Harrison Neves Marciano

Using Quaternions to Avoid Singularities in UAV-UGV Multi Robot Systems

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Engenharia Elétrica do Centro Tecnológico da Universidade Federal do Espírito Santo, como requisito parcial para obtenção do Grau de Mestre em Engenharia Elétrica, na linha de pesquisa Robótica, Controle e Automação.

Universidade Federal do Espírito Santo – UFES
Centro Tecnológico
Programa de Pós-Graduação em Engenharia Elétrica

Orientador: Prof. Dr. Mário Sarcinelli Filho
Coorientador: Prof. Dr. Alexandre Santos Brandão

Vitória, Espírito Santo
2021

Harrison Neves Marciano

Using Quaternions to Avoid Singularities in UAV-UGV Multi Robot Systems

Vitória, Espírito Santo

2021

Ficha catalográfica disponibilizada pelo Sistema Integrado de
Bibliotecas - SIBI/UFES e elaborada pelo autor

Marciano, Harrison, 1992-
M319u Using Quaternions to Avoid Singularities in UAV-UGV Multi Robot Systems / Harrison Marciano. - 2021.
84 f. : il.

Orientador: Mário Sarcinelli Filho.
Coorientador: Alexandre Santos Brandão.
Dissertação (Mestrado em Engenharia Elétrica) - Universidade Federal do Espírito Santo, Centro Tecnológico.

1. Formação Heterogênea de Robôs Móveis. 2. Descrição da Formação em Quatérnios. 3. Descrição Euclidiana da Formação. 4. Paradigma de Controle Baseado em Estruturas Virtuais. 5. Seguimento de Trajetória. 6. Singularidades de Formação. I. Sarcinelli Filho, Mário. II. Santos Brandão, Alexandre. III. Universidade Federal do Espírito Santo. Centro Tecnológico. IV. Título.

CDU: 621.3

HARRISON NEVES MARCIANO

USING QUATERNIONS TO AVOID SINGULARITIES IN UAV-UGV MULTI ROBOT SYSTEMS

Dissertação submetida ao programa de Pós-Graduação em Engenharia Elétrica do Centro Tecnológico da Universidade Federal do Espírito Santo, como requisito parcial para a obtenção do Grau de Mestre em Engenharia Elétrica.

Aprovada em 25 de agosto de 2021.

COMISSÃO EXAMINADORA

Mário Sarcinelli Filho

Prof. Dr. Mário Sarcinelli Filho
Universidade Federal do Espírito Santo
Orientador

Prof. Dr. Ignacio Agustin Mas
Instituto Tecnológico de Buenos Aires
Consejo Nacional de Investigaciones Científicas y Técnicas - CONICET
Argentina

Lucas Vago Santana

Prof. Dr. Lucas Vago Santana
Instituto Federal de Educação, Ciência e Tecnologia do
Espírito Santo - Campus Linhares

Aos meus familiares. Que eu nunca deixe de valorizá-los.

Agradecimentos

Eu gostaria de agradecer primeiramente à minha família, cujo o apoio foi imprescindível nesta empreitada. À minha mãe, Elizabeth, pela infindável paciência e ao meu pai, Paulo, pelo inesgotável orgulho. Ao meu irmão Pablo, pelas conversas sempre produtivas e a minha irmã Ester, pelas conversas sempre divertidas.

Gostaria de agradecer a meus amigos de LAI, Anthony, Daniel, Diego, Mauro, Valentim e Vinícius que tornaram essa jornada muito mais divertida e interessante. A meus orientadores Mário Sarcinelli Filho e Alexandre Brandão, cuja disposição e disponibilidade são uma grande inspiração.

Também gostaria de agradecer a meu amigo Diego Fagundes, sempre presente nas necessidades. A João Soares e Elizabeth Ramenghi por serem o casal mais "descolado" que um dia eu conheci e a suas filhas Dahyl e Julia, que um dia, eu espero, leiam o trabalho do "dinho" Marciano. Também agradeço a meu amigo de longa data, Thiago Goulart, que apesar dos altos e baixos não desistiu de um amigo. Agradeço a Vinícius Nunes, Gutierry Rocha e Estêvão Carvalho, os quais eu não vejo mais tanto quanto gostaria, mas sempre tornam nossos encontros em grandes eventos.

Agradeço ao VRC e a todos os amigos que pude conhecer através do Rúgbi. Obrigado por experiências que eu nunca pretendi passar e por me darem força para fazer coisas que eu nunca imaginei fazer.

Por fim, agradeço ao CNPq - Conselho Nacional de Desenvolvimento Científico e Tecnológico, agência de fomento do Ministério da Ciência, Tecnologia, Inovação e Comunicações, assim como à FAPES - Fundação de Amparo à Pesquisa e Inovação do Espírito Santo, agência estadual de fomento à pesquisa científica e tecnológica, pelo apoio financeiro que disponibilizou os equipamentos e materiais usados nesta pesquisa. Em particular também agradeço à FAPES a bolsa de estudos que me foi outorgada, a qual me permitiu dedicar-me exclusivamente ao Mestrado.

Harrison Neves Marciano

"You live and learn. At any rate, you live."
Douglas Noel Adams

Resumo

Controlar uma formação de um veículo terrestre não tripulado (VTNT) e um veículo aéreo não tripulado (VANT) com base no paradigma de estruturas virtuais envolve relacionar as variáveis que descrevem a formação, chamadas variáveis da formação, com as posições dos robôs, chamadas variáveis dos robôs. Essa relação gera o que se chama transformações do espaço da formação para o espaço dos robôs, e vice-versa. Uma questão importante que surge ao adotar tal formulação é que pode haver posições relativas dos veículos que caracterizam singularidades de formação. A caracterização de tal formação com base em coordenadas esféricas (distância e orientações relativas) apresenta um ponto de singularidade quando o VANT está diretamente acima do VTNT, preparando-se para pousar nele, por exemplo. Esta Dissertação de Mestrado tem seu foco exatamente nessa situação, tratando do pouso do VANT sobre o VTNT, que emula uma estação base estática ou móvel para o VTNT. Nesse caso, a singularidade é o fato de o VANT não ter a garantia de estar alinhado lateralmente com o VTNT, não permitindo, assim, obter com precisão o estado atual da formação ao iniciar a manobra de pouso. Para lidar com esta questão, esta tese propõe uma representação diferente da formação, livre de singularidades, com base em quatérnios unitários, e um controlador de formação, com base nesta formulação, para guiar a formação em tarefas de seguimento de trajetória. Além do desenvolvimento teórico, o texto também apresenta resultados de experimentos executados, utilizando uma formação composta por um VTNT *Pioneer 3-DX* e um VANT *Bebop 2*, que validam a formulação e o controlador propostos.

Palavras-Chave: Quatérnios; Singularidade de Formação; Seguimento de Trajetória; Paradigma de Estrutura Virtual; Formação Heterogênea de Robôs Móveis.

Abstract

To control a formation composed by a ground unmanned vehicle (UGV) and an unmanned aerial vehicle (UAV) based on the virtual structure paradigm, the focus of this thesis, involves to relate the variables describing the formation, referred to as formation variables, to the positions of the robots, referred to as robots variables. Such relationship generates transformations from the formation space to the robots space, and vice-versa. An important issue that arises when adopting such a formulation is that there may be relative positions of the vehicles which characterize formation singularities. The characterization of such a formation based on spherical coordinates (distance and relative orientations) presents a singularity point when the UAV is just above the UGV, preparing to land on it, for instance. This thesis focuses exactly in such situation, dealing with the landing of UAV on the UGV, which emulates a static or moving base station for the UAV. In such case, the singularity is the fact that the UAV is not guaranteed to be laterally aligned with the UGV, thus not allowing to precisely get the current state of the formation when starting the landing maneuver. To deal with this issue this thesis proposes a singularity-free representation of the formation, based on unit quaternions, and a formation controller, based on this formulation, to guide the formation in trajectory-tracking tasks. Besides the theoretical development, the text also provides results of experiments run using a formation composed by a *Pioneer 3-DX* UGV and a *Bebop 2* UAV, which validate the proposed formulation and controller.

Key-words: Quaternions; Formation Singularity; Trajectory-Tracking Task; Virtual Structure Paradigm; Mobile Robots Heterogeneous Formation.

Sumário

Lista de ilustrações

Lista de tabelas

List of Abreviations

UAV	Unmanned Aerial Vehicle
UGV	Unmanned Ground Vehicle
ROS	Robot Operating System
CS	Cluster Space
RS	Robot Space
IMU	Inertial Measurement Unit
MAR	Mobile Augmented Reality
EKF	Extended Kalman Filter
WF	World Frame
PF	Pioneer Frame
BF	Bebop Frame
DoF	Degrees of Freedom
FS	Formation Space

1 Introduction

There are many definitions of Robots, which vary according to the passage of time, social context, application in daily life and technological improvements. The Oxford Dictionary defines a robot as "... *a machine capable of carrying out a complex series of actions automatically, especially one programmable by a computer*" (OXFORD, 2009). Letting aside the modern concept of computers, there can be found examples of "machines" which could be included in this definition throughout History. One could trace the beginnings of Robotics to the great Greek engineer Ctesibius, who applied his knowledge of pneumatic systems and hydraulics to produce the first (known) organ and water clock with moving figures (ROSHEIM, 1994). Many other attempts to build self-operating machines can be found in ancient and medieval civilizations around the world (DUBS, 1960; GANCHY, 2009; HILL, 1991).

During the *Renaissance*, Leonardo da Vinci became notorious, among many other reasons, for his revolutionary ideas for flying machines, which demonstrated to be in general unfeasible, but still held important concepts that were further improved through time and helped in the development of this technology centuries ahead (ISAACSON, 2017). During the First Industrial Revolution, Richard Arkwright invented the first fully automated spinning mill driven by water power. This invention was then used by Oliver Evans in order to build an automatic wheat flour mill, much likely the first completely automated mill ever made (LIU, 2019).

Despite the enormous changes promoted by the two first industrial revolutions, industrial process still relied on human work for more complex functions before the 20-th century. Steam and internal combustion engines could replace (and actually increase) human physical strength, but decisions on whether or not lift a load, turn the machinery on, precisely position a screw-nut, and other complex functions still could not be totally accomplished without human intervention. Only with the understanding and domain of electricity together with the development of control theory, a more sophisticate usage of automation could thrive.

During, and greatly due to, the First and Second World Wars, major advances in automation were achieved. Electro-mechanical relay logic allowed considerable improvement for industry. Still, it was the technology based on solid-state digital logic that transformed completely the world and allowed the advances that can be observed today, not only in the industry, but also in the daily life of people. Automation is no longer a feature present only on plants and industries, but accessible for most of human population at certain level. Improvement on software, data and signal processing and machine learning in the second half of the 20-th century and throughout the 21-st century, intensified and promoted the access to systems with some level of decision-making capability, ranging from majorly human dependent to completely autonomous.

There are many different ways to classify autonomous systems (of which robots represent a sub-category), depending on their construction or application. One possible classification, when referring directly to robots, includes manipulator, mobile and manipulator-mobile robots. Surely, this is not a perfect definition, which is natural, once each definition is an attempt to make useful classification for a certain application or field of study. Nonetheless, it grasps many important aspects. Manipulators arms used in the automotive industry or in orbit of the earth for manipulation and repairs in the International Space Station (ISS) are examples of manipulator robots, while military and civil drones, as well as anthropomorphized (or even with an animal shape) robots are examples of mobile robots. Mobile robots with the capability to manipulate objects, by their turn, are included in the manipulator-mobile definition.

Mobile robots are of special interest for their potential to affect peoples' daily life. Self-driving cars represent a tremendously substantial innovation, being target of many research about their control system and other technologies that should be associated with them, such as pedestrian detection (CARDOSO et al., 2020; XU et al., 2018; ARIA, 2019; SARCINELLI et al., 2019). For goods transportation, self-driving vehicles could be included in the more generic definition of Unmanned Ground Vehicles (UGV), as long as no human is required for the vehicle to accomplish its task. Now drones, or more precisely Unmanned Aerial Vehicles (UAV), despite of being commonly associated to military applications, possess very interesting usages for civilians, ranging from merely toys to devices used for image capture in surveillance, environmental and agriculture inspections, search-and-rescue missions, gas leak detection and power-line inspections, for instance (FLOREANO; WOOD, 2015; HASSANALIAN; ABDELKEFI, 2017; DAPONTE et al., 2015).

It is not surprising that applications involving UGV and UAV formations arose in recently years, including agricultural inspections (ZHANG; KOVACS, 2012; MURUGAN; GARG; SINGH, 2017) and load transportation (PALUNKO; CRUZ; FIERRO, 2012; Pizetta; Brandão; Sarcinelli-Filho, 2020; PIZETTA; BRANDãO; SARCINELLI-FILHO, 2019), to quote some examples .

Several researches place UGV-UAV formations as feasible solutions for logistic and delivery companies, attacking different problems related to this application, such as path following (BACHETI; BRANDÃO; SARCINELLI-FILHO, 2020), route planning (GAO et al., 2020; MATHEW; SMITH; WASLANDER, 2015), land and take-off procedures (RABELO; BRANDãO; SARCINELLI-FILHO, 2021), task prioritization (MOREIRA; BRANDÃO; SARCINELLI-FILHO, 2019) and payload deliveries (YAKIMENKO et al., 2011), for instance. Others works deal with problems related to forest monitoring and fire detection strategies using teams of UGV-UAV (GHAMRY; KAMEL; ZHANG, 2016)

Among these many issues related to UGV-UAV formations, one important aspect that cannot be ignored is the landing procedure. Many of the applications previously described relies on the fact that the UGV usually possess a greater autonomy, being able to cover most of the distances, while the UAV's energy can be saved for a designated specific situation, thus increasing the set energy-autonomy. However, the landing procedure must be carefully executed, once that a failure landing could derail the whole operation.

As an example, in April 19^{th} 2021, the first flight in another planet was performed by the "Mars Helicopter" Ingenuity (a UAV), which was sent to the "Red Planet" together with the rover Perseverance[1]. For this mission, the goal was to demonstrate that it was possible to fly on Mars, what could be unfeasible, once its atmospheric volume is less than 1% of Earth's total. Once it has been proved possible, for future missions, it could be considered a formation where the rover (UGV) wander most of the distances and preserve the helicopter (UAV) to explore specific regions, such as canyons, mountains and craters (see Figure 1). This example illustrates how important is the precision for certain operations associated to robot formations, once it must be done autonomously for no human could manually replace the UAV over the UGV if it lands incorrectly. Besides, the distances between the Earth and Mars vary from 54.6 millions Km to 401 millions Km, meaning that any control signal sent from Earth would take between 182 seconds and 1,342 seconds, which are prohibitive signal delays for this control system.

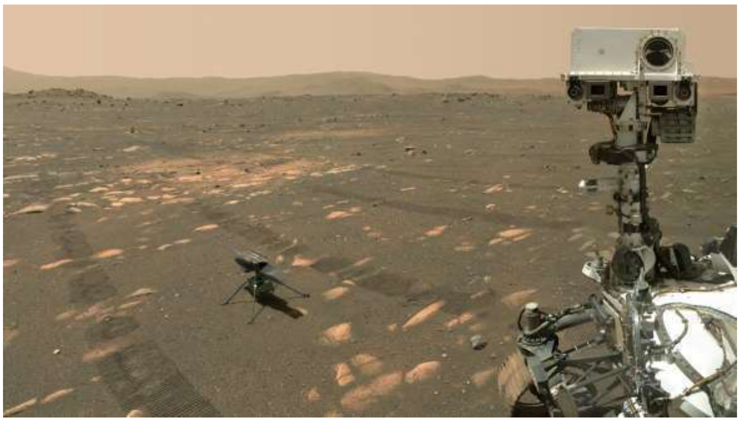


Figura 1 – "Selfie" of Perseverance and Ingenuity in Mars

Source: <https://apod.nasa.gov/apod/image/2107/PIA24542_fig2.jpg>

Therefore, it is reasonable to assume that certain operations associated to robot formations must be carried out as carefully as possible, avoiding sources for mistakes. The landing procedure is an example where some issues may arise. One recurrent problem is the formation singularities. In order to understand this situation, first it is necessary to explain the control

[1] See <https://mars.nasa.gov/technology/helicopter/#People-Profiles>

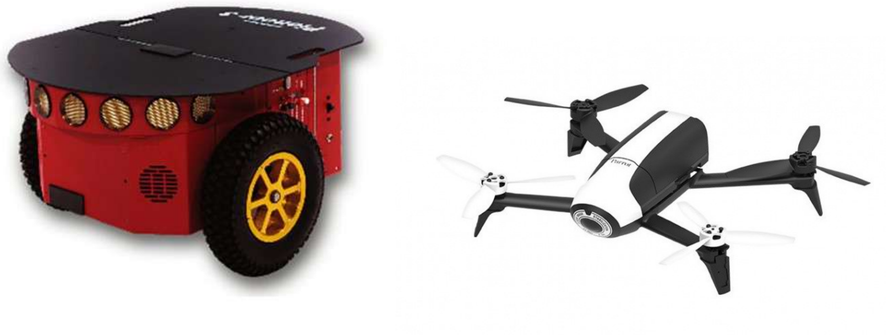
paradigm of virtual structures for controlling robot formations. These structures are made in such a way that the robots considered form a geometric figure, such as lines or triangles (RABELO; BRANDãO; SARCINELLI-FILHO, 2021; Kitts; Mas, 2009), and the variables considered represent an aspect of the figure and not of the robots, such as its geometric center or one of its edges, its length or area, and so on. The controller guides the robots to move accordingly in order to keep the desired shape. All variables considered in order to describe the virtual formation can be agglutinated in a single entity, denominated as cluster. In order to precisely characterize this entity, all variables must be well defined, with no misrepresentation. Formation singularities are then situations where one or more of such variables become undefined.

Singularities may arise due to physical restrictions, such as a line describing a formation between two robots with length zero, meaning that both robots should occupy the same position at the same time, what is obviously impossible, because in this case one robot would be mounted on the other, thus not characterizing a formation. Other possibility are singularities caused by the mathematical representation chosen for the formation. For those cases, what causes the Cluster Space (CS) to become undefined are the kinematic equations that map the Robot Space (RS) into the CS. Euclidean descriptions (meaning mathematical descriptions based on Euclidean space and geometry) of the kinematic equations usually present points or regions where singularities arise. It is important to emphasize that those singularities appear due to the mathematical description chosen and does not represent a physical limitation of the formation, as previously exemplified. It is unsettling that the mere choice of a mathematical representation may bring problems to the control system. Fortunately, there are options regarding to formation singularities. One of those options is the representation of the cluster based on quaternions.

The quaternions number system is an extension of the complex numbers. They provide an unique and powerful tool for characterizing the relationships among 3D orientation frames that the orthonormal axes themselves and traditional representations of them (Euclidean representations) are unable to supply. One important aspect is that quaternions provide a double covering (2:1 mapping) of the ordinary 3D rotation, this way avoiding misrepresentations that occur for most Euclidean representations (HANSON, 2006).

Due to its potential, this thesis proposes a quaternion based description for a UAV-UGV formation where no points of singularity occur for any attainable form of the CS. This is done through experiments which follow the control paradigm of virtual structures for multi robot systems, previously explained. A UAV together with a UGV form the limits of a virtual line linking both robots. The point of interest for control of such a virtual line will not be, in this work, its centroid, as in other virtual structures, such as in (Kitts; Mas, 2009), but one of its limits (the UGV current position). The task proposed for the formation is to follow a given trajectory, keeping a certain shape for the formation, which is the UAV exactly over the UGV. This specific situation corresponds to the virtual structure exactly in the vertical position, and is necessary for the landing of the UAV over the UGV. The problem is that this situation represents a singularity for the most usual Euclidean description. Moreover, a quaternion based description also allows the formation to reach configurations that would be impossible when considering an alternative Euclidean description, which does not present singularities for the landing procedure, but looses attainability of the CS.

For the validation experiments reported ahead, it is considered an unicycle-like nonholonomic platform *Pioneer 3-DX*, from *Adept Mobile Robots*, the UGV, and a quadrotor *Bebop 2*[2], from *Parrot Drones SAS*, the UAV, which are shown in Figure 2.



(a) UGV used for experiments

(b) UAV used for experiments

Figura 2 – UAV-UGV robots used for the experiments reported in this work.

The experiments were run in a structured environment, where the positions and velocities of the robots, as well as their orientations, were acquired by the high precision motion capture system *OptiTrack*[3], configured with eight cameras. Figure 3 shows the robots used to run the experiments, and three of the eight cameras as well. Therefore, the data acquisition necessary for this work depends on measurements made by an equipment inherent to indoor applications, which may appear as a weakness. Although it is clear that the *OptiTrack* system is unfeasible for outdoor applications, the main goal of this work is to demonstrate how a quaternion description for a UAV-UGV formation can be superior to most Euclidean descriptions when formation singularities are an issue. Feasible sensing system for outdoor application involving UAV and UGV formations is, by itself, a subject for another work.

[2] See <https://www.parrot.com/us/drones/parrot-bebop-2-power-pack-fpv>

[3] See <https://optitrack.com/motion-capture-robotics/>

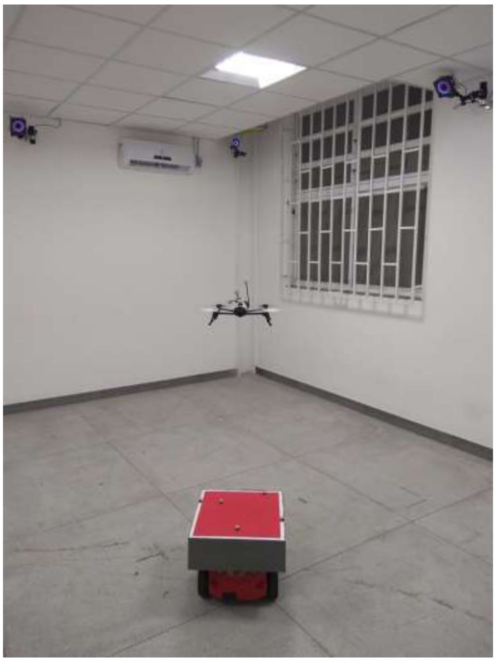


Figura 3 – Pioneer 3-DX and Bebop 2 with their *OptiTrack* markers and three of eight cameras

1.1 State of the Art

As mentioned before, quaternions are helpful tools to describe three-dimensional rotations and orientations for many applications. Radavelli and his co-authors (RADAVELLI et al., 2012) compare a dual-quaternion description (which includes translation in the description) with the Denavit-Hartenberg convention (see (SPONG; VIDYASAGAR, 1989)) for the kinematic description of robot manipulators. They conclude that dual-quaternions are computationally more costly, because their multiplications is more complex. On the other hand, dual-quaternions take advantages from homogeneous matrices in the storage point of view, since homogeneous matrices require 12 numbers to represent six degrees of freedom, whereas dual-quaternions require just eight. Besides, quaternions and dual-quaternions provide robot kinematics representations without singularities, which is a major problem regarding homogeneous matrices derived from Euclidean descriptions.

In (FIGUEREDO et al., 2013) it is presented a control analysis strategy for robot manipulators which uses dual-quaternion space to avoid decoupling the end-effector rotational and translational dynamics and representation singularities. In (XIAN et al., 2004) it is presented a class of task-space tracking controller for robot manipulators which describes the end-effector orientation in terms of the unit quaternions without singularities. Others situations where quaternions are used for robot manipulators mainly as an alternative way to avoid representation singularities are presented in (PHAM et al., 2010), (BRAGANZA et al., 2005) and (AYDIN; KUCUK, 2006).

Quaternions are also useful for sensing devices, such as in (YURTMAN; BARSHAN; FIDAN, 2018), where a wearable sensor orientation is computed through rotations between consecutive time samples represented by unit quaternions in the Earth frame. In this work, it has been observed that representing rotational transformations by rotation matrices, instead of quaternions, degrades the accuracy, what reinforce some advantages of quaternions over the most conventional representations.

In (YUN; BACHMANN, 2006) it is presented a quaternion-based Kalman Filter for human body motion tracking, designed for real-time estimation of the orientation of human limb segments. This work also highlights that using quaternions improves computational efficiency and provides singularity avoidance. By its turn, in (VARGHESE; CHANDRA; KUMAR, 2015) it is proposed a dual-quaternion based Inertial Measurement Unit (IMU) and vision sensor fusion framework for accurate tracking in Mobile Augmented Reality (MAR), as an extension of the quaternion based Extended Kalman Filter (EKF) developed in (KUMAR et al., 2014), resulting in a dual-quaternion based EKF. Others examples of quaternions use for sensing devices may be observed in (CHOU; KAMEL, 1991), (TADANO; TAKEDA; MIYAGAWA, 2013) and in (RENAUDIN; COMBETTES, 2014), as a solution for problems related to singularities, among others issues.

Therefore, it can be observed that many systems which depend on some kind of oriented framework are susceptible to problems related to singularities. This is not different for formations of unmanned mobile vehicles. In (YANG; GU; HU, 2005) it is proposed a leader-follower formation composed by two nonholonomic robots where a structural formation singularity exists. This issue is circled by applying a linearization at the castor wheel position, rather than the center of the driving wheels. Simulations presented an acceptable result, as long as the restrictions derived from the linearization at the castor wheels position are respected, which is rather limited. In (DAI et al., 2019) it is also used nonholonomic robots for a leader-follower formation, where the singularity issue is solved by considering a bearing angle constraint, therefore limiting the attainability of the formation as a whole. Even though those works manage to solve the control issue proposed bypassing somehow the singularities, these solutions are obtained compromising precision, generality or attainability of the formation, which is unsettling.

In (RABELO; BRANDãO; SARCINELLI-FILHO, 2021) it is proposed a controller for the landing of an UAV on static or moving platforms (an UGV). For such endeavor, it is used an Euclidean description and a virtual structure control paradigm. More specifically, the virtual line segment connecting both UAV and UGV is controlled, regarding its length and orientation. For this set up, the singularities arise exactly when the UAV hovers over the UGV, which represents precisely the required formation configuration for the landing procedure. In (ERNANDES-NETO; SARCINELLI-FILHO; BRANDÃO, 2019) it is proposed a different Euclidean description for a similar problem, where no singularity arises when the UAV hovers over the UGV. Nevertheless, there is still a region of singularity which has been moved elsewhere. That way, even though the singularity for the moment of landing is solved, it is done by compromising the CS attainability of the formation.

In (MAS; KITTS, 2017) it is proposed a quaternion based description for a formation between two UAVs, as well as a quaternion-based controller for a leader-follower trajectory tracking task. Through some adjustments, it is possible to adapt this description for a UAV-UGV formation, now using the virtual structure control paradigm, also performing a trajectory tracking task. This allows the formation to execute the trajectory tracking task, as it has been done in (RABELO; BRANDãO; SARCINELLI-FILHO, 2021), without the singularity issue during the landing procedure and with no loss in the CS attainability as it happens in (ERNANDES-NETO; SARCINELLI-FILHO; BRANDÃO, 2019). Besides, in this work real experiments have been run in order to validate the proposal, whereas in (MAS; KITTS, 2017) the validation was made only through simulations. Therefore, this work strengthens and complements the findings of (MAS; KITTS, 2017) and provides a feasible solution for existing problems in (RABELO; BRANDãO; SARCINELLI-FILHO, 2021) and in (ERNANDES-NETO; SARCINELLI-FILHO; BRANDÃO, 2019).

1.2 Motivation

As automation improves, it shows up as a powerful tool for many application fields, from the more prosaic (such as grocery shopping) to the more knowledge expending (such as spacial exploration). The first flight in another planet made by the Ingenuity drone is a remarkable achievement for planetary exploration, once it widens the range of possibilities for future missions, allowing mankind to improve observations and data acquisition in other worlds. All Galilean moons (Io, Europa, Ganymede and Callisto) and even the dwarf planet Pluto are known to have thin atmosphere (thinner than Mars' atmosphere), meanwhile Titan (Saturn's moon), the second largest moon in the solar system, has an atmosphere estimated to be 45% denser than Earth's. Being able to fly these objects is a challenge in itself. Therefore, the recent flight of a drone on Mars served as an important rehearsal for similar missions, keeping in mind that Europa and Titan are long-time desired targets for exploratory missions by NASA and other space agencies.

Wherever the next spacial mission aims to go to, now it is possible to consider a UAV-UGV formation, in which the UGV carries the UAV by most paths, saving UAV energy to explore specific regions and take overview footages from landscapes with a better definition than those acquired by probes in orbit of planets, moons or dwarf planets. As such, the landing of the UAV over the UGV is one of the many issues related to such missions. Moreover, it must be carried as carefully as possible, once that the distances involved make any remote control impossible, being necessary for the robots formation to perform all procedures autonomously. Therefore, a quaternion based description for this robot formation presents itself as a suitable starting point, once it does not present formation singularities throughout its CS.

As it has happened many times in History, technologies developed for fields of study apparently disconnected to more earthly applications and needs demonstrated themselves remarkably transforming in society. The internet and smartphones are very well known examples. Similarly, the advances in robot formation controllers applied to interplanetary exploration can be converted to exploration of dangerous areas on Earth, such as mines or nearby forest fires. Delivery companies may adapt this technology for vehicles formation, where the ground vehicle moves through the main roads and avenues, letting for the aerial vehicles the last mile deliveries, thus minimizing delivering time and energy consumption. Thinking on indoor applications, automated warehouses may use UAV-UGV formations in such way that the UAV finds, carries and lands the goods over the UGV, which, by its turns, takes the load to a next section of the distribution line. For all these examples, it is essential for the UAV to land or hover over the UGV, demonstrating how harmful it might be for the whole system if singularities arise exactly at this moment.

Therefore, in order to deal with singularities that may arise for Euclidean descriptions of robot formations, the present work proposes a quaternion-based description for a UAV-UGV formation performing a trajectory tracking task, exploring formations where singularities would arise for most Euclidean descriptions. Besides, it is explored the fact that the quaternion-based description for such formations has no loss in the attainability of the CS, as it happens for some Euclidean descriptions, which attempt to avoid singularities when the UAV hovers over the UGV.

1.3 Objectives

The more general goal of this thesis is to develop a controller for a UAV-UGV formation based on quaternions, capable of performing trajectory tracking tasks keeping configurations that would represent singularities for Euclidean descriptions, to demonstrate that quaternion descriptions for mobile robots are as functional as Euclidean descriptions without issues related to singularities. The cluster condition in which the UAV hovers over the UGV simulates usual configurations in applications that range from spacial exploration to automated warehouses, and represents situations where singularities usually arise for most Euclidean descriptions. Besides, as a secondary goal, it is going to be analyzed how the CS attainability is affected by quaternion-based and Euclidean descriptions. Furthermore the quaternion-based description here proposed is validated through the experiments described in Chapter 5.

1.4 Text Structure

This thesis is organized according to the following structure:

Chapter 1: Introduction

Chapter introducing the main concepts related to this thesis, as well as the state of the art and the motivation for this work, also highlighting the goals intended to be achieved here.

Chapter 2: Quaternions Background

Chapter aiming to provide the most important notions of quaternions, the history of its development, its role in group theory and the main concepts required for the development of this work.

Chapter 3: Experimental Setup

In this chapter, it is explained the infra-structure available for the experiments here developed, as well as some major limitations and how they affected the final results.

Chapter 4: Formation Description and Controller Design

Chapter explaining the quaternion description for the robot formation considered, which is compared to two different Euclidean descriptions for the same formation, in order to demonstrate the strengths of the quaternion description in comparison to the Euclidean ones. Furthermore, the quaternion based controller that is used for the trajectory tracking task is developed.

Chapter 5: Experimental Results and Discussions

In this chapter the experiments run are explained, in order to clarify each one's goals and the results obtained are presented. The results found are further discussed, in order to highlight the most important conclusions.

Chapter 6: Conclusion

This chapter ranks the main conclusions and achievements of this work, as well as proposes further research lines related to it.

Concluding this chapter, it is worth mentioning that this research allowed the publication of the followings papers:

1. Harrison Neves Marciano, Alexandre Santos Brandão e Mário Sarcinelli Filho, "Singularity-Free Quaternion Representation to Control a UGV-UAV Formation Performing Trajectory-Tracking Tasks", 2021 International Conference on Unmanned Aircraft Systems (ICUAS), Athens, Greece, 2021, pp. 656-665.

2. Harrison Neves Marciano, Alexandre Santos Brandão e Mário Sarcinelli Filho, "Uso de Quatérnios para Solução do Problema de Singularidade de uma Formação VANT-VTNT", 14th IEEE/IAS International Conference on Industry Applications, São Paulo, Brazil, 2021 (accepted paper).

3. Harrison Neves Marciano, Anthony Oliveira Pinto, Vinicius Pacheco Bacheti, Mauro Sergio Mafra Moreira, Alexandre Santos Brandão e Mário Sarcinelli-Filho, "Modelagem de Alto Nível e Controle do Quadrimotor Bebop 2", XXIII Congresso Brasileiro de Automática (CBA2020), Porto Alegre, RS, 2020, doi: 10.48011/asba.v2i1.1114

4. Anthony Oliveira Pinto, Harrison Neves Marciano, Vinicius Pacheco Bacheti, Mauro Sergio Mafra Moreira, Alexandre Santos Brandão e Mário Sarcinelli-Filho, "High-Level Modeling and Control of the Bebop 2 Micro Aerial Vehicle", 2020 International Conference on Unmanned Aircraft Systems (ICUAS), Athens, Greece, 2020, pp. 939-947, doi: 10.1109/ICUAS48674.2020.9213941

2 Quaternions Background

This section gives some basic properties of quaternions, besides some historical aspects of its creation and use.

2.1 Quaternions Development

In (NEVES, 2008) is described the most famous story about the "discovery" of quaternions and the man behind it. Thereit is told that on the 16-th day of October, 1843, William Rowan Hamilton walked to the Council of the Royal Irish Academy, beside his wife, where he would attend and preside. Although his wife talked to him eventually, his mind was taken by a train of thoughts which led him to a so persecuted and desired result. The excitement for his conclusion rushed him toward his notebook in order to write his findings down. Not yet satisfied, Hamilton went to the Brougham Bridge, where he cut with a knife the fundamental formula which contains the solution for the problem that he had been trying to solve for more than one decade, namely

$$i^2 = j^2 = k^2 = ijk = -1 \tag{2.1}$$

The original carving faded away long ago. However, a plaque has been placed at the Brougham Bridge as a remembrance of Hamilton achievement, as it can be seen in Figure 4. For the rest of his life, Hamilton attempted to popularize quaternions having some success locally, in Dublin. As the time progressed, nearby the end of the XIX century, quaternions began to be replaced, by the scientific community, in favor of the vector analysis, which could represent the same phenomena as quaternions, being more straightforward and clearer in notation, relegating quaternions to restrict fields of mathematics and physics. Not only being relegated, quaternions were considered as harmful for some notorious mathematicians of the period. Oliver Heaviside once said[1]:

> *"I came later to see that as far as the vector analysis I required was concerned, the quaternion was not only not required, but was a positive evil of no inconsiderable magnitude."*
>
> *Oliver Heaviside*

[1] See <https://www.youtube.com/watch?v=d4EgbgTm0Bg>

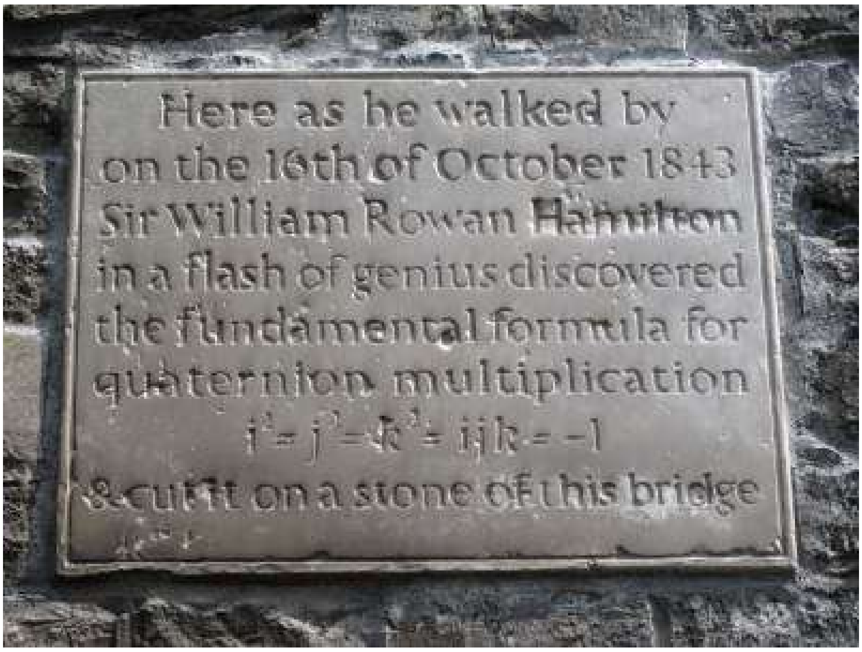

Here as he walked by
on the 16th of October 1843
Sir William Rowan Hamilton
in a flash of genius discovered
the fundamental formula for
quaternion multiplication
$i^2 = j^2 = k^2 = ijk = -1$
& cut it on a stone of this bridge

Figura 4 – Plaque containing the fundamental formula of quaternions

Furthermore, Lord Kelvin said:

> *"Quaternions came from Hamilton after his really good work had been done; and, though beautifully ingenious, have been an unmixed evil to those who have touched them in any way, including Clerk Maxwell.*
>
> *Lord Kelvin*

It is even believed that the Mad Hatter's Tea Party scene in Alice in Wonderland, where it is quoted *"... Why, you might just as well say that I see what I eat is the same thing as I eat what I see"*, mocks with the non-commutative nature of quaternions.

Quaternions were truly relegated to minor fields of study for approximately a century, until, in the late 20-th century, professionals and researchers on graphic design, robotics and any field where three-dimensional rotation in space were necessary, revived it due to its pragmatic and elegant way to describe and to compute three-dimensional rotations. Besides, quaternions demonstrated to be computationally more efficient than other methods and less susceptible to numerical errors that arise when using such other methods. Quaternions also show to be quite useful for quantum mechanics, since the actions they describe in four dimensions are considerably suitable to mathematically describe two-state systems in quantum mechanics.

Hamilton had none of this in his mind when he began to study what would become the quaternions, obviously. To him, all started due to his interest on defining a multiplication for triplets which possessed the same properties of the complex numbers and that could be interpreted as rotations in space, analogously to rotations in planes, as it also happens with complex numbers. His first attempts involved the addition of another complex number "j", orthogonal to the real numbers and to "i", holding the same property of "i", whose square is equal to -1. He struggled for many years with the fact that for this definition the distributive property did not hold. The product between "i" and "j" did not represent anything inside the set he had defined. This situation was overcome just that October 16-th, when Hamilton conceived a fourth spatial dimension, orthogonal to the other three, allowing the creation of a consistent number system set with an effective geometric meaning.

Indeed, the German mathematician Ferdinand Georg Frobenius, in 1878, proved that the finite dimension associative division algebras over the real numbers are isomorphic only for three number systems, namely the real one, the complex one and the quaternions. This explains why Hamilton could not solve his problem by simply adding another imaginary element, being necessary a fourth one. Loosing a little the restrictions and removing the associative condition, but keeping a notion of norm, Adolf Hurwitz (also German) proved, in 1898, that it is possible to add another algebra represented by the Octonions to the other three (PEREIRA; ABREU,), which will not be discussed in this thesis, but also possesses interesting applications in mathematics.

A more explicit way to express Hurwitz's finding is the proof that

$$\left(a_1^2 + a_2^2 + ... + a_n^2\right) = \left(b_1^2 + b_2^2 + ... + b_n^2\right)\left(c_1^2 + c_2^2 + ... + c_n^2\right) \tag{2.2}$$

is true only for $n = 1,\ 2,\ 4,\ 8$.

At first, Hamilton did not know what was the nature of 'k', but eventually he concluded that it had a nature similar to that of 'i' and 'j', an orthogonal line in relation to the others, with modulus 1. Besides, quaternions multiplication preserves the distributive and associative properties, but not the commutative for the imaginary elements, which makes quaternions a four-dimensional non-commutative associative normed division algebra over the real numbers.

Section 2.2 will explore somewhat further such aspects.

2.2 Properties of Quaternions

2.2.1 Quaternions Algebra

The quaternion set $\mathbb{H}$ is a vector space over $\mathbb{R}^4$, which can be defined as:

$$\mathbb{H} = \{w + xi + yj + zk \ \mid \ w, x, y, z \in \mathbb{R},\ \text{and}\ \ i^2 = j^2 = k^2 = ijk = -1\} \tag{2.3}$$

From (2.3) it can be derived that:

$$\begin{aligned} ij &= -ji = k; \\ jk &= -kj = i; \\ ki &= -ik = j. \end{aligned} \tag{2.4}$$

Proofs: using

$$ijk = -1 \quad \therefore \quad ijk^2 = -k \quad \therefore \quad -ij^2 = -kj \quad \therefore \quad i^2 = -kji \quad \therefore \quad -1 = -kji \quad \therefore \quad kji = 1, \tag{2.5}$$

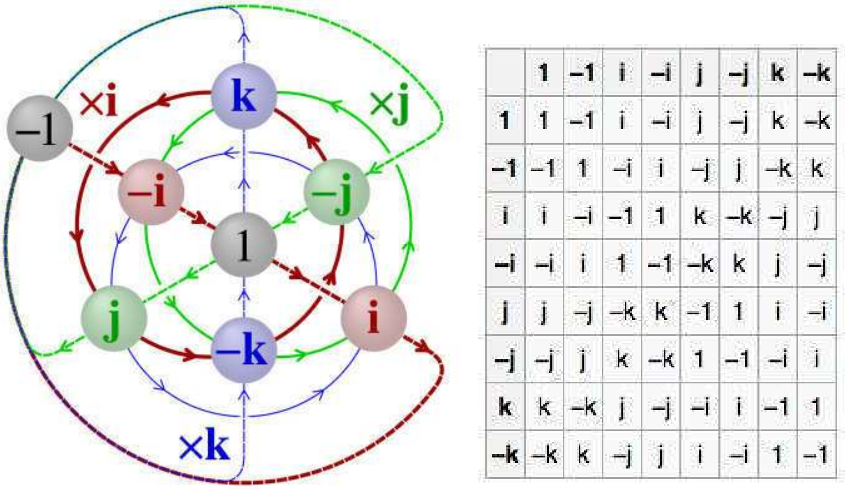
and also using (2.3), it can be concluded that

$$\begin{aligned} ijk = -1 \quad &\therefore \quad ijk^2 = -k \quad \therefore \quad -ij = -k \quad \therefore \quad ij = k; \\ kji = 1 \quad &\therefore \quad k^2ji = k \quad \therefore \quad -ji = k \quad \therefore \quad ji = -k; \\ ijk = -1 \quad &\therefore \quad i^2jk = -i \quad \therefore \quad -jk = -i \quad \therefore \quad jk = i; \\ kji = 1 \quad &\therefore \quad kji^2 = i \quad \therefore \quad -kj = i \quad \therefore \quad kj = -i. \end{aligned} \tag{2.6}$$

Now, using (2.6), it can be obtained

$$\begin{aligned} ji = -k \quad &\therefore \quad ji^2 = -ki \quad \therefore \quad -j = -ki \quad \therefore \quad ki = j; \\ ij = k \quad &\therefore \quad i^2j = ik \quad \therefore \quad -j = ik \quad \therefore \quad ik = -j. \end{aligned} \tag{2.7}$$

Figure 5 shows a diagram with the quaternion multiplicative relationships, as well as the multiplicative table for quaternions.



	1	−1	i	−i	j	−j	k	−k
1	1	−1	i	−i	j	−j	k	−k
−1	−1	1	−i	i	−j	j	−k	k
i	i	−i	−1	1	k	−k	−j	j
−i	−i	i	1	−1	−k	k	j	−j
j	j	−j	−k	k	−1	1	i	−i
−j	−j	j	k	−k	1	−1	−i	i
k	k	−k	j	−j	−i	i	−1	1
−k	−k	k	−j	j	i	−i	1	−1

(a) Cayley Q8 Graph

(b) Quaternions Multiplicative Table

Figura 5 – Multiplicative relationships between quaternions.

2.2.2 Quaternion Group of Symmetry

A symmetry of an object is a geometric transformation which leaves the object unchanged (HART; SEGERMAN, 2016). There are 8 possible transformations over a square that would keep it undistinguished from how it started. A circle may have infinite transformations and still it would be impossible to differentiate it from its initial form. Each action over an object that keeps its form unchanged is a symmetry of this object and the set of these symmetries form a group of symmetry. Moreover, not only the actions that describe a group are relevant, but the interactions between those actions also are. Similarly to a sequence of rotations which could be described as a single one having the same final result, a sequence of actions of a group must be equivalent to a single action which also belongs to that group. Rotating a square 90º counter-clockwise and then flipping it around the vertical axis is equivalent to simply flip it over the diagonal axis which passes through the upper right corner and the bottom left corner[2].

More strictly, a *group* is a set, together with an operation on its members, which satisfies certain axioms. Those axioms are not chosen randomly, obviously, but they are the minimum rules restrictions over a geometric object under which all transformations keep it invariant. In this subsection it is considered an operator ($*$), which could be an addition, a multiplication or even a made up operator. It does not matter as long as the axioms are respected. Let G be a set of elements. Then, G will form a *group* if:

1. $a,\ b \in G$ are any two members, then $a * b$ is defined, and also a member of G[3];
2. the associative law holds: $(a * b) * c = a * (b * c)$ for every three $a,\ b,\ c \in G$;
3. there is an "identity element", called e, so that $e * a = a * e$ for all $a \in G$.

   There is another rule that does not prevent a set to form a *group*, although being important in order to define it. Such a rule is
4. for every element $a \in G$ there is another element $b \in G$ with the property that $a * b = b * a = e$. The element b is usually written as a^{-1}.

When these four properties hold for all pairs of elements in G, G is called a *commutative group* or, more commonly, an *abelian group*. When the three first of such properties hold, but the fourth one does not hold for even one pair of elements, G is called as a *non-commutative* or a *non-abelian group*.

[2] See <https://www.youtube.com/watch?v=mvmuCPvRoWQ&t=254>

[3] Interestingly, this was one of the aspects that made Hamilton to struggle the most. Remember that he could not give a meaning to $i \circ j$ that would still belong to the set he had defined. Conceiving a fourth dimension back then was somewhat shocking, even though it seems complete acceptable for abstraction purposes today.

Quaternions form a finite *group*, called (quite originally) as the *quaternion group* Q_8, which is a *non-abelian group* of order eight. It is given by,

$$Q_8 = \langle \bar{e},\ i,\ j,\ k \ \mid\ \bar{e}^2 = e, i^2 = j^2 = k^2 = ijk = \bar{e} \rangle, \tag{2.8}$$

which is isomorphic to the eight-element closed to multiplication subset,

$$S = \{1,\ i,\ j,\ k,\ -1,\ -i,\ -j,\ -k\}. \tag{2.9}$$

Group Theory is a remarkably relevant field of study in mathematics, being the basis for several theories and applications. Besides, it represents some fundamental structures in nature. Therefore, it cannot be ignored how interesting is the fact that quaternions form a group of symmetry of their own, which emphasizes how important quaternions can be for fields of study going beyond the application here described.

2.2.3 Quaternions Representation

A usual notation for a given quaternion is

$$q \in \mathbb{H} \ \mid\ q = q_0 + q_1 i + q_2 j + q_3 k, \ \text{where } q_0, q_1, q_2, q_3 \in \mathbb{R}, \tag{2.10}$$

or, in a more compact writing,

$$q \in \mathbb{H} \ \mid\ q = q_0 + \mathbf{q}, \tag{2.11}$$

where $\mathbf{q} = q_1 i + q_2 j + q_3 k$ represents the imaginary or vector part of a quaternion (to avoid confusion with the imaginary elements of a quaternion, hereinafter in this thesis the $\mathbf{q}$ part of a quaternion will be referred to as its vector part). By its turn, q_0 represents the real part of the quaternion.

Similarly to the complex numbers, quaternions have an exponential representation, which is

$$q = ||q|| e^{\frac{\theta}{2}\hat{\mathbf{n}}} = ||q|| (\cos\frac{\theta}{2} + \text{sen}\frac{\theta}{2}\hat{\mathbf{n}}), \tag{2.12}$$

where $||q||$ is the modulus of the quaternion, θ is the rotation applied by it and $\hat{\mathbf{n}}$ is an unit vector representing the axis around which the rotation is performed, such that,

$$\hat{\mathbf{n}} = \frac{\mathbf{q}}{||\mathbf{q}||} = \frac{q_1 i + q_2 j + q_3 k}{\sqrt{q_1^2 + q_2^2 + q_3^2}}, \tag{2.13}$$

Similarly to a multiplication between complex numbers, which causes a rotation in the plane and a stretching or squeezing, depending on the modulus of the complex number, the multiplication of two quaternions also does that, now in the three-dimensional space.

Therefore, in order to apply spatial rotation without distorting the original length of the vector, the quaternion used should be normalized, so that,

$$\hat{q} = \frac{q}{||q||} = \frac{q_0 + q_1 i + q_2 j + q_3 k}{\sqrt{q_0^2 + q_1^2 + q_2^2 + q_3^2}} = e^{\frac{\theta}{2}\hat{\mathbf{n}}} = (\cos\frac{\theta}{2} + \operatorname{sen}\frac{\theta}{2}\hat{\mathbf{n}}). \tag{2.14}$$

Normalized quaternions are also known as unity length quaternions, or simply unit quaternions. Furthermore, they can be expressed as a conjugated form, in which the vector part has its signal inverted, or

$$q^* = (q_0,\ -\mathbf{q}). \tag{2.15}$$

Applying the results in (2.6) and (2.7), it can be derived the general non-commutative multiplication rule for two quaternions. Let it be,

$$q = q_0 + q_1 i + q_2 j + q_3 k \ \text{ and } \ p = p_0 + p_1 i + p_2 j + p_3 k. \tag{2.16}$$

Then,

$$\begin{aligned} q \circ p = q_0p_0 + q_1p_1 + q_2p_2 + q_3p_3 + i\,(q_0p_1 + q_1p_0 - q_2p_3 + q_3p_2) + \\ j\,(q_0p_2 + q_1p_3 + q_2p_0 - q_3p_1) + k\,(q_0p_3 - q_1p_2 + q_2p_1 + q_3p_0)\,, \end{aligned} \tag{2.17}$$

where $\circ$ represents the multiplication between two quaternions.

Now, considering q and p as 4x1 vectors, the multiplication rule can be written more compactly as,

$$q \circ p = (q_0p_0 - \mathbf{q}\cdot\mathbf{p},\ q_0\mathbf{p} + p_0\mathbf{q} + \mathbf{q}\times\mathbf{p})^T\,. \tag{2.18}$$

As a result,

$$q \circ q^* = q_0^2 + q_1^2 + q_2^2 + q_3^2 = ||q||^2. \tag{2.19}$$

To conclude, once all the quaternions considered in this work are unit quaternions, they are such that,

$$\hat{q} \circ \hat{q}^* = ||\hat{q}||^2 = 1. \tag{2.20}$$

2.2.4 Three-Dimensional Rotation and Orientation

In order to represent three-dimensional rotations by quaternions, the first step is to write the point or vector which the rotation will be applied to in such a way that,

$$p = 0 + xi + yj + zk. \tag{2.21}$$

Then, following (2.14) it is defined a normalized three-dimensional axis $\hat{\mathbf{n}}$, around which the rotation will be performed, and the angle θ, the rotation angle to be applied to p. Finally, the new position is found multiplying p by the unit quaternion $\hat{q}$, derived from $\hat{\mathbf{n}}$ and θ, to the left, and by $\hat{q}^*$ to the right. Notice that successive rotations can be applied, what means to repeat this processes continuously. This can be better summarized as

$$p' = \hat{q}_n \circ \hat{q}_{n-1} \circ \ \ldots \ \circ \hat{q}_2 \circ \hat{q}_1 \circ p \circ \hat{q}_1^* \circ \hat{q}_2^* \circ \ \ldots \ \circ \hat{q}_{n-1}^* \circ \hat{q}_n^*, \tag{2.22}$$

where,

$$\hat{q}_n = q_{n0} + q_{n1}i + q_{n2}j + q_{n3}k = q_{n0} + \mathbf{q}_n \tag{2.23}$$

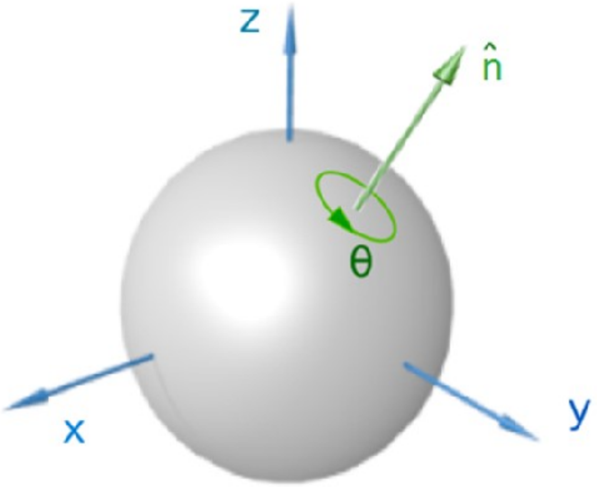


Figura 6 – Three-dimensional rotation around axis $\hat{\mathbf{n}}$ by angle θ

Not only rotations can be characterized by quaternions. They can also be used to express the spatial attitude or orientation, through $\hat{\mathbf{n}}$ and θ. A UGV, moving in a plane surface, would always have a $\hat{\mathbf{n}} = k$, while a UAV, which can move freely in the 3-D space, could have any combination of $q_1 i$, $q_2 j$ and $q_3 k$ in order to form $\hat{\mathbf{n}}$, whenever $\sqrt{q_1^2 + q_2^2 + q_3^2} = 1$. As a consequence of (2.12), θ obeys $0 \leqslant \theta < 4\pi$, rather than $0 \leqslant \theta < 2\pi$ in order to represent all possible quaternions. It implies that the quaternions $\hat{q}$ and $-\hat{q}$ represent the same orientation and generate the same three-dimensional rotation. Therefore, quaternions accomplish a double covering (2:1 mapping) of ordinary 3D rotations (HANSON, 2006).

This fact allows a continuous transition between orientation angles. For instance, consider a UGV on a plane surface, starting navigating with an orientation of $\hat{q} = 1 + 0i + 0j + 0k$. If a counter-clockwise rotation is applied to it, when it finishes a complete revolution its orientation will have varied continuously up to $\hat{q} = -1 + 0i + 0j + 0k$, which also represents the initial orientation. If another revolution is applied at the same direction, it will vary, also continuously, until $\hat{q} = 1 + 0i + 0j + 0k$. Therefore, the abrupt transition between 2π and 0 does not occur for this orientation description.

Tabela 1 – Examples of spherical geometry definitions

Description	**Equation**	**Embedding Dimension**
Circle (S^1)	$q \cdot q = q_0^2 + q_1^2 = 1$	$\mathbb{R}^2$
Sphere (S^2)	$q \cdot q = q_0^2 + q_1^2 + q_2^2 = 1$	$\mathbb{R}^3$
Hypersphere (S^3)	$q \cdot q = q_0^2 + q_1^2 + q_2^2 + q_3^2 = 1$	$\mathbb{R}^4$

2.2.5 Unit Quaternions Visualization

An unit quaternion is no more than a point on an unit four-dimensional hypersphere. Therefore, in order to visualize it, the first thing to do is to compare how it is possible to visualize a three-dimensional sphere as circles in a plane, and, then, to apply the analogous process with the four-dimensional hypersphere using three-dimensional spheres.

Usually, spheres are three-dimensional objects. However, mathematically speaking they could be generalized to any dimension, as long as all set of points that constitute it are equally apart from its origin and form a closed curve or surface. An unit sphere, then, holds these properties and the fact that the distance between any point and the origin is equal to 1. The table 1 summarizes this definition.

Starting with a sphere (S^2), once its radius is specified, in this case 1, the equation describing it has two degrees of freedom, such that,

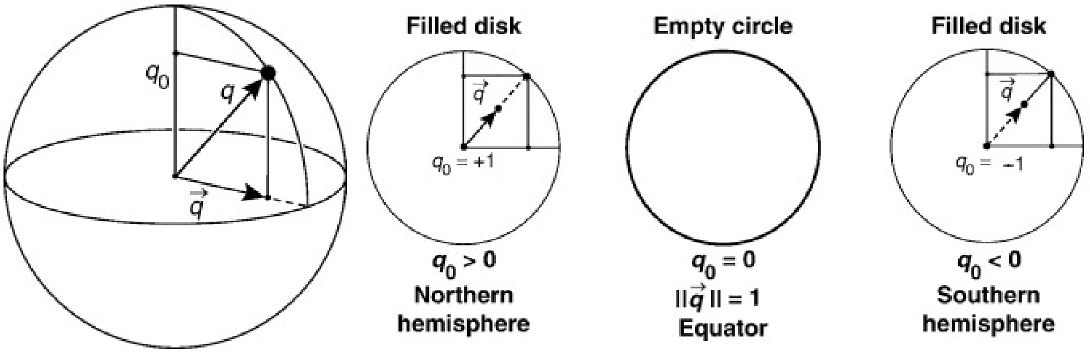
$$q_0 = \pm\sqrt{1 - (q_1)^2 - (q_2)^2}. \tag{2.24}$$

where the positive values of q_0 represent the northern hemisphere and the negative values the southern hemisphere. If all points that form the sphere surface are projected to the intersection between a plane passing through the equator and the sphere, then both the northern and the southern hemispheres are going to create a filled disk each one. Meanwhile, $q_0 = 0$ represents the equator, being the only projection which represents precisely the original region on the sphere. This way, the whole sphere (S^2) can be mapped using one dimension less, in this case using three circles (S^1).

Now, repeating this reasoning to an unit hypersphere (S^3), the equation which describes it is such that

$$q_0 = \pm\sqrt{1 - (q_1)^2 - (q_1)^2 - (q_2)^2}. \tag{2.25}$$

where, similarly to the previous case, the positive values of q_0 represent the northern "hemisphere" and the negative values the southern "hemisphere". It is impossible to human minds to picture a four-dimensional hypersphere. Nevertheless, analogously to the (S^2) case, projecting all points that form the hypersphere surface to the intersection between a "space" passing through the equator and the hypersphere, then the northern and the southern "hemispheres" are going to create a filled ball each. As before, $q_0 = 0$ represents the "equator", being the only projection which represents precisely the original region on the hypersphere.



(a) Sphere (S^2)

(b) (S^2) represented by three (S^1)

Figura 7 – Representing a sphere (S^2) with one dimension lower.

Source: (HANSON, 2006)

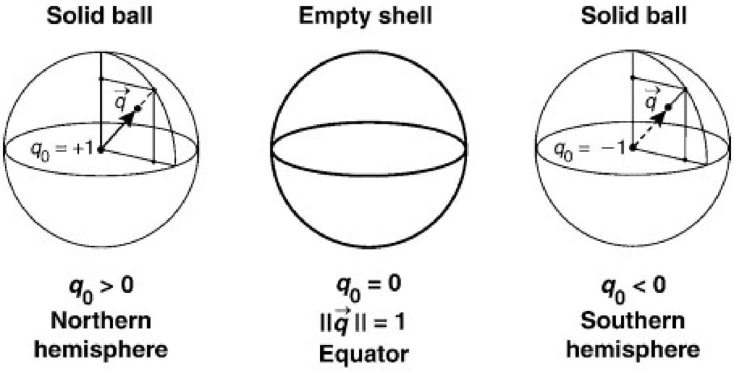


Figura 8 – Representing a hypersphere (S^3) with one dimension lower.

Source: (HANSON, 2006)

This way, even though it is humanly impossible to picture a four-dimensional sphere, this method allows the visualization of it, lessening its obscurity.

2.2.6 Quaternions Logarithm and Exponential

Dealing with logarithms and exponential for quaternions can be useful for numeric differentiation, interpolation or even to normalize unit quaternions, as it is going to be explained in the next section. Besides, once quaternions have an exponential representation, logarithms and exponential functions applied to them arise naturally.

Consider a quaternion as characterized in (2.12). Using its exponential representation, a logarithm applied to it is such that,

$$\ln(q) = \ln(||q||) + \frac{\theta}{2}\hat{\mathbf{n}}. \tag{2.26}$$

Therefore, for an unit quaternion one gets,

$$\ln\left(\hat{q}\right) = 0 + \frac{\theta}{2}\hat{\mathbf{n}}. \tag{2.27}$$

Now, also using a quaternion as characterized in (2.12), an exponential function applied to it results in

$$\begin{aligned} \exp\left(q\right) &= e^{(q_0)}e^{(q_1 i + q_2 j + q_3 k)} \\ \exp\left(q\right) &= e^{q_0}e^{||\mathbf{q}||\frac{\mathbf{q}}{||\mathbf{q}||}} \\ \exp\left(q\right) &= e^{q_0}e^{||\mathbf{q}||\hat{\mathbf{n}}}. \end{aligned} \tag{2.28}$$

Using the trigonometric form of (2.12) it is also possible to infer that

$$\exp\left(q\right) = e^{\left(||q|| \cos\frac{\theta}{2}\right)}e^{||q|| \operatorname{sen}\frac{\theta}{2}\hat{\mathbf{n}}}. \tag{2.29}$$

It is important to highlight that the identity property

$$f(f^{-1}(x)) = f^{-1}(f(x)) = x, \tag{2.30}$$

does not hold for all interval. Nevertheless, for this work, in the situations where it is needed, it is applied,

$$\exp\left(\ln\left(\hat{q}\right)\right) = \hat{q}, \tag{2.31}$$

which holds for the intervals considered.

2.2.7 Quaternions Differentiation

In order to transform the control references between the CS to the World Frame (WF), it is essential to define the mapping functions that perform the transformations between them and compute the Jacobian that describes such transformations in terms of time derivatives, as well as partial differentiation between quaternions. Thus, this section will present the most relevant expressions used in this thesis without proving. More detailed explanations can be obtained in (JIA, 2019).

Consider the case of a quaternion function $q(t)$ describing how the orientation of a moving object, represented by its body frame, varies with respect to a fixed WF. Let ω be the angular velocity of the body frame with respect to the WF. Therefore,

$$\dot{q} = \frac{1}{2}\omega q. \tag{2.32}$$

If $\dot{q}$ is known, then ω can be computed as

$$\omega = 2\dot{q}q^{*}. \tag{2.33}$$

Then, using (2.33), the second derivative is

$$\begin{aligned}\ddot{q} &= \frac{1}{2}\left(\dot{\omega}q + \omega\dot{q}\right)\\ \ddot{q} &= \frac{1}{2}\dot{\omega}q + \frac{1}{4}\omega\omega q\\ \ddot{q} &= \left(-\frac{1}{4}||\omega||^2 + \frac{1}{2}\dot{\omega}\right)q\end{aligned} \tag{2.34}$$

There can be also find the angular acceleration if the first and second time derivatives of q are both known. It can be shown that,

$$\dot{\omega} = 2\left(\ddot{q}q^* - \left(\dot{q}q^*\right)^2\right) \tag{2.35}$$

Finally, a possible numerical approach to compute the next quaternion using the existing information relies on Euler's method, in which,

$$q_{k+1} = q_k + \frac{1}{2}T\omega_k q_k, \tag{2.36}$$

where T represents the time step considered, q_k and ω_k are the quaternion and the angular velocity at the time kT, respectively, and q_{k+1} is the value of the quaternion at the next time step.

Due to computational iterations, the norm of q_{k+1}, considering that it has been used an unit quaternion, decreases until it becomes zero. A possible solution is to make

$$q_{k+1} \leftarrow \frac{q_{k+1}}{||q_{k+1}||} \tag{2.37}$$

after each iteration, in order to keep the norm equal to 1. That is a simple solution, though it also propagates errors, being harmful in long runs. An alternative is to plug q_{k+1} into (2.31), but pushing that

$$\ln\left(q_{k+1}\right) = \ln\left(||q_{k+1}||\right) + \frac{\theta_{k+1}}{2}\hat{\mathbf{n}} = 0 + \frac{\theta_{k+1}}{2}\hat{\mathbf{n}}, \tag{2.38}$$

once that this would be the expected result without numerical errors. The argument is proportionally less affected by the numeric errors. Therefore, after the exponential, the value found represents better what should be expected from q_{k+1}. In (ZHAO; WACHEM, 2013), those and some other methods for quaternion integration are presented and compared.

Now, for partial differentiation between quaternions, (JIA, 2019) presents straightforward solutions with no need for numerical approximations. Let us consider two quaternions q and p, such that

$$q \text{ and } p \in \mathbb{H} \;\mid\; q = q_0 + \mathbf{q} \text{ and } p = p_0 + \mathbf{p}. \tag{2.39}$$

Considering q and p as the column vectors $\begin{bmatrix} q_0 & q_1 & q_2 & q_3 \end{bmatrix}^\top$ and $\begin{bmatrix} p_0 & p_1 & p_2 & p_3 \end{bmatrix}^\top$, respectively, the derivative of $q \circ p$ with respect to q is,

$$\frac{\partial \left(q \circ p\right)}{\partial q} = \begin{bmatrix} p_0 & -\mathbf{p}^\top \\ \mathbf{p} & p_0 \mathbf{I_{3x3}} - \mathbf{p}\times \end{bmatrix}_{4x4} . \tag{2.40}$$

Now, the partial derivative of $q \circ p^*$ with respect to q is

$$\frac{\partial \left(q \circ p^*\right)}{\partial q} = \begin{bmatrix} p_0 & \mathbf{p}^\top \\ -\mathbf{p} & p_0 \mathbf{I_{3x3}} + \mathbf{p}\times \end{bmatrix}_{4x4} . \tag{2.41}$$

Similarly, the partial derivative of $q \circ p$ with respect to p is,

$$\frac{\partial \left(q \circ p\right)}{\partial p} = \begin{bmatrix} q_0 & -\mathbf{q}^\top \\ \mathbf{q} & q_0 \mathbf{I_{3x3}} + \mathbf{q}\times \end{bmatrix}_{4x4} . \tag{2.42}$$

Besides,

$$\frac{\partial \left(q \circ p^*\right)}{\partial p} = \begin{bmatrix} q_0 & \mathbf{q}^\top \\ \mathbf{q} & -q_0 \mathbf{I_{3x3}} - \mathbf{q}\times \end{bmatrix}_{4x4} , \tag{2.43}$$

where $\mathbf{I_{3x3}}$ is an unit matrix and $\mathbf{q}\times$ and $\mathbf{p}\times$ are 3×3 anti-symmetric matrices given by

$$\begin{aligned} \mathbf{q}\times &= \begin{bmatrix} 0 & -q_3 & q_2 \\ q_3 & 0 & -q_1 \\ -q_2 & q_1 & 0 \end{bmatrix}_{3x3} \quad \text{and} \\ \mathbf{p}\times &= \begin{bmatrix} 0 & -p_3 & p_2 \\ p_3 & 0 & -p_1 \\ -p_2 & p_1 & 0 \end{bmatrix}_{3x3} . \end{aligned} \tag{2.44}$$

At last, for the derivatives of the product $q \circ \mathbf{v} \circ q^*$ involving a quaternion q and the vector part $\mathbf{v}$ of another quaternion v, with respect to $\mathbf{v}$, the result is,

$$\frac{\partial \left(q \circ \mathbf{v} \circ q^*\right)}{\partial \mathbf{v}}_{3x3} = \left(q_0^2 - ||\mathbf{q}||^2\right) \mathbf{I_{3x3}} + 2\mathbf{q}\mathbf{q}^\top + 2q_0\mathbf{q}\times, \tag{2.45}$$

whereas,

$$\frac{\partial \left(q^* \circ \mathbf{v} \circ q\right)}{\partial \mathbf{v}}_{3x3} = \left(q_0^2 - ||\mathbf{q}||^2\right) \mathbf{I_{3x3}} + 2\mathbf{q}\mathbf{q}^\top - 2q_0\mathbf{q} \times . \tag{2.46}$$

By its turn, for the derivatives of the product $q \circ \mathbf{v} \circ q^*$ with respect to q the result is,

$$\frac{\partial \left(q \circ \mathbf{v} \circ q^*\right)}{\partial q}_{3x4} = 2\left(\left[\begin{array}{c:c} q_0\mathbf{v} + \mathbf{q} \times \mathbf{v} & -\mathbf{v}\mathbf{q}^\top + \left(\mathbf{v} \cdot \mathbf{q}\right) \mathbf{I_{3x3}} + \mathbf{q}\mathbf{v}^\top - q_0\mathbf{v}\times \end{array}\right]\right) \tag{2.47}$$

3 Experimental Setup

This chapter deals with the infrastructure used to run the experiments designed to validate the proposal here described, involving not only the formation description based on quaternions but also the controller designed to deal with such a description, both discussed in Chapter 4. It is detailed both the UGV and UAV used to accomplish the robot formation, as well as their kinematic and dynamic (when relevant) equations, in order to describe their behavior when flying. It is also detailed the motion capture system used to acquire the data correspondent to the positions of the robots in the formation, thus getting the current state of the formation, as well as the software needed in order to make it operational. Moreover, it is discussed some restrictions imposed by the space limitation of the flying arena, as well as the equipment used for processing the algorithms and communication between the robots and the ground station where the control system runs.

3.1 Unmanned Mobile Robots

The main characteristics of the two platforms used to run the validating experiments for the system here proposed are considerably relevant for the understanding of the context in which this solution has been elaborated. Without this contextualization, it becomes more difficult for future designs or researches to apply this proposal (or aspects of it) in their solutions, if desired. Thus, the main characteristics and considerations made about the two platforms are here presented.

3.1.1 Unmanned Ground Vehicle: Pioneer 3-DX

As mentioned in Chapter 1, the UGV considered for this work is the *Pioneer 3-DX*, an unicycle-like nonholonomic platform. As explained in (MOREIRA, 2020), it has two independently driven wheels, positioned at its sides, together with a caster wheel to support its equilibrium, located at its backside. This configuration allows it to move forward and backward, as well as around its own vertical axis keeping the center of the line section between both wheels unchanged. It is also capable of performing combinations of linear and angular velocities, resulting in motions described by circumference arcs, by applying different velocity values to each wheel, as exemplified in Figure 9. Due to such characteristics, it is activated by linear velocity commands in its x^{p}[1] axis and by angular velocity commands in its z^p axis.

[1] Superscript p indicates Pioneer's frame

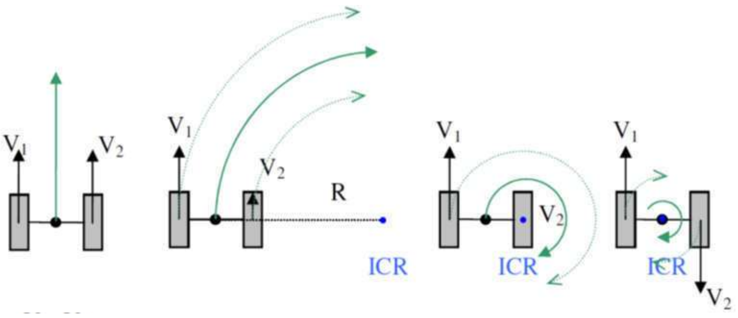


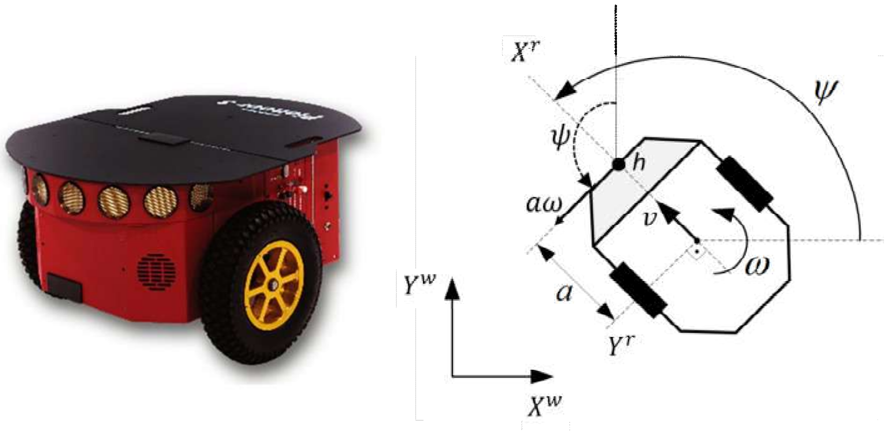
Figura 9 – Pioneer 3-DX Mobility (ICR means Instantaneous Center of Rotation)

Therefore, the velocity commands sent to the UGV can be expressed as

$$\mathbf{u} = \begin{bmatrix} v & \omega \end{bmatrix}^{\top}. \tag{3.1}$$

where v and ω represent the instantaneous linear and angular velocity commands, respectively.

As for the kinematic model that represents the Pioneer 3-DX UGV, it depends on the point of interest for control chosen, which is located in a certain distance a from the center of the virtual axis connecting the two driven wheels, as represented in Figure 10.



(a) UGV used for experiments (b) Pioneer 2D Model

Figura 10 – Kinematic Model Representation for the UGV.

For such a configuration, in order to convert velocities commands between the World Frame (WF) and the Pioneer Frame (PF), it is used the direct kinematic model of an unicycle-like robot, given by

$$\mathbf{A}_p = \begin{bmatrix} \cos\psi & -a\,\mathrm{sen}\psi \\ \mathrm{sen}\psi & a\cos\psi \end{bmatrix}, \tag{3.2}$$

whereas to convert from the WF to the PF it is used its inverse, given by

$$\mathbf{A}_p^{-1} = \begin{bmatrix} \cos\psi & \mathrm{sen}\psi \\ -\frac{\mathrm{sen}\psi}{a} & \frac{\cos\psi}{a} \end{bmatrix}. \tag{3.3}$$

In matrix notation, the transformation from the PF to the WF is written as

$$\begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} = \mathbf{A}_p \begin{bmatrix} v \\ \omega \end{bmatrix}, \tag{3.4}$$

and the transformation from the WF to the PF is written as

$$\begin{bmatrix} v \\ \omega \end{bmatrix} = \mathbf{A}_p^{-1} \begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix}, \tag{3.5}$$

where $\begin{bmatrix} \dot{x} & \dot{y} \end{bmatrix}^\top$ are the velocities in the WF.

3.1.2 Unmanned Aerial Vehicle: Bebop 2

As also mentioned in Chapter 1, the UAV used in this work is the quadrotor *Bebop 2*, from Parrot Drones SAS. It is equipped with a full HD eye-fish 14 mega-pixels frontal camera and an optical stabilizer, besides a second camera, which faces down (the images acquired by such cameras can be accessed through the software provided by the manufacturer). It also has a dualcore CPU processor and an embedded quadcore GPU. It also has embedded instruments, such as altitude sensors, GPS module, accelerometers, battery charge level sensor, such that it is able to deliver information about the current state of the vehicle, updated each 200 ms interval.

In order to establish communication with the *Bebop 2*, it was used the *Bebop_Autonomy* driver, alongside with the Robot Operating System (ROS). The *Bebop_Autonomy* driver was developed at the Autonomy Lab from the Simon Fraser University, in C++. It offers a range of topics and resources which allow the user to access internal data from the *Bebop 2*, as well as send commands to it. Subsection 3.3 will provide more details about ROS.

From the operational point of view, the UAV has 6 Degrees of Freedom (6 DoF). It can activate linear velocity commands in all three axes x^{b}[2], y^b and z^b, as well as angular velocity commands around the same axes. These angular velocity commands correspond to the Roll, Pitch and Yaw movements, represented by the angles ϕ, θ and ψ, respectively. Combinations of such movements are performed by controlling the forces generated by each one of the rotors, as indicated in Figure 11.

Roll and Pitch movements are not of interest in this work, however, for not being necessary for the applications considered, and due to the risk involved in performing such movements in an indoor environment. For similar reasons, the quadrotor *Bebop 2* was configured to have small bound limits for linear and angular velocities along and around its z^b axis. Furthermore,

[2] Superscript b indicates Bebop's frame

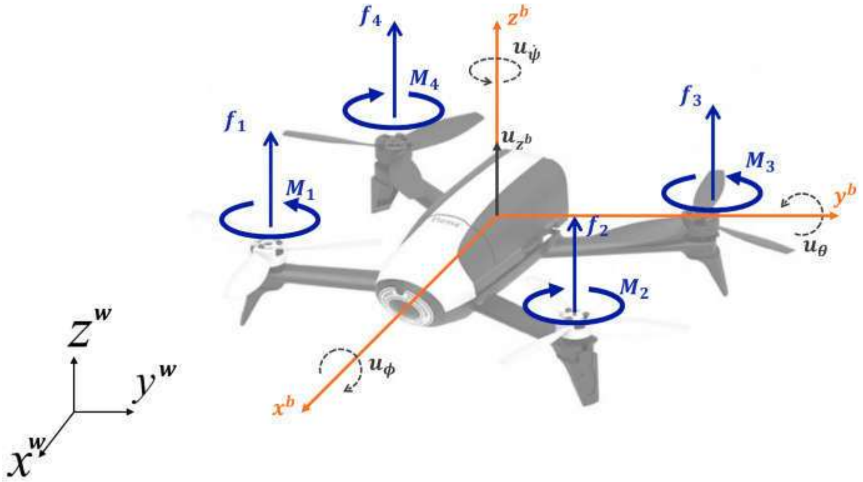


Figura 11 – Bebop 2 Mobility

the Roll and Pitch movements were limited to ±5°, so that it was possible to model the UAV as a fully actuated system (see (Pinto et al., 2020)), whose velocity commands are

$$\mathbf{u} = \begin{bmatrix} u_{v_x} & u_{v_y} & u_{\dot{z}} & u_{\dot{\psi}} \end{bmatrix}^\top, \tag{3.6}$$

without considerable loss in the precision of the model.

Likewise for the Pioneer 3-DX, the kinematic model of the *Bebop 2* also considers it as a mass-less virtual point which can respond to the velocity commands instantaneously. Previous experiments, however, demonstrated that only a kinematic model (and controller) was not enough to ensure an acceptable performance for the UAV considered. Hence, a dynamic model was also used for the *Bebop 2*, but not considered for the Pioneer 3-DX, once it has shown acceptable performance when applying just a kinematic controller. Thus, its dynamics, for the objectives of this thesis, is not considered.

Taking into account only the four DoF specified in (3.6), and considering that the Pitch and Roll movements are small enough so that they may be ignored when the UAV moves alongside the x^b and y^b axes (see (Pinto et al., 2020)), its matrix of direct kinematics is

$$\mathbf{A}_b = \begin{bmatrix} \cos\psi & -\operatorname{sen}\psi & 0 & 0 \\ \operatorname{sen}\psi & \cos\psi & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{3.7}$$

and its matrix of inverse kinematics is

$$\mathbf{A}_b^{-1} = \begin{bmatrix} \cos\psi & \operatorname{sen}\psi & 0 & 0 \\ -\operatorname{sen}\psi & \cos\psi & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{3.8}$$

so that the transformation of the velocities from the Bebop Frame (BF) to the WF is written as

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ \dot{\psi} \end{bmatrix} = \mathbf{A}_b \begin{bmatrix} u_{\dot{x}} \\ u_{\dot{y}} \\ u_{\dot{z}} \\ u_{\dot{\psi}} \end{bmatrix}, \tag{3.9}$$

and the transformation from the WF to the PF is written as

$$\begin{bmatrix} u_{\dot{x}} \\ u_{\dot{y}} \\ u_{\dot{z}} \\ u_{\dot{\psi}} \end{bmatrix} = \mathbf{A}_b^{-1} \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ \dot{\psi} \end{bmatrix}, \tag{3.10}$$

where $\begin{bmatrix} \dot{x} & \dot{y} & \dot{z} & \dot{\psi} \end{bmatrix}^\top$ are the linear velocities and the yaw velocity, in the WF.

Now, taking into account the dynamics of the UAV, a new model is considered, described as (SANTANA; BRANDãO; SARCINELLI-FILHO, 2016)

$$\ddot{\mathbf{X}} = \mathbf{f}_1\mathbf{u}_d - \mathbf{f}_2\dot{\mathbf{X}}, \tag{3.11}$$

where,

$$\mathbf{f}_1 = \begin{bmatrix} \cos\psi & -\,\text{sen}\psi & 0 & 0 \\ \text{sen}\psi & \cos\psi & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} K_1 & 0 & 0 & 0 \\ 0 & K_3 & 0 & 0 \\ 0 & 0 & K_5 & 0 \\ 0 & 0 & 0 & K_7 \end{bmatrix} = \mathbf{A}_b\mathbf{K}_1, \tag{3.12}$$

$$\mathbf{f}_2 = \begin{bmatrix} \cos\psi & -\,\text{sen}\psi & 0 & 0 \\ \text{sen}\psi & \cos\psi & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} K_2 & 0 & 0 & 0 \\ 0 & K_4 & 0 & 0 \\ 0 & 0 & K_6 & 0 \\ 0 & 0 & 0 & K_8 \end{bmatrix} = \mathbf{A}_b\mathbf{K}_2, \tag{3.13}$$

$\dot{\mathbf{X}} = \begin{bmatrix} \dot{x} & \dot{y} & \dot{z} & \dot{\psi} \end{bmatrix}^\top$, $\ddot{\mathbf{X}} = \begin{bmatrix} \ddot{x} & \ddot{y} & \ddot{z} & \ddot{\psi} \end{bmatrix}^\top$ and $\mathbf{u}_d$ is a vector with the control signals sent to the UAV. As for the parameters $K_1, \cdots, K_8$, they are estimated using experimental values gathered according to the procedure described in (Pinto et al., 2020). The values thus obtained for such parameters, which are the values used in this thesis, are presented in Table 2.

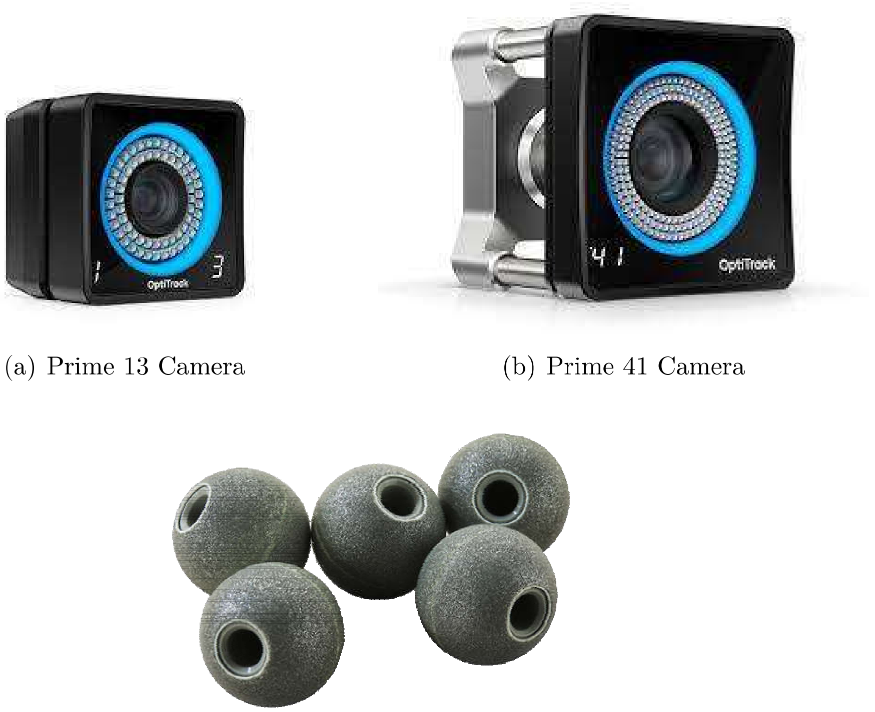
3.2 Motion Capture System Optitrack

As mentioned in (MOREIRA, 2020), the Motion Capture System *Optitrack* consists in a proprietary source three-dimensional localization system, commercially sold by Natural

Tabela 2 – Parameters Values from the Identified Dynamic Model

K_1	K_2	K_3	K_4	K_5	K_6	K_7	K_8
0.8417	0.1823	0.8354	0.1710	3.966	4.001	9.8524	4.7295

Point, Inc. This system assembles a virtual rigid body through the detection of retro-reflective capture markers in the body of interest. The three-dimensional location is then figured out by forming a high luminosity blob around the markers using images captured by cameras installed around the laboratory. Those cameras, as well as the retro-reflective capture markers are shown in Figure 12.



(a) Prime 13 Camera

(b) Prime 41 Camera

(c) Retro-reflective Capture Markers

Figura 12 – *Optitrack* System Cameras and Markers.

In this work the *Optitrack* configuration corresponds to eight cameras, four of them of model Prime 13 and the other four of model Prime 41. The difference between them is that the Prime 41 model has a greater angular opening, in comparison with the Prime 13 model. The retro-reflective capture markers are designed to reflect incoming light back to its source. IR light emitted from the cameras is reflected by passive markers and detected by the camera's sensor[3].

[3] See <https://v22.wiki.optitrack.com/index.php?title=Markers>

For each rigid body whose movement should be tracked, at least four markers were used, placed asymmetrically over the object, in order to ensure that the software could distinguish one from another, in this case the UGV and the UAV. Both robots with their markers are shown in Figure 13.

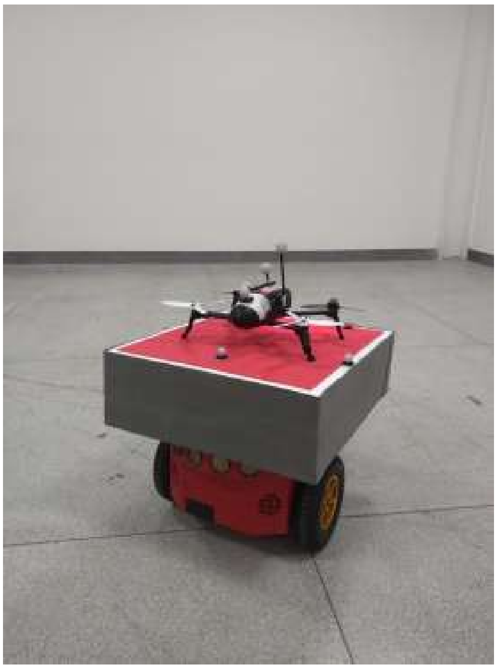


Figura 13 – Pioneer 3-DX and Bebop 2 with their *Optitrack* markers

Finally, the software used to capture the images and to allow the object tracking is the *Motive*, also a property of NaturalPoint, Inc. This software provides an interface where the user can define the rigid body formed by the markers detected and save or transmit this data through the network available. The *Motive* software computes the 3D positions of the bodies using the 2D images provided by multiple cameras positioned throughout the room, in such a way that it is possible to triangulate the bodies 3D position with the minimum of blind spots possible. It is also capable to track 6 DoF from the rigid body measured, being three positioning coordinates and three orientation coordinates. It is also possible to acquire the orientation already as a quaternion, which was the choice made in this work.

3.3 Robot Operating System (ROS)

ROS is an operational system specified for robots usage, being a flexible open source framework, which can accomplish the low level communication among a wide variety of robots. It is composed by a set of libraries and tools which enable the development of more complex behaviors for the robots, individually or as a group.

Broadly speaking, the ROS operates through a central server named *Master Node*, which is responsible to coordinate all communication among the other nodes in the network. This *Master Node* has the information about the elements that compose the network, as well as all published and received messages. Any new node informs the *Master Node* about its topics and which messages it publishes or listens. If a node needs some information from the network, it requires from the *Master Node* a signature requisition or a topic subscription. If that information is available in the network, the *Master Node* informs to the requesting node the address where that information is published, and the link between both nodes is established.

In this work, it was used a Linux server to run the ROS *Master Node*, alongside with the *Bebop 2* communication driver, the *Bebop_Autonomy* library aforementioned. The codes are written in C++ and Python, and are responsible for the low level communication between the computer running the *Master Node* and the quadrotor.

As for the communication between the *Master Node* and the *Pioneer 3-DX* UGV, it is performed using a *Raspberry PI 3* model B mini-computer, installed over the *Pioneer 3-DX*, in which the *Ubuntu* Operating System is installed. Such a resource is used to run the ROS and the communication nodes connecting the UGV and the *Windows 10* Operating System computer.

Furthermore, there is a Listener/Publisher node, which is responsible to control both robots, the UGV and the UAV. It runs the control loop, as well as the interruption procedures required to ensure the safety of the assets and personnel during the experiments. This node runs in another computer, over the *Windows 10* Operating System with the *Matlab R2018a* software installed. Inside the *Matlab* script it is created the node which establishes the communication with the Linux server. This way, the necessary communication between the two robots and the computer running the control system is established.

It was used the *ROS-Kinect Kame* version for this work, installed in the 16.04 version of the *Ubuntu* Operating System. There are more recent versions for ROS and for the *Ubuntu* Operating System. However, until the moment in which this thesis has been written, the used versions were required in order to guarantee the compatibility with the *Bebop_Autonomy* communication driver, embedded in the *Bebop 2* quadrotor.

3.4 Space Limitations

The testing environment is a well illuminated, clean, predominant white room, with blocking-light curtains in all windows, so that the external luminosity does not disturb the *Optitrack* measurements. The room is 6 meters long, 5 meters wide and has 3 meters of height. Taking 1 meter from the width for the chairs and tables, it can be considered that the flight arena corresponds to an area of $6 \times 4 = 24m^2$, correspondent to a useful volume of $24 \times 3 = 72m^3$.

This space is rather limiting, regarding the navigation of the UGV-UAV formation. Hence the experiments were run considering trajectories with moderated speeds $\dot{x}_d$, and $\dot{y}_d$. Moreover, for some cases the maximum velocity allowed for the UAV was limited, as it is going to be explained in Chapter 5.

3.5 Infra-Structure Overview

The whole infra-structure can be then listed as follows, considering all aspects aforementioned.

- *Pioneer 3-DX* UGV;
- *Bebop 2* UAV;
- *Optitrack* motion capture system, with 8 cameras connected to a DELL XPS *Windows 10* desktop computer, with an INTEL Core I7 processor, 24GB of RAM memory, 500GB of SSD disc and shared GPU, running the softwares *Motive*[4] and *Matlab*;
- Samsung Notebook with an INTEL Core I7 processor, 8GB of RAM DDR4 memory and a rigid disc of 1TB, operating with *Linux*;
- Tplink Gigabit Model router with wi-fi transmission of 2.4GHz and 5GHz;
- Gigabit Switch with a PoE feeding capability;
- XBOX360 Joystick, computer model.

Motive is a proprietary software of Natural Point Inc., the provider of the *OptiTrack* system, responsible for getting the instantaneous positions of the UGV and the UAV and their relative orientation, this last one as a quaternion, as aforementioned. As for the *Matlab*, it is necessary to run the script correspondent to the control system. Finally, the notebook is necessary to run the ROS system to create the communication structure linking the computer running the control system and the two robots.

It is worth mentioning that the joystick was used to overwrite the commands sent by the controller if necessary, characterizing a safety controller for emergencies. An overview of the whole setup is shown in Figure 14.

[4] See <https://optitrack.com/software/>

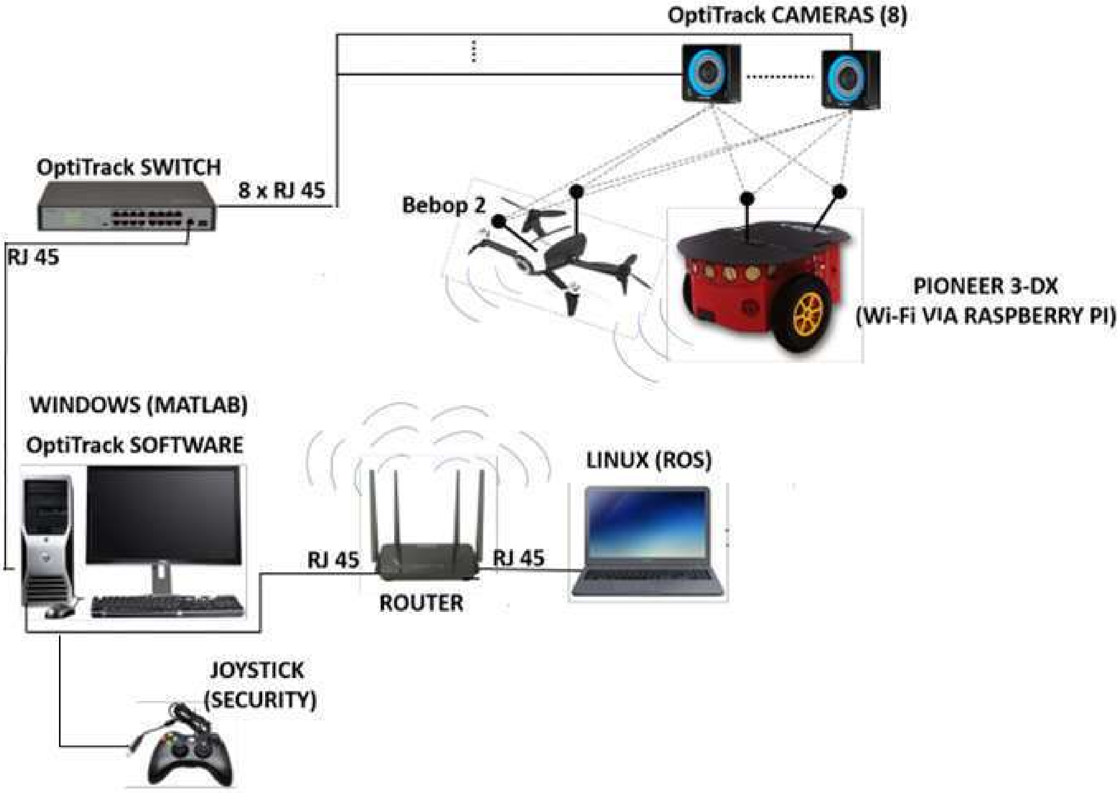


Figura 14 – Experimental Setup Adopted

4 Formation Description and Controller Design

In this chapter, two Euclidean descriptions of the UGV-UAV formation are introduced, for the sake of comparison with the quaternion-based description here proposed. All these formations consider the virtual structure control paradigm, where the three-dimensional line segment connecting both robots is the virtual structure to be controlled. They also consider a trajectory tracking task, with the possibility that the UAV lands on the UGV. Therefore, the UAV hovers over the UGV for a while, at certain point of the experiments, just before starting falling down on the UGV (this moment characterizes the specific formation configuration being explored in this work).

4.1 Euclidean Descriptions of the UGV-UAV Formation

Euclidean descriptions for an UGV-UAV formation are usually formulated considering the three-dimensional position of the robots in space, the distance between them and the angles defined taking in account the virtual structure and predefined axes or planes. This is a straightforward approach, for being easily visualized and having a simpler geometry than most descriptions. Nonetheless, Euclidean descriptions are susceptible to formation singularities, depending on how they are described. In Subsection 4.1.1 it is presented an Euclidean description in which the singularity appears in the formation configuration exactly when the UAV hovers over the UGV. On the other hand, in Subsection 4.1.2 it is presented a possible solution, still considering an Euclidean description, for the problem of singularity, with the downside that restrictions are imposed to the CS.

4.1.1 Singularity arising when the UAV hovers over the UGV

In (RABELO; BRANDãO; SARCINELLI-FILHO, 2021) it is used an Euclidean description for an application in which the UAV should land on a static or moving platform, considering as virtual structure the line segment connecting both robots, in a trajectory tracking task. The infrastructure adopted to run the validation was similar to the one used here (positions were gathered using the *Optitrack* system, but using the *AR.Drone 2.0* quadrotor as the UAV).

As for the cluster characterization, such work considers six variables: the three-dimensional position of the point of interest for control of the cluster $\mathbf{x}_c = \begin{bmatrix} x_c & y_c & z_c \end{bmatrix}^\top$, the distance between both robots ρ_c, the angle between the x^{w}[1] axis and the projection of the virtual line over the xy^w plane α_c and the angle between the virtual line and the xy^w plane β_c. Therefore, the cluster space is represented by the variables $\mathbf{c} = \begin{bmatrix} x_c & y_c & z_c & \rho_c & \alpha_c & \beta_c \end{bmatrix}^\top$. Figure 15 summarizes this formulation, considering the *Bebop 2* quadrotor as the UAV.

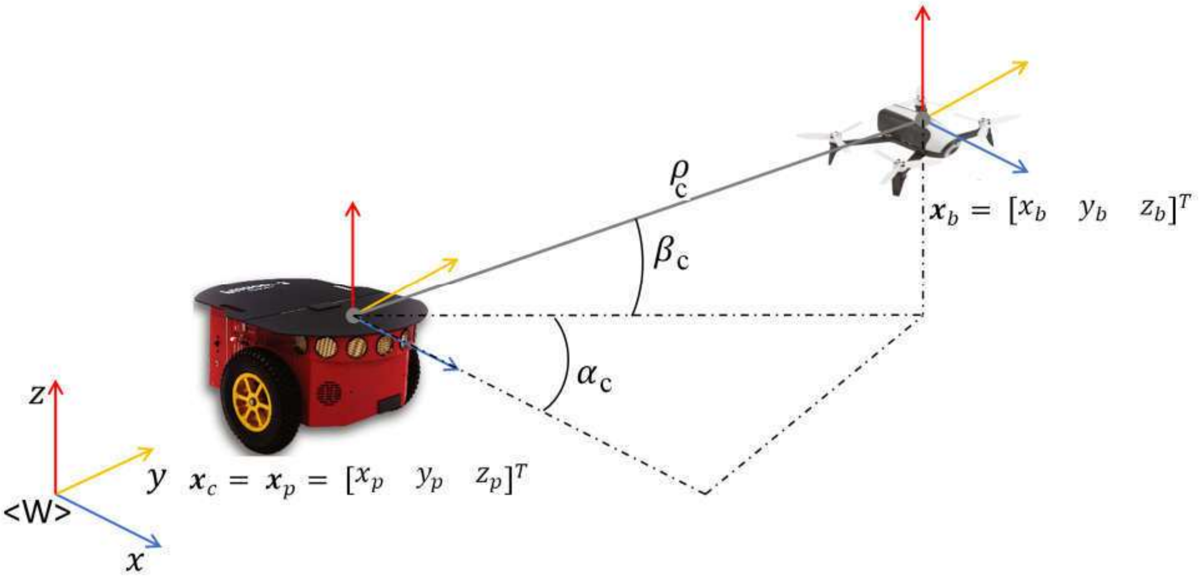

ρ_c
β_c
$\boldsymbol{x}_b = [x_b \quad y_b \quad z_b]^T$
α_c
z
y
<W>
x
$\boldsymbol{x}_c = \boldsymbol{x}_p = [x_p \quad y_p \quad z_p]^T$

Figura 15 – Euclidean Description for Virtual Structure Paradigm, as in (RABELO; BRANDãO; SARCINELLI-FILHO, 2021)

In such a figure $\mathbf{x}_p = \begin{bmatrix} x_p & y_p & z_p \end{bmatrix}^\top$ and $\mathbf{x}_b = \begin{bmatrix} x_b & y_b & z_b \end{bmatrix}^\top$ are the *Pioneer 3-DX* and *Bebop 2* three-dimensional positions, respectively. Note that the point of interest for control $\mathbf{x}_c$ is the same as the *Pioneer 3-DX* position, such that $\mathbf{x}_c = \mathbf{x}_p$. The transformation mapping the robots space into the cluster space, for this description, called direct or forward transformation, is characterized as

$$x_c = x_p, \tag{4.1}$$

$$y_c = y_p, \tag{4.2}$$

$$z_c = z_p, \tag{4.3}$$

$$\rho_c = \sqrt{(x_b - x_p)^2 + (y_b - y_p)^2 + (z_b - z_p)^2}, \tag{4.4}$$

$$\alpha_c = \arctan\left(\frac{y_b - y_p}{x_b - x_p}\right), \tag{4.5}$$

[1] Superscript w refers to the World frame

$$\beta_c = \arctan\left(\frac{z_b - z_p}{\sqrt{(x_b - x_p)^2 + (y_b - y_p)^2}}\right), \tag{4.6}$$

whereas the inverse transformation, mapping the cluster space into the RS, is given by

$$x_p = x_c, \tag{4.7}$$

$$y_p = y_c, \tag{4.8}$$

$$z_p = z_c, \tag{4.9}$$

$$x_b = x_c + \rho_c \cos(\alpha_c) \cos(\beta_c) \tag{4.10}$$

$$y_b = y_c + \rho_c \,\mathrm{sen}(\alpha_c) \cos(\beta_c) \tag{4.11}$$

$$z_b = z_c + \rho_c \,\mathrm{sen}(\beta_c) \tag{4.12}$$

In (4.10) and (4.11) one can observe that when $\beta_c = 90°$ there is no projection of the virtual straight line of length ρ_c over the xy^w plane, which characterizes a formation singularity. Nonetheless, this configuration represents exactly the formation required to start the procedure for the landing of the UAV on the UGV. Actually, this configuration might be desired even when the UAV is not supposed to land on the UGV (it might be desired for a task of placing a package over the UGV, for instance). A consequence of this singularity is that α_c becomes undefined and may cause an unpredictable oscillation. This makes the UAV to oscillate as well, what is harmful to applications where stability is a requirement, such as to land or to deliver a package.

4.1.2 Limited cluster space due to the Euclidean representation chosen

In (ERNANDES-NETO; SARCINELLI-FILHO; BRANDÃO, 2019), by its turn, it is proposed an Euclidean description formulated in such a way that singularities do not arise when the UAV hovers over the UGV. Nevertheless, this is accomplished by restricting the CS attainability. This effect can be better perceived by observing the transformations between robots space to cluster space and cluster space to robots space associated to this description. Figure 15 summarizes this formulation.

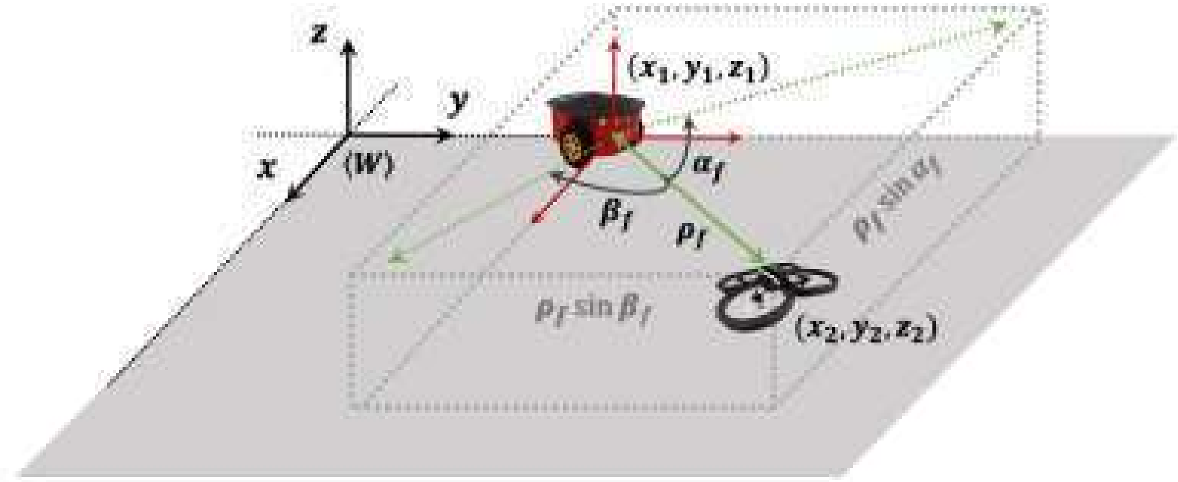


Figura 16 – Euclidean Description for Virtual Structure Paradigm with no Singularity when the UAV Hovers over the UGV (ERNANDES-NETO; SARCINELLI-FILHO; BRANDÃO, 2019)

Using the same notation adopted in (ERNANDES-NETO; SARCINELLI-FILHO; BRANDÃO, 2019), the direct transformation is characterized as

$$x_f = x_1, \tag{4.13}$$

$$y_f = y_1, \tag{4.14}$$

$$z_f = z_1, \tag{4.15}$$

$$\rho_f = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2}, \tag{4.16}$$

$$\alpha_f = \text{sen}^{-1}\left(\frac{x_2 - x_1}{\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2}}\right), \tag{4.17}$$

$$\beta_f = \text{sen}^{-1}\left(\frac{y_2 - y_1}{\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2}}\right), \tag{4.18}$$

whereas the inverse transformation is given by

$$x_1 = x_f, \tag{4.19}$$

$$y_1 = y_f, \tag{4.20}$$

$$z_1 = z_f, \tag{4.21}$$

$$x_2 = x_f + \rho_f \operatorname{sen}(\alpha_f), \tag{4.22}$$

$$y_2 = y_f + \rho_f \operatorname{sen}(\beta_f), \tag{4.23}$$

$$z_2 = z_f + \rho_f \sqrt{1 - \operatorname{sen}^2(\alpha_f) - \operatorname{sen}^2(\beta_f)}, \tag{4.24}$$

where $\mathbf{x}_f = \begin{bmatrix} x_f & y_f & z_f \end{bmatrix}^\top$ represents the point of interest for control, $\mathbf{x}_1 = \begin{bmatrix} x_1 & y_1 & z_1 \end{bmatrix}^\top$ represents the *Pioneer 3-DX* position, $\mathbf{x}_2 = \begin{bmatrix} x_2 & y_2 & z_2 \end{bmatrix}^\top$ represents the *AR.Drone 2.0* position (this was the quadrotor used in that work), ρ_f represents the distance between both robots and α_f and β_f are the angles used to describe this formation.

Notice that (4.22), (4.23) and (4.24) imply that when $\alpha_f = \beta_f = 0°$ the UAV position is $\mathbf{x}_2 = \begin{bmatrix} x_1 & y_1 & z_1 + \rho_f \end{bmatrix}^\top$. Therefore, the UAV will be hovering the UGV without singularities, solving any issue related to it for this configuration. In the other hand, one can observe by (4.24) that α_f and β_f are limited by each other. The maximum value that α_f may assume is the complement of β_f and vice-versa. Taking an extreme case, if $\alpha_f = 90°$ precisely, then β_f will have to be precisely 0°, otherwise z_2 will become a complex number, what is not compatible with its own nature, since it is a Cartesian coordinate (this also characterizes a singularity).

In order to avoid this sort of problem, for this Euclidean description, it has been considered a restricted range of variation for α_f and β_f, such that $0° \leq \alpha_f \leq 45°$ and $0° \leq \beta_f \leq 45°$. The landing platform over the UGV is considered to be the level 0 of altitude, such that with the α_f and β_f restricted range it is not possible for the UAV to navigate at the same level of the landing platform, if desired. Therefore, this description solves the singularity issue of the first description (during the landing procedure) by moving it elsewhere. Even though it might be enough for several applications, it is not ideal, once the Formation Space (FS) or CS attainability is compromised.

4.2 Unit Quaternion-Based Description of the UGV-UAV Formation

The quaternion based description for this formation is capable to solve the singularity issue, without losses in the attainability of the CS. It was adapted from the description given in (MAS; KITTS, 2017), where a formation involving two UAVs is considered, rather than the heterogeneous formation here proposed. The formulations established for the UAVs can be easily adapted to the UGV used in this work, once it behaves similarly to an UAV, although subject to some restrictions (the UGV does not move along the z^p and the y^p axes).

Furthermore, in (MAS; KITTS, 2017) it is also considered the virtual structure paradigm, with the virtual structure being the line segment connecting both robots, the same as here, but the point of interest for control is located in the center of such line. For the description here proposed, the point of interest for control is changed to coincide with the *Pioneer 3-DX* current position, to make easier to land the UAV on the UGV. Figure 17 summarizes this formulation.

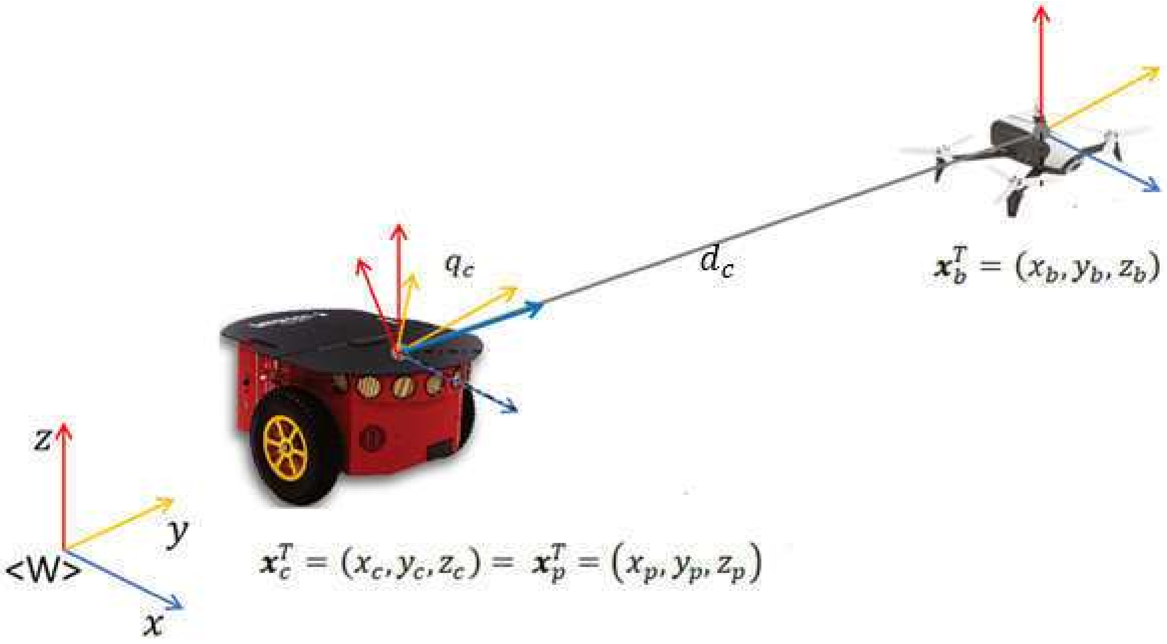

q_c
d_c
$x_b^T = (x_b, y_b, z_b)$
z
y
<W>
x
$x_c^T = (x_c, y_c, z_c) = x_p^T = (x_p, y_p, z_p)$

Figura 17 – Quaternion Based Description for Virtual Structure Paradigm

Taking in account all these considerations, the direct transformation (from the robots space to the cluster space) for the quaternion-based description is characterized as

$$\mathbf{x}_c = \mathbf{x}_p, \tag{4.25}$$

$$\hat{q}_c = [\cos\frac{\eta}{2}\ ;\ \text{sen}\frac{\eta}{2}\hat{\mathbf{n}}],\ \text{and} \tag{4.26}$$

$$d_c = ||\mathbf{x_b} - \mathbf{x_p}||, \tag{4.27}$$

where

$$\eta = atan2\left(\frac{||\hat{i} \times (\mathbf{x_b} - \mathbf{x_p})||}{\hat{i} \cdot (\mathbf{x_b} - \mathbf{x_p})}\right),\ \text{and} \tag{4.28}$$

$$\mathbf{\hat{n}_{bp}} = \frac{\hat{\imath} \times (\mathbf{x_b} - \mathbf{x_p})}{||\hat{\imath} \times (\mathbf{x_b} - \mathbf{x_p})||}, \tag{4.29}$$

with $\hat{\imath} = \begin{bmatrix}1, & 0, & 0\end{bmatrix}^\top$.

As for the inverse transformation (from the cluster space to the robots space), it is characterized as

$$\mathbf{x}_p = \mathbf{x}_c, \text{ and} \tag{4.30}$$

$$\mathbf{x}_b = \mathbf{x}_c + \hat{q}_c \circ \mathbf{d}_c \circ \hat{q}_c^*, \tag{4.31}$$

where $\mathbf{x}_c$, $\mathbf{x}_p$, $\mathbf{x}_b$ and $\mathbf{d}_c$ are pure vector quaternions, such that $\mathbf{x}_c = (0,\ x_c i,\ y_c j,\ z_c k)$, $\mathbf{x}_p = (0,\ x_p i,\ y_p j,\ z_p k)$, $\mathbf{x}_b = (0,\ x_b i,\ y_b j,\ z_b k)$ and $\mathbf{d}_c = (0,\ d_c i,\ 0,\ 0)$. As for $\hat{q}_c^*$, it is the conjugated form of $\hat{q}_c$, defined as $\hat{q}_c^* = (q_{c0},\ -\mathbf{q_c})$. Besides, $\mathbf{x}_c$ is the cluster point of interest for control, $\mathbf{x}_p$ is the *Pioneer 3-DX* position, $\mathbf{x}_b$ is the *Bebop 2* position and d_c is the scalar distance between both robots. Moreover, $\mathbf{\hat{n}_{bp}}$ is a normalized vector orthogonal to the x^w axis and the line section connecting the UAV and the UGV, coming from the UGV to the UAV. Finally, η is the angle between the x^w axis and the line section connecting both robots, while $\hat{q}_c$ is an unit quaternion used to orientate such a line.

Interestingly, if $(\mathbf{x_b} - \mathbf{x_p}) = \hat{\imath} = (1,\ 0,\ 0)$, meaning that the line section is aligned with the x^w axis and both robots are at the same level, it would imply that $\mathbf{\hat{n}_{bp}} = (0,\ 0,\ 0)$[2] and $\eta = 0°$, therefore $\hat{q}_c = 1 + 0i + 0j + 0k$. This value represents the identity quaternion and is orthogonal to all 3D space. If a purely three-dimensional description had been used, it would be impossible to define a single vector orthogonal to $(\mathbf{x_b} - \mathbf{x_p})$ and $\hat{\imath} = (1,\ 0,\ 0)$, once they are the same in this case. In fact, all vectors coming from the origin inside the $y^w z^w$ plane would be orthogonal to them, implying in an undefined value. Quaternions, with their four-dimensional characteristic, allow an unique[3] definition for an orthogonal element to both vectors.

4.3 The Unit Quaternion-Based Controller

Once the direct and inverse transformations between the CS and the WF have been defined, it is possible to design a controller based on the quaternion description here adopted.

[2] By (4.29), there would be a division by 0. This is fixed by imposing $||\hat{\imath} \times (\mathbf{x_b} - \mathbf{x_p})|| = 1$, when it should be 0.

[3] Actually, there are two: $\hat{q}_c = -1 + 0i + 0j + 0k$ is also orthogonal to the x^w axis, but it would imply in $\eta = 360°$, which is the same orientation. As mentioned in Subsection 2.2.3, quaternions make a double covering of the 3D space. In this work, $\hat{q}_c$ is never converted from the quaternion domain to the 3D Euclidean domain, therefore no error due misrepresentation occurs.

The current position of each robot is measured in the WF and the current state of the Cluster is then computed through (4.25), (4.26) and (4.27), such that

$$\mathbf{c} = \begin{bmatrix} \mathbf{x_c} & \hat{q}_c & \mathbf{d}_c \end{bmatrix}^{\top}_{10\times 1}. \tag{4.32}$$

The velocity references are computed in the CS and then converted to the WF by the inverse Jacobian matrix, defined as

$$\mathbf{J}^{-1}(\mathbf{c}) = \left(\frac{\partial \mathbf{x}}{\partial \mathbf{c}} \right), \text{ or} \tag{4.33}$$

$$\mathbf{J}^{-1}(\mathbf{c}) = \begin{bmatrix} \frac{\partial \mathbf{x}_p}{\partial \mathbf{x}_c} & \frac{\partial \mathbf{x}_p}{\partial q_c} & \frac{\partial \mathbf{x}_p}{\partial \mathbf{d}_c} \\ \frac{\partial \mathbf{x}_b}{\partial \mathbf{x}_c} & \frac{\partial \mathbf{x}_b}{\partial q_c} & \frac{\partial \mathbf{x}_b}{\partial \mathbf{d}_c} \end{bmatrix}_{6x10}, \tag{4.34}$$

where $\frac{\partial \mathbf{x_p}}{\partial \mathbf{x_c}} = \frac{\partial \mathbf{x_b}}{\partial \mathbf{x_c}} = \mathbf{I_{3x3}}$, $\frac{\partial \mathbf{x_p}}{\partial q_c} = \mathbf{0_{3x4}}$ and $\frac{\partial \mathbf{x_p}}{\partial \mathbf{d_c}} = \mathbf{0_{3x3}}$. As for $\frac{\partial \mathbf{x_b}}{\partial q_c}$ and $\frac{\partial \mathbf{x_b}}{\partial \mathbf{d_c}}$,

$$\frac{\partial \mathbf{x_b}}{\partial q_c}_{3x4} = \frac{\partial \mathbf{x_c}}{\partial q_c} + \frac{\partial \left(q_c \mathbf{d_c} q_c^* \right)}{\partial q_c}, \tag{4.35}$$

where

$$\frac{\partial \mathbf{x_c}}{\partial q_c}_{3x4} = \mathbf{0_{3x4}} \tag{4.36}$$

and $\frac{\partial (q_c \mathbf{d_c} q_c^*)}{\partial q_c}$ can be obtained through (2.47), while

$$\frac{\partial \mathbf{x_b}}{\partial \mathbf{d_c}}_{3x3} = \frac{\partial \mathbf{x_c}}{\partial \mathbf{d_c}} + \frac{\partial \left(q_c \mathbf{d_c} q_c^* \right)}{\partial \mathbf{d_c}}, \tag{4.37}$$

where

$$\frac{\partial \mathbf{x_c}}{\partial \mathbf{d_c}}_{3x3} = \mathbf{0_{3x3}}, \tag{4.38}$$

with $\frac{\partial (q_c \mathbf{d_c} q_c^*)}{\partial \mathbf{d_c}}_{3x3}$ obtained through (2.45).

This way, the inverse Jacobian matrix can be written as

$$\mathbf{J}^{-1}(\mathbf{c}) = \begin{bmatrix} \mathbf{I}_{3x3} & \mathbf{0}_{3x4} & \mathbf{0}_{3x3} \\ \mathbf{I}_{3x3} & \frac{\partial (q_c \mathbf{d_c} q_c^*)}{\partial q_c}_{3x4} & \frac{\partial (q_c \mathbf{d_c} q_c^*)}{\partial \mathbf{d_c}}_{3x3} \end{bmatrix}_{6x10}, \tag{4.39}$$

where

$$\frac{\partial \left(q_c \mathbf{d_c} q_c^* \right)}{\partial q_c}_{3x4} = \left[2 \left(\left[\begin{array}{c:c} q_{c0}\mathbf{d}_c + \mathbf{q}_c \times \mathbf{d}_c & -\mathbf{d}_c \mathbf{q}_c^{\top} + \left(\mathbf{d}_c \cdot \mathbf{q}_c \right) \mathbf{I_{3x3}} + \mathbf{q}_c \mathbf{d}_c^{\top} - q_{c0} \mathbf{d}_c \times \end{array} \right] \right) \right], \text{ and} \tag{4.40}$$

$$\frac{\partial \left(q_c \mathbf{d_c} q_c^* \right)}{\partial \mathbf{d_c}}_{3x3} = \left[\left(q_{c0}^2 - ||\mathbf{q}_c||^2 \right) \mathbf{I_{3x3}} + 2\mathbf{q}_c \mathbf{q}_c^{\top} + 2q_{c0}\mathbf{q}_c \times \right]. \tag{4.41}$$

As mentioned in (MAS; KITTS, 2017), despite not being a square matrix, the inverse Jacobian matrix is full rank (its rank is equal to the total DoF of the physical system), thus not having singularity problems.

4.3.1 Kinematic Controller

In (MARCIANO; BRANDÃO; SARCINELLI-FILHO, 2021), it is presented the proportional controller adapted from (MAS; KITTS, 2017). Once the inverse Jacobian matrix has been computed, the transformation from the CS to the WF of the control references is such that

$$\dot{\mathbf{x}}_r = \mathbf{J}^{-1}(\mathbf{c})\dot{\mathbf{c}}_r, \tag{4.42}$$

where $\dot{\mathbf{x}}_r$ and $\dot{\mathbf{c}}_r$ are the control references in the WF and in the CS, respectively. As for $\dot{\mathbf{c}}_r$, it is obtained through proportional controllers derived as

$$\dot{\mathbf{x}}_{cr} = \dot{\mathbf{x}}_{cd} - \mathbf{K}_x \tanh \tilde{\mathbf{x}}_c \tag{4.43}$$

and

$$\dot{d}_{cr} = \dot{d}_{cd} - K_d \tanh \tilde{d}_c, \tag{4.44}$$

where $\dot{\mathbf{x}}_{cr}$ and $\dot{d}_{cr}$ are the velocity references, $\dot{\mathbf{x}}_{cd}$ and $\dot{d}_{cd}$ are the desired velocities, all of them in the CS, $\mathbf{K_x}$ is a three-by-three diagonal matrix of positive constant gains, K_d is a positive constant and $\tilde{\mathbf{x}}_c = \mathbf{x}_c - \mathbf{x}_{cd}$ and $\tilde{d}_c = d_c - d_{cd}$ are the cluster position errors, and, finally, $\mathbf{x}_{cd}$ and d_{cd} are the desired cluster position and distance between the robots, respectively.

For the cluster orientation reference, the control law is

$$\dot{q}_{cr} = \frac{1}{2} q_c \circ \left(0,\ -sgn\left(\tilde{q}_{c0}\right)\mathbf{K}_w \tilde{\mathbf{q}}_c\right), \tag{4.45}$$

considering that $\tilde{q}_c = q_c{\circ}q_{cd} = \left(\tilde{q}_{c0},\ \tilde{\mathbf{q}}_c\right)$, where q_{cd} is the desired cluster orientation represented as a quaternion and $\mathbf{K}_w$ is a three-by-three diagonal matrix of positive constant gains. Note that the vector part of $\tilde{q}_c$ should be dealt as a three-by-one vector.

Besides,

$$\dot{\mathbf{x}}_r = \left[\dot{\mathbf{x}}_{pr},\ \dot{\mathbf{x}}_{br}\right]^{\top}_{6x1} = \left[\dot{x}_{pr},\ \dot{y}_{pr},\ \dot{z}_{pr},\ \dot{x}_{br},\ \dot{y}_{br},\ \dot{z}_{br}\right]^{\top}_{6x1}, \tag{4.46}$$

and

$$\dot{\mathbf{c}}_r = \left[\dot{\mathbf{x}}_{cr},\ \dot{q}_{cr}, \dot{\mathbf{d}}_{cr}\right]^{\top}_{10x1} = \left[\dot{x}_{cr},\ \dot{y}_{cr},\ \dot{z}_{cr},\ \dot{q}_{c0r},\ \dot{q}_{c1r},\ \dot{q}_{c2r},\ \dot{q}_{c3r},\ \dot{d}_{c1r},\ \dot{d}_{c2r},\ \dot{d}_{c3r}\right]^{\top}_{10x1},$$

where $\dot{\mathbf{d}}_{cr} = \left[\dot{d}_{cr},\ 0,\ 0\right]^{\top}_{3x1}$.

4.3.2 Dynamic Compensation

As it has already been mentioned, the kinematic controller designed considers both robots as mass-less virtual points which can reach desired velocities instantaneously. This is, obviously, untrue for a physical robot, though previous experiments demonstrate that, with this assumption, the UGV achieves an acceptable performance for the trajectory tracking task proposed in this thesis. The UAV, however, demonstrated a non acceptable performance, showing that only a kinematic controller is not enough to it. That is why a dynamic compensation module, as proposed in (SANTOS et al., 2019), was added, to consider the dynamic effects associated to the UAV in flight.

The dynamic compensation law, based on the model of (SANTOS et al., 2019), is defined as

$$\mathbf{v}_2^D = \mathbf{K}_u^{-1}(\dot{\mathbf{v}}_{r_2} + \mathbf{L}_d(\mathbf{v}_{r_2} - \mathbf{v}_2) + \mathbf{K}_v\mathbf{v}_2), \tag{4.47}$$

where $\mathbf{v}_2^D$ is the output of the dynamic compensator. Besides,

$$\mathbf{K}_u = \operatorname{diag}\left(\begin{bmatrix} K_1 & K_3 & K_5 & K_7 \end{bmatrix}\right) \tag{4.48}$$

and

$$\mathbf{K}_v = \operatorname{diag}\left(\begin{bmatrix} K_2 & K_4 & K_6 & K_8 \end{bmatrix}\right) \tag{4.49}$$

are diagonal matrices using the coefficients of Table 2, $\mathbf{L}_d$ is a diagonal positive definite gain matrix, and $\dot{\mathbf{v}}_{r_2}$ is the time derivative of $\mathbf{v}_{r_2}$, the velocity command generated by the kinematic controller.

This controller uses the feedback linearization control technique described in (KHALIL, 2002) to change the dynamics of the velocity error to an asymptotically stable linear one, thus guaranteeing the asymptotic convergence of the real UAV velocity $\mathbf{v}_2$ to the desired value $\mathbf{v}_{r_2}$.

4.3.3 UAV Orientation Controller

In (MAS; KITTS, 2017) the orientation of both UAVs are dealt with using quaternions, and are included in the Jacobian matrix. For simplicity, in this thesis it was adopted an usual Euclidean proportional controller, just for the UAV orientation, since the UGV orientation is given by its motion direction, defined by the trajectory being tracked. Regarding the *Bebop 2*, the UAV used in this thesis, its low level controller receives commands of angular velocity around the axis z^b in rad/s. Therefore, it seemed to be more practical to compute the angular velocity command already in rad/s, instead of computing it in quaternions and then convert to rad/s, thus justifying the decision of implementing a controller based on an euclidean characterization. Furthermore, the UAV orientation has nothing to do with the formation singularity, once it does not affect any characteristic of the formation, which depends only on the robots positions, not on their orientation. Nonetheless, a fully description based on quaternions requires not only the orientation controller to be expressed completely by quaternions, but also the kinematic and dynamic models of the robots. Due to time restrictions and deadlines, this work could not reach that stage, but there is the intention of describing all aspects related to orientation in this system in quaternions.

The UAV orientation can be chosen as better as it suits the application. Considering the case where it is desired that the UAV observes the trajectory taken by the UGV, it was chosen a controller which makes the UAV orientation be the same as the UGV orientation. That could not be the case, however, as it might be of interest other configurations where the UAV watches the UGV surrounds, for instance.

Let ψ_p and ψ_b be the orientations of the *Pioneer 3-DX* (UGV) and the *Bebop 2* (UAV) in euclidean angles, respectively. Considering that the UAV should be oriented to the same direction of the UGV, the current orientation of the UGV represents the desired orientation for the UAV, or, $\psi_{bd} = \psi_p$. The orientation error is then

$$\tilde{\psi}_b = \psi_{bd} - \psi_b, \tag{4.50}$$

such that a proportional controller for the UAV orientation can be designed as

$$\dot{\psi}_{br} = K_\psi \tanh \tilde{\psi}_b. \tag{4.51}$$

Because ψ_{bd} is defined as the UGV current orientation, in order to compute $\dot{\psi}_{bd}$ it would be necessary to take the numerical differentiation. However, this may introduce a considerable amount of noise in the system, so that to avoid this just a proportional control law is adopted. Besides the fact that the UAV orientation is not relevant to the main goal of this work, the orientation error can be considerably decreased with a well tuned gain K_ψ.

4.4 Control System Overview

Summarizing the whole control system explained in this chapter, there are two main regions in the control structure: the Cluster Space and the World Frame. The actions involved in the control loop are

- the desired positions, velocities and orientations are given in the CS;
- the three-dimensional positions of the robots are measured in the WF
- the current values of the formation variables are obtained using the Forward Transformation and the robot variables;
- the control references are computed in the CS using the cluster current state and the desired references;
- the inverse Jacobian converts the control references from the CS to the WF;
- the matrix of inverse kinematics of each robot converts the control references from the WF to the RS;

- a dynamic compensation is applied to the UAV kinematic control reference;
- each robot receives its control references, which causes them to move accordingly to the desired trajectory.
- new positions are measured and the process is repeated, till finishing the navigation.

Figure 18 shows a flowchart representing the sequence of actions just described.

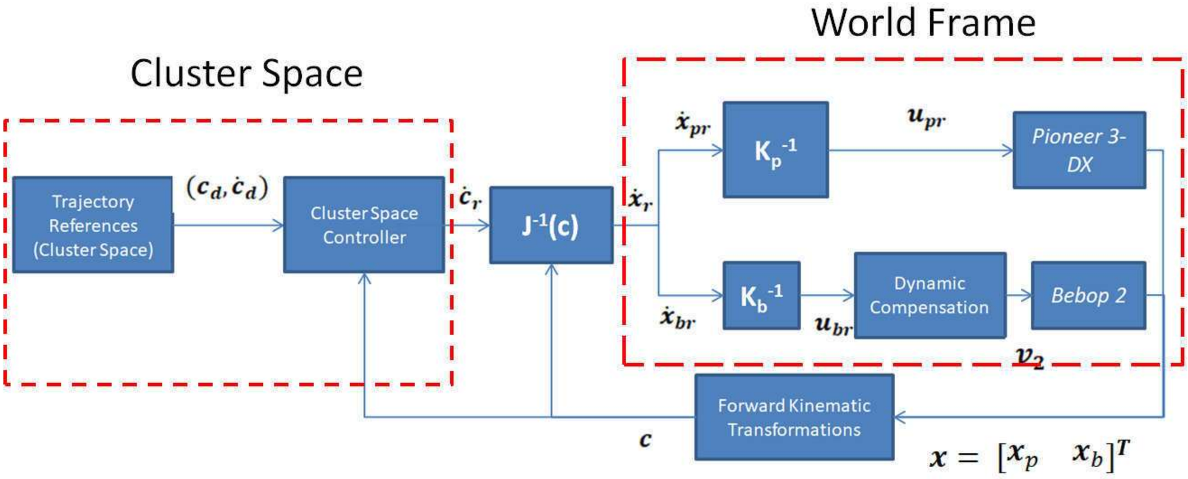


Figura 18 – Flowchart of the whole formation control system

5 Experimental Results and Discussions

Three main experiments were run to validate the formation description and controller proposed in this thesis. Two of them consisted on trajectory tracking tasks considering a configuration for the UGV-UAV formation in which the UAV hovers over the UGV, for a circular trajectory and for a lemniscate trajectory. The last trajectory was selected to better excite the system dynamics. These two experiments aimed to demonstrate that the quaternion-based description proposed in this work is not affected by singularities for configurations where the UAV hovers over the UGV, as it happens in (RABELO; BRANDãO; SARCINELLI-FILHO, 2021).

The third experiment considers the same trajectory tracking task, though considering a configuration where the UAV initially hovers in front of the UGV, at the same level of the landing platform, and, then, after 60s, alternate to the same configuration of the two previous experiments. In this case a circular trajectory was tracked, due to the fact that the lemniscate curve demanded a greater radius, and, because the UAV would navigate in front of the UGV in part of the experiment, hence the total area necessary for the experiment was increased. Therefore, for safety issues, a circular trajectory was chosen. This experiment aimed to demonstrate that the quaternion-based description proposed in this work has no singularities for any attainable configuration of the CS.

For all the experiments the values adopted for the constants so far mentioned are

$$\mathbf{L}_d = \mathrm{diag}\left(\begin{bmatrix}2 & 2 & 1.8 & 5\end{bmatrix}\right),$$

$$K_\psi = 3,$$

$$a = 0.15\ m,$$

$$\mathbf{K}_\omega = \mathbf{I}_{3x3},$$

$$\mathbf{K}_x = \mathbf{I}_{3x3},$$

and

$$K_d = 1$$

.

5.1 Tracking a Circular Trajectory with the UAV Hovering over the UGV

In this experiment the UAV stays hovering over the UGV all the time, after taking-off. The UAV begins in the ground and moves accordingly, to assemble the desired configuration. The circular trajectory to be tracked is described as

$$\begin{aligned} x_{cd} &= r_x \operatorname{sen}(\omega_d t) + x_0, \\ y_{cd} &= r_y \cos(\omega_d t) + y_0, \\ z_{cd} &= z_0, \end{aligned} \tag{5.1}$$

for which

$$\begin{aligned} \dot{x}_{cd} &= \omega_d r_x \cos(\omega_d t), \\ \dot{y}_{cd} &= -\omega_d r_y \operatorname{sen}(\omega_d t), \\ \dot{z}_{cd} &= 0, \end{aligned} \tag{5.2}$$

where $r_x = 1\ m$, $r_y = 1\ m$, $\omega_d = \frac{\pi}{15}\ rad/s$, and $x_0 = y_0 = z_0 = 0\ m$. As for the desired distance between the robots, it was adopted $d_{cd} = 1.5\ m$.

Through (4.26) one can demonstrate that the desired orientation for the cluster, in this case, is the unit quaternion given by

$$\hat{q}_{cd} = \frac{\sqrt{2}}{2} + 0i - \frac{\sqrt{2}}{2}j + 0k. \tag{5.3}$$

Figures 19 shows a three-dimensional representation of the trajectory tracked by the cluster, whereas Figures 20 and 21, by their turn, summarize the results of such an experiment.

Figure 20 shows, in the left side, the cluster variables (its position $\mathbf{x}_c$ and the distance d_c between the two robots). As for its right side, it shows the errors in such variables with respect to the desired values. Such results allow observing that the cluster was capable to accomplish the trajectory tracking task with an absolute error below 0.1 m. Moreover, it remained over the trajectory, after reaching it, configuring a steady-state situation. It is also important to remark that the point of interest for control of the cluster coincides with the Pioneer position $\mathbf{x}_p$. Hence, these graphics also represent the behavior of the Pioneer along the time.

Now regarding Figure 21, its left side shows the cluster orientation in quaternions, whereas its right side shows its error with respect to the desired orientation. It is important to remember that the unit quaternion $\hat{q}_c = 1 + 0i + 0j + 0k$ represents the identity element, meaning that it represents the action of "doing nothing", in terms of control. Therefore, the expectancy is that the error converge to it when the controller guides correctly the cluster

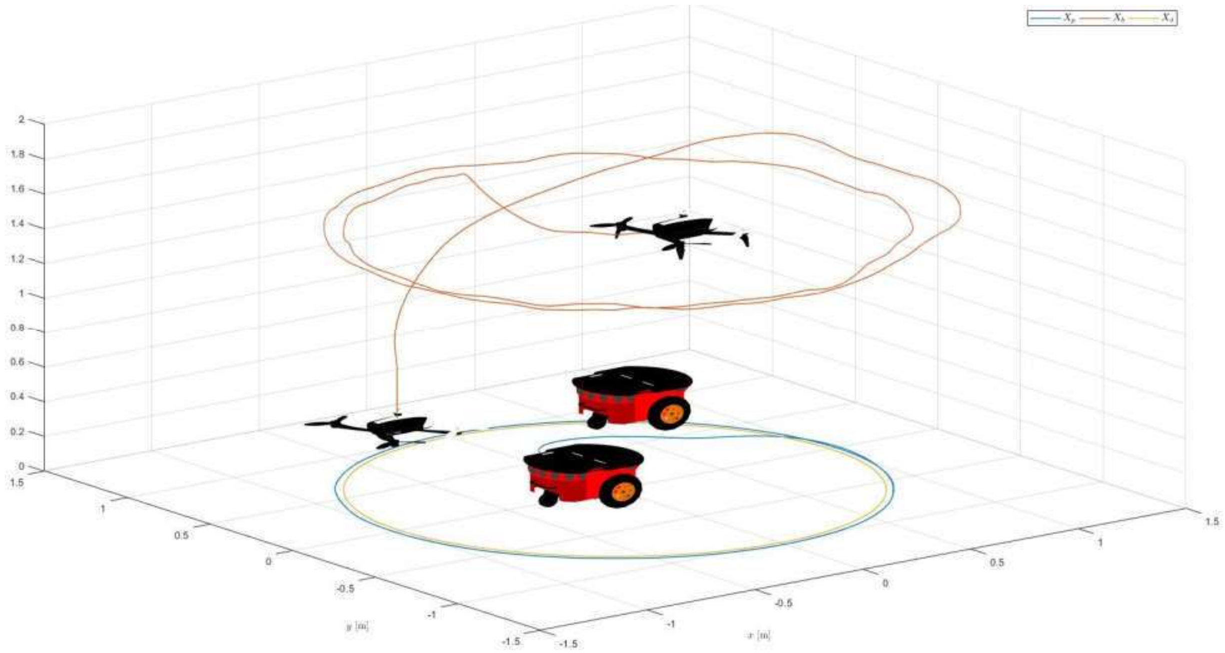


Figura 19 – Three Dimensional Circular Trajectory of Each Robot.

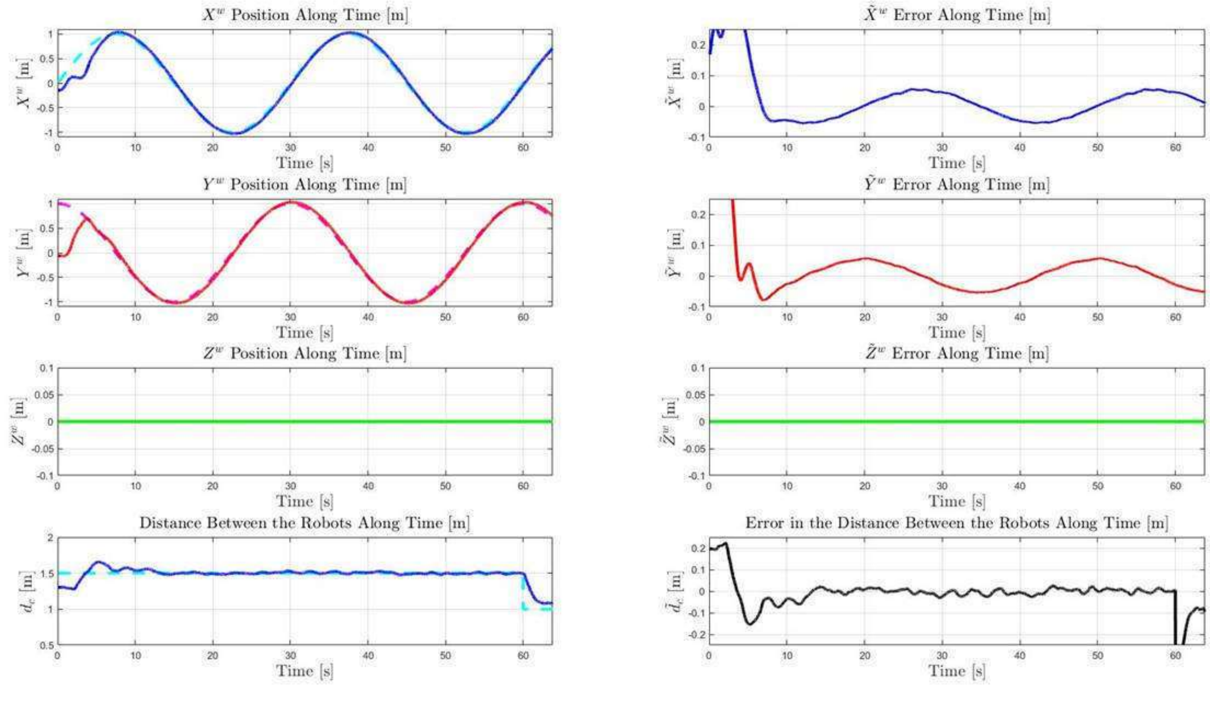


Figura 20 – Cluster Variables Along Time for a Circular Trajectory: Position and Distance Between Robots.

to track the desired orientation. Therefore, the conclusion is that the cluster orientation $\hat{q}_c$ also reached its desired value, with some oscillation in steady state, but within an acceptable margin.

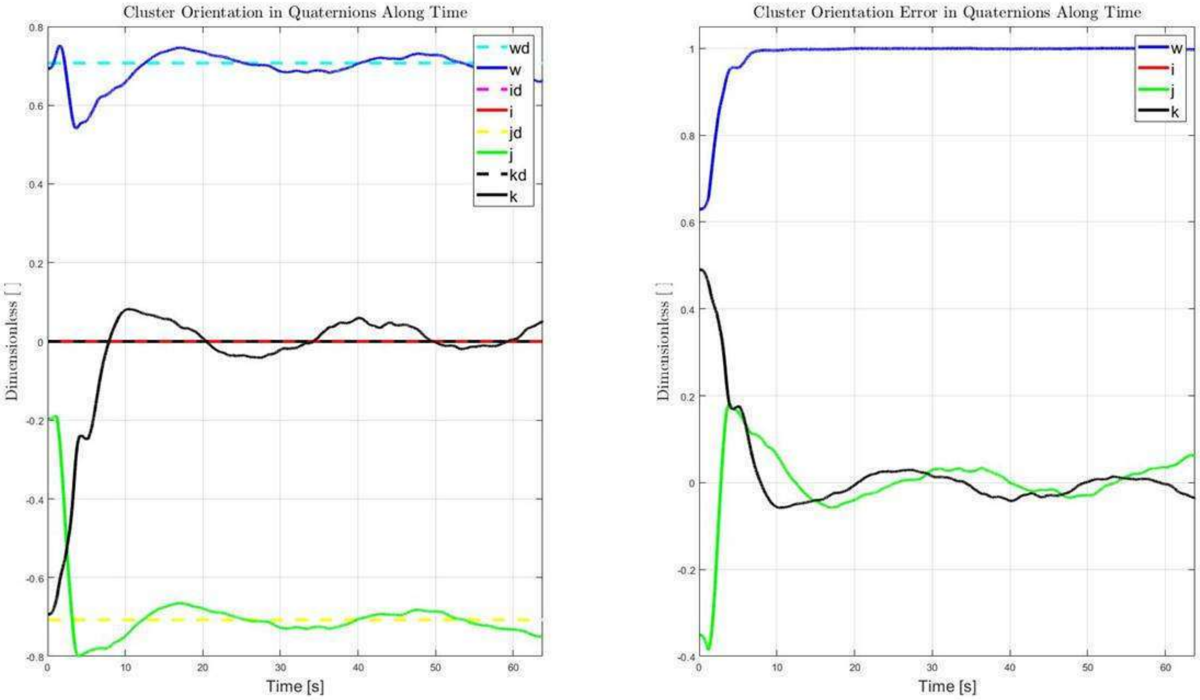


Figura 21 – Cluster Variables Along Time for a Circular Trajectory: Orientation in Quaternions.

The fact that the cluster reached its designated trajectory implies that individually the robots also did that. Figures 22 and 23 show the temporal variation of path traveled by the Pioneer and its orientation, respectively.

As mentioned before, the UGV trajectory coincides with the cluster one. Furthermore, due to the fact that the UGV considered depends on its orientation to perform the trajectory, a low error in position implies in a low error for the orientation, as it can be observed in Figure 23, where the error, aside the transient, remains close to the identity.

Now, for the Bebop, Figures 24 and 25 show the temporal variation of the path traveled by it and its orientation, respectively.

The UAV presented a greater error in comparison to the UGV, what was expected, once that aerial vehicles tend to be more unstable than ground vehicles, mainly when flying close to the ground, when the ground effect is meaningful. However, the maximum absolute error in steady state did not overcame 0.2 m, which is acceptable for this application. The UAV orientation was also capable to track the desired reference, which in this case was the UGV orientation. There was a persistent delay in steady state, which can be observed in Figure 25. It was expected due to the controller used (a proportional one), which is not capable to eliminate completely the error, but can decrease it considerably with a well tuned gain. As mentioned before, the UAV orientation does not affect the cluster proposed in this work,

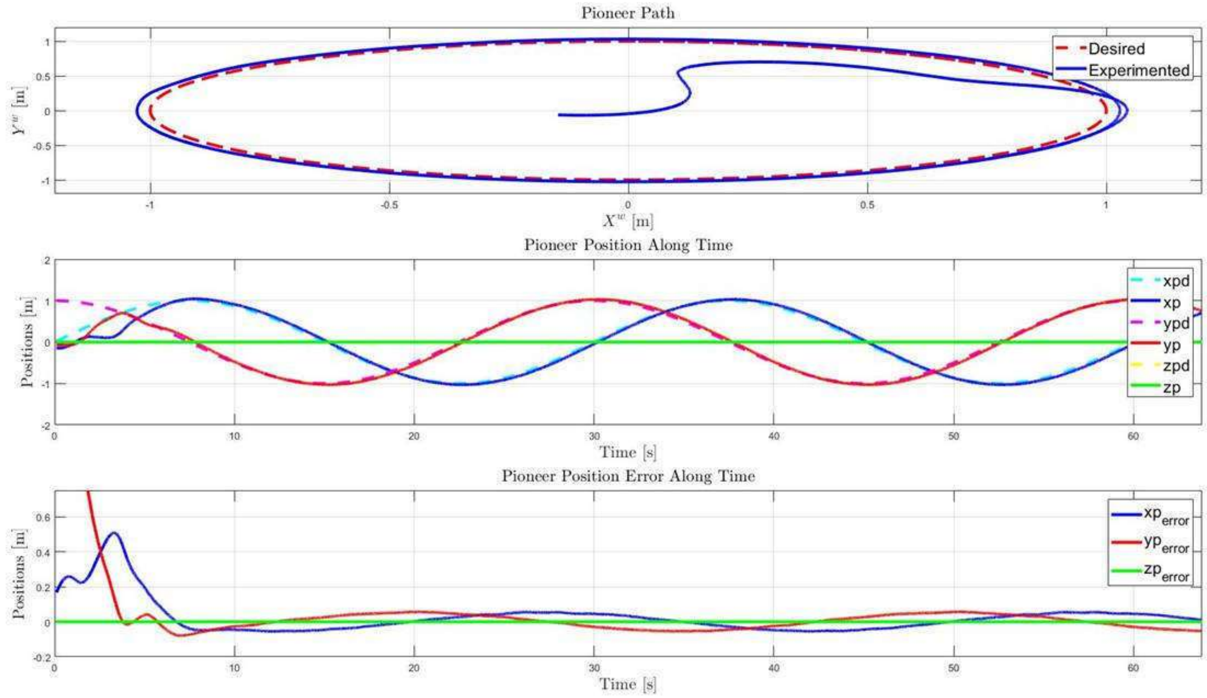


Figura 22 – Pioneer Trajectory Along Time for a Circular Trajectory.

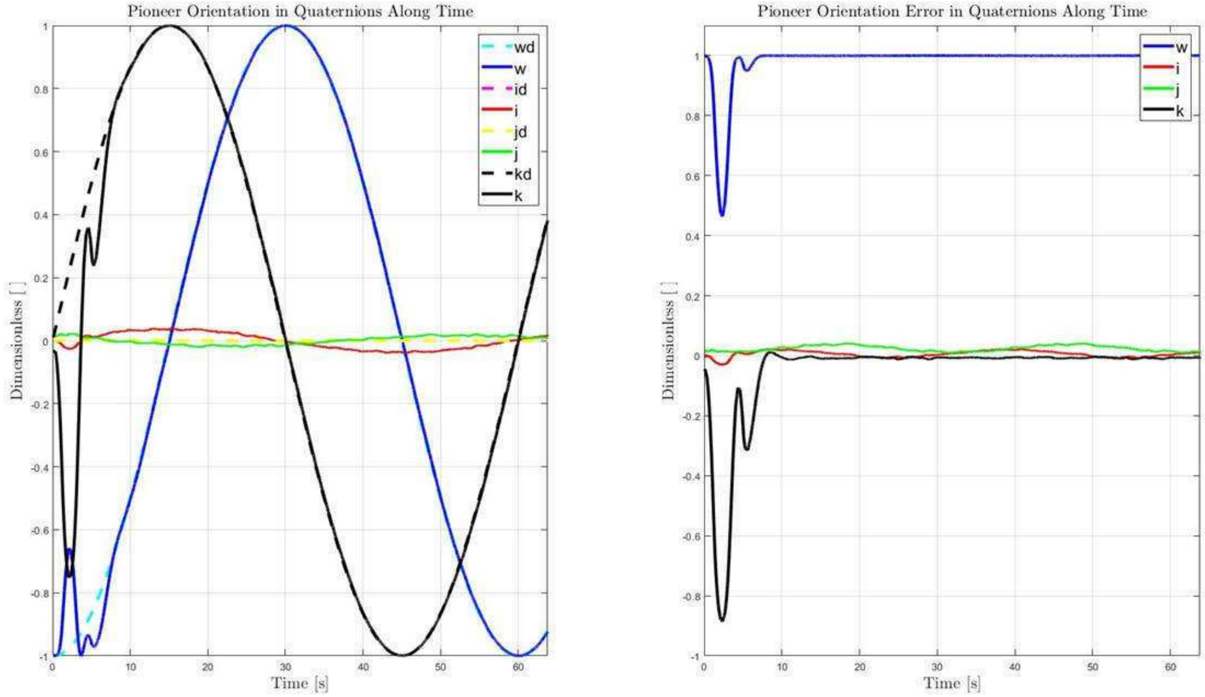


Figura 23 – Pioneer Orientation in Quaternions Along Time for a Circular Trajectory.

although it is intended to include the UAV orientation in the cluster, in order to unify the whole system representation. As an auxiliary resource to understand the experiment, a video

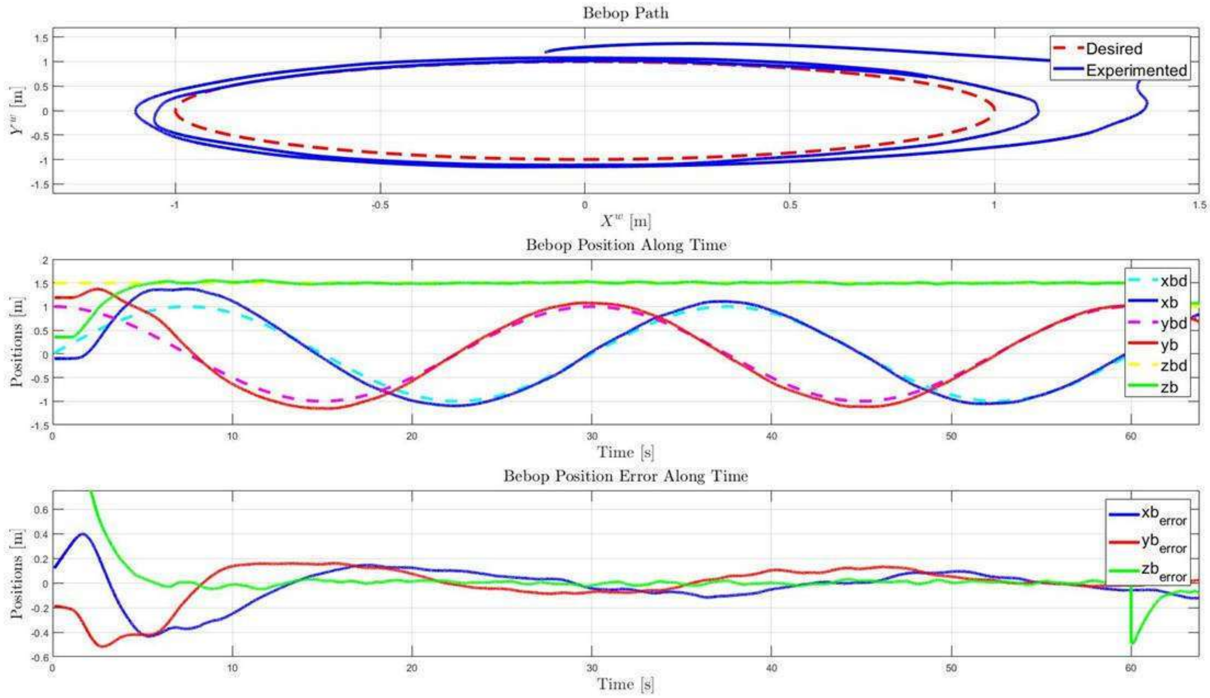


Figura 24 – Bebop Trajectory Along Time for a Circular Trajectory.

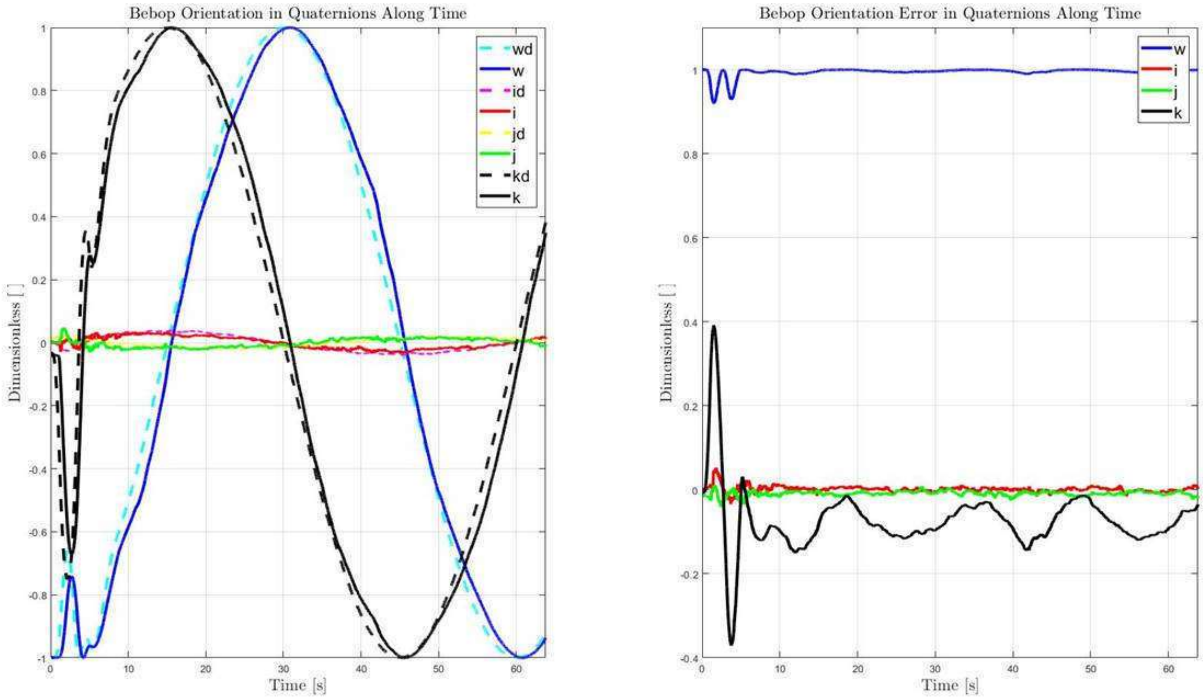


Figura 25 – Bebop Orientation in Quaternions Along Time for a Circular Trajectory.

is available in the link <https://www.youtube.com/watch?v=jJy_OTnBXuE&t=0s>.

5.2 Tracking of a Lemniscate of Bernouli Trajectory with the UAV Hovering over the UGV

In this experiment the UAV still stays hovering over the UGV all the time, after taking-off. Nevertheless, a lemniscate curve is adopted as the trajectory to be tracked, in order to provide a greater excitation of the system dynamics. The UAV begins in the ground and moves accordingly, to assemble the desired configuration. The lemniscate trajectory is described as

$$\begin{aligned} x_{cd} &= r_x \operatorname{sen}(\omega_d t) + x_0, \\ y_{cd} &= r_y \operatorname{sen}(2\omega_d t) + y_0 \\ z_{cd} &= z_0, \end{aligned} \tag{5.4}$$

for which,

$$\begin{aligned} \dot{x}_{cd} &= \omega_d r_x \cos{(\omega_d t)}, \\ \dot{y}_{cd} &= 2\omega_d r_y \cos{(2\omega_d t)}, \\ \dot{z}_{cd} &= 0, \end{aligned} \tag{5.5}$$

where $r_x = 1\ m$, $r_y = 1.5\ m$, $\omega_d = \frac{\pi}{30}\ rad/s$, and $x_0 = y_0 = z_0 = 0\ m$. Besides, the desired distance between both robots is $d_{cd} = 1.5\ m$.

In this case, the desired cluster orientation $\hat{q}_{cd}$ is the same as the one given in Section 5.1 (see (5.3)).

Figures 26 shows a three-dimensional representation of the trajectory tracked by the cluster. As for Figures 27 and 28, they summarize the results of such an experiment.

Figure 27 shows, in the left side, the cluster variables, specifically its position $\mathbf{x}_c$ and the distance d_c between both robots. As for its right side, it shows the errors of such variables in comparison to the desired values. Observing these results, it can also be concluded that the cluster was capable to accomplish the trajectory tracking task with an absolute error lower than 0.1 m in X^w and lower than 0.15 m in Y^w, which is slightly worse than the previous experiment. This is not a surprise, once that the curve considered here is more demanding, in terms of control actions. Furthermore, the cluster remains stable, once it enters its steady state, with a persistent error oscillation, as it can be observed in Figure 27.

With regard to Figure 28, the left side shows the cluster orientation in quaternions, whereas the right side shows its error in comparison to the desired orientation. It can be observed that the cluster orientation was capable to reach the desired reference, although with some persistent oscillation in steady state, which again is due to the fact that the Lemniscate curve

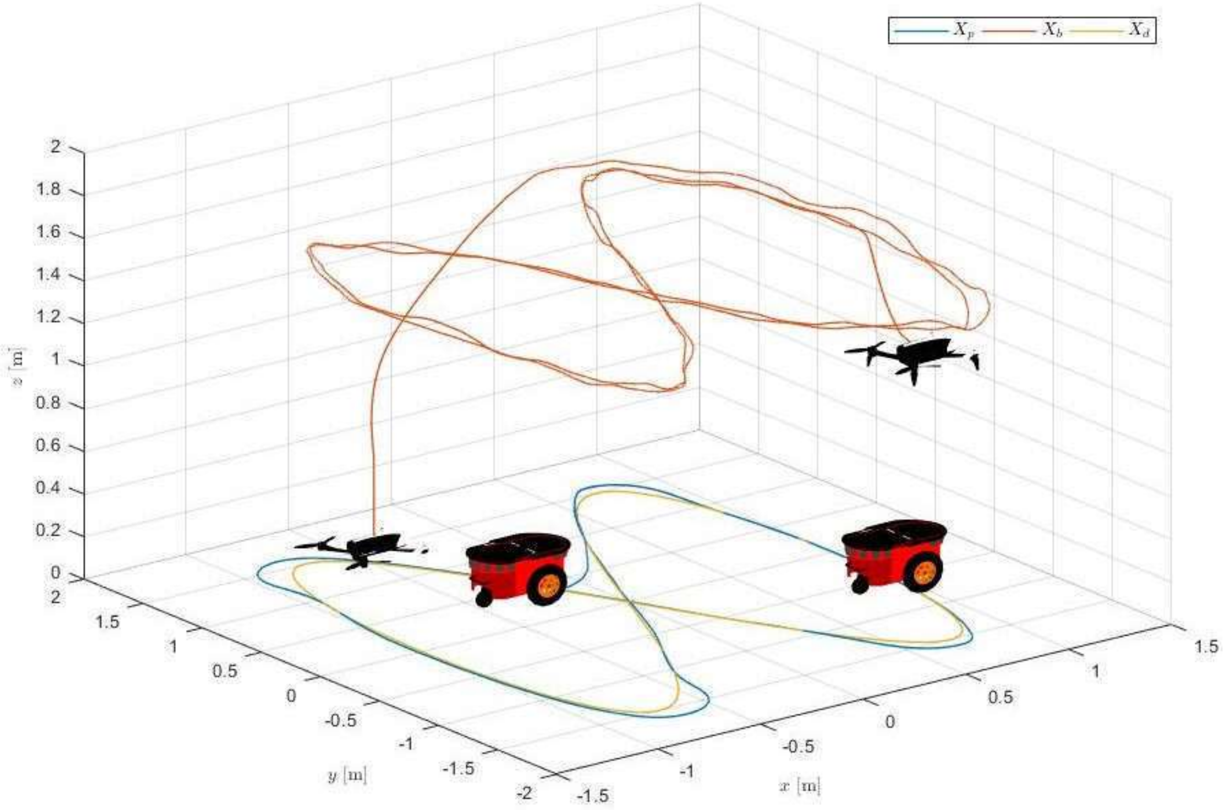


Figura 26 – Three Dimensional Lemniscate Trajectory of Each Robot.

is more challenging than a circumference, in terms of control actions. Nonetheless, such an error is within the bounds of tolerance for the applications here considered.

Once more, as the cluster reaches its designated trajectory, both robots also do that. Figures 29 and 30 show the temporal variation of the path traveled by the Pioneer and its orientation, respectively. Observing both figures, it can be seen that the trajectory has been successfully tracked by the Pioneer, which implies in the orientation error converging to the identity as can be seen in Figures 29.

Now, for the Bebop, Figures 31 and 32 show the temporal variation of the path traveled by it and its orientation, respectively.

The UAV presented a greater error in comparison to the UGV, what again was expected, for the same reasons given in Section 5.1. However, the maximum absolute error in steady state did not overcame 0.2 m, which is acceptable for this application. The UAV orientation was also capable to track the desired reference, which in this case was the UGV orientation, still with more difficult than the registered for the circular trajectory. There was also a persistent delay in steady state, which can be observed in Figure 32, again for the reason explained in Section 5.1. As an auxiliary resource to understand the experiment, a video is available in the link <https://www.youtube.com/watch?v=dc4mIU3fRGg>.

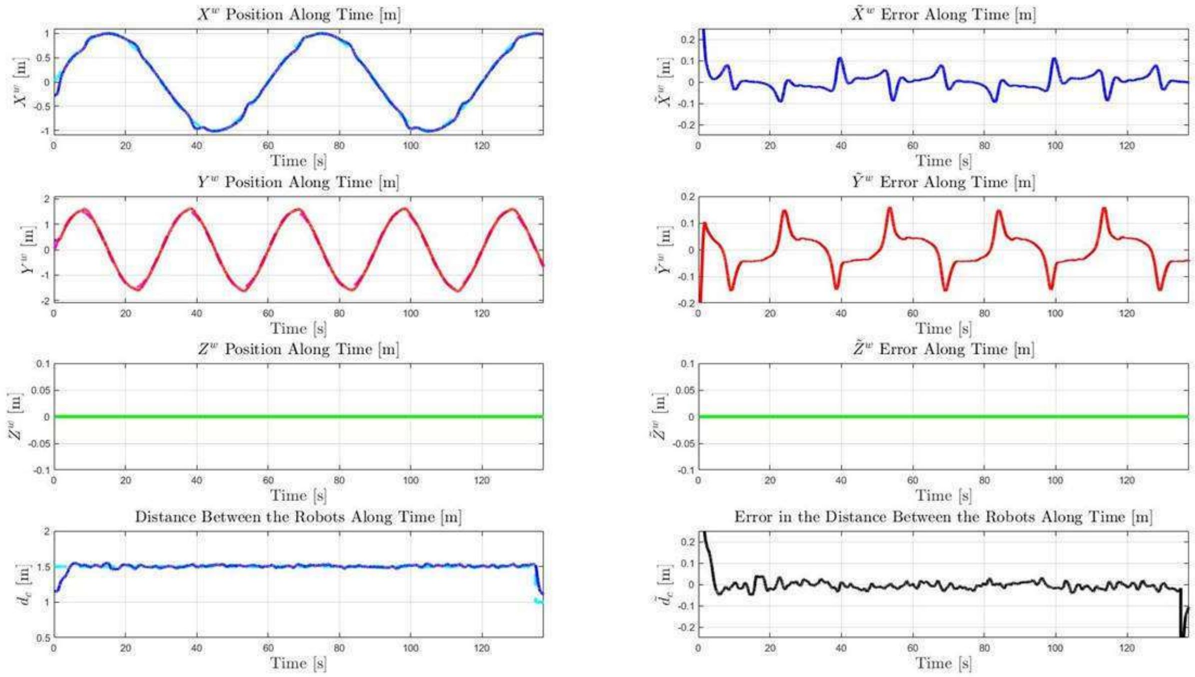


Figura 27 – Cluster Variables Along Time for a Lemniscate Trajectory: Position and Distance Between Robots.

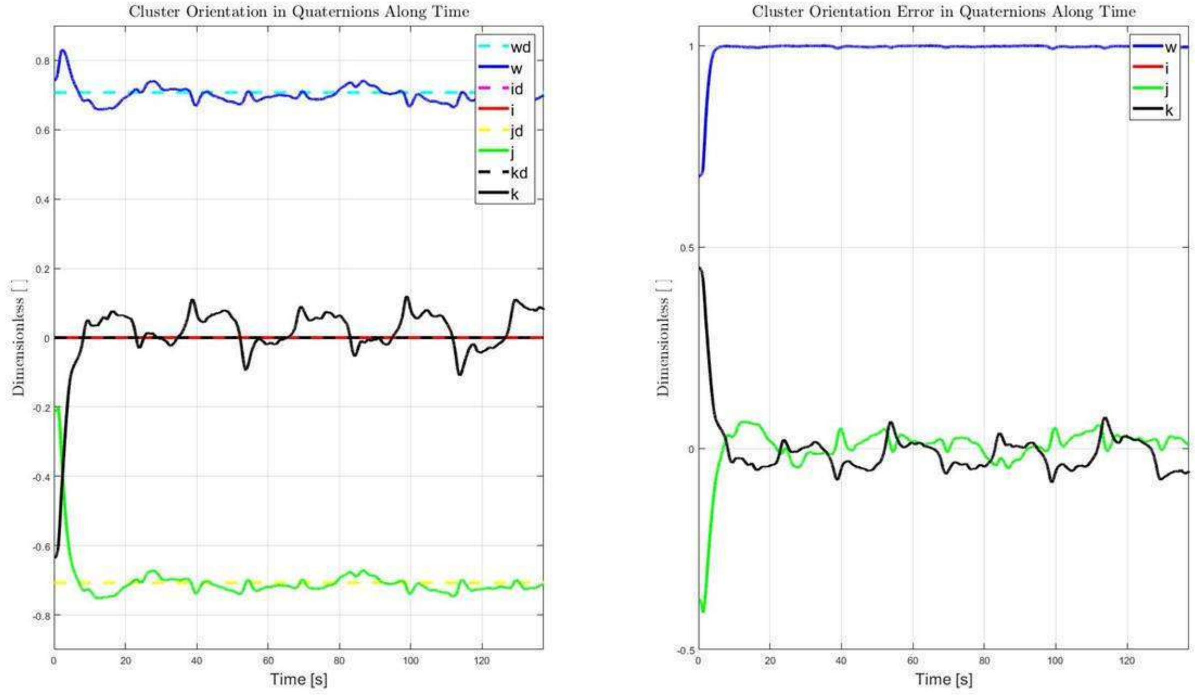


Figura 28 – Cluster Variables Along Time for a Lemniscate Trajectory: Orientation in Quaternions.

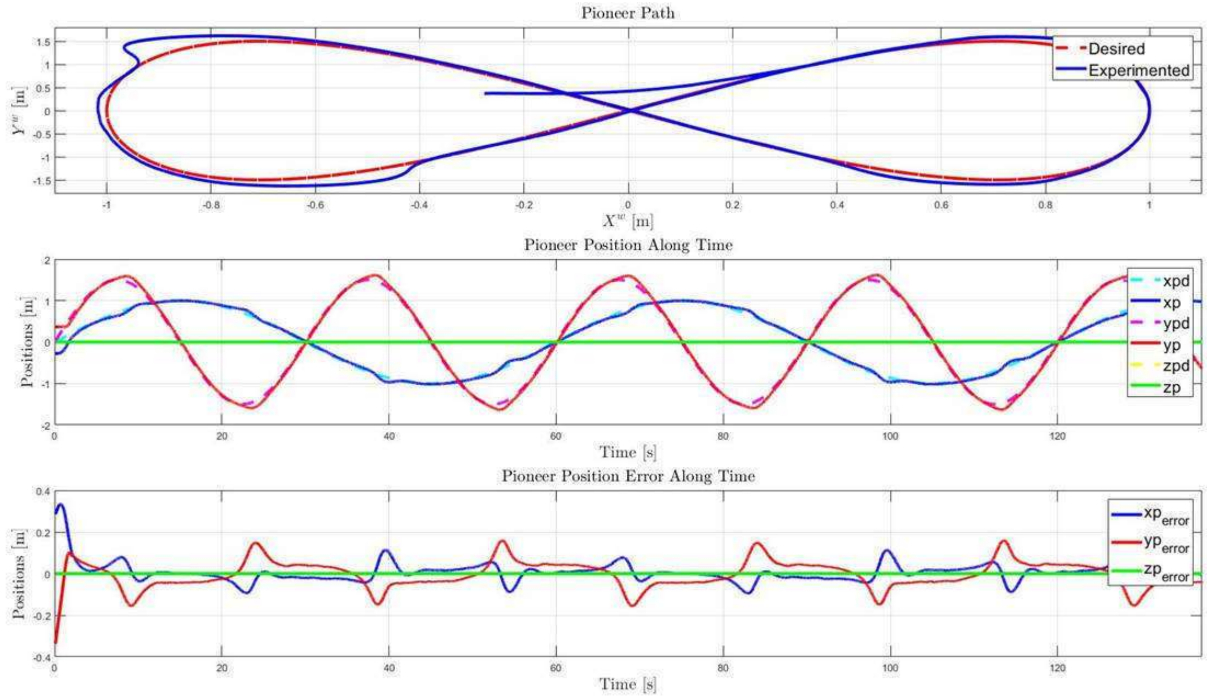


Figura 29 – Pioneer Trajectory Along Time for a Lemniscate Trajectory.

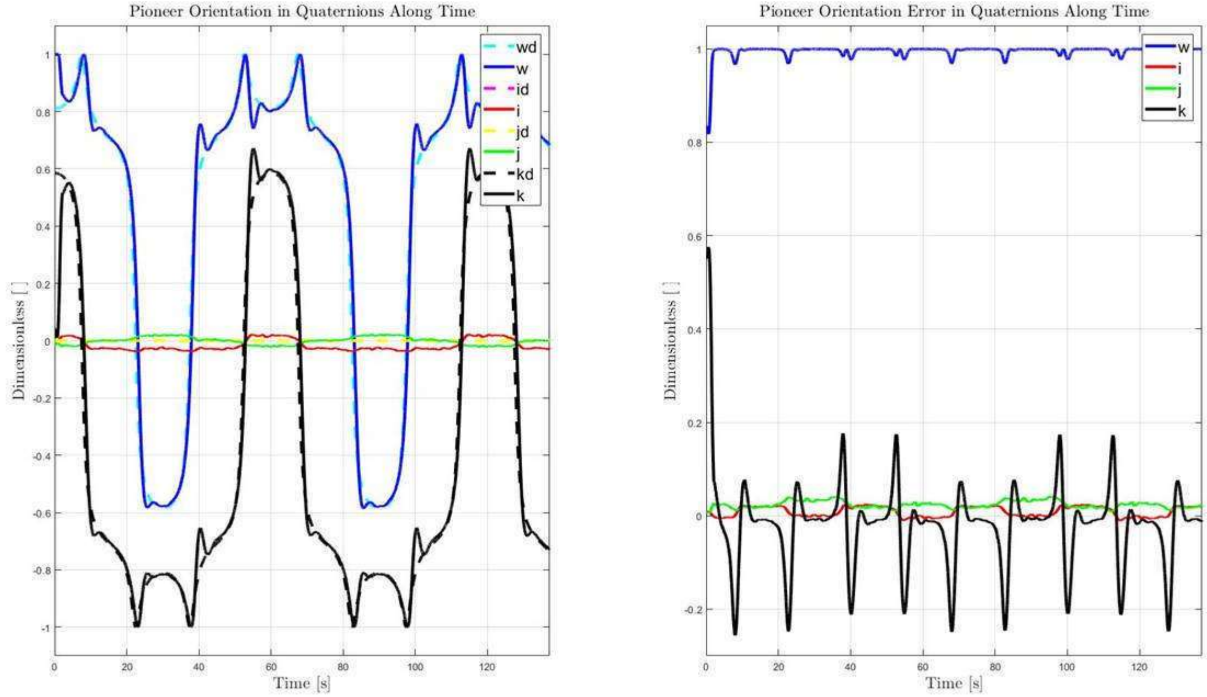


Figura 30 – Pioneer Orientation in Quaternions Along Time for a Lemniscate Trajectory.

Despite some oscillations around the desired positions and orientations along time, the quaternion based controller proposed in this thesis demonstrated itself as reliable for the

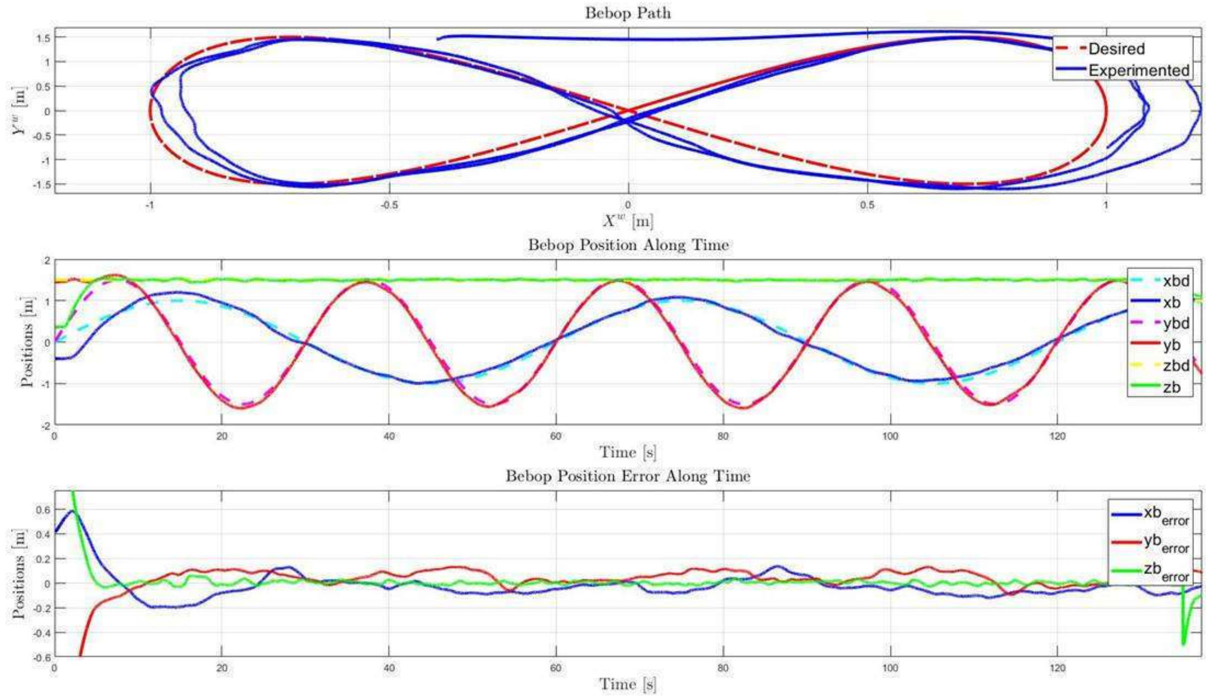


Figura 31 – Bebop Trajectory Along Time for a Lemniscate Trajectory.

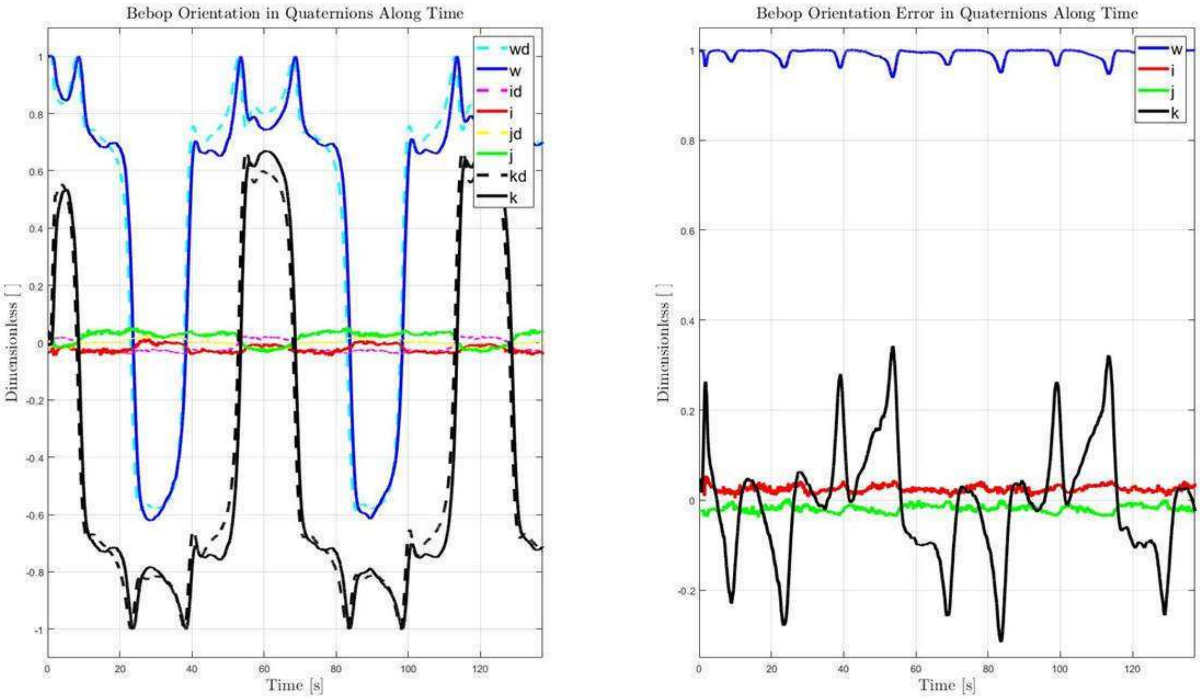


Figura 32 – Bebop Orientation in Quaternions Along Time for a Circular Trajectory.

trajectory tracking task proposed with a configuration which would represent a singularity for the Euclidean description given in (RABELO; BRANDãO; SARCINELLI-FILHO, 2021).

Besides, no variable became undefined for any moment of the experiment, demonstrating that formation singularities do not arise for this controller with such a configuration.

5.3 Tracking a Circular Trajectory with the UAV Initially in Front of the UGV and After Hovering over It

Finally, in the third experiment, the UAV stays in front of the UGV for the first 60s after taking-off. Then, the UAV hovers over the UGV, as it happened in the two previous experiments, until the experiment time is over. The circular trajectory to be tracked now is described as

$$\begin{aligned} x_{cd} &= r_x \operatorname{sen}(\omega_d t) + x_0, \\ y_{cd} &= r_y \cos(\omega_d t) + y_0, \\ z_{cd} &= z_0, \end{aligned} \tag{5.6}$$

for which

$$\begin{aligned} \dot{x}_{cd} &= \omega_d r_x \cos(\omega_d t), \\ \dot{y}_{cd} &= -\omega_d r_y \operatorname{sen}(\omega_d t), \\ \dot{z}_{cd} &= 0, \end{aligned} \tag{5.7}$$

where $r_x = 0.5\ m$, $r_y = 0.5\ m$, $\omega_d = \frac{\pi}{30}\ rad/s$, and $x_0 = y_0 = z_0 = 0\ m$. As for the desired distance between the robots, it was adopted $d_{cd} = 0.75\ m$.

Notice that the radius of this circle, as well as the distance between the two robots, is reduced in comparison to the first experiment. As mentioned in Section 3.4, the useful width of the laboratory where the experiments were run was 4 meters. The origin of the WF is in the middle of the room, so that just 2 meters were available for the formation maneuvering in each half of the smallest side of the laboratory. Taking this in account, in the first 60s, the UAV would hover at the same level of the landing platform and at a distance d_c from the UGV. Adding the radius r of the circumference, the bound

$$\mathbf{r} + d_c < \frac{W}{2} \tag{5.8}$$

should be respected, where W is the room width.

Hence, as a safety measure, $\mathbf{r}$ and d_{cd} were reduced to the values above. Furthermore, the maximum velocity of the UAV was reduced to 40% of its total, which had a considerable effect in the UAV capability to respond to the controller signals, as it is going to be shown ahead in this section. Nevertheless, it does not invalidate the fact that the quaternion based controller was able to make the formation navigate in a configuration which would not be possible considering the Euclidean description used in (ERNANDES-NETO; SARCINELLI-FILHO; BRANDÃO, 2019).

Another important aspect of the formation for this experiment is that the Cluster Orientation is no longer constant all the time, as it had been the case for the first two experiments. For the first 60s, $\hat{q}_{cd}$ can be computed by using (4.10), (4.11) and (4.12), considering $\beta_c = 0°$, $\alpha_c = atan2\left(\dot{y}_{cd},\ \dot{x}_{cd}\right)$, $\rho_c = d_{cd}$, $x_c = x_{cd}$, $y_c = y_{cd}$ and $z_c = z_{cd}$ and, finally, applying (4.26). After this period, the cluster orientation becomes again the constant given by (5.3).

Figures 33 shows a three-dimensional representation of the path traveled by the cluster, whereas Figures 34 and 35, by their turn, summarize the results of such an experiment.

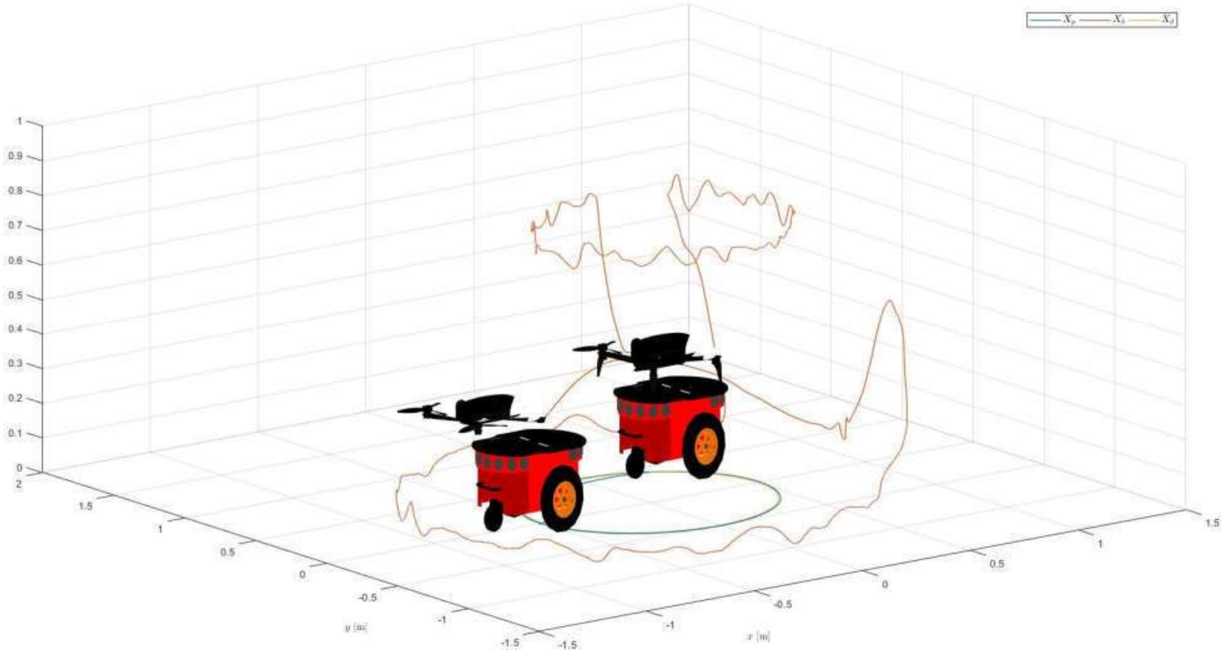


Figura 33 – Three Dimensional Circular Trajectory for Each Robot with Transition between Desired Configurations.

Analyzing Figure 34, it can be observed that the controller was capable to accomplish the trajectory tracking for the point of interest for control, as well as to keep the distance between the robots. Regarding Figure 35, however, it can be observed a greater difficult to track the desired cluster orientation, due to the facts previously mentioned. Nonetheless, the error is acceptable for this application. Besides, the orientation clearly follows its desired value, though with some delay.

Once the cluster trajectory has been successfully tracked, the same is expected for the Pioneer, what effectively happens, as it can be seen in Figures 36 and 37. The position error had an absolute value lower than 0.05m along the whole experiment. The reduction on the trajectory speed in comparison to the first experiment (from $\omega_d = \frac{\pi}{15}\ rad/s$ to $\omega_d = \frac{\pi}{30}\ rad/s$)

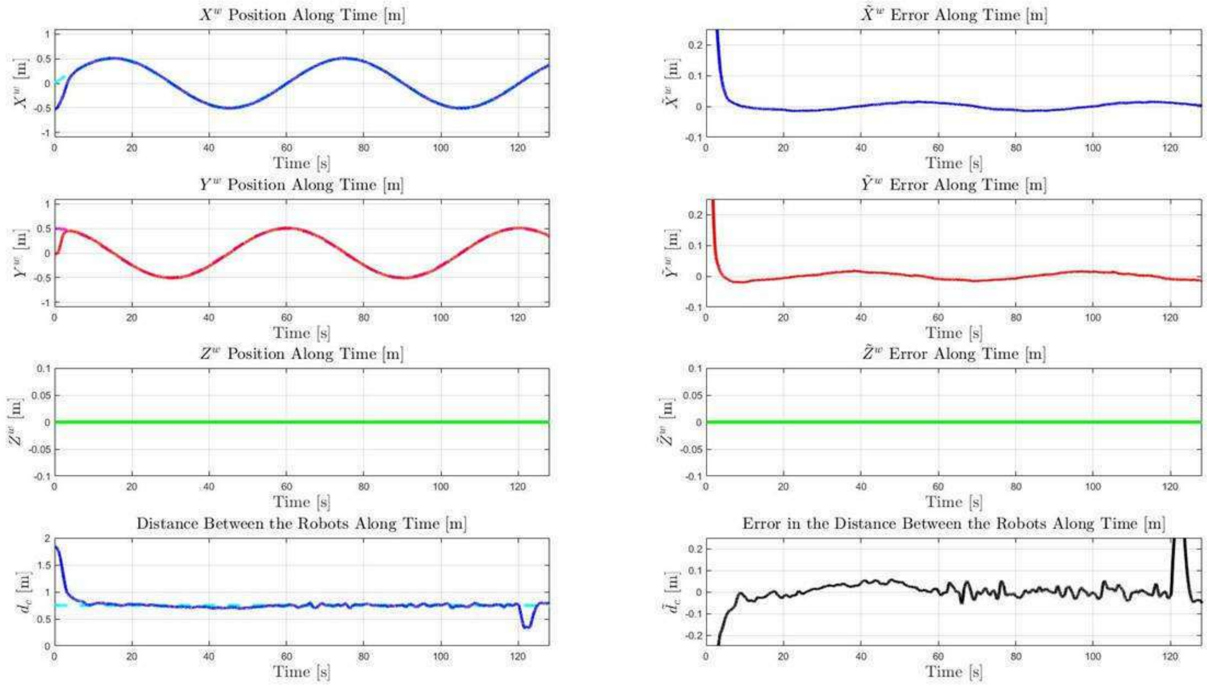


Figura 34 – Cluster Variables Along Time for a Circular Trajectory Varying the Cluster Orientation: Position and Distance Between Robots.

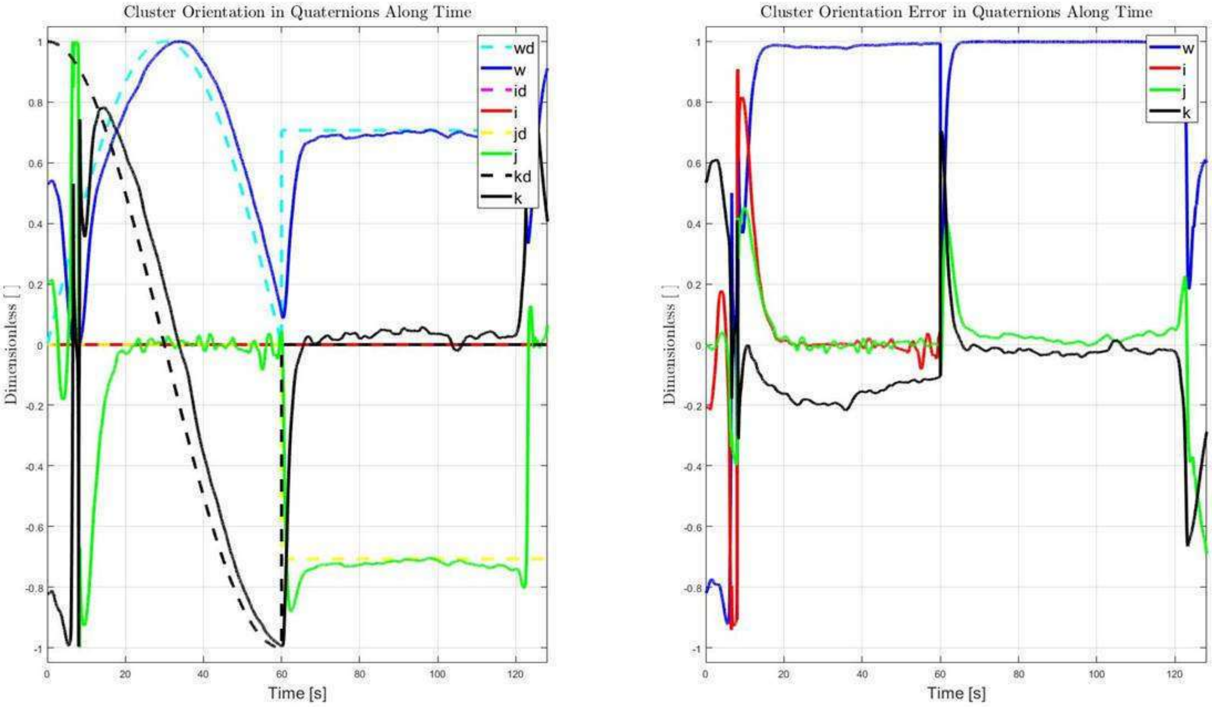


Figura 35 – Cluster Variables Along Time for a Circular Trajectory Varying the Cluster Orientation: Orientation in Quaternions.

for safety issues is the obvious reason for this improvement. Consequently, the Pioneer orientation also had a reduced error, with a slight delay, as it can be observed in Figure 37.

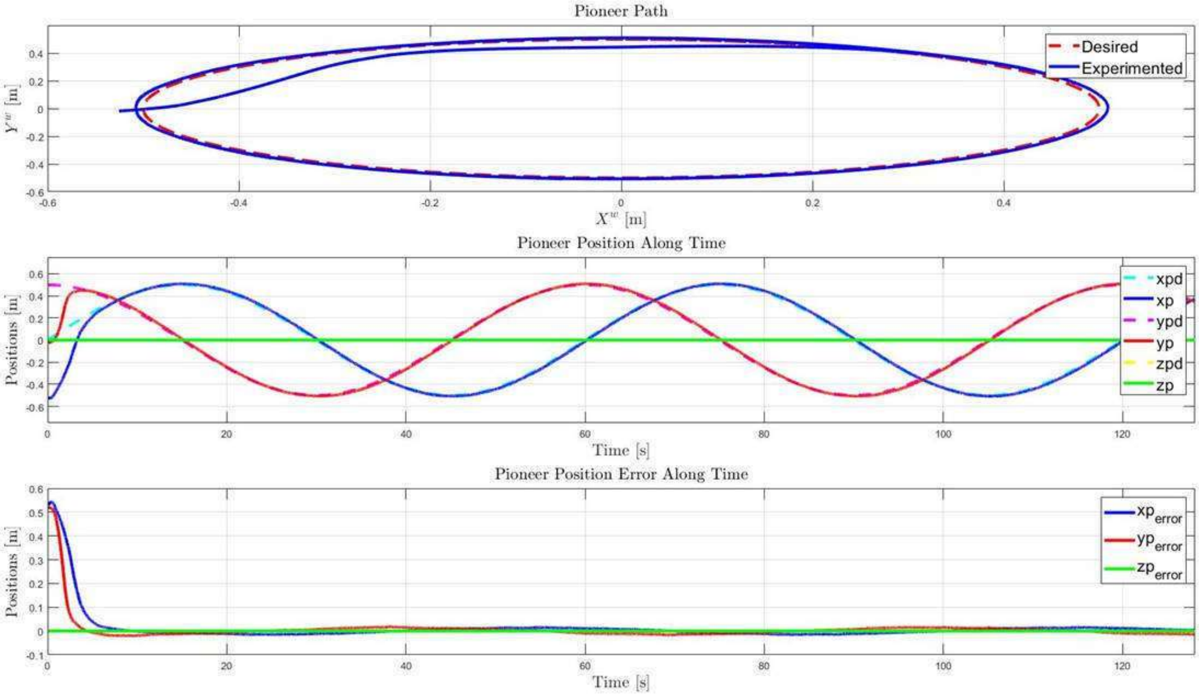


Figura 36 – Pioneer Trajectory Along Time for a Circular Trajectory Varying the Cluster Orientation.

For the Bebop, on the other hand, the performance has not been as efficient as it was for the Pioneer in the first section of the experiment, as it can be seen in Figures 38 and 39. This is due to the reduction of the Bebop maximum velocity, which limits its capability to decrease the error, what is more visible when the UAV has to navigate in front of the UGV (a greater velocity was necessary). The maximum absolute position error along time for the UAV was around 0.4m which is not negligible, although it can be improved by allowing a greater velocity to the UAV. Furthermore, it does not invalidate the fact that the navigation was successfully accomplished considering a formation configuration where the UAV hovers at the same level of the landing platform.

The second section of this experiment had a lower position error along time (lower than 0.2m) in comparison to the first section of the experiment. This is explained by the fact that this formation configuration is less demanding for the controller, such that even with its maximum velocity reduced, the Bebop was able to remain in the desired formation configuration while the designated trajectory was tracked.

It is also important to remark that, in this experiment, there was two transients: one at the beginning of the experiment and the second after 60s, as it can be seen in Figure 38,

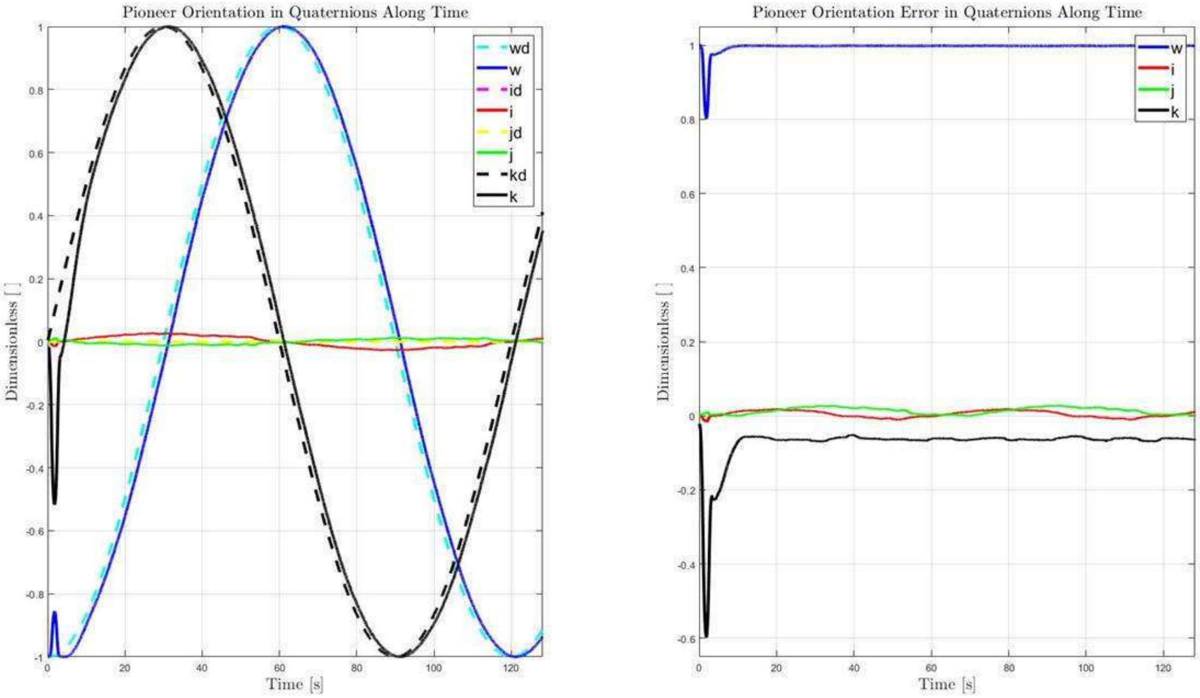


Figura 37 – Pioneer Orientation in Quaternions Along Time for a Circular Trajectory Varying the Cluster Orientation.

due to the transition between the desired formation configurations. In both regions of the experiment, no variable became undefined in any moment, meaning that no singularities had arisen for the quaternion based formation here proposed.

As a conclusion, after analyzing the results obtained from these three experiments, it can be observed that the quaternion based formation description, as well as the quaternion based controller, both proposed in this thesis, were able to accomplish the trajectory tracking task for a heterogeneous formation involving a UAV and a UGV. Moreover, it did not presented formation singularities for any configuration which would have had this sort of issue if an Euclidean description had been chosen. Therefore, it is possible to claim that a quaternion based description for mobile robots formation is capable to accomplish the same tasks of Euclidean based description without problems related to singularities, thus been more robust in contexts where formation singularities are a concern.

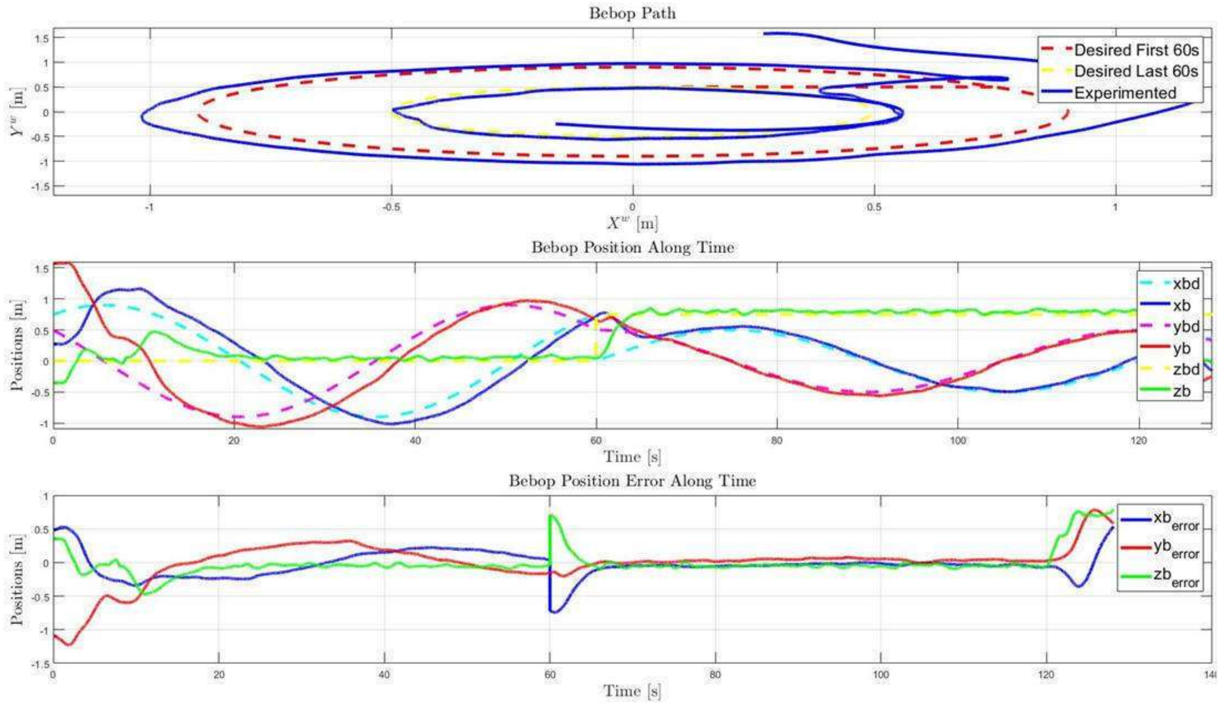


Figura 38 – Bebop Trajectory Along Time for a Circular Trajectory Varying the Cluster Orientation

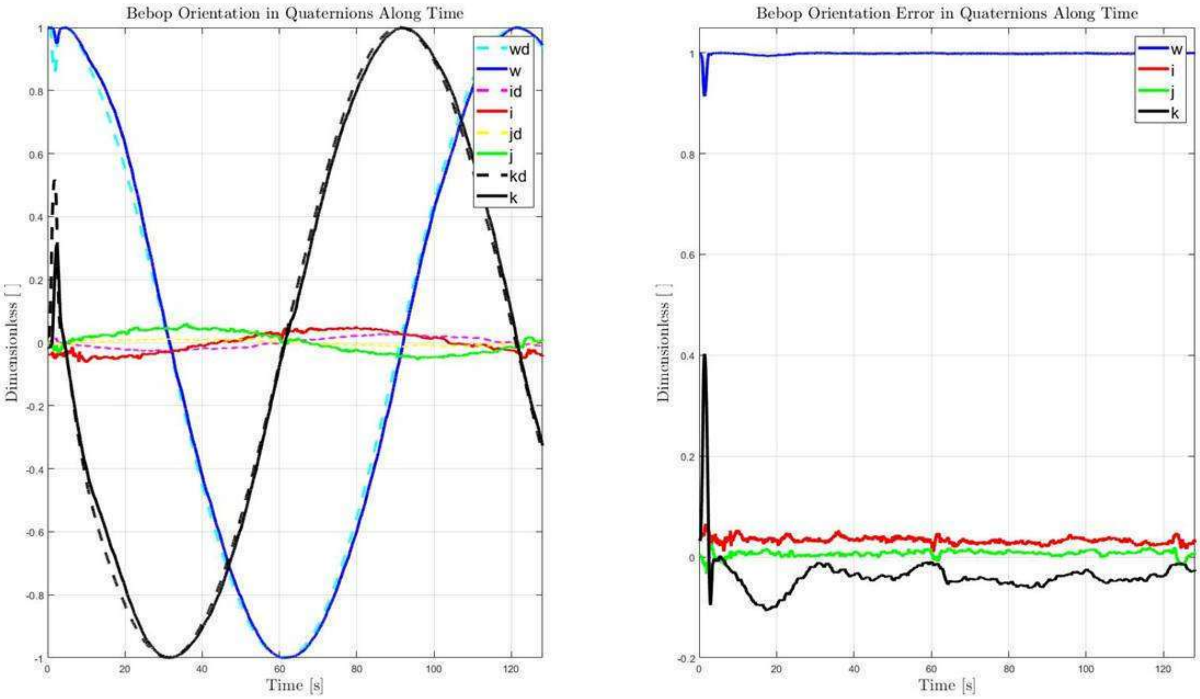


Figura 39 – Bebop Orientation in Quaternions Along Time for a Circular Trajectory Varying the Cluster Orientation

6 Conclusion

This thesis addresses the problem of controlling a formation composed by a UGV and a UAV, considering a situation in which the UAV should land on the UGV, when accomplishing trajectory-tracking tasks. Similarly to what is done in (RABELO; BRANDãO; SARCINELLI-FILHO, 2021) and (ERNANDES-NETO; SARCINELLI-FILHO; BRANDÃO, 2019), this situation corresponds to the last step of a cargo delivery application, in which the UAV gets back to its base and lands on it.

However, in such works undesirable problems arise. In the first case, the state of the UGV-UAV formation just before the landing of the UAV on the UGV, what means the UAV hovering above the UGV for a while, is a singularity for the Euclidean description of the formation there adopted, which can cause undesired instability at the landing moment. As for the second case, the position of the UAV with respect to the UGV is restricted, in consequence of the Euclidean description there adopted to describe the formation.

To circumvent such problems this thesis proposes a new description of the formation, based on quaternions, regarding the orientation of the straight line linking the two vehicles. As a consequence of such a formation description, a new controller is also proposed, to deal with the variables associated to the new description.

Finally, experiments are run, replicating the undesired conditions in (RABELO; BRANDãO; SARCINELLI-FILHO, 2021) and in (ERNANDES-NETO; SARCINELLI-FILHO; BRANDÃO, 2019), and the results show that the description here adopted for the formation, associated to the new controller proposed, is able to guarantee that the mentioned problems no more occur, thus validating the quaternion-based description and controller here proposed.

As future work it is planned to extend the quaternion-based description to consider not only the orientation of the virtual structure correspondent to the straight line linking the two robots, the UGV and the UAV, but also the UAV orientation as it is made through simulations in (MAS; KITTS, 2017). The inclusion of a dual-quatenion based description seems to be a natural way of improving the performance of the UAV-UGV formation, once it allows including the translation into the same element. Besides, it is also intended to extend the proposal here presented to path-following tasks, which is more suitable as motion control strategy for cargo delivery applications, for allowing to select the magnitude of the desired velocity.

Referências

ARIA, M. A survey of self-driving urban vehicles development. In: IOP PUBLISHING. *IOP Conference Series: Materials Science and Engineering*. [S.l.], 2019. v. 662, n. 4, p. 042006. Citado na página 16.

AYDIN, Y.; KUCUK, S. Quaternion based inverse kinematics for industrial robot manipulators with euler wrist. In: IEEE. *2006 IEEE international conference on mechatronics*. [S.l.], 2006. p. 581–586. Citado na página 20.

BACHETI, V. P.; BRANDÃO, A. S.; SARCINELLI-FILHO, M. Path-following with a ugv-uav formation considering that the uav lands on the ugv. In: IEEE. *2020 International Conference on Unmanned Aircraft Systems (ICUAS)*. [S.l.], 2020. p. 488–497. Citado na página 16.

BRAGANZA, D. et al. Tracking control for robot manipulators with kinematic and dynamic uncertainty. In: IEEE. *Proceedings of the 44th IEEE Conference on Decision and Control*. [S.l.], 2005. p. 5293–5297. Citado na página 20.

CARDOSO, V. B. et al. A large-scale mapping method based on deep neural networks applied to self-driving car localization. In: IEEE. *2020 International Joint Conference on Neural Networks (IJCNN)*. [S.l.], 2020. p. 1–8. Citado na página 16.

CHOU, J. C.; KAMEL, M. Finding the position and orientation of a sensor on a robot manipulator using quaternions. *The international journal of robotics research*, Sage Publications Sage CA: Thousand Oaks, CA, v. 10, n. 3, p. 240–254, 1991. Citado na página 21.

DAI, S.-L. et al. Adaptive leader–follower formation control of nonholonomic mobile robots with prescribed transient and steady-state performance. *IEEE Transactions on Industrial Informatics*, IEEE, v. 16, n. 6, p. 3662–3671, 2019. Citado na página 21.

DAPONTE, P. et al. Metrology for drone and drone for metrology: Measurement systems on small civilian drones. In: IEEE. *2015 IEEE Metrology for Aerospace (MetroAeroSpace)*. [S.l.], 2015. p. 306–311. Citado na página 16.

DUBS, H. Science and civilization in china: Volume 2, history of scientific thought. by joseph needham, frs, with the research assistance of wang ling, ph. d.(cambridge university press. 1956. pp. xxiv+ 697. price 80s. net.). *Philosophy*, Cambridge University Press, v. 35, n. 133, p. 167–168, 1960. Citado na página 15.

ERNANDES-NETO, V.; SARCINELLI-FILHO, M.; BRANDÃO, A. S. Trajectory-tracking of a heterogeneous formation using null space-based control. In: IEEE. *2019 International Conference on Unmanned Aircraft Systems (ICUAS)*. [S.l.], 2019. p. 187–195. Citado 7 vezes nas páginas 11, 21, 22, 51, 52, 73 e 78.

FIGUEREDO, L. et al. Robust kinematic control of manipulator robots using dual quaternion representation. In: IEEE. *2013 IEEE International Conference on Robotics and Automation*. [S.l.], 2013. p. 1949–1955. Citado na página 20.

FLOREANO, D.; WOOD, R. J. Science, technology and the future of small autonomous drones. *Nature*, Nature Publishing Group, v. 521, n. 7553, p. 460–466, 2015. Citado na página 16.

GANCHY, S. *Islam and science, medicine, and technology*. [S.l.]: The Rosen Publishing Group, Inc, 2009. Citado na página 15.

GAO, W. et al. Commanding cooperative ugv-uav with nested vehicle routing for emergency resource delivery. *IEEE Access*, IEEE, v. 8, p. 215691–215704, 2020. Citado na página 16.

GHAMRY, K. A.; KAMEL, M. A.; ZHANG, Y. Cooperative forest monitoring and fire detection using a team of uavs-ugvs. In: IEEE. *2016 International Conference on Unmanned Aircraft Systems (ICUAS)*. [S.l.], 2016. p. 1206–1211. Citado na página 16.

HANSON, A. J. *Visualizing Quaternions*. [S.l.]: Elsevier, 2006. Citado 3 vezes nas páginas 18, 33 e 35.

HART, V.; SEGERMAN, H. The quaternion group as a symmetry group. In: *The Best Writing on Mathematics 2015*. [S.l.]: Princeton University Press, 2016. p. 141–153. Citado na página 30.

HASSANALIAN, M.; ABDELKEFI, A. Classifications, applications, and design challenges of drones: A review. *Progress in Aerospace Sciences*, Elsevier, v. 91, p. 99–131, 2017. Citado na página 16.

HILL, D. R. Mechanical engineering in the medieval near east. *Scientific American*, JSTOR, v. 264, n. 5, p. 100–105, 1991. Citado na página 15.

ISAACSON. *Leonardo da Vinci*. [S.l.]: Intrínseca, 1ª edição, 2017. Citado na página 15.

JIA, Y.-B. Quaternions. *Course Com S*, v. 477, p. 577, 2019. Citado 2 vezes nas páginas 36 e 37.

KHALIL, H. *Nonlinear Systems*. [S.l.]: Prentice Hall, 2002. (Pearson Education). ISBN 9780130673893. Citado na página 58.

Kitts, C. A.; Mas, I. Cluster space specification and control of mobile multirobot systems. *IEEE/ASME Transactions on Mechatronics*, v. 14, n. 2, p. 207–218, 2009. Citado na página 18.

KUMAR, K. et al. An improved tracking using imu and vision fusion for mobile augmented reality applications. *arXiv preprint arXiv:1411.2335*, 2014. Citado na página 21.

LIU, T. P. *The weaver's knot*. [S.l.]: Cornell University Press, 2019. Citado na página 15.

MARCIANO, H. N.; BRANDÃO, A. S.; SARCINELLI-FILHO, M. Singularity-free quaternion representation to control a ugv-uav formation performing trajectory-tracking tasks. In: IEEE. *2021 International Conference on Unmanned Aircraft Systems (ICUAS)*. [S.l.], 2021. p. 656–665. Citado na página 57.

MAS, I.; KITTS, C. Quaternions and dual quaternions: Singularity-free multirobot formation control. 2017. Citado 6 vezes nas páginas 22, 54, 56, 57, 58 e 78.

MATHEW, N.; SMITH, S. L.; WASLANDER, S. L. Planning paths for package delivery in heterogeneous multirobot teams. *IEEE Transactions on Automation Science and Engineering*, IEEE, v. 12, n. 4, p. 1298–1308, 2015. Citado na página 16.

MOREIRA, M. S. M. Controle de uma formação vtnt-vant baseado em espaço nulo. Universidade Federal do Espírito Santo, 2020. Citado 2 vezes nas páginas 39 e 43.

MOREIRA, M. S. M.; BRANDÃO, A. S.; SARCINELLI-FILHO, M. Null space based formation control for a uav landing on a ugv. In: IEEE. *2019 International Conference on Unmanned Aircraft Systems (ICUAS)*. [S.l.], 2019. p. 1389–1397. Citado na página 16.

MURUGAN, D.; GARG, A.; SINGH, D. Development of an adaptive approach for precision agriculture monitoring with drone and satellite data. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, IEEE, v. 10, n. 12, p. 5322–5328, 2017. Citado na página 16.

NEVES, R. C. *Os quatérnios de Hamilton e o Espaço*. Tese (Doutorado) — Dissertação de Mestrado, 2008. Citado na página 26.

OXFORD, O. E. *Oxford English Dictionary*. [S.l.]: Oxford: Oxford University Press, 2009. Citado na página 15.

PALUNKO, I.; CRUZ, P.; FIERRO, R. Agile load transportation: Safe and efficient load manipulation with aerial robots. *IEEE Robotics Automation Magazine*, v. 19, n. 3, p. 69–79, September 2012. ISSN 1558-223X. Citado na página 16.

PEREIRA, P.; ABREU, A. C. Os teoremas de frobenius e hurwitz. Citado na página 28.

PHAM, H.-L. et al. Position and orientation control of robot manipulators using dual quaternion feedback. In: IEEE. *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*. [S.l.], 2010. p. 658–663. Citado na página 20.

Pinto, A. O. et al. High-level modeling and control of the bebop 2 micro aerial vehicle. In: *2020 International Conference on Unmanned Aircraft Systems (ICUAS)*. Athens, Greece: [s.n.], 2020. p. 939–947. Citado 2 vezes nas páginas 42 e 43.

PIZETTA, I. H. B.; BRANDãO, A. S.; SARCINELLI-FILHO, M. Cooperative load transportation using three quadrotors. In: *2019 International Conference on Unmanned Aircraft Systems (ICUAS)*. Atlanta, GA, USA: [s.n.], 2019. p. 644–650. ISSN 2373-6720. Citado na página 16.

Pizetta, I. H. B.; Brandão, A. S.; Sarcinelli-Filho, M. Load transportation by quadrotors in crowded workspaces. *IEEE Access*, v. 8, p. 223941–223951, 2020. Citado na página 16.

RABELO, M. F. S.; BRANDãO, A. S.; SARCINELLI-FILHO, M. Landing a UAV on static or moving platforms using a formation controller. *IEEE Systems Journal*, IEEE, v. 15, n. 1, p. 37–45, 2021. Citado 10 vezes nas páginas 11, 16, 18, 21, 22, 49, 50, 61, 71 e 78.

RADAVELLI, L. et al. A comparative study of the kinematics of robots manipulators by denavit-hartenberg and dual quaternion. *Mecánica Computacional*, v. 31, n. 15, p. 2833–2848, 2012. Citado na página 20.

RENAUDIN, V.; COMBETTES, C. Magnetic, acceleration fields and gyroscope quaternion (magyq)-based attitude estimation with smartphone sensors for indoor pedestrian navigation. *Sensors*, Multidisciplinary Digital Publishing Institute, v. 14, n. 12, p. 22864–22890, 2014. Citado na página 21.

ROSHEIM, M. E. *Robot evolution: the development of anthrobotics*. [S.l.]: John Wiley & Sons, 1994. Citado na página 15.

SANTANA, L. V.; BRANDãO, A. S.; SARCINELLI-FILHO, M. Navigation and cooperative control using the ar.drone quadrotor. *Journal of Intelligent & Robotic Systems*, v. 84, n. 1, p. 327–350, 2016. Citado na página 43.

SANTOS, M. C. P. et al. An adaptive dynamic controller for quadrotor to perform trajectory tracking tasks. *Journal of Intelligent & Robotic Systems*, Springer Nature BV, v. 93, n. 1-2, p. 5–16, 2019. Citado na página 58.

SARCINELLI, R. et al. Handling pedestrians in self-driving cars using image tracking and alternative path generation with frenét frames. *Computers & Graphics*, Elsevier, v. 84, p. 173–184, 2019. Citado na página 16.

SPONG, M. W.; VIDYASAGAR, M. *Robot Dynamics and Control*. 1st. ed. [S.l.]: Wiley, 1989. ISBN 9780471612438. Citado na página 20.

TADANO, S.; TAKEDA, R.; MIYAGAWA, H. Three dimensional gait analysis using wearable acceleration and gyro sensors based on quaternion calculations. *Sensors*, Multidisciplinary Digital Publishing Institute, v. 13, n. 7, p. 9321–9343, 2013. Citado na página 21.

VARGHESE, A.; CHANDRA, M. G.; KUMAR, K. Dual quaternion based imu and vision fusion framework for mobile augmented reality. In: IEEE. *2015 IEEE 9th International Symposium on Intelligent Signal Processing (WISP) Proceedings*. [S.l.], 2015. p. 1–6. Citado na página 21.

XIAN, B. et al. Task-space tracking control of robot manipulators via quaternion feedback. *IEEE Transactions on Robotics and Automation*, IEEE, v. 20, n. 1, p. 160–167, 2004. Citado na página 20.

XU, E. et al. Pioneer p3-dx robot to achieve self driving car. In: IEEE. *2018 IEEE International Conference on Electro/Information Technology (EIT)*. [S.l.], 2018. p. 0234–0239. Citado na página 16.

YAKIMENKO, O. A. et al. Precise autonomous aerial payload delivery system integrated with uav and ugv. In: ROYAL AERONAUTICAL SOCIETY, AUSTRALIAN DIVISION; ENGINEERS AUSTRALIA. *AIAC14: Fourteenth Australian Aeronautical Conference*. [S.l.], 2011. p. 49. Citado na página 16.

YANG, E.; GU, D.; HU, H. Nonsingular formation control of cooperative mobile robots via feedback linearization. In: IEEE. *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems*. [S.l.], 2005. p. 826–831. Citado na página 21.

YUN, X.; BACHMANN, E. R. Design, implementation, and experimental results of a quaternion-based kalman filter for human body motion tracking. *IEEE transactions on Robotics*, IEEE, v. 22, n. 6, p. 1216–1227, 2006. Citado na página 21.

YURTMAN, A.; BARSHAN, B.; FIDAN, B. Activity recognition invariant to wearable sensor unit orientation using differential rotational transformations represented by quaternions. *Sensors*, Multidisciplinary Digital Publishing Institute, v. 18, n. 8, p. 2725, 2018. Citado na página 21.

ZHANG, C.; KOVACS, J. M. The application of small unmanned aerial systems for precision agriculture: a review. *Precision agriculture*, Springer, v. 13, n. 6, p. 693–712, 2012. Citado na página 16.

ZHAO, F.; WACHEM, B. V. A novel quaternion integration approach for describing the behaviour of non-spherical particles. *Acta Mechanica*, Springer, v. 224, n. 12, p. 3091–3109, 2013. Citado na página 37.