# Quaternion Modeling and Control Approaches

H. Abaunza, P. Castillo, and R. Lozano

## Contents


Introduction . . . 2
Quaternion Background . . . 3
  Quaternion Operations . . . 4
Quaternion Dynamic Modeling . . . 6
  Quadrotor Quaternion Dynamical Model . . . 7
  Decoupling the Vehicle's Dynamics . . . 8
  Quaternion Attitude Trajectory . . . 10
Quadrotor Quaternion Control Example . . . 11
  Translational Controller . . . 11
  Attitude Trajectory Strategy . . . 12
  Rotational Controller . . . 12
  Simulation Example . . . 13
Dual Quaternion Background . . . 14
  Operations and Definitions for Dual Quaternions . . . 17
  Dual Quaternion Kinematics . . . 18
  Multiple Dual Quaternion Transformation . . . 20
Quadrotor Dual Quaternion Model . . . 21
Dual Quaternion Quadrotor Control Example . . . 22
  Dual Vector Arrays, Matrices, and Products . . . 22
  Controller Design . . . 23
  Simulation Example . . . 25
Discussion . . . 28
References . . . 29



H. Abaunza (✉) · P. Castillo · R. Lozano
Sorbonne Universités, Université de Technologie de Compiègne, CNRS, Compiègne cedex, France
e-mail: habaunza@hds.utc.fr; castillo@hds.utc.fr; rlozano@hds.utc.fr




K. P. Valavanis, G. J. Vachtsevanos (eds.), *Handbook of Unmanned Aerial Vehicles*,
https://doi.org/10.1007/978-3-319-32193-6_179-1

**Abstract**

Quaternions are an alternative to the classical Euler angles for mathematically describing mechanical systems, including unmanned aerial vehicles (UAVs). In this chapter, the most important concepts of unit and dual quaternions are explained to give a clear basis for readers when working with quaternions. In the first section, unit quaternions are presented to describe a simple yet complete dynamic model for the rotational and translational dynamics of UAVs. In the second section, dual quaternions are explained, which are useful when describing robotic systems with multiple rotations and translations. The dynamic model and a controller example are developed in both sections to illustrate each approach.

**Keywords**

Unmanned aerial vehicles · Multirotors · Quadrotor · Axis-angle rotation · Euler-Rodrigues · Quaternions · Dual quaternions · UAV modeling · Control · Lyapunov

## Introduction

Unmanned aerial vehicles have been increasing in popularity over the last years, every day with more people interested on developing new applications for these robotic systems. Some of them demand more complex capabilities and represent a challenge for the designers of their vehicles.

Most researchers have relied on the classical Euler angles (Goldstein 1962) to represent the dynamics of UAVs. This approach is intuitive and easy to implement, especially when the design and application are simplified, e.g., maintaining small angles, slow movements, or not including mechanically complex components. However, when more arduous designs, tasks, or applications are involved, Euler angle representations encounter some problems, such as complicated nonlinear mathematical expressions, singularities, and gimbal locks.

Quaternions provide an interesting alternative for representing the rotation of rigid bodies and have great advantages compared to Euler angles, i.e., lack of discontinuities and gimbal lock, and provision of mathematical simplicity. When multiple rotations and translations are considered in more complex systems, dual quaternions become very useful to describe the transformation of rigid bodies, also offering mathematical simplicity.

This chapter introduces an elementary basis for understanding and applying unit and dual quaternions for describing dynamic models and designing controllers for UAVs. The quadrotor configuration is used as an example of this approach; however, this methodology can be extended for other types of aircraft. A comprehensive explanation of unit quaternions, their use for describing rotations, their main operations, and their relationships with vectors are provided in section "Quaternion Background". A general quaternion-based dynamic model, which can be applied to describe any rigid body, is developed in section "Quaternion Dynamic Modeling", while in section "Quadrotor Quaternion Control Example", an example of a quadrotor dynamic model and control is introduced. Similarly, the basis of dual quaternions, their operators, and relationships with simultaneous rotations and translations are explained in section "Dual Quaternion Background". A general dynamic model for any rigid body using dual quaternions is described in section "Quadrotor Dual Quaternion Model". In section "Dual Quaternion Quadrotor Control Example", a model that describes a quadrotor is proposed. Finally, some discussions about this work are presented in section "Discussion".

## Quaternion Background

Quaternions were proposed by Hamilton in the nineteenth century as a three-dimensional version of complex numbers (represented as one real and one imaginary part) (Altmann 1989). They are also known as "hypercomplex" numbers (the hypercomplex space is noted as $\mathbb{H}$) since they can be represented as one real plus three imaginary numbers as $\boldsymbol{q} \triangleq q_0 + q_1\hat{i} + q_2\hat{j} + q_3\hat{k}$, where $\hat{i}, \hat{j}, \hat{k} \in \mathbb{I}$, such that $\hat{i}^2 = \hat{j}^2 = \hat{k}^2 = \hat{i}\hat{j}\hat{k} = -1$, and $q_0, \cdots, q_3 \in \mathbb{R}$. Another common representation of a quaternion is using one scalar number and a vector as $\boldsymbol{q} \triangleq q_0 + \overline{q}$, with $\overline{q} = [q_1\ q_2\ q_3]^T$. Due to its mathematical and geometrical advantages, this notation will be used throughout this chapter. In this work, letters with bars over them represent vectors in 3D space $\overline{(.)} \in \mathbb{R}^3$. Since the space of three-dimensional vectors is included in the quaternion space, then vectors can be treated as quaternions with a scalar part equal to zero in all of the quaternion operations.

Considering the rotation illustrated in Fig. 1 as vector $\overline{\alpha} = [\alpha_x\ \alpha_y\ \alpha_z]^T$ with magnitude $\alpha = ||\overline{\alpha}||$ in radians, acting on an axis represented as a unitary vector $\overline{u} = \overline{\alpha}/||\overline{\alpha}||$, the axis-angle representation of this rotation is denoted as

$$\overline{\alpha} = \alpha\overline{u}. \tag{1}$$

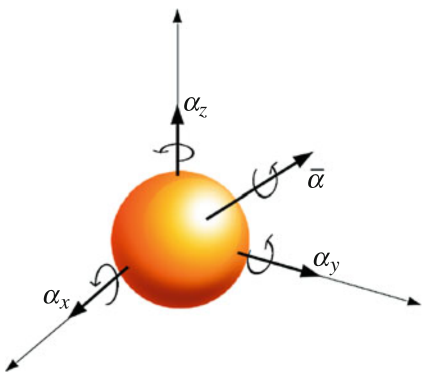


**Fig. 1** Axis-angle representation of a rigid body rotation

It is widely known that a simple rotation, with magnitude $\phi$ in radians, over a plane can be represented using the Euler formula as $e^{\hat{i}\phi} = \cos\phi + \hat{i}\sin\phi$ (Spring 1986). In the nineteenth century, a French banker named Olinde Rodrigues expanded the Euler formula to include three-dimensional rotations using quaternions. This expression, known as the Euler-Rodrigues formula is the exponential mapping of the axis-angle representation of a rotation, defined as

$$\boldsymbol{q} = e^{\frac{1}{2}\alpha\overline{u}} = \cos(\alpha/2) + \overline{u}\sin(\alpha/2), \tag{2}$$

notice that $||\boldsymbol{q}|| = 1$, thus $\boldsymbol{q}$ can be called a *unit quaternion*.

Inversely, the axis-angle representation of a rotation can be derived from a quaternion using the logarithmic mapping

$$\overline{\alpha} = 2\ln\boldsymbol{q}, \tag{3}$$

with

$$\ln\boldsymbol{q} := \begin{cases} [0\ 0\ 0]^T & ,\text{if } ||\overline{q}|| = 0 \\ \dfrac{\overline{q}}{||\overline{q}||}\arccos q_0 & ,\text{if } ||\overline{q}|| \neq 0 \end{cases}. \tag{4}$$

## Quaternion Operations

Because of its significance, historically as well as in the definition of the quaternion space, the main operation of quaternions is the multiplication. Other operations and properties arise from this definition, like the conjugate and the norm. Considering $\boldsymbol{q}, \boldsymbol{r} \in \mathbb{H}$, the quaternion operations are defined as:

### Product

$$\boldsymbol{q} \otimes \boldsymbol{r} := (q_0 r_0 - \overline{q}\cdot\overline{r}) + (q_0\overline{r} + r_0\overline{q} + \overline{q}\times\overline{r}). \tag{5}$$

It is noteworthy that quaternion product is not commutative, which means that $\boldsymbol{q}\otimes\boldsymbol{r} \neq \boldsymbol{r}\otimes\boldsymbol{q}$. This is because of the same noncommutativity property of the cross product used in the definition.

### Sum

The sum of quaternions is simply defined as the sum of each of its elements,

$$\boldsymbol{q} + \boldsymbol{r} := q_0 + r_0 + \overline{q} + \overline{r}. \tag{6}$$

The set of all quaternions with operations addition and multiplication defines a noncommutative division ring. See Kuipers (1999) for more information on this matter.

## Conjugate

The conjugate of a unit quaternion is defined as

$$\boldsymbol{q}^* := q_0 - \overline{q}\,, \tag{7}$$

while the conjugate of a product of quaternions is

$$(\boldsymbol{q} \otimes \boldsymbol{r})^* = \boldsymbol{r}^* \otimes \boldsymbol{q}^*, \tag{8}$$

which can be proved by expanding the corresponding products.

## Norm

The norm of a quaternion is defined by

$$||\boldsymbol{q}||^2 := \boldsymbol{q} \otimes \boldsymbol{q}^* = q_0^2 + q_1^2 + q_2^2 + q_3^2\,. \tag{9}$$

## 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

$$\begin{aligned} &\boldsymbol{q}^{-1} := \frac{\boldsymbol{q}^*}{||\boldsymbol{q}||} \\ &\boldsymbol{q} \otimes \boldsymbol{q}^{-1} = \boldsymbol{q}^{-1} \otimes \boldsymbol{q} = 1 \end{aligned}\,. \tag{10}$$

## Vector Rotation

Considering $\overline{p} \in \mathbb{R}^3$ as a 3D vector in a first reference frame (e.g., the earth coordinates) and $\overline{p}'$ as the same vector but now with respect to a new reference frame (e.g., a vehicle's moving coordinates), then $\overline{p}$ can be transformed into $\overline{p}'$ using a double quaternion product as

$$\overline{p}' = \boldsymbol{q}^{-1} \otimes \overline{p} \otimes \boldsymbol{q} = \boldsymbol{q}^* \otimes \overline{p} \otimes \boldsymbol{q}, \tag{11}$$

where the quaternion $\boldsymbol{q}$ represents the rotation of the second reference frame with respect to the first one. Note this rotation does not affect the vector's magnitude.

## Derivative

The derivative of any unit quaternion can be obtained by differentiating (11) as

$$\dot{\overline{p}}' = \dot{\boldsymbol{q}}^{-1} \otimes \overline{p} \otimes \boldsymbol{q} + \boldsymbol{q}^{-1} \otimes \overline{p} \otimes \dot{\boldsymbol{q}} = \dot{\boldsymbol{q}}^{-1} \otimes \boldsymbol{q} \otimes \overline{p}' + \overline{p}' \otimes \boldsymbol{q}^{-1} \otimes \dot{\boldsymbol{q}}. \tag{12}$$

Since $\boldsymbol{q}$ is an unit quaternion, then $\boldsymbol{q}^{-1} \otimes \boldsymbol{q} = 1$ and $\dot{\boldsymbol{q}}^{-1} \otimes \boldsymbol{q} + \boldsymbol{q}^{-1} \otimes \dot{\boldsymbol{q}}$, which yields

$$\dot{\overline{p}}' = \overline{p}' \otimes \boldsymbol{q}^{-1} \otimes \dot{\boldsymbol{q}} - \boldsymbol{q}^{-1} \otimes \dot{\boldsymbol{q}} \otimes \overline{p}' = 2(\boldsymbol{q}^{-1} \otimes \dot{\boldsymbol{q}}) \times \overline{p}'. \tag{13}$$

Note $\dot{\overline{p}}'$ is the translational velocity of the vector, $\dot{\overline{p}}' = \omega \times \overline{p}'$, where $\omega$ is the rotational velocity of $\overline{p}'$, thus

$$\omega \times \overline{p}' = 2(\boldsymbol{q}^{-1} \otimes \dot{\boldsymbol{q}}) \times \overline{p}', \tag{14}$$

which can be reduced to

$$\omega = 2(\boldsymbol{q}^{-1} \otimes \dot{\boldsymbol{q}}) \Rightarrow \dot{\boldsymbol{q}} = \frac{1}{2}\boldsymbol{q} \otimes \omega. \tag{15}$$

### Axis-Angle Representation and Angular Velocity

An interesting property can be obtained by differentiating the Euler-Rodrigues formula (2) in its exponential expression

$$\dot{\boldsymbol{q}} = \frac{d}{dt} e^{\frac{1}{2}\overline{\alpha}} = \frac{1}{2} e^{\frac{1}{2}\overline{\alpha}} \otimes \dot{\overline{\alpha}} = \frac{1}{2}\boldsymbol{q} \otimes \dot{\overline{\alpha}}. \tag{16}$$

By comparing (15) and (16), it is clearly seen that

$$\dot{\overline{\alpha}} = \omega. \tag{17}$$

### Exponential

The exponential of a quaternion is computed as

$$e^{\boldsymbol{q}} = e^{q_0}\left(\cos ||\overline{q}|| + \frac{\overline{q}}{||\overline{q}||} \sin(||\overline{q}||)\right). \tag{18}$$

## Quaternion Dynamic Modeling

The translational and rotational state of any given rigid body can be expressed as $\mathbf{x} := \begin{bmatrix} \xi & \dot{\xi} & \boldsymbol{q} & \omega \end{bmatrix}^T$ where $\xi \in \mathrm{R}^3$ symbolizes the body's position in the inertial frame, $\dot{\xi}$ its velocity; $\boldsymbol{q} = q_0 + \begin{bmatrix} q_1 & q_2 & q_3 \end{bmatrix}^T$ defines the vehicle orientation with respect to the inertial frame, represented as a unit quaternion; and $\omega = \begin{bmatrix} \omega_x & \omega_y & \omega_z \end{bmatrix}^T$ represents the rotational velocity in the body's frame located on its center of mass. Therefore, following Newton's equations of motion, the dynamic model of any rigid body expressed with unit quaternions is

$$\dot{\mathbf{x}} = \begin{bmatrix} \dot{\xi} \\ \ddot{\xi} \\ \dot{\boldsymbol{q}} \\ \dot{\omega} \end{bmatrix} = \begin{bmatrix} \dot{\xi} \\ m^{-1} F_I \\ \frac{1}{2}\boldsymbol{q} \otimes \omega \\ J^{-1}\left(\tau - \omega \times J\,\omega\right) \end{bmatrix}, \tag{19}$$

where $J$ is the inertia matrix; $\tau$ represents the total torque, both with respect to the body's reference frame; and $F_I$ defines the total force applied to the body in the inertial coordinate system.

Applying a logarithmic mapping to the attitude, the body's state variable is defined as $\mathbf{y} = \left[\xi\ \dot{\xi}\ \overline{\alpha}\ \dot{\overline{\alpha}}\right]^T$; the dynamic model becomes

$$\dot{\mathbf{y}} = \begin{bmatrix} \dot{\xi} \\ \ddot{\xi} \\ \dot{\overline{\alpha}} \\ \ddot{\overline{\alpha}} \end{bmatrix} = \begin{bmatrix} \dot{\xi} \\ m^{-1} F_I \\ \dot{\overline{\alpha}} \\ J^{-1}\left(\tau - \dot{\overline{\alpha}} \times J\,\dot{\overline{\alpha}}\right) \end{bmatrix}. \tag{20}$$

Equations (19) and (20) can be used to describe any mechanical system including aerial vehicles. In the next subsection, the dynamics of a quadrotor will be represented using this approach.

## Quadrotor Quaternion Dynamical Model

The quadrotor is a geometrically simple aerial vehicle. It consists on four parallel rotors with blades attached to a frame in cross configuration. The direction of the rotation of each blade is selected such that all torques on the rotors cancel out in stationary flight.

If the assembly of the mechanical components is considered to be symmetric, and some effects as blade flapping and the misalignment on the motors' axes could be considered small enough, we can assume the forces and torques which act on the vehicle are only the ones illustrated on Fig. 2.

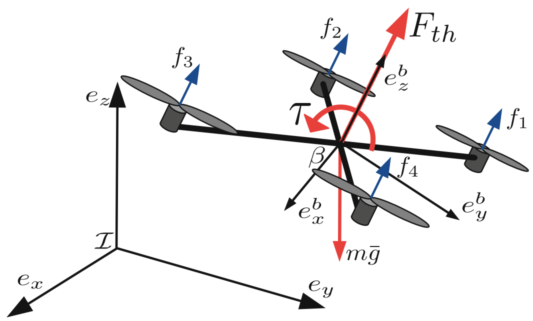


**Fig. 2** Quadrotor free body diagram

Here, two reference frames are considered. $\mathcal{I} = \begin{bmatrix} e_x & e_y & e_z \end{bmatrix}^T$ defines the fixed inertial coordinates, and $\beta = \begin{bmatrix} e_x^b & e_y^b & e_z^b \end{bmatrix}^T$ represents the moving body frame located on the vehicle's center of mass, $F_{th} = \begin{bmatrix} 0 & 0 & \sum_{i=1}^{3} f_i \end{bmatrix}^T$ symbolizes the total thrust force generated by the rotating blades, and the total torque is given by

$$\tau = \begin{bmatrix} l \; (f_1 + f_4 - f_2 - f_3) \\ l \; (f_1 + f_2 - f_3 - f_4) \\ \sum_{i=1}^{4} (-1)^{i+1} \; \tau_i \end{bmatrix}, \tag{21}$$

where $l$ defines the perpendicular distance between any motor and axes, $e_x^b$ or $e_y^b$. Note that both $F_{th}$ and $\tau$ act on $\beta$, but applying a quaternion rotation from (11), it is easy to change their reference frame to $\mathcal{I}$. Thus, the quadrotor's dynamic model is expressed as

$$\dot{\mathbf{x}}_{\text{quad}} = \frac{d}{dt} \begin{bmatrix} \xi \\ \dot{\xi} \\ \boldsymbol{q} \\ \omega \end{bmatrix} = \begin{bmatrix} \dot{\xi} \\ \boldsymbol{q} \otimes \frac{F_{th}}{m} \otimes \boldsymbol{q}^* + \bar{g} \\ \frac{1}{2} \boldsymbol{q} \otimes \omega \\ J^{-1} \; (\tau - \omega \times J \; \omega) \end{bmatrix}, \tag{22}$$

where $\bar{g} = \begin{bmatrix} 0 \; 0 \; g \end{bmatrix}^T$ corresponds to the gravity's vector. Defining $F_{th}^I = \boldsymbol{q} \otimes F_{th} \otimes \boldsymbol{q}^* + m\bar{g}$, the model can be represented using axis-angle notation (Cariño and Abaunza 2015):

$$\dot{\mathbf{x}}'_{\text{quad}} = \frac{d}{dt} \begin{bmatrix} \xi \\ \dot{\xi} \\ \bar{\alpha} \\ \dot{\bar{\alpha}} \end{bmatrix} = \begin{bmatrix} \dot{\xi} \\ m^{-1} F_{th}^I \\ \dot{\bar{\alpha}} \\ J^{-1} \left( \tau - \dot{\bar{\alpha}} \times J \, \dot{\bar{\alpha}} \right) \end{bmatrix}. \tag{23}$$

Note from (22) and (23) that the quadrotor's rotational and translational dynamics are coupled, due to the orientation of $F_{th}^I$ depending on the vehicle's attitude $\boldsymbol{q}$. Nevertheless, using an appropriate approach and some properties of unit quaternions, the quadrotor can be easily stabilized despite its underactuated nature.

## Decoupling the Vehicle's Dynamics

Since the attitude subsystem of the quadrotor is completely actuated, we address it in this subsection.

## Quaternion Rotational Model

From (22), the state vector for the rotational dynamics is

$$\dot{\mathbf{x}}_r = \frac{d}{dt}\begin{bmatrix} \boldsymbol{q} \\ \omega \end{bmatrix} = \begin{bmatrix} \frac{1}{2}\boldsymbol{q} \otimes \omega \\ J^{-1}\,(\tau - \omega \times J\,\omega) \end{bmatrix}, \tag{24}$$

or from (23):

$$\dot{\mathbf{x}}_r' = \begin{bmatrix} \dot{\overline{\alpha}} \\ \ddot{\overline{\alpha}} \end{bmatrix} = \begin{bmatrix} \dot{\overline{\alpha}} \\ J^{-1}\left(\tau - \dot{\overline{\alpha}} \times J\,\dot{\overline{\alpha}}\right) \end{bmatrix}. \tag{25}$$

Choosing an appropriate torque input $\tau = J\tau_u + \dot{\overline{\alpha}} \times J\,\dot{\overline{\alpha}}$, where $\tau_u$ is a virtual control input, (25) can be exactly transformed to a system expressed as

$$\dot{\mathbf{x}}_r' = \begin{bmatrix} \dot{\overline{\alpha}} \\ \tau_u \end{bmatrix}. \tag{26}$$

If (26) is stabilized using any appropriate controller $\tau_u$, then the axis-angle orientation $\overline{\alpha}$ and its angular velocity $\omega$ will converge to zero, which means the quaternion attitude will converge to $\boldsymbol{q}_0 = 1 + [0\ 0\ 0]^T$.

Given a desired attitude trajectory defined by a reference quaternion $\boldsymbol{q}_d$ and its angular velocity $\omega_d$, (24) and (25) can be defined in terms of the error quaternion $\boldsymbol{q}_e \triangleq \boldsymbol{q}_d^* \otimes \boldsymbol{q}$ as

$$\frac{d}{dt}\begin{bmatrix} \boldsymbol{q}_e \\ \omega_e \end{bmatrix} = \begin{bmatrix} \frac{1}{2}\boldsymbol{q}_e \otimes \omega_e \\ J^{-1}\,(\tau - \omega_e \times J\,\omega_e) \end{bmatrix}, \tag{27}$$

with its linear equivalent represented as

$$\dot{\mathbf{e}}_r = \begin{bmatrix} \dot{\overline{\alpha}}_e \\ \tau_u \end{bmatrix}, \tag{28}$$

if $\tau_u$ is correctly designed in terms of the attitude error. Then $\overline{\alpha}_e$ will converge to zero, implying $\boldsymbol{q}_e \to \boldsymbol{q}_0$, and $\boldsymbol{q} \to \boldsymbol{q}_d$.

## Quadrotor Translational Model

From (23), the translational dynamics are given by

$$\dot{\mathbf{x}}_t = \frac{d}{dt}\begin{bmatrix} \xi \\ \dot{\xi} \end{bmatrix} = \begin{bmatrix} \dot{\xi} \\ m^{-1}F_{th}^I \end{bmatrix}. \tag{29}$$

Since (29) is a linear system, an appropriate desired force $F_{th}^I$ can be designed such that $\mathbf{x}_t$ and $\dot{\mathbf{x}}_t$ converge to zero. If a position error is defined as $\xi_e = \xi - \xi_d$, where $\xi_d$ represents a desired position for the UAV, then the translational error dynamics can be written as

$$\dot{\mathbf{e}}_t = \frac{d}{dt}\begin{bmatrix} \xi_e \\ \dot{\xi}_e \end{bmatrix} = \begin{bmatrix} \dot{\xi}_e \\ m^{-1}F^I_{th} \end{bmatrix}. \tag{30}$$

Consequently, if an adequate controller is designed for $F^I_{th}$, the position error will converge to zero, meaning the quadrotor can be stabilized in any desired position.

## Quaternion Attitude Trajectory

Supposing an appropriate $F^I_{th}$ is designed to stabilize the translational subsystem (30), then an attitude trajectory can be computed such that the quadrotor's thrust vector $F_{th}$ is pointed toward the desired control force. This trajectory is derived from the shortest rotation between both vectors and represented with a desired quaternion $\boldsymbol{q}_d$.

Recalling the Euler-Rodrigues formula from (2), $\boldsymbol{q}_d$ is defined as

$$\boldsymbol{q}_d = e^{\frac{1}{2}\alpha_d \bar{u}_d} = \cos(\alpha_d/2) + \bar{u}_d \sin(\alpha_d/2), \tag{31}$$

where $\bar{u}_d$ and $\alpha_d$ denote, respectively, the axis and the angle of the shortest rotation between $F_{th}$ and $F^I_{th}$. Defining $\hat{F}_{th}$ and $\hat{F}^I_{th}$ as the normalized vectors of $\hat{F}_{th}$ and $\hat{F}_{th}$, respectively (note $\hat{F}_{th} = [0\ 0\ 1]^T$ is constant), the cross product between these vectors is defined as

$$\hat{F}_{th} \times \hat{F}^I_{th} = \bar{u}_d \sin(\alpha_d), \tag{32}$$

while the scalar product is given by

$$\hat{F}_{th} \cdot \hat{F}^I_{th} = \cos(\alpha_d). \tag{33}$$

From the definition of the quaternion product, (32) and (33) can be combined as

$$\hat{F}^I_{th} \otimes \hat{F}^*_{th} = -\hat{F}^I_{th} \cdot \hat{F}^*_{th} + \hat{F}^I_{th} \times \hat{F}^*_{th} = \hat{F}_{th} \cdot \hat{F}^I_{th} + \hat{F}_{th} \times \hat{F}^I_{th} = \cos(\alpha_d) + \bar{u}_d \sin(\alpha_d)\,. \tag{34}$$

Since (34) expresses twice the desired rotation needed in (31), some exponential and logarithmic properties are then applied, thus resulting in

$$\boldsymbol{q}_d = e^{\dfrac{\ln(\hat{F}^I_{th} \otimes \hat{F}^*_{th})}{2}}. \tag{35}$$

Note since $\hat{F}_{th}$ only acts in the vertical axis of the quadrotor, then $\boldsymbol{q}_d$ will only compute rotations around the $xy$ plane of the inertial frame. Considering $\psi_d$ as a desired rotation around the $z$ axis of the vehicle's body frame, the desired quaternion can be enhanced as

$$\boldsymbol{q}_d = e^{\frac{\ln(\hat{F}_{th}^I \otimes \hat{F}_{th}^*)}{2}} \otimes \boldsymbol{q}_z = e^{\frac{\ln(\hat{F}_{th}^I \otimes \hat{F}_{th}^*)}{2}} \otimes e^{\frac{[0\ 0\ \psi_d]^T}{2}}. \tag{36}$$

## Quadrotor Quaternion Control Example

In this section, a quaternion controller is designed as an example using Lyapunov theory based on the proposed model from section

In this section, a quaternion controller is designed as an example using Lyapunov theory theory based on the proposed model from section "Quaternion Dynamic Modeling", a similar controller has been presented in (Abaunza et al. 2017a).

### Translational Controller

First, from (30), the linear translational subsystem can be written as

$$\dot{\mathbf{e}}_t = \begin{bmatrix} 0^{3\times3} & I^{3\times3} \\ 0^{3\times3} & 0^{3\times3} \end{bmatrix} \begin{bmatrix} \xi_e \\ \dot{\xi}_e \end{bmatrix} + \begin{bmatrix} 0^{3\times3} \\ m^{-1} I^{3\times3} \end{bmatrix} F_{th}^I = A_t \mathbf{e}_t + B_t F_{th}^I. \tag{37}$$

Propose the following positive-definite function:

$$V_t = \frac{1}{2} \mathbf{e}_t \cdot \mathbf{e}_t, \tag{38}$$

where its derivative is given by

$$\dot{V}_t = \mathbf{e}_t \cdot \dot{\mathbf{e}}_t = \mathbf{e}_t \cdot \left[ A_t \mathbf{e}_t + B_t F_{th}^I \right]. \tag{39}$$

Proposing

$$F_{th}^I = K_t \mathbf{e}_t, \tag{40}$$

where $K_t$ contains the control gains and is defined as

$$K_t = \begin{bmatrix} K_{ptx} & 0 & 0 & K_{dtx} & 0 & 0 \\ 0 & K_{pty} & 0 & 0 & K_{dty} & 0 \\ 0 & 0 & K_{ptz} & 0 & 0 & K_{dtz} \end{bmatrix}, \tag{41}$$

Equation (39) can be rewritten as

$$\dot{V}_t = \mathbf{e}_t \cdot [A_t + B_t K_t] \, \mathbf{e}_t. \tag{42}$$

In order to asymptotically stabilize the translational subsystem, the gains from $K_t$ must be chosen such that the real part of the complex eigenvalues of $(A_t + B_t K_t)$ is negative, implying the following inequalities must be met:

$$\left.\begin{array}{r} \mathrm{re}\big(K_{dtx} + (K_{dtx}^2 + 4K_{ptx}m)^{1/2}\big)/(2m) \\ \mathrm{re}\big(K_{dty} + (K_{dty}^2 + 4K_{pty}m)^{1/2}\big)/(2m) \\ \mathrm{re}\big(K_{dtz} + (K_{dtz}^2 + 4K_{ptz}m)^{1/2}\big)/(2m) \\ \mathrm{re}\big(K_{dtx} - (K_{dtx}^2 + 4K_{ptx}m)^{1/2}\big)/(2m) \\ \mathrm{re}\big(K_{dty} - (K_{dty}^2 + 4K_{pty}m)^{1/2}\big)/(2m) \\ \mathrm{re}\big(K_{dtz} - (K_{dtz}^2 + 4K_{ptz}m)^{1/2}\big)/(2m) \end{array}\right\} < 0. \tag{43}$$

If the above inequalities are satisfied, then asymptotic stability is ensured for system (37) since

$$V_t > 0, \quad \dot{V}_t < 0 \quad \forall \quad \mathbf{e} \neq 0. \tag{44}$$

## Attitude Trajectory Strategy

Note the controller proposed in section "Translational Controller" results in a force $F_{th}^I$ that may be pointing at any direction in a 3D space. Since the quadrotor's thrust vector $F_{th}$ always points in the body's vertical axis, a reference quaternion $\boldsymbol{q}_d$ must be used such that $\boldsymbol{q} \rightarrow \boldsymbol{q}_d \Rightarrow \boldsymbol{q} \otimes \frac{F_{th}}{m} \otimes \boldsymbol{q}^* \rightarrow F_{th}^I$. This is computed by rewriting (36) as

$$\boldsymbol{q}_d = e^{\dfrac{\ln\left(F_{th}^I \otimes (-[0\ 0\ 1]^T)\right)}{2}} \otimes \boldsymbol{q}_z = e^{\dfrac{\ln\left(F_{th}^I \otimes (-[0\ 0\ 1]^T)\right)}{2}} \otimes e^{\dfrac{[0\ 0\ \psi_d]^T}{2}}, \tag{45}$$

where $\psi_d$ is an input, given manually to the quadrotor.

## Rotational Controller

Considering $\boldsymbol{q}_e \triangleq \boldsymbol{q}_d^* \otimes \boldsymbol{q}$ and $\overline{\alpha}_e = 2\ln(\boldsymbol{q}_e)$, $\dot{\overline{\alpha}}_e = \omega - 2\frac{d}{dt}\ln(\boldsymbol{q}_d)$, the same methodology from section "Translational Controller" is now applied to the rotational error model in its axis-angle representation from (28).

$$\dot{\mathbf{e}}_r = \frac{d}{dt}\begin{bmatrix} \overline{\alpha}_e \\ \dot{\overline{\alpha}}_e \end{bmatrix} = \begin{bmatrix} 0^{3\times3} & I^{3\times3} \\ 0^{3\times3} & 0^{3\times3} \end{bmatrix}\mathbf{e}_r + \begin{bmatrix} 0^{3\times3} \\ m^{-1}I^{3\times3} \end{bmatrix}\tau_u = A_r\mathbf{e}_r + B_r\tau_u. \tag{46}$$

Proposing a positive-definite function with its derivative as

$$V_r = \frac{1}{2}\mathbf{e}_r \cdot \mathbf{e}_r \;\rightarrow\; \dot{V}_r = \mathbf{e}_r \cdot [A_r\mathbf{e}_r + B_r\tau_u], \tag{47}$$

The controller can be defined as

$$\tau_u = K_r \mathbf{e}_r \;\Rightarrow\; \tau = J K_r \mathbf{e}_r + \dot{\bar{\alpha}}_e \times J\, \dot{\bar{\alpha}}_e, \tag{48}$$

with control gains given by

$$K_r = \begin{bmatrix} K_{prx} & 0 & 0 & K_{drx} & 0 & 0 \\ 0 & K_{pry} & 0 & 0 & K_{dry} & 0 \\ 0 & 0 & K_{prz} & 0 & 0 & K_{drz} \end{bmatrix}, \tag{49}$$

such that

$$V_r > 0, \quad \dot{V}_r < 0 \;\; \forall \;\; \mathbf{e}_r \neq 0. \tag{50}$$

Therefore, asymptotic stability for the rotational subsystem is ensured as long as the following conditions are met:

$$\left.\begin{array}{r} \mathrm{re}\big(K_{drx} + (K_{drx}^2 + 4K_{prx})^{1/2}\big)/2 \\ \mathrm{re}\big(K_{dry} + (K_{dry}^2 + 4K_{pry})^{1/2}\big)/2 \\ \mathrm{re}\big(K_{drz} + (K_{drz}^2 + 4K_{prz})^{1/2}\big)/2 \\ \mathrm{re}\big(K_{drx} - (K_{drx}^2 + 4K_{prx})^{1/2}\big)/2 \\ \mathrm{re}\big(K_{dry} - (K_{dry}^2 + 4K_{pry})^{1/2}\big)/2 \\ \mathrm{re}\big(K_{drz} - (K_{drz}^2 + 4K_{prz})^{1/2}\big)/2 \end{array}\right\} < 0. \tag{51}$$

## Simulation Example

Numerical simulations are useful to validate algorithms and to better understand system dynamics. Here, numerical simulations of the model presented in (22) and the controllers from (40) and (48) are illustrated. The dynamic model was coded using MATLAB, and a simple numerical integration technique based on Riemann's sums can be used to compute the system's translational and rotational state variables.

The model parameters were taken from a custom-made quadrotor and estimated using computer-assisted design and finite element techniques such that

$$m = 1.3\,\mathrm{kg}\;,\; J = \begin{bmatrix} 0.177 & 0 & 0 \\ 0 & 0.177 & 0 \\ 0 & 0 & 0.354 \end{bmatrix} \mathrm{kg\,m^2}\,. \tag{52}$$

For the controllers, their gains were empirically selected as

$$K_t = -\begin{bmatrix} 8 & 0 & 0 & 4 & 0 & 0 \\ 0 & 8 & 0 & 0 & 4 & 0 \\ 0 & 0 & 4 & 0 & 0 & 4 \end{bmatrix}, \quad K_r = -\begin{bmatrix} 50 & 0 & 0 & 10 & 0 & 0 \\ 0 & 50 & 0 & 0 & 10 & 0 \\ 0 & 0 & 50 & 0 & 0 & 10 \end{bmatrix}, \tag{53}$$

such that the conditions from (43) and (51) are met.

A flight scenario is considered in which the quadrotor takes off from the origin to an altitude of 1m, then a $r = 2$ m circular path is followed, while a rotation around the vehicle's $z$ axis is tracked such that the quadrotor's front is always pointed toward the center of the circle. This is achieved by designing a circular path $\xi_d$ as

$$\xi_d = \begin{cases} [0\ 0\ 1]^T & \forall\ t < 5\text{s} \\ \begin{bmatrix} -r\cos(t-5) + r \\ -r\sin(t-5) \\ 0 \end{bmatrix} & \forall\ t > 5\text{s} \end{cases}, \tag{54}$$

and a complement to $\boldsymbol{q}_z$ from (45) by adapting (35) such that

$$\delta_{xy} = [1\ 0\ 0]^T - \xi \quad , \quad \boldsymbol{q}_z = e^{\dfrac{\ln\left(\delta_{xy} \otimes (-[1\ 0\ 0]^T)\right)}{2}}. \tag{55}$$

The controllers from (40) and (48) compute the forces and torques required to stabilize the vehicle in the desired references, shown in Fig. 3.

The desired position is tracked by the effects of the control force as depicted on Fig. 4. The quadrotor's front is symbolized by an arrow in Fig. 4d. Note its direction is pointed toward the center of the circle.

The trajectory computed by (45) and (55) is represented in Fig. 5. It is important to remark that, although the rotation of the vehicle completes more than one complete tour around the $z$ axis, the attitude is continuous. This lack of discontinuity points is one of the main advantages of using quaternion approaches.

Finally, to better illustrate the vehicle's rotational behavior, the axis-angle representation of the quadrotor's attitude, reference, and error are illustrated on Fig. 6 for each axis. Note the axis-angle representation uncovers the discontinuities that are present when completing a rotation of $2\pi$ [rad].

## Dual Quaternion Background

Sections "Quaternion Background", "Quaternion Dynamic Modeling", and "Quadrotor Quaternion Control Example" separate rotational and translational dynamics on the proposed models and controllers. This is very common when dealing with relatively simple systems such as quadrotors and other similar types of aircraft, but this might complicate things when multiple rotations and translations are involved.

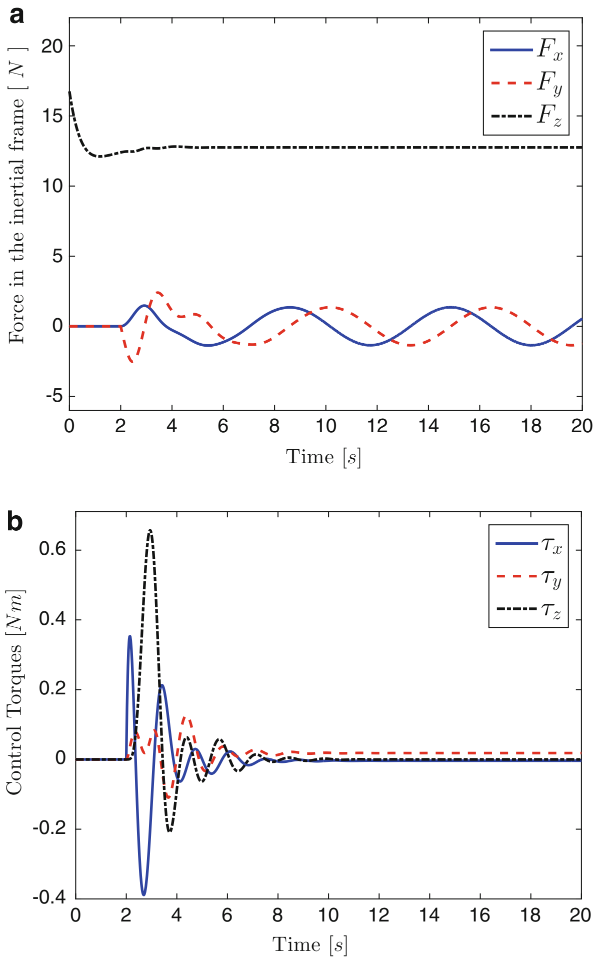


**Fig. 3** Quadrotor unit quaternion control inputs. (**a**) Translational controller. (**b**) Rotational controller

Such case could occur, for example, in digital animations, multi-agent robotic systems, and autonomous aerial manipulators, see (Abaunza et al. 2017b).

Over many years, researchers came with the idea of expanding the concept of the quaternion to include more complex geometrical transformations. Such ideas started developing in the realm of non-Euclidean geometry, analysis, and topology. In 1873, an English geometer named William Kingdon Clifford made a first sketch of an expansion of Hamilton's quaternions, by adding a second quaternion which is multiplied by a new imaginary term; he gave this concept the name of *biquaternion* (Lifford 1873; Buchheim 1885). Later on, Alexander McAulay introduced in 1898 a *nullipotent term* $\epsilon$ to the biquaternion (McAulay 1898). This term can be defined

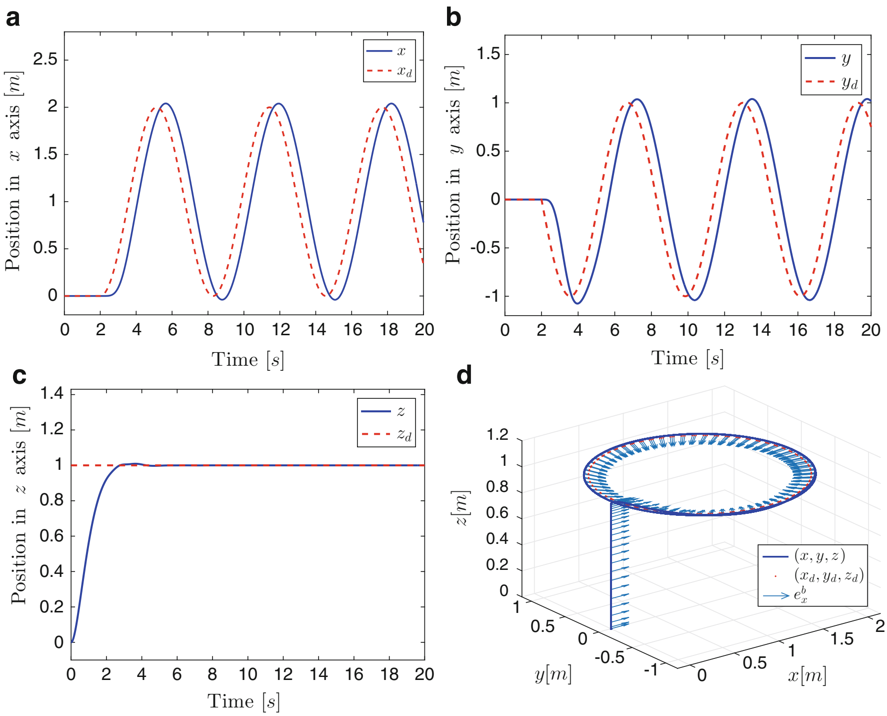


**Fig. 4** Quadrotor position tracking using a unit quaternion controller. (**a**) Quadrotor's $x$ axis position and reference. (**b**) Quadrotor's $y$ axis position and reference. (**c**) Quadrotor's altitude regulation. (**d**) Vehicle's translation and orientation tracking

in any way, as long as it respects the property:

$$\epsilon \neq 0 \ \mid \ \epsilon^2 = 0; \tag{56}$$

with this component, McAulay developed dual quaternion algebra, using the term of *octonions*. Finally, in 1985, Aleksandr Kotelnikov studied the applications of octonions and biquaternions on kinematics (Kotelnikov 1895), concluding that the properties of unit quaternions are also valid for those numbers; hence, they were eventually known as *Dual Quaternions*, which are defined as

$$\hat{q} = \boldsymbol{q}_R + \boldsymbol{q}_D \epsilon, \quad \epsilon \neq 0, \epsilon^2 = 0, \tag{57}$$

where $\boldsymbol{q}_R$, $\boldsymbol{q}_D \in \mathbb{H}$ are known as the real and dual parts of $\hat{q}$, respectively.

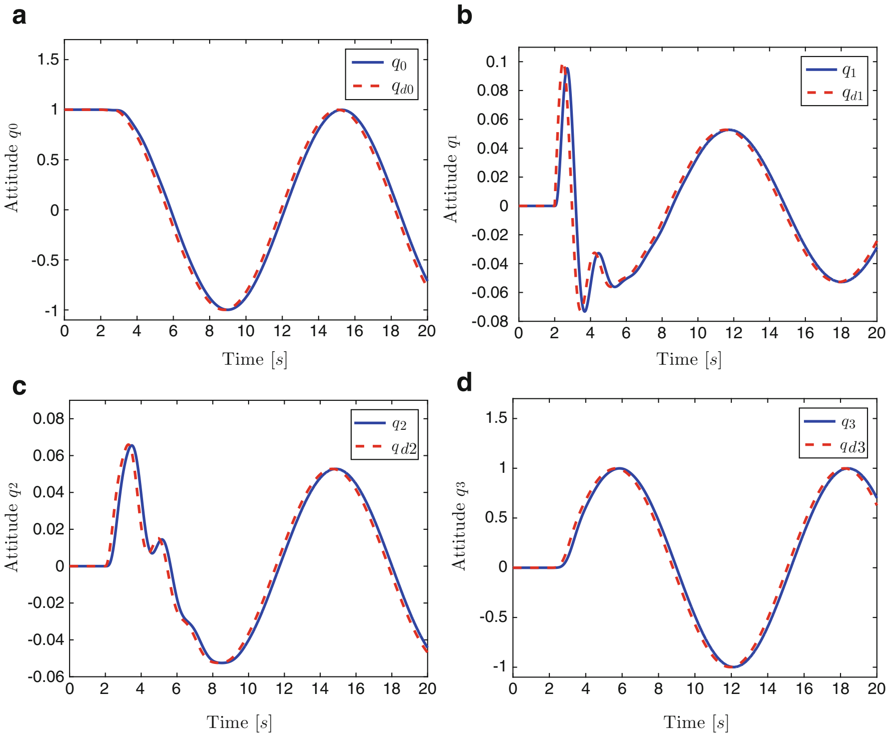


**Fig. 5** Quadrotor attitude trajectory, tracked by a unit quaternion controller

## Operations and Definitions for Dual Quaternions

### Sum

Let $\hat{q}_1$ and $\hat{q}_2$ be dual quaternions, then $\hat{q}_1 + \hat{q}_2 = \boldsymbol{q}_{1R} + \boldsymbol{q}_{2R} + [\boldsymbol{q}_{1D} + \boldsymbol{q}_{2D}]\,\epsilon$.

### Product

The multiplication between dual quaternions is defined as $\hat{q}_1 \otimes \hat{q}_2 = \boldsymbol{q}_{1R} \otimes \boldsymbol{q}_{2R} + [\boldsymbol{q}_{1R} \otimes \boldsymbol{q}_{2D} + \boldsymbol{q}_{1D} \otimes \boldsymbol{q}_{2R}]\,\epsilon$.

### Norm

The norm of a dual quaternion is represented as $||\hat{q}||^2 = \hat{q} \otimes \hat{q}^*$. 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 written as $\hat{q}^* = \boldsymbol{q}_R^* + \boldsymbol{q}_D^*\epsilon$. Since in this work we are dealing only with unit dual quaternions, $\hat{q}^* = \hat{q}^{-1}$.

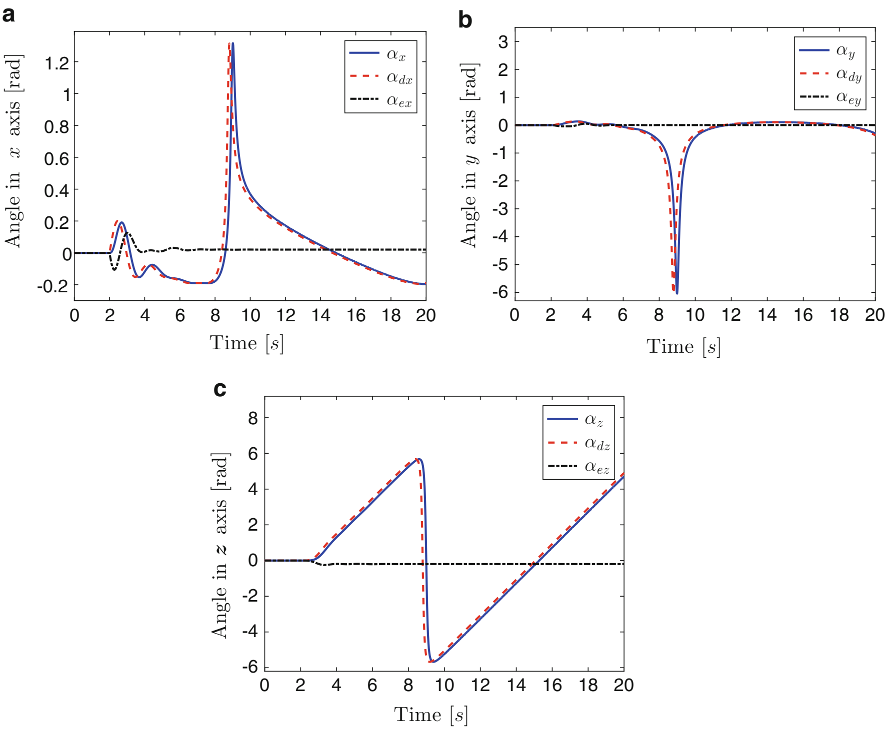


**Fig. 6** Quadrotor attitude in axis-angle representation, controlled by a unit quaternion algorithm. (**a**) Quadrotor's angle and error in $x$ axis. (**b**) Quadrotor's angle and error in $y$ axis. (**c**) Quadrotor's angle and error in $z$ axis

## 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.

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 $\hat{\bar{k}} \cdot \hat{\bar{v}} = K_R\bar{v}_R + K_D\bar{v}_D\epsilon$, where $K_R$ and $K_D$ are $3 \times 3$ diagonal matrices with entries $k_{r1}, k_{r2}, k_{r3}$ and $k_{d1}, k_{d2}, k_{d3}$, respectively.

## Dual Quaternion Kinematics

Supposing a rigid body is subject to a translation $\xi$ expressed in the inertial reference frame (or $\xi_\beta \in \mathbb{R}^3$ if its represented with respect to the body's reference frame), followed by a rotation represented by a quaternion $\boldsymbol{q}$, then its dual quaternion transformation can be expressed as

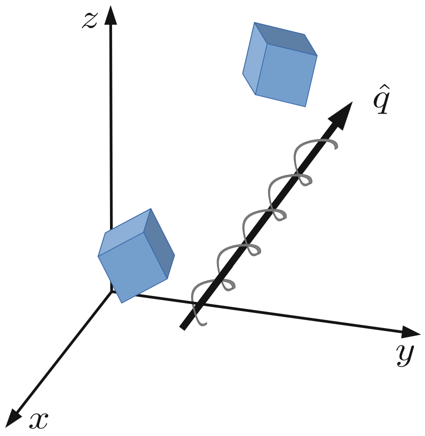


**Fig. 7** Simultaneous rotation and translation of a rigid body

$$\hat{q} = \boldsymbol{q} + \frac{1}{2}\xi \otimes \boldsymbol{q}\epsilon = \boldsymbol{q} + \frac{1}{2}\boldsymbol{q} \otimes \xi_\beta \epsilon, \tag{58}$$

where $\otimes$ is a quaternion product. Since $||\boldsymbol{q}|| = 1$, then $\hat{q}$ is considered to be a unit dual quaternion.

The derivate of the previous equation is obtained by differentiating (58) as

$$\begin{aligned}
\dot{\hat{q}} &= \dot{\boldsymbol{q}} + \frac{1}{2}\left[\dot{\boldsymbol{q}} \otimes \xi_\beta + \boldsymbol{q} \otimes \dot{\xi}_\beta\right]\epsilon \\
&= \frac{1}{2}\boldsymbol{q} \otimes \omega + \left[\frac{1}{4}\boldsymbol{q} \otimes \omega \otimes \xi_\beta + \frac{1}{2}\boldsymbol{q} \otimes \dot{\xi}_\beta\right]\epsilon \\
&= \frac{1}{2}\boldsymbol{q} \otimes \omega + \left[\frac{1}{2}\boldsymbol{q} \otimes (\omega \times \xi_\beta) + \frac{1}{4}\boldsymbol{q} \otimes \xi_\beta \otimes \omega + \frac{1}{2}\boldsymbol{q} \otimes \dot{\xi}_\beta\right]\epsilon \\
&= \frac{1}{2}\left(\boldsymbol{q} + \frac{\boldsymbol{q} \otimes \xi_\beta}{2}\epsilon\right) \otimes \left(\omega + [\omega \times \xi_\beta + \dot{\xi}_\beta]\epsilon\right).
\end{aligned} \tag{59}$$

Define

$$\hat{\zeta} \triangleq \omega + [\omega \times \xi_\beta + \dot{\xi}_\beta]\epsilon, \tag{60}$$

where $\hat{\zeta}$ is the *twist* dual vector (combination of angular and translational velocities). Finally, the expression for the derivate of a dual quaternion can be obtained

$$\dot{\hat{q}} = \frac{1}{2}\hat{q} \otimes \hat{\zeta}. \tag{61}$$

The twist dual vector can be interpreted as a screw, containing a translation vector along its length, and a rotation that around itself. See Figure 7.

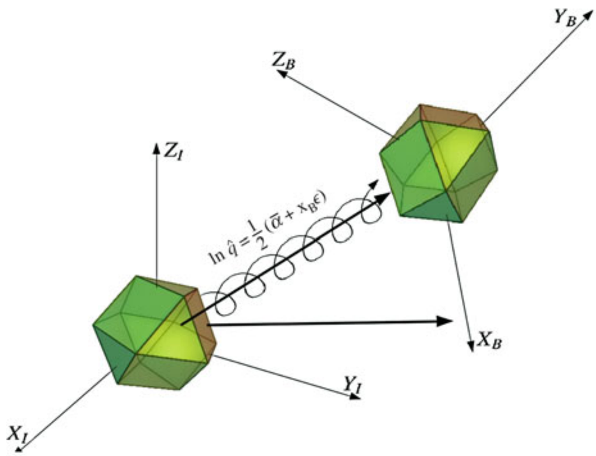


**Fig. 8** Logarithmic representation of a rigid body transformation

## Dual Quaternion Logarithm

A dual quaternion can be transformed by $\ln \hat{q} = \frac{1}{2}(\bar{\alpha} + \xi_\beta \epsilon)$, where $\bar{\alpha} = 2 \ln \boldsymbol{q}$ represents the body's rotation given by a unit quaternion logarithmic mapping and $\xi_\beta$ denotes the position vector in the body frame.

This yields a relationship between a dual quaternion and the screw representation of simultaneous rotation and translation, illustrated in Fig. 8, and defined as

$$\bar{\alpha} + \xi_\beta = 2 \ln \hat{q}\,. \tag{62}$$

## Multiple Dual Quaternion Transformation

Some robots involve successions of rotations and translations, such as aerial manipulators (a quadrotor provided with an attached robotic arm). Dual quaternions become useful in this case. For example, in Fig. 9, $\hat{q}_1$ represents the transformation from reference frame $R_1$ to $R_2$, $\hat{q}_2$ denotes the relative transformation from $R_2$ to $R_3$, and $\hat{q}_3$ defines the total transformation from $R_1$ to $R_3$.

The total transformation from $R_1$ to $R_3$ can be computed using a dual quaternion product as

$$\hat{q}_3 = \hat{q}_1 \otimes \hat{q}_2. \tag{63}$$

If more transformations are required, this expression can be extended by adding as much dual quaternions as needed.

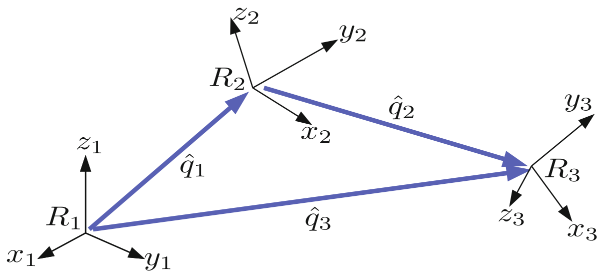


**Fig. 9** Simultaneous dual quaternion transformations can be expressed as a one, using the dual quaternion product

### Dual Quaternion Inverse Transformations

From (63), inverse kinematics can be easily obtained just by multiplying by the conjugate of any required transformation. For example, if $\hat{q}_2$ and $\hat{q}_3$ are given, $\hat{q}_1$ can be computed as

$$\hat{q}_3 \otimes \hat{q}_2^* = \hat{q}_1 \otimes \hat{q}_2 \otimes \hat{q}_2^* \Rightarrow \hat{q}_1 = \hat{q}_3 \otimes \hat{q}_2^*. \tag{64}$$

Similarly, if $\hat{q}_1$ and $\hat{q}_3$ are known, $\hat{q}_2$ can be computed as

$$\hat{q}_1^* \otimes \hat{q}_3 = \hat{q}_1^* \otimes \hat{q}_1 \otimes \hat{q}_2 \Rightarrow \hat{q}_2 = \hat{q}_1^* \otimes \hat{q}_3. \tag{65}$$

## Quadrotor Dual Quaternion Model

Consider the quadrotor as a rigid body, and describing its rotation and translation with respect to the body's reference frame as a dual quaternion $\hat{q}_v$. Its dynamic model is given by

$$\begin{bmatrix} \dot{\hat{q}}_v \\ \dot{\hat{\zeta}}_v \end{bmatrix} = \begin{bmatrix} \frac{1}{2}\hat{q}_v \otimes \hat{\zeta}_v \\ \hat{F}_v + \hat{u}_v \end{bmatrix}, \tag{66}$$

with

$$\hat{\zeta}_v = \dot{\alpha}_v + [\dot{\alpha}_v \times \xi_v + \dot{\xi}_v]\epsilon,$$

where $\dot{\alpha}_v$ expresses the vehicle's angular velocity and $\xi_v$ defines its position with respect to its local frame.

The control and external forces are included in the terms $\hat{F}_v$ and $\hat{u}_v$ as

$$\begin{aligned} \hat{F}_v &= a_v + (a_v \times \xi_v + \dot{\alpha}_v \times \dot{\xi}_v)\epsilon \\ \hat{u}_v &= J_v^{-1}\tau_v + [J_v^{-1}\tau_v \times \xi_v + m_v^{-1}F_v]\epsilon\,, \\ a_v &= -J_v^{-1}(\dot{\alpha}_v \times J_v\dot{\alpha}_v) \end{aligned} \tag{67}$$

where $J_v$ denotes the quadrotor's inertia matrix, $m_v$ represents its mass, $\tau_v \in \mathbb{R}^3$ corresponds to the total torque, and $F_v = F_{th} + \boldsymbol{q}^* \otimes m\bar{g} \otimes \boldsymbol{q}$ is the total force containing the thrust and gravity vectors in the body's reference frame.

In the case of a symmetric vehicle, it is considered that the center of mass is located in the structure's geometric center. The effects of the combination of $\dot{\alpha}_v$ and $\dot{\xi}_v$ can be considered to be nonexistent, thus simplifying the quadrotor's model, which can be rewritten as

$$\begin{bmatrix} \dot{\hat{q}}_v \\ \dot{\hat{\zeta}}_v \end{bmatrix} = \begin{bmatrix} \frac{1}{2}\hat{q}_v \otimes \hat{\zeta}_v \\ \hat{u}_v \end{bmatrix}, \tag{68}$$

with $\hat{\zeta}_v = \dot{\alpha}_v + \dot{\xi}_v\epsilon$, and

$$\hat{u}_v = -J_v^{-1}(\dot{\alpha}_v \times J_v\dot{\alpha}_v - \tau_v) + [m_v^{-1}F_v]\epsilon. \tag{69}$$

## Dual Quaternion Quadrotor Control Example

A dual quaternion control law based on dual quaternions can be easily developed to stabilize the model from section "Quadrotor Dual Quaternion Model". Here, Lyapunov theory is used to validate the design of a simple feedback controller; nevertheless, this methodology can be applied to use other control techniques.

### Dual Vector Arrays, Matrices, and Products

Let two dual vectors be $\hat{g} = \bar{g}_R + \bar{g}_D\epsilon$ and $\hat{h} = \bar{h}_R + \bar{h}_D\epsilon$. A new kind of product $\odot$ between dual vectors is proposed as

$$\hat{g} \odot \hat{h} = \bar{g}_R \cdot \bar{h}_R + \bar{g}_D \cdot \bar{h}_D\epsilon. \tag{70}$$

Note the output of product $\odot$ is a dual scalar number. Let this operator behave on dual vector arrays on the following manner, equivalently to conventional algebraic operations

$$\begin{bmatrix} \hat{g}_1 \\ \hat{g}_2 \end{bmatrix}^T \odot \begin{bmatrix} \hat{h}_1 \\ \hat{h}_2 \end{bmatrix} = \begin{bmatrix} \hat{g}_1 \ \hat{g}_2 \end{bmatrix} \odot \begin{bmatrix} \hat{h}_1 \\ \hat{h}_2 \end{bmatrix} = \hat{g}_1 \cdot \hat{h}_1 + \hat{g}_2 \cdot \hat{h}_2,$$

$$\begin{bmatrix} \hat{g}_1 & \hat{g}_2 \\ \hat{g}_3 & \hat{g}_4 \end{bmatrix} \odot \begin{bmatrix} \hat{h}_1 \\ \hat{h}_2 \end{bmatrix} = \begin{bmatrix} \hat{g}_1 \cdot \hat{h}_1 + \hat{g}_2 \cdot \hat{h}_2 \\ \hat{g}_3 \cdot \hat{h}_1 + \hat{g}_4 \cdot \hat{h}_2 \end{bmatrix}, \tag{71}$$

where $\hat{g}_i, \hat{h}_i, i = 1, \ldots, 4$ are dual vectors. Observe the product between transposed dual vector arrays result in a dual vector.

## Controller Design

Given a desired dual quaternion $\hat{q}_d = \boldsymbol{q}_d + \frac{1}{2}\boldsymbol{q}_d \otimes \xi_d \epsilon$, where $\boldsymbol{q}_d$ is an attitude reference, defined similarly as in section "Attitude Trajectory Strategy", and $\xi_d$ denotes the position which the quadrotor is desired to converge to, then a dual quaternion error can be defined as

$$\hat{q}_e = \hat{q}_d^* \otimes \hat{q}_v. \tag{72}$$

Let $\hat{\mathbf{x}}$ be a vertical array of dual vectors defined as the state of the quadrotor, thus

$$\hat{\mathbf{x}} = \begin{bmatrix} 2 \ln \hat{q}_e \\ \hat{\zeta}_e \end{bmatrix} \quad \rightarrow \quad \dot{\hat{\mathbf{x}}} = \begin{bmatrix} \hat{\zeta}_e \\ \hat{u}_v \end{bmatrix} = \begin{bmatrix} 0 & \hat{I} \\ 0 & 0 \end{bmatrix} \odot \hat{\mathbf{x}} + \begin{bmatrix} 0 \\ \hat{I} \end{bmatrix} \odot \hat{u}_v = A \odot \hat{\mathbf{x}} + B \odot \hat{u}_v, \tag{73}$$

where the velocity errors are included in $\hat{\zeta}_e = \dot{\alpha}_e + \dot{\xi}_e \epsilon$, the control torques and forces are considered as $\hat{u}_v = -J_v^{-1}(\dot{\alpha}_e \times J_v \dot{\alpha}_e - \tau_v) + [m_v^{-1} F_v]\epsilon$, $\hat{I} = [1\ 1\ 1]^T + [1\ 1\ 1]^T \epsilon$ denotes a constant dual vector, and $A, B$ are matrices containing dual quaternion entries.

A dual scalar positive-definite function

$$\hat{V} = \frac{1}{2}\hat{\mathbf{x}}^T \odot \hat{\mathbf{x}} = V_R + V_D \epsilon, \quad V_R, V_D \in \mathbb{R}^+ \tag{74}$$

can be proposed, being its derivative

$$\dot{\hat{V}} = \hat{\mathbf{x}}^T \odot \dot{\hat{\mathbf{x}}} = \hat{\mathbf{x}}^T \odot (A \odot \hat{\mathbf{x}} + B \odot \hat{u}_v). \tag{75}$$

Defining a state feedback controller

$$\hat{u}_v = 2\hat{k}_p \cdot \ln \hat{q}_e + \hat{k}_d \cdot \hat{\zeta}_e = \begin{bmatrix} \hat{k}_p & \hat{k}_d \end{bmatrix} \odot \hat{\mathbf{x}}, \tag{76}$$

being $\hat{k}_p = [k_{pRx}\ k_{pRy}\ k_{pRz}]^T + [k_{pDx}\ k_{pDy}\ k_{pDz}]^T \epsilon$ and $\hat{k}_d = [k_{dRx}\ k_{dRy}\ k_{dRz}]^T + [k_{dDx}\ k_{dDy}\ k_{dDz}]^T \epsilon$ dual vectors denoting control gains, then (75) yields

$$
\begin{aligned}
\dot{\hat{V}} &= \hat{\mathbf{x}} \odot \left(A \odot \hat{\mathbf{x}} + B \odot \begin{bmatrix}\hat{k}_p \ \hat{k}_d\end{bmatrix} \odot \hat{\mathbf{x}}\right) \\
&= \hat{\mathbf{x}} \odot \left(\begin{bmatrix}0 & \hat{I} \\ 0 & 0\end{bmatrix} + \begin{bmatrix}0 \\ \hat{I}\end{bmatrix} \odot \begin{bmatrix}\hat{k}_p \ \hat{k}_d\end{bmatrix}\right) \odot \hat{\mathbf{x}} \\
&= \hat{\mathbf{x}} \odot \begin{bmatrix}0 & \hat{I} \\ \hat{k}_p & \hat{k}_d\end{bmatrix} \odot \hat{\mathbf{x}} \\
&= \begin{bmatrix}\bar{\alpha} \\ \dot{\bar{\alpha}}\end{bmatrix} \odot \begin{bmatrix}0^{3\times3} & I^{3\times3} \\ K_{pR} & K_{dR}\end{bmatrix}\begin{bmatrix}\bar{\alpha} \\ \dot{\bar{\alpha}}\end{bmatrix} + \begin{bmatrix}\xi \\ \dot{\xi}\end{bmatrix} \cdot \begin{bmatrix}0^{3\times3} & I^{3\times3} \\ K_{pD} & K_{dD}\end{bmatrix}\begin{bmatrix}\xi \\ \dot{\xi}\end{bmatrix}\epsilon,
\end{aligned}
\tag{77}
$$

and

$$
\dot{V}_R = \begin{bmatrix}\bar{\alpha} \\ \dot{\bar{\alpha}}\end{bmatrix} \odot \begin{bmatrix}0^{3\times3} & I^{3\times3} \\ K_{pR} & K_{dR}\end{bmatrix}\begin{bmatrix}\bar{\alpha} \\ \dot{\bar{\alpha}}\end{bmatrix} \quad , \quad \dot{V}_D = \begin{bmatrix}\xi \\ \dot{\xi}\end{bmatrix} \cdot \begin{bmatrix}0^{3\times3} & I^{3\times3} \\ K_{pD} & K_{dD}\end{bmatrix}\begin{bmatrix}\xi \\ \dot{\xi}\end{bmatrix}. \tag{78}
$$

If $\dot{V}_R, \dot{V}_D < 0 \ \forall \ \hat{\mathbf{x}} \neq 0$, then $2\ln\hat{q}_e \to 0, \hat{\xi} \to 0$ and $\hat{q}_e \to 1$; the quadrotor's pose $\hat{q}$ asymptotically converges to $\hat{q}_d$ if the real part of all the eigenvectors of matrices $\begin{bmatrix}0^{3\times3} & I^{3\times3} \\ K_{pR} & K_{dR}\end{bmatrix}$ and $\begin{bmatrix}0^{3\times3} & I^{3\times3} \\ K_{pD} & K_{dD}\end{bmatrix}$ is negative definite; therefore, the components of $\hat{k}_p$ and $\hat{k}_d$ need to be selected according to the following inequalities:

$$
\begin{aligned}
&\left.\begin{array}{r}
\mathrm{re}\big(k_{dRx} + (k_{dRx}^2 + 4k_{pRx})^{1/2}\big)/2 \\
\mathrm{re}\big(k_{dRy} + (k_{dRy}^2 + 4k_{pRy})^{1/2}\big)/2 \\
\mathrm{re}\big(k_{dRz} + (k_{dRz}^2 + 4k_{pRz})^{1/2}\big)/2 \\
\mathrm{re}\big(k_{dRx} - (k_{dRx}^2 + 4k_{pRx})^{1/2}\big)/2 \\
\mathrm{re}\big(k_{dRy} - (k_{dRy}^2 + 4k_{pRy})^{1/2}\big)/2 \\
\mathrm{re}\big(k_{dRz} - (k_{dRz}^2 + 4k_{pRz})^{1/2}\big)/2
\end{array}\right\} < 0, \\
&\left.\begin{array}{r}
\mathrm{re}\big(k_{dDx} + (k_{dDx}^2 + 4k_{pDx})^{1/2}\big)/2 \\
\mathrm{re}\big(k_{dDy} + (k_{dDy}^2 + 4k_{pDy})^{1/2}\big)/2 \\
\mathrm{re}\big(k_{dDz} + (k_{dDz}^2 + 4k_{pDz})^{1/2}\big)/2 \\
\mathrm{re}\big(k_{dDx} - (k_{dDx}^2 + 4k_{pDx})^{1/2}\big)/2 \\
\mathrm{re}\big(k_{dDy} - (k_{dDy}^2 + 4k_{pDy})^{1/2}\big)/2 \\
\mathrm{re}\big(k_{dDz} - (k_{dDz}^2 + 4k_{pDz})^{1/2}\big)/2
\end{array}\right\} < 0.
\end{aligned}
\tag{79}
$$

*Remark:* From (76) and following (69), the control forces and torques can be easily computed as

$$
\hat{u}_v = 2\hat{k}_p \cdot \ln\hat{q}_e + \hat{k}_d \cdot \hat{\zeta}_e = -J_v^{-1}(\dot{\bar{\alpha}}_v \times J_v\dot{\bar{\alpha}}_v - \tau_v) + [m_v^{-1}F_v]\epsilon. \tag{80}
$$

Separating real and dual components yields

$$\tau_v = J_v\big(K_{pR}\bar{\alpha}_e + K_{dR}\dot{\bar{\alpha}}_e\big) + \dot{\bar{\alpha}}_e \times J\,\dot{\bar{\alpha}}_e, \quad F_v = -m_v\big(K_{pD}\xi_e + K_{dD}\dot{\xi}_e\big), \quad (81)$$

which are equivalent to the unit quaternion approach from (40) and (48). If other control strategies are used, the same methodology can be used to extract the actual forces and torques that will be used as inputs for the quadrotor.

Using the same strategy as in section "Attitude Trajectory Strategy", the attitude component $\boldsymbol{q}_d$ of the dual quaternion reference $\hat{q}_d$ can be computed using the control force, following (45).

## Simulation Example

Taking into account the model from (66), and applying the dual quaternion controller (76), a numerical simulation is done using MATLAB and a simple Riemann's integration technique.

The flight scenario, model parameters, and control gains were considered to be the same as from section "Simulation Example", giving a better comparison point of view between both approaches. In this case the control gains are defined as

$$\hat{k}_p = -\begin{bmatrix}50\\50\\50\end{bmatrix} - \begin{bmatrix}8\\8\\4\end{bmatrix}\epsilon\ , \quad \text{and} \quad \hat{k}_d = -\begin{bmatrix}10\\10\\10\end{bmatrix} - \begin{bmatrix}4\\4\\4\end{bmatrix}\epsilon\ . \quad (82)$$

Notice that conditions from (79) are met.

Recall that the flight scenario is a $r = 2$m circular path with a take off stage, defined in the inertial reference frame as

$$\xi_d = \begin{cases} \begin{bmatrix}0 \ 0 \ 1\end{bmatrix}^T & \forall\ t < 5\text{s} \\ \begin{bmatrix} -r\cos(t-5) + r \\ -r\sin(t-5) \\ 0 \end{bmatrix} & \forall\ t > 5\text{s} \end{cases} . \quad (83)$$

Computing $\boldsymbol{q}_d$ from (45), and defining $\boldsymbol{q}_z$ as

$$\delta_{xy} = \begin{bmatrix}1 \ 0 \ 0\end{bmatrix}^T - \xi \cdot \begin{bmatrix}1 \ 1 \ 0\end{bmatrix}^T \quad , \quad \boldsymbol{q}_z = e^{\dfrac{\ln\left(\delta_{xy} \otimes (-[1\ 0\ 0]^T)\right)}{2}} , \quad (84)$$

the dual quaternion reference can be defined as

$$\hat{q}_d = \boldsymbol{q}_d + \frac{1}{2}\xi_d \otimes \boldsymbol{q}_d . \quad (85)$$

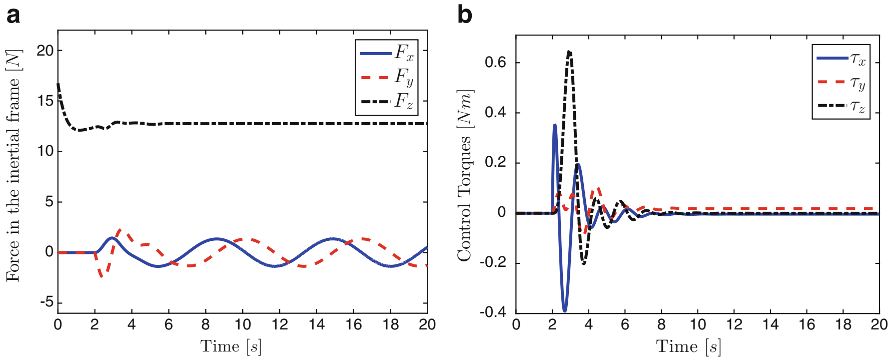


**Fig. 10** Quadrotor dual quaternion control inputs. (**a**) Translational controller. (**b**) Rotational controller

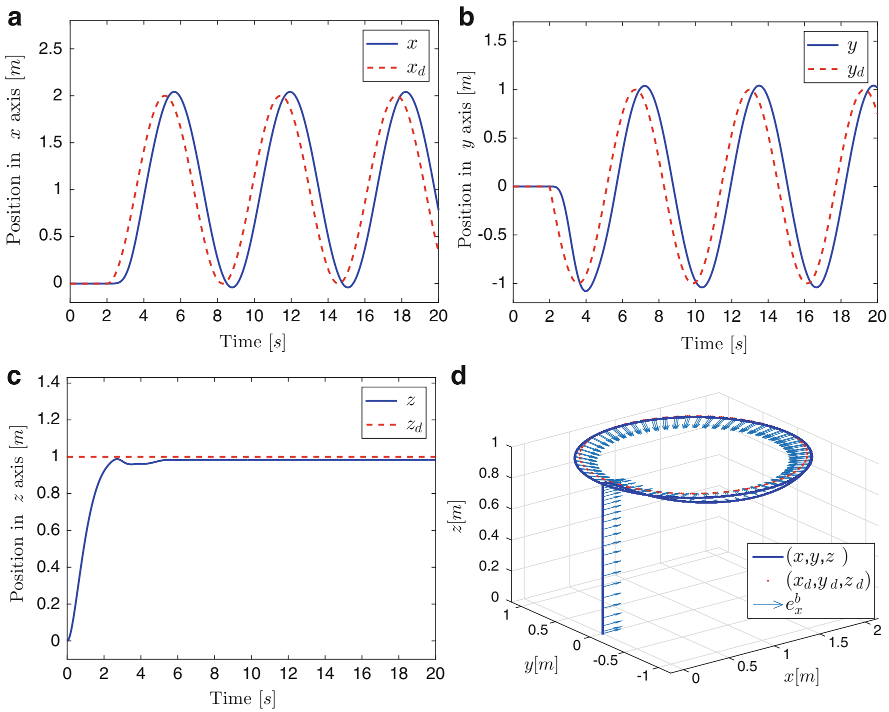


**Fig. 11** Quadrotor position tracking using a dual quaternion controller. (**a**) Quadrotor's $x$ axis position and reference. (**b**) Quadrotor's $y$ axis position and reference. (**c**) Quadrotor's altitude regulation. (**d**) Vehicle's translation and orientation tracking

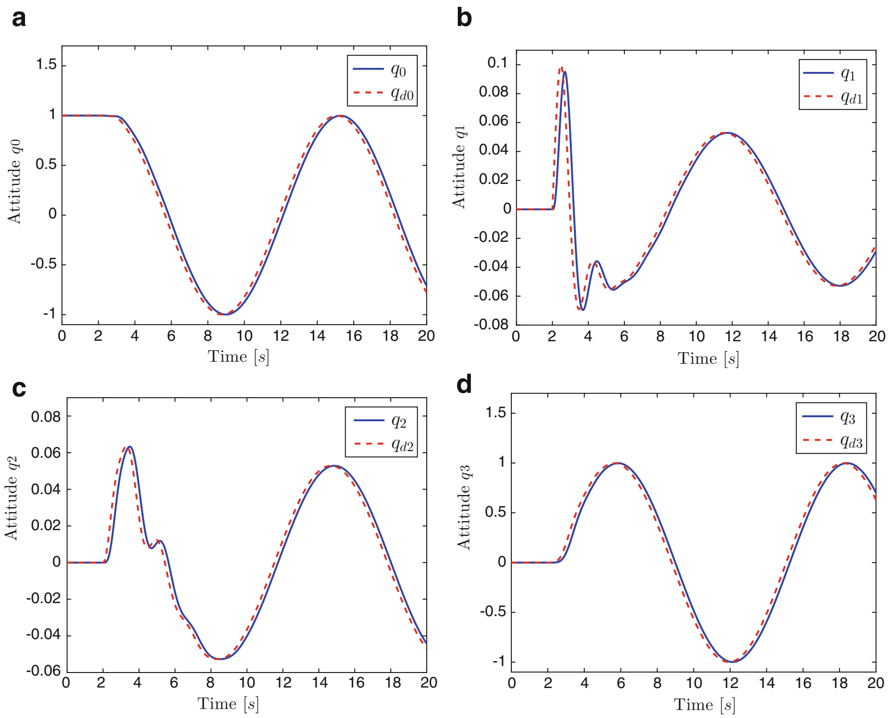


**Fig. 12** Quadrotor attitude trajectory, tracked by a dual quaternion controller

The controller computes the forces and torques required to stabilize the vehicle, as illustrated in Fig. 10.

The position reference tracking is depicted in Fig. 11, while the vehicle path with its front symbolized by an arrow is illustrated in Fig. 11d.

The attitude trajectory is represented in Fig. 12. Similarly as on the unit quaternion approach, the vehicle turns around the $z$ axis while remaining singularity-free.

Finally, the axis-angle equivalent attitude representation is illustrated on Fig. 13. The axis-angle representation also uncovers the discontinuities that are not present on dual quaternions.

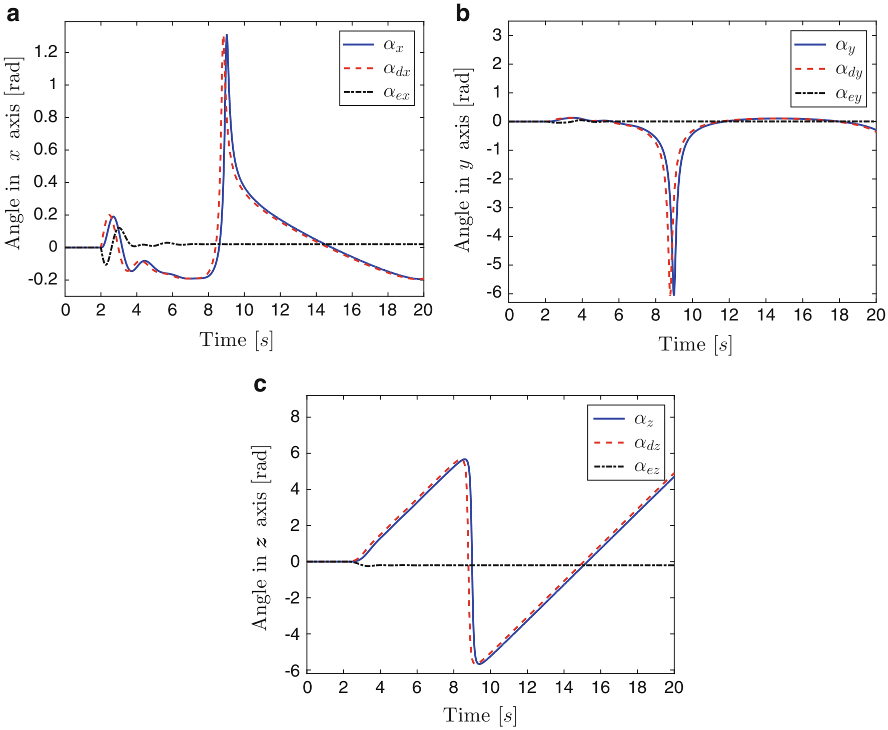


**Fig. 13** Quadrotor attitude axis-angle representation, controlled by a dual quaternion algorithm. (**a**) Quadrotor's angle and error in $x$ axis. (**b**) Quadrotor's angle and error in $y$ axis. (**c**) Quadrotor's angle and error in $z$ axis

## Discussion

Quaternions, as a mathematical concept, have been known for more than a century. Nevertheless, their application in modeling and control of robotic systems, more specifically unmanned aerial vehicles, is relatively recent. Until the last few years, most researchers have chosen more conservative approaches as the Euler-Lagrange or Newton-Euler methodologies to describe dynamic models and design controllers. However, intuitive representation has been surpassed by its many limitations, like the presence of important nonlinearities, undesired effects such as the Gimball-Lock, and an inherent complexity when multiple rotations and translations are present.

The goal of this chapter is to provide a clear introduction to the use of quaternions for describing the dynamics and designing controllers for unmanned aerial systems. The objectives are to present the main concepts which are needed to apply this methodology to any aerial system and to present the particular application on quadrotors as an example. Although this approach can appear to be less intuitive and difficult to conceptualize, we believe that once the basic concepts are understood, the application of quaternions can really simplify dynamic models and help in the design of better controllers.

Finally, dual quaternions provide a powerful tool for modeling more complex systems; since multiple rotations and translations can be expressed as simple quaternion products, the computation of direct and inverse kinematics, as well as the synthesis of dynamic models, can be considerably simplified.

## References


H. Abaunza, E. Ibarra, P. Castillo, A. Victorino, Quaternion based control for circular UAV trajectory tracking, following a ground vehicle: real-time validation, in *The 20th IFAC World Congress*, Toulouse, 2017

H. Abaunza, P. Castillo, R. Lozano, A. Victorino, Dual quaternion modeling and control of a quad-rotor aerial manipulator. J. Intell. Robot. Syst. (JINT) 2017. Springer, **88**(2-4), 267–283

S.L. Altmann, Hamilton, Rodrigues, and the quaternion scandal. Math. Mag. **62**, 291–308 (1989)

A. Buchheim, A memoir on Biquaternions. Am. J. Math. **7**(4), 293–326 (1885)

J. Cariño, H. Abaunza, P. Castillo, Quad-rotor quaternion control, in *The International Conference on Unmanned Aircraft Systems (ICUAS)*, Denver, 2015

H. Goldstein, *Classical Mechanics*, vol. 4 (Pearson Education, India, 1962)

J.B. Kuipers, *Quaternions and Rotation Sequences*, vol. 66 (Princeton University Press, Princeton, 1999)

A.P. Kotelnikov, Screw calculus and some applications to geometry and mechanics. Annuals of Imperial University of Kazan, 1895

W.K. Lifford, Preliminary sketch of Biquaernions. Proc. Lond. Math. Soc. **4**(64/65), 381–395 (1873)

A. McAulay, *Octonions: A Development of Clifford's Bi-Quaternions* (University Press, Cambridge, 1898)

K.W. Spring, Euler parameters and the use of quaternion algebra in the manipulation of finite rotations: a review. Mech. Mach. Theory **21**(5), 365–373 (1986)