Quaternion Guidance and Control of Quadrotor

Tom Stian Andersen
Department of Electrical Engineering
The Arctic University of Norway
NO-8514 Narvik, Norway
Email: tom.s.andersen@uit.no

Raymond Kristiansen
Department of Electrical Engineering
The Arctic University of Norway
NO-8514 Narvik, Norway
Email: raymond.kristiansen@uit.no

Abstract—**This paper addresses the problem of trajectory tracking for underactuated quadrotors. A quaternion based guidance law is proposed which feeds into an attitude control system based on a PD+ control law. The desired attitude from the guidance law is defined such that the attitude control system tries to align the position error along the axis of the translational actuator. Simulation results are provided and discussed to demonstrate the proposed method.**

I. Introduction

Over the recent years there has been a surge of interest in small UAVs and in particular quadrotor UAVs. They have a wide range of applications for instance in surveillance, surveying, transportation, search and rescue and structural inspection. The increasing availability of small, light and cheap sensor platforms and powerful embedded processors allows for highly sophisticated guidance, navigation and control algorithms to be used to highten the level of autonomy of quadrotors. In recent years nonlinear control has been employed to deal with the nonlinearities of the quadrotor dynamics although classical linear techniques are still prevalent. One of the earlier works on Quadrotor control is that of [1] where PID and LQ methods are compared. In [2] a tracking controller is developed directly on SE(3) and almost global exponential attractivness is shown. In [3] integral backstepping is used for attitude, altitude and position control which enables the quadrotor to fly autonomously and avoid obstacles. In [4] differential flatness of the quadrotor dynamics are exploited to generate optimal safe trajectories for the quadrotor to follow and a nonlinear controller is used to ensure convergence to the desired trajectories. In [5] a highly sophisticated nonlinear model for a quadrotor UAV is developed, the model was also simulated with an NMPC-controller applied to it and the simulation shows stable and near optimal performance. A common theme in most quadrotor literature is the usage of Euler angles for the attitude parameterization. Euler angles are known to have singularities which limit the quadrotors ability to follow arbitrary trajectories. In this work quaternions are used for the attitude representations since they are singularity free and more compact than rotation matrices. A quaternion based guidance algorithm is developed which enables a high degree of flexibility in specifying the desired performance of the quadrotor through tuning gains.

This paper is organized in the following way. Section II introduces quaternions, which is the main tool in this paper, and some useful properties. Section III details the dynamic model for the quadrotor in using quaternions. Section IV derives the quaternion guidance algorithm. In Section V nonlinear attitude and translational controllers are presented. In Section VI simulations are performed to show the potential of the approach and a brief conclusion is given in Section VI highlighting some possible extensions of this work.

II. Preliminaries

A. Notation

Vectors are denoted as lower-case bold letters while scalars are non-bold for instance $\boldsymbol{x} \in \mathbb{R}^n$ is an n-dimensional vector while $a \in \mathbb{R}$ is a scalar. Matrices are upper-case bold letters. The set of positive real numbers is denoted $\mathbb{R}_+$. The time derivative of a vector is denoted as $\dot{\boldsymbol{x}} = \frac{d\boldsymbol{x}}{dt}$ and the second time derivative is denoted as $\ddot{\boldsymbol{x}} = \frac{d^2\boldsymbol{x}}{dt^2}$. Coordinate reference frames are denoted $\mathcal{F}_{(\cdot)}$, and superscripts are used to denote a variables frame of reference, such that the vector $\boldsymbol{x}^A$ is referenced in $\mathcal{F}_A$. The inner-product between two vectors $\boldsymbol{x} \in \mathbb{R}^n$ and $\boldsymbol{y} \in \mathbb{R}^n$ is written as $\langle \boldsymbol{x}, \boldsymbol{y} \rangle = (\boldsymbol{x})^T \boldsymbol{y}$. The euclidian norm is denoted as $\|\boldsymbol{x}\| = \langle \boldsymbol{x}, \boldsymbol{x} \rangle^{\frac{1}{2}}$.

B. Reference frames

a) NED frame: This coordinate reference frame denoted $\mathcal{F}_n$ has its x-axis pointing north, its y-axis pointing east and its z-axis pointing down completing a right-handed coordinate system see Figure 1. The origin of the NED frame is usually chosen to be at the intial position of the rigid-body in question. This can for instance be the take-off point in case of a quadrotor UAV or pre-defined point in inertial space. In this paper $\mathcal{F}_n$ is also assumed to be inertial.

b) Body frame: This coordinate reference frame denoted $\mathcal{F}_b$ is fixed at the quadrotor's centre of mass. The x-axis points from the center of mass through the first propeller, y-axis points from the center of mass throught the fourth propeller and the z-axis points downwards completing the right handed coordinate reference frame. The frame represents the orientation of the quadrotor with respect to $\mathcal{F}_n$ as seen in Figure 1.

c) Desired frame: This coordinate reference frame denoted $\mathcal{F}_d$ is defined by the guidance system, and represents the quadrotor's desired orientation.

C. Quaternions

In this section a brief overview of quaternions are given, more complete formulations can be found in literature *cf.* [6], [7]. Quaternions are well known to offer a singularity free parameterization of a rigid-body's orientation using only four parameters in contrast to nine parameters for the direction cosine matrices. Similary to direction cosine matrices they can also be used to transform vectors between reference frames. Quaternions can be compactly represented as an element in $\mathbb{R}^4$ as

$$\boldsymbol{q} = \begin{bmatrix} q_0 \\ \boldsymbol{q}_v \end{bmatrix}, \qquad q_0 \in \mathbb{R}, \boldsymbol{q}_v \in \mathbb{R}^3 \tag{1}$$

and can be used to represent a rotation through the formula

$$\boldsymbol{q} = \begin{bmatrix} \cos\frac{\theta}{2} & \boldsymbol{u}_\theta \sin\frac{\theta}{2} \end{bmatrix} \tag{2}$$

where $\theta \in \mathbb{R}$ is the angle of rotation and $\boldsymbol{u}_\theta \in \mathbb{R}^3$ is a unit vector pointing in the direction of axis of rotation. A vector $\boldsymbol{u}^b \in \mathbb{R}^3$ can be rotated from $\mathcal{F}_b$ to $\mathcal{F}_n$ by the use of the sandwich product

$$\boldsymbol{u}^n = \boldsymbol{q}_{n,b} \otimes \boldsymbol{u}^b \otimes \boldsymbol{q}^*_{n,b} \tag{3}$$

where the quaternion product is defined as

$$\boldsymbol{q}_1 \otimes \boldsymbol{q}_2 = \begin{bmatrix} q_{1,0}q_{2,0} - \boldsymbol{q}^T_{1,v}\boldsymbol{q}_{2,v} \\ q_{1,0}\boldsymbol{q}_{2,v} + q_{2,0}\boldsymbol{q}_{1,v} + \boldsymbol{q}_{1,v} \times \boldsymbol{q}_{2,v} \end{bmatrix} \tag{4}$$

and $(\cdot)^*$ is the quaternion conjugate defined as

$$\boldsymbol{q}^* = \begin{bmatrix} q_0 \\ -\boldsymbol{q}_v \end{bmatrix}^T.$$

The norm of a quaternion can be defined through the quaternion product as

$$\|\boldsymbol{q}\| = \sqrt{\boldsymbol{q}^* \otimes \boldsymbol{q}}$$

and should always be equal to unity to ensure that lengths are preserved when using (3). Therefore in this paper all quaternions are assumed to be unit quaternions. Several quaternions can also be combined using the quaternion product to represent composite rotations as

$$\boldsymbol{q}_{a,c} = \boldsymbol{q}_{a,b} \otimes \boldsymbol{q}_{b,c}$$

and difference in rotations can be defined as

$$\boldsymbol{q}_{b,d} = \boldsymbol{q}^*_{a,b} \otimes \boldsymbol{q}_{a,d}.$$

The quaternion kinematics is defined as [6]

$$\dot{\boldsymbol{q}}_{a,b} = \frac{1}{2}\boldsymbol{\omega}^a_{a,b} \otimes \boldsymbol{q}_{a,b} = \frac{1}{2}\boldsymbol{q}_{a,b} \otimes \boldsymbol{\omega}^b_{a,b} \tag{5}$$

where $\boldsymbol{\omega}^{(\cdot)}_{a,b} \in \mathbb{R}^3$ is the angular velocity of $\mathcal{F}_b$ relative to $\mathcal{F}_a$, expressed either in $\mathcal{F}_a$ or $\mathcal{F}_b$.

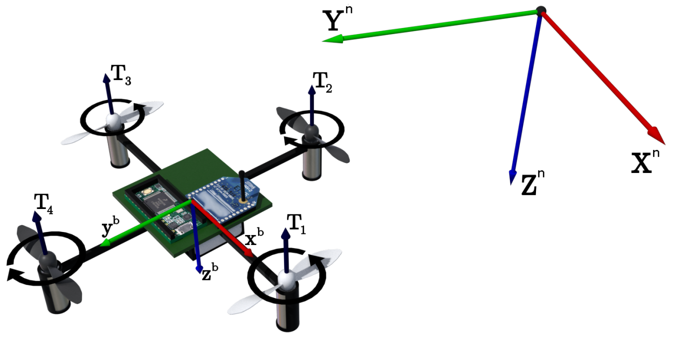

Fig. 1. Illustration of reference frames often used in quadrotor literature *c.f.* [8].

III. Modeling

A. Quadrotor dynamics

In the quadrotor dynamics literature *c.f.* [9], [5], [2], [4], [8], the Newton-Euler equations of motion, with reference frames as shown in Figure 1, for a quadrotor is commonly defined as

$$\dot{\boldsymbol{v}}^n = \frac{1}{m}\left(\boldsymbol{f}^n_G - \boldsymbol{R}^n_b \boldsymbol{f}^b_T\right) \tag{6}$$

$$\dot{\boldsymbol{\omega}}^b_{n,b} = \left(\boldsymbol{J}^b\right)^{-1}\left(\boldsymbol{\tau}^b - \boldsymbol{\omega}^b_{n,b} \times \left(\boldsymbol{J}^b \boldsymbol{\omega}^b_{n,b}\right)\right) \tag{7}$$

where $\boldsymbol{v}^n \in \mathbb{R}^3$ in the inertial velocity of the quadrotor, $\boldsymbol{f}^n_G \in \mathbb{R}^3$ is the gravity vector, $\boldsymbol{f}^b_T \in \mathbb{R}^3$ is the quadrotor's thrust vector, $\boldsymbol{R}^n_b \in \mathbb{R}^{3x3}$ is the orientation of the quadrotor parameterized using quaternions, $\boldsymbol{\omega}^b_{n,b} \in \mathbb{R}^3$ is the angular velocity of the quadrotor, $\boldsymbol{J}^b = \mathrm{diag}\left\{J_x, J_y, J_z\right\} \in \mathbb{R}^{3x3}$ is the quadrotor's inertia matrix and $\boldsymbol{\tau}^b \in \mathbb{R}^3$ is the applied torques on the quadrotor. The input thrust and torques can be calculating according to [5] as

$$\begin{aligned} \boldsymbol{f}^b_T &= -\begin{bmatrix} 0 \\ 0 \\ c_T\left(\sigma_1^2 + \sigma_2^2 + \sigma_3^2 + \sigma_4^2\right) \end{bmatrix} \\ \boldsymbol{\tau}^b &= \begin{bmatrix} d_r c_T\left(\sigma_4^2 - \sigma_2^2\right) \\ d_r c_T\left(\sigma_1^2 - \sigma_3^1\right) \\ d_r c_T\left(\sigma_1^2 + \sigma_3^2 - \sigma_2^1 - \sigma_4^2\right) \end{bmatrix} \end{aligned} \tag{8}$$

where $\sigma_i, i = 1, 2, 3, 4$ is the angular velocity of the ith propeller, c_T and c_Q are the thrust coefficient and the torque coeffiecient respectively and d_r is the distance from the centre of the quadrotor to the centre of the propellers which is assumed to be equal for all propellers.

IV. Guidance

The guidance algorithm is inspired from [10], but is modified to allow for flexibility in tuning the performance of guidance algorithm. Let the desired position $\boldsymbol{p}^n_d$ be represented as point on a given trajectory and let the position error be defined as

$$\boldsymbol{e}^n = \boldsymbol{p}^n_d - \boldsymbol{p}^n \tag{9}$$

where $\boldsymbol{p}^n \in \mathbb{R}^3$ denotes the current position of the quadrotor. The only translation actuator the quadrotor has is the thrust along the negative *x-axis* in $\mathcal{F}_b$. Therefore the desired orientation can be represented

$$\boldsymbol{e}^d = \begin{bmatrix} 0 & 0 & -\|\boldsymbol{e}^n\| \end{bmatrix}^T = \boldsymbol{q}^*_{n,d} \otimes \boldsymbol{e}^n \otimes \boldsymbol{q}_{n,d}. \tag{10}$$

The quaternion representing the desired orientation can be calculated as

$$\boldsymbol{q}_{n,d} = \begin{bmatrix} \cos\frac{\theta}{2} & \boldsymbol{k}\sin\frac{\theta}{2} \end{bmatrix}^T \tag{11}$$

where

$$\theta = \arccos\left(\frac{(\boldsymbol{e}^d)^T\boldsymbol{e}^n}{\|(\boldsymbol{e}^d)^T\boldsymbol{e}^n\|}\right) \tag{12}$$

and

$$\boldsymbol{k} = \frac{\boldsymbol{e}^d \times \boldsymbol{e}^n}{\|\boldsymbol{e}^d \times \boldsymbol{e}^n\|}. \tag{13}$$

An issue with this guidance law is that θ will be large for most values of $\boldsymbol{e}^d$ and $\boldsymbol{e}^n$. For instance in the case when the altitud of the quadrotor's position is equal to the altitude of the desired position. In these cases θ will be $\frac{\pi}{2}$ and the quadrotor will consequently try to tilt itself 90° resulting in loss of altitude since there will be no thrust to cancel the effects of gravity. As the quadrotor falls θ will become smaller such that the quadrotor will eventually recover if enough thrust is available, but if $\boldsymbol{p}^n_d$ is at low altitude it can mean that the quadrotor crashes with the ground before it recovers. More influence over the behaviour of the quadrotor is gained by redefining (12) using the $\arctan$ function as

$$\theta = k_1 \arctan\left(k_2\|\boldsymbol{e}^n\|\right) \tag{14}$$

where $k_1, k_2 \in \mathbb{R}$. In (14) the constant k_1 influences the maximum tilt allowed by the quadrotor while k_2 influences how aggressive the quadrotor's approach should be. Since θ and $\boldsymbol{k}$ are defined seperately it is not guaranteed that $\boldsymbol{q}_{n,d}$ is a unit-quaternion and therefore has to be normalized after it has been calculated. The desired angular velocity is obtained by taking the derivative of (10)

$$\begin{aligned} \dot{\boldsymbol{e}}^d &= \dot{\boldsymbol{q}}^*_{n,d} \otimes \boldsymbol{e}^n \otimes \boldsymbol{q}_{n,d} + \boldsymbol{q}^*_{n,d} \otimes \dot{\boldsymbol{e}}^n \otimes \boldsymbol{q}_{n,d} + \boldsymbol{q}^*_{n,d} \otimes \boldsymbol{e}^n \otimes \dot{\boldsymbol{q}}_{n,d} \\ &= \boldsymbol{q}^*_{n,d} \otimes \dot{\boldsymbol{e}}^n \otimes \boldsymbol{q}_{n,d} + \boldsymbol{\omega}^d_{n,d} \times \boldsymbol{e}^d \end{aligned}$$

which can be rewritten as

$$\boldsymbol{e}^d \times \boldsymbol{\omega}^d_{n,d} = \boldsymbol{q}^*_{n,d} \otimes \dot{\boldsymbol{e}}^n \otimes \boldsymbol{q}_{n,d} - \dot{\boldsymbol{e}}^d. \tag{15}$$

Using the skew-symmetric operator and the Moore-Penrose pseudoinverse, (15) can be solved with respect to $\boldsymbol{\omega}^d_{n,d}$ as

$$\boldsymbol{\omega}^d_{n,d} = -S\left(\boldsymbol{e}^d\right)^\dagger \left(\boldsymbol{q}^*_{n,d} \otimes \dot{\boldsymbol{e}}^n \otimes \boldsymbol{q}_{n,d} - \dot{\boldsymbol{e}}^d\right) \tag{16}$$

where

$$S\left(\boldsymbol{e}^d\right)^\dagger = \begin{bmatrix} 0 & \frac{1}{\|\boldsymbol{e}^n\|} & 0 \\ -\frac{1}{\|\boldsymbol{e}^n\|} & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad \dot{\boldsymbol{e}}^d = \begin{bmatrix} 0 \\ 0 \\ \frac{-(\dot{\boldsymbol{e}}^n)^T\boldsymbol{e}^n}{\|\boldsymbol{e}^n\|} \end{bmatrix}.$$

This implies that $S\left(\boldsymbol{e}^d\right)^\dagger \dot{\boldsymbol{e}}^d = 0$ and (15) reduces to

$$\boldsymbol{\omega}^d_{n,d} = -S\left(\boldsymbol{e}^d\right)^\dagger \boldsymbol{q}^*_{n,d} \otimes \dot{\boldsymbol{e}}^n \otimes \boldsymbol{q}_{n,d}. \tag{17}$$

The desired angular acceleration can be found either by using a linear filter or by differentiating (15).

Remark: The degenerate case when $\|\boldsymbol{e}^d \times \boldsymbol{e}^n\| = 0$ needs to be handled explicitly and this is done by setting $\boldsymbol{k} = \boldsymbol{0}$, which gives $\boldsymbol{q}_{n,d} = \boldsymbol{q}_I$ after normalization. Furthermore in the case when $\|\boldsymbol{e}^n\| = 0$, then $\dot{\boldsymbol{e}}^d = \boldsymbol{0}$ and $S\left(\boldsymbol{e}^d\right)^\dagger = \boldsymbol{0}_{3\text{x}3}$.

This guidance law only specifies the amount of desired roll and pitch for the quadrotor, but can easily be extended to include a desired yaw. If we let $\boldsymbol{q}_{yaw}$ represent the desired yaw then the desired orientation can be redefined to be

$$\boldsymbol{q}_{n,d} = \boldsymbol{q}_{roll,pitch} \otimes \boldsymbol{q}_{yaw} \tag{18}$$

where $\boldsymbol{q}_{roll,pitch}$ is defined as in (11). This extended desired orientation is then used in (17) to defined the desired angular velocity.

V. Control

The approach to the nonlinear control of the quadrotor follows that of [4] and [2].

A. Attitude control

Let the guidance system provide the attitude controller with the signals $\boldsymbol{q}_{n,d} \in \mathbb{R}^4$ and $\boldsymbol{\omega}^d_{n,d}, \dot{\boldsymbol{\omega}}^d_{n,d} \in \mathbb{R}^3$. Then the quaternion error is defined as

$$\boldsymbol{q}_{d,b} = \boldsymbol{q}^*_{n,d} \otimes \boldsymbol{q}_{n,b} \tag{19}$$

and the angular velocity error is defined as

$$\boldsymbol{\omega}^b_{d,b} = \boldsymbol{\omega}^b_{n,b} - \boldsymbol{q}_{d,b} \otimes \boldsymbol{\omega}^d_{n,d} \otimes \boldsymbol{q}_{d,b}. \tag{20}$$

The torque can then be defined as

$$\tau = \boldsymbol{\omega}^b_{n,b} \times \boldsymbol{J}^b\boldsymbol{\omega}^b_{n,b} + \boldsymbol{J}^b\dot{\boldsymbol{\omega}}^b_{n,d} - k_q\boldsymbol{q}_{v_{d,b}} - k_\omega\boldsymbol{\omega}^b_{d,b} \tag{21}$$

which is a PD+ controller with a feedforward term.

B. Translational control

Given a trajectory defined by $\boldsymbol{p}^n_d, \dot{\boldsymbol{p}}^n_d, \ddot{\boldsymbol{p}}^n_d \in \mathbb{R}^3$ the distance and velocity errors are defined as

$$\boldsymbol{e}^n = \boldsymbol{p}^n_d - \boldsymbol{p}^n \tag{22}$$

$$\dot{\boldsymbol{e}}^n = \dot{\boldsymbol{p}}^n_d - \boldsymbol{v}^n. \tag{23}$$

The total thrust can then be defined as

$$\boldsymbol{f}^n_T = m\ddot{\boldsymbol{p}}^n_d + \boldsymbol{f}_g - k_p\boldsymbol{e}^n - k_d\dot{\boldsymbol{e}}^n \tag{24}$$

which is a PD+ controller with feedforward compensation. To get the thrust vector in the body frame we take the projection of (24) with $\boldsymbol{u}_z = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}^T$

$$\boldsymbol{f}^b_T = \left(\left(\boldsymbol{f}^n_T\right)^T \boldsymbol{u}_z\right)\boldsymbol{u}_z. \tag{25}$$

TABLE I
PARAMATERS FOR AN ASCTEC PELICAN IN [11]

Parameter	Value	Unit
Quadrotor mass, m	1.27022	kg
X-axis inertia, J_{xx}	0.04338	kg·m^2
Y-axis inertia, J_{yy}	0.04338	kg·m^2
Z-axis inertia, J_{zz}	0.07050	kg·m^2
Distance to rotor, d_r	0.149352	m
Thrust coefficient, c_T	$1.5 \cdot 10^{-5}$	
Torque coefficient, c_Q	$1.9 \cdot 10^{-7}$	

VI. SIMULATION

This section the quadrotor dynamics in (6) and (7) are simulated with the guidance scheme from Section IV and the controllers from Section V. The gain parameters were chosen as $k_1 = 1/3$, $k_2 = 1$, $k_q = 10$, $k_\omega = 3$, $k_p = 1$ and $k_d = 2$. The quadrotor parameters can be found in Table VI. The desired trajectory is defined as

$$\boldsymbol{p}_d^n = \begin{bmatrix} c_1 \sin c_2 t & c_1 \cos c_2 t & c_3 - c_4 \sin c_5 t \end{bmatrix}^T \text{[m]}$$
$$\dot{\boldsymbol{p}}_d^n = \begin{bmatrix} c_1 c_2 \cos c_2 t & -c_1 c_2 \sin c_2 t & -c_4 c_5 \cos c_5 t \end{bmatrix}^T \text{[m/s]}$$
$$\ddot{\boldsymbol{p}}_d^n = \begin{bmatrix} -c_1 c_2^2 \sin c_2 t & -c_1 c_2^2 \cos c_2 t & c_4 c_5^2 \sin c_5 t \end{bmatrix}^T \text{[m/s}^2\text{]}$$

where $c_1 = 5$ m, $c_2 = 0.6$ rad/s, $c_3 = -10$ m, $c_4 = 0.5$ m and $c_5 = 0.3$ rad/s. The initial conditions were set as

$$\boldsymbol{q}_{n,b} = \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix}^T$$
$$\boldsymbol{\omega}_{n,b}^b = \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}^T \text{[rad/s]}$$
$$\boldsymbol{p}^n = \begin{bmatrix} 1 & 1.4 & 0.2 \end{bmatrix}^T \text{[m]}$$
$$\boldsymbol{v}^n = \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}^T \text{[m/s]}$$
$$\sigma_i = 0 \text{ [rad/s]}, \qquad i = 1, 2, 3, 4.$$

The thrust was limited to $0N \leq T \leq 30$N. It can be seen from Figure 2 and Figure 3 the quadrotor converges towards the desired trajectory and follows it. From Figure 5 it can be seen that the error never converges to $\boldsymbol{q}_I$ and the reason for this is the choice of k_1 which limits the tilt of the quadrotor to about 30°. This error could be made smaller by letting $k_1 \to 1$, but there comes a point where the quadrotor will tilt too much and will be unable to compensate for gravity when the quadrotor's height matches that of the desired height resulting in the quadrotor falling.

VII. CONCLUSIONS

In this paper a simplified model for a quadrotor using quaternions was presented. A quaternion based guidance scheme was presented which provides reference signals to the underlying attitude control system. Simulations were performed which shows that the quadrotor converges to and follows the desired trajectory. Future work would include formulating a stability proof using lyapunov analysis.

ACKNOWLEDGMENT

This research has been funded by the Norwegian Research Council and is part of the Arctic Earth Observation and Surveillance Technologies project 195142/I60.

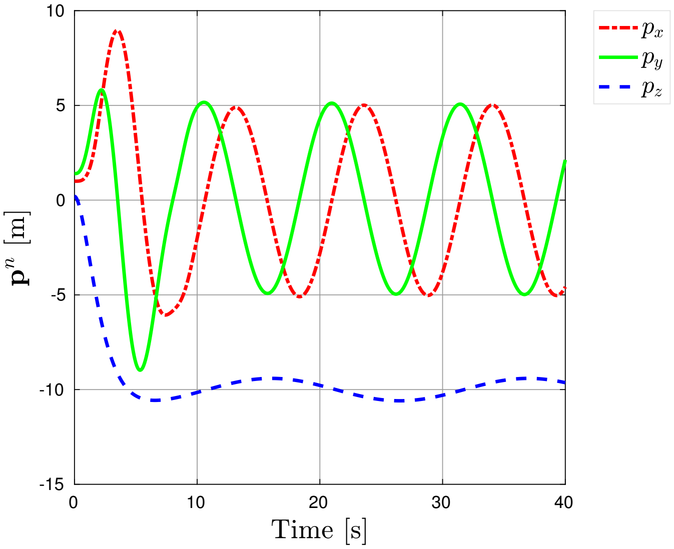

Fig. 2. Quadrotor's inertial position.

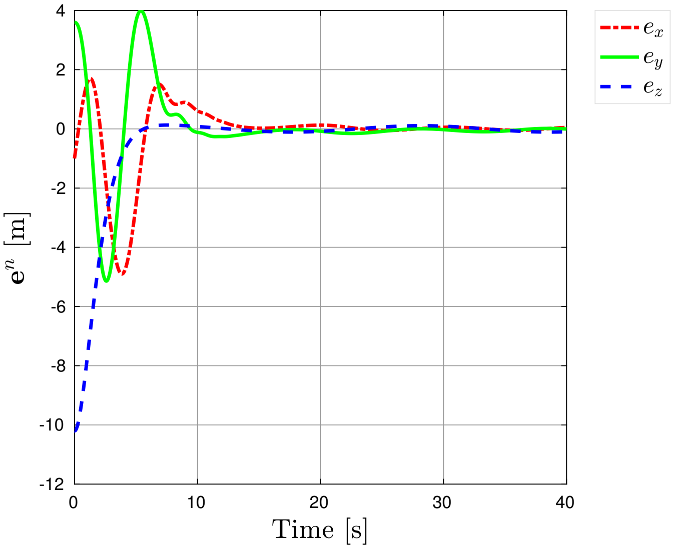

Fig. 3. Position error.

REFERENCES

[1] S. Bouabdallah, A. Noth, and R. Siegwart, "PID vs LQ control techniques applied to an indoor micro quadrotor," in *International Conference on Intelligent Robots and Systems*, vol. 3, 2004, pp. 2451–2456.

[2] T. Lee, M. Leoky, and N. H. McClamroch, "Geometric tracking control of a quadrotor UAV on SE(3)," in *IEEE Conference on Decision and Control*, 2010, pp. 5420–5425.

[3] S. Bouabdallah and R. Siegwart, "Full control of a quadrotor," in *International Conference on Intelligent Robots and Systems*, 2007, pp. 153–158.

[4] D. Mellinger and V. Kumar, "Minimum snap trajectory generation and control for quadrotors," in *Robotics and Automation*, 2011, pp. 2520–2525.

[5] M. Bangura and R. Mahony, "Nonlinear dynamic modeling for high performance control of a quadrotor," in *Proceedings of Australasian Conference on Robotics and Automation*, 2012.

[6] O. Egeland and J. T. Gravdahl, *Modeling and Simulation for Automatic Control*. Marine Cybernetics, 2002.

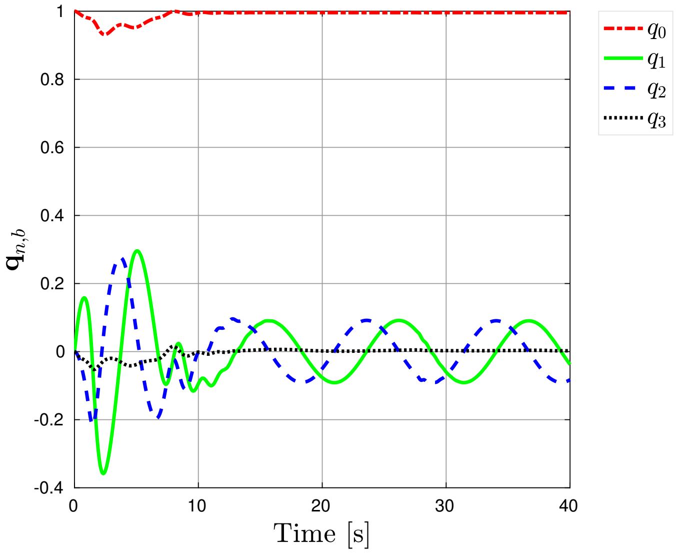

Fig. 4. Quadrotor attitude.

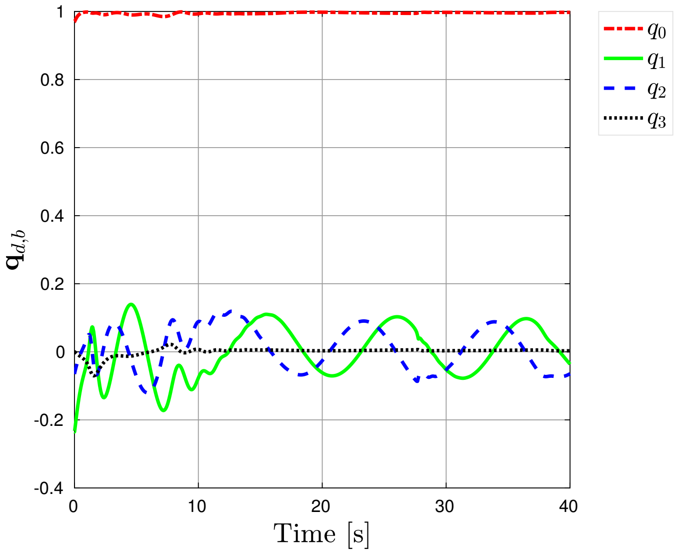

Fig. 5. The error attitude.

[7] E. B. Dam, M. Koch, and M. Lillholm, “Quaternion, interpolation and animation,” University of Copenhagen, Tech. Rep., 1998.

[8] M. D. Hua, T. Hamel, P. Morin, and C. Samson, “Introduction to feedback control of underactuated VTOL vehicles: A review of basic control design ideas and principles,” *IEEE Control Systems*, vol. 33, no. 1, pp. 61–75, Feb 2013.

[9] R. Mahony, V. Kumar, and P. Corke, “Multirotor aerial vehicles: Modeling, estimation and control of quadrotor,” *IEEE Robotics Automation Magazine*, vol. 19, no. 3, pp. 20 – 32, 2012.

[10] E. Oland, “A command-filtered backstepping approach to autonomous inspections using a quadrotor,” in *Mediterranean Conference on Control and Automation*, 2016, pp. 65–70.

[11] B. L. Stevens, F. L. Lewis, and E. N. Johnson, *Aircraft Control and Simulation: Dynamics, Control Design, and Autonomous Systems*. John Wiley & Sons, 2016.

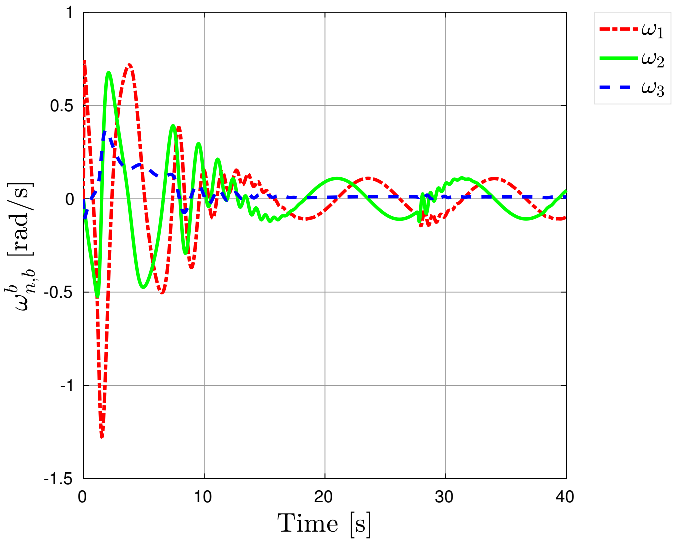

Fig. 6. Quadrotor angular velocities.

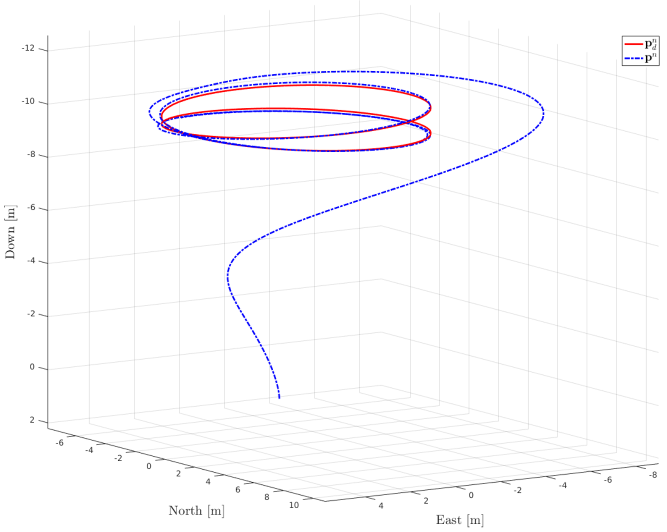

Fig. 7. Quadrotor converging to and following the desired trajectory.