control Symposium (IFAC)*, 2002.

[52] M. W. Spong, S. Hutchinson, and M. Vidyasagar, *Robot modeling and control*. John Wiley & Sons New York, 2006.

[53] P. Castillo, R. Lozano, and A. Dzul, "Stabilization of a mini rotorcraft with four rotors," *IEEE Control Systems Magazine*, vol. 25, no. 6, pp. 45–55, 2005.

[54] K. P. Valavanis, *Advances in Unmanned Aerial Vehicles: State of the Art and the Road to Autonomy*, ser. Intelligent Systems, Control and Automation: Science and Engineering. Springer, 2007, vol. 33.

[55] P. Pounds, R. Mahony, and P. Corke, "Modelling and control of a large quadrotor robot," *Control Engineering Practice*, vol. 18, no. 7, pp. 691–699, 2010.

[56] S. Bouabdallah, "Design and control of quadrotors with application to autonomous flying," Master's thesis, Lausanne Polytechnic University, France, 2007.

[57] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Journal of basic Engineering*, vol. 82, no. 1, pp. 35–45, 1960.

[58] J. L. Marins, X. Yun, E. R. Bachmann, R. B. McGhee, and M. J. Zyda, "An extended kalman filter for quaternion-based orientation estimation using marg sensors," in *Intelligent Robots and Systems, 2001. Proceedings. 2001 IEEE/RSJ International Conference on*, vol. 4. IEEE, 2001, pp. 2003–2011.

[59] A. M. Sabatini, "Quaternion-based extended kalman filter for determining orientation by inertial and magnetic sensing," *Biomedical Engineering, IEEE Transactions on*, vol. 53, no. 7, pp. 1346–1356, 2006.

[60] A. J. Baerveldt and R. Klang, "A low-cost and low-weight attitude estimation system for an autonomous helicopter," in *Intelligent Engineering Systems, 1997. INES'97. Proceedings., 1997 IEEE International Conference on*, 1997, pp. 391–395.

[61] R. Mahony, T. Hamel, and J.-M. Pflimlin, "Nonlinear complementary filters on the special orthogonal group," *IEEE Trans. on Automatic Control*, vol. 53, no. 5, pp. 1203–1218, 2008.

[62] M. Euston, P. Coote, R. Mahony, J. Kim, and T. Hamel, "A complementary filter for attitude estimation of a fixed-wing uav," in *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on*. IEEE, 2008, pp. 340–345.

[63] Y. B. Wilker, "The quaternion formalism for möbius groups in four or fewer dimensions," *Linear algebra and its applications*, vol. 190, pp. 99–136, 1993.

[64] L. A. Sandino, M. Bejar, K. Kondak, and A. Ollero, "Advances in modeling and control of tethered unmanned helicopters to enhance hovering performance," *Journal of Intelligent & Robotic Systems*, vol. 73, no. 1-4, pp. 3 – 18, 2014.

[65] U. Leuthäusser, "Viscoelastic theory of climbing ropes," vol. 1, 2012. [Online]. Available: http://sigmadewe.com/fileadmin/user_upload/pdf-Dateien/Physics_of_climbing_ropes.pdf

[66] Y. M. de Haan and G. M. Sluimer, "Standard linear solid model for dynamic and time dependent behavior of building materials," *Heron*, vol. 46, 2001.

[67] I. Percival and D. Richards, *Introduction to Dynamics*. Cambridge University Press, 1991.

[68] D. A. Wells, *Theory and problems of Lagrangian dynamics.* Schaum, 1967.

[69] G. Benettin, "Appunti per il corso di fisica matematica," 2011.

[70] A. Isidori, *Nonlinear Control Systems, 3rd edition.* Springer, 1995.

[71] K. Ogata, *Modern Control Engineering Fourth Edition.* Prentice Hall, Upper Saddle River, NJ, 2002.

[72] Btcraze. (2012) Crazyflie nano quadrotor. [Online]. Available: http://www.bitcraze.se/

[73] ROS. (2011) Ros: Robot operating system. [Online]. Available: http://www.ros.org/

[74] V. Grabe, M. Riedel, H. H. Bülthoff, P. Robuffo Giordano, and A. Franchi, "The TeleKyb framework for a modular and extendible ROS-based quadrotor control," in *6th European Conference on Mobile Robots*, Barcelona, Spain, Sep. 2013.

[75] C. Secchi, A. Franchi, H. H. Bülthoff, and P. Robuffo Giordano, "Bilateral control of the degree of connectivity in multiple mobile-robot teleoperation," in *2013 IEEE Int. Conf. on Robotics and Automation*, Karlsruhe, Germany, May 2013.

[76] T. Nestmeyer, M. Riedel, J. Lächele, S. Hartmann, F. Botschen, P. Robuffo Giordano, and A. Franchi, "Interactive demo: Haptic remote control of multiple uavs with autonomous cohesive behavior," in *Int. Work. on Towards Fully Decentralized Multi-Robot Systems: Hardware, Software and Integration, at 2013 IEEE Int. Conf. on Robotics and Automation*, Karlsruhe, Germany, May 2013.