

Full Quaternion Based Attitude Control for a Quadrotor

Emil Fresk and George Nikolakopoulos

Abstract—The aim of this article is to present a novel quaternion based control scheme for the attitude control problem of a quadrotor. A quaternion is a hyper complex number of rank 4 that can be utilized to avoid the inherent geometrical singularity when representing rigid body dynamics with Euler angles or the complexity of having coupled differential equations with the Direction Cosine Matrix (DCM). In the presented approach both the quadrotor's attitude model and the proposed non-linear Proportional squared (P^2) control algorithm have been implemented in the quaternion space, without any transformations and calculations in the Euler's angle space or DCM. Throughout the article, the merits of the proposed novel approach are being analyzed and discussed, while the efficacy of the suggested novel quaternion based controller are being evaluated by extended simulation results.

I. INTRODUCTION

The area of Unmanned Aerial Vehicles (UAV) and especially the one of having the capability for Vertical Take-Off and Landing (VTOL) as the quadrotor, has been in the focus of the research and development efforts, mainly due to their efficiency in accomplishing complex missions [1].

For achieving the desired performance most frequently the trajectory generation problem is being divided into two subproblems: a) the attitude problem and b) the translation problem, while as it has been proven [2, 3], these systems can be cascade interconnected. For the examined case of a quadrotor UAV, the position controller (translation) is generating reference attitude set-points for the attitude controller and thus the control problem of quadrotors has been confronted using several different approaches from research-leading teams worldwide, with famous works to include linear [4–6], and nonlinear controllers [7–11]. Although it has been proved that the aforementioned control strategies manage to stably navigate a quadrotor, the problem of designing optimized controllers that will be able to: a) provide fine-smoothed control actions for attitude stabilization and trajectory tracking, b) make use of model-knowledge for more accurate navigation, c) preserve robustness against sudden and unpredicted external disturbances, is still an open challenge. One of the constraints that the control engineers are facing, when dealing with the attitude problem is still a fundamental problem in dynamics due to the fact that finite rotation of a rigid body does not obey the laws of vector addition, e.g. commutativity, while the attitude characteristics of the rigid body cannot be extracted by integrating the body's angular velocity.

The authors are with the Department of Computer, Electrical and Space Engineering, Luleå University of Technology, SE-97187, Luleå, Sweden
Corresponding Author's email: geonik@ltu.se

However, when working with rotations, whether it's estimators or controllers, there has been one approach utilized more than any other, when creating models: the Newton–Euler equations [12], which is able to describe the combined translational and rotational dynamics of the rigid body. Although this modeling approach is considered a fundamental one, still it has three drawbacks. Firstly, it is solely based on Euler angles, which have the merit of being intuitive, but per definition these angles cannot define certain orientations as it suffers from singularities that result in an problem known as “gimbal lock” [13]. This problem is the loss of one degree of freedom in a three-dimensional space that occurs, when two of the rotational axes align and locking together. Secondly, it is very computationally expensive. Calculating sines and cosines takes a lot of performance and can very fast become unmanageable especially if it's being implemented on low cost hardware. Thirdly, when creating estimators or controllers that is necessary to utilize the Jacobian of the system states, the computational cost is even greater, as during these calculations some times all the matrix elements will have one or more sines or cosines to compute, which quickly can overwhelm the system.

For overcoming these problems, three solutions can be followed: a) guarantee that the system will keep inside the bounds of Euler angles, b) the utilization of a Direction Cosine Matrix (DCM) approach, and c) the quaternion approach. If the quadrotor has been only designed for simple stable flight, the first one might work, but in the case that unknown external disturbances (e.g. wind gusts) are being applied on the vehicle and result in flipping the aircraft (turning it upside-down), the Euler angle approach would not be able to compensate this. For the second approach DCM is constructed by translating the x , y and z body fixed coordinate system in a 3×3 matrix, while no matter of how this coordinate frame has been rotated, the matrix will still represent this transformation as the most significant merit of this approach is the non-suffering from the singularities that the Euler angles have. However, the DCM suffers from the constraint that each axis must be orthogonal to the other axes and should also be of a unit length. When rotating a DCM it must be multiplied with another DCM and the derivative of the DCM results in a 3×3 matrix and into a system of nine coupled differential equations (states) to solve (six if the 3rd axis is calculated from the cross product of the other two).

In the quaternion approach the previous mentioned limitations do not exist as one can directly translate a quaternion into a DCM and vice versa, however the quaternion and its corresponding derivative have four values and the only constraint is that it must be of unit length. This translates into

a system of only four coupled differential equations/states, greatly decreasing the computational cost and keeping the overall complexity low [14]. Due to the fact that the quaternion is a complex number it's sometimes hard to get an intuitive feeling for what it represents, but the direct coupling to a DCM makes the translation easy.

The novelty of this article stems from proposing a quaternion based non-linear P^2 controller, for solving the attitude problem of a quadrotor. In the proposed methodology, both the quadrotor modeling and the controller design will be made in the quaternion space, without the utilization of Euler angles or DCM. Until now in the relative scientific literature, only a few references are available utilizing quaternions controllers for the attitude stabilization problem as it has been described in [15], [16] and the references there in. These approaches, they do however convert the quaternion error back to Euler angles and regulate on these angles instead of regulating the quaternion directly. The drawback of these approaches is the fact that suffer from the non-linearities and singularities of Euler angles, plus the extra processing power needed to convert the Euler angles to quaternions and vice versa. As it is going to be analyzed in the sequel, the suggested novel control scheme does not suffer from these shortcomings, can be applied in the full three dimensional rotational domain, while it can be generalized as it is independent of the rigid body.

The rest of the article is structured as it follows. In Section II the fundamental properties and the corresponding algebra of the quaternion mathematics are being presented, while in Section III the quaternion based quadrotor modeling is analyzed. In Section IV the novel full quaternion based control scheme is established and in Section V extended simulation results that prove the efficiency of the proposed scheme are depicted. Finally, in Section VI conclusions are drawn.

II. QUATERNION MATH

For consistency reasons, and for building the mathematical background for following the proposed modeling and control scheme, this section is going to present the basic algebraic concepts behind the idea of quaternions. For a more comprehensive analysis and an in depth description of this mathematical tool, the reader is referred to the following publications [17] [18].

A quaternion is a hyper complex number of rank 4, which can be represented in many ways, while equations (1–2) represent two of the most popular approaches. The quaternion units from q_1 to q_3 are called the vector part of the quaternion, while q_0 is the scalar part.

$$\mathbf{q} = q_0 + q_1 \mathbf{i} + q_2 \mathbf{j} + q_3 \mathbf{k} \quad (1)$$

$$\mathbf{q} = [q_0 \ q_1 \ q_2 \ q_3]^T \quad (2)$$

Multiplication of two quaternions \mathbf{p} , \mathbf{q} is being performed by the Kronecker product, denoted as \otimes , and the outcome is presented in the following equations. If \mathbf{p} represents one rotation and \mathbf{q} represents another rotation $\mathbf{p} \otimes \mathbf{q}$ represents the combined rotation. It's important to note that quaternion

multiplication is non-commutative, just as rotations are non-commutative.

$$\begin{aligned} \mathbf{p} \otimes \mathbf{q} &= \begin{bmatrix} p_0 q_0 - p_1 q_1 - p_2 q_2 - p_3 q_3 \\ p_0 q_1 + p_1 q_0 + p_2 q_3 - p_3 q_2 \\ p_0 q_2 - p_1 q_3 + p_2 q_0 + p_3 q_1 \\ p_0 q_3 + p_1 q_2 - p_2 q_1 + p_3 q_0 \end{bmatrix} \\ \mathbf{p} \otimes \mathbf{q} &= Q(\mathbf{p})\mathbf{q} = \begin{bmatrix} p_0 & -p_1 & -p_2 & -p_3 \\ p_1 & p_0 & -p_3 & p_2 \\ p_2 & p_3 & p_0 & -p_1 \\ p_3 & -p_2 & p_1 & p_0 \end{bmatrix} \begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{bmatrix} \\ &= \bar{Q}(\mathbf{q})\mathbf{p} = \begin{bmatrix} q_0 & -q_1 & -q_2 & -q_3 \\ q_1 & q_0 & q_3 & -q_2 \\ q_2 & -q_3 & q_0 & q_1 \\ q_3 & q_2 & -q_1 & q_0 \end{bmatrix} \begin{bmatrix} p_0 \\ p_1 \\ p_2 \\ p_3 \end{bmatrix} \end{aligned}$$

The norm/length of a quaternion is defined, just as for any complex number, as depicted in equation (3). All quaternions in the presented approach are assumed to be of unitary length and thus are called unit quaternions.

$$\text{Norm}(\mathbf{q}) = \|\mathbf{q}\| = \sqrt{q_0^2 + q_1^2 + q_2^2 + q_3^2} \quad (3)$$

The complex conjugate of a quaternion has the same definition as normal complex numbers. The sign of the complex part is switched as in equation (4).

$$\text{Conj}(\mathbf{q}) = \mathbf{q}^* = [q_0 \ -q_1 \ -q_2 \ -q_3]^T \quad (4)$$

The inverse of a quaternion is defined as in equation (5), as the normal inverse of a complex number. Moreover, if the length of the quaternion is unitary then the inverse is the same as its conjugate.

$$\text{Inv}(\mathbf{q}) = \mathbf{q}^{-1} = \frac{\mathbf{q}^*}{\|\mathbf{q}\|^2} \quad (5)$$

The derivative of a quaternion requires some algebraic manipulation and can be represented as [18]: a) as in equation (6) in case that the angular velocity vector is in the fixed frame of reference, and b) as in equation (7) if the angular velocity vector is in the body frame of reference. It's important to note that these notations have been provided with respect to the left hand notation, and that for having them in the right hand notation the ω quaternion must be conjugated.

$$\dot{\mathbf{q}}_{\omega}(\mathbf{q}, \omega) = \frac{1}{2} \mathbf{q} \otimes \begin{bmatrix} 0 \\ \omega \end{bmatrix} = \frac{1}{2} Q(\mathbf{q}) \begin{bmatrix} 0 \\ \omega \end{bmatrix} \quad (6)$$

$$\dot{\mathbf{q}}_{\omega'}(\mathbf{q}, \omega') = \frac{1}{2} \begin{bmatrix} 0 \\ \omega' \end{bmatrix} \otimes \mathbf{q} = \frac{1}{2} \bar{Q}(\mathbf{q}) \begin{bmatrix} 0 \\ \omega' \end{bmatrix} \quad (7)$$

where $\omega = [\omega_x, \omega_y, \omega_z]^T$. If a quaternion is a unit quaternion it can be used as a rotation operator. However the transformation is not built up by only one quaternion multiplication but two, the normal and its conjugate, as shown in equation (8). This rotates the vector \mathbf{v} from the fixed frame to the body frame represented by \mathbf{q} .

$$\mathbf{w} = \mathbf{q} \otimes \begin{bmatrix} 0 \\ \mathbf{v} \end{bmatrix} \otimes \mathbf{q}^* \quad (8)$$

This rotation in equation (8) can be rewritten by replacing v with the x , y and z axis, as it is being displayed in the following equations (9-10) and (11).

$$R_x(\mathbf{q}) = \mathbf{q} \otimes \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} \otimes \mathbf{q}^* = \begin{bmatrix} q_0^2 + q_1^2 - q_2^2 - q_3^2 \\ 2(q_1q_2 + q_0q_3) \\ 2(q_1q_3 - q_0q_2) \end{bmatrix} \quad (9)$$

$$R_y(\mathbf{q}) = \mathbf{q} \otimes \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} \otimes \mathbf{q}^* = \begin{bmatrix} 2(q_1q_2 - q_0q_3) \\ q_0^2 - q_1^2 + q_2^2 - q_3^2 \\ 2(q_2q_3 + q_0q_1) \end{bmatrix} \quad (10)$$

$$R_z(\mathbf{q}) = \mathbf{q} \otimes \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \otimes \mathbf{q}^* = \begin{bmatrix} 2(q_1q_3 + q_0q_2) \\ 2(q_2q_3 - q_0q_1) \\ q_0^2 - q_1^2 - q_2^2 + q_3^2 \end{bmatrix} \quad (11)$$

It should be noted that in the examined case, only the vector part of the quaternion has been extracted, resulting in a rotation matrix, which rotates a point in a fixed coordinate system, as depicted in equation (12). When rotating a coordinate system, the angle sign changes and provides equation (13), while the same result arises when conjugating the quaternion in equation (8).

$$R(\mathbf{q}) = [R_x(\mathbf{q}) \quad R_y(\mathbf{q}) \quad R_z(\mathbf{q})] \quad (12)$$

$$R(\mathbf{q}) = \begin{bmatrix} R_x(\mathbf{q})^T \\ R_y(\mathbf{q})^T \\ R_z(\mathbf{q})^T \end{bmatrix} \quad (13)$$

The rotation can also be represented using a rotation vector as denoted in equation (14), where \mathbf{u} is the rotation axis (unit vector) and α is the angle of rotation. Using this notation can have many benefits when creating an error or specifying a reference as it has a direct physical connection.

$$\mathbf{q} = \cos\left(\frac{\alpha}{2}\right) + \mathbf{u} \sin\left(\frac{\alpha}{2}\right) \quad (14)$$

Finally, for representing quaternion rotations in a more intuitive manner, the conversion from Euler angles to quaternion and from quaternion to Euler angle can be performed by utilizing the following two equations respectively. This property is very useful in case that the aim is to represent an orientation in angles, while retaining the overall dynamics of the system in a quaternion form.

$$\mathbf{q} = \begin{bmatrix} \cos(\phi/2) \cos(\theta/2) \cos(\psi/2) + \sin(\phi/2) \sin(\theta/2) \sin(\psi/2) \\ \sin(\phi/2) \cos(\theta/2) \cos(\psi/2) - \cos(\phi/2) \sin(\theta/2) \sin(\psi/2) \\ \cos(\phi/2) \sin(\theta/2) \cos(\psi/2) + \sin(\phi/2) \cos(\theta/2) \sin(\psi/2) \\ \cos(\phi/2) \cos(\theta/2) \sin(\psi/2) - \sin(\phi/2) \sin(\theta/2) \cos(\psi/2) \end{bmatrix} \quad (15)$$

$$\begin{bmatrix} \phi \\ \theta \\ \psi \end{bmatrix} = \begin{bmatrix} \text{atan2}(2(q_0q_1 + q_2q_3), q_0^2 - q_1^2 - q_2^2 + q_3^2) \\ \text{asin}(2(q_0q_2 - q_3q_1)) \\ \text{atan2}(2(q_0q_3 + q_1q_2), q_0^2 + q_1^2 - q_2^2 - q_3^2) \end{bmatrix} \quad (16)$$

III. QUATERNION BASED QUADROTOR MODELING

For modeling the attitude dynamics of the quadrotor, as the one depicted in Figure 1, it has been assumed that the structure is rigid and symmetrical, the center of gravity and

the body fixed frame origin coincide, the propellers are rigid, the bias throttle to counteract the effect of gravity can be neglected and only the differential forces created by the propellers has an effect on rotation.

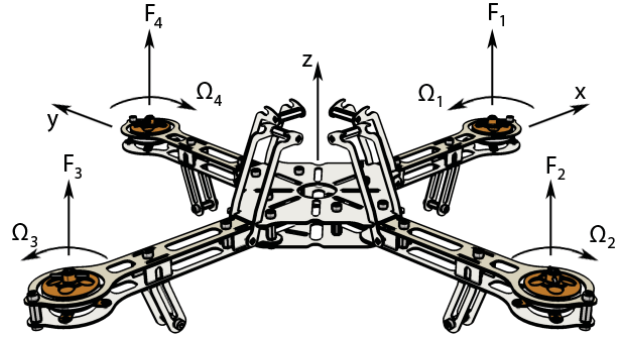


Fig. 1. A sketch of the Luleå University of Technology Quadrotor without propellers attached. Ω_{1-4} denotes the rotational speed of each motor, F_{1-4} is the force generated by each motor and x , y and z is the body fixed coordinate system.

For modeling the physics of the quadrotor two alternative approaches could be followed: a) the full physical model can be derived by the utilization of Newtons' laws of motion and producing a frame dependent model, or b) to use the Euler-Newton equations for translational and rotational dynamics of a rigid body, as in equation (17). The utilization of the second approach greatly simplifies the derivation of the model as the only unknown in the derived model is the connection among the control signal and the corresponding torque.

$$\begin{bmatrix} F \\ \tau \end{bmatrix} = \begin{bmatrix} m & 0 \\ 0 & I_{cm} \end{bmatrix} \begin{bmatrix} a_{cm} \\ \dot{\omega} \end{bmatrix} + \begin{bmatrix} 0 \\ \omega \times (I_{cm} \cdot \omega) \end{bmatrix}, \quad (17)$$

where ω is defined as:

$$\omega = \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix}$$

For deriving the full dynamics of the quadrotor's rigid body rotations, the right hand quaternion derivative from equation (7) should be combined with the rotation dynamics from equation (17), which results in an equation system describing the entire rotation dynamics of the Quadrotor on a quaternion form, as it has been also presented in equation (18):

$$\begin{cases} \dot{\mathbf{q}} = -\frac{1}{2} \begin{bmatrix} 0 \\ \omega \end{bmatrix} \otimes \mathbf{q} \\ \dot{\omega} = I_{cm}^{-1} \cdot \tau - I_{cm}^{-1} [\omega \times (I_{cm} \cdot \omega)] \end{cases} \quad (18)$$

When modeling the control signal to torque relation, various linear and non-linear system approximations [12] have been proposed. In the presented approach this relation has been simplified to identity matrix. A more detailed approach, taking under consideration the physics behind rotors' dynamics can be taken under consideration without losing the impact of the presented approach. The obtained results

in this Section can be directly generalized for modeling the attitude dynamics of other types of UAV frames than the Quadrotor, as long as the control signal to torque relationship can be found.

IV. CONTROLLER SYNTHESIS

In this section a feedback control scheme for the attitude stabilization of the quadrotor aircraft will be presented. Initially and in order to be able to propose a proper control scheme, the inputs and outputs of the system must be a priori known. The system presented in equation (18) suggests that measurements of the quaternion and angular rates are needed in order to measure the system's state and calculate the necessary driving torque to rotate it.

As it has been stated before, the novelty of this article stems from proposing a control scheme for the attitude problem of a quadrotor completely in the quaternion space. In the proposed approach, all the measurements and the calculations have been made by utilizing quaternions, while no transformation to Euler angles and rotations have been executed. The resulting controller has no issues with singularities and can be straightforwardly implemented, while retaining simplicity.

For utilizing quaternions in the error calculation between the desired \mathbf{q}_{ref} and the measured quaternion based response of the quadrotor \mathbf{q}_m , an error quaternion, denoted as \mathbf{q}_{err} , should be calculated. This is done by multiplying the reference, \mathbf{q}_{ref} , with the conjugate of the estimated quaternion, \mathbf{q}_m , as it has been presented in equation (19). This Kronecker product will calculate the difference quaternion, which can be utilized to produce the error around each axis of rotation or:

$$\mathbf{q}_{err} = \mathbf{q}_{ref} \otimes \mathbf{q}_m^* \quad (19)$$

The vector part directly connects the quaternion to the sine of the error from equation (14), which results in an axis error as depicted in equation (20). In case that the reference is demanding a rotation more than π radians, the closest rotation is the inverted direction and this is found by examining q_0 . If $q_0 < 0$ then the desired orientation is more than π radians away and the closest rotation is the conjugate of \mathbf{q}_{err} , negating the axis error in equation (20).

$$\text{Axis}_{err} = \begin{bmatrix} q_1^{err} \\ q_2^{err} \\ q_3^{err} \end{bmatrix} \quad (20)$$

When designing the controller the simplest form available has been chosen, which is a non-linear P^2 -controller formulation, as it has been depicted in Figure 2. In the proposed approach, an inner loop proportional controller P_ω for the angular velocity and an outer loop proportional controller P_q for the angular velocity reference tracking, have been effectively combined for creating a non-linear P^2 -controller for the attitude regulation problem. The overall mathematical formulation of the proposed P^2 control scheme is being

denoted as it follows:

$$\tau = -P_q \cdot \begin{bmatrix} q_1^{err} \\ q_2^{err} \\ q_3^{err} \end{bmatrix} - P_\omega \cdot \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} \quad (21)$$

It should be noted that equation (21) is derivative free and straightforward to be implemented (low computational cost). Moreover, the noise immunity of this design is only as good as the measured/estimated quaternion and angular velocity, as the suggested scheme will directly amplify noise just as much as the corresponding errors. Something that is worth

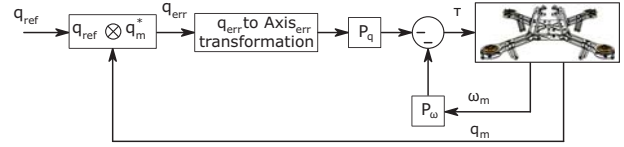


Fig. 2. Block diagram of the full nonlinear P^2 quaternion based control scheme

noting is the fact that the P^2 design will always drive the error to zero thanks to the double integrator in the non-linear dynamics. This reduces controller complexity and without an integrator there is no negative phase shift added from the controller.

V. SIMULATION RESULTS

All simulations have been carried out on the nonlinear quadrotor model presented in [10], which takes into account a wide set of the aerodynamic forces and moments acting on the system, including the hub and friction forces, the rolling moments and –up to some extent– the variations in the aerodynamic coefficients, due to the motion of the quadrotor inside the atmosphere.

The parameters of the quadrotor model have been set as $I_{xx} = I_{yy} = 6.5 \cdot 10^{-4} \text{kg} \cdot \text{m}^2$ and $I_{zz} = 1.2 \cdot 10^{-3} \text{kg} \cdot \text{m}^2$. The aforementioned values correspond to the CAD model analysis presented in Figure 1, with respect to the analysis provided in [19]. The proposed quaternion based controller have been evaluated under three fundamental tracking test cases which are: a) constant rotation - step input, b) periodical reference - sinusoidal input, and c) complex maneuver - flip. In all the simulated cases additive corrupting zero mean noise of 0.1 amplitude affecting the measurements have been considered. Bounds on the control action have been applied, for performing a more realistic evaluation. These bounds have been set as $\pm 4 \text{Nm}$ for all the motors, while the gains of the nonlinear P^2 controller have been set as: $P_q = 20$ and $P_\omega = 4$ after small fine tuning in simulations.

In the first case a step response has been considered where an one rad reference step around each axis has been requested, at different time instances. The results obtained from each axis can be depicted in Figure 3, while the corresponding control action is provided in Figure 4.

From the obtained responses, it can be observed that the proposed control scheme performs very well with a very small overshoot and a very good reference tracking. All

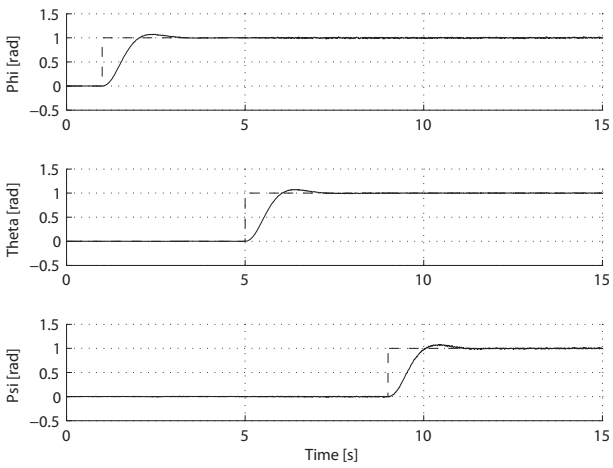


Fig. 3. Quadrotor step responses. The reference signal and the system's responses have been denoted by the dashed and solid lines respectively. All the graphs have been indicated in radians for more intuitive display.

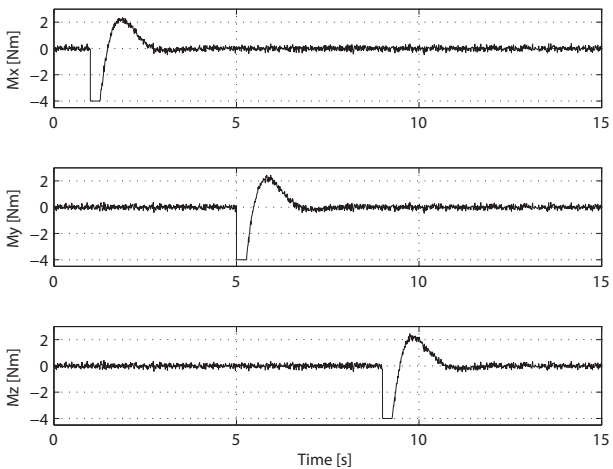


Fig. 4. Controller effort during step responses

errors go quickly to zero and no strange effects can be witnessed from the effect of noise or the non-linearities. To provide a realistic simulated evaluation, bounds on the control action have been considered. The control signal saturates when the step is introduced but quickly goes back to its linear region. Although the existence of the corrupting noise, the quadrotor's performance has not been significantly affected mainly to the effect of the double integrator dynamics of the system.

The second evaluation test-case has considered the problem of tracking a 0.5 radian amplitude sine wave with a frequency of 1 radian/s. The applied referenced waves were phase shifted on the y and z axis for driving the torques not be in the same phase. The tracking results obtained are depicted in Figure 5, while the corresponding control actions are being displayed in Figure 6.

From the obtained results it is obvious that, as in the previous testing case, the proposed control scheme is able

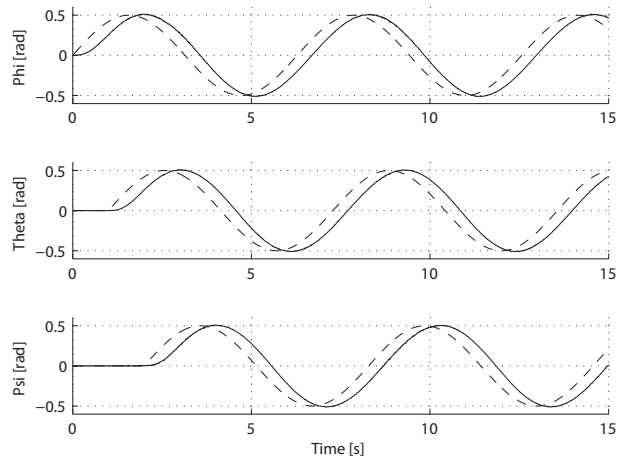


Fig. 5. Controller tracking an 1 rad/s sine wave. Input/output shown in radians for more intuitive display. The dashed line is reference and the continuous line is output.

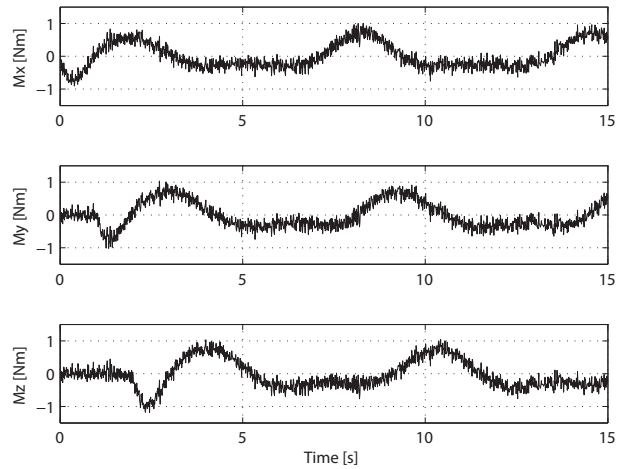


Fig. 6. Controller control action during tracking.

to provide a very good and fast tracking with a small phase shift of about 0.5 seconds. Also it should be stated that the proposed controller has the merit of performing quite big and dynamic changes in the quadrotor's attitude, without the problem of saturating the control signals for tracking these fast changes. The effects of the corrupting noise are more identifiable when the amplitude of the control effort is small, due to the fact that the gains of the quaternion controller have been fixed, while this noise has no direct and severe effect on the overall controlled quadrotor.

For the final evaluation test-case, a most popular maneuver has been considered as well, which is the 360 degree flip. The control signal was generated by ramping from 0 to 2π radians so the controller wouldn't take the shortest route, while the simulation result are depicted in Figure 7. From observing the response of the quadrotor during flip it is obvious that the maneuver has been executed without any problem. In the case of the Euler angles, this flip would have been subjected to massive non-linearities but, as seen in Figure

8, the quaternion has no non-linearities nor singularities and thus can therefore perform the flip without any problem. The control signal does only saturate a little and this is because the controller has been tuned for smooth transitions (small gain values). For a faster flip a more aggressive tuning could be used or a direct control on the angular velocity reference could be applied until the attitude has made half the flip and then reconnect the attitude controller.

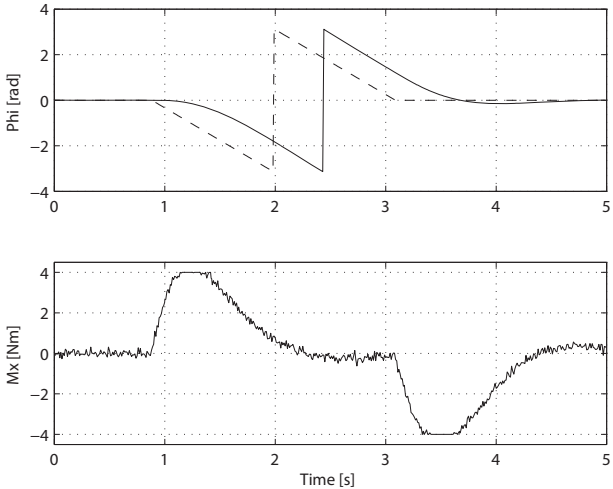


Fig. 7. Controller doing a flip. Input/output shown in radians for more intuitive display. The dashed line is reference and the continuous line is output.

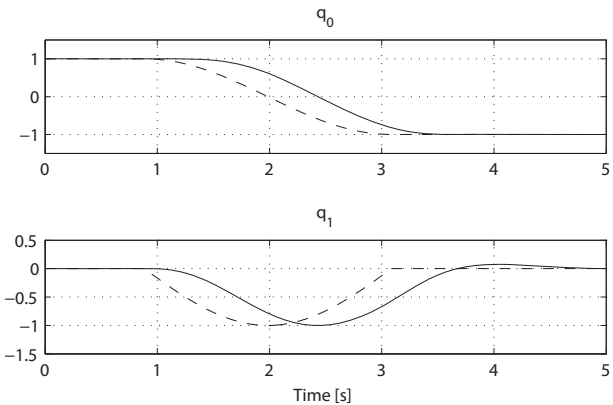


Fig. 8. Controller doing a flip (displayed in quaternion data q_0 and q_1). The dashed line is reference and the continues line is output.

VI. CONCLUSIONS

In this article a full quaternion based attitude controller for a quadrotor has been presented. The overall aim and the novelty of the article is to present a modeling and control scheme development approach, fully implemented in the quaternion space for avoiding the the non-linearities and the computational cost usually connected with the Euler and rotation angles. The presented approach, has been applied directly to the quaternion error, without any conversions for

solving the attitude problem. Extended simulation results have been presented that proved the efficiency of the suggested scheme.

REFERENCES

- [1] K. Alexis, G. Nikolakopoulos, A. Tzes, and L. Dritsas, "Coordination of helicopter UAVs for aerial Forest-Fire surveillance," in *Applications of Intelligent Control to Engineering Systems*. Springer Netherlands, June 2009, pp. 169–193.
- [2] G. Raffo, M. Ortega, and F. Rubio, "An integral predictive/nonlinear control structure for a quadrotor helicopter," *Automatica*, vol. 46, no. 1, pp. 29–39, 2010.
- [3] K. Alexis, G. Nikolakopoulos, and A. Tzes, "Switching model predictive attitude control for a quadrotor helicopter subject to atmospheric disturbances," *Control Engineering Practice*, vol. 19, no. 10, pp. 1995–2007, October 2011.
- [4] G. M. Hoffmann, H. Huang, S. L. Waslander, and C. J. Tomlin, "Quadrotor helicopter flight dynamics and control: Theory and experiment," in *Proc. of the American Institute of Aeronautics and Astronautics (AIAA) Guidance, Navigation, and Control Conference*, SC, USA, 2007.
- [5] S. Bouabdallah, A. Noth, and R. Siegwart, "PID vs LQ control techniques applied to an indoor micro quadrotor," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, Sendai, Japan, 2004, pp. 2451–2456.
- [6] J. Gadewadikar, F. Lewis, K. Subbarao, K. Peng, and B. Chen, "H-Infinity static output-feedback control for rotorcraft," *Journal of Intelligent and Robotic Systems*, vol. 54, no. 4, pp. 629–646, Apr. 2009.
- [7] R. Mahony, V. Kumar, and P. Corke, "Multirotor aerial vehicles: Modeling, estimation and control of quadrotor," *IEEE Robotics and Automation Magazine*, vol. 19, no. 3, pp. 20–32, September 2012.
- [8] A. Benallegue, A. Mokhtari, and L. Fridman, "Feedback linearization and high order sliding mode observer for a quadrotor UAV," in *International Workshop on Variable Structure Systems (VSS'06)*, Alghero, Sardinia, 2006, pp. 365–372.
- [9] S. L. Waslander, G. M. Hoffmann, J. S. Jang, and C. J. Tomlin, "Multi-agent quadrotor testbed control design: Integral sliding mode vs. reinforcement learning," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robotics and Systems*, Alberta, Canada, 2005, pp. 468–473.
- [10] S. Bouabdallah and R. Siegwart, "Full control of a quadrotor," in *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2007, pp. 153–158.
- [11] A. Das, F. Lewis, and K. Subbarao, "Backstepping approach for controlling a quadrotor using lagrange form dynamics," *Journal of Intelligent and Robotic Systems*, vol. 56, no. 1, pp. 127–151, 2009.
- [12] T. Bresciani, "Modelling, identification and control of a quadrotor helicopter," Ph.D. dissertation, Lund University, 2008.
- [13] J. Harrison, J. Gallagher, and E. Grace, "An algorithm providing all-attitude capability for three-gimballed inertial systems," *IEEE Transactions on Aerospace and Electronic Systems*, vol. AES-7, no. 3, pp. 532–543, 1971.
- [14] J. T. T. H. . T. C. Grant Baldwin, Robert Mahony, "Complementary filter design on the special euclidean group so(3)," *IEEE Conference on Decision and Control*, vol. 44, 2005.
- [15] E. Stingu and F. Lewis, "Design and implementation of a structured flight controller for a 6dof quadrotor using quaternions," in *17th Mediterranean Conference on Control and Automation*, Thessaloniki, Greece, 2009.
- [16] K. A. W. J.-Y. Joshi, S.M., "Robust attitude stabilization of spacecraft using nonlinear quaternion feedback," *IEEE Transactions on Automatic Control*, vol. 40, 1995.
- [17] J. B. Kuipers, *Quaternions and Rotation Sequences*, P. U. Press, Ed., 1998.
- [18] J. Diebel, "Representing attitude: Euler angles, unit quaternions, and rotation vectors," 2006.
- [19] A. Tayebi and S. McGilvray, "Attitude stabilization of a VTOL quadrotor aircraft," *IEEE Transactions on Control Systems Technology*, vol. 14, no. 3, pp. 562–571, 2006.