Report

Nonlinear Quadrocopter Attitude Control Technical Report

Author(s):
Brescianini, Dario; Hehn, Markus; D'Andrea, Raffaello

Publication Date:
2013

Permanent Link:

https://doi.org/10.3929/ethz-a-009970340 →

Nonlinear Quadrocopter Attitude Control

Technical Report

Dario Brescianini, Markus Hehn and Raffaello D'Andrea
{bdario,hehnm,rdandrea}@ethz.ch

Institute for Dynamic Systems and Control (IDSC)
ETH Zürich

October 10, 2013

Abstract

This report addresses the issue of nonlinear attitude control for quadrocopters. A globally stable attitude controller based on unit quaternions is proposed and its stability and robustness is proven. In addition, a method to prioritize the control of the crucial pitch and roll angles over yaw is introduced. Finally, experiments with the proposed nonlinear controller are conducted which demonstrate its performance.

1 Introduction

Quadrocopters are a popular research subject and a vast variety of literature on quadrocopter attitude control exists. Often, the attitude is described using Euler angles, which is a very natural way of describing orientation, especially for fixed-wing aircrafts. Controllers are then derived from linearizations of the nonlinear attitude control problem around hover conditions. This yields reasonable results for small pitch and roll angles. More recently, the research has been directed towards high maneuverability. For aggressive maneuvers, i.e. extreme changes in attitude, however, these locally stable controllers have significant deficiencies which yield poor performance or even worse, failure to stabilize the vehicle.
In order to exploit the full potential of quadrocopters, globally stable attitude controllers should be applied. Furthermore, Euler angles give rise to singularities which have to be avoided. This limits the set of all possible motions for purely mathematical reasons which have nothing to do with the physical constraints of the quadrocopter and thus Euler angles are not appropriate for global attitude control. It is crucial for global attitude control that a parameterization is chosen which is able to represent attitudes globally and is neither geometrically nor kinematically singular. In [2], a survey of attitude descriptions with their suitability for rigid body attitude control is given. Commonly, globally stable attitude controllers are based on unit quaternions [14, 4, 13, 12], but there exist also some controllers representing the attitude using rotation matrices [2, 9, 10].
This report is the result of a semester project carried out in fall 2011, with the objective of developing and implementing a globally stable attitude controller for the quadrocopters used in the Flying Machine Arena[1]. It is now published as a technical report to serve as a reference for the applied attitude controller in the Flying Machine Arena.
The remainder of this report is organized as follows: In Section 2, background material on attitude representation and the quadrocopter setup is provided. A globally asymptotically stable control law is then proposed in Section 3. Finally, experimental results for the proposed controller are shown in Section 4 and the report is concluded in Section 5.

2 Preliminaries

2.1 Attitude Representation

The attitude of a rigid body can be described by a rotation between a reference coordinate frame and a body-fixed coordinate frame. There exist several rotation parametrizations such as Euler angles (3 parameters), unit quaternions (4 parameters) and rotation matrices (9 parameters). All three-parameter representations suffer from singularities and all k-parameter representations ($k > 3$) have $k - 3$ constraints [11].

2.1.1 Rotation Matrix

A rotation R_{BI} is a linear transformation which maps the linear space $\mathcal{I} \mathrel{\widehat{=}} \mathbb{R}^3$ to the linear space $\mathcal{B} \mathrel{\widehat{=}} \mathbb{R}^3$ while preserving length and right-handedness. From these properties, it follows directly that $R_{BI} \in SO(3)$. Consider the orthogonal bases $\vec{\bar{e}}^I \in \mathcal{I}$ and $\vec{\bar{e}}^B \in \mathcal{B}$, where $\vec{\bar{e}}^I$ is defined as

$$\vec{\bar{e}}^I := \left(\vec{e}^I_x, \vec{e}^I_y, \vec{e}^I_z\right) \tag{1}$$

and a vector $\vec{r}$ expressed in the coordinate frames $\mathcal{I}$ and $\mathcal{B}$, i.e. ${}_I\vec{r}$ and ${}_B\vec{r}$, respectively. Then the linear transformation R_{BI} that maps a vector ${}_I\vec{r}$ to ${}_B\vec{r}$

$${}_B\vec{r} = R_{BI}\,{}_I\vec{r}, \tag{2}$$

[1] http://www.flyingmachinearena.org

is given by a 3×3 matrix

$$R_{BI} = \left({}_B\vec{e}_x^I, {}_B\vec{e}_y^I, {}_B\vec{e}_z^I \right). \tag{3}$$

The column vectors of R_{BI} are the three orthogonal unit basis vectors of frame $\mathcal{I}$ represented in frame $\mathcal{B}$. Similarly, the three row vectors correspond to the basis vectors of frame $\mathcal{B}$ represented in frame $\mathcal{I}$.

2.1.2 Euler Angles

Euler angles are based on the fact that any rotation matrix can be constructed by consecutively applying three elementary rotation matrices. The nine entries of the rotation matrix can then be parametrized by the rotation angles of the three elementary rotations. A very common rotation sequence for aircraft applications is the ZYX-sequence (Fig. 1):

1. Rotation about the initial z-axis by yaw angle ψ ($\mathcal{I} \rightarrow \mathcal{K}$).
2. Rotation about the new (rotated) y-axis by pitch angle θ ($\mathcal{K} \rightarrow \mathcal{L}$).
3. Rotation about the new (rotated) x-axis by roll angle ϕ ($\mathcal{L} \rightarrow \mathcal{B}$).

R_{BI} can be written as

$$\begin{aligned} R_{BI} &= R_{BL} R_{LK} R_{KI} && (4) \\ &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & c_\phi & s_\phi \\ 0 & -s_\phi & c_\phi \end{bmatrix} \begin{bmatrix} c_\theta & 0 & -s_\theta \\ 0 & 1 & 0 \\ s_\theta & 0 & c_\theta \end{bmatrix} \begin{bmatrix} c_\psi & s_\psi & 0 \\ -s_\psi & c_\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} && (5) \\ &= \begin{bmatrix} c_\theta c_\psi & c_\theta s_\psi & -s_\theta \\ -c_\phi s_\psi + s_\phi s_\theta c_\psi & c_\phi c_\psi + s_\phi s_\theta s_\psi & s_\phi c_\theta \\ s_\phi s_\psi + c_\phi s_\theta c_\psi & -s_\phi c_\psi + c_\phi s_\theta s_\psi & c_\phi c_\theta \end{bmatrix} && (6) \end{aligned}$$

where $c_{(\cdot)}$ and $s_{(\cdot)}$ are abbreviations for $\cos(\cdot)$ and $\sin(\cdot)$ respectively. The inverse mapping from rotation matrix to Euler angles is then

$$\begin{aligned} \phi &= \arctan2\left(R_{23}, R_{33}\right) && (7) \\ \theta &= -\arcsin\left(R_{13}\right) && (8) \\ \psi &= \arctan2\left(R_{12}, R_{11}\right) && (9) \end{aligned}$$

with R_{ij} the entry in the i-th row and j-th column of R_{BI}.

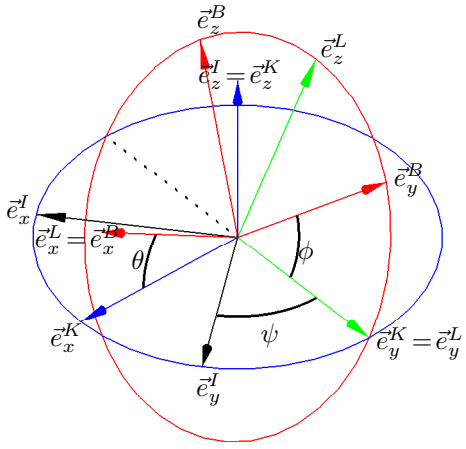

Figure 1: ZYX-Euler angles.

2.1.3 Unit quaternion

Every rotation can be parametrized by a single rotation about a fixed axis described by the unit vector $\vec{k}$ and a rotation angle α. This is known as eigenaxis rotation and embodies the shortest rotation between two orientations. Based on this, a unit quaternion is defined as

$$\mathbf{q} = \begin{bmatrix} q_0 & q_1 & q_2 & q_3 \end{bmatrix}^\top = \begin{bmatrix} q_0 \\ \mathbf{q}_{1:3} \end{bmatrix} = \begin{bmatrix} \cos(\frac{\alpha}{2}) \\ \vec{k}\sin(\frac{\alpha}{2}) \end{bmatrix}. \tag{10}$$

The adjoint, norm and inverse of a quaternion $\mathbf{q}$ are

$$\bar{\mathbf{q}} = \begin{bmatrix} q_0 \\ -\mathbf{q}_{1:3} \end{bmatrix}, \tag{11}$$

$$\|\mathbf{q}\| = \sqrt{q_0^2 + q_1^2 + q_2^2 + q_3^2}, \tag{12}$$

$$\mathbf{q}^{-1} = \frac{\bar{\mathbf{q}}}{\|\mathbf{q}\|}. \tag{13}$$

The multiplication of two quaternions q and p is then defined[2] by

$$\mathbf{q} \cdot \mathbf{p} := Q(\mathbf{q})\mathbf{p} \tag{14}$$

where

$$Q(\mathbf{q}) := \begin{bmatrix} q_0 & -q_1 & -q_2 & -q_3 \\ q_1 & q_0 & -q_3 & q_2 \\ q_2 & q_3 & q_0 & -q_1 \\ q_3 & -q_2 & q_1 & q_0 \end{bmatrix}. \tag{15}$$

Note that the quaternion, which corresponds to a rotation matrix $\mathbf{I}$ is given by

$$\mathbf{q_I} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}. \tag{16}$$

The rotation of a vector $\vec{r}$ by a quaternion $\mathbf{q}$ is computed by

$$\mathbf{p}(\vec{r}_r) = \mathbf{q} \cdot \mathbf{p}(\vec{r}) \cdot \bar{\mathbf{q}} \tag{17}$$

where $\vec{r}_r$ is the rotated vector and $\mathbf{p}(\cdot)$ is the quaternion representation of a vector:

$$\mathbf{p}(\vec{r}) = \begin{bmatrix} 0 \\ \vec{r} \end{bmatrix}. \tag{18}$$

The mapping from Euler angles to unit quaternions for the ZYX-sequence is

$$\mathbf{q}(\psi, \theta, \phi) = \begin{bmatrix} c_{\frac{\phi}{2}} c_{\frac{\theta}{2}} c_{\frac{\psi}{2}} + s_{\frac{\phi}{2}} s_{\frac{\theta}{2}} s_{\frac{\psi}{2}} \\ -c_{\frac{\phi}{2}} s_{\frac{\theta}{2}} s_{\frac{\psi}{2}} + c_{\frac{\theta}{2}} c_{\frac{\psi}{2}} s_{\frac{\phi}{2}} \\ c_{\frac{\phi}{2}} c_{\frac{\psi}{2}} s_{\frac{\theta}{2}} + s_{\frac{\phi}{2}} c_{\frac{\psi}{2}} s_{\frac{\psi}{2}} \\ c_{\frac{\phi}{2}} c_{\frac{\theta}{2}} s_{\frac{\psi}{2}} - s_{\frac{\phi}{2}} c_{\frac{\psi}{2}} s_{\frac{\theta}{2}} \end{bmatrix}. \tag{19}$$

[2]Note: For ease of interpretation, the quaternion multiplication used in this report differs from the one defined in [3]. With the definition in (14), the quaternion multiplication $\mathbf{q} \cdot \mathbf{p}$ corresponds to first a rotation of $\mathbf{q}$ and afterwards a rotation of $\mathbf{p}$ in the new coordinate system.

2.1.4 Remarks on attitude representation

Both, rotation matrices and unit quaternions, offer singularity-free representations of attitude. Note however, that the space of unit quaternions $\mathbb{S}^3$ double covers the space of physical attitudes $SO(3)$. Hence, unit quaternions are not unique. In fact, each pair of antipodal unit quaternions $\pm\mathbf{q} \in \mathbb{S}^3$ corresponds to the same physical attitude [2]. For attitude control tasks in $\mathbb{S}^3$, this implies that a controller has to be designed which stabilizes a disconnected set of equilibrium points. Nevertheless, unit quaternions are often the chosen parametrization in practise for attitude control (including this report) since they are a *minimal* globally non-singular parametrization. Furthermore, unit quaternions offer an easy insight on what is happening on a geometrical level.

2.2 Quadrocopter

In the following, we consider a quadrocopter with a body-fixed frame $\mathcal{B}$ and an inertial frame $\mathcal{I}$ as shown in Fig. 2. The inputs to the quadrocopter are the four motor thrust forces. The

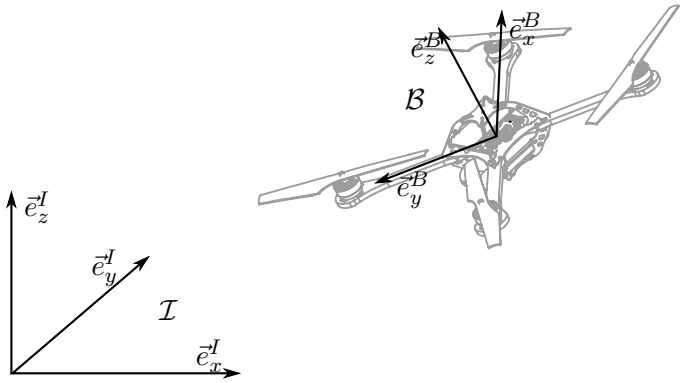

Figure 2: Quadrocopter with the body-fixed coordinate frame $\mathcal{B}$ and a reference frame $\mathcal{I}$.

outputs are the position, velocity and acceleration as well as the attitude and angular body rate. As the model has more outputs than inputs, it is clear that not all outputs can be controlled independently. In the FMA, a cascaded control structure is used to control the quadrocopters, with a position controller as the outermost control loop (Fig. 3). This control approach is

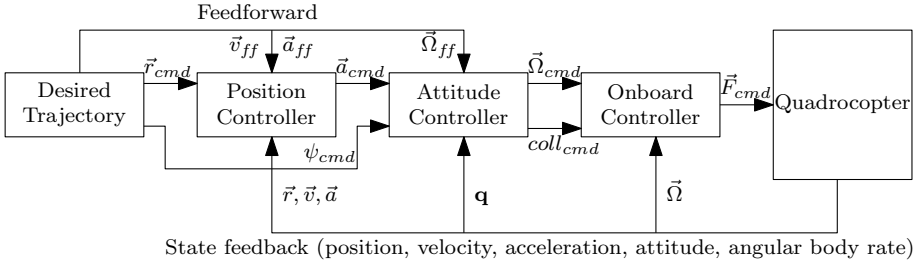

Figure 3: Cascaded control architecture.

known as time-scale separation and is valid as long as the inner loops are significantly faster than the outer loops. In this report, only the attitude control loop is taken into consideration. It is assumed that the dynamics of the onboard control loop and the quadrocopter are much faster such that the system to be controlled can be modelled as a rigid body with the desired

angular body rate $\vec{\Omega}$ as direct input. The quadrocopter dynamics can then be modelled as

$$\dot{\mathbf{q}} = \begin{bmatrix} \dot{q_0} \\ \dot{\mathbf{q}}_{1:3} \end{bmatrix} = \frac{1}{2}\mathbf{q} \cdot \mathbf{p}\left(\vec{\Omega}\right) = \begin{bmatrix} -\frac{1}{2}\mathbf{q}_{1:3}^{\top}\vec{\Omega} \\ \frac{1}{2}\left(S\left(\mathbf{q}_{1:3}\right) + q_0\mathbf{I}\right)\vec{\Omega} \end{bmatrix} \tag{20}$$

with $S(\mathbf{q}_{1:3})$ denoting the skew-matrix of $\mathbf{q}_{1:3}$

$$S(\mathbf{q}_{1:3}) = \begin{bmatrix} 0 & -q_3 & q_2 \\ q_3 & 0 & -q_1 \\ -q_2 & q_1 & 0 \end{bmatrix}. \tag{21}$$

3 Attitude Control Design

As a first step, the commanded acceleration from the position loop and the commanded yaw angle have to be converted into a desired attitude $\mathbf{q_{cmd}}$ (Section 3.2). The objective is then to design a feedback law which stabilizes the quadrocopter at any desired physical attitude. Since any physical attitude in $SO(3)$ corresponds to two antipodal quaternions in $\mathbb{S}^3$, this can be achieved by stabilizing the attitude $\mathbf{q}$ at $\pm\mathbf{q}_{cmd}$. When neglecting this fact, quaternion-based controllers can cause undesirable phenomena such as unwinding, where the rigid body rotates unnecessarily through a full rotation [7, 2]. To solve the problem of unwinding, the controller must satisfy

$$\vec{\Omega}_{cmd}(\mathbf{q}) = \vec{\Omega}_{cmd}(-\mathbf{q}). \tag{22}$$

This can easily be seen, since $\mathbf{q}$ and $-\mathbf{q}$ represent the same physical attitude, the control output should be the same as well. This issue can be solved by changing the sign of $\mathbf{q}$ whenever $q_0 < 0$. Changing the sign however results in a discontinuous controller. Interestingly, this is not a drawback since it can be shown that all continuous state-feedback control laws are at most *almost* globally stabilizing.

3.1 Control Law

Consider the following control law:

$$\vec{\Omega}_{cmd}(\mathbf{q}) = \frac{2}{\tau}\mathrm{sgn}(q_{e,0})\mathbf{q}_{\mathbf{e},1:3}, \qquad \mathrm{sgn}(q_{e,0}) = \begin{cases} 1, & q_{e,0} \geq 0 \\ -1, & q_{e,0} < 0 \end{cases} \tag{23}$$

where

- τ = first-order system time constant [s],
- $\mathbf{q_e} := \mathbf{q}^{-1} \cdot \mathbf{q_{cmd}}$ error measure, representing the rotation from $\mathbf{q}$ to $\mathbf{q_{cmd}}$.

Then, $\pm\mathbf{q_{cmd}}$ is a globally asymptotically stable equilibrium point of (20).

Proof. Without loss of generality, set $\mathbf{q_{cmd}} = \mathbf{q_I}$. Define an autonomous hybrid automaton [6] $H = (Z, Q, f, Init, Dom, E, G, R)$ with

- $Z = \{z_1, z_2\}$, two discrete states, corresponding to the upper and lower hemisphere of $\mathbb{S}^3$.
- $Q = \mathbb{S}^3 \subset \mathbb{R}^4$, continuous state $\mathbf{q}$, which lies in the three-sphere and represents the current attitude.

- A vector field

$$f(z,\mathbf{q}) = \begin{cases} \frac{1}{2}\mathbf{q}\cdot\mathbf{p}\left(\vec{\Omega}_{cmd}\left(z,\mathbf{q}\right)\right), & \text{if } z = z_1 \\ -\frac{1}{2}\mathbf{q}\cdot\mathbf{p}\left(\vec{\Omega}_{cmd}\left(z,\mathbf{q}\right)\right), & \text{if } z = z_2 \end{cases} \tag{24}$$

  with

$$\vec{\Omega}_{cmd}(z,\mathbf{q}) = \begin{cases} \frac{2}{\tau}\mathbf{q_e}_{,1:3} = -\frac{2}{\tau}\mathbf{q}_{1:3}, & \text{if } z = z_1 \\ -\frac{2}{\tau}\mathbf{q_e}_{,1:3} = \frac{2}{\tau}\mathbf{q}_{1:3}, & \text{if } z = z_2 \end{cases} \tag{25}$$

  describing the evolution of $\mathbf{q}$ in time when the quaternion is in the upper hemisphere z_1 or in the lower hemisphere z_2 respectively.

- Domains $Dom(z_1) = Dom(z_2) = \mathbb{S}^3$.
- Edges $E = \{(z_1, z_2), (z_2, z_1)\}$, representing the possibility to go from z_1 to z_2 and vice versa.
- Guards $G(z_1, z_2) = \{\mathbf{q} \in \mathbb{S}^3 | q_0 < 0\}$, and $G(z_2, z_1) = \{\mathbf{q} \in \mathbb{S}^3 | q_0 < 0\}$
- Reset maps $R(z_1, z_2, \mathbf{q}) = R(z_2, z_1, \mathbf{q}) = -\mathbf{q}$, since a rotation of 180° about an axis is equal to a rotation of −180° about the same axis.

Now, consider the following Lyapunov candidate function:

$$V(z,\mathbf{q}) = \mathbf{q}_{1:3}^\top \mathbf{q}_{1:3} + (q_0 - 1)^2 \quad \forall z \in Z \tag{26}$$

Differentiating (26) with respect to time and inserting (24) and (20) yields

$$\begin{aligned} \dot{V}(z,\mathbf{q}) &= \frac{\partial V(z,\mathbf{q})}{\partial \mathbf{q}} f(z,\mathbf{q}) && (27)\\ &= \begin{bmatrix} 2\left(q_0 - 1\right) & 2\mathbf{q}_{\mathbf{1:3}}^\top \end{bmatrix} \begin{bmatrix} \frac{1}{2}\mathbf{q}_{\mathbf{1:3}}^\top \frac{2}{\tau}\mathbf{q}_{\mathbf{1:3}} \\ -\frac{1}{2}\left(S\left(\mathbf{q}_{\mathbf{1:3}}\right) + q_0\mathbf{I}\right)\frac{2}{\tau}\mathbf{q}_{\mathbf{1:3}} \end{bmatrix} && (28)\\ &= \frac{2}{\tau}\left(q_0 - 1\right)\mathbf{q}_{\mathbf{1:3}}^\top \mathbf{q}_{\mathbf{1:3}} - \frac{2}{\tau}\mathbf{q}_{\mathbf{1:3}}^\top \left(S(\mathbf{q}_{\mathbf{1:3}}) + q_0\mathbf{I}\right)\mathbf{q}_{\mathbf{1:3}} && (29)\\ &= -\frac{2}{\tau}\mathbf{q}_{1:3}^\top (S(\mathbf{q}_{1:3}) + \mathbf{I})\mathbf{q}_{1:3} && (30)\\ &= -\frac{2}{\tau}\mathbf{q}_{1:3}^\top \mathbf{q}_{1:3} \quad \forall z \in Z && (31) \end{aligned}$$

By Lyapunovs stability theorem for hybrid systems [6], the attitude $\mathbf{q_I}$ is a stable equilibrium because:

1. $V(z,\cdot) = 0$ if and only if $\mathbf{q} = \mathbf{q_I}$,
2. $V(z,\mathbf{q}) > 0 \quad \forall \mathbf{q}, (z,\mathbf{q}) \in Dom(z)\backslash\{\mathbf{q_I}\}$, and
3. $\dot{V}(z,\mathbf{q}) = \frac{\partial V(z,\mathbf{q})}{\partial \mathbf{q}} f(z,\mathbf{q}) \leq 0 \quad \forall \mathbf{q}, (z,\mathbf{q}) \in Dom(z)$.

Furthermore, $\dot{V}(z,\mathbf{q}) < 0 \quad \forall z \in Z, \quad \forall \mathbf{q} \in Dom(z)\backslash\{\mathbf{q_I}\}$ and $\dot{V}(z,\mathbf{q_I}) = 0 \; \forall z \in Z$. Note that $V(z,\mathbf{q})$ does not jump when z changes its state, therefore $V(z,\mathbf{q})$ is strictly decreasing, meaning that $\mathbf{q_I}$ is a globally asymptotically stable equilibrium point. □

Remark 1. *The error measure* $\mathbf{q_e}$ *represents the rotation from the current attitude* $\mathbf{q}$ *to the desired attitude* $\mathbf{q_{cmd}}$*. By taking the sign of* $\mathbf{q_e}$*, it is ensured that the rotation angle is always less or equal to* 180°*. This means that the controller always rotates the rigid body in such a way that the rotation angle is minimal. Note however, a minimal rotation angle does in general not imply a time-optimal rotation maneuver [1, 5]. This can easily be seen if the angular body rates are limited, i.e.* $|\Omega_i| \leq \Omega_{i,max}, \quad i \in \{x, y, z\}$ *and* $\Omega_{i,max} \neq \Omega_{j,max}$ *for* $i \neq j$*.*

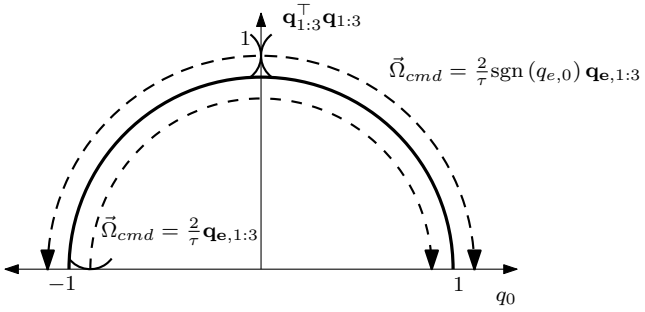

Figure 4: The proposed control law stabilizes the attitude to $q_0 = 1$ or $q_0 = -1$. However, there is a discontinuity at $q_0 = 0$. If the sign of $q_{e,0}$ is not taken into consideration, the controller would only be almost globally asymptotically stable with $q_0 = -1$ as an unstable equilibrium point [8].

Remark 2. *A visual interpretation of the control law* (23) *for* $\mathbf{q_{cmd}} = \mathbf{q_I}$ *is shown in Fig. 4.*

Remark 3. *By definition,* $\|\mathbf{q}\| = 1 \quad \forall \mathbf{q} \in \mathbb{S}^3$*, and therefore* $\|\mathbf{q}_{\mathbf{e},1:3}\| \leq 1$*. The maximum possible control output is thus limited to* $\|\vec{\Omega}_{cmd}\| \leq \frac{2}{\tau}$*. This is especially an advantage for systems with saturations.*

Remark 4. *Assume, without loss of generality, that* $\mathbf{q_{cmd}} = \mathbf{q_I}$*. Inserting the control law* (23) *into the state equation* (20) *yields*

$$\dot{\mathbf{q}} = \begin{bmatrix} \dot{q}_0 \\ \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \end{bmatrix} = \frac{1}{\tau} \begin{bmatrix} q_1^2 + q_2^2 + q_3^2 \\ -q_0 q_1 \\ -q_0 q_2 \\ -q_0 q_3 \end{bmatrix}. \tag{32}$$

For small deviations from the equilibrium $\mathbf{q_I}$ *(*$q_0 \approx 1, \mathbf{q}_{1:3} \ll 1$*),* (32) *simplifies to*

$$\dot{\mathbf{q}} = \begin{bmatrix} \dot{q}_0 \\ \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \end{bmatrix} = \frac{1}{\tau} \begin{bmatrix} 0 \\ -q_1 \\ -q_2 \\ -q_3 \end{bmatrix}, \tag{33}$$

which is an uncoupled first-order system with time-constant τ[3]*.*

3.1.1 Robustness

Although the proposed controller achieves global asymptotic stability of any desired attitude, it can be shown that the controller is not robust to arbitrarily small measurement noise. Following closely the proof in [8], a noise signal $\mathbf{q_{noise}}$ can be constructed such that when starting arbitrarily close to the discontinuity, then $\mathbf{q}$ stays close to the discontinuity for all time.
Let the current attitude $\mathbf{q}$ be

$$\mathbf{q} = \begin{bmatrix} \cos\left(\frac{\alpha}{2}\right) \\ \vec{k}\sin\left(\frac{\alpha}{2}\right) \end{bmatrix}. \tag{34}$$

Define $\mathbf{q_{noise}}$ to be

$$\mathbf{q_{noise}}(\mathbf{q}) := \begin{bmatrix} \cos\left(\frac{\beta(\mathbf{q})}{2}\right) \\ \vec{k}\sin\left(\frac{\beta(\mathbf{q})}{2}\right) \end{bmatrix}, \tag{35}$$

[3]Note that τ represents the time when the quaternion error decays to $\frac{1}{e}$ and *not* when the error in terms of Euler angles decays to $\frac{1}{e}$.

where $\vec{k}$ is eigenaxis of the attitude $\mathbf{q}$ and

$$\beta(\mathbf{q}) = \begin{cases} \beta_0, & \text{if } 0 < \alpha \leq \pi \\ -\beta_0, & \text{if } \pi < \alpha < 2\pi \\ -\beta_0, & \text{if } 0 \geq \alpha > -\pi \\ \beta_0, & \text{if } -\pi \geq \alpha > -2\pi \end{cases}, \quad \text{with } 0 < \beta_0 < \pi. \tag{36}$$

If the measured attitude $\widetilde{\mathbf{q}}$ is the current attitude rotated by the noise signal, i.e.

$$\widetilde{\mathbf{q}} = \mathbf{q} \cdot \mathbf{q_{noise}}(\mathbf{q}) = \begin{bmatrix} \cos\left(\frac{\alpha+\beta(\mathbf{q})}{2}\right) \\ \vec{k}\sin\left(\frac{\alpha+\beta(\mathbf{q})}{2}\right) \end{bmatrix}, \tag{37}$$

then, the kinematic equation for $\mathbf{q_{cmd}} = \mathbf{q_I}$ is

$$\dot{\mathbf{q}} = \begin{cases} \frac{1}{\tau}\mathbf{q} \cdot \mathbf{p}\left(\text{sgn}(q_0)\widetilde{\mathbf{q}}_{\mathbf{1:3}}\right), & \text{if } |\pi - \alpha| \leq \beta_0 \\ -\frac{1}{\tau}\mathbf{q} \cdot \mathbf{p}\left(\text{sgn}(q_0)\widetilde{\mathbf{q}}_{\mathbf{1:3}}\right), & \text{if } |\pi - \alpha| > \beta_0. \end{cases} \tag{38}$$

Define the set of all discontinuities $\mathcal{M} := \{\mathbf{q} \in \mathbb{S}^3 : q_0 = 0\}$, which corresponds to all attitudes that are 180° rotation from $\mathbf{q_{cmd}}$. Now consider the Lyapunov candidate function $V_{\mathcal{M}}(\mathbf{q}) = q_0^2$, with $V_{\mathcal{M}}(\mathbb{S}^3 \backslash \mathcal{M}) > 0$ and $V_{\mathcal{M}}(\mathcal{M}) = 0$. Then, for $|\pi - \alpha| \leq \beta_0$,

$$\begin{aligned} \dot{V}_{\mathcal{M}}(\mathbf{q}) &= 2q_0\dot{q}_0 && (39) \\ &= -\frac{1}{\tau}|q_0|\,\mathbf{q}_{\mathbf{1:3}}^{\top}\widetilde{\mathbf{q}}_{\mathbf{1:3}} && (40) \\ &= -\frac{1}{\tau}|q_0|\,\vec{k}^{\top}\vec{k}\sin\left(\frac{\alpha}{2}\right)\sin\left(\frac{\alpha+\beta(\mathbf{q})}{2}\right) && (41) \\ &\leq 0. && (42) \end{aligned}$$

Thus, starting from any attitude with $|\pi - \alpha| \leq \beta_0$, the system does not converge to the desired attitude.
In [8], a discontinuous controller with a hysteric memory state is proposed to avoid this unwanted equilibrium. The hysteric memory state defines the rotation direction and is always chosen such that the body rotates in direction of the shortest transition. If $|\pi - \alpha| \leq \beta_0$, the hysteric memory state does not get updated and the rotation direction remains constant, thus leaving the region where an arbitrarily small noise signal can destabilize the controller. In practice, a hysteric memory state is not necessary if the control law is implemented on a discrete-time controller. Since the output of a discrete-time controller is constant between two updates, the rotation direction does not get changed during that time period and hence, a discrete-time controller behaves like a continuous-time controller with hysteresis.

3.2 Desired Attitude

Given the quadrocopter setup in Fig. 2 with the body-fixed frame $\mathcal{B}$, it is clear that the quadrocopter can only accelerate in direction of $\vec{e}_z^{\mathcal{B}}$. Therefore, the commanded acceleration $\vec{a}_{cmd}$ has to be transformed into a target orientation $\mathbf{q_{cmd}}$ such that the corresponding z-axis is aligned with the desired acceleration. As with every pointing application, the rotation about the pointing direction is irrelevant, i.e. the rotation about the thrust direction $\vec{e}_z^{\mathcal{B}}$ has no influence on the translational behaviour of the quadrocopter. It makes therefore sense to split up the control task into two parts:

- **Reduced attitude control:** Only the crucial pointing direction of the thrust is controlled. The yaw angle is not controlled directly, but $\mathbf{q_{cmd}}$ is always chosen such that no rotation about the yaw axis is induced.
- **Full attitude control:** The pointing direction of the thrust vector as well as the yaw angle is controlled. $\mathbf{q_{cmd}}$ is chosen such that the corresponding z-axis is aligned with ${}_I\vec{e}^B_{cmd,z}$ and $\psi = \psi_{cmd}$.

3.2.1 Reduced Attitude Control

Given any desired acceleration $\vec{a}_{cmd}$, the desired thrust and direction can be computed by

$$ {}_I\vec{e}^B_{cmd,z} = \frac{{}_I\vec{a}_{cmd}}{\|{}_I\vec{a}_{cmd}\|}, \tag{43} $$

$$ coll_{cmd} = \|{}_I\vec{a}_{cmd}\|. \tag{44} $$

The reduced error quaternion $\mathbf{q_{e,red}}$ (Fig. 5), which rotates the quadrocopter from the current attitude to the desired attitude is then

$$ \mathbf{q_{e,red}} = \begin{bmatrix} \cos\left(\frac{\alpha}{2}\right) \\ \vec{k}\sin\left(\frac{\alpha}{2}\right) \end{bmatrix} = \begin{bmatrix} \cos\left(\frac{\alpha}{2}\right) \\ \sin\left(\frac{\alpha}{2}\right)\left(\frac{{}_I\vec{e}^B_z \times {}_I\vec{e}^B_{cmd,z}}{\|{}_I\vec{e}^B_z \times {}_I\vec{e}^B_{cmd,z}\|}\right) \end{bmatrix}, \tag{45} $$

where ${}_Ie^B_z$ is the corresponding z-axis of the current attitude $\mathbf{q}$ and α is the angle between the current thrust direction and the desired thrust direction:

$$ \alpha = \arccos\left({}_I\vec{e}^{B\top}_z \, {}_I\vec{e}^B_{cmd,z}\right). \tag{46} $$

Finally, the desired attitude can be produced by

$$ \mathbf{q_{cmd,red}} = \mathbf{q} \cdot \mathbf{q_{e,red}}. \tag{47} $$

Note that last entry of $\mathbf{q_{e,red}}$ is always zero because $\vec{k} \perp \vec{e}^B_z$, which implies that $\Omega_{cmd,z} = 0$. Due to the structure of the Euler angles, this does not mean that the yaw angle ψ is constant.

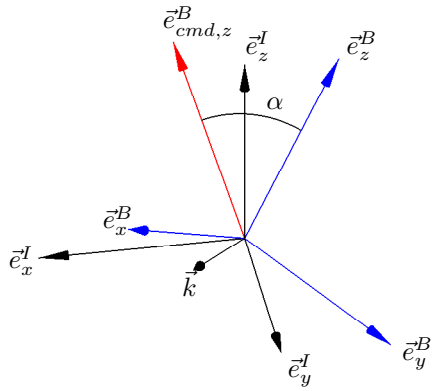

Figure 5: Geometric interpretation of $\mathbf{q_{e,red}}$.

3.2.2 Full Attitude Control

A desired yaw angle ψ_{cmd} and thrust direction ${}_{\mathcal{I}}\vec{e}^{\mathcal{B}}_{cmd,z}$ fully define the attitude. Pitch and roll angles, θ_{cmd} and ϕ_{cmd}, can thus be recovered from ψ_{cmd} and ${}_{\mathcal{I}}\vec{e}^{\mathcal{B}}_{cmd,z}$ using projections of ${}_{\mathcal{I}}\vec{e}^{\mathcal{B}}_{cmd,z}$ onto the intermediate coordinate frames $\mathcal{K}$ and $\mathcal{L}$.
First, the desired pointing direction ${}_{\mathcal{I}}\vec{e}^{\mathcal{B}}_{cmd,z}$ is expressed in the $\mathcal{K}$-frame, which is rotated by ψ_{cmd} about the initial z-axis:

$$ {}_{\mathcal{K}}\vec{e}^{\mathcal{B}}_{cmd,z} = R_{\mathcal{KI}}\, {}_{\mathcal{I}}\vec{e}^{\mathcal{B}}_{cmd,z} = \begin{bmatrix} \cos(\psi_{cmd}) & \sin(\psi_{cmd}) & 0 \\ -\sin(\psi_{cmd}) & \cos(\psi_{cmd}) & 0 \\ 0 & 0 & 1 \end{bmatrix} {}_{\mathcal{I}}\vec{e}^{\mathcal{B}}_{cmd,z}. \tag{48} $$

The pitch angle θ_{cmd} can then easily be reconstructed by projecting the pointing vector onto the $\vec{e}^{\mathcal{K}}_x, \vec{e}^{\mathcal{K}}_z$-plane (Fig. 6):

$$ \theta_{cmd} = \arctan\left(\frac{{}_{\mathcal{K}}\vec{e}^{\mathcal{B}}_{cmd,z,1}}{{}_{\mathcal{K}}\vec{e}^{\mathcal{B}}_{cmd,z,3}} \right). \tag{49} $$

Expressing ${}_{\mathcal{I}}\vec{e}^{\mathcal{B}}_{cmd,z}$ in the $\mathcal{L}$-frame, which is rotated about $\vec{e}^{\mathcal{K}}_y$ by the newly computed computed pitch angle θ_{cmd} yields

$$ {}_{\mathcal{L}}\vec{e}^{\mathcal{B}}_{cmd,z} = R_{\mathcal{LK}}\, {}_{\mathcal{K}}\vec{e}^{\mathcal{B}}_{cmd,z} = \begin{bmatrix} \cos(\theta_{cmd}) & 0 & -\sin(\theta_{cmd}) \\ 0 & 1 & 0 \\ \sin(\theta_{cmd}) & 0 & \cos(\theta_{cmd}) \end{bmatrix} {}_{\mathcal{K}}\vec{e}^{\mathcal{B}}_{cmd,z}. \tag{50} $$

Again, projecting ${}_{\mathcal{I}}\vec{e}^{\mathcal{B}}_{cmd,z}$ onto the $\vec{e}^{\mathcal{L}}_y, \vec{e}^{\mathcal{L}}_z$-plane, it can be seen from Fig. 7 that the roll angle ϕ_{cmd} is given by

$$ \phi_{cmd} = \arctan2\left(-{}_{\mathcal{L}}\vec{e}^{\mathcal{B}}_{cmd,z,2}, {}_{\mathcal{L}}\vec{e}^{\mathcal{B}}_{cmd,z,3} \right). \tag{51} $$

Finally, the desired attitude can be constructed using (19):

$$ \mathbf{q_{cmd,full}} = \mathbf{q}\left(\psi_{cmd}, \theta_{cmd}, \phi_{cmd}\right). \tag{52} $$

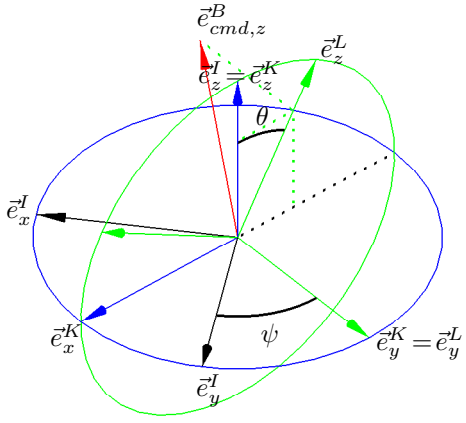

Figure 6: Projection onto the $\vec{e}^{\mathcal{K}}_x, \vec{e}^{\mathcal{K}}_z$-plane.

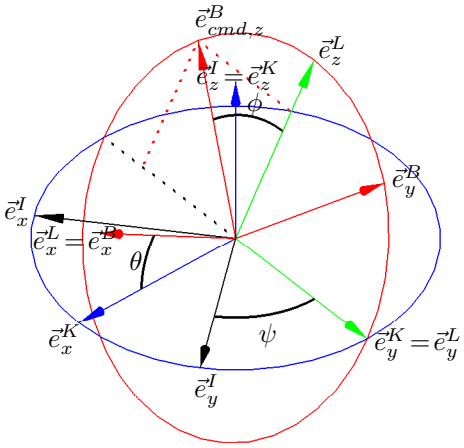

Figure 7: Projection onto the $\vec{e}^{\,L}_{y}, \vec{e}^{\,L}_{z}$-plane.

3.2.3 Mixing Full and Reduced Attitude Control

Although it is not necessary that yaw is controlled in order to follow any desired trajectory, it is nevertheless sometimes desirable. In practice, using full attitude control ($\mathbf{q_{cmd}} = \mathbf{q_{cmd,full}}$) does not yield good results. The dynamics of Ω_z are much slower than for Ω_x and Ω_y. The reason for this is that Ω_z can not be controlled by applying differential thrust but only by exploiting the air drag of the propellers. When controlling all angular rates with the same gain $\frac{1}{\tau}$, either the controller is very slow for the pitch and roll angles or there is a lot of overshoot in yaw. Furthermore, since with the control structure $\|\vec{\Omega}_{cmd}\|$ is limited, a lot of control effort might be wasted to control yaw although it is not crucial for the quadrocopter motion. This can become important for very aggressive maneuvers. One possibility to overcome these issues is to mix reduced and full attitude control. Define $\mathbf{q_{mix}} := \mathbf{q}^{-1}_{\mathbf{cmd,red}} \cdot \mathbf{q_{cmd,full}}$ to be the rotation between $\mathbf{q_{cmd,red}}$ and $\mathbf{q_{cmd,full}}$. By definition, $\mathbf{q_{mix}}$ always has the form

$$\mathbf{q_{mix}} = \begin{bmatrix} \cos\left(\frac{\alpha_{mix}}{2}\right) \\ 0 \\ 0 \\ \sin\left(\frac{\alpha_{mix}}{2}\right) \end{bmatrix}, \tag{53}$$

where α_{mix} is the rotation angle between $\mathbf{q_{cmd,red}}$ and $\mathbf{q_{cmd,full}}$ about ${}_B\vec{e}^{\,B}_{z}$. Choosing $\mathbf{q_{cmd}}$ to lie anywhere in between the two attitudes $\mathbf{q_{cmd,red}}$ and $\mathbf{q_{cmd,full}}$, i.e.

$$\mathbf{q_{cmd}} = \mathbf{q_{cmd,red}} \cdot \begin{bmatrix} \cos\left(\frac{p\alpha_{mix}}{2}\right) \\ 0 \\ 0 \\ \sin\left(\frac{p\alpha_{mix}}{2}\right) \end{bmatrix}, \qquad \text{with } p \in [0,1]\,, \tag{54}$$

then it is guaranteed that the resulting rotation yields correct pitch and roll angles. Although the controller does only correct p-fraction of the actual yaw error, it nevertheless converges to the desired yaw angle as $t \to \infty$. $\Omega_{cmd,z}$ is now limited to

$$|\Omega_{cmd,z}| \leq \frac{2\sin\left(\frac{p\pi}{2}\right)}{\tau}. \tag{55}$$

3.2.4 Limiting Maximum Tilt Angle

For some applications, e.g. carrying loads, it might be wanted that the quadrocopter can only tilt up to some maximum tilt angle α_{max}. This can be implemented right after the step in (43). First, the tilt angle α_{tilt} is defined as the angle between $\vec{e}_z^{\,I}$ and $\vec{e}_{cmd,z}^{\,B}$:

$$\alpha_{tilt} = \arccos\left({}_I\vec{e}_z^{\,I\top}\,{}_I\vec{e}_{cmd,z}^{\,B}\right) \tag{56}$$

If $|\alpha_{tilt}| > \alpha_{max}$, the angle can be limited using (17). First, the rotation axis is computed,

$$\vec{k} = \frac{{}_I\vec{e}_z^{\,I} \times {}_I\vec{e}_{cmd,z}^{\,B}}{\|{}_I\vec{e}_z^{\,I\top}\,{}_I\vec{e}_{cmd,z}^{\,B}\|}, \tag{57}$$

and a tilt rotation is defined:

$$\mathbf{q_{tilt}} = \begin{bmatrix} \cos\left(\frac{\alpha_{tilt}}{2}\right) \\ \vec{k}\sin\left(\frac{\alpha_{tilt}}{2}\right) \end{bmatrix}. \tag{58}$$

The new target thrust direction, which lies within a cone with an aperture of $2\alpha_{max}$, is then

$$\mathbf{p}\left({}_I\vec{e}_{new\ cmd,z}^{\,B}\right) = \mathbf{q_{tilt}} \cdot \mathbf{p}\left({}_I\vec{e}_{cmd,z}^{\,B}\right) \cdot \bar{\mathbf{q}}_{\mathbf{tilt}}. \tag{59}$$

It is worth mentioning that this does not prevent the quadrocopter from flipping over. Although the reachable set of attitudes is now constrained, the quadrocopter might get rotated by some external momentum to an extreme attitude. The shortest angular rotation to recover to the commanded attitude can then still involve performing a flip.

3.3 Heuristic

The proposed control law (23) always stabilizes the quadrocopter to the desired orientation with the shortest possible rotation. This is a very desirable result, however, it was assumed that $\vec{\Omega}$ can be controlled directly and infinitely fast, which is in practise not true. Imagine the case where the quadrocopter has a yaw error of almost 180°. It does not make a big difference in time if the quadrocopter rotates about $\vec{k}$ or $-\vec{k}$. However, if $|\Omega_z|$ is large, it is obviously faster to rotate in the same direction as Ω_z than to accelerate in the opposite direction. The question, whether it is faster to undertake the longer rotation or not, is not trivial. In fact, the rise time of $\vec{\Omega}$ depends on many parameters, e.g. thrust magnitude and the value of $\vec{\Omega}$ itself and no algebraic expression exists to answer this question. To keep the computational effort small, a simple heuristic is used.

Here, only the heuristic for errors in yaw is introduced, however, it works the same way for pitch and roll angles. Assume that the quadrocopter has an angular body rate of $\vec{\Omega}$. The rotation axis of $\mathbf{q_{mix}}$ is either $\vec{k} = (0,0,1)^\top$ or $\vec{k} = (0,0,-1)^\top$. If the quadrocopter already rotates in the same direction as commanded, i.e. $\vec{k}^\top\vec{\Omega} \geq 0$, then obviously nothing has to be changed. Otherwise, if $\vec{k}^\top\vec{\Omega} < 0$, it might be faster, depending on the error in yaw, to undertake the longer rotation. The decision criterion is

$$\vec{k}^\top\vec{\Omega} < 0 \quad \text{AND} \quad \left|\vec{k}^\top\vec{\Omega}\right| \geq \text{threshold}\left(\alpha_{mix}\right). \tag{60}$$

For errors in yaw of almost 180°, the threshold should be close to 0 and should strictly increase for smaller error angles. A good threshold was found experimentally to be

$$\text{threshold}\left(\alpha_{mix}\right) = \frac{\pi - \alpha_{mix}}{\pi}\Omega_{z,min}. \tag{61}$$

Whenever $\left|\vec{k}^\top \vec{\Omega}\right|$ is above the threshold, it is better to continue in this direction. To gain as much time as possible, the rotation until an error angle of 180° should be conducted with full speed, i.e.

$$\mathbf{q_{e,full}} = \mathbf{q_{e,red}} \cdot \begin{bmatrix} \cos\left(\frac{p\pi}{2}\right) \\ -\vec{k}\sin\left(\frac{p\pi}{2}\right) \end{bmatrix}. \tag{62}$$

Note that it has to be ensured that once the rotation direction is decided, it does not get changed anymore. Therefore, the commanded rotation rate should be well above the threshold which implies that

$$\Omega_{z,min} < \frac{2\sin\left(\frac{p\pi}{2}\right)}{\tau}. \tag{63}$$

Because unit quaternions are normalized and large errors in pitch and roll can decrease the $\Omega_{cmd,z}$, this is not ensured when $\Omega_{z,min} = 2\sin(p\pi/2)/\tau$.

Remark 1. *The above heuristic also works for $\Omega_{x,y}$. However, $\Omega_{x,y}$ can be controlled much faster and therefore the time earning is not that large anymore. Experiments have shown that the heuristic is only beneficial for extremely large angular rates in pitch and roll, which are almost out of the measurement range of the gyroscopic sensors. Furthermore, the heuristic is designed for a single rotation about a principle axis of the quadrocopter. It is not totally clear what happens in the 3D case. It is therefore recommended to only apply the heuristic to the yaw axis, which can be controlled only slowly and therefore the time earnings are the highest.*

Remark 2. *Instead of the heuristic* (60)*, it was also attempted to solve the problem by simulating the state equation* (20) *forward in time, once with a rotation in the shorter direction and once in opposite direction. However, the results were not any better than with the above heuristic. Moreover, the simulation was computationally very expensive which made it useless for real-time uses.*

Remark 3. *If using the above heuristic, then stability and robustness of the controller would have to be proven again. Although the results are quite intuitive, the proof is not done here.*

4 Results

4.1 Time constant

The proposed controller (23) is based upon the assumption that $\vec{\Omega}$ can be controlled directly. This assumption certainly holds for large τ. In order to determine how small τ can be chosen such that the above assumption still holds, a series of step responses in attitude are analyzed.

4.1.1 Step in Pitch Angle

Starting from hovering, a step of 45° in pitch angle ϕ is commanded. As the problem is symmetric in pitch and roll, only the step response for pitch is analyzed. Fig. 8 shows the result for various τ and p such that $\tau_{yaw} = 0.02\text{s}$ (64).
For $\tau \geq 0.1\text{s}$, q_2 behaves like a first order system. For $\tau < 0.1\text{s}$, the assumption that $\vec{\Omega}$ can be controlled directly does not hold anymore. At time $t = 0.2\text{s}$, the commanded angular pitch rate $\Omega_{cmd,y}$ is equal to zero, but $\Omega_y(t = 0.2\text{s}) \gg 0^{\text{rad}}/\text{s}$ and hence, the system exhibits even overshoot. Therefore, fast responses with almost no overshoot can be achieved with $\tau \in [0.08\text{s}, 0.15\text{s}]$.

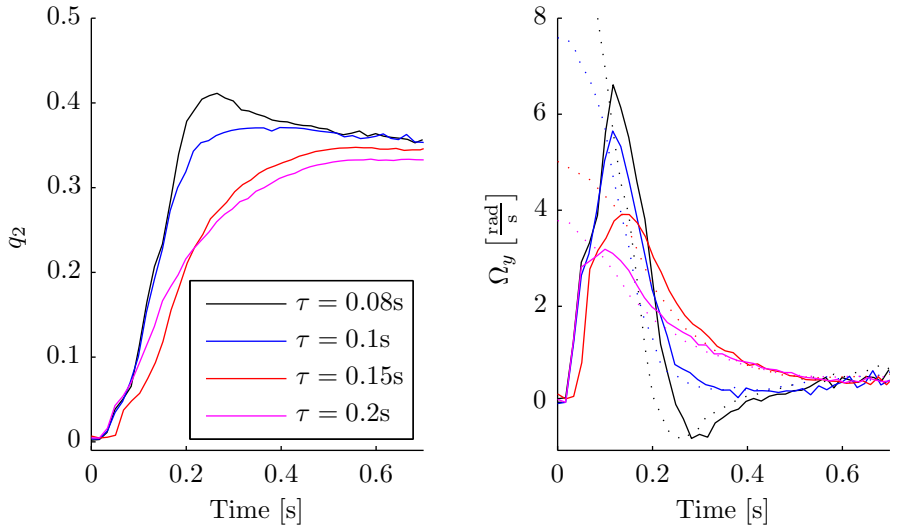

Figure 8: Step response of q_2 and Ω_y. The dotted lines are the corresponding $\Omega_{cmd,y}$.

4.1.2 Step in Yaw Angle

A step input of magnitude 180° in yaw is given into the attitude controller at $t = 0$s, starting from hovering with $\psi = 0°$. The results for different τ ($p = 1$) are shown in Fig. 9.
For $\tau \geq 0.2$s, q_3 certainly behaves very similar to a first-order system. For smaller τ, the system begins to show overshoot since the fundamental assumption is hurt. This can also be seen in Fig. 9, where Ω_z reached saturation for $\tau = 0.1$s even when decelerating. For controlling yaw, τ should therefore lie in the range of $[0.2\text{s}, 0.4\text{s}]$.

Remark 1. *In Fig. 9, the response of q_3 for $\tau = 0.5$s does oscillate slightly. This is due to the small control gain of this time constant which cannot prevent the quadrocopter from oscillating about its pitch and roll axis.*

Remark 2. *The above results strongly depend on the total thrust. For both experiments, τ was analyzed with a collective thrust that allows to maintain altitude.*

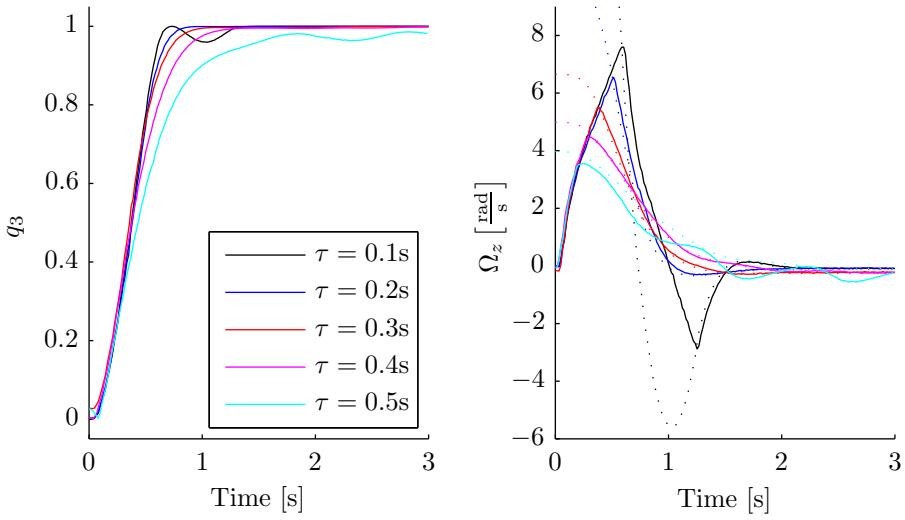

Figure 9: Step response of q_3 and Ω_z. The dotted lines are the corresponding $\Omega_{cmd,z}$.

4.2 Mixing Full and Reduced Attitude Control

The experiments in Section 4.1 have shown that a good controller gain for the yaw axis is significantly smaller than for roll and pitch axis. One method to implement this is to mix full and reduced attitude control and to choose the prioritization factor p of (54) to be

$$p = \frac{\tau}{\tau_{yaw}}, \tag{64}$$

where τ_{yaw} is a time constant which attains good responses in yaw and τ is the desired time constant of the overall controller (for pitch and roll). Equation (64) originated from a linearization of the control law with mixed attitude control for small errors in yaw. If p is chosen according to (64), then for small errors in yaw, the controller behaves just like one with a time constant of τ_{yaw} instead of τ. For a very aggressive controller with $\tau = 0.08$s and $\tau_{yaw} = 0.2$s, p is

$$p = \frac{0.08\text{s}}{0.2\text{s}} = 0.4. \tag{65}$$

In Fig 10, an experiment with different values for p and $\tau = 0.08$s is shown. Starting from a position with Euler angles $(\psi, \theta, \phi) = (180°, 45°, 0°)$, a step of 180° in yaw is commanded at $t = 0$s.
As a benchmark, the results with $p = 0$ are plotted in Fig. 10a. The rotation about the pitch angle does not change significantly for different values of p and the correct tilt angle is achieved in all cases shortly after 0.2s. But examining Fig. 10 more carefully, it can be seen that with increasing p, the pitch angle is corrected slightly slower. This is due to the increasing amount of control power that is taken up by yaw with larger p. Remarkable is that the larger p is, the faster yaw is controlled. For $p = 0.4$, the correct attitude is achieved at $t = 0.88$s whereas for $p = 0.267$ and $p = 0.133$, the desired attitude is not yet achieved at $t = 1$s.

Remark 1. *For extremely large angular rates about $\vec{e}_z^{\,\mathcal{B}}$, mixed attitude control achieves again better results than pure reduced attitude control. If $|\Omega_z|$ is large, than the eigenaxis $\vec{k}$ of $\mathbf{q_{e,red}}$ rotates with a speed of Ω_z in the $\vec{e}_x^{\,\mathcal{B}}, \vec{e}_y^{\,\mathcal{B}}$-plane. This implies that the rotational velocity of the motors increase and decrease constantly. However, there are limits on the motor dynamics and the motor speed can not be controlled infinitely fast. Further, since $\Omega_{cmd,z} = 0$, a lot of control effort is used by the onboard controller to break down Ω_z, leaving almost no control power to rotate the quadrocopter about its pitch and roll axis.*

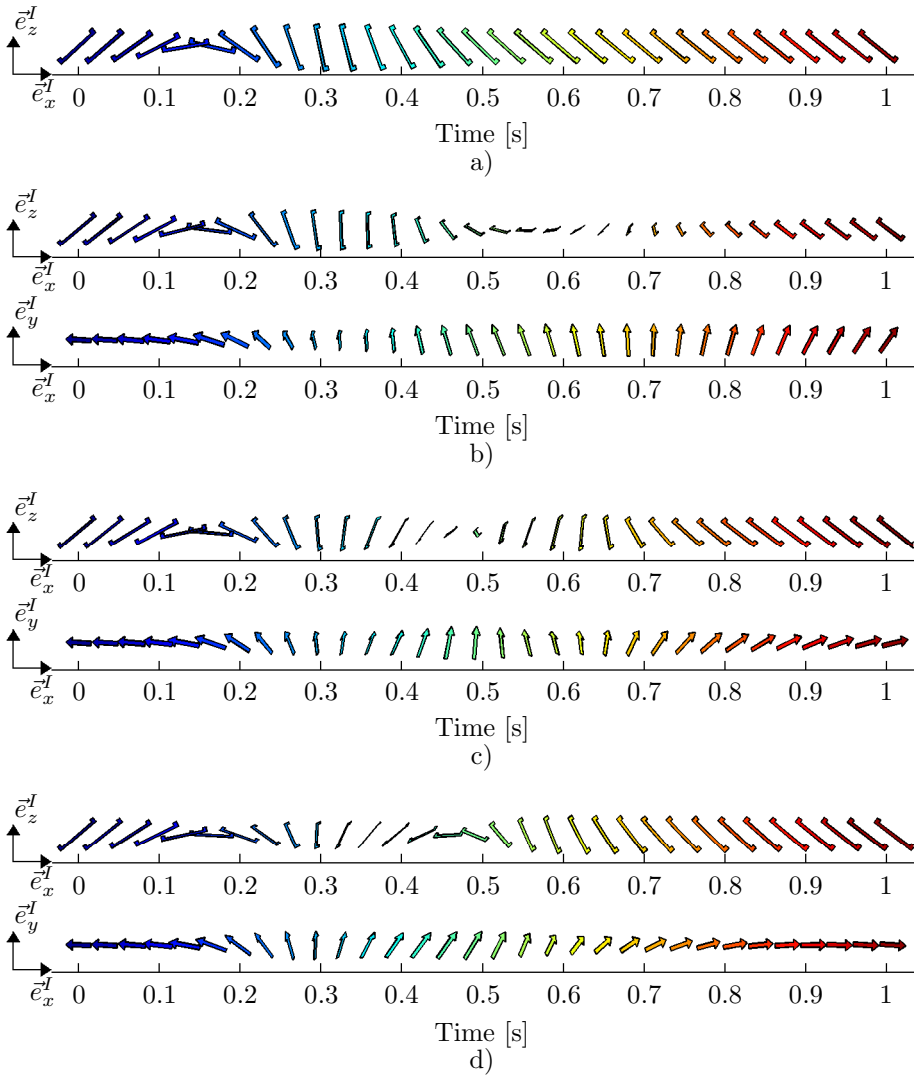

Figure 10: View of the quadrocopter in the xz-plane and xy-plane respectively for different prioritization factors: a) $p = 0.0$, b) $p = 0.133$, c) $p = 0.267$ and d) $p = 0.4$.

4.3 Heuristic

An experiment with the heuristic (60) enabled is carried out to show its effectiveness. Starting from hovering, the quadrocopter rotates about its yaw axis with different angular rates. Once the yaw angle ψ reaches 90°, the quadrocopter is commanded to return to $\psi_{cmd} = 0°$. The control parameters for this experiment are $\tau = 0.1$s, $p = 0.3$ and $\Omega_{z,min} = 4\frac{\text{rad}}{\text{s}}$. The results are shown in Fig. 11.

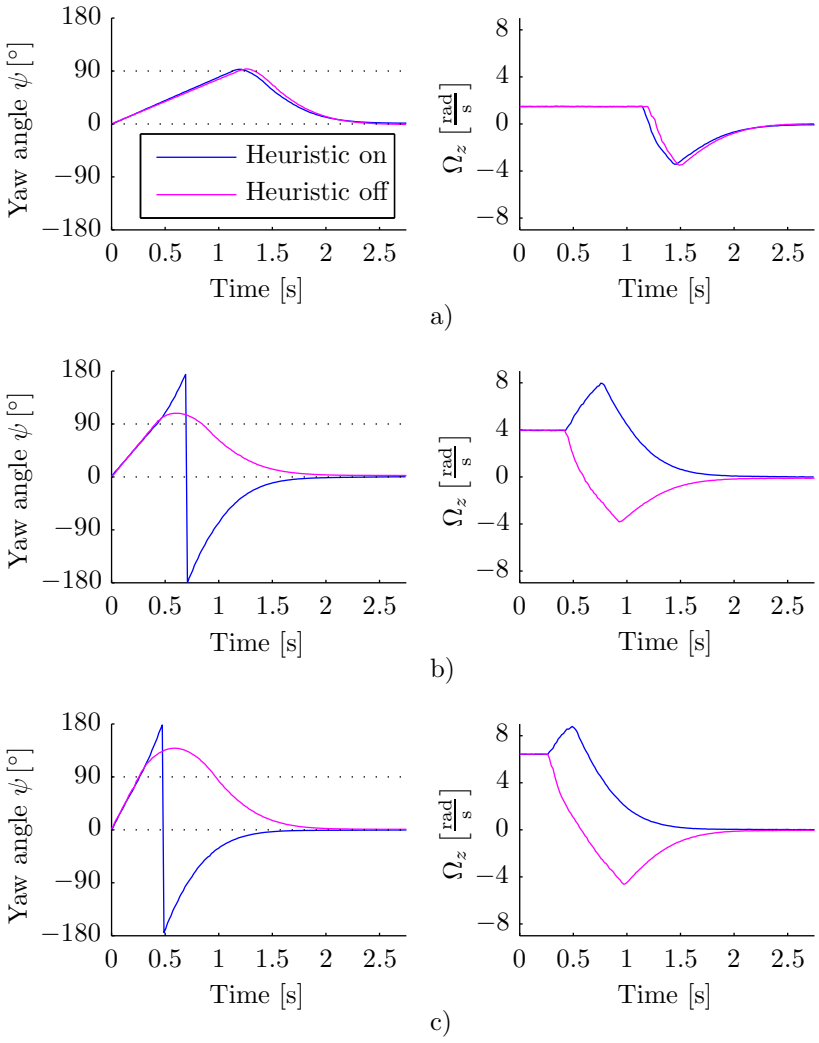

Figure 11: Experiment testing the heuristic for different angular rates: a) $\Omega_z = 1.5\frac{\text{rad}}{\text{s}}$, b) $\Omega_z = 4\frac{\text{rad}}{\text{s}}$ and c) $\Omega_z = 6.5\frac{\text{rad}}{\text{s}}$.

Clearly, for slow angular rates, there is no difference if the heuristic is enabled or not (Fig. 11a). But if Ω_z is beyond a certain threshold, it is faster to undertake the longer rotation and hence, the controller with the heuristic enabled attains better results.

5 Conclusion

A nonlinear attitude controller based on unit quaternions for the quadrocopters in the Flying Machine Arena has been developed. The control inputs are a desired acceleration and yaw angle, which are then converted to a commanded target orientation. It was assumed that the

onboard controller and quadropcopter dynamics are much faster than the attitude control loop and hence, the quadrocopter can be controlled directly by $\vec{\Omega}$. The stability and robustness of the controller were then proven.
By experiments, the performance of the nonlinear controller was demonstrated. The proposed attitude controller has many benefits, especially for aggressive maneuvers. Due to the global asymptotic stability, the proposed controller is able to recover the quadrocopter to hovering from arbitrary attitudes and can track any feasible trajectory. Furthermore, it was shown that the control of the crucial pitch and roll angles can be prioritized by mixing full and reduced attitude control. The proposed heuristic is able improve the time to reach a desired attitude, however, it only yields good results for rotations about the yaw axis.
Nevertheless, there are still many problems to be solved. Currently, the rise time of $\vec{\Omega}$ depends on the commanded thrust. This can cause very slow rotation rates if the commanded thrust is low although the whole rotation would take only a few tenths of a second and thus, a large thrust would have only minor influence on the translational motion. Furthermore, it would be nice if the prioritization factor p is adapted dynamically depending on the attitude error. Large errors in pitch and roll should be controlled first, but if no errors in pitch and roll occur, then p can be large such that errors in yaw are corrected in a reasonable time. Experiments also revealed that a yaw Euler angle ψ is not well suited to define a desired rotation about the quadrocopter's z-axis. Due to the pitch angle θ, which is only defined in a range of $[\frac{-\pi}{2}, \frac{\pi}{2}]$, jumps in ψ occur if the pitch angle would exceed its range. If these jumps are not considered in the input trajectory, the quadrocopter may rotate unintentionally about its yaw axis.
Currently, the proposed controller is model independent and can therefore be used for various flying vehicles. However, it might be beneficial in terms of time to combine the attitude and onboard controller and take model parameters such as moments of inertia into consideration.

References

[1] K. Bilimoria and B. Wie. Time-optimal three-axis reorientation of a rigid spacecraft. *Journal of Guidance Control and Dynamics*, 16(3):446–452, 1993.

[2] N.A. Chaturvedi, A.K. Sanyal, and N.H. McClamroch. Rigid-body attitude control. *Control Systems, IEEE*, 31(3):30 –51, june 2011.

[3] J. Diebel. Representing attitude: Euler angles, unit quaternions, and rotation vectors, 2006.

[4] J. Dvorak, M. de Lellis, and Z. Hurak. Advanced control of quadrotor using eigenaxis rotation. In *Control Applications (CCA), 2011 IEEE International Conference on*, pages 153 –158, sept. 2011.

[5] T. Lee, M. Leok, and N.H. McClamroch. Time optimal attitude control for a rigid body. In *American Control Conference, 2008*, pages 5210 –5215, june 2008.

[6] J. Lygeros, S. Sastry, and C. Tomlin. Hybrid systems: Foundations, advanced topics and applications.

[7] C.G. Mayhew, R.G. Sanfelice, and A.R. Teel. On quaternion-based attitude control and the unwinding phenomenon. In *American Control Conference (ACC), 2011*, pages 299 –304, 29 2011-july 1 2011.

[8] C.G. Mayhew, R.G. Sanfelice, and A.R. Teel. Quaternion-based hybrid control for robust global attitude tracking. *Automatic Control, IEEE Transactions on*, 56(11):2555 –2566, nov. 2011.

[9] D. Mellinger and V. Kumar. Minimum snap trajectory generation and control for quadrotors. In *ICRA*, pages 2520–2525. IEEE, 2011.

[10] K Rudin, M.-D. Hua, G. Ducard, and S. Bouabdallah. A robust attitude controller and its application to quadrotor helicopters. In *18th IFAC World Congress*, 2011.

[11] J. Stuelpnagel. On the parametrization of the three-dimensional rotation group. *SIAM Review*, 6(4):pp. 422–430, 1964.

[12] A. Tayebi. Unit quaternion-based output feedback for the attitude tracking problem. *Automatic Control, IEEE Transactions on*, 53(6):1516 –1520, july 2008.

[13] A. Tayebi and S. McGilvray. Attitude stabilization of a vtol quadrotor aircraft. *Control Systems Technology, IEEE Transactions on*, 14(3):562 – 571, may 2006.

[14] J.T.-Y. Wen and K. Kreutz-Delgado. The attitude control problem. *Automatic Control, IEEE Transactions on*, 36(10):1148 –1162, oct 1991.