CENTRO DE INVESTIGACIÓN Y DE ESTUDIOS AVANZADOS DEL INSTITUTO POLITÉCNICO NACIONAL

UNIDAD ZACATENCO

PROGRAMA DE SISTEMAS AUTÓNOMOS DE NAVEGACION AÉREA Y SUBMARINA

"Vehículo Cuadrirotor con Brazo Manipulador"

T E S I S

Que presenta

Hernán Abaunza González

Para obtener el grado de

MAESTRO EN CIENCIAS

EN SISTEMAS AUTÓNOMOS DE NAVEGACIÓN AÉREA Y SUBMARINA

Directores de la Tesis:

Dr. Rogelio Lozano Leal

Dr. Pedro Castillo García

México, D.F. Octubre, 2015

CENTRO DE INVESTIGACIÓN Y DE ESTUDIOS AVANZADOS DEL INSTITUTO POLITÉCNICO NACIONAL

UNIDAD ZACATENCO

PROGRAMA DE SISTEMAS AUTÓNOMOS DE NAVEGACION AÉREA Y SUBMARINA

"Quadrotor Vehicle with a Manipulator Arm"

T H E S I S

Presented by

Hernán Abaunza González

Submitted in fulfillment of the degree of

MASTER OF SCIENCE

IN

SISTEMAS AUTÓNOMOS DE NAVEGACIÓN AÉREA Y SUBMARINA

Thesis supervisors:

PhD. Rogelio Lozano Leal

PhD. Pedro Castillo García

México, D.F. October, 2015

"The best scientist is open to experience and begins with romance - the idea that anything is possible."

Ray Bradbury

Agradecimientos

Antes que nada, quiero agradecer a las instituciones que apoyaron este trabajo desde el inicio. Al Consejo Nacional de Ciencia y Tecnología (CONACyT), ya que por medio de su programa de becas para posgrados nacionales me brindó el apoyo económico suficiente para solventar mis estudios de maestría. Agradezco también al Centro de Investigación y de Estudios Avanzados del Instituto Politécnico Nacional (CINVESTAV - IPN), ya que mediante su programa de Sistemas Autónomos de Navegación Aérea y Submarina se desarrolló este proyecto. Especialmente agradezco a todos los integrantes del laboratorio UMI-LAFMIA, a los profesores que me brindaron las herramientas necesarias para trabajar en este campo de la ciencia, a los investigadores que me brindaron ayuda cuando me encontré con obstáculos, y a mis compañeros con los que he convivido durante estos años.

Quiero agradecer también a mis tutores, al los doctores Rogelio Lozano Leal y Pedro Castillo García, por haberme corregido en mis errores, guiado cuando no encontraba el camino a seguir, y presionado para continuar con el ritmo requerido para la conclusión de este trabajo. Agradezco al Dr. Rogelio por haberme permitido trabajar con él, y por brindarme un objetivo a seguir durante esta tesis, y al Dr. Pedro por haberme guiado una vez más en la ejecución del proyecto, y por haberme ayudado en cada paso hasta alcanzar el objetivo.

Agradezco a mi esposa Brenda Ivette García Maya por apoyarme en todo momento durante esta tesis, por acompañarme en momentos de estrés y de alegría, por levantarse conmigo cuando he tenido clases temprano, por soportar mis migrañas, y por escuchar mis historias sobre cuaterniones.

A mis padres Héctor Abaunza Gavidia y Naela González Benítez por haber procurado una buena eduación para mi y por apoyarme en toda decisión que he tomado, por brindarme consejos en todo momento, y por animarme a perseguir mis metas aunque el camino sea difícil.

Gracias a mi compañero y colega Jossué Cariño Escobar, por todos los proyectos que hemos desarrollado juntos, por las cosas que hemos aprendido en nuestras estancias y por toda la experiencia que hemos podido compartir. Especialmente por ayudarme a explorar el mundo de los cuaterniones, que después de todo resultaron no ser "magia obscura".

Dedicatoria

A mi familia, que ha estado siempre dispuesta a ofrecerme una mano cuando la he necesitado, a mi esposa Brenda, a mis padres Héctor y Naela, a mis hermanos Gaby, Héctor y Tatiana.

Contents

List of Figures

Abbreviations

DoF	**D**egree **o**f **F**reedom
UAV	**U**nmanned **A**erial **V**ehicle
MAV	**M**icro **A**erial **V**ehicle
VTOL	**V**ertical **T**ake-off and **L**anding
PVTOL	**P**lanar **V**ertical **T**ake-off and **L**anding
RF	**R**adio **F**requency
3D	**3**-**D**imensional
2D	**2**-**D**imensional
CAD	**C**omputer **A**ided **D**esign
CAM	**C**omputer **A**ided **M**anufacturing

Summary

Unmanned aerial vehicles have experienced huge advances in terms of scientific research and development, the implementation of more powerful digital electronics have opened the application of more complex sensing techniques and control schemes, thus permitting the conception of interesting applications.

Aerial manipulators have been proposed as an alternative to solve problems like complex object handling in large areas, or sample recollection in exploration missions. Some researchers have proposed mathematical models and experimental platforms consisting on UAVs with robotic arms, but their mathematical solutions tend to be really complex, and sometimes need difficult control schemes to properly function.

This thesis work proposes a mathematical model for a quadrotor vehicle with a manipulator arm, based on unit quaternions and dual quaternions. The obtained model is simpler in comparison with other works, thus permitting the application of simple control schemes for its stabilization.

Two control laws are proposed to fully stabilize the system, The first one is based on unit quaternions to separately stabilize the position and orientation of the vehicle, the second one is based on dual quaternions, which make possible to stabilize simultaneously the transformation of the UAV. The kinematic and dynamic effects of the robotic arm are considered as dual quaternions, and the torques that it applies on the vehicle are compensated in both of the proposed control laws.

The proposed model and control schemes are then validated with numerical simulations and in experiments in a real UAV prototype, demonstrating the stability of the platform in closed-loop system.

Key Words. Quaternion, Dynamics, Mechanical Arm, Manipulator, Dual Numbers, UAV, Drones, Multirotor, Quadrotor

Resumen

Los vehículos aéreos no tripulados han experimentado grandes avances científicos y tecnológicos en los últimos años, la implementación de componentes electrónicos más potentes ha permitido la aplicación de técnicas de sensado y esquemas de control más complejos, permitiendo la concepción de aplicaciones muy interesantes.

Los manipuladores aéreos han sido propuestos como una alternativa para resolver problemas como la manipulación de objetos en grandes áreas, o para la recolección de muestras en misiones de exploración. Algunos investigadores han propuesto modelos matemáticos y plataformas experimentales que consisten de vehículos aéreos con brazos robóticos, pero sus soluciones matemáticas tienden a ser muy complejas, y tienden a necesitar esquemas de control complejos para funcionar apropiadamente.

Este trabajo de tesis propone un modelo matemático para un vehículo cuadrirotor con brazo mecánico basado en cuaterniones unitarios y cuaterniones duales. El modelo obtenido es más simple en comparación con otros trabajos, por lo que permite la aplicación de esquemas de control más simples para su estabilización.

Dos leyes de control son propuestas para estabilizar completamente el sistema. La primera está basada en cuaterniones unitarios para estabilizar la posición y orientación del vehículo de manera separada, la segunda está basada en cuaterniones duales para lograr la estabilización simultanea de la transformación del quadrirotor. Los efectos cinemáticos y dinámicos del brazo mecánico son considerados usando cuaterniones duales, y los pares que causan en el vehículo son compensados en ambas leyes de control.

El modelo propuesto y los esquemas de control fueron validados con simulaciones numéricas y experimentos en un prototipo de vehículo cuadrirotor, demostrando la estabilidad de la plataforma con la implementación de las leyes de control propuestas.

Palabras Clave. Cuaterniones, Dinámica, Brazo Mecánico, Manipulador, Números Duales, UAV, Drones, Multirotor, cuadrirotor

Chapter 1

Introduction

In the last years, unmanned aerial vehicles (UAVs) have gained popularity in many areas including military, industry, scientific research, and even as end consumer applications.

UAVs offer great advantages in terms of speed and maneuverability in comparison to ground vehicles and other platforms, one of the most popular types of aerial platforms is the quadrotor (also known as quadcopter), which consists in a simple array of four motors and propellers. The quadrotor offers great maneuverability and the possibility of stabilization in a given position and orientation in the space.

1.1 Current Research Topics

During the last years, many researchers have developed new designs that take advantage of both rotary and fixed wing configurations to accomplish specific tasks and missions.

Some applications mentioned in Valavanis [2008] and in Cariño [2015] include:

- **Swarm Operation:** Several UAVs flying together with mutual communication that cooperate to perform certain tasks.

- **Fault-tolerant flight:** The capacity of an UAV to fly in the presence of faults using redundancy with automatic control to stabilize the system.
- **SLAM:** It refers "Simultaneous Location And Mapping", when an UAV utilizes its sensors to build a map and locate itself inside of it, see Cariño [2015].
- **UAV with attached manipulator:** A robotic manipulator is an actuator that manipulates objects, the most common configuration in industry is the robotic arm, which has a restricted workspace in a finite area. By adding the ability to fly in a wider space, a quadrotor with an attached robotic arm could provide much more flexibility for tasks that require movement in a more spacious environment. Prof. Anibal Ollero from the University of Seville, and his team (FADA-CATEC [2011]) have been working in this concept in the last years, and have accomplished many significant advances.

1.2 Problem Statement

The possibility of manipulate or transport payloads, instruments, and other objects using multi-rotor UAVs opens a variety of possibilities such as retrieving samples for exploration missions, or automated building of structures.

The problem begins with the fact that the total amount of degrees of freedom (DoF) of the system will be the sum of the DoF of the quadcopter and the robotic limb.

Although a quadrotor platform has six degrees of freedom (DoF), and it can move trough any position in the 3-dimensional space and is able to adopt any given orientation, only four of those DoF are stable, and two of them are unstable.

This can be seen in the simplified model for a quadrotor in terms of its mass, orientation, and control inputs given in Castillo et al. [2006] as

$$\begin{aligned} \ddot{x} &= -\tfrac{1}{m}\sin\theta u_1 \\ \ddot{y} &= \tfrac{1}{m}\cos\theta\sin\phi u_1 \\ \ddot{z} &= \tfrac{1}{m}\cos\theta\cos\phi u_1 \\ \ddot{\phi} &= \tau_\phi \\ \ddot{\theta} &= \tau_\theta \\ \ddot{\psi} &= \tau_\psi \end{aligned} \tag{1.1}$$

where θ, ϕ, and ψ denote the vehicle's orientation (in terms of Euler angles), $\tau_\phi, \tau_\theta, \tau_\psi$, and u_1 are the control torques and throttle input respectively, and x, y and z represent its position in the 3-D space.

Equation (1.1) shows that the equilibrium points of the system are restricted to the conditions $\theta = 0$ and $\phi = 0$, this means that the vehicle can be stabilized at any position in x, y and z as long as the conditions are respected.

By adding a 2 DoF robotic limb whose actuated axes are parallel to the θ and ψ angles of the quadrotor, then the complete system will be completely actuated, thus the 6 DoF of the end effector of the robotic arm will be stable.

1.3 Objectives

The main objective of this work is the modeling, control, simulation and experimental implementation of a quadrotor vehicle with an attached manipulator arm in terms of the position and orientation of its end actuator.

This can be divided in 6 secondary objectives:

1. Synthesize the kinematic and dynamic models for a quadrotor UAV with an attached 2 DoF robotic limb.

2. Obtain the desired orientation and position reference for the UAV from the desired reference of the robotic limb usind inverse kinematics.

3. Propose a control law that stabilizes the end effector of the arm in a desired position and orientation.

4. Corroborate the stability of the proposed control law with numeric simulations.

5. Build a quadrotor platform and a 2 DoF robotic arm that can be mounted in the frame.

6. Corroborate the stability of the proposed control law with an experimental validation on the proposed platform.

1.4 Methodology

Several approaches were considered to obtain the kinematic and dynamic models, it was observed that quaternions offer a practical and mathematically simple way to represent rotations in the 3D space, particularly dual quaternions are a convenient tool to describe multiple and simultaneous rotations and translations.

Firstly, the quadrotor dynamic model was obtained by using the Newton-Euler approach, and describing its rotational attitude with unit quaternions and its translational dynamics separately. A control law was proposed to exactly linearize the system, and then to globally stabilize it. Then, it was observed that it is necessary to apply multiple and simultaneous transformations (rotations and translations) to describe the orientation and position of the final actuator, so the possibility of using dual quaternions was explored.

The dynamic model of the quadrotor using dual quaternions was obtained also using the Newton Euler approach, the kinematic and dynamic model of the complete platform (including the robotic limb) was then obtained using also dual quaternions.

A dual quaternion regulator was then proposed to stabilize the system in position and orientation simultaneously, an algebraic relationship between the quadrotor's rotational and translational dynamics was implemented to globally stabilize the system.

Both proposed models and control laws were simulated using Python programming language.

Lastly, the results were validated in the experimental platform that was built in the context of the project.

1.5 State of the Art

1.5.1 Quadrotor Systems

Quadrotors are one of the most popular platforms for UAV research due to their mathematical simplicity, flight capabilities, and relatively low-cost implementation. Many researchers have made important advances with quadcopters in the last years.

Castillo et al. [2006] presented an important contribution where they explained in a didactic manner, the modeling and control for many types of UAVs including the PVTOL, mini-helicopters, and quadrotors (among others).

In García et al. [2010] the authors propose a position stabilization method based in vision algorithms which estimate the UAV's position in the 3D space with respect to a landing pad in the ground.

The main problem that needs to be resolved in any application of quadcopters is the stabilization of the platform in both orientation and position. Many control approaches have been proposed to solve this problem, such as simple PID controllers as in Junior et al. [2013], others have proposed non-linear control approaches such as Fermi Guerrero Castellanos [2013].

More complex control schemes have also been proposed such as backstepping-based inverse optimal controls (see Wang Jian and Hongxu [2013]) among many others. Part of this complexity is justified because the most used quadcopter model is based on the description of its rotation using Euler angles, other approaches can be explored such that the system's description becomes simpler thus more easy to stabilize.

1.5.2 Quadrotor Manipulator Systems

Because of all of the benefits and offered possibilities quadrotor manipulators have been an important focus of interest for many researchers in the last years.

Ghadiok [2011] presented in his Masters thesis a simple gripper manipulation system that implements autonomous indoor aerial gripping using a low-cost quadrotor. The major challenges that this project had to overcome were the precise positioning, sensing and manipulation trough the stabilization of disturbances due to the interaction with the object. Position sensing was achieved by using Simultaneous Location and Mapping (SLAM).

Khalifa et al. [2012] presented the modeling and control of a quadrotor platform with a robotic manipulator, this team has presented various contributions in which they proposed different control approaches to control their platform as shown in Khalifa et al. [2013] and Khalifa et al. [2015].

A different quadrotor platform that uses a buoyancy envelope is presented in Korpela et al. [2012] in order to provide a better stability in the system. The team has proposed also a complex 16 DoF manipulator which consists in multiple individual manipulators.

Kim et al. [2013] obtained a kinematic and dynamic model using Euler angles for a quadrotor with a two DoF robotic arm, and then stabilized the complete system using an adaptive sliding mode controller.

Similarly, Arleo et al. [2013] proposed a kinematic and a dynamic model using Euler angles to describe the position and orientation of the complete system. A feedback regulator is proposed to stabilize the system.

Other works like Yang and Dongjun [2014] and Danko and Oh [2014] have explored more ambitious propositions in which manipulators have more actuated joints, thus adding complexity with redundant DoF.

Orsag et al. [2011] presented a different design for the manipulator where three 2-DoF manipulators are implemented to interact with the environment using hook ad push/pull actions on objects.

One particular work that calls special attention is the european project called ARCAS (Aerial Robotics Cooperative Assembly System), which proposes the development and experimental validation of a cooperative free-flying robot system for assembly and structure construction, with a wide variety of possible applications going from the building of platforms for evacuation of people, to the inspection and maintenance of facilities (among many others). This project is conformed by a consortium of european universities and research centers from France, Spain, Italy, Germany and Switzerland, they have published more than 40 scientific articles since 2011 covering many related topics such as mechanical design (see for example Cano [2013],), modeling, control, trajectory planning, optimization, and vision.

FIGURE 1.1: Aerial 6 DoF manipulator proposed by the ARCAS project

Many of the reviewed works have accomplished very interesting and promising results, however, the models they proposed were usually very large, non-linear, and their inverse kinematics were difficult to calculate, this is because the most common approach to represent multiple rotations and translations (as when working with robotic manipulators) is trough Euler Angles.

1.5.3 Quaternion-based Approaches

As it will be explained in sections 2.2.1 and 2.3.1, unit quaternions offer a very convenient way to express rotations of a rigid body, and dual quaternions expand this advantage to simultaneous rotations and translations, see Spring [1986], Kuipers [1999], Nojan Madinehi [2013] and Horn [1987].

Several applications of quaternions in the modeling of UAVs (particularly multicopter platforms) have been proposed by various researchers.

For example, a linear model is obtained trough a special feedback linearization in Long et al. [2012] where the utilized platform is an omnicopter MAV.

Alaimo et al. [2013] use unit quaternions to describe the dynamic model of a hexacopter, the authors use a relatively simple feedback controller to stabilize the system.

A similar modeling approach is presented in Fresk and Nikolakopoulos [2013] where quaternions are used to describe the dynamic model of a quadrotor. The authors implement a proportional squared control to stabilize the attitude of the vehicle. An LQR regulator is similarly proposed in Reyes-Valeria et al. [2013] parting from the quaternion-based model for a quadrotor.

Our team has published a contribution where we propose the modeling and control of a quadrotor platform using unit quaternions and a special trajectory to globally stabilize the system (in both position and orientation), see Cariño et al. [2015].

Dual quaternions are also being studied to describe simultaneously the attitude and position of UAVs. Zhao et al. [2014] implemented dual quaternions to control the flight of a fixed-wing UAV.

Wang and Yu [2010] proposed a feedback linearization regulator using unit dual quaternions, although it is not directly applicable to quadrotors, it provides a basis to implement a simple feedback controller to globally stabilize any UAV platform.

In Nuno and Tsiotras [2013] the authors proposed a control that simultaneously stabilizes position and attitude using dual quaternions, they show that this implementation is simple and relatively straightforward.

If the description of multiple rotations and translations is required when a quadcopter is combined with a robotic limb, then dual quaternions seem to provide a simple system description which simplifies modeling and control.

Chapter 2

Quaternion Modeling

Quaternions are a very useful mathematical tool for representing the rotation of a rigid body, they have great advantages with respect to the more commonly used Euler angles representation such as lack of discontinuities and mathematical simplicity. A brief description of quaternion algebra is presented in section 2.2.1.

Simultaneous rotation and translation (also called transformation) can be described using unitary quaternions if a vector in the body frame is rotated to the inertial frame using a quaternion product. Since the objective of this work is to obtain the model and control for a quadrotor with a robotic arm with various links, the transformation of all of the elements of the platform would ve very complex, thus another mathematical tool is used to simplify the modeling of the robotic arm.

Dual Quaternions are another mathematical tool used to describe the transformation of a rigid body, they also offer mathematical simplicity, and more simple calculations for the kinematics of robotic arms. A brief description of dual quaternion algebra is presented in section 2.3.1.

3 sections compose this chapter; in the first one, previous work is presented, in the next one the mathematical model of the vehicle is obtained using unitary quaternions, in the last the model is acquired using the dual quaternion approach. Firstly the

equations only consider the aerial platform, after that, the mathematical model is completed with the inclusion of the robotic arm.

2.1 Previous Work

Many researchers have made efforts to model the dynamics of aerial vehicles, in Castillo et al. [2006], a detailed method for modeling a quadrotor is presented using Euler angles representation, in this book, an equivalency between quaternions and Euler angles is presented as a mathematical tool for control design.

Not many authors have adopted the quaternion-based representation for modeling aerial vehicles, probably because, in contrast with the Euler angles representation, they lack of a direct intuitive visualization for the rotation.

Nevertheless, some researchers have noticed that the advantages of quaternions overpass the disadvantages. For example, Fresk and Nikolakopoulos [2013] present a full attitude control based on quaternions using a state feedback from the axis-angle representation of the orientation, Alaimo et al. [2013] introduce a mathematical model for an hexarotor vehicle using the Euler-Lagrange approach, Izaguirre-Espinosa [2015] proposes a technique for stabilizing the position-yaw tracking of quadrotors, although they use a traditional Euler angles representation for the vehicle's attitude, their control for the z axis is presented using a quaternion approach. Other control techniques such as optimal, LQR, and feedback linearization are presented in Wang Jian and Hongxu [2013], Reyes-Valeria et al. [2013], and David J. Cappelleri [2012].

Other authors have noticed that the dual quaternion approach can provide advantages when stabilizing both orientation and position of a rigid body, Jin and Wang [2013] presents an introduction to dual quaternions, and a modeling of a quadrotor UAV using the Lagrangian formulation, Nuno and Tsiotras [2013] proposes a feedback control without linear and angular velocity feedback using dual quaternions, and finally, Wang and Yu [2010] propose a feedback regulator that globally stabilizes a

fully actuated system using dual quaternions, this work can be expanded for use with under-actuated systems (such as a quadrotor).

2.2 Model Using Unit Quaternions.

2.2.1 Quaternion Background

Quaternions are "hypercomplex" numbers, which means that they have three imaginary parts $\hat{i}, \hat{j}, \hat{k}$ instead of one compared to complex numbers. They can be used to describe in a very simple mathematical and computational way rotations in three-dimensional space. When many methods use trigonometric functions, which are non-linear and suffer from numerical inaccuracy, quaternion rotations are simple in that they only need multiplications, divisions and sums to be implemented.

2.2.1.1 Notation.

In this work, over lined letters represent vectors in 3D space $\overline{(\cdot)} \in \mathbb{R}^3$. A quaternion is a four tuple that belongs to the $\mathbb{H}$ quaternion space. It can be seen as a number that contains one real part and three imaginary parts multiplied by their corresponding imaginary units $\hat{i}, \hat{j}, \hat{k} \in \mathbb{I}$

$$\begin{aligned} &q_0, q_1, q_2, q_3 \in \mathbb{R}; \boldsymbol{q} \in \mathbb{H}; \hat{i}, \hat{j}, \hat{k} \in \mathbb{I} \\ &\boldsymbol{q} \coloneqq q_0 + q_1\,\hat{i} + q_2\,\hat{j} + q_3\,\hat{k} \end{aligned} \tag{2.1}$$

As there are three different imaginary parts, these are often viewed as a vector in $\mathbb{R}^3$ space. Thus, $\mathbb{R}^3$ space can be seen as a subspace of $\mathbb{H}$ space and a $\mathbb{R}^3$ vector can be considered a pure imaginary quaternion.

$$\begin{aligned} &\overline{q} \in \mathbb{R}^3 \\ &\boldsymbol{q} = q_0 + \begin{bmatrix} q_1 \\ q_2 \\ q_3 \end{bmatrix} = q_0 + \overline{q} \end{aligned} \tag{2.2}$$

2.2.1.2 Quaternion Operations.

Because of its significance, historically as well as in the definition of the quaternion space itself, the main operation of quaternions is the multiplication. Other operations and properties arise from this definition, like the conjugate and the norm.

Product. The quaternion product between quaternions $\boldsymbol{q}, \boldsymbol{p} \in \mathbb{H}$, expressed as a sum between a scalar real part and an imaginary vector $\boldsymbol{q} = q_0 + \overline{q}; \boldsymbol{p} = p_0 + \overline{p}$, is defined in the following manner

$$\boldsymbol{q} \otimes \boldsymbol{p} \coloneqq (q_0 + p_0 - \overline{q} \cdot \overline{p}) + (q_0\,\overline{p} + p_0\,\overline{q} + \overline{q} \times \overline{p}) \tag{2.3}$$

Some properties can be seen from this definition. One of the most important is that quaternion product is not commutative. Which means that $\boldsymbol{q} \otimes \boldsymbol{p} \neq \boldsymbol{p} \otimes \boldsymbol{q}$. This is because of the same non-commutativity property of the cross product used on the definition.

Sum. The sum of quaternions $\boldsymbol{q}$ and $\boldsymbol{p}$ is simply defined as the sum of each of its elements, like this

$$\begin{aligned} &\boldsymbol{q}, \boldsymbol{r} \in \mathbb{H} \\ &\boldsymbol{q} + \boldsymbol{r} \coloneqq q_0 + r_0 + \overline{q} + \overline{r} \end{aligned} \tag{2.4}$$

The set of all quaternions with operations addition and multiplication defines a non-commutative division ring. See Kuipers [1999] for more information on this matter.

Conjugate. The conjugate of quaternion $\boldsymbol{q}$ is denoted by

$$\boldsymbol{q}^* \coloneqq q_0 - \overline{q} \tag{2.5}$$

The conjugate of a product of quaternions is

$$(\boldsymbol{q} \otimes \boldsymbol{r})^* = \boldsymbol{r}^* \otimes \boldsymbol{q}^* \tag{2.6}$$

This can be proved by expanding the corresponding products.

Norm. The norm of a quaternion is defined by

$$||\boldsymbol{q}||^2 \coloneqq \boldsymbol{q} \otimes \boldsymbol{q}^* = q_0^2 + q_1^2 + q_2^2 + q_3^2 \tag{2.7}$$

Inverse. The quaternion product forms a closed-loop group, that is, the product of two non-null quaternions is another quaternion. This means that for any non-null quaternion there exists an inverse quaternion such that

$$\begin{aligned} \boldsymbol{q}^{-1} &\coloneqq \frac{\boldsymbol{q}^*}{||\boldsymbol{q}||} \\ \boldsymbol{q} \otimes \boldsymbol{q}^{-1} &= \boldsymbol{q}^{-1} \otimes \boldsymbol{q} = 1 \end{aligned} \tag{2.8}$$

Derivative Let r be any given vector (quaternion with zero scalar part) fixed in an initial reference frame . let r' be the same vector but rotated to another reference frame such that

$$r' = \boldsymbol{q}^{-1} \otimes r \otimes \boldsymbol{q} \tag{2.9}$$

If we differentiate (2.9), then

$$\dot{r}' = \dot{\boldsymbol{q}}^{-1} \otimes r \otimes \boldsymbol{q} + \boldsymbol{q}^{-1} \otimes r \otimes \dot{\boldsymbol{q}} \tag{2.10}$$

From equations (2.9) and (2.10) we obtain

$$\dot{r}' = \dot{\boldsymbol{q}}^{-1} \otimes \boldsymbol{q} \otimes r' + r' \otimes \boldsymbol{q}^{-1} \otimes \dot{\boldsymbol{q}} \tag{2.11}$$

Since $\boldsymbol{q}$ is a unitary quaternion, then $\boldsymbol{q}^{-1} \otimes \boldsymbol{q} = 1$, and

$$\dot{\boldsymbol{q}}^{-1} \otimes \boldsymbol{q} + \boldsymbol{q}^{-1} \otimes \dot{\boldsymbol{q}} = 0 \tag{2.12}$$

Then it follows from (2.11) and (2.12) that

$$\dot{r}' = r' \otimes \boldsymbol{q}^{-1} \otimes \dot{\boldsymbol{q}} - \boldsymbol{q}^{-1} \otimes \dot{\boldsymbol{q}} \otimes r' \tag{2.13}$$

Now, the scalar part (real part) of $\boldsymbol{q}^{-1} \otimes \dot{\boldsymbol{q}}$ is

$$Re(\boldsymbol{q}^{-1} \otimes \dot{\boldsymbol{q}}) = \dot{\boldsymbol{q}}_0 \boldsymbol{q}_0 + \dot{\boldsymbol{q}}_1 \boldsymbol{q}_1 + \dot{\boldsymbol{q}}_2 \boldsymbol{q}_2 + \dot{\boldsymbol{q}}_3 \boldsymbol{q}_3 = 0$$

Thus, we can say that the product $\boldsymbol{q}^{-1} \otimes \dot{\boldsymbol{q}}$ is a vector (a quaternion with real part equal to zero).

Then, we can say that

$$\dot{r}' = r' \otimes \boldsymbol{q}^{-1} \otimes \dot{\boldsymbol{q}} - \boldsymbol{q}^{-1} \otimes \dot{\boldsymbol{q}} \otimes r' = 2(\boldsymbol{q}^{-1} \otimes \dot{\boldsymbol{q}}) \times r' \tag{2.14}$$

Since $\dot{r}'$ is the translational velocity of the vector, we can say that $\dot{r}' = \omega \times r'$, where ω is the rotational velocity of r', thus

$$\omega \times r' = 2(\boldsymbol{q}^{-1} \otimes \dot{\boldsymbol{q}}) \times r' \tag{2.15}$$

Since r' can be any vector, the expression is reduced to

$$\omega = 2(\boldsymbol{q}^{-1} \otimes \dot{\boldsymbol{q}}) \Rightarrow \dot{\boldsymbol{q}} = \frac{1}{2}\boldsymbol{q} \otimes \omega \tag{2.16}$$

2.2.2 Unit Quaternions.

If the norm of $\boldsymbol{q}$ it has unitary norm $||\boldsymbol{q}|| = 1$, then it can be called unitary quaternion. Unitary quaternions are often used to represent rotations in 3D space because they offer some advantages over other methods of representation such as lack of singularities. They do not have gimball-lock effect, and they have mathemathical and computational simplicity because all the operations need only multiplications and sums.

2.2.2.1 Euler-Rodríguez Equation of Rotation.

Euler stated in his theorem for rotation of rigid bodies that any rotation of a rigid body can be expressed as a rotation with respect to a fixed axis and a certain amount or angle. This rotation in a 3D space can be represented using unitary quaternions as

$$\begin{aligned} \overline{p}' &= \boldsymbol{q}^{-1} \otimes \overline{p} \otimes \boldsymbol{q} = \boldsymbol{q}^{*} \otimes \overline{p} \otimes \boldsymbol{q} \\ \boldsymbol{q} &\coloneqq \cos\left(\frac{\theta}{2}\right) + \overline{u}\sin\left(\frac{\theta}{2}\right) \end{aligned} \tag{2.17}$$

where $\overline{p} \in \mathbb{R}^3$ is a 3D vector in the original reference frame, $\overline{p}'$ represents the same vector as $\overline{p}$ but now with respect to a new reference frame. $\overline{u} \in \mathbb{R}^3$ means the direction of the axis of rotation. θ defines the angle of rotation around the axis of rotation.

In equation (2.17), it can be seen that a double quaternion product can be used to rotate any vector from one reference frame into another, and this rotation does not affect the vector's magnitude.

It can be seen that a quaternion $\boldsymbol{q}$ gives the same rotation as the quaternion $-\boldsymbol{q}$. If we imagine the fixed axis $\bar{u}$, it can be easily seen that two rotation magnitudes with respect to this axis can translate to the same orientation, those are θ and $-2\pi + \theta$ since $\boldsymbol{q} = \cos\left(\frac{\theta}{2}\right) + \overline{u}\sin\left(\frac{\theta}{2}\right)$ and $-\boldsymbol{q} = \cos\left(\frac{-2\pi+\theta}{2}\right) + \overline{u}\sin\left(\frac{-2\pi+\theta}{2}\right)$. This duality can be used to assure that the rotation is applied with the smallest magnitude as possible.

2.2.2.2 Axis-Angle representation

From the Euler-Rodríguez formula (2.17) it is possible to obtain a direct relationship between the so called "Axis-Angle" representation and unit quaternions for certain rotation in order to express the rotation as a single vector $\overline{\theta} \in \mathbb{R}^3$, with $||\overline{\theta}|| = \theta$ representing the rotational magnitude and $\overline{u} = \dfrac{\overline{\theta}}{||\overline{\theta}||}$. This representation gives a more intuitive way to understand the rotation of the rigid body.

This relationship can be represented by the quaternion logarithmic mapping, which is given by

$$\ln \boldsymbol{q} := \begin{cases} \ln q_0 & , \text{if} \quad ||\overline{q}|| = 0 \\ \ln ||\boldsymbol{q}|| + \dfrac{\overline{q}}{||\overline{q}||} \arccos \dfrac{q_0}{||\boldsymbol{q}||} & , \text{if} \quad ||\overline{q}|| \neq 0 \end{cases} \tag{2.18}$$

$$||\boldsymbol{q}|| = 1$$

$$\ln \boldsymbol{q} = \frac{\overline{q}}{||\overline{q}||} \arccos q_0$$

It can be seen that if we are working only with unitary quaternions ($||\boldsymbol{q}|| = 1$), then the logarithmic mapping is reduced to

$$\ln \boldsymbol{q} = \frac{\overline{q}}{||\overline{q}||} \arccos q_0 \tag{2.19}$$

Using equation (2.19) and (2.17), we have the following axis-angle representation from an unit quaternion

$$\overline{\theta} = 2 \ln \boldsymbol{q} \tag{2.20}$$

The inverse relationship that can return a quaternion from its axis angle representation is called quaternion exponential mapping, it can be said that this is the inverse of the quaternion logarithmic mapping

$$\boldsymbol{e}^{\boldsymbol{q}} := \begin{cases} ||\overline{q}|| = 0, & \boldsymbol{e}^{q_0} \\ ||\overline{q}|| \neq 0, & \boldsymbol{e}^{||\boldsymbol{q}||} \left(\cos \dfrac{||\boldsymbol{q}||}{2} + \dfrac{\overline{q}}{||\overline{q}||} \sin \dfrac{||\boldsymbol{q}||}{2} \right) \end{cases} \tag{2.21}$$

2.2.3 Quadcopter Dynamic Model.

A quadcopter can be considered as a rigid body that has 6 DoF, but only four of them are stable, this is because the platform can not move in the orientation states without affecting its position.

The force that the motors apply to the vehicle with respect to the body reference frame can only act in the z axis, but its orientation in the inertial frame can be changed by controlling the orientation of the platform.

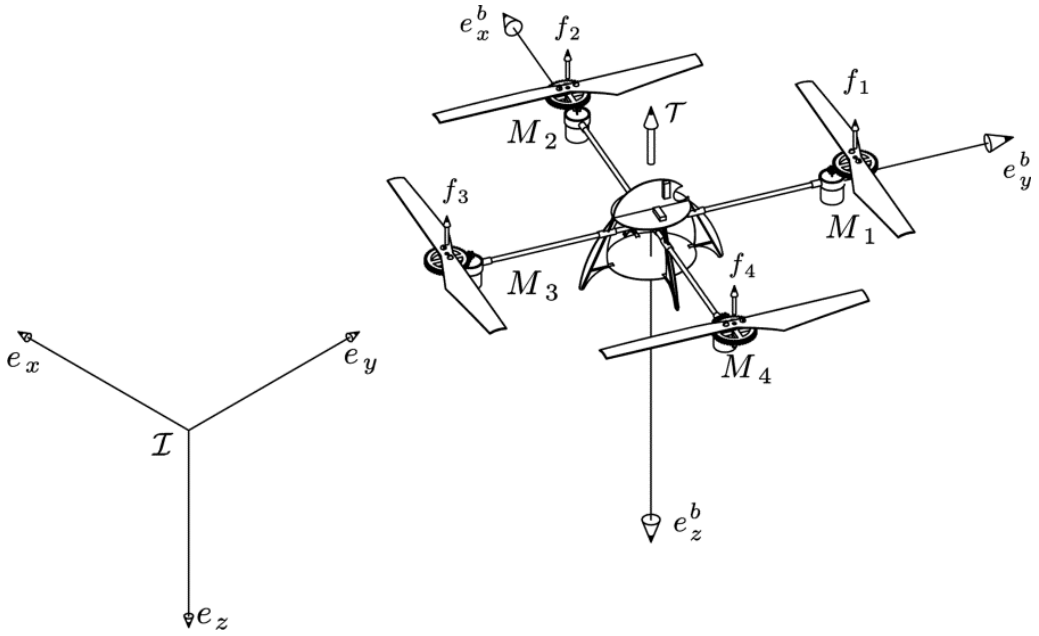

FIGURE 2.1: Quadrotor body diagram. Image taken from Izaguirre-Espinosa [2015]

It is important to note that the forces and torques that the motors apply to the body are given by

$$\begin{aligned} f_i = &\quad k_i\,\omega_{M_i}^2 \\ \tau = &\begin{bmatrix} l\,(f_1 + f_4 - f_2 - f_3) \\ l\,(f_1 + f_2 - f_3 - f_4) \\ \sum_{i=1}^{4} (-1)^{i+1}\,\tau_i \end{bmatrix} \end{aligned} \tag{2.22}$$

where $\omega, \tau \in \mathbb{R}^3$ define the angular velocity and the applied torque respectively, $l = \dfrac{L}{\sqrt{2}}$, L means the distance between the vehicle's center of mass and any motor (the platform is supposed to be symmetrical), $f_i, \tau_i, k_i, \omega_{M_i} \in \mathbb{R}, i = 1, 2, 3, 4$ represent the force, torque, aerodynamic constant, and angular velocity of each one of the four motors.

As any rigid body, the state vector of the quadrotor can be expressed in terms of its position and its orientation as

$$x_{quad} \coloneqq \begin{bmatrix} p^T & \dot{p}^T & \boldsymbol{q}^T & \omega^T \end{bmatrix}^T \tag{2.23}$$

where

- $p = \begin{bmatrix} x & y & z \end{bmatrix}^T$ is the translational position in the inertial frame.
- $\dot{p} = \begin{bmatrix} \dot{x} & \dot{y} & \dot{z} \end{bmatrix}^T$ denotes the translational velocity vector.
- $\boldsymbol{q} = q_0 + \begin{bmatrix} q_1 & q_2 & q_3 \end{bmatrix}^T$ defines the vehicle orientation with respect to the inertial frame, represented as a unit quaternion.
- $\omega = \begin{bmatrix} \omega_x & \omega_y & \omega_z \end{bmatrix}^T$ represents the rotational velocity in the body frame.

Therefore, the dynamic model can be separated into two subsystems, the first one corresponds to the rotational and the second one to the translational dynamics.

2.2.3.1 Quaternion Rotational Model.

The state vector for the rotational dynamics must be defined

$$x_{rot} \coloneqq \begin{bmatrix} \boldsymbol{q}^T & \omega^T \end{bmatrix}^T \tag{2.24}$$

The rotational dynamical model is obtained by the differentiation of $\boldsymbol{q}$ in equation (2.16)

$$\dot{x}_{rot} = \begin{bmatrix} \dot{\boldsymbol{q}} \\ \dot{\omega} \end{bmatrix} = \begin{bmatrix} \frac{1}{2}\boldsymbol{q} \otimes \omega \\ J^{-1}\left(\tau - \omega \times J\,\omega\right) \end{bmatrix} \tag{2.25}$$

where J represents the inertia matrix, $\dot{\omega}$ is obtained from Newton-Euler equations of motion, see Goldstein [1962]. $\tau \in \mathbb{R}^3$ defines the vector that contains the torques

applied to the body, these torques are the sum of the control input torques and the external torques. Then we can say that $\tau = \tau_u + \tau_{ext}$.

Since this work consists in the addition of a robotic arm to the quadcopter, we can say that τ_{ext} represents the torques that the arm apply to the vehicle.

Equilibrium Points. Equilibrium points are defined as a constant solution to a differential equation, in order to find an equilibrium point, it is necessary to find the states and control inputs x^*, τ^* where $\dot{x} = 0$.

Taking this into account, the equilibrium points of the attitude system (2.25) can be calculated when $\dot{x}_{rot} = \overline{0}$, thus it follows that

$$\Rightarrow \quad x^*_{rot} = \begin{bmatrix} \boldsymbol{q}^* \\ \overline{0} \end{bmatrix}, \tau^* = \overline{0} \tag{2.26}$$

This corroborates the intuitive approach for finding the equilibrium points; when the vehicle's orientation is constant, the torques and the angular velocities must be equal to zero.

2.2.3.2 Quadcopter Translational Dynamics.

The state variable for the translational model is given by

$$x_{pos} = \begin{bmatrix} p^T & \dot{p}^T \end{bmatrix}^T \tag{2.27}$$

According to Newton's equations, the total force that acts on the body in the inertial frame can be obtained multiplying the acceleration and the mass (note that F_t is expressed in the body frame)

$$\boldsymbol{q} \otimes F_t \otimes \boldsymbol{q}^* = m\ddot{p} \Rightarrow \ddot{p} = F_t$$

The total force that acts on the vehicle consists on the sum of the control force F_u, and the external forces F_{ext} expresed by $F_t = F_u + F_{ext}$.

Since the quadrotor is a under-actuated system, the only force in the body reference frame that can be used to control the platform is the trust force vector F_u, in this case F_{ext} corresponds to the gravity force, which acts with respect to the inertial frame, thus

$$\dot{x}_{pos} = \begin{bmatrix} \dot{p} \\ \ddot{p} \end{bmatrix} = \begin{bmatrix} \dot{p} \\ \boldsymbol{q} \otimes \dfrac{F_u}{m} \otimes \boldsymbol{q}^* + \overline{g} \end{bmatrix} \tag{2.28}$$

with

$$F_u = \begin{bmatrix} 0 & 0 & \sum_{i=1}^{4} f_i \end{bmatrix}^T \tag{2.29}$$

where m stands for the quadcopter's mass, and $\overline{g} = \begin{bmatrix} 0 & 0 & g \end{bmatrix}^T$ corresponds to the gravity's vector in the inertial frame.

Joining models (2.25) and (2.28), as shown in the derivate of the state from equation (2.23), the complete model for the quadrotor is obtained

$$\dot{x}_{quad} = \frac{d}{dt} \begin{bmatrix} p \\ \dot{p} \\ \boldsymbol{q} \\ \omega \end{bmatrix} = \begin{bmatrix} \dot{p} \\ \boldsymbol{q} \otimes \dfrac{F_u}{m} \otimes \boldsymbol{q}^* + \overline{g} \\ \frac{1}{2}\, \boldsymbol{q} \otimes \omega \\ J^{-1}\, (\tau - \omega \times J\, \omega) \end{bmatrix} \tag{2.30}$$

Since the thrust vector is fixed in the body frame, and the attitude is variable according to equation (2.25), then the position can be controlled using the attitude subsystem, thus the platform can be globally stabilized.

2.2.4 Quadrotor with Jointed Arm Effects

We already know that τ_{ext} represents the torques that the arm apply to the vehicle, we consider that the robotic arm has two links, the first one is constant, and goes

from the center of mass of the vehicle. The second one has two rotational axises that rotate between the x and y axis of the vehicle reference frame.

From this statement, we can say that equation (2.31) is true, where τ_c is the torque that the constant link causes to the vehicle, and τ_b is the torque caused by the rotational link.

$$\tau_{ext} = \tau_c + \tau_b \tag{2.31}$$

The first torque is given by the product $\tau = r \times F$, which says that any torque can be calculated using the cross product between the force that causes it, and the radius from the point where F is applied, and the rotational center, thus

$$\tau_c = r_{cv} \times m_c \vec{g}_v + J_c \omega_v$$

where r_{cv} is the constant distance from the center of mass of the quadrotor, to the center of rotation of the arm's first motor, m_c is the mass of the link, J_c is its inertia matrix, and $\vec{g}_v$ is the gravity acceleration vector in the vehicle's reference frame.

$$\bar{g}_v = \boldsymbol{q}_v^* \otimes \bar{g} \otimes \boldsymbol{q}_v$$

So the final expression for τ_c becomes

$$\tau_c = r_{cv} \times m_c[\boldsymbol{q}_v^* \otimes \vec{g} \otimes \boldsymbol{q}_v] + J_c \omega_v \tag{2.32}$$

The second torque is similarly given by

$$\tau_b = r_{bvt} \times m_b \vec{g}_v + J_b(\omega_v + \omega_b)$$

where r_{bvt} is the total distance from the center of mass of the quadrotor to the center of mass of the second link, and is given by $r_{bvt} = r_c + r_{bv}$, and ω_b is the angular

velocity with respect to the first motor of the arm, m_b is the mass of the link, and J_b is it's inertia matrix.

r_{bv} represents the position of the center of mass of the arm with respect to its first articulation, in the vehicle's reference frame

$$r_{bv} = \boldsymbol{q}_b \otimes r_b \otimes \boldsymbol{q}_b^*$$

where $\boldsymbol{q}_b$ represents the rotation of the arm with respect to the body frame.

So the final expression for τ_b comes to

$$\tau_b = (\boldsymbol{q}_b \otimes r_b \otimes \boldsymbol{q}_b^* + r_c) \times m_b \vec{g}_v + J_b(\omega_v + \omega_b) \tag{2.33}$$

2.3 Model Using Dual Quaternions.

2.3.1 Quaternion Background

2.3.1.1 Dual Numbers.

A dual number is defined as

$$\begin{aligned} &a, b \in \mathbb{R}, \epsilon \neq 0, \epsilon^2 = 0 \\ &\hat{a} = a + \epsilon b \end{aligned} \tag{2.34}$$

Where the *hat* $\hat{a}$ denotes that $\hat{a}$ is a dual number, which consists in a real part (a) and a dual part (b)

2.3.1.2 Dual Vectors.

Dual Vectors are a generalization of dual numbers where both real and dual parts are n-dimensional vectors. In this work, we will use three-dimensional vectors when referring to dual vectors.

Dot-Product for Dual Vectors Definition: Let $\hat{\bar{v}} = \bar{v}_r + \bar{v}_d\epsilon$ and $\hat{\bar{k}} = \bar{k}_r + \bar{k}_d\epsilon$ be dual vectors with $\bar{v}_r, \bar{v}_d, \bar{k}_r, \bar{k}_d \in \mathbb{R}^3$, then its dot product is given by

$$\hat{\bar{k}} \cdot \hat{\bar{v}} = K_r\bar{v}_r + K_d\bar{v}_d\epsilon \tag{2.35}$$

Where K_r and K_d are diagonal matrices with 3×3 shape with diagonal entries k_{r1}, k_{r2}, k_{r3} and k_{d1}, k_{d2}, k_{d3} respectively.

2.3.1.3 Dual Quaternions.

Dual quaternions are dual numbers with both real and dual parts given by quaternions, i.e. $\hat{q} = \boldsymbol{q}_r + \boldsymbol{q}_d\epsilon$, with $\boldsymbol{q}_r, \boldsymbol{q}_d \in \mathbb{H}$.

Some of the most important operations for dual quaternions are

Sum Let $\hat{q}_1$ and $\hat{q}_2$ be dual quaternions, then

$$\hat{q}_1 + \hat{q}_2 = \boldsymbol{q}_{1r} + \boldsymbol{q}_{2r} + [\boldsymbol{q}_{1d} + \boldsymbol{q}_{2d}]\,\epsilon \tag{2.36}$$

Product The multiplication between dual quaternions is defined as

$$\hat{q}_1 \otimes \hat{q}_2 = \boldsymbol{q}_{1r} \otimes \boldsymbol{q}_{2r} + [\boldsymbol{q}_{1r} \otimes \boldsymbol{q}_{2d} + \boldsymbol{q}_{1d} \otimes \boldsymbol{q}_{2r}]\,\epsilon \tag{2.37}$$

Norm The norm of a dual quaternion is defined as

$$||\hat{q}||^2 = \hat{q} \otimes \hat{q}^* \tag{2.38}$$

Note that if $||\hat{q}||^2 = 1 + 0\epsilon$, then $\hat{q}$ is called a *unit dual quaternion.*

Conjugation The conjugation of a dual quaternion is defined as

$$\hat{q}^* = \boldsymbol{q}_r^* + \boldsymbol{q}_d^*\epsilon \tag{2.39}$$

Since in this work we are dealing only with unit dual quaternions, then we can say that $\hat{q}^* = \hat{q}^{-1}$

Logarithmic Mapping The multiplication between dual quaternions is defined as

$$\ln \hat{q} = \frac{1}{2}(\bar{\theta} + T_B \epsilon), \tag{2.40}$$

where $\bar{\theta} = 2 \ln \boldsymbol{q}$ is given by a quaternion logarithmic mapping.

This operation is commonly used to transform between a dual quaternion and its axis-angle-translation equivalence Wang and Yu [2010].

Dual Quaternion Derivate When a dual quaternion is representing a simultaneous rotation and translation, it is defined as

$$\hat{q} \triangleq \boldsymbol{q} + \frac{\boldsymbol{q} \otimes T}{2} \epsilon \quad , \quad \boldsymbol{q} \in \mathbb{H}, T \in \mathbb{R}^3 \tag{2.41}$$

Where $\boldsymbol{q}$ represents the orientation of the body, and T represents its position with respect to the body frame.

The derivate of equation (2.41) given as

$$\dot{\hat{q}} = \dot{\boldsymbol{q}} + \frac{1}{2}\left[\dot{\boldsymbol{q}} \otimes T + \boldsymbol{q} \otimes \dot{T}\right] \epsilon \tag{2.42}$$

Applying the quaternion derivative from equation (2.16)

$$\dot{\hat{q}} = \frac{1}{2}\boldsymbol{q} \otimes \omega + \left[\frac{1}{4}\boldsymbol{q} \otimes \omega \otimes T + \frac{1}{2}\boldsymbol{q} \otimes \dot{T}\right] \epsilon \tag{2.43}$$

Applying the property from equation (A.1) we obtain that

$$\dot{\hat{q}} = \frac{1}{2}\boldsymbol{q} \otimes \omega + \left[\frac{1}{2}\boldsymbol{q} \otimes (\omega \times T) + \frac{1}{4}\boldsymbol{q} \otimes T \otimes \omega + \frac{1}{2}\boldsymbol{q} \otimes \dot{T}\right] \epsilon \tag{2.44}$$

Factorizing

$$\dot{\hat{q}} = \frac{1}{2}\left(\boldsymbol{q} + \frac{\boldsymbol{q} \otimes T}{2}\epsilon\right) \otimes \left(\omega + [\omega \times T_B + \dot{T}_B]\epsilon\right) \tag{2.45}$$

Let's define

$$\hat{\xi} \triangleq \omega + [\omega \times T_B + \dot{T}_B]\epsilon \tag{2.46}$$

where $\hat{\xi}$ is the *twist* dual vector (combination of angular and translational velocities).

Finally, we obtain the final expression for the derivate of a dual quaternion

$$\dot{\hat{q}} = \frac{1}{2}\hat{q} \otimes \hat{\xi} \tag{2.47}$$

2.3.2 Unit Dual Quaternions.

Let $\hat{q}$ be a dual quaternion given by

$$\hat{q} = \boldsymbol{q} + \frac{\boldsymbol{q} \otimes T_B}{2}\epsilon \tag{2.48}$$

where $\boldsymbol{q}$ is the unit quaternion that defines the attitude of the body, T_B is the translation vector with respect to the body frame (also a quaternion with real part equal to zero), and $\epsilon^2 = 0, \epsilon \neq 0$.

If $\hat{q}$ is a unit dual quaternion, then it can be used to describe simultaneously, the rotation and translation of the body with respect to the inertial frame.

Some advantages of using this representation, contrary to the separated rotational and translational sub-models, is that it is possible to describe various rotations and translations using only dual quaternion products, thus providing computational and mathematical simplicity to the model and the control algorithms.

2.3.2.1 Dual Quaternion Logarithmic Mapping

In section 2.3.1, it is shown that there exists a direct relationship between the "Axis-Angle-Translation" notation and a dual quaternion trough the Logarithmic mapping

$\ln \hat{q} = \frac{1}{2}(\bar{\theta} + T_B\epsilon)$ in equation (2.40).

This representation offers a more intuitive way to understand the transformation (simultaneous rotation and translation for a rigid body with respect of two reference frames).

2.3.3 Quadcopter Dual Quaternion Kinematics.

As done when modeling the vehicle with unit quaternions, it is considered that the quadrotor is a rigid body with 6 DoF and is under-actuated as shown in equations (2.22) and (2.29).

The state vector of the quadrotor can be expressed in terms of its orientation, position, and twist as

$$x := \begin{bmatrix} \hat{q} \\ \hat{\xi} \end{bmatrix} = \begin{bmatrix} \boldsymbol{q} + \frac{\boldsymbol{q} \otimes T}{2}\epsilon \\ \omega + [\omega \times T + \dot{T}]\epsilon \end{bmatrix} \tag{2.49}$$

where

- $T = \begin{bmatrix} x & y & z \end{bmatrix}^T$ is the translational position in the body frame.
- $\dot{T} = \begin{bmatrix} \dot{x} & \dot{y} & \dot{z} \end{bmatrix}^T$ denotes the translational velocity vector in the body frame.
- $\boldsymbol{q} = q_0 + \begin{bmatrix} q_1 & q_2 & q_3 \end{bmatrix}^T$ defines the vehicle orientation with respect to the inertial frame, represented as a unit quaternion.
- $\omega = \begin{bmatrix} \omega_x & \omega_y & \omega_z \end{bmatrix}^T$ represents the rotational velocity in the body frame.

2.3.4 Quadcopter Dual Quaternion Dynamic Model.

The dynamic model of a quadcopter using Newton-Euler approach with dual quaternions is described in Wang and Yu [2010] as follows

Remember that rotational dynamics of a rigid body is given by

$$J\dot{\omega} + \omega \times (J\omega) = \tau \tag{2.50}$$

According to Newton's equations of motion, the translational dynamic of a rigid body relative to its body frame is

$$m\ddot{T} = F \tag{2.51}$$

Differentiating equation (2.46), it can be obtained that

$$\dot{\hat{\xi}} = \dot{\omega} + [\dot{\omega} \times T_B + \omega \times \dot{T}_B + \ddot{T}_B]\epsilon \tag{2.52}$$

Substituting equations (2.50) and (2.51) in equation (2.52), the dynamic model is obtained

$$x = \begin{bmatrix} \hat{q} \\ \hat{\xi} \end{bmatrix}, \quad \dot{x} = \begin{bmatrix} \dot{\hat{q}} \\ \dot{\hat{\xi}} \end{bmatrix} = \begin{bmatrix} \frac{1}{2}\hat{q} \otimes \hat{\xi} \\ \hat{F} + \hat{u} \end{bmatrix} \tag{2.53}$$

where

$$\begin{cases} \hat{F} &= a + (a \times T_B + \omega \times \dot{T}_B)\epsilon \\ \hat{u} &= J^{-1}\tau + [J^{-1}\tau \times T_B + m^{-1}F]\epsilon \\ a &= -J^{-1}(\omega \times J\omega) \end{cases} \tag{2.54}$$

The relationships between the input torques and forces are

$$\begin{bmatrix} F_{th} \\ \tau_{u_x} \\ \tau_{u_y} \\ \tau_{u_z} \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^{4} k_i\,\omega_i^2 \\ l\,(k_1\,\omega_1^2 - k_2\,\omega_2^2 - k_3\,\omega_3^2 + k_4\,\omega_4^2) \\ l\,(k_1\,\omega_1^2 + k_2\,\omega_2^2 - k_3\,\omega_3^2 - k_4\,\omega_4^2) \\ \sum_{i=1}^{4} \tau_i\,(-1)^i \end{bmatrix} \tag{2.55}$$

where $k_i\,\omega_i^2$ defines the thrust of the propeller of motor i with respect to its angular velocity ω_i, l is the distance from the center of mass to the motor axis of action and τ_i denotes the torque of motor i.

Equilibrium Points. In order to find the equilibrium points, it is necessary to find the states and control inputs $x^*, \hat{u}^*$ where $\dot{x} = 0$.

Taking this into account, the equilibrium points are

$$\Rightarrow x^* = \begin{bmatrix} \hat{q}^* \\ \hat{0} \end{bmatrix} = \begin{bmatrix} \boldsymbol{q}^* + \frac{\boldsymbol{q}^* \otimes T^*}{2}\epsilon \\ \hat{0} \end{bmatrix}, \ \hat{u}^* = \hat{0} \begin{cases} \Rightarrow F = 0 \Rightarrow F_u = -F_{ext} \\ \Rightarrow \tau = 0 \Rightarrow \tau_u = -\tau_{ext} \end{cases} \tag{2.56}$$

This corroborates the intuitive approach for finding the equilibrium points; the platform will stay still at any position and orientation in space, as long as its angular velocity, and translational velocity equal to zero, and also as long as the control forces and torques counteract the external ones.

2.3.5 Quadrotor with Jointed Arm Model

Once both models were stated and tested, it was seen that dual quaternions offer more advantages when it comes to modeling for robotic arms, since their mathematical operations make it simpler when applying multiple rotations and translations.

The problem is stated as a robotic arm with three imaginary links, as shown in figure 2.2. The position and orientation of the final gripper can be obtained by combining rotations and translations from the inertial frame to the final actuator, all by using dual quaternions, being

- $\hat{q}_v$ the dual quaternion that describes transformation from the inertial frame to the quadrotor's reference frame.
- $\hat{q}_c$ the constant dual quaternion that describes transformation from the quadrotor's reference frame to the first articulation of the robotic arm.
- $\hat{q}_b$ the dual quaternion that describes transformation from the first articulation of the robotic arm to the final actuator.

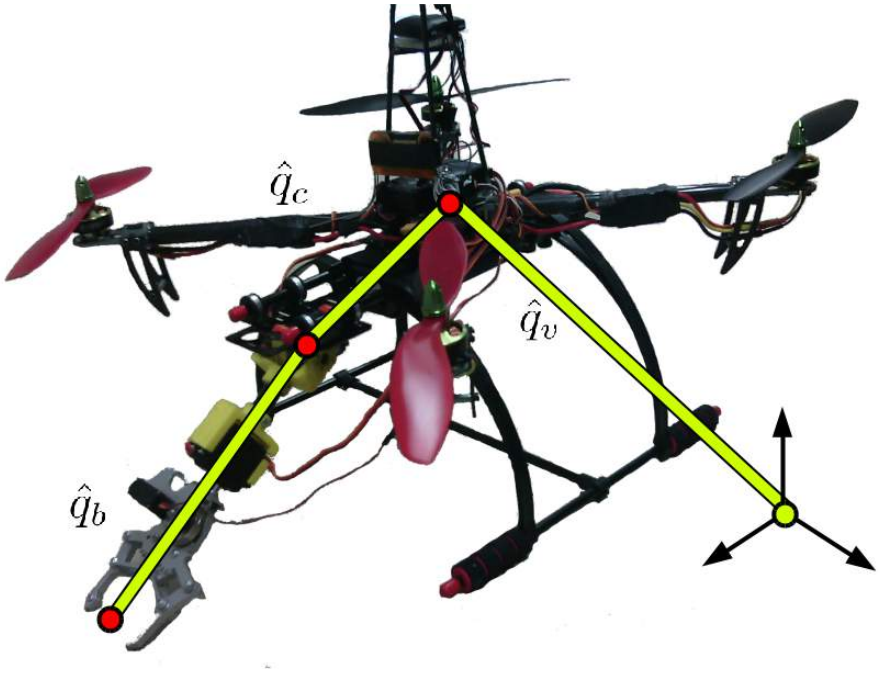

FIGURE 2.2: Complete platform seen as a robotic arm with seven DoF

The following assumptions are done

- The center of mass of the quadrotor is located in the geometrical origin of the vehicle's reference frame
- The link corresponding to $\hat{q}_c$ has no mass since it is considered inside of the quadrotor.
- The center of mass of the robotic arm is located in its ending (this is acceptable, since the gripper is made of metal, and the other pieces are made of plastic or carbon fiber)

2.3.5.1 Multiple Transformations

Consider three dual quaternions that describe three simultaneous rotations and translations

$$\hat{q}_1 = \boldsymbol{q}_1 + \frac{\boldsymbol{q}_1 \otimes T_{B_1}}{2}\epsilon \quad , \quad \hat{q}_2 = \boldsymbol{q}_2 + \frac{\boldsymbol{q}_2 \otimes T_{B_2}}{2}\epsilon \quad , \quad \hat{q}_3 = \boldsymbol{q}_3 + \frac{\boldsymbol{q}_3 \otimes T_{B_3}}{2}\epsilon$$

where $\hat{q}_1$ describes the simultaneous rotation and translation from the inertial frame to a first body frame, $\hat{q}_2$ describes the rotation and translation from the first body frame to a second body frame, and $\hat{q}_3$ from the second body frame to a third body frame.

The final transformation from the inertial frame to the third body frame (with respect to the third body's reference frame) is given by

$$\hat{q}_{total} = \hat{q}_1 \otimes \hat{q}_2 \otimes \hat{q}_3 \tag{2.57}$$

2.3.5.2 Complete Dynamic Model

Using the quadrotor dynamical model in equation (2.53) we define

$$x \triangleq \begin{bmatrix} \hat{q}_v \\ \hat{\xi}_v \\ \hat{q}_f \\ \hat{\xi}_f \end{bmatrix} \Rightarrow \dot{x} = \begin{bmatrix} \dot{\hat{q}}_v \\ \dot{\hat{\xi}}_v \\ \dot{\hat{q}}_f \\ \dot{\hat{\xi}}_f \end{bmatrix} = \begin{bmatrix} \frac{1}{2}\hat{q}_v \otimes \hat{\xi}_v \\ \hat{F}_v + \hat{u}_v \\ \frac{1}{2}\hat{q}_f \otimes \hat{\xi}_f \\ \hat{F}_f + \hat{u}_f \end{bmatrix} \tag{2.58}$$

where $\hat{q}_v$ describes the transformation of the quadrotor platform, $\hat{q}_f$ describes the transformation of the final actuator, and $\hat{\xi}_v$ and $\hat{\xi}_f$ are their respective twists, with

$$\hat{q}_f = \hat{q}_v \otimes \hat{q}_c \otimes \hat{q}_b \tag{2.59}$$

In order to obtain the detailed model, the Newton-Euler approach is used in each link.

Quadrotor Imaginary Link

For the first link, the resolution for $\dot{\hat{q}}_v$ is trivial, since equations (2.53) and (2.54) already describe it.

$$\begin{bmatrix} \dot{\hat{q}}_v \\ \dot{\hat{\xi}}_v \end{bmatrix} = \begin{bmatrix} \frac{1}{2}\hat{q}_v \otimes \hat{\xi}_v \\ \hat{F}_v + \hat{u}_v \end{bmatrix} \tag{2.60}$$

with

$$\hat{\xi}_v = \omega_v + [\omega_v \times T_v + \dot{T}_v]\epsilon$$

where ω_v is the angular velocity vector with respect of the vehicle's reference frame, T_v is its position, and.

$$\begin{cases} \hat{F}_v &= a_v + (a_v \times T_v + \omega_v \times \dot{T}_v)\epsilon \\ \hat{u}_v &= J_v^{-1}\tau_v + [J_v^{-1}\tau_v \times T_v + m_v^{-1}F_v]\epsilon \\ a_v &= -J_v^{-1}(\omega_v \times J_v\omega_v) \end{cases},$$

where J_v is the inertia matrix of the aerial platform, $\tau_v = \tau_{v-u} + \tau_{v-ext}$ is the sum of the torques that act on the vehicle, and $F_v = F_{v-u} + F_{v-ext}$ corresponds to the sum of the forces.

Robotic Arm's Link

In this subsection, for the sake of simplicity, we will consider a simpler notation for quaternion products, that means that $\hat{q}_a \otimes \hat{q}_b$ is equivalent to $\hat{q}_a\hat{q}_b$, and $\boldsymbol{q}_a \otimes \boldsymbol{q}_b$ is equivalent to $\boldsymbol{q}_a\boldsymbol{q}_b$.

For $\hat{q}_f$, we can obtain from equation (2.59) that

$$\hat{q}_f = \left(\boldsymbol{q}_v + \frac{1}{2}\boldsymbol{q}_v T_v \epsilon\right)\left(\boldsymbol{q}_0 + \frac{1}{2}\boldsymbol{q}_0 T_c \epsilon\right)\left(\boldsymbol{q}_b + \frac{\boldsymbol{q}_b \otimes T_b}{2}\epsilon\right), \tag{2.61}$$

Then expanding the product

$$\hat{q}_f = \boldsymbol{q}_v\boldsymbol{q}_b + \frac{1}{2}\left[\boldsymbol{q}_v\boldsymbol{q}_b T_b + \boldsymbol{q}_v(T_c + T_v)\boldsymbol{q}_b\right]\epsilon \tag{2.62}$$

The inverse of $\hat{q}_f$ will be needed later

$$\hat{q}_f^* = \boldsymbol{q}_b^*\boldsymbol{q}_v^* - \frac{1}{2}\left[T_b\boldsymbol{q}_b^*\boldsymbol{q}_v^* + \boldsymbol{q}_b^*(T_c + T_v)\boldsymbol{q}_v^*\right]\epsilon \tag{2.63}$$

Differentiating (2.62) the derivate of the first state is obtained

$$\begin{aligned}\dot{\hat{q}}_f &= \dot{\boldsymbol{q}}_v\boldsymbol{q}_b + \boldsymbol{q}_v\dot{\boldsymbol{q}}_b + \frac{1}{2}\dot{\boldsymbol{q}}_v\boldsymbol{q}_bT_b\epsilon + \frac{1}{2}\boldsymbol{q}_v\dot{\boldsymbol{q}}_bT_b\epsilon + \frac{1}{2}\boldsymbol{q}_v\boldsymbol{q}_b\dot{T}_b\epsilon + \frac{1}{2}\dot{\boldsymbol{q}}_v(T_c+T_v)\boldsymbol{q}_b\epsilon + \frac{1}{2}\boldsymbol{q}_v\dot{T}_v\boldsymbol{q}_b\epsilon \\ &\quad + \frac{1}{2}\boldsymbol{q}_v(T_c+T_v)\dot{\boldsymbol{q}}_b\epsilon \\ &= \dot{\boldsymbol{q}}_v\boldsymbol{q}_b + \boldsymbol{q}_v\dot{\boldsymbol{q}}_b + \frac{1}{2}\left[\dot{\boldsymbol{q}}_v\boldsymbol{q}_bT_b + \boldsymbol{q}_v\dot{\boldsymbol{q}}_bT_b + \boldsymbol{q}_v\boldsymbol{q}_b\dot{T}_b + \dot{\boldsymbol{q}}_v(T_c+T_v)\boldsymbol{q}_b + \boldsymbol{q}_v\dot{T}_v\boldsymbol{q}_b + \boldsymbol{q}_v(T_c+T_v)\dot{\boldsymbol{q}}_b\right]\epsilon\end{aligned} \tag{2.64}$$

We know that the differentiate of any dual quaternion that expresses a transformation is given by equation (2.47), thus the derivate of (2.62) is given by

$$\dot{\hat{q}}_f = \frac{1}{2}\hat{q}_f\hat{\xi}_f \Rightarrow \hat{\xi}_f = 2\hat{q}_f^*\dot{\hat{q}}_f \tag{2.65}$$

Substituting (2.62) and (2.64) to (2.65) a larger expression for $\hat{\xi}_f$ is obtained

$$\begin{aligned}\hat{\xi}_f &= 2\left(\boldsymbol{q}_b^*\boldsymbol{q}_v^* - \tfrac{1}{2}\left[T_b\boldsymbol{q}_b^*\boldsymbol{q}_v^* + \boldsymbol{q}_b^*(T_c+T_v)\boldsymbol{q}_v^*\right]\epsilon\right)\left(\dot{\boldsymbol{q}}_v\boldsymbol{q}_b + \boldsymbol{q}_v\dot{\boldsymbol{q}}_b + \tfrac{1}{2}\left[\dot{\boldsymbol{q}}_v\boldsymbol{q}_bT_b\right.\right. \\ &\quad \left.\left. + \boldsymbol{q}_v\dot{\boldsymbol{q}}_bT_b + \boldsymbol{q}_v\boldsymbol{q}_b\dot{T}_b + \dot{\boldsymbol{q}}_v(T_c+T_v)\boldsymbol{q}_b + \boldsymbol{q}_v\dot{T}_v\boldsymbol{q}_b + \boldsymbol{q}_v(T_c+T_v)\dot{\boldsymbol{q}}_b\right]\epsilon\right) \\ &= 2\boldsymbol{q}_b^*\boldsymbol{q}_v^*(\dot{\boldsymbol{q}}_v\boldsymbol{q}_b + \boldsymbol{q}_v\dot{\boldsymbol{q}}_b) - \left[T_b\boldsymbol{q}_b^*\boldsymbol{q}_v^* + \boldsymbol{q}_b^*(T_c+T_v)\boldsymbol{q}_v^*\right]\dot{\boldsymbol{q}}_v\boldsymbol{q}_b\epsilon - \left[T_b\boldsymbol{q}_b^*\right. \\ &\quad \left. + \boldsymbol{q}_b^*(T_c+T_v)\right]\dot{\boldsymbol{q}}_b\epsilon + \left[\boldsymbol{q}_b^*\boldsymbol{q}_v^*\dot{\boldsymbol{q}}_v\boldsymbol{q}_bT_b + \boldsymbol{q}_b^*\dot{\boldsymbol{q}}_bT_b + \dot{T}_b + \boldsymbol{q}_b^*\boldsymbol{q}_v^*\dot{\boldsymbol{q}}_v(T_c+T_v)\boldsymbol{q}_b\right. \\ &\quad \left. + \boldsymbol{q}_b^*\dot{T}_v\boldsymbol{q}_b + \boldsymbol{q}_b^*(T_c+T_v)\dot{\boldsymbol{q}}_b\right]\epsilon\end{aligned} \tag{2.66}$$

Applying now the derivate of a unit quaternion from equation (2.16)

$$\begin{aligned}\hat{\xi}_f = \boldsymbol{q}_b^*\omega_v\boldsymbol{q}_b + \omega_b + \frac{1}{2}[&-T_b\times\boldsymbol{q}_b^*\omega_v\boldsymbol{q}_b - \boldsymbol{q}_b^*(T_c+T_v)\times\omega_v\boldsymbol{q}_b - T_b\times\omega_b \\ &+ \boldsymbol{q}_b^*\omega_v\boldsymbol{q}_b\times T_b + \omega_b\times T_b + \dot{T}_b + \boldsymbol{q}_b^*\omega_v\times(T_c+T_v)\boldsymbol{q}_b + 2\boldsymbol{q}_b^*\dot{T}_v\boldsymbol{q}_b]\epsilon\end{aligned}$$

Applying the properties from equations (A.6) and (A.1)

$$\hat{\xi}_f = \boldsymbol{q}_b^*\omega_v\boldsymbol{q}_b + \omega_b + \left[\omega_b\times T_b + \boldsymbol{q}_b^*\omega_v\boldsymbol{q}_b\times T_b + \boldsymbol{q}_b^*\omega_v\times(T_c+T_v)\boldsymbol{q}_b + \boldsymbol{q}_b^*\dot{T}_v\boldsymbol{q}_b + \frac{1}{2}\dot{T}_b\right]\epsilon$$

In our prototype, the length of the robotic arm T_b is constant, thus the vector $\dot{T}_b$ is zero

$$\hat{\xi}_f = \boldsymbol{q}_b^* \omega_v \boldsymbol{q}_b + \omega_b + \left[\omega_b \times T_b + \boldsymbol{q}_b^* \omega_v \boldsymbol{q}_b \times T_b + \boldsymbol{q}_b^* \omega_v \times (T_c + T_v)\boldsymbol{q}_b + \boldsymbol{q}_b^* \dot{T}_v \boldsymbol{q}_b\right] \epsilon \tag{2.67}$$

Differentiating now equation (2.67)

$$\begin{aligned}\dot{\hat{\xi}}_f = {} & \dot{\boldsymbol{q}}_b^* \omega_v \boldsymbol{q}_b + \boldsymbol{q}_b^* \dot{\omega}_v \boldsymbol{q}_b + \boldsymbol{q}_b^* \omega_v \dot{\boldsymbol{q}}_b + \dot{\omega}_b + [\dot{\omega}_b \times T_b + \dot{\boldsymbol{q}}_b^* \omega_v \boldsymbol{q}_b \times T_b \\ & + \boldsymbol{q}_b^* \dot{\omega}_v \boldsymbol{q}_b \times T_b + \boldsymbol{q}_b^* \omega_v \dot{\boldsymbol{q}}_b \times T_b + \dot{\boldsymbol{q}}_b^* \omega_v \times (T_c + T_v)\boldsymbol{q}_b + \boldsymbol{q}_b^* \dot{\omega}_v \times (T_c + T_v)\boldsymbol{q}_b \\ & + \boldsymbol{q}_b^* \omega_v \times \dot{T}_v \boldsymbol{q}_b + \boldsymbol{q}_b^* \omega_v \times (T_c + T_v)\dot{\boldsymbol{q}}_b + \dot{\boldsymbol{q}}_b^* \dot{T}_v \boldsymbol{q}_b + \boldsymbol{q}_b^* \ddot{T}_v \boldsymbol{q}_b + \boldsymbol{q}_b^* \dot{T}_v \dot{\boldsymbol{q}}_b\Big] \epsilon\end{aligned} \tag{2.68}$$

The expressions for the derivate of the orientation of the arm and the vehicle, and their derivates are $\dot{\boldsymbol{q}}_b = \frac{1}{2}\boldsymbol{q}_b \omega_b$, $\dot{\boldsymbol{q}}_b^* = -\frac{1}{2}\omega_b \boldsymbol{q}_b^*$, $\dot{\boldsymbol{q}}_v = \frac{1}{2}\boldsymbol{q}_v \omega_v$, and $\dot{\boldsymbol{q}}_v^* = -\frac{1}{2}\omega_v \boldsymbol{q}_v^*$, thus

$$\begin{aligned}\dot{\hat{\xi}}_f = {} & -\tfrac{1}{2}\omega_b \times \boldsymbol{q}_b^* \omega_v \boldsymbol{q}_b + \boldsymbol{q}_b^* \dot{\omega}_v \boldsymbol{q}_b + \tfrac{1}{2}\boldsymbol{q}_b^* \omega_v \boldsymbol{q}_b \times \omega_b + \dot{\omega}_b \\ & + \big[\dot{\omega}_b \times T_b - \tfrac{1}{2}\omega_b \times \boldsymbol{q}_b^* \omega_v \boldsymbol{q}_b \times T_b + \boldsymbol{q}_b^* \dot{\omega}_v \boldsymbol{q}_b \times T_b + \tfrac{1}{2}\boldsymbol{q}_b^* \omega_v \boldsymbol{q}_b \times \omega_b \times T_b \\ & - \tfrac{1}{2}\omega_b \times \boldsymbol{q}_b^* \omega_v \times (T_c + T_v)\boldsymbol{q}_b + \boldsymbol{q}_b^* \dot{\omega}_v \times (T_c + T_v)\boldsymbol{q}_b + \boldsymbol{q}_b^* \omega_v \times \dot{T}_v \boldsymbol{q}_b \\ & + \tfrac{1}{2}\boldsymbol{q}_b^* \omega_v \times (T_c + T_v)\boldsymbol{q}_b \omega_b - \tfrac{1}{2}\omega_b \times \boldsymbol{q}_b^* \dot{T}_v \boldsymbol{q}_b + \boldsymbol{q}_b^* \ddot{T}_v \boldsymbol{q}_b + \tfrac{1}{2}\boldsymbol{q}_b^* \dot{T}_v \boldsymbol{q}_b \times \omega_b\Big] \epsilon\end{aligned} \tag{2.69}$$

Now applying property from equation (A.6)

$$\begin{aligned}\dot{\hat{\xi}}_f = {} & \boldsymbol{q}_b^* \omega_v \boldsymbol{q}_b \times \omega_b + \boldsymbol{q}_b^* \dot{\omega}_v \boldsymbol{q}_b + \dot{\omega}_b \\ & + \left[\dot{\omega}_b \times T_b + \boldsymbol{q}_b^* \omega_v \boldsymbol{q}_b \times \omega_b \times T_b + \boldsymbol{q}_b^* \left(\dot{\omega}_v \times (T_c + T_v) + \omega_v \times \dot{T}_v\right) \boldsymbol{q}_b \right. \\ & \left. + \boldsymbol{q}_b^* \omega_v \times (T_c + T_v)\boldsymbol{q}_b \times \omega_b + \boldsymbol{q}_b^* \dot{T}_v \boldsymbol{q}_b \times \omega_b + \boldsymbol{q}_b^* \ddot{T}_v \boldsymbol{q}_b + \boldsymbol{q}_b^* \dot{\omega}_v \boldsymbol{q}_b \times T_b\right] \epsilon\end{aligned} \tag{2.70}$$

Separating now the components for the angular acceleration of the final actuator from equation (2.70), the following expression is defined

$$\dot{\omega}_f \triangleq \boldsymbol{q}_b^* \omega_v \boldsymbol{q}_b \times \omega_b + \boldsymbol{q}_b^* \dot{\omega}_v \boldsymbol{q}_b + \dot{\omega}_b \tag{2.71}$$

Then, equation (2.70) is reduced to

$$\dot{\hat{\xi}}_f = \dot{\omega}_f + \Big[\dot{\omega}_f \times T_b + \boldsymbol{q}_b^* \left(\omega_v \times (T_c + T_v) + \dot{T}_v\right) \boldsymbol{q}_b \times \omega_b \\ + \boldsymbol{q}_b^* \left(\ddot{T}_v + \dot{\omega}_v \times (T_c + T_v) + \omega_v \times \dot{T}_v\right) \boldsymbol{q}_b\Big] \epsilon \quad (2.72)$$

Now, the components of the translational acceleration are separated, thus defining

$$\ddot{T}_f \triangleq \boldsymbol{q}_b^* \left(\ddot{T}_v + \dot{\omega}_v \times (T_c + T_v) + \omega_v \times \dot{T}_v\right) \boldsymbol{q}_b \quad (2.73)$$

Finally, the final expression for the derivate of the second state is given by

$$\dot{\hat{\xi}}_f = \dot{\omega}_f + \left[\dot{\omega}_f \times T_b + \boldsymbol{q}_b^* \left(\omega_v \times (T_c + T_v) + \dot{T}_v\right) \boldsymbol{q}_b \times \omega_b + \ddot{T}_f\right] \epsilon \quad (2.74)$$

Chapter 3

Quaternion Control

After the two versions of the model were presented, two control laws are now proposed to stabilize the system in a desired position, two sections compose this chapter, the first one is an exact linearization that stabilizes the orientation of the vehicle, then a trajectory is calculated to stabilize its position, the second one is a feedback regulator that simultaneously stabilizes the vehicle in position and orientation using dual quaternions, a trajectory is also calculated to stabilize the position by modifying the orientation.

3.1 Unit Quaternion Control

For the first control approach a special feedback is stated such that the non-linear model of the quadrotor becomes linear, in this section the robotic arm is considered as an external torque such that it can be compensated.

3.1.1 Attitude algorithm

From and using the logarithmic mapping from (2.20), the rotational dynamic model of the quadrotor from (2.25) can be expressed as

$$x_{att} = \begin{bmatrix} \bar{\theta} \\ \omega \end{bmatrix} \Rightarrow \dot{x}_{att} = \begin{bmatrix} \omega \\ J^{-1}\,(\tau - \omega \times J\,\omega) \end{bmatrix} \tag{3.1}$$

The objective is to force the system to have a linear behavior, thus we propose a virtual control such that this can be accomplished

$$\tau = J\,u + \omega \times J\,\omega \tag{3.2}$$

being u the new control input, then (3.1) yields

$$\dot{x}_{att} = Ax_{att} + Bu \tag{3.3}$$

with $A = \begin{bmatrix} \overline{0} & I_3 \\ \overline{0} & \overline{0} \end{bmatrix}$, and $B = \begin{bmatrix} \overline{0} \\ I_3 \end{bmatrix}$.

Remember that $\tau = \tau_u + \tau_{ext}$, so the torques that the motors have to apply to the vehicle are

$$\tau_u = J\,u + \omega \times J\,\omega - \tau_{ext} \tag{3.4}$$

3.1.1.1 State Feedback Control

Proposing the following control law

$$u = -K_{att}x_{att} \tag{3.5}$$

where $K_{att} > 0$ denotes the feedback gains, thus equation (3.3) becomes a closed-loop system

$$\dot{x}_{att} = Ax_{att} - BK_{att}x_{att} \tag{3.6}$$

Therefore, if the gain matrix K_{att} is selected such that the eigenvalues of $A - BK_{att}$ have all negative real parts, then system (2.25) (and its equivalence (3.1)) becomes exponentially stable, and converges asymptotically to the origin.

3.1.1.2 Error Stabilization

The quaternion error is defined as follows

$$\boldsymbol{q}_e = \boldsymbol{q} \otimes \boldsymbol{q}_d^* \tag{3.7}$$

Note that if $\boldsymbol{q} \to \boldsymbol{q}_d$, then $\boldsymbol{q} \simeq \boldsymbol{q}_d \otimes \boldsymbol{q}_d^* = \boldsymbol{q}_0$, where $\boldsymbol{q}_0 = 1 + [0 \;\; 0 \;\; 0]^T$.

Hence, if the control law is implemented using the quaternion error as

$$u = -K_{att} x_{att-e} \tag{3.8}$$

where

$$x_{att-e} = \begin{bmatrix} (2 \ln \boldsymbol{q}_d)^T & \omega_e^T \end{bmatrix}^T, \;\; \omega_e = \frac{d}{dt} 2 \ln \boldsymbol{q}_d \tag{3.9}$$

then, the orientation will converge asymptotically to the desired reference

$$x_{att-e} \to \bar{0} \;\; \Rightarrow \;\; \boldsymbol{q}_e \to \boldsymbol{q}_0 \;\; \Rightarrow \;\; \boldsymbol{q} \to \boldsymbol{q}_d$$

3.1.1.3 Robotic Arm Effects

From equation (2.33) the torques that the robotic arm produce on the platform can be calculated. and incorporated in equation (3.4) as

$$\tau_u = J\,u + \omega \times J\,\omega - (\boldsymbol{q}_b \otimes r_b \otimes \boldsymbol{q}_b^* + r_c) \times m_b \vec{g}_v - J_b(\omega_v + \omega_b) \tag{3.10}$$

3.1.2 Position Strategy

Rewriting equation (2.28), we have that

$$\dot{x}_{pos} = \begin{bmatrix} \dot{p} \\ \ddot{p} \end{bmatrix} = \begin{bmatrix} \dot{p} \\ \boldsymbol{q} \otimes \dfrac{F_u}{m} \otimes \boldsymbol{q}^* + \overline{g} \end{bmatrix} \tag{3.11}$$

3.1.2.1 Feedback Linearization

Define

$$u_p = \boldsymbol{q} \otimes \frac{F_u}{m} \otimes \boldsymbol{q}^* \tag{3.12}$$

The following exact linearization is proposed

$$u_p = u_{pos} - \overline{g} \tag{3.13}$$

with u_{pos} denoting the new control input.

Then rewriting (2.28) we have

$$\begin{aligned} \dot{x}_{pos} &= \begin{bmatrix} \overline{0} & I_3 \\ \overline{0} & \overline{0} \end{bmatrix} x_{pos} + \begin{bmatrix} \overline{0} \\ I_3 \end{bmatrix} u_{pos} \\ &= A_{pos}\, x_{pos} + B_{pos}\, u_{pos} \end{aligned} \tag{3.14}$$

3.1.2.2 State Feedback Control

Proposing the following control law that uses the position error to stabilize the system in a desired reference

$$u_{pos} = -K_{pos}\,(x_{pos} - x_{pos_d}) \tag{3.15}$$

with x_{pos_d} as the desired position state, then this controller exponentially stabilizes subsystem (3.14) if the gain matrix K_{pos} is selected such that the eigenvalues of

$A_{pos} - B_{pos}K_{pos}$ have all negative real parts.

3.1.3 Attitude Trajectory

The quadrotor is an under-actuated system, and the only force that can be applied to stabilize the vehicle is F_u, which can only act in the z-axis of the body frame, according with equation (2.29).

Notice that the quaternion norm is $||\boldsymbol{q}|| = 1$ thus, the direction of the input vector $u_p = \boldsymbol{q} \otimes \frac{F_u}{m} \otimes \boldsymbol{q}^*$ depends exclusively on the vehicle's orientation, and its magnitude depends entirely of the magnitude of the vector $\frac{F_u}{m}$, which is also a control input of the quadcopter.

Notice that the orientation of the vehicle is completely actuated, then the position of the quadrotor with respect to the inertial frame can be controlled by defining a trajectory for its orientation.

If u_p is the force in the inertial frame that stabilizes the system (2.28) at the desired position, then there exists a rotation $\boldsymbol{q}_d$ that makes the vector F_u coincide with u_p in the inertial frame.

The algebraic relationship between $F_u, \boldsymbol{q}_d$, and u_p is

$$\begin{aligned} \boldsymbol{q}'_d &= (b \cdot u_p + ||u_p||) + b \times u_p \\ \boldsymbol{q}_d &= \frac{\boldsymbol{q}'_d}{||\boldsymbol{q}'_d||} \\ F_u &= b\,||u_p||\,m \end{aligned} \tag{3.16}$$

where $b = [0 \;\; 0 \;\; 1]^T$ is a unitary vector denoting the axis in which the thrust acts in the body frame.

The angular velocity trajectory can be obtained by differentiating the desired orientation

$$\omega_d = \frac{d}{dt}(2 \ln \boldsymbol{q}_d) \tag{3.17}$$

Therefore, the quaternion error is calculated using (3.16), following equation (3.7), and then implemented in the attitude algorithm such that $\boldsymbol{q} \approxeq \boldsymbol{q}_d$, which implies that $u_p \approxeq \boldsymbol{q} \otimes \dfrac{F_u}{m} \otimes \boldsymbol{q}^*$ and then, the model (2.28) is stabilized alongside with the model (2.25).

3.2 Dual Quaternion Control

Unit quaternions provide a simple way of stabilizing the platform, but when the references of the quadrotor depend on the orientation and position of the robotic arm, the calculations become more complex.

It was shown in section 2.3 that dual quaternions are a simpler way to deal with multiple rotations and translations, thus a control scheme that stabilizes the system using dual quaternions is needed.

The effects of the robotic arm are included such that the transformation of the quadrotor vehicle depends on the references of the final actuator. The dynamic effects are included as external torques and forces.

3.2.1 Proposed Control Law

A control law is proposed such that the sub-system $\dot{\hat{\xi}} = \hat{F} + \hat{u}$ from equation (2.53) is exactly linearized

$$\hat{u} = 2\hat{k}_p \cdot \ln \hat{q} + \hat{k}_v \cdot \hat{\xi} - \hat{F} \tag{3.18}$$

where $\hat{k}_p = \bar{k}_{pr} + \bar{k}_{pd}\epsilon$ and $\hat{k}_v = \bar{k}_{vr} + \bar{k}_{vd}\epsilon$ are dual vectors that contain the control gains with $\bar{k}_{pr}, \bar{k}_{pd}, \bar{k}_{vr}, \bar{k}_{vd} \in \mathbb{R}^3$.

Theorem 1: The control law (3.18) in (2.53) assures that the dual quaternion $\hat{q}$ converges asymptotically to the identity dual quaternion $\hat{q}_0 = 1 + [0,0,0]^T + \left[0 + [0,0,0]^T\right]\epsilon$, using appropriate $\hat{k}_p$ and $\hat{k}_v$.

Proof: Substituting (3.18) in (2.53)

$$\begin{bmatrix} \dot{\hat{q}} \\ \dot{\hat{\xi}} \end{bmatrix} = \begin{bmatrix} \frac{1}{2}\hat{q} \otimes \hat{\xi} \\ 2\hat{k}_p \cdot \ln \hat{q} + \hat{k}_v \cdot \xi \end{bmatrix}$$

rewriting the $\dot{\hat{\xi}}$ dynamics

$$\dot{\hat{\xi}} = 2\hat{k}_p \cdot \ln \hat{q} + \hat{k}_v \cdot \hat{\xi} \tag{3.19}$$

Applying the definition of the logarythmic mapping and the twist to (3.19), we obtain

$$\dot{\hat{\xi}} = \hat{k}_p \cdot (\bar{\theta} + T\epsilon) + \hat{k}_v \cdot (\omega + [\omega \times T + \dot{T}]\epsilon) \tag{3.20}$$

If we compare (3.20) with (2.52), it can be seen (using the dual vector dot-product) that the rotational acceleration is given by

$$\dot{\omega} = K_{pr}\bar{\theta} + K_{vr}\omega \tag{3.21}$$

and the translational acceleration by

$$\ddot{T} = K_{pd}T + K_{vd}(\omega \times T + \dot{T}) - (\dot{\omega} \times T + \omega \times \dot{T}), \tag{3.22}$$

where $K_{pr}, K_{pd}, K_{vr}, K_{vd} \in \mathbb{R}^{3\times 3}$ are the diagonal matrices of $\bar{k}_{pr}, \bar{k}_{pd}, \bar{k}_{vr}$ and $\bar{k}_{vd}$ respectively.

Taking in account that $\omega = \dot{\bar{\theta}}$, then we obtain the following equation for the rotational acceleration

$$\ddot{\bar{\theta}} - K_{vr}\dot{\bar{\theta}} - K_{pr}\bar{\theta} = 0 \tag{3.23}$$

Using Laplace transformation, we can obtain the following expression

$$\bar{\Theta}_1 s^2 - K_{vr}\bar{\Theta}s - K_{pr}\bar{\Theta} = 0$$

$$s^2 - K_{vr}s - K_{pr} = 0 \tag{3.24}$$

If $\bar{k}_{vr}$ and $\bar{k}_{pr}$ are selected such that the roots of (3.24) are defined negative, then $\bar{\theta}_1$ as well as ω_1, and $\dot{\omega}_1$ will converge to zero asymptotically.

We can now gather all the forces that are originated from the rotational velocity and acceleration defining

$$F(\omega, \dot{\omega}) = -K_{vd}(\omega \times T) + \dot{\omega} \times T + \omega \times \dot{T} \tag{3.25}$$

It was already proved that $\bar{\theta}$, ω, and $\dot{\omega}$ converge asymptotically to zero, then this implies that when $t \to \infty$, then $F(\omega_1, \dot{\omega}_1) \to 0$.

Then, there is a finite time such that equation (3.25) will be so small that can be neglected, and equation (3.22) could be reduced to

$$\ddot{T} = K_{pd}T + K_{vd}\dot{T} \tag{3.26}$$

Using Laplace transform, we obtain the following polynomial

$$\mathbf{T}s^2 - K_{vd}\mathbf{T}s - K_{pd}\mathbf{T} = 0$$

$$s^2 - K_{vd}s - K_{pd} = 0 \tag{3.27}$$

If $\bar{k}_{vd}$ and $\bar{k}_{pd}$ are elected such that the roots of (3.27) are defined negative, then T as well as $\dot{T}$, and $\ddot{T}$ converge to zero asymptotically. ■

3.2.2 Application in a Quadrotor Platform

It is known that quadrotors are under-actuated systems, although the input torques can act over the three axises (x, y, z), the input forces are considered to only act in the z axis, see equation (2.55), for this reason, the control law proposed in (3.18) can not be implemented directly in our vehicle.

It is important to design a dual quaternion based control that can be implemented in our quadrotor platform. A trajectory for the attitude must be calculated using a relationship between a desired force in the inertial frame and a reference in orientation.

Substituting the description of the twist from (2.46) in (3.18)

$$\begin{aligned} \hat{u} &= 2\hat{k}_p \cdot \ln \hat{q} + \hat{k}_v \cdot (\omega + [\omega \times T + \dot{T}]\epsilon) - \hat{F} \\ &= J^{-1}\tau + [J^{-1}\tau \times T + m^{-1}F]\epsilon \end{aligned} \tag{3.28}$$

Then

$$\hat{k}_p \cdot (\bar{\theta} + T\epsilon) + \hat{k}_v \cdot (\omega + [\omega \times T + \dot{T}]\epsilon) - a - (a \times T + \omega \times \dot{T})\epsilon$$
$$= J^{-1}\tau + [J_1^{-1}\tau \times T + m^{-1}F1]\epsilon$$

$$K_{pr}\bar{\theta} + K_{pd}T\epsilon + K_{vr}\omega + K_{vd}[\omega \times T + \dot{T}]\epsilon - a - (a \times T + \omega_1 \times \dot{T})\epsilon$$
$$= J^{-1}\tau + [J^{-1}\tau \times T + m^{-1}F]\epsilon \tag{3.29}$$

Separating the real and dual parts from (3.29) we obtain the expressions for the input torques and forces

$$\begin{aligned} \tau &= \tau_u + \tau_{ext} = J(K_{pr}\bar{\theta} + K_{vr}\omega - a) \\ \tau_u &= J(K_{pr}\bar{\theta} + K_{vr}\omega - a) - \tau_{ext} \end{aligned} \tag{3.30}$$

remember that τ_{ext} can be obtained from equation (2.33).

$$\begin{aligned} F &= F_u + F_{ext} \\ &= m\left(K_{pd}T + K_{vd}[\omega \times T + \dot{T}] - (a_1 \times T + \omega_1 \times \dot{T}) - J_1^{-1}\tau_1 \times T\right) \\ F_u &= m\left(K_{pd}T + K_{vd}[\omega \times T + \dot{T}] - (a_1 \times T + \omega_1 \times \dot{T}) - J_1^{-1}\tau_1 \times T\right) - F_{ext} \end{aligned} \tag{3.31}$$

Equation (3.30) determines the torques that are applied as an input by the control law. As it was shown, there is a finite time when the angular velocity and acceleration can be neglected because when $t \to \infty$, then $\bar{\theta} \to 0$, $\omega \to 0$, $\dot{\omega} \to 0$, and $\tau_1 \to 0$ y

$a \to 0$. Then equation (3.31) can be reduced to

$$F_u = m\left(K_{pd}T + K_{vd}[\omega \times T + \dot{T}] - (a_1 \times T + \omega_1 \times \dot{T})\right) - F_{ext} \tag{3.32}$$

In the case of a quadrotor platform, it can be considered that F_{ext} is the gravity force in the body reference given by $F_{ext} = m\boldsymbol{q}^* \otimes \bar{g} \otimes \boldsymbol{q}$, thus

$$F_u = m\left(K_{pd}T + K_{vd}[\omega \times T + \dot{T}] - (a_1 \times T + \omega_1 \times \dot{T})\right) - m\boldsymbol{q}^* \otimes \bar{g} \otimes \boldsymbol{q} \tag{3.33}$$

It is important to note that the desired control force that is necessary to move the body to the desired reference is given in equation (3.33) and is referenced to the body frame. This force can be expressed in the inertial frame by defining

$$F_{uI} \triangleq \boldsymbol{q} \otimes F_u \otimes \boldsymbol{q}^* \tag{3.34}$$

3.2.2.1 Error Stabilization

The quaternion error is defined as follows

$$\hat{q}_e = \hat{q} \otimes \hat{q}_d^* = \boldsymbol{q}_e + \frac{\boldsymbol{q}_e \otimes T_e}{2}\epsilon \tag{3.35}$$

where

$$\hat{q} = \boldsymbol{q} + \frac{\boldsymbol{q} \otimes T}{2}\epsilon \ , \ \hat{q}_d = \boldsymbol{q}_d + \frac{\boldsymbol{q}_d \otimes T_d}{2}\epsilon$$

being q_d the desired attitude, and T_d the desired position in the body frame, then

$$\begin{aligned} \hat{q}_e &= \left(\boldsymbol{q} + \tfrac{\boldsymbol{q}\otimes T}{2}\epsilon\right) \otimes \left(\boldsymbol{q}_d + \tfrac{\boldsymbol{q}_d \otimes T_d}{2}\epsilon\right)^* \\ &= \left(\boldsymbol{q} + \tfrac{\boldsymbol{q}\otimes T}{2}\epsilon\right) \otimes \left(\boldsymbol{q}_d^* - \tfrac{T_d \otimes \boldsymbol{q}_d^*}{2}\epsilon\right) \\ &= \left(\boldsymbol{q} \otimes \boldsymbol{q}_d^* + \tfrac{\boldsymbol{q}\otimes(T - T_d)\otimes \boldsymbol{q}_d^*}{2}\epsilon\right) \ \Rightarrow \ T_e = \boldsymbol{q}_d \otimes (T - T_d) \otimes \boldsymbol{q}_d^* \end{aligned} \tag{3.36}$$

Note that if $\hat{q} \to \hat{q}_d$, then $\boldsymbol{q} \to \boldsymbol{q}_d$, and $T \to T_d$, thus $\hat{q} \simeq \hat{q}_d \otimes \hat{q}_d^* = \hat{q}_0$, where $\hat{q}_0 = 1 + [0 \ 0 \ 0]^T + \left[0 + [0 \ 0 \ 0]^T\right]\epsilon$. Therefore, if the control law from equation

(3.18) is implemented using the dual quaternion error as

$$\hat{u} = 2\hat{k}_p \cdot \ln \hat{q}_e + \hat{k}_v \cdot \hat{\xi}_e - \hat{F} \tag{3.37}$$

with

$$\hat{\xi}_e = \omega_e + [\omega_e \times T_e + \dot{T}_e]\epsilon$$

Then the error will converge to $\hat{q}_0$, thus the transformation will converge asymptotically to the desired reference $\hat{q}_d$.

3.2.3 Trajectory Calculation

We have already discussed that the quadrotor is an under-actuated system, the trajectory from equation (3.16) must be calculated to fully stabilize the system.

If F_{uI} is the force in the inertial frame that stabilizes the system (2.28) at the desired position, then there exists a trajectory $\boldsymbol{q}_d$ that makes the vector F_{th} from equation (2.55) coincide with u_p in the inertial frame.

Following the algebraic relationship from equation (3.16), we obtain

$$\begin{aligned} \boldsymbol{q}'_d &= (b \cdot F_{uI} + ||F_{uI}||) + b \times F_{uI} \\ \boldsymbol{q}_d &= \frac{\boldsymbol{q}'_d}{||\boldsymbol{q}'_d||} \\ F_{th} &= b\,||F_{uI}|| \end{aligned} \tag{3.38}$$

and in consequence

$$\omega_d = \frac{d}{dt}(2 \ln \boldsymbol{q}_d) \tag{3.39}$$

Therefore, the quaternion error is calculated using (3.38), following equation (3.35), and then implemented in the control law from (3.18) such that $\hat{q} \cong \hat{q}_d$, which implies that $F_{uI} \cong \boldsymbol{q} \otimes \frac{F_{th}}{m} \otimes \boldsymbol{q}^*$, thus the model (2.53) converges asymptotically to the desired position T_d with orientation $\boldsymbol{q} = \boldsymbol{q}_0$.

3.2.4 Definition of the References

One of the objectives of this work is to obtain a control scheme where the final effector of the robotic arm stabilizes in a desired position and orientation, this transformation can be given as a dual quaternion

$$\hat{q}_{fd} = \boldsymbol{q}_{fd} + \frac{\boldsymbol{q}_{fd} \otimes T_{fd}}{2}\epsilon \tag{3.40}$$

Where the position is given in the body reference frame as $T_{fd} = 0{+}[p_{fd-x}, p_{fd-y}, p_{fd-z}]^T$ and $\boldsymbol{q}_{fd} = \boldsymbol{q}_{xy} \otimes \boldsymbol{q}_z$ is the desired orientation of the gripper.

$\boldsymbol{q}_{xy}$ represents the desired orientation in the x and y axises, and $\boldsymbol{q}_z$ gives the orientation around the z axis.

Note that $\boldsymbol{q}_{xy}$ depends only on the inclination of the robotic arm because the quadrotor needs to be stabilized in its equilibrium point with respect to the x and y axes, thus

$$\boldsymbol{q}_{xy} = \begin{bmatrix} \boldsymbol{q}_{xy_0} \\ \boldsymbol{q}_{xy_1} \\ \boldsymbol{q}_{xy_2} \\ \boldsymbol{q}_{xy_3} \end{bmatrix} = \begin{bmatrix} \cos(||\theta_{xy}||/2) \\ \bar{e}_0 \sin(||\theta_{xy}||/2) \\ \bar{e}_1 \sin(||\theta_{xy}||/2) \\ 0 \end{bmatrix}$$

$\bar{e} \in \mathbb{R}^3$ denotes the orientation given as a unit vector

$$\bar{e} = \begin{bmatrix} \bar{e}_0 \\ \bar{e}_1 \\ \bar{e}_2 \end{bmatrix} = \begin{bmatrix} \theta_x \\ \theta_y \\ \theta_z \end{bmatrix}, \quad ||\bar{e}|| = 1$$

where θ_x and θ_y are references which values are between -1 and 1, and $\theta_z = 0$

$\theta_{xy} \in \mathbb{R}^3$ is a vector that represents the desired orientation in axis-angle representation, and is given by

$$\theta_{xy} = \begin{bmatrix} \theta_x \alpha_{max_x} \\ \theta_y \alpha_{max_y} \\ \theta_z \alpha_{max_z} \end{bmatrix},$$

where α_{max_x}, α_{max_y}, and α_{max_z} are the maximum angles that the arm can be rotated. Similarly, the quaternion reference for the z axis is defined as

$$\boldsymbol{q}_z = \begin{bmatrix} \boldsymbol{q}_{z_0} \\ \boldsymbol{q}_{z_1} \\ \boldsymbol{q}_{z_2} \\ \boldsymbol{q}_{z_3} \end{bmatrix} = \begin{bmatrix} \cos(\theta_z) \\ 0 \\ 0 \\ \sin(\theta_z) \end{bmatrix} \tag{3.41}$$

where θ_z is the desired angle for the final effector around the z axis.

3.2.4.1 Inverse Kinematics

In order to stabilize the system, the desired transformation for the vehicle must be defined, this is accomplished by taking the reference of the final efector $\hat{q}_{fd}$, and using inverse kinematics to obtain the respective reference for the quadrotor $\hat{q}_{vd}$.

From equation (2.59), the transformation of the vehicle can be obtained with

$$\hat{q}_v = \hat{q}_f \otimes \hat{q}_b^* \otimes \hat{q}_c^* \tag{3.42}$$

If $\hat{q}_f$ is replaced with $\hat{q}_{fd}$ from equation (3.40), then the reference for the quadrotor platform is given by

$$\hat{q}_{vd} = \hat{q}_{fd} \otimes \hat{q}_b^* \otimes \hat{q}_c^* \tag{3.43}$$

with

$$\hat{q}_{vd} = \boldsymbol{q}_{vd} + \frac{\boldsymbol{q}_{vd} \otimes T_{vd}}{2} \epsilon. \tag{3.44}$$

As it was before stated the vehicle's orientation must be stabilized in a trajectory defined in equation (3.38), but this orientation doesn't give a reference around the z axis because this angle does not affect the direction of the force F_{uI}.

Let $\boldsymbol{q}_z$ be a quaternion that defines the desired angle to be rotated around the z axis, then the reference $\boldsymbol{q}_d$ given by the trajectory is completed with $\boldsymbol{q}_z$ using a quaternion product as

$$\boldsymbol{q}_{vd} = \boldsymbol{q}_d \otimes \boldsymbol{q}_z \tag{3.45}$$

The reference for the orientation of the mechanical arm is given by

$$\boldsymbol{q}_{bd} = \boldsymbol{q}_{xz} \ ,$$

Thus the reference is calculated using equation (3.42), then $\boldsymbol{q}_{vd}$ is replaced using (3.45), and then the error is calculated using equation (3.35)

Chapter 4

Numeric Validation

Systems proposed in chapter 2 (equations (2.53), and (2.60)) were simulated in order to validate the closed-loop system stability, using the Python programming language, along with the SciPy, NumPy, and Python Systems Control libraries, see Jones et al. [2001], Van Der Walt et al. [2011], and Goppert et al. [2014]. The quadrotor models (2.30) and (2.53) were simulated by integrating numerically $\dot{x}$ using SciPy's *integrate* function for a period of ten and five seconds respectively.

4.1 Unit Quaternion Control Simulations

The control inputs (3.5) and (3.15) were implemented using K_{att} and K_{pos}, which were determined using a linear quadratic regulator using the function *lqr()* included in the python control library. The results of this simulations were presented by our team in Cariño et al. [2015]

$$K_{att} = \begin{bmatrix} 100 & 0 & 0 & 14.142 & 0 & 0 \\ 0 & 100 & 0 & 0 & 14.142 & 0 \\ 0 & 0 & 3.16\text{e}-6 & 0 & 0 & 2.51e-3 \end{bmatrix}$$

$$K_{pos} = \begin{bmatrix} 3.1622 & 0 & 0 & 2.7063 & 0 & 0 \\ 0 & 3.1622 & 0 & 0 & 2.7063 & 0 \\ 0 & 0 & 3.1622 & 0 & 0 & 2.7063 \end{bmatrix}$$

The quaternion operations were previously defined using NumPy's vector and cross products, and then included in a custom library. The parameters used were the following.

$$J = \begin{bmatrix} 0.0118976 & 0 & 0 \\ 0 & 0.0118976 & 0 \\ 0 & 0 & 0.0218816 \end{bmatrix}$$

$$p_0 = \begin{bmatrix} 1 & 2 & 0 \end{bmatrix}^T$$

$$\dot{p}_0 = \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}^T$$

$$\omega_0 = \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}^T$$

$$p_d = \begin{bmatrix} 1 & 2 & 0 \end{bmatrix}^T$$

$$\dot{p}_d = \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}^T$$

$$m = 1.3\,kg$$

4.1.1 Attitude Stabilization

The first simulation was defined to stabilize the attitude of the quadrotor. Figures 4.1 and 4.2 show the stabilization of the vehicle states, for example in Figure 4.1, it can be seen that the quaternion stabilizes in the equilibrium point. Similarly, in

Figure 4.2 we can observe that the angular velocities converge to zero. This means that the vehicle will converge exponentially to a given orientation and stay still in it.

A trajectory is calculated from the position control (see section 4.1.2) $\boldsymbol{q}_d$ comes from the position control, the desired attitude is illustrated with $\boldsymbol{q}_d$ in Figure 4.3. Figure 4.4 illustrates the behavior of the quaternion error. Since the quaternion error is defined as $\boldsymbol{q}_e = \boldsymbol{q} \otimes \boldsymbol{q}_d^{-1}$, so in this case the attitude ($\boldsymbol{q}$) converges to the desired trajectory ($\boldsymbol{q}_d$), making the error $\boldsymbol{q}$ converge to the unit quaternion.

On the other hand, Figure 4.5 depicts the angular velocity errors in a logarithmic scale, it can be seen that the angular velocity converges to zero at an infinite time, since the error values approach to $-\infty$, and the inverse of the logarithmic scale $e^{-\infty} \to 0$. In practical terms it can be considered that the system's angular velocity really converges to zero.

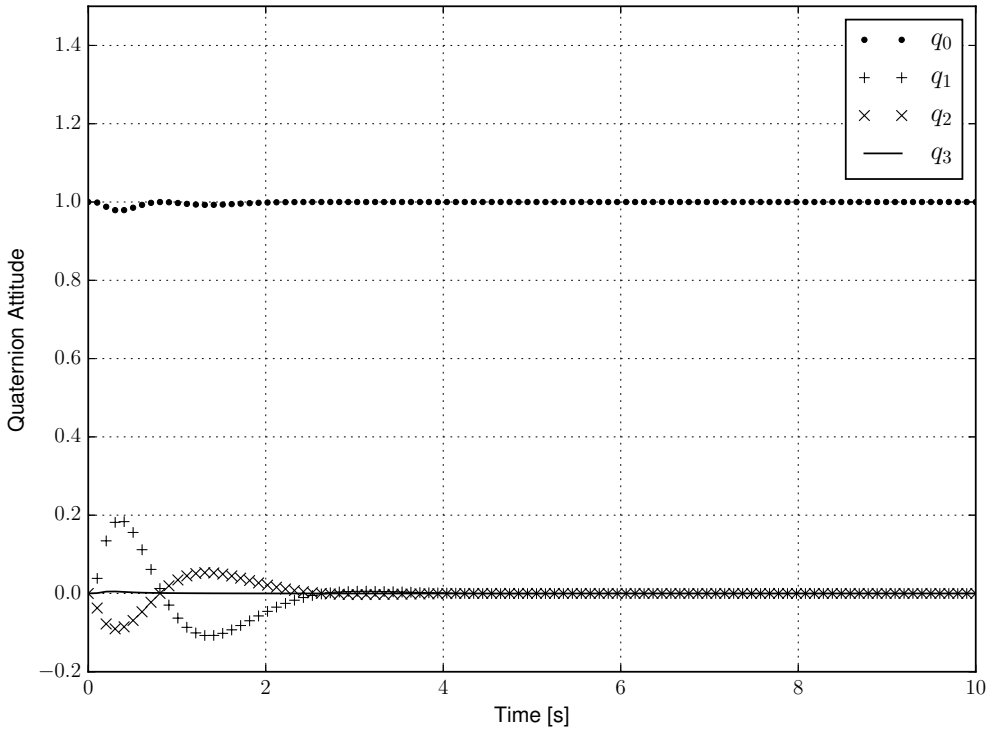

FIGURE 4.1: Quadcopter attitude (q).

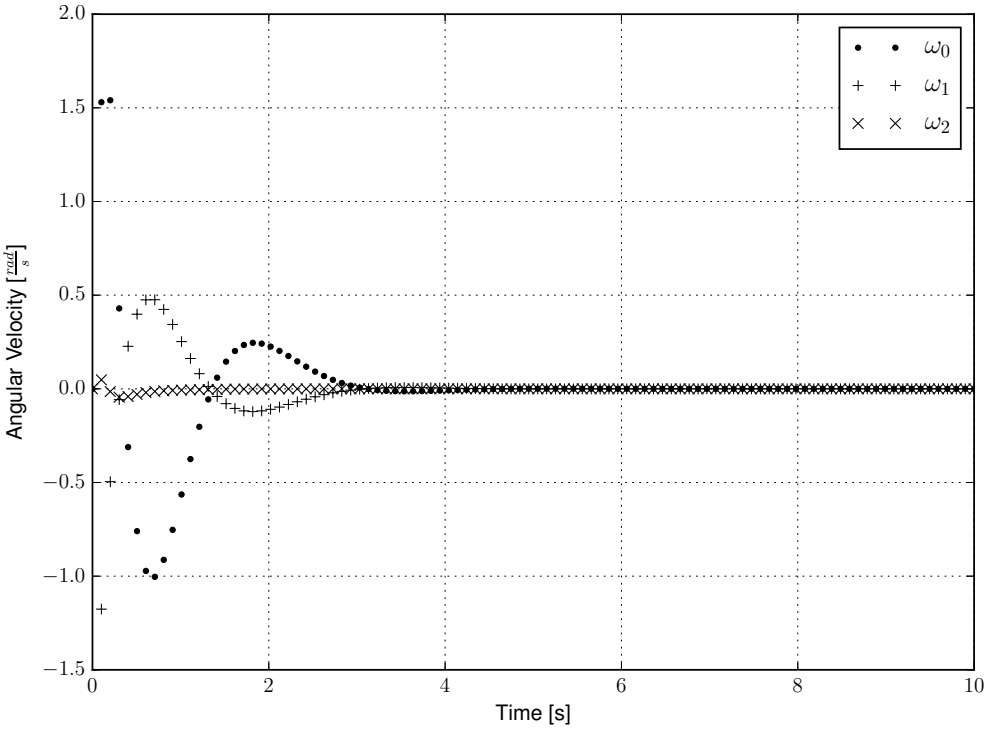

FIGURE 4.2: Quadcopter rotational velocity ω.

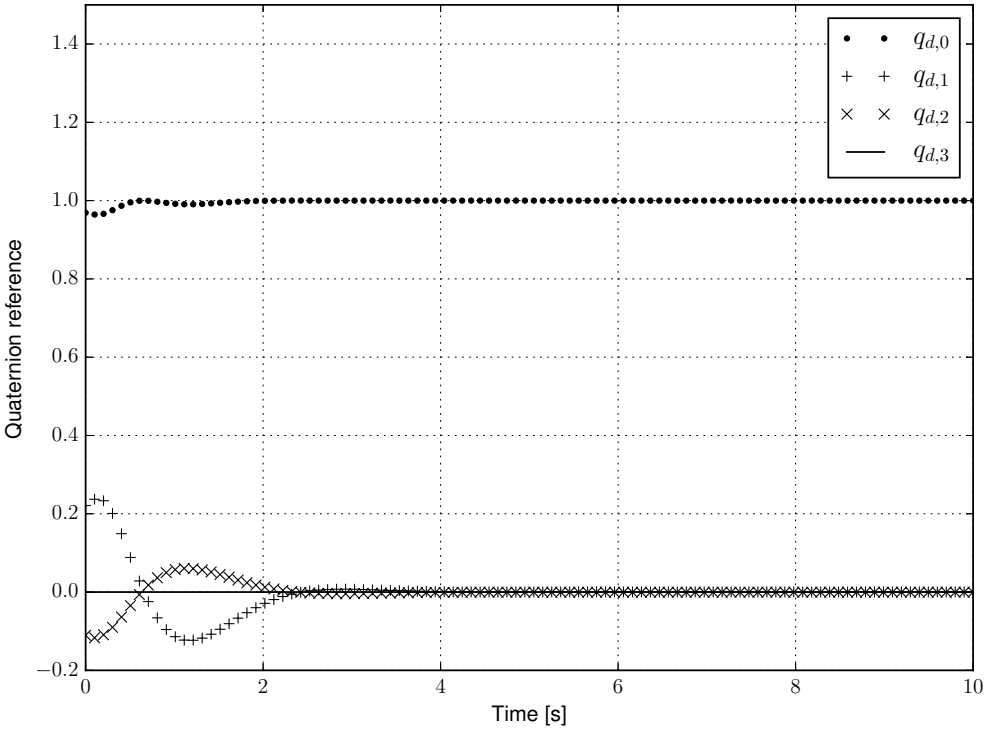

FIGURE 4.3: Quadcopter attitude reference (q_d).

4.1.2 Translational Stabilization

An initial position $p_0 = [1, 2, 0]$ was given, then a desired equilibrium point $p_d = [0, 0, 1]$ was proposed as a reference to globally stabilize the system.

Figures 4.6 and 4.7 show that the system's position exponentially converges from the initial condition p_0 to the reference p_d.

Figures 4.8 and 4.9 introduce the translational error in a logarithmic scale, we can see that it converges to zero in an infinite time. This error is insignificant in practical terms.

The resulting error is used to calculate the attitude reference trajectory shown before in section 4.1.1, see Figure 4.3. In Figure 4.9 the translational velocity errors in a logarithmic scale are depicted, here it can be seen that the angular velocity converges to zero at an infinite time. In practical terms it can be considered that the system's

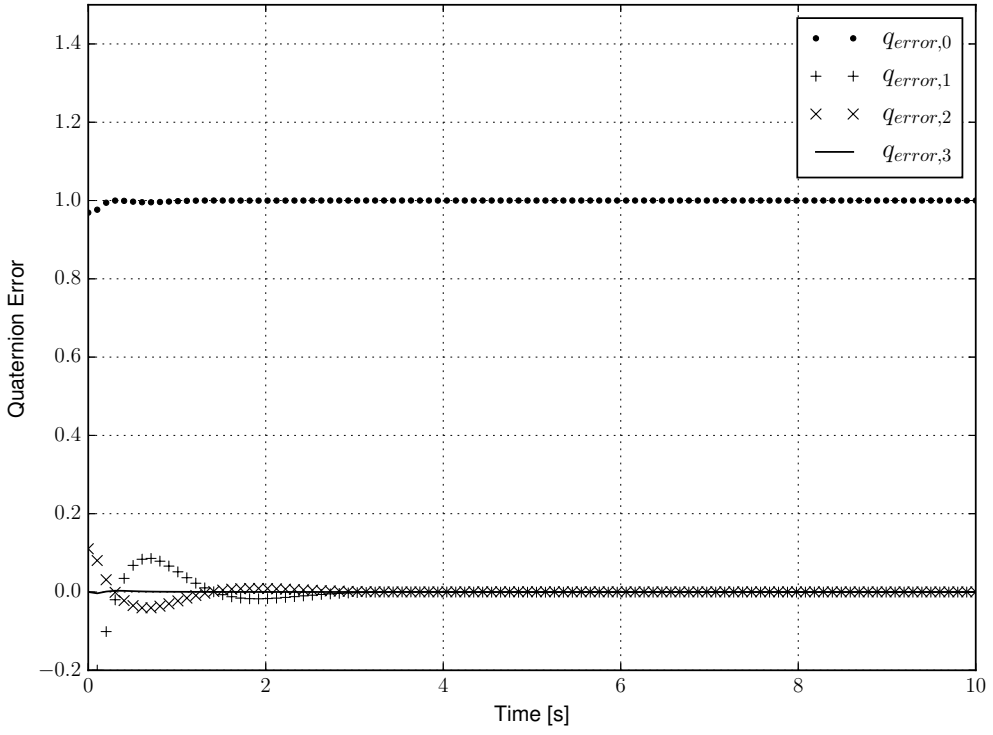

FIGURE 4.4: Quadcopter attitude error (q_e).

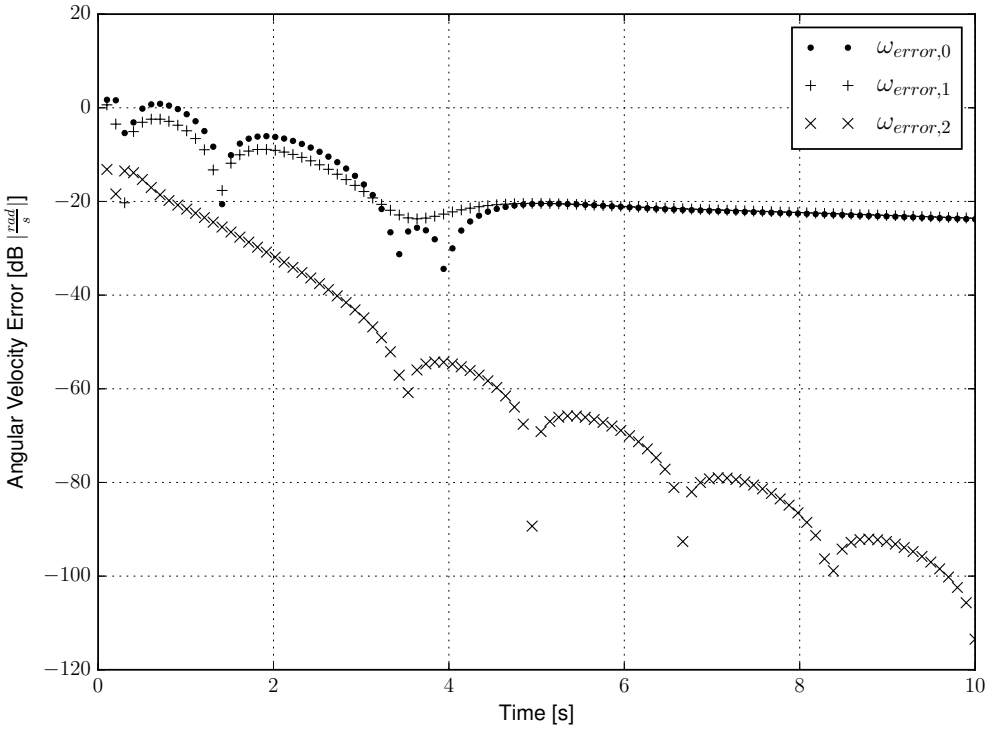

FIGURE 4.5: Quadcopter attitude velocity error (ω_e).

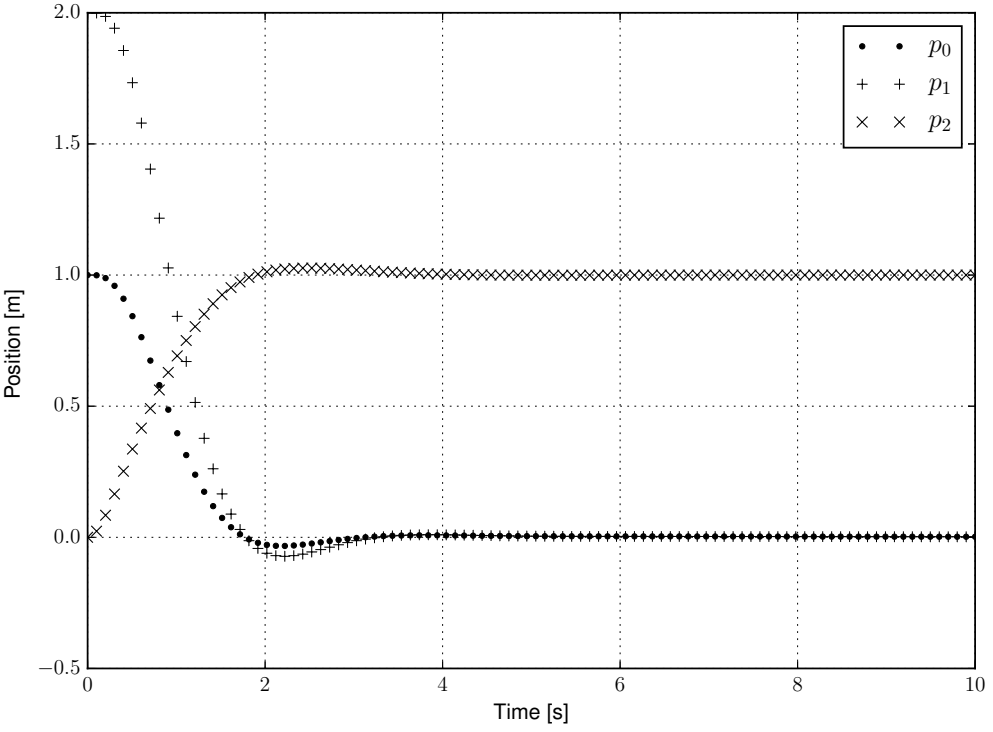

FIGURE 4.6: Quadcopter translational position (p).

angular velocity really converges to zero. Figure 4.10 shows the system inputs τ_{ux}, τ_{uy}, and τ_{uz}, notice that they converge to zero when the system stabilizes.

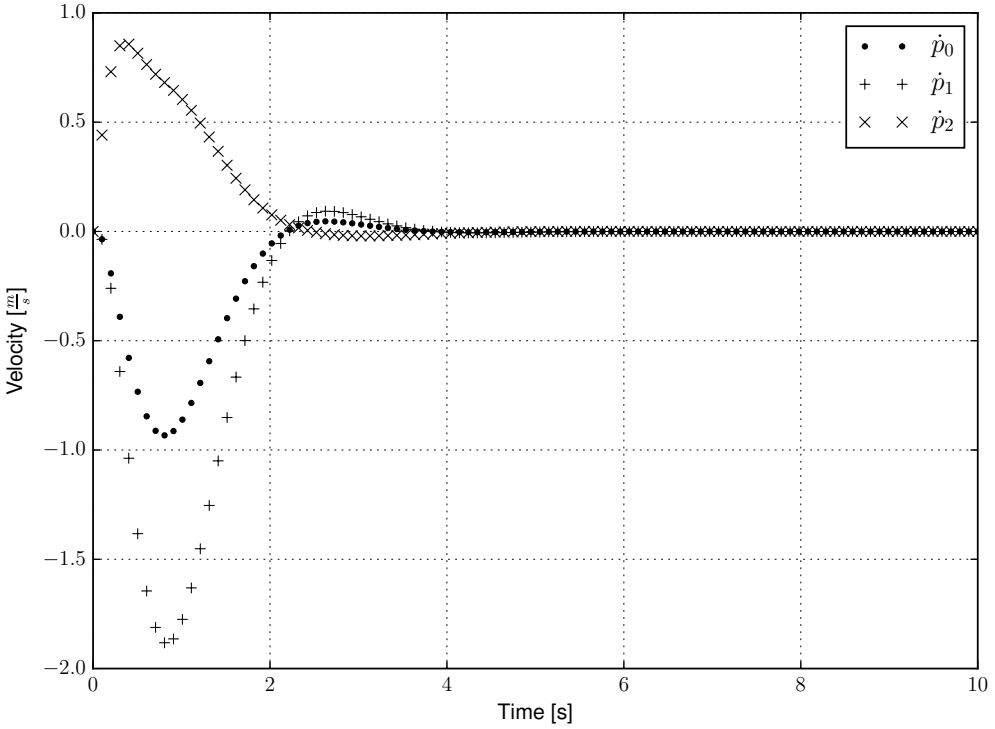

FIGURE 4.7: Quadcopter translational velocity ($\dot{p}$).

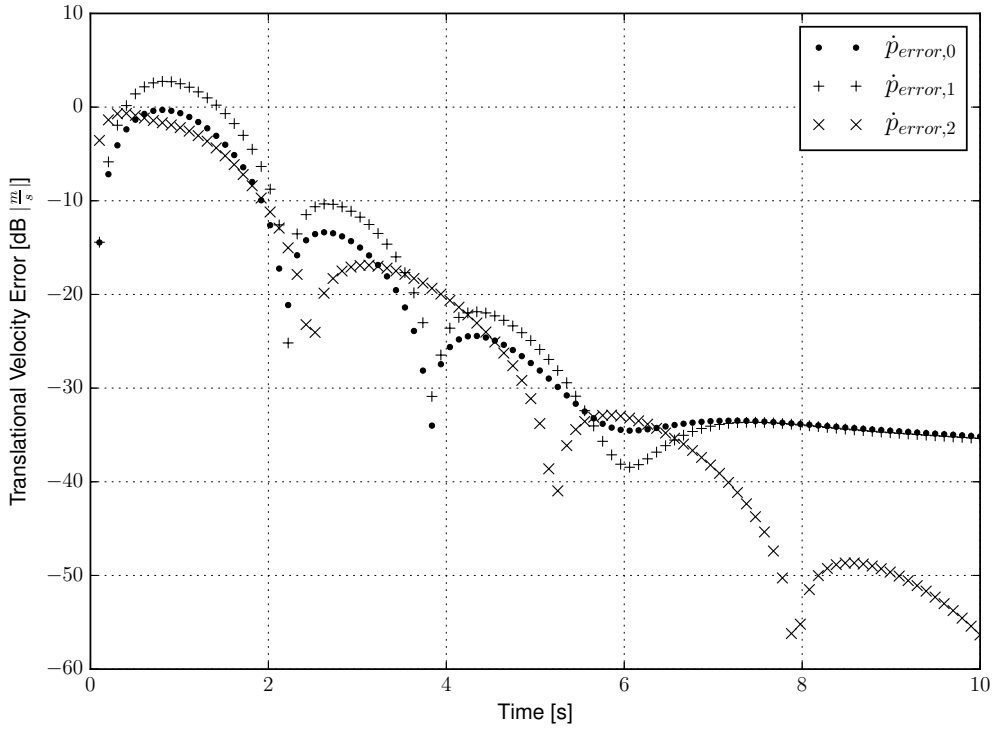

FIGURE 4.9: Quadcopter translational velocity error.

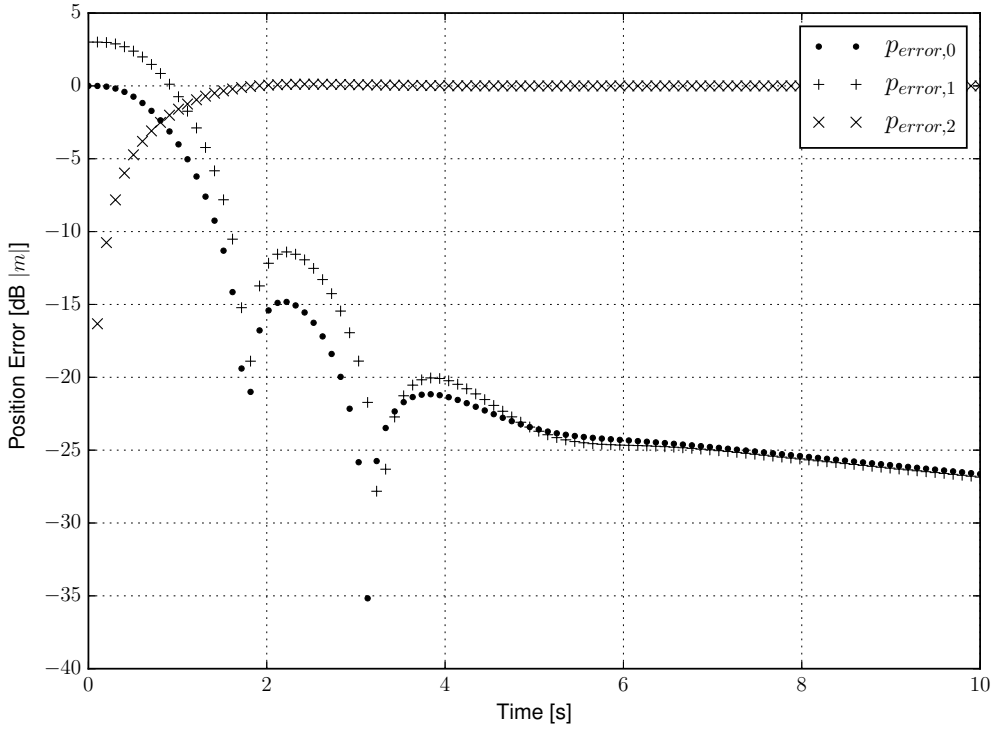

FIGURE 4.8: Quadcopter translational position error.

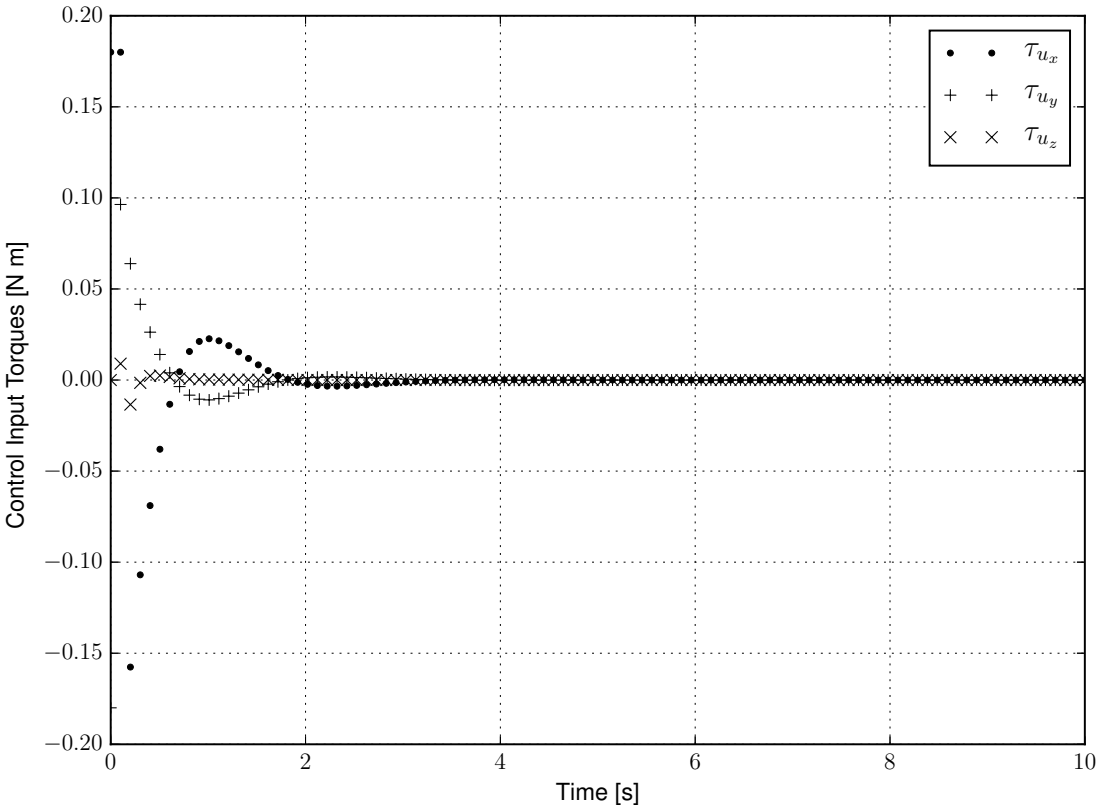

FIGURE 4.10: Quadcopter input.

4.2 Dual Quaternion Control Simulations

Before the platform with the robotic arm is simulated, the control law from equation (3.18) was numerically validated considering only the quadcopter vehicle. The control gains $\hat{k}_p$ and $\hat{k}_v$ were selected in base of matrices K_{att} and K_{pos}, and making some empirical modifications

$$\hat{k}_p = \begin{bmatrix} 100 \\ 100 \\ 3.16 \end{bmatrix} + \begin{bmatrix} 3.1622 \\ 3.1622 \\ 3.1622 \end{bmatrix} \epsilon \quad , \quad \hat{k}_v = \begin{bmatrix} 14.142 \\ 14.142 \\ 2.51 \end{bmatrix} + \begin{bmatrix} 2.7063 \\ 2.7063 \\ 2.7063 \end{bmatrix} \epsilon \tag{4.1}$$

The dual quaternion operations were previously defined using NumPy's vector and cross products, and then included in a custom library. The vehicle's parameters and initial values $\hat{q}_{ini}$ were considered as

$$\begin{aligned}
J &= \begin{bmatrix} 0.0118976 & 0 & 0 \\ 0 & 0.0118976 & 0 \\ 0 & 0 & 0.0218816 \end{bmatrix} \\
\hat{q}_{ini} &= \boldsymbol{q}_{ini} + \frac{\boldsymbol{q}_{ini} \otimes T_{ini}}{2} \epsilon \\
\boldsymbol{q}_{ini} &= \frac{1 + \begin{bmatrix} 0 & 0 & \frac{\pi}{10} \end{bmatrix}^T}{\text{norm}\left(1 + \begin{bmatrix} 0 & 0 & \frac{\pi}{10} \end{bmatrix}^T\right)} \\
T_{ini} &= \begin{bmatrix} 1 & 1 & 1 \end{bmatrix}^T \\
\dot{T}_{ini} &= \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}^T \\
\omega_{ini} &= \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}^T \\
T_d &= \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}^T \\
\dot{T}_d &= \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}^T \\
m &= 1.3\, kg
\end{aligned}$$

4.2.1 Dual Quaternion Stabilization

The results of the simulation were divided in order to understand them in a more intuitive manner, the attitude and the position were extracted from the dual quaternion and plotted to better understand the behavior of the quadrotor.

An initial position T_{ini} was given, then a desired equilibrium point T_d was proposed as a reference to globally stabilize the system. Figures 4.11 and 4.12 show that the system's position exponentially converges from the initial condition to the reference.

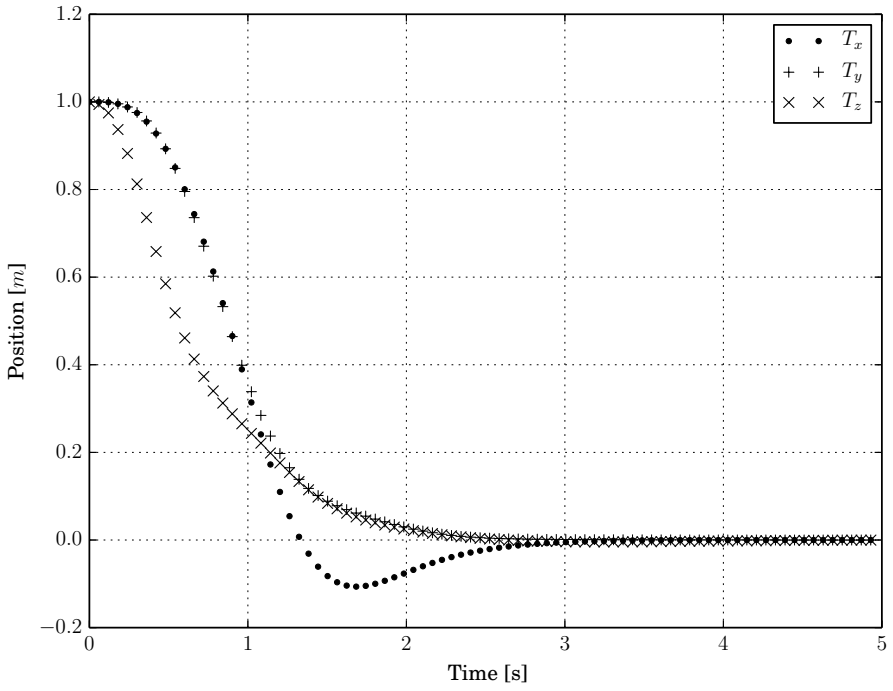

FIGURE 4.11: Quadcopter translational position (p) using dual quaternions.

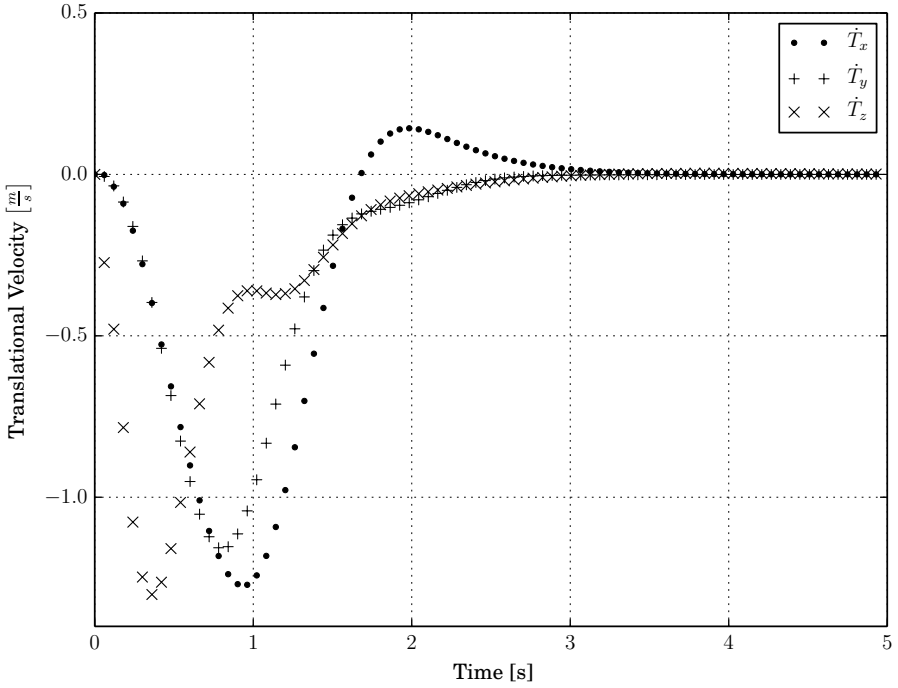

FIGURE 4.12: Quadcopter translational velocity ($\dot{p}$) using dual quaternions.

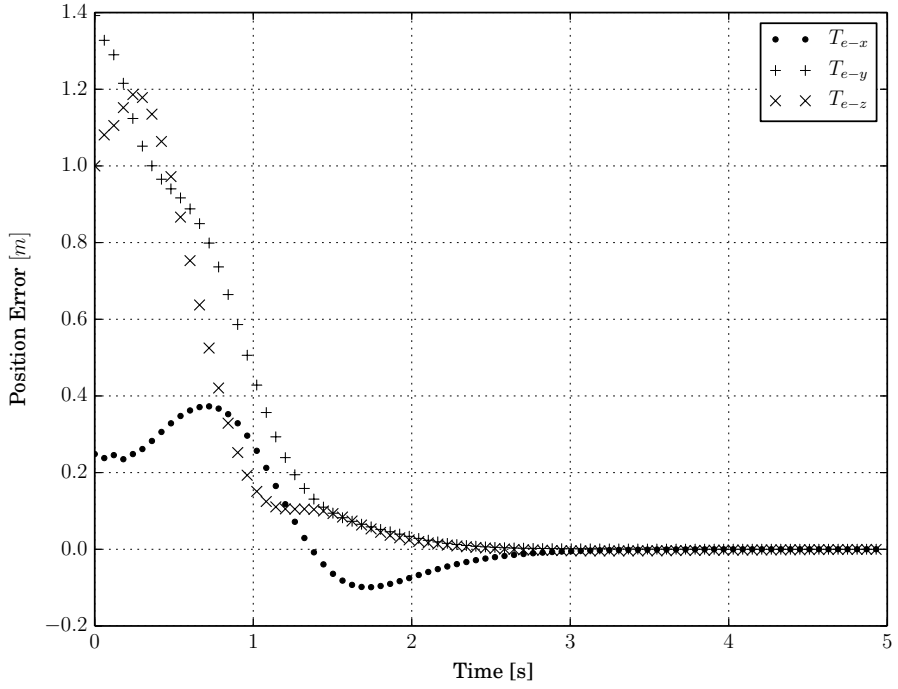

FIGURE 4.13: Quadcopter translational position error using dual quaternions.

The resulting error is used to calculate the attitude reference trajectory $\boldsymbol{q}_d$ shown before in section 4.3.1, see Figure 4.14. Figure 4.15 shows the system inputs τ_{ux}, τ_{uy}, and τ_{uz}, notice that they converge to zero when the system stabilizes in the desired trajectory.

It can be seen in Figure 4.16 that the quaternion stabilizes in the equilibrium point q_0. Similarly, Figure 4.17 shows that the angular velocities converge to zero. This means that the vehicle follows the trajectory $\boldsymbol{q}_d$ complimented with the desired reference around the "z" axis, thus stabilizing the dual quaternion to the equilibrium point.

Figures 4.18 and 4.19 show the attitude and angular velocity errors respectively. It can be seen that $\boldsymbol{q}_e$ converges to the unit quaternion while ω_e stabilizes in zero.

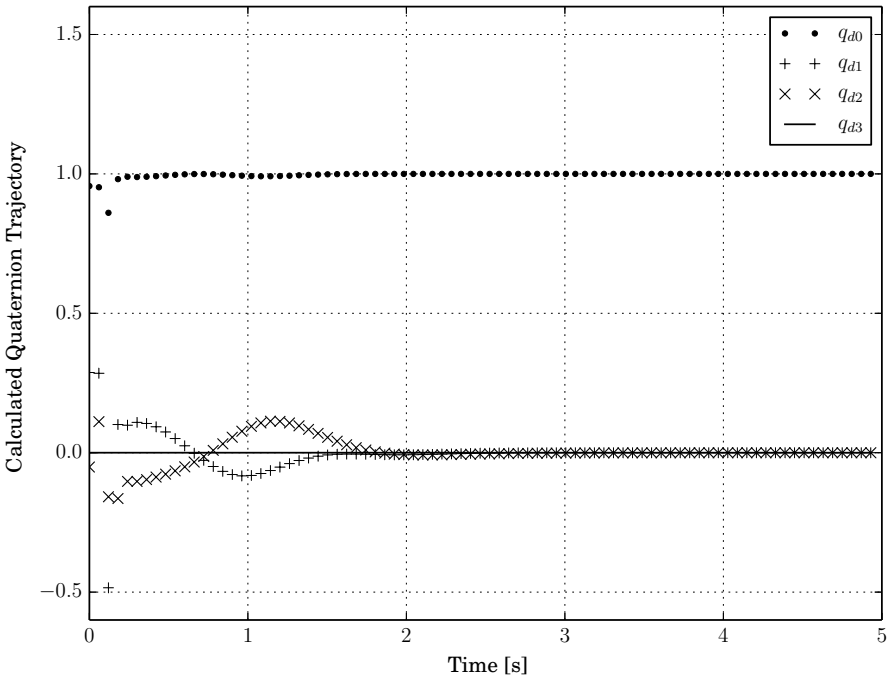

FIGURE 4.14: Quadcopter attitude reference (q_d) using dual quaternions.

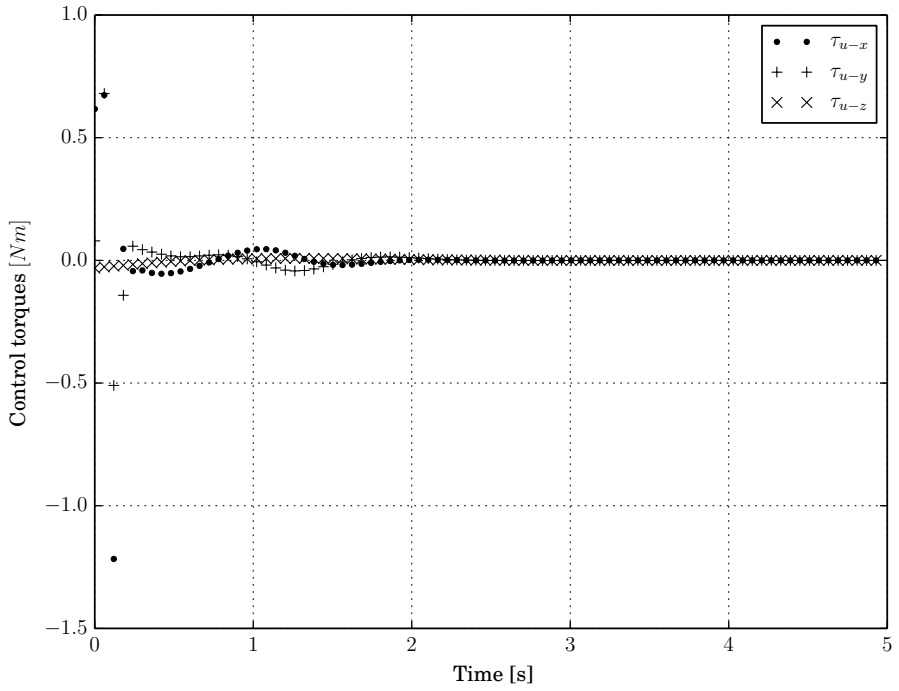

FIGURE 4.15: Quadcopter input using dual quaternions

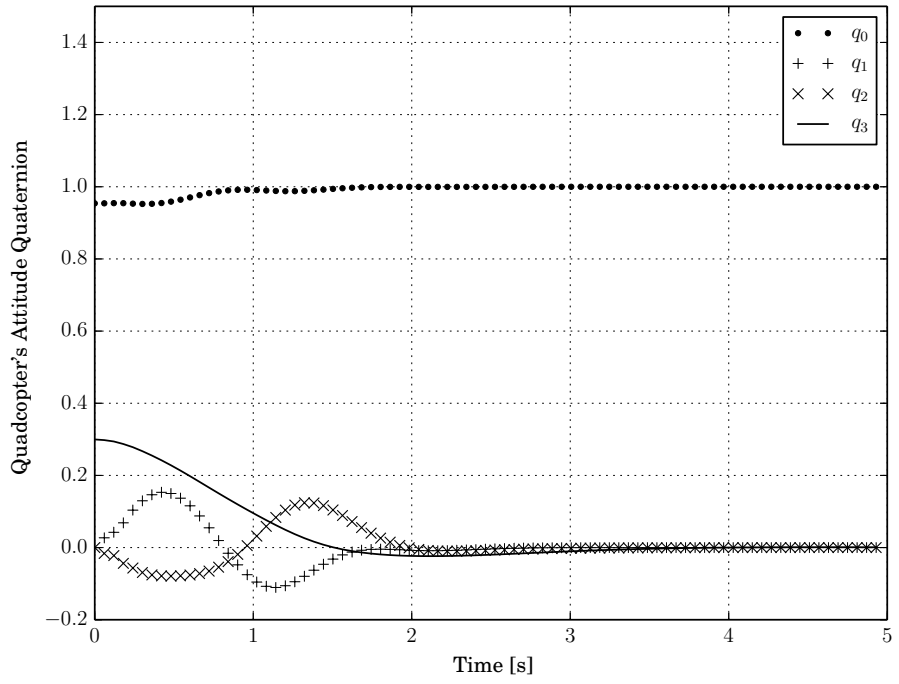

FIGURE 4.16: Quadcopter attitude (q) using dual quaternions.

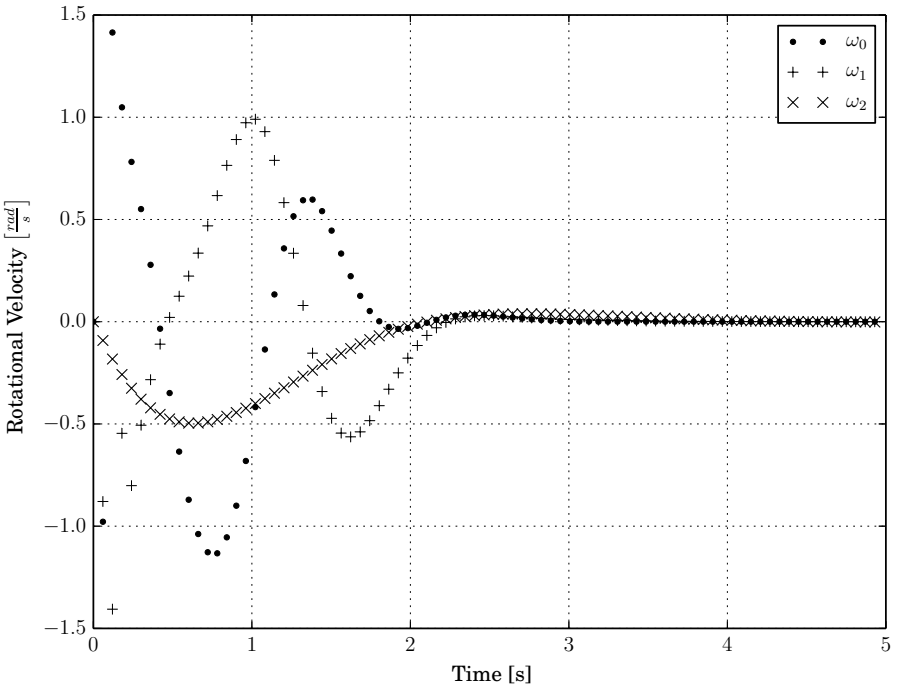

FIGURE 4.17: Quadcopter rotational velocity ω using dual quaternions.

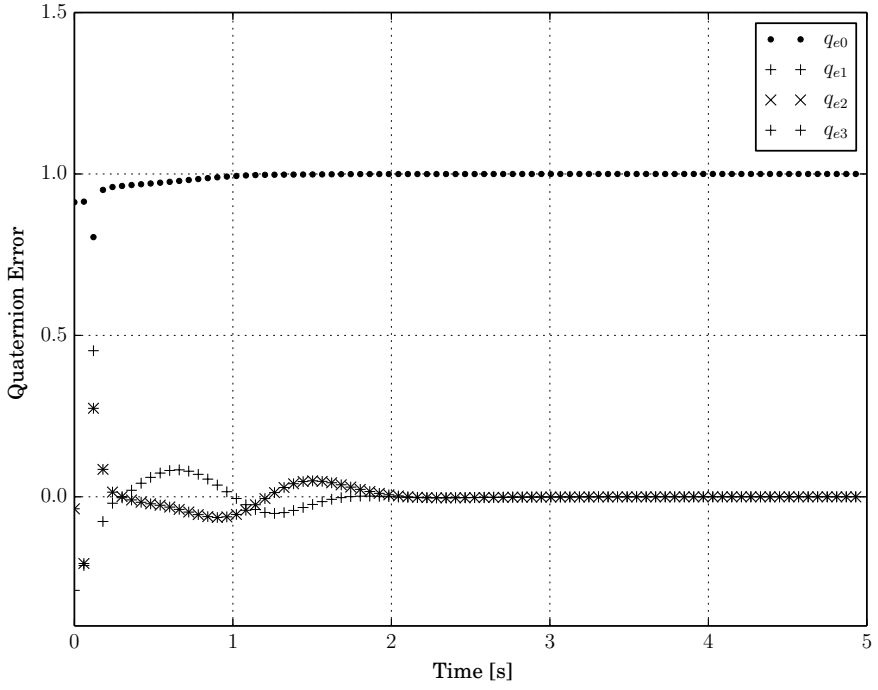

FIGURE 4.18: Quadcopter attitude error (q_e) using dual quaternions.

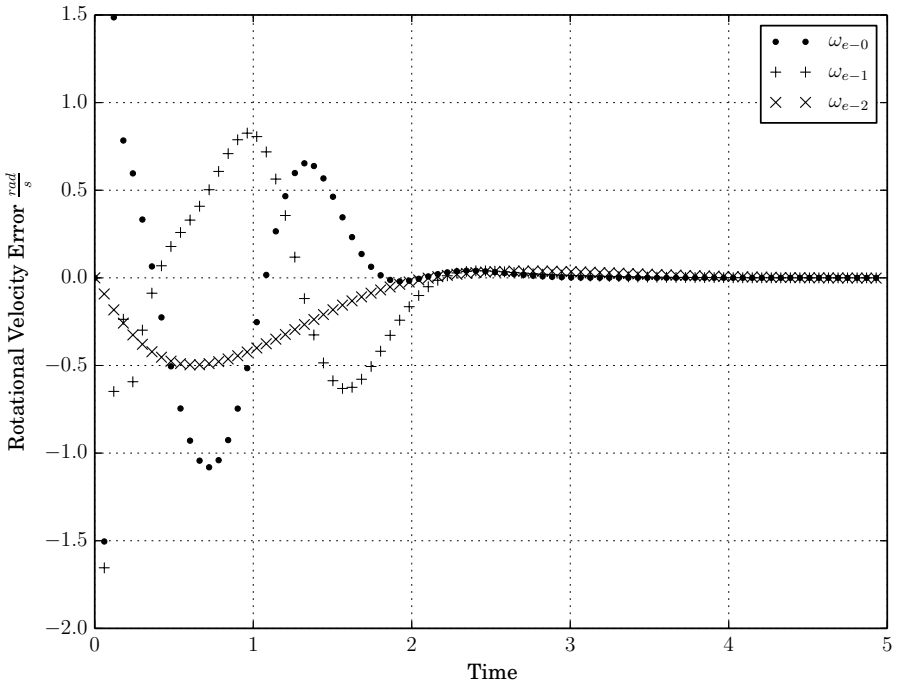

FIGURE 4.19: Quadcopter attitude velocity error (ω_e) using dual quaternions.

4.3 Quadrotor with Robotic Arm Simulations

The robotic arm effects were incorporated to the simulation, a desired reference for the final effector $\hat{q}_f$ was given as a dual quaternion, the reference for the quadrotor platform was calculated using equation (3.42). The torques that the arm cause on the vehicle were compensated using equation (2.33) to calculate the external torques of equation (3.30).

This simulation used the same values for $\hat{k}_p$ and $\hat{k}_v$ as well as for the vehicle's parameters as in section 4.2, the same custom library was also used in this simulation.

$$
\begin{aligned}
J &= \begin{bmatrix} 0.0118976 & 0 & 0 \\ 0 & 0.0118976 & 0 \\ 0 & 0 & 0.0218816 \end{bmatrix} \\
\hat{q}_{ini} &= \boldsymbol{q}_{ini} + \frac{\boldsymbol{q}_{ini} \otimes T_{ini}}{2}\epsilon \\
\boldsymbol{q}_{ini} &= \frac{1 + \begin{bmatrix} 0 & 0 & \frac{\pi}{10} \end{bmatrix}^T}{\text{norm}\left(1 + \begin{bmatrix} 0 & 0 & \frac{\pi}{10} \end{bmatrix}^T\right)} \\
T_{ini} &= \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}^T \\
\dot{T}_{ini} &= \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}^T \\
\omega_{ini} &= \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}^T \\
T_{fd-I} &= \begin{bmatrix} 0.5 & 0.5 & 0.5 \end{bmatrix}^T \\
\dot{T}_{fd} &= \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}^T \\
m &= 1.3\,kg \\
m_b &= 0.2kg
\end{aligned} \tag{4.2}
$$

4.3.1 Dual Quaternion Stabilization

In order to better understand the behavior of the system, the plots corresponding to the quadrotor's orientation and position are firstly presented, then the graphs that describe the orientation and position of the end effector are shown.

It is supposed that the end effector begins the simulation in a point given by $\hat{q}_{ini}$, then a desired equilibrium point $\hat{q}_{fd}$ was proposed following equation (3.40), as a reference to globally stabilize the system in a desired position in the inertial frame T_{fd-I}, considering that the orientation of the vehicle will follow a rotation given in axis-angle representation by $\bar{\theta}_b = [\pi/4, -\pi/4, \pi/10]^T$. Using the inverse kinematics obtained from equation (3.43) and (3.44), a position reference shown in Figure 4.20 is calculated for the quadrotor vehicle.

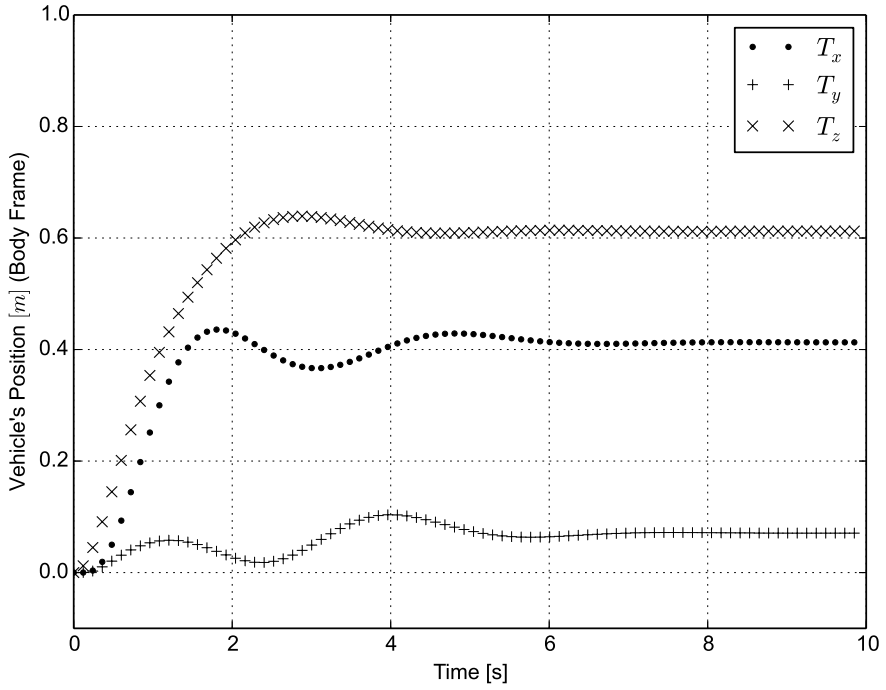

FIGURE 4.20: Quadcopter translational position (T), with arm effects.

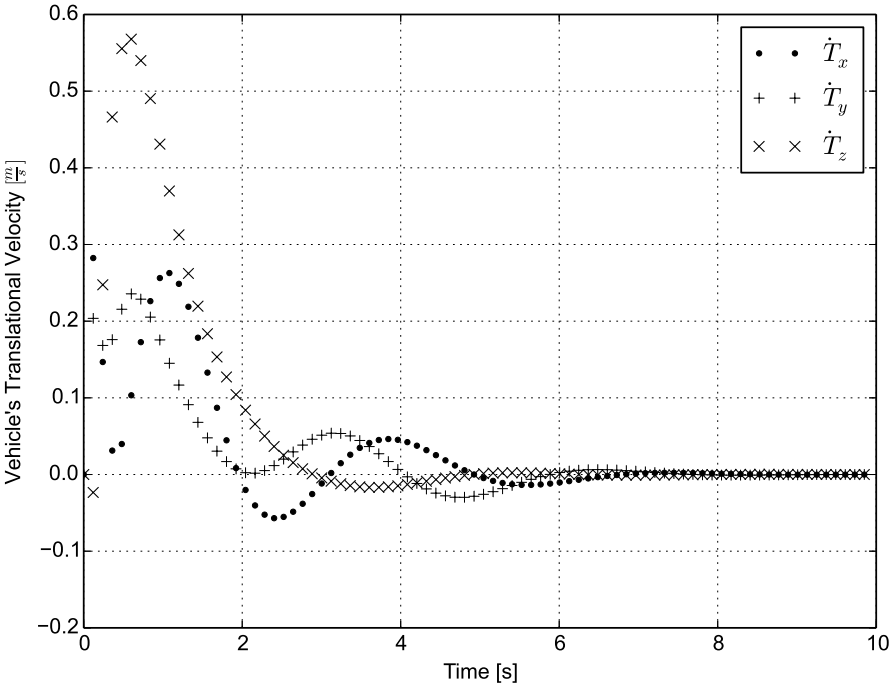

FIGURE 4.21: Quadcopter translational velocity ($\dot{T}$), with arm effects.

Figure 4.23 displays the trajectory calculated with the position error, this reference is used in the control law to calculate the input torques τ_{ux}, τ_{uy}, and τ_{uz} shown in Figure 4.24.

It can be seen in Figure 4.25 that the quaternion stabilizes in the equilibrium point q_d, in which the vehicle's only rotation is around the z axis. Figure 4.26 shows that the angular velocities converge to zero. This means that the vehicle follows the trajectory $\boldsymbol{q}_d$ complimented with the desired reference around the z axis, thus stabilizing the dual quaternion to the equilibrium point.

Figures 4.27 and 4.28 show the attitude and angular velocity errors respectively. It can be seen that $\boldsymbol{q}_e$ converges to the unit quaternion while ω_e stabilizes in zero.

Finally, Figures 4.29, 4.30 and 4.31 represent respectively the orientation quaternion and final position (in the body and inertial frame) of the end effector.

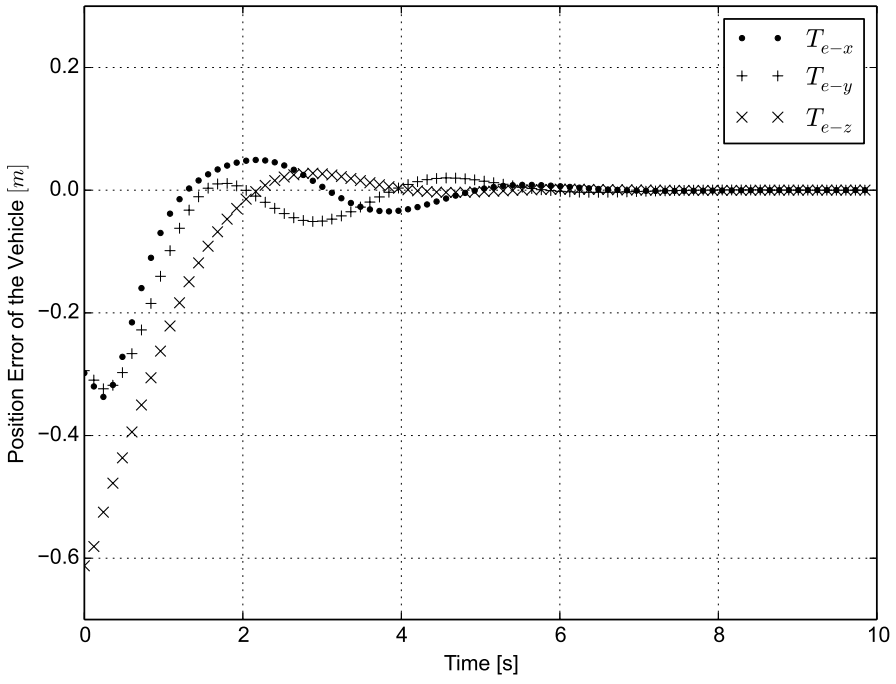

FIGURE 4.22: Quadcopter translational position error, with arm effects.

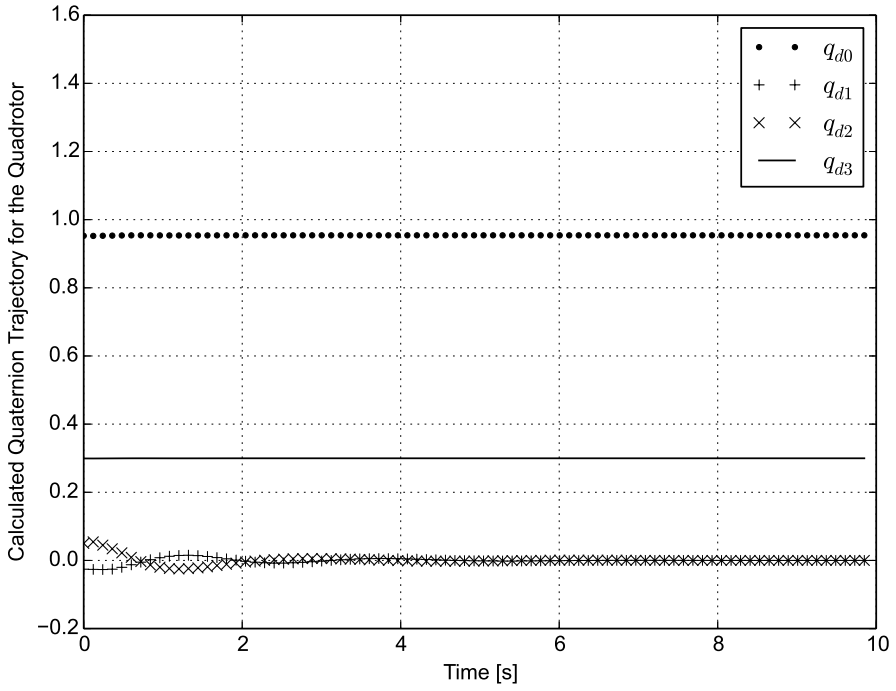

FIGURE 4.23: Quadcopter attitude reference (q_d) using dual quaternions.

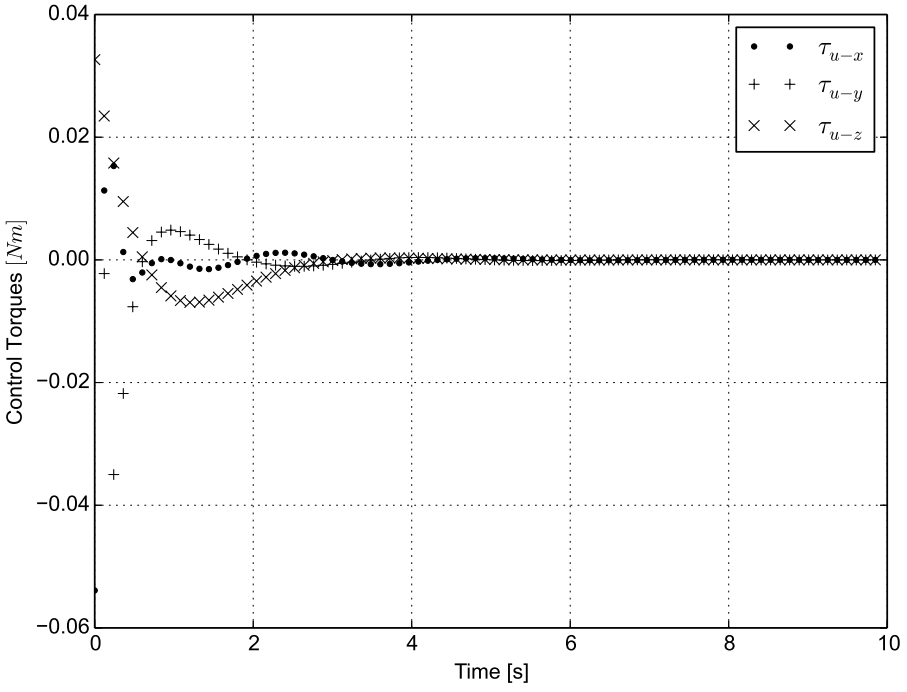

FIGURE 4.24: Quadcopter input using dual quaternions

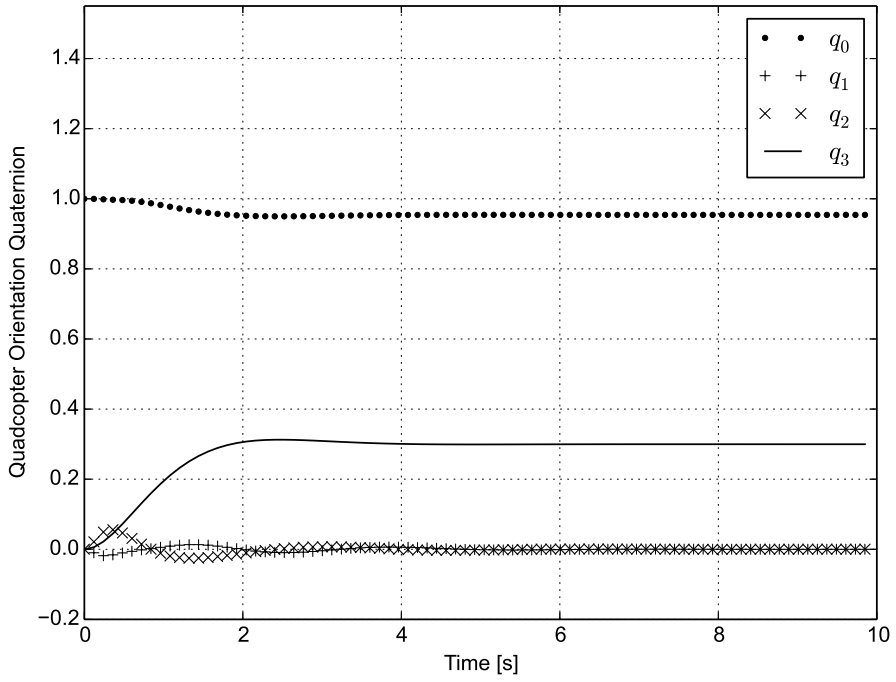

FIGURE 4.25: Quadcopter attitude (q), with arm effects.

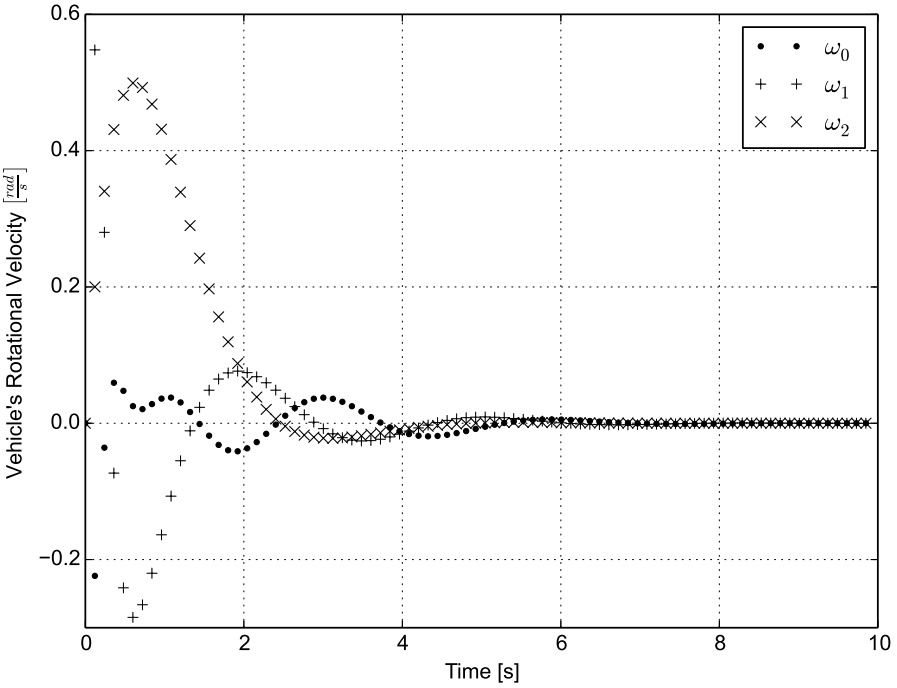

FIGURE 4.26: Quadcopter rotational velocity ω, with arm effects.

It can be seen that at the final dual quaternion $\hat{q}_f$, which is composed by the final rotation $\boldsymbol{q}_f$ and translation T_f, the position and orientation of the end effector converge asymptotically to the desired reference.

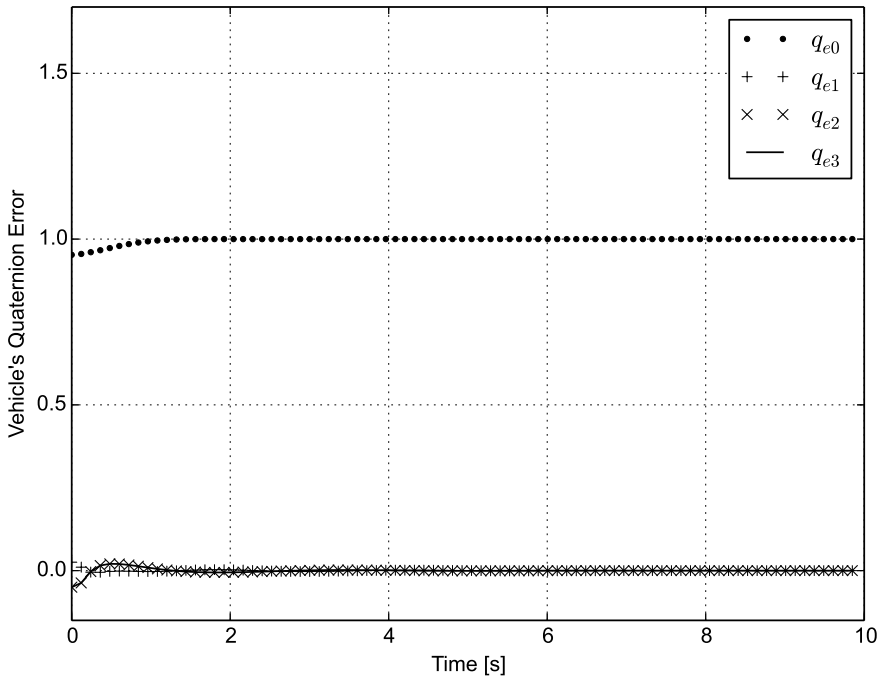

FIGURE 4.27: Quadcopter attitude error (q_e), with arm effects.

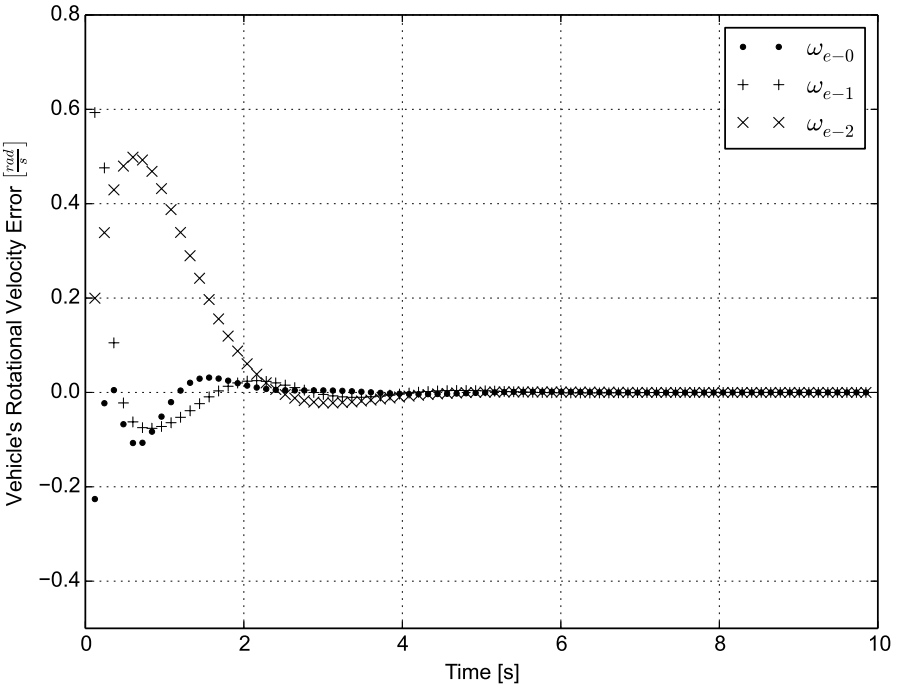

FIGURE 4.28: Quadcopter attitude velocity error (ω_e), with arm effects.

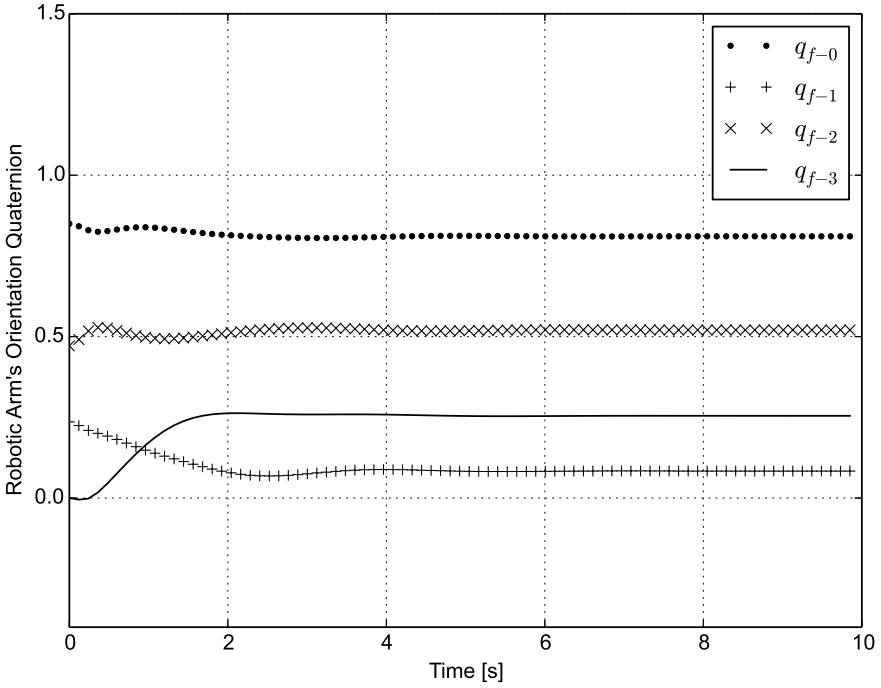

FIGURE 4.29: Attitude quaternion for the robotic arm (q_b).

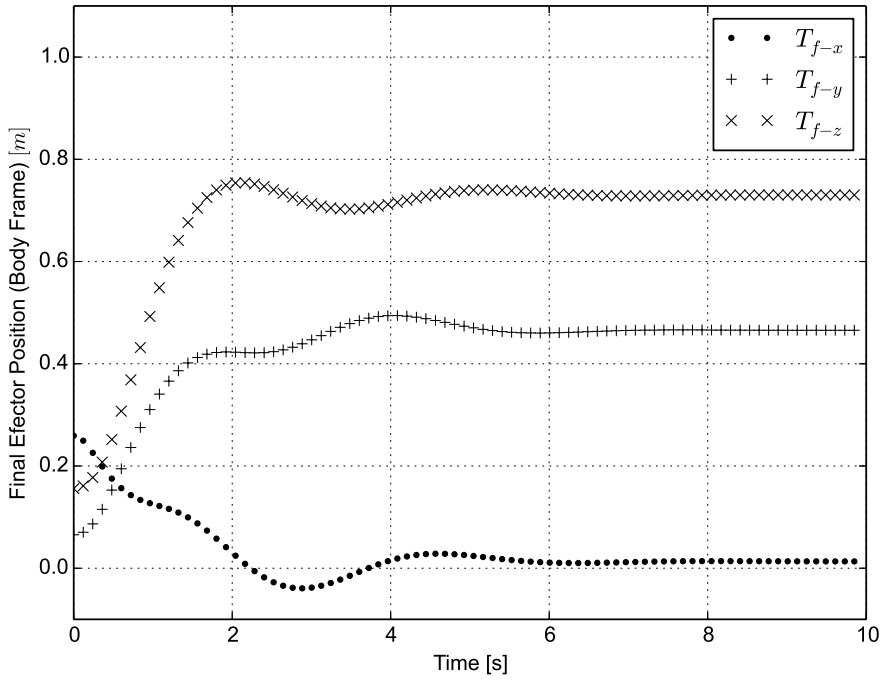

FIGURE 4.30: Final position for the end effector (T_f) in the body frame.

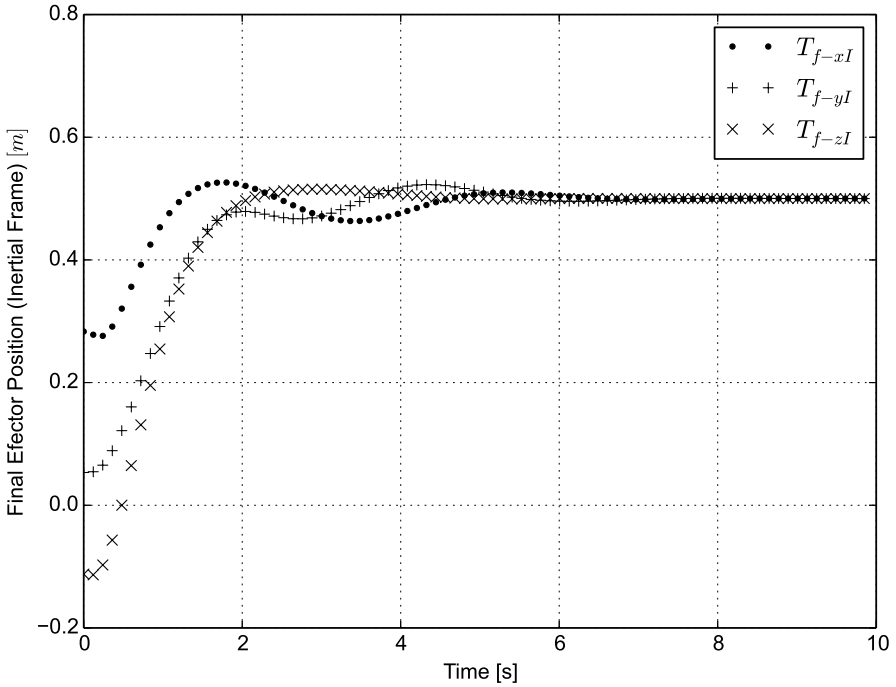

FIGURE 4.31: Final position for the end effector (T_{f-I}) in the inertial frame.

Chapter 5

Experimental Validation

The proposed control scheme was implemented in a quadrotor UAV platform. First, a control law to stabilize the vehicle's orientation using unit quaternions was validated, then the dual quaternion controller was implemented.

The dynamic effects of the robotic arm were calculated using equation (3.10) while the position and orientation reference for the quadcopter was computed using inverse kinematics from equation (3.42).

5.1 System Description

The proposed quadrotor UAV was built in the UMI-LAFMIA laboratory with the following technical description:

- Carbon fiber structure.
- 4 brushless motors with 14 poles each.
- 4 ESC speed controllers with 10 A capacity.
- 2.4 GHz Radio transmitter-receiver.
- 4 cell Li-Po battery with 5000 mAh charge capacity.

- 32 bits Pixhawk micro-computer for flight control.
  - Built-in accelerometers, gyroscopes, magnetometer, and barometer.
  - Arducopter firmware.
- PX4-Flow (optical flow) sensor.

FIGURE 5.1: Proposed quadrotor platform

Figure 5.1 shows the quadrotor platform that was used to validate the proposed control schemes before the robotic arm was taken into account.

Using direct measures and some estimations from a CAD software, and considering the mechanical properties of the main components of the UAV, the following parameters were obtained:

$$
\begin{aligned}
m_v &= 1.3kg \\
J_v &= \begin{bmatrix} 0.010334 & 0 & 0 \\ 0 & 0.010322 & 0 \\ 0 & 0 & 0.016985 \end{bmatrix}
\end{aligned}
$$

5.1.1 Robotic Arm Description

The robotic arm was based on metal gear digital servo motors, alongside a 2 DoF Robot arm with gripper and included servos.

(A) TGY-8103 servomotor (B) 2 DoF gripper

FIGURE 5.2: Mechanical actuators used in the robotic arm

One link was required to unite the mechanical actuator, the pieces that compose this link were designed using a CAD software, and built in a 3D printer.

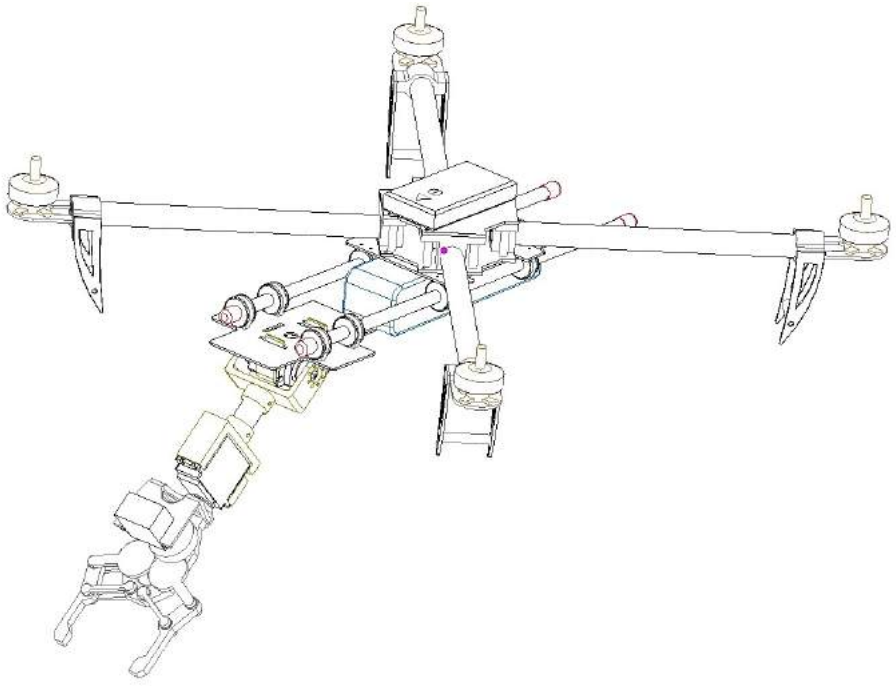

FIGURE 5.3: CAD sketch of the quadcopter with its robotic arm

Taking direct measures, and based on the CAD tools from the mechanical design, the following parameters for the robotic arm were obtained.

$$m_c = 0.109kg \quad , \quad J_c = \begin{bmatrix} 0.002245 & 0 & 0 \\ 0 & 0.000467 & 0 \\ 0 & 0 & 0.001923 \end{bmatrix}$$

$$m_b = 0.210kg \quad , \quad J_b = \begin{bmatrix} 0.004050 & 0 & 0 \\ 0 & 0.000103 & 0 \\ 0 & 0 & 0.004055 \end{bmatrix}$$

FIGURE 5.4: Quadrotor UAV with robotic arm

5.2 Experimental Results

Three different experiments were made in order to validate each proposed scheme. Firstly, the control law stated in equation (3.8) was implemented without considering the robotic arm effects (considering $\tau_{ext} = [0, 0, 0]^T$). Then the robotic arm effects were added following equation (2.31) to verify that the vehicle remains stable to the disturbances. Lastly, the control law from equation (3.30) based on dual quaternions was implemented to validate the position strategy.

5.2.1 Orientation Experiments

The resulting plots corresponding to the experiments that exclude the arm effects. The atitude, orientation reference (given by the radio controller), attitude error, angular velocity and input torques of the first flight test are shown in Figures 5.5 - 5.9.

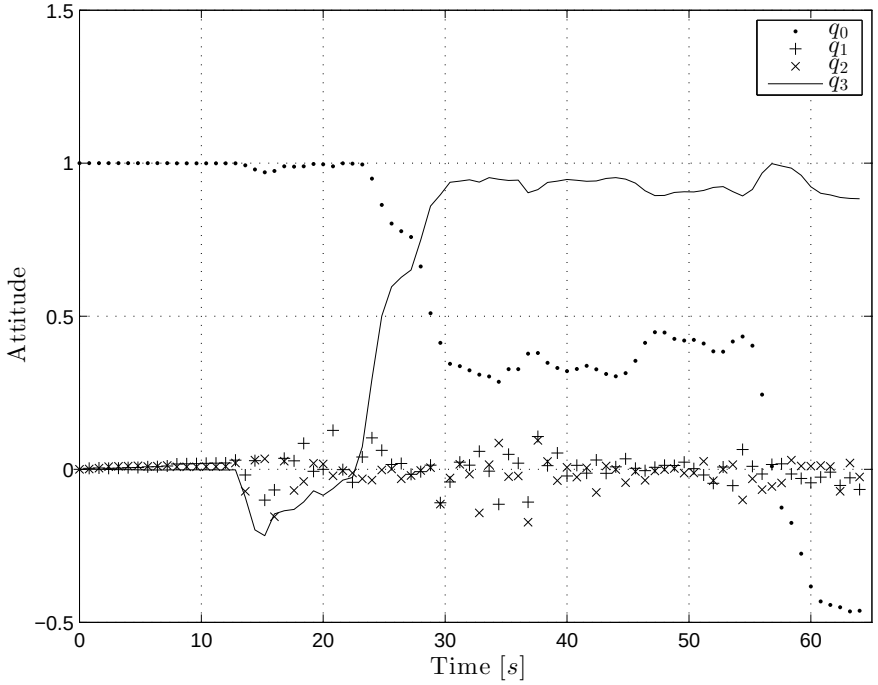

FIGURE 5.5: Quadrotor's attitude

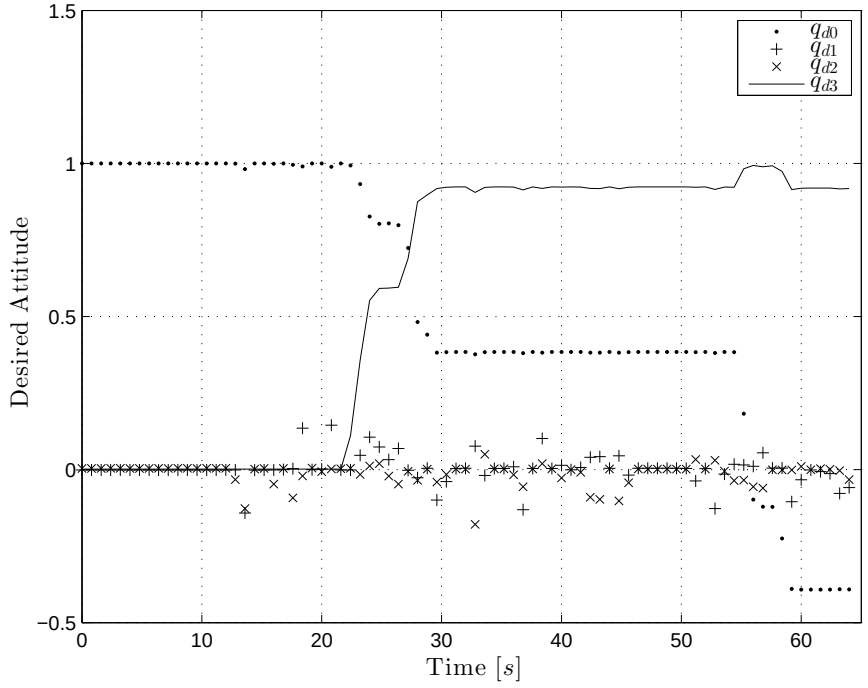

FIGURE 5.6: Quadrotor's attitude reference

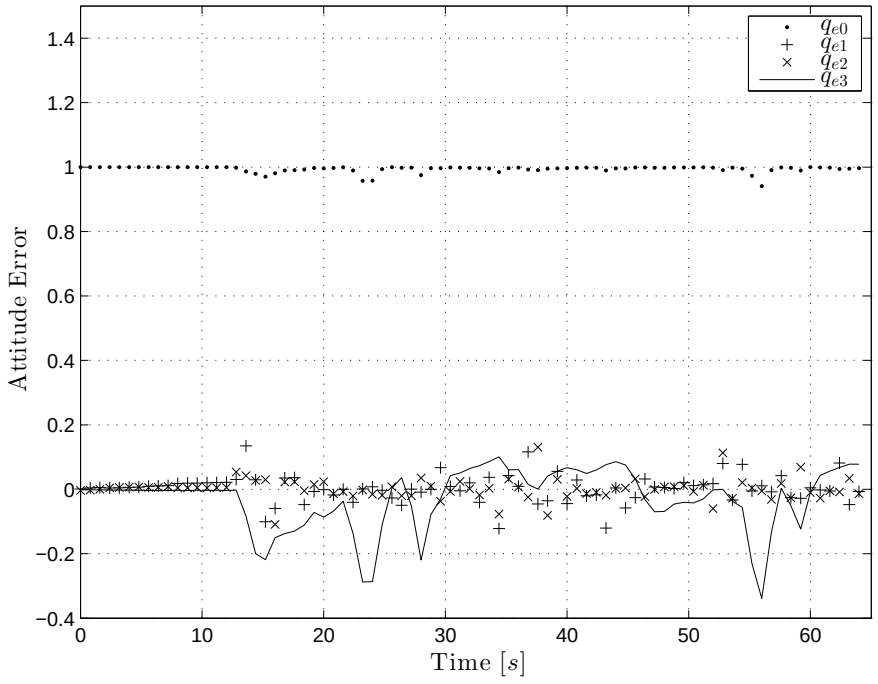

FIGURE 5.7: Quadrotor's attitude error

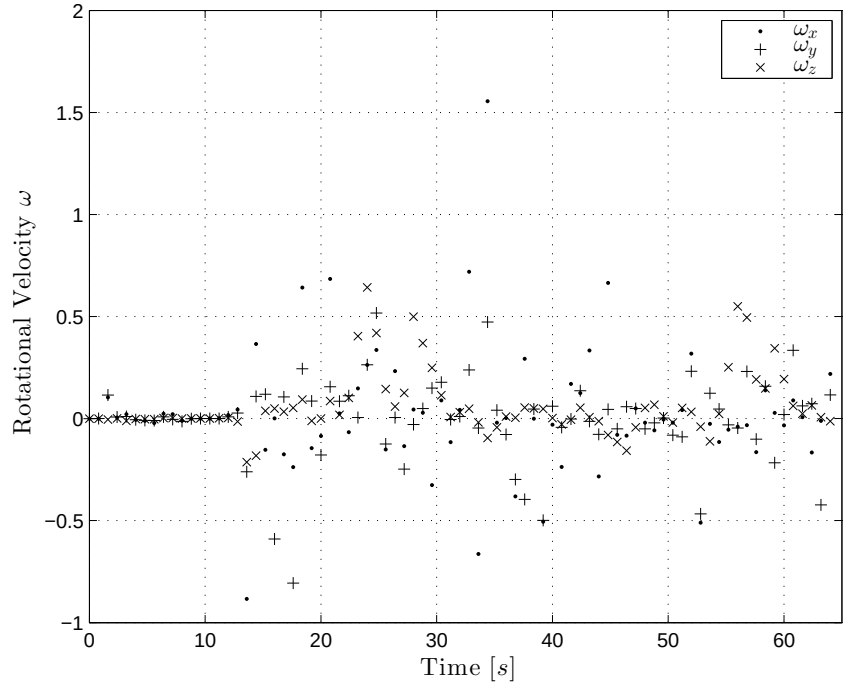

FIGURE 5.8: Quadrotor's angular velocity

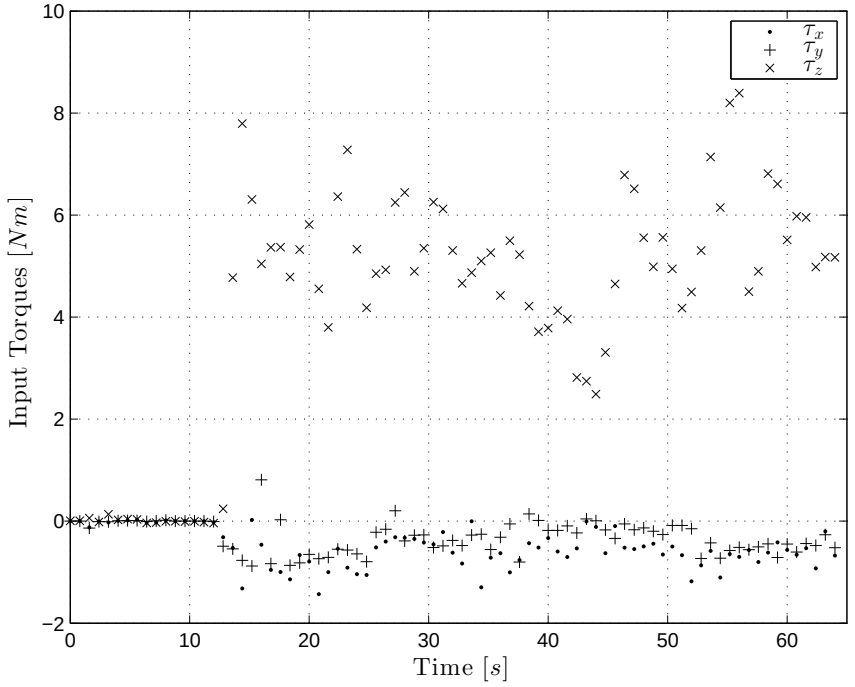

FIGURE 5.9: Quadrotor's input torques

5.2.2 Orientation Experiments With Robotic Arm

In this second experiment the vehicle's atitude, orientation reference (given by the radio controller), attitude error, angular velocity and input torques are shown in Figures 5.10, 5.11, 5.12, 5.13, and 5.15.

In this case the effects of the robotic arm are considered, thus its orientation and torques are shown in Figures 5.14 and 5.15.

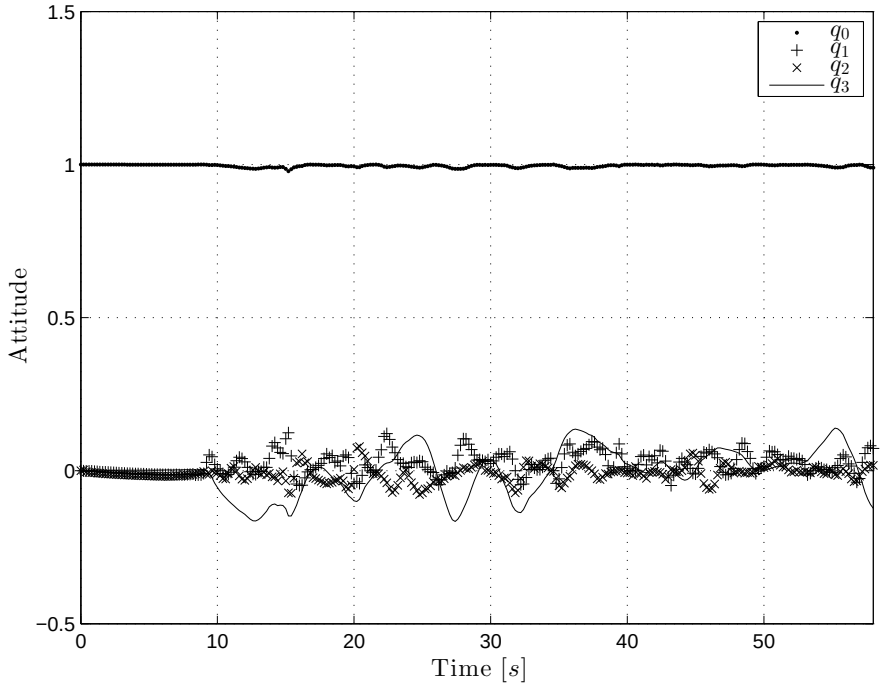

FIGURE 5.10: Quadrotor's attitude

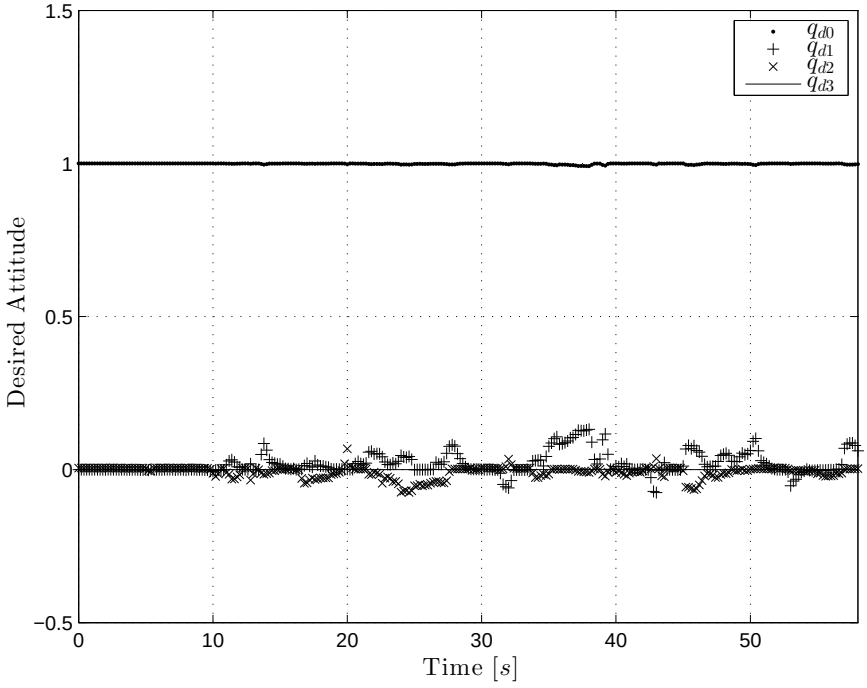

FIGURE 5.11: Quadrotor's attitude reference

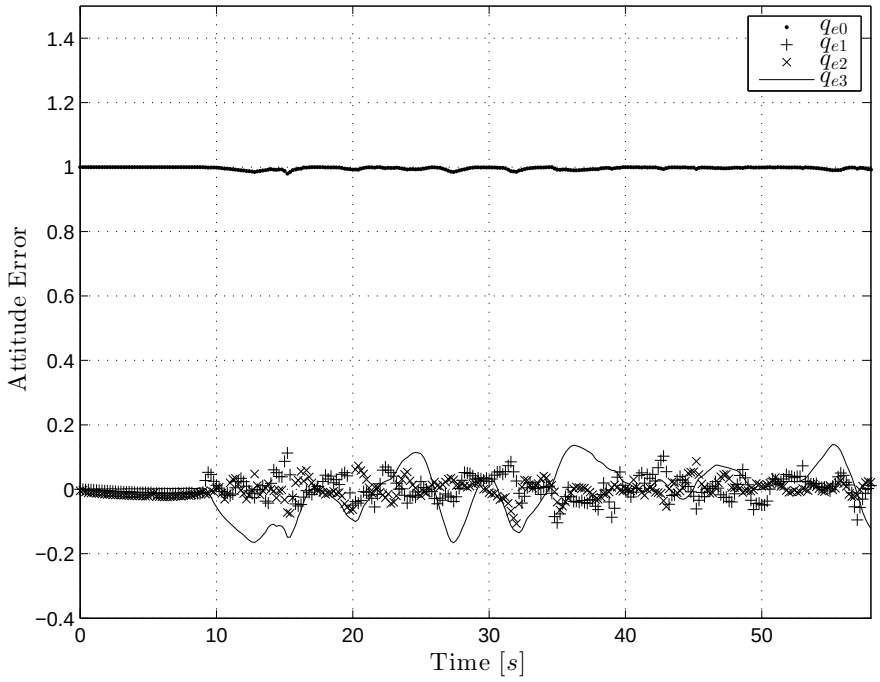

FIGURE 5.12: Quadrotor's attitude error

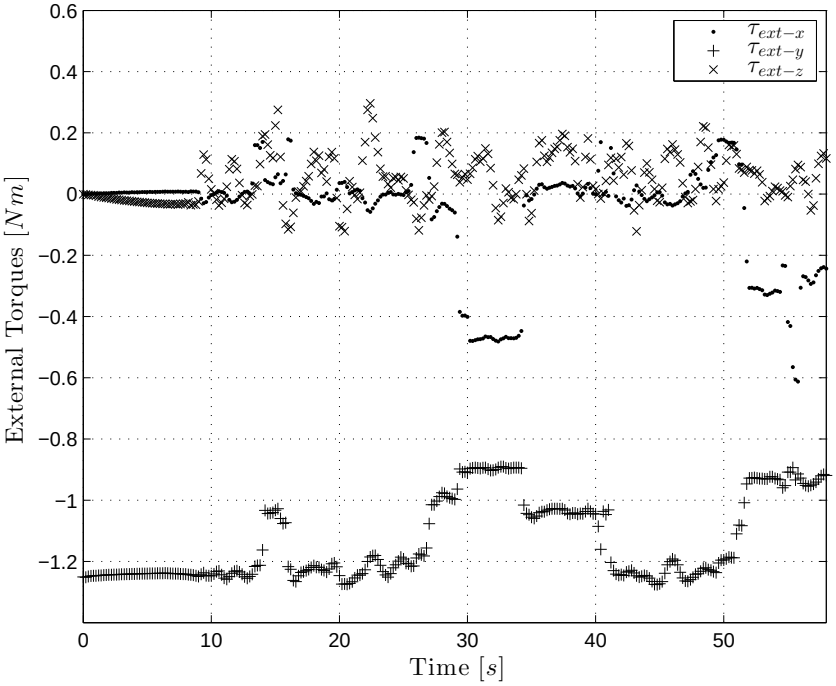

FIGURE 5.13: Robotic arm's torques

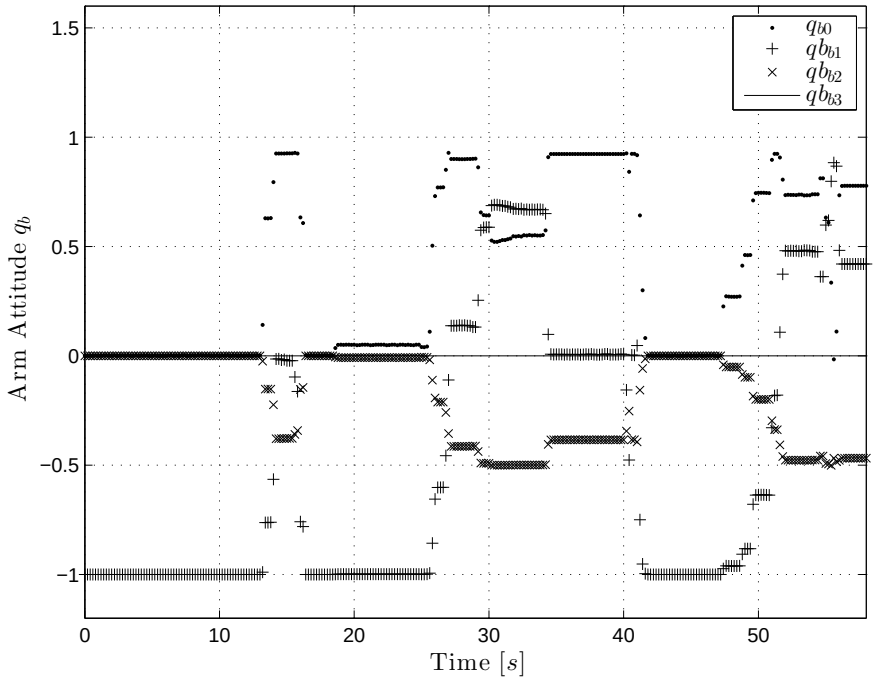

FIGURE 5.14: Robotic arm's orientation

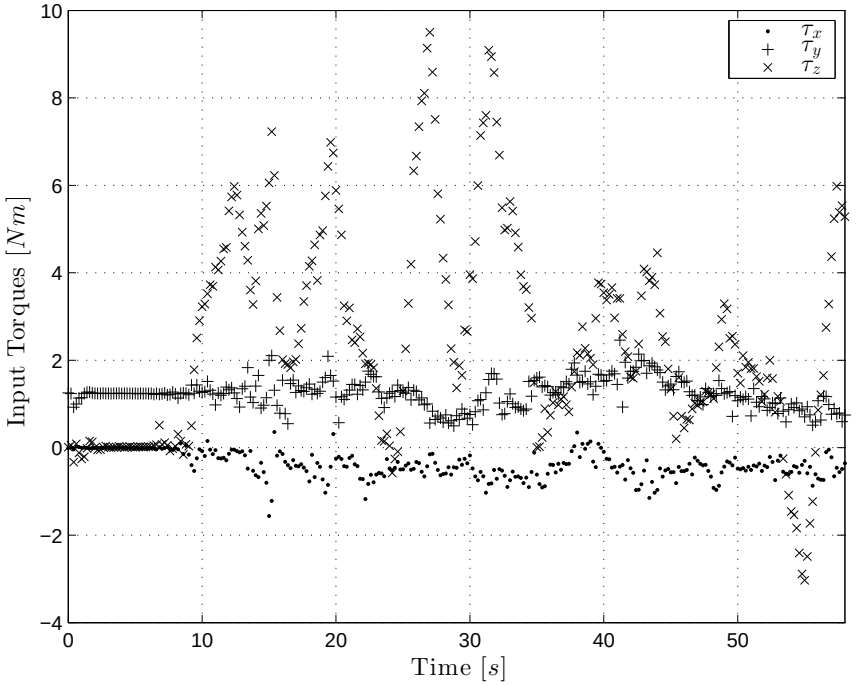

FIGURE 5.15: Quadrotor's external torques

5.2.3 Position Experiments

In the last experiment the global stabilization of the vehicle is achieved by implementing the dual quaternion based control law. The following graphs present the vehicle's position, attitude, orientation trajectory, attitude error, angular velocity and input torques in Figures 5.16 - 5.21.

In this experiment the robotic arm's torques were considered and displayed in Figure 5.22, and the orientation of the end effector is shown in Figure 5.23.

The vehicle was set to stabilize its linear velocity to zero, a better sensing technique is needed to make a feedback in position for the x and w axes such as SLAM algorithms, or Opti-Track cameras. The radio controller was used only to elevate the vehicle to the desired altitude and to start the control law.

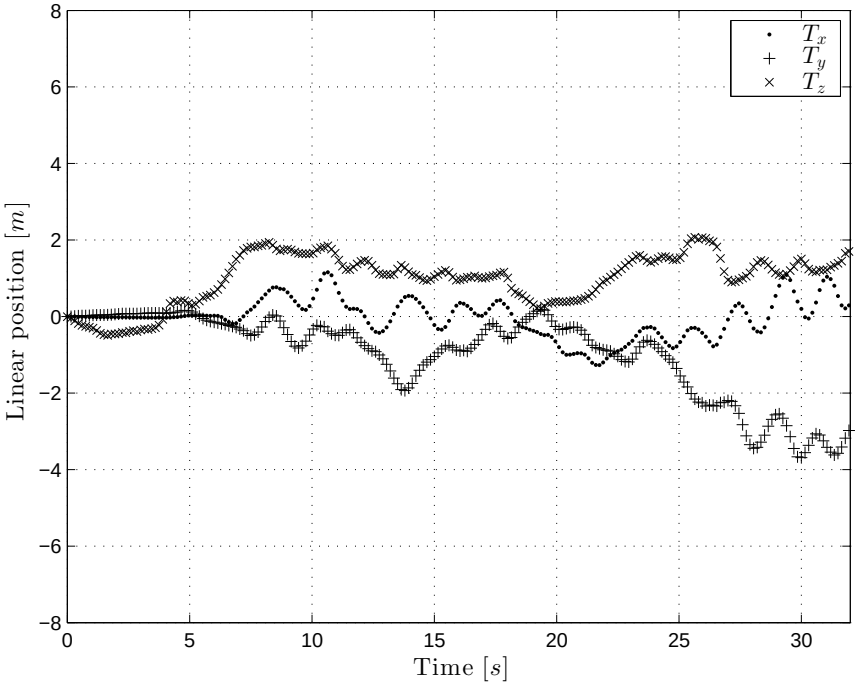

FIGURE 5.16: Quadrotor's position

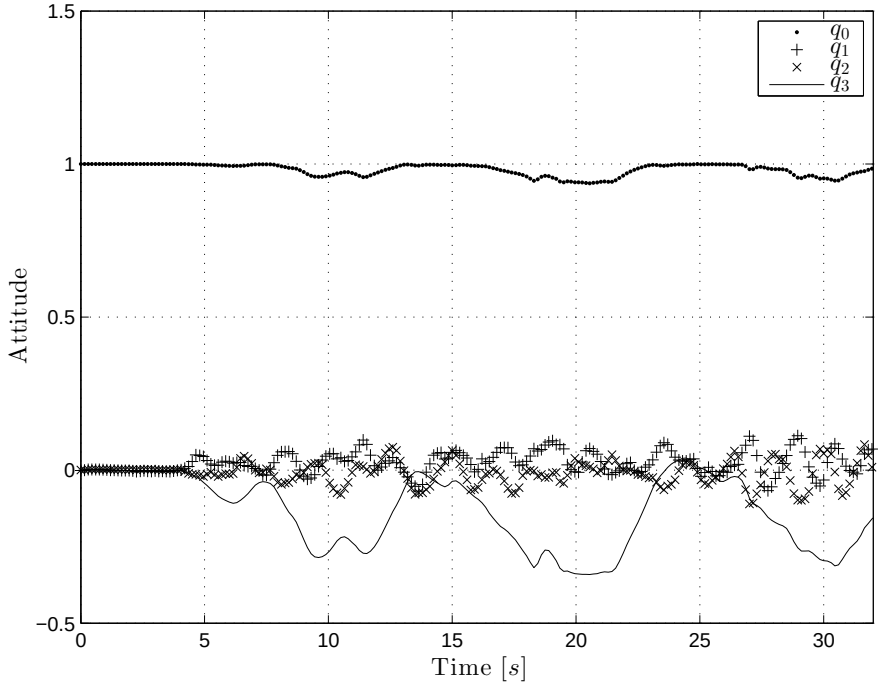

FIGURE 5.17: Quadrotor's orientation

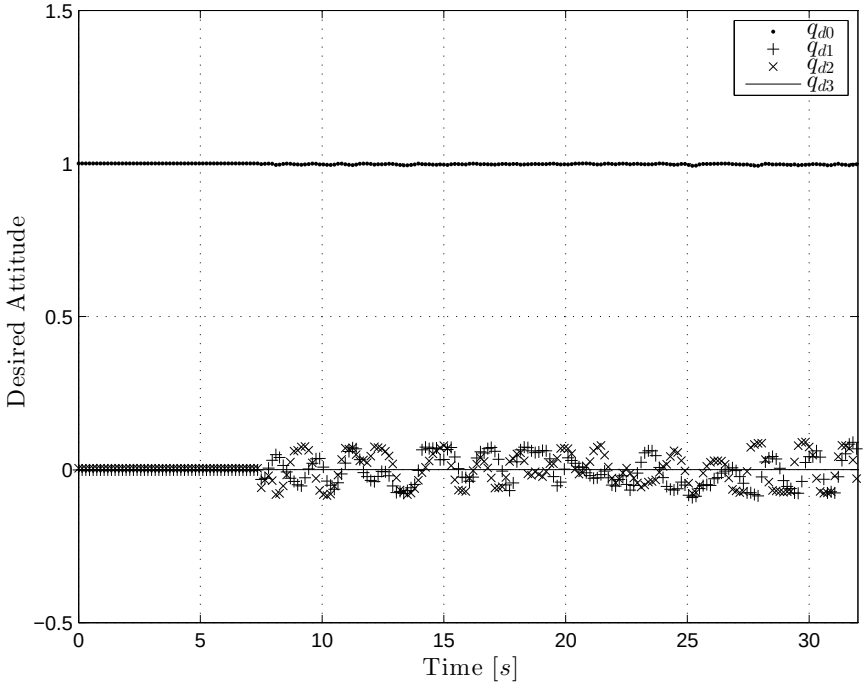

FIGURE 5.18: Quadrotor's orientation trajectory

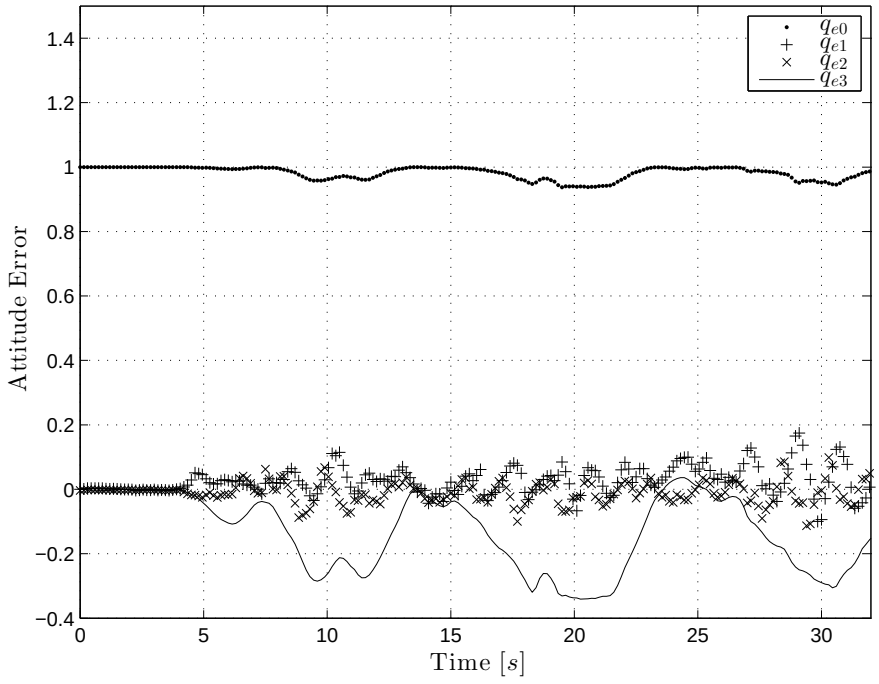

FIGURE 5.19: Quadrotor's orientation error

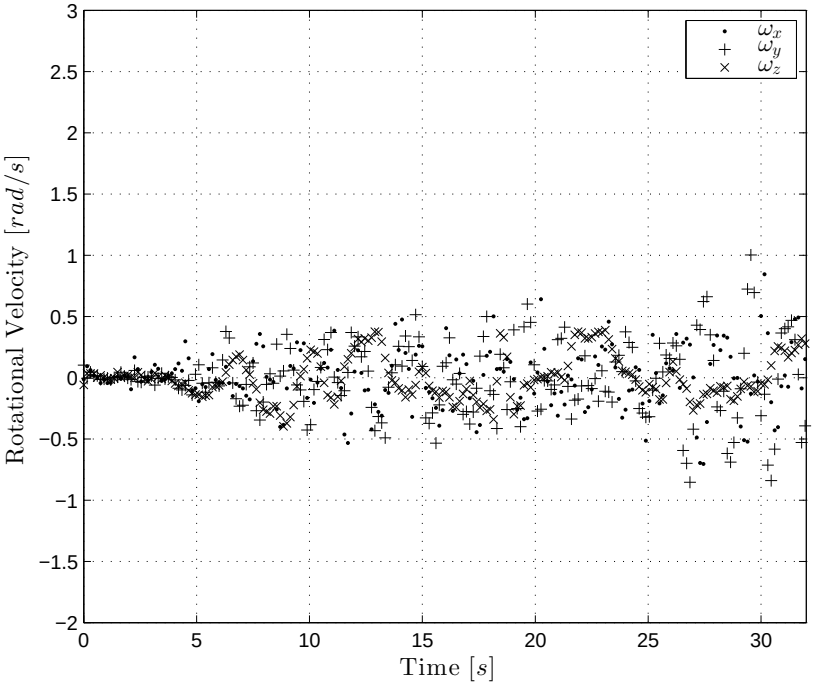

FIGURE 5.20: Quadrotor's rotational velocity

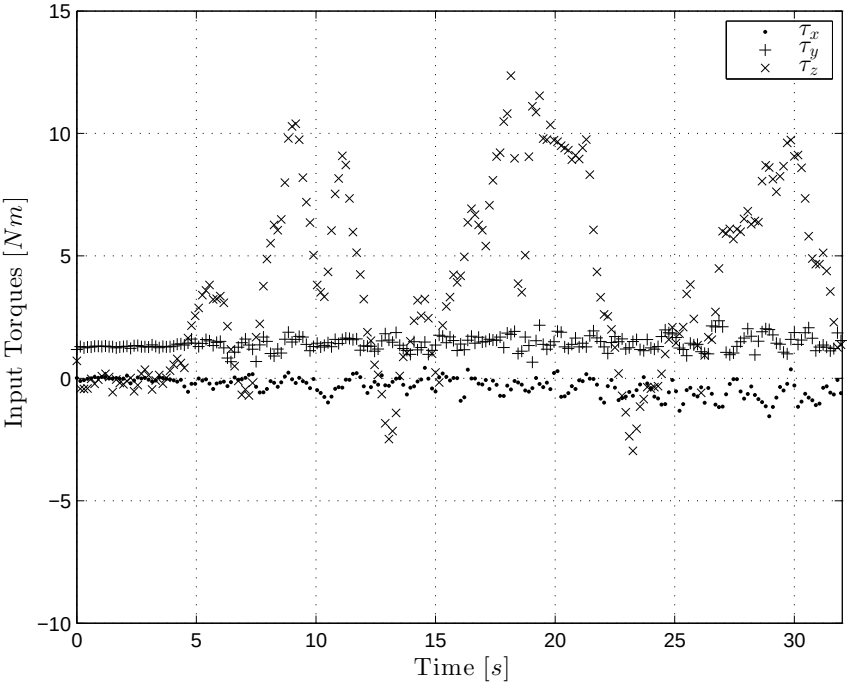

FIGURE 5.21: Quadrotor's input torques

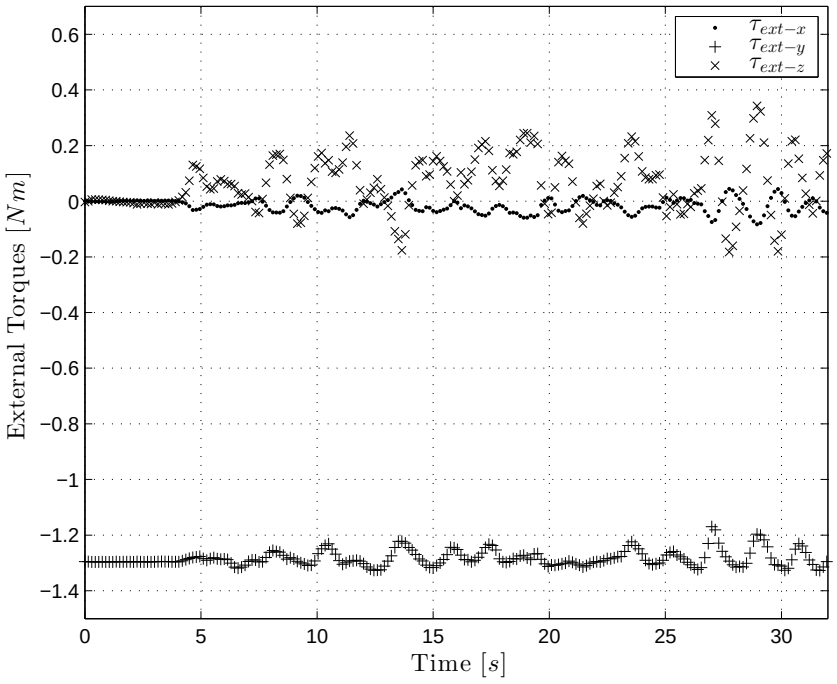

FIGURE 5.22: External torques caused by the robotic arm

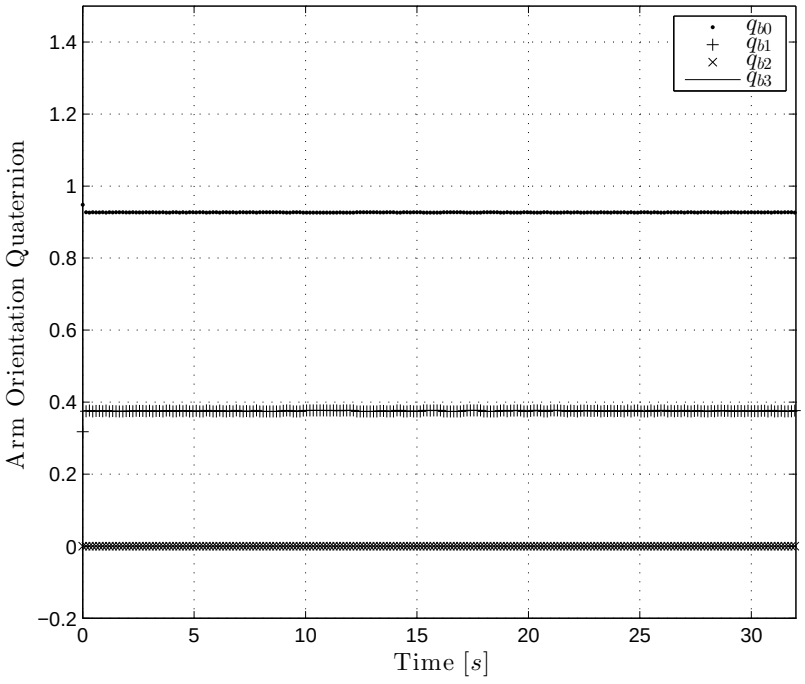

FIGURE 5.23: End effector's orientation

Chapter 6

Conclusions and Future Work

Quadrotor manipulators are platforms that promise many opportunities for scientific research, and applications in civil, commercial, and industrial fields.

This thesis work proposed a model to describe the kinematics and dynamics of a 6 DoF aerial manipulator using a quadrotor UAV and a 2 DoF robotic arm, unit quaternions and dual quaternions were proposed as tools to describe the orientation and position of the vehicle, the arm, and the end effector.

Quaternions turned to be a very useful tool to achieve a mathematically simple model, that can be exactly linearized, and stabilized with simple feedback control laws.

Dual quaternions offered a simple way to express multiple transformations by using only quaternion products and sums, this simplified the calculations for the inverse kinematics required to determine the position and orientation references for the quadrotor and the 2 DoF robotic limb.

The 2 DoF robot arm was sufficient to prove this concept and to compensate the limitations of the quadcopter UAV, but certain applications may require more complex actuators and more articulations. Dual quaternions could simplify the modeling and control of larger robotic limbs, such that the multiple transformation can be expressed as simple products between dual quaternions and vectors.

The proposed control laws were tested in simulations and experiments, they shown a very good performance considering their relative simplicity.

6.1 Future work

Some obstacles were found during the development of this project, more research is needed to obtain a more robust platform. Some opportunities of improvement were found that can lead to more scientific work of our team.

1. Implementation of better measurements of position
    - SLAM techniques to locate the vehicle in the 3D space.
    - Artificial vision to locate a target or an objective
    - Estimator-observer schemes to have better approximations of the position measurements
2. Proposition of adaptable control laws to estimate the change of mass and inertia at the moment of grabbing an object with the gripper.
3. Robust control laws to obtain a better resistance against non considered disturbances.
4. Cooperative algorithms to achieve certain tasks such as complex object manipulation or construction.

Appendix A

Mathematical Proofs

A.1 Property 1.

$$v_1 \times v_2 = \tfrac{1}{2}(v_1 \otimes v_2 - v_2 \otimes v_1) \quad , \quad v_1, v_2 \in \mathbb{R}^3 \tag{A.1}$$

A.1.1 Proof:

Since we know that:

$$v_1 \otimes v_2 = -v_1 \cdot v_2 + v_1 \times v_2 \tag{A.2}$$

$$v_2 \otimes v_1 = -v_2 \cdot v_1 + v_2 \times v_1 \tag{A.3}$$

Thus:

$$v_1 \times v_2 = \frac{1}{2}(-v_1 \cdot v_2 + v_1 \times v_2 + v_2 \cdot v_1 - v_2 \times v_1) \tag{A.4}$$

$$v_1 \times v_2 = \frac{1}{2}(v_1 \times v_2 - v_2 \times v_1) \tag{A.5}$$

It is well known that

$$-\, v_2 \times v_1 = v_1 \times v_2 \tag{A.6}$$

Because of property (A.6), we demonstrate the equality:

$$\frac{1}{2}(v_1 \times v_2 - v_2 \times v_1) = \frac{1}{2}(v_1 \times v_2 + v_1 \times v_2) = v_1 \times v_2 \quad \blacksquare \tag{A.7}$$

Bibliography

A Alaimo, V Artale, C Milazzo, A Ricciardello, and L Trefiletti. Mathematical modeling and control of a hexacopter. In *Unmanned Aircraft Systems (ICUAS), 2013 International Conference on*, pages 1043–1050. IEEE, 2013.

G. Arleo, Caccavale F., G. Muscio, and Pierri F. Control of quadrotor aerial vehicles equipped with a robotic arm. *1st Mediterranean Conference on Control and Automation*, 2013.

Ricardo Campa and Karla Camarillo. Unit quaternions: A mathematical tool for modeling, path planning and control of robot manipulators. *Robot manipulators, M. Ceccarelli (ed.), In-Teh*, pages 21–48, 2008.

Raul Cano. Diseno mecanico de un manipulador aereo de 6 gdl para la construccion de estructuras de barras con uavs. Master's thesis, Universidad de Sevilla, 2013.

Jossué Cariño. Simultaneous localization and mapping on a quadrotor platform. Master's thesis, Centro de Investigación y de Estudios Avanzados del IPN, 2015.

Jossué Cariño, Hernán Abaunza, and Pedro Castillo. Quadrotor quaternion control. *Unmanned Aerial Systems (ICUAS), 2015 IEEE International Conference on*, 2015.

Pedro Castillo, Rogelio Lozano, and Alejandro Dzul. *Modelling and control of mini-flying machines*. Springer Science & Business Media, 2006.

Todd W. Danko and Paul Y. Oh. Design and control of a hyper-redundant manipulator for mobile manipulating unmanned aerial vehicles. *Journal of Intelligent Robotic Systems*, 2014.

Sean Lyttle Nicholas Pagano David J. Cappelleri, Yangbo Long. Design and quaternion-based attitude control of the omnicopter mav using feedback linearization. *Proceedings of the ASME 2012 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, 2012.

FADA-CATEC. ARCAS: Aerial robotics cooperative assembly system, 2011. URL `http://www.arcas-project.eu/`. [Online; accessed 2015-09-15].

Sylvain Durand Nicolas Fermi Guerrero Castellanos, J. J. Tellez-Guzman. Attitude stabilization of a quadrotor by means of event-triggered nonlinear control. *Journal of Intelligent and Robotic Systems*, 73:123–135, 2013.

Emil Fresk and George Nikolakopoulos. Full quaternion based attitude control for a quadrotor. In *Control Conference (ECC), 2013 European*, pages 3864–3869. IEEE, 2013.

L.R. García, E. Rondon, A. Sanchez, A. Dzul, and R. Lozano. Stabilizationd and trajectory tracking of a quad-rotor using vision. *Journal of Intelligent and Robotic Systems*, 2010.

Vaibhav Ghadiok. Autonomous aerial manipulation using a quadrotor. Master's thesis, Utah State University, 2011.

Herbert Goldstein. *Classical mechanics*, volume 4. Pearson Education India, 1962.

James Goppert et al. Python control systems library, 2014. URL `http://sourceforge.net/projects/python-control/`. [Online; accessed 2015-02-05].

Berthold KP Horn. Closed-form solution of absolute orientation using unit quaternions. *JOSA A*, 4(4):629–642, 1987.

John D Hunter. Matplotlib: A 2d graphics environment. *Computing in science and engineering*, 9(3):90–95, 2007.

Carlos Izaguirre-Espinosa. Position–yaw tracking of quadrotors. *Journal of Dynamic Systems, Measurement, and Control*, 137:061011–1, 2015.

ZhiHao Jin and YinXun Wang. Modeling of the quadrotor uav based on screw theory via dual quaternion. *AIAA Modeling and Simulation Technologies (MST) Conference*, 2013.

Eric Jones, Travis Oliphant, Pearu Peterson, et al. SciPy: Open source scientific tools for Python, 2001. URL `http://www.scipy.org/`. [Online; accessed 2015-02-05].

Jose C. V. Junior, Julio C. De Paula, Gideon V. Leandro, and Mario C. Bonfim. Stability control of a quad-rotor using a pid controller. *Brazilian Journal of Instrumentation and Control*, 2013.

Ahmed Khalifa, Mohamed Fanni, Ahmed Ramadan, and Abo-Ismail Ahmed. Modeling and control of a new quadrotor manipulation system. *First International Conference on Innovative Engineering Systems*, 2012.

Ahmed Khalifa, Mohamed Fanni, Ahmed Ramadan, and Abo-Ismail Ahmed. Adaptive intelligent controller design for a new quadrotor manipulation system. *International Conference on Systems, Man and Cybernetics*, 2013.

Ahmed Khalifa, Mohamed Fanni, Ahmed Ramadan, and Abo-Ismail Ahmed. Controller design of a new quadrotor manipulation system based on robust internal-loop compensator. *International Conference on Autonomous Robot Systems and Competitions*, 2015.

Wisama Khalil. Dynamic modeling of robots using recursive newton-euler techniques. *ICINCO2010*, 2010.

Suseong Kim, Seungwon Choi, and H. Jin Kim. Aerial manipulation using a quadrotor with a two dof robotic arm. *International Conference on Intelligent Robots and Systems*, 2013.

Christopher Korpela, Matko Orsag, Todd Danko, Bryan Kobe, Clayton McNeil, Robert Pisch, and Paul Oh. Flight stability in aerial redundant manipulators. *International Conference on Robotics and Automation*, 2012.

Jack B Kuipers. *Quaternions and rotation sequences*, volume 66. Princeton university press Princeton, 1999.

Min-Fan Ricky Lee, Fu Hsin Steven Chiu, and Chen Zhuo. 6 dof manipulator design for maneuvering autonomous aerial mobile robot. In *System Integration (SII), 2013 IEEE/SICE International Symposium on*, pages 173–178. IEEE, 2013.

Yangbo Long, Sean Lyttle, Nicholas Pagano, and David J. Cappelleri. Design and quaternion-based attitude control of the omnicopter mav using feedback linearization. *Proceedings of the ASME 2012 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, 2012.

Lorenz Meier, Petri Tanskanen, Friedrich Fraundorfer, and Marc Pollefeys. Pixhawk: A system for autonomous flight using onboard computer vision. pages 2992–2997, 2011.

Abdelhamid Tayebi Nojan Madinehi. Rigid body attitude estimation: An overview and comparative study. 2013.

Fillipe Nuno and Panagiotis Tsiotras. Simultaneous position and attitude control without linear and angular velocity feedback using dual quaternions. *American Control Conference*, 2013.

Katsuhiko Ogata and Yanjuan Yang. *Modern control engineering*. Prentice-Hall Englewood Cliffs, 1970.

Matko Orsag, Korpela Christopher, and Paul Oh. Modeling and control of mm-uav: Mobile manipulating unmanned aerial vehicle. *Journal of Intelligent Robotic Systems*, 2011.

Elias Reyes-Valeria, Rogerio Enriquez-Caldera, Sergio Camacho-Lara, and Jose Guichard. Lqr control for a quadrotor using unit quaternions: Modeling and simulation. *Electronics, Communications and Computing (CONIELECOMP), 2013 International Conference on*, pages 172–178, 2013.

Jean-Jacques E Slotine, Weiping Li, et al. *Applied nonlinear control*, volume 199. Prentice-Hall Englewood Cliffs, NJ, 1991.

Kerry W Spring. Euler parameters and the use of quaternion algebra in the manipulation of finite rotations: a review. *Mechanism and machine theory*, 21(5):365–373, 1986.

Kimon P Valavanis. *Advances in unmanned aerial vehicles: state of the art and the road to autonomy*, volume 33. Springer Science & Business Media, 2008.

Stefan Van Der Walt, S Chris Colbert, and Gael Varoquaux. The numpy array: a structure for efficient numerical computation. *Computing in Science & Engineering*, 13(2):22–30, 2011.

Xiangke Wang and Changbin Yu. Feedback linearization regulator with coupled attitude and translation dynamics based on unit dual quaternion. In *Intelligent Control (ISIC), 2010 IEEE International Symposium on*, pages 2380–2384. IEEE, 2010.

Li Jie-Wang Jianwen Wang Jian, An Honglei and Ma Hongxu. Backstepping-based inverse optimal attitude control of quadrotor. *International Journal of Advanced Robotic Systems*, 10, 2013.

Hyunsoo Yang and Lee Dongjun. Dynamics and control of quadrotor with robotic manipulator. *International Conference on Robotics and Automation*, 2014.

Yunyun Zhao, Wang Xiangke, and Zhu Huayong. The unit dual quaternion based flight control for a fixed-wing unmanned aerial vehicle. *Proceedings of 2014 IEEE Chinese Guidance, Navigation and Control Conference*, 2014.