Theory and Math Behind RPG Quadrotor Control

Matthias Faessler and Flavio Fontana

February 23, 2018

Contents

1 Introduction

In this document, we summarized the theory and math behind the quadrotor control algorithms developed at the *Robotics and Perception Group* as they are implemented in our rpg_quadrotor_control repository. If you spot any typos or mistakes or want to improve or add something, please open an issue or send us a pull request on `https://github.com/uzh-rpg/rpg_quadrotor_control`.

2 Mechanics

2.1 Euler's first law

The linear momentum of a rigid body, $\mathbf{p}$ is defined as

$$\mathbf{p} := m \cdot \mathbf{v}_S, \tag{1}$$

where m is the mass of the body, and $\mathbf{v}_S$ is the velocity of the center of mass. Euler's first law states that the change of linear momentum is equal to the sum of all external forces $\mathbf{F}$:

$$\dot{\mathbf{p}} = \mathbf{F}, \tag{2}$$

Differentiating (1) and inserting in (2) gives

$$\dot{\mathbf{p}} = m \cdot \mathbf{a}_S = \mathbf{F}, \tag{3}$$

where $\mathbf{a}_S$ is the acceleration of the center of mass.

2.2 Euler's second law

The angular momentum of a rigid body with respect to a stationary point O is defined as

$$\mathbf{L}_\mathrm{O} := \mathbf{R}_\mathrm{OP} \times \mathbf{p} + \mathbf{I}_\mathrm{P} \cdot \boldsymbol{\Omega} + m \cdot \mathbf{R}_\mathrm{PS} \times \mathbf{v}_\mathrm{P}, \tag{4}$$

where P is an arbitrary point on the body, and $\mathbf{I}_\mathrm{P}$ is the second moment of inertia matrix with respect to a rotation around P, $\boldsymbol{\Omega}$ is the angular velocity of the body. Evaluating (4) not for an arbitrary P but at the center of mass $P = S$ we find

$$\mathbf{L}_\mathrm{O} = \mathbf{R}_\mathrm{OS} \times \mathbf{p} + \mathbf{I}_\mathrm{S} \cdot \boldsymbol{\Omega}. \tag{5}$$

Euler's second law states that the change of angular momentum is equal to the sum of all external moments $\mathbf{M}_\mathrm{O}$ with respect to O:

$$\dot{\mathbf{L}}_\mathrm{O} = \mathbf{M}_\mathrm{O}. \tag{6}$$

Differentiating (5) and inserting in (6) gives

$$\dot{\mathbf{L}}_\mathrm{O} = \mathbf{R}_\mathrm{OS} \times \dot{\mathbf{p}} + \mathbf{I}_\mathrm{S} \cdot \boldsymbol{\Psi} + \boldsymbol{\Omega} \times \mathbf{I}_\mathrm{S} \cdot \boldsymbol{\Omega} = \mathbf{M}_\mathrm{O}, \tag{7}$$

where $\boldsymbol{\Psi}$ is the angular acceleration of the body. We can further rewrite (7) as

$$\begin{aligned}
\dot{\mathbf{L}}_\mathrm{O} - \mathbf{R}_\mathrm{OS} \times \dot{\mathbf{p}} = \mathbf{I}_\mathrm{S} \cdot \boldsymbol{\Psi} + \boldsymbol{\Omega} \times \mathbf{I}_\mathrm{S} \cdot \boldsymbol{\Omega} &= \mathbf{M}_\mathrm{O} - \mathbf{R}_\mathrm{OS} \times \dot{\mathbf{p}}, \\
&= \mathbf{M}_\mathrm{O} + \mathbf{R}_\mathrm{SO} \times \dot{\mathbf{p}}, \\
&= \mathbf{M}_\mathrm{O} + \mathbf{R}_\mathrm{SO} \times \mathbf{F}, \\
\mathbf{I}_\mathrm{S} \cdot \boldsymbol{\Psi} + \boldsymbol{\Omega} \times \mathbf{I}_\mathrm{S} \cdot \boldsymbol{\Omega} &= \mathbf{M}_\mathrm{S}.
\end{aligned} \tag{8}$$

2.3 Summary

Note that we can not only evaluate (3) and (8) in the world frame W but in any rotating coordinate systems A and C.

$$m \cdot {}_\mathrm{A}\mathbf{a}_\mathrm{S} = {}_\mathrm{A}\mathbf{F}, \tag{9}$$

$${}_\mathrm{C}\mathbf{I}_\mathrm{S} \cdot {}_C\boldsymbol{\Psi} + {}_\mathrm{C}\boldsymbol{\Omega} \times {}_\mathrm{C}\mathbf{I}_\mathrm{S} \cdot {}_\mathrm{C}\boldsymbol{\Omega} = {}_\mathrm{C}\mathbf{M}_\mathrm{S}. \tag{10}$$

3   Attitude Representations

This is not an introduction to coordinate transformations but a small summary of the important math we need to describe the dynamics of a quadrotor.

3.1   Rotation Matrices and Euler Angles

We denote the rotation matrix that converts from system B to W as $\mathbf{R}_{\mathrm{WB}}$ and the translation of system B with respect to system W as $\mathbf{t}_{WB}$, respectively. To convert a vector $\mathbf{c}$ from the body frame B to the world frame W we use

$$_{\mathrm{W}}\mathbf{c} = \mathbf{R}_{\mathrm{WB}} \cdot {}_{\mathrm{B}}\mathbf{c}. \tag{11}$$

Correspondingly, to convert a point $\mathbf{p}$ from the body frame B to the world frame W we use

$$_{\mathrm{W}}\mathbf{p} = \mathbf{R}_{\mathrm{WB}} \cdot {}_{\mathrm{B}}\mathbf{p} \; + \; {}_{\mathrm{W}}\mathbf{t}_{\mathrm{WB}}. \tag{12}$$

Later on Euler angles can be used. It is really important to keep in mind that with Euler angles the order of each single rotation is important. When you are using formulas with euler angles, always check what convention they are using. Here we will use the $z - y - x$ convention:

1. yaw ψ around the z body axis

2. pitch θ around the new y body axis

3. roll ϕ around the new x body axis

$$\mathbf{R}_{\mathrm{WB}} = \mathbf{R}_{\mathrm{z}}(\psi)\mathbf{R}_{\mathrm{y}}(\theta)\mathbf{R}_{\mathrm{x}}(\phi), \tag{13}$$

where

$$\mathbf{R}_{\mathrm{z}}(\psi) = \begin{bmatrix} \cos(\psi) & -\sin(\psi) & 0 \\ \sin(\psi) & \cos(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix}, \tag{14}$$

$$\mathbf{R}_{\mathrm{y}}(\theta) = \begin{bmatrix} \cos(\theta) & 0 & \sin(\theta) \\ 0 & 1 & 0 \\ -\sin(\theta) & 0 & \cos(\theta) \end{bmatrix}, \tag{15}$$

$$\mathbf{R}_{\mathrm{x}}(\phi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\phi) & -\sin(\phi) \\ 0 & \sin(\phi) & \cos(\phi) \end{bmatrix}. \tag{16}$$

Putting it all together we get

$$\mathbf{R}_{\mathrm{WB}} = \begin{bmatrix} \mathrm{c}(\psi)\,\mathrm{c}(\theta) & \mathrm{c}(\psi)\,\mathrm{s}(\theta)\,\mathrm{s}(\phi) - \mathrm{s}(\psi)\,\mathrm{c}(\phi) & \mathrm{c}(\psi)\,\mathrm{s}(\theta)\,\mathrm{c}(\phi) + \mathrm{s}(\psi)\,\mathrm{s}(\phi) \\ \mathrm{s}(\psi)\,\mathrm{c}(\theta) & \mathrm{s}(\psi)\,\mathrm{s}(\theta)\,\mathrm{s}(\phi) + \mathrm{c}(\psi)\,\mathrm{c}(\phi) & \mathrm{s}(\psi)\,\mathrm{s}(\theta)\,\mathrm{c}(\phi) - \mathrm{c}(\psi)\,\mathrm{s}(\phi) \\ -\mathrm{s}(\theta) & \mathrm{c}(\theta)\,\mathrm{s}(\phi) & \mathrm{c}(\theta)\,\mathrm{c}(\phi) \end{bmatrix}, \tag{17}$$

$$= \begin{bmatrix} \mathbf{x}_{\mathrm{B}} & \mathbf{y}_{\mathrm{B}} & \mathbf{z}_{\mathrm{B}} \end{bmatrix}, \tag{18}$$

with $\mathbf{x}_{\mathrm{B}}$, $\mathbf{y}_{\mathrm{B}}$ and $\mathbf{z}_{\mathrm{B}}$ being the orthogonal basis vectors of the Body coordinate system B represented in world coordinates W as illustrated in Fig. 1. The angular velocity of coordinate system B with respect to the world frame W is denoted as $\boldsymbol{\omega}_{\mathrm{WB}}$. For the next step, we define the skew symmetric matrix

$$\hat{\boldsymbol{\omega}}_{\mathrm{WB}} = \begin{bmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{bmatrix}. \tag{19}$$

The angular velocity is defined as

$$_{\mathrm{W}}\hat{\boldsymbol{\omega}}_{\mathrm{WB}} := \dot{\mathbf{R}}_{\mathrm{WB}} \cdot \mathbf{R}_{\mathrm{WB}}^{T}, \tag{20}$$

or in body frame B

$$\mathrm{_B}\hat{\boldsymbol{\omega}}_{\mathrm{WB}} = \mathbf{R}_{\mathrm{BW}} \cdot {}_{\mathrm{W}}\hat{\boldsymbol{\omega}}_{\mathrm{WB}} \cdot \mathbf{R}_{\mathrm{BW}}^T, \tag{21}$$

$$= \mathbf{R}_{\mathrm{BW}} \cdot \dot{\mathbf{R}}_{\mathrm{WB}} \cdot \mathbf{R}_{\mathrm{WB}}^T \cdot \mathbf{R}_{\mathrm{BW}}^T, \tag{22}$$

$$= \mathbf{R}_{\mathrm{WB}}^\top \cdot \dot{\mathbf{R}}_{\mathrm{WB}}. \tag{23}$$

Evaluating (20) and (23) for $\boldsymbol{\omega}_{\mathrm{WB}}$ we get

$$
{}_{\mathrm{W}}\boldsymbol{\omega}_{\mathrm{WB}} = \begin{bmatrix} \dot{\phi}\,\mathrm{c}(\theta)\,\mathrm{c}(\psi) - \dot{\theta}\,\mathrm{s}(\psi) \\ \dot{\phi}\,\mathrm{c}(\theta)\,\mathrm{s}(\psi) + \dot{\theta}\,\mathrm{c}(\psi) \\ -\dot{\phi}\,\mathrm{s}(\theta) + \dot{\psi} \end{bmatrix} = \begin{bmatrix} \mathrm{c}(\theta)\,\mathrm{c}(\psi) & -\mathrm{s}(\psi) & 0 \\ \mathrm{c}(\theta)\,\mathrm{s}(\psi) & \mathrm{c}(\psi) & 0 \\ -\mathrm{s}(\theta) & 0 & 1 \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix}, \tag{24}
$$

and

$$
{}_{\mathrm{B}}\boldsymbol{\omega}_{\mathrm{WB}} = \begin{bmatrix} \dot{\phi} - \mathrm{s}(\theta)\dot{\psi} \\ \mathrm{c}(\phi)\dot{\theta} + \mathrm{s}(\phi)\,\mathrm{c}(\theta)\dot{\psi} \\ -\mathrm{s}(\phi)\dot{\theta} + \mathrm{c}(\phi)\,\mathrm{c}(\theta)\dot{\psi} \end{bmatrix} = \begin{bmatrix} 1 & 0 & -\mathrm{s}(\theta) \\ 0 & \mathrm{c}(\phi) & \mathrm{s}(\phi)\,\mathrm{c}(\theta) \\ 0 & -\mathrm{s}(\phi) & \mathrm{c}(\phi)\,\mathrm{c}(\theta) \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix}. \tag{25}
$$

3.2 Quaternions

Instead of using Euler angles or rotation matrices, the orientation of a quadrotor can also be represented as a quaternion. We denote

$$\mathbf{q}_{\mathrm{WB}} = \begin{bmatrix} q_w & q_x & q_y & q_z \end{bmatrix}^\top \tag{26}$$

as the quaternion that describes the orientation of coordinate frame B with respect to coordinate frame W, where q_w is the real part of the quaternion. Note that $\mathbf{q}_{\mathrm{WB}}$ is what you get from optitrack. The adjoint, norm, and inverse of the quaternion, $\mathbf{q}$, are

$$\bar{\mathbf{q}} = \begin{bmatrix} q_w & -q_x & -q_y & -q_z \end{bmatrix}^\top, \tag{27}$$

$$\|\mathbf{q}\| = \sqrt{q_w^2 + q_x^2 + q_y^2 + q_z^2}, \tag{28}$$

$$\mathbf{q}^{-1} = \frac{\bar{\mathbf{q}}}{\|\mathbf{q}\|}. \tag{29}$$

3.2.1 Quaternion Multiplication

Quaternion multiplication is not commutative. The Quaternion multiplication between quaternions $\mathbf{q}$ and $\mathbf{p}$ is defined as

$$\mathbf{q} \otimes \mathbf{p} = \mathbf{Q}\,(\mathbf{q}) \cdot \mathbf{p} = \bar{\mathbf{Q}}\,(\mathbf{p}) \cdot \mathbf{q}, \tag{30}$$

$$\mathbf{p} \otimes \mathbf{q} = \mathbf{Q}\,(\mathbf{p}) \cdot \mathbf{q} = \bar{\mathbf{Q}}\,(\mathbf{q}) \cdot \mathbf{p}, \tag{31}$$

$$\tag{32}$$

where

$$
\mathbf{Q}\,(\mathbf{q}) = \begin{bmatrix} q_w & -q_x & -q_y & -q_z \\ q_x & q_w & -q_z & q_y \\ q_y & q_z & q_w & -q_x \\ q_z & -q_y & q_x & q_w \end{bmatrix}, \tag{33}
$$

$$
\bar{\mathbf{Q}}\,(\mathbf{q}) = \begin{bmatrix} q_w & -q_x & -q_y & -q_z \\ q_x & q_w & q_z & -q_y \\ q_y & -q_z & q_w & q_x \\ q_z & q_y & -q_x & q_w \end{bmatrix}. \tag{34}
$$

We find that

$$\mathbf{Q}\,(\bar{\mathbf{q}}) = \mathbf{Q}\,(\mathbf{q})^\top, \tag{35}$$

$$\bar{\mathbf{Q}}\,(\bar{\mathbf{q}}) = \bar{\mathbf{Q}}\,(\mathbf{q})^\top. \tag{36}$$

3.2.2 Rotating a Vector by a Quaternion

To rotate a vector $\mathbf{v}$ by a quaternion $\mathbf{q}$ we use the notion $\mathbf{q} \odot \mathbf{v}$. To rotate a vector ${}_B\mathbf{v}$ represented in body coordinates B into world coordinates W we can apply

$$\begin{bmatrix} 0 \\ {}_\text{W}\mathbf{v} \end{bmatrix} = \begin{bmatrix} 0 \\ \mathbf{q}_\text{WB} \odot {}_\text{B}\mathbf{v} \end{bmatrix} = \bar{\mathbf{Q}}^\top \left(\mathbf{q}_\text{WB} \right) \cdot \mathbf{Q} \left(\mathbf{q}_\text{WB} \right) \cdot \begin{bmatrix} 0 \\ {}_\text{B}\mathbf{v} \end{bmatrix}. \tag{37}$$

This equation will be more clear by looking at the derivations for transforming a quaternion into a rotation matrix as described in Section 3.3.5.

3.2.3 Quaternion Rates to Angular Velocity

The time derivative of the unit quaternion is the vector of quaternion rates. The quaternion rates $\dot{\mathbf{q}}$ are related to the angular velocities similar to (20).

$$\begin{bmatrix} 0 \\ {}_\text{W}\boldsymbol{\omega}_\text{WB} \end{bmatrix} = 2\dot{\mathbf{q}}_\text{WB} \otimes \bar{\mathbf{q}}_\text{WB}, \tag{38}$$

$$= 2\mathbf{Q}\left(\dot{\mathbf{q}}_\text{WB} \right) \bar{\mathbf{q}}_\text{WB}, \tag{39}$$

$$= 2\bar{\mathbf{Q}}\left(\bar{\mathbf{q}}_\text{WB} \right) \dot{\mathbf{q}}_\text{WB}, \tag{40}$$

or in body fixed frame B

$$\begin{bmatrix} 0 \\ {}_\text{W}\boldsymbol{\omega}_\text{WB} \end{bmatrix} = 2\bar{\mathbf{q}}_\text{WB} \otimes \dot{\mathbf{q}}_\text{WB}. \tag{41}$$

More compactly, we can write

$$_\text{W}\boldsymbol{\omega}_\text{WB} = 2\mathbf{W}\left(\mathbf{q}_\text{WB} \right) \dot{\mathbf{q}}_\text{WB}, \tag{42}$$

$$_\text{W}\boldsymbol{\omega}_\text{WB} = 2\mathbf{W}'\left(\mathbf{q}_\text{WB} \right) \dot{\mathbf{q}}_\text{WB}, \tag{43}$$

with $\mathbf{W}\left(\mathbf{q}_\text{WB} \right)$ and $\mathbf{W}'\left(\mathbf{q}_\text{WB} \right)$ as defined in Section 3.3.5.

3.3 Changing Transformation Representation

This section lists the common conversions between different orientation representations.

3.3.1 Rotation Matrix to Euler Angles

For this conversion, we denote R_{ij} as the element in row i and column j of $\mathbf{R}_\text{WB}$.

$$\phi = \text{atan2}(R_{32}, R_{33}) \tag{44}$$

$$\theta = \text{atan2}(R_{31}, \sqrt{R_{32}^2 + R_{33}^2}) \tag{45}$$

$$\psi = \text{atan2}(R_{21}, R_{11}) \tag{46}$$

3.3.2 Euler Angles to Rotation Matrix

$$\mathbf{R}_\text{WB} = \begin{bmatrix} \text{c}(\psi)\,\text{c}(\theta) & \text{c}(\psi)\,\text{s}(\theta)\,\text{s}(\phi) - \text{s}(\psi)\,\text{c}(\phi) & \text{c}(\psi)\,\text{s}(\theta)\,\text{c}(\phi) + \text{s}(\psi)\,\text{s}(\phi) \\ \text{s}(\psi)\,\text{c}(\theta) & \text{s}(\psi)\,\text{s}(\theta)\,\text{s}(\phi) + \text{c}(\psi)\,\text{c}(\phi) & \text{s}(\psi)\,\text{s}(\theta)\,\text{c}(\phi) - \text{c}(\psi)\,\text{s}(\phi) \\ -\text{s}(\theta) & \text{c}(\theta)\,\text{s}(\phi) & \text{c}(\theta)\,\text{c}(\phi) \end{bmatrix} \tag{47}$$

3.3.3 Quaternion to Euler Angles

$$\phi = \text{atan2}(2q_w q_x + 2q_y q_z,\ q_w^2 - q_x^2 - q_y^2 + q_z^2) \tag{48}$$

$$\theta = -\text{asin}(2q_x q_z - 2q_w q_y) \tag{49}$$

$$\psi = \text{atan2}(2q_w q_z + 2q_x q_y,\ q_w^2 + q_x^2 - q_y^2 - q_z^2) \tag{50}$$

3.3.4 Euler Angles to Quaternion

$$\mathbf{q}_{\text{WB}} = \begin{bmatrix} q_w \\ q_x \\ q_y \\ q_z \end{bmatrix} = \begin{bmatrix} \cos\left(\frac{\phi}{2}\right)\cos\left(\frac{\theta}{2}\right)\cos\left(\frac{\psi}{2}\right) + \sin\left(\frac{\phi}{2}\right)\sin\left(\frac{\theta}{2}\right)\sin\left(\frac{\psi}{2}\right) \\ \sin\left(\frac{\phi}{2}\right)\cos\left(\frac{\theta}{2}\right)\cos\left(\frac{\psi}{2}\right) - \cos\left(\frac{\phi}{2}\right)\sin\left(\frac{\theta}{2}\right)\sin\left(\frac{\psi}{2}\right) \\ \cos\left(\frac{\phi}{2}\right)\sin\left(\frac{\theta}{2}\right)\cos\left(\frac{\psi}{2}\right) + \sin\left(\frac{\phi}{2}\right)\cos\left(\frac{\theta}{2}\right)\sin\left(\frac{\psi}{2}\right) \\ \cos\left(\frac{\phi}{2}\right)\cos\left(\frac{\theta}{2}\right)\sin\left(\frac{\psi}{2}\right) - \sin\left(\frac{\phi}{2}\right)\sin\left(\frac{\theta}{2}\right)\cos\left(\frac{\psi}{2}\right) \end{bmatrix} \tag{51}$$

3.3.5 Quaternion to Rotation Matrix

Unit quaternions are quaternions with unity norm. Throughout this section, we assume that

$$\|\mathbf{q}\| = 1. \tag{52}$$

To convert a vector $\mathbf{c}$ from the body frame B to the world frame W we use

$$\begin{bmatrix} 0 \\ _{\text{W}}\mathbf{c} \end{bmatrix} = \mathbf{q} \cdot \begin{bmatrix} 0 \\ _{\text{B}}\mathbf{c} \end{bmatrix} \cdot \mathbf{q}^{-1}, \tag{53}$$

$$= \mathbf{q} \cdot \begin{bmatrix} 0 \\ _{\text{B}}\mathbf{c} \end{bmatrix} \cdot \bar{\mathbf{q}}, \tag{54}$$

$$= \bar{\mathbf{Q}}\left(\mathbf{q}\right)^{\top} \mathbf{Q}\left(\mathbf{q}\right) \begin{bmatrix} 0 \\ _{\text{B}}\mathbf{c} \end{bmatrix}, \tag{55}$$

$$= \begin{bmatrix} 1 & \mathbf{0}^{\top} \\ \mathbf{0} & \mathbf{R}_{\text{WB}} \end{bmatrix} \begin{bmatrix} 0 \\ _{\text{B}}\mathbf{c} \end{bmatrix}. \tag{56}$$

From (55) and (56), it can be seen that the rotation matrix can be as

$$\mathbf{R}_{\text{WB}} = \begin{bmatrix} q_w^2 + q_x^2 - q_y^2 - q_z^2, & 2q_xq_y - 2q_wq_z, & 2q_wq_y + 2q_xq_z \\ 2q_wq_z + 2q_xq_y, & q_w^2 - q_x^2 + q_y^2 - q_z^2, & 2q_yq_z - 2q_wq_x \\ 2q_xq_z - 2q_wq_y, & 2q_wq_x + 2q_yq_z, & q_w^2 - q_x^2 - q_y^2 + q_z^2 \end{bmatrix}. \tag{57}$$

To rewrite (55) and (56), we define

$$\mathbf{W}(\mathbf{q}_{\text{WB}}) = \begin{bmatrix} -q_x & q_w & -q_z & q_y \\ -q_y & q_z & q_w & -q_x \\ -q_z & -q_y & q_x & q_w \end{bmatrix}, \tag{58}$$

$$\mathbf{W}'(\mathbf{q}_{\text{WB}}) = \begin{bmatrix} -q_x & q_w & q_z & -q_y \\ -q_y & -q_z & q_w & q_x \\ -q_z & q_y & -q_x & q_w \end{bmatrix}, \tag{59}$$

which allows us to write (57) as

$$\mathbf{R}_{\text{WB}} = \mathbf{W}(\mathbf{q}_{\text{WB}})\mathbf{W}'(\mathbf{q}_{\text{WB}})^{\top}. \tag{60}$$

Note that just as with rotation matrices, sequences of rotations are represented by products of quaternions. That is, for unit quaternions it holds that

$$\mathbf{R}\left(\mathbf{q} \cdot \mathbf{p}\right) = \mathbf{R}\left(\mathbf{q}\right)\mathbf{R}\left(\mathbf{p}\right). \tag{61}$$

Getting a quaternion directly from a rotation matrix is not straight forward but is also not used very often in our applications.

4 Dynamics of a Quadrotor

4.1 Coordinate System

For all our quadrotor math, we make use of the coordinate conventions as illustrated in Fig. 1 with the x-axis $\mathbf{x}_{\text{B}}$ pointing forward, the y-axis $\mathbf{y}_{\text{B}}$ pointing left and the z-axis $\mathbf{z}_{\text{B}}$ pointing up. Therefore, we define the gravity vector as $\mathbf{g} = [0 \ 0 \ -g]^{\top}$. The rotor numbering and rotation directions can also be seen in Fig. 1.
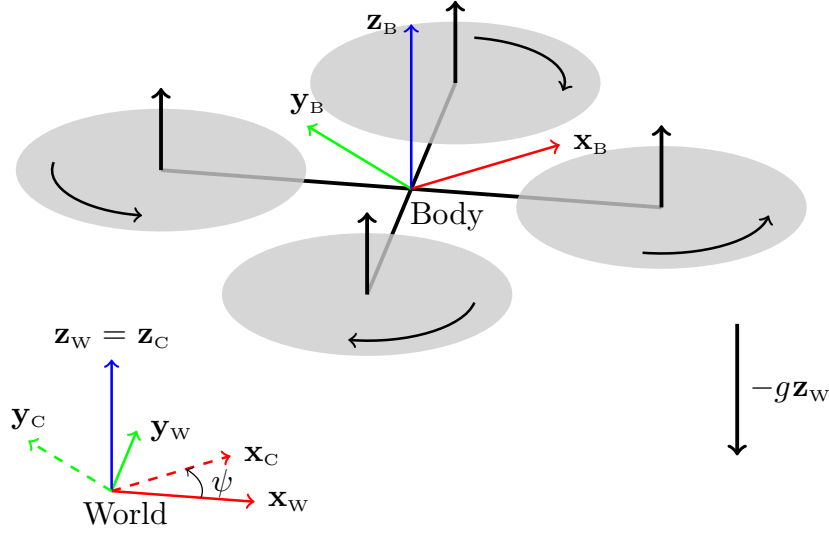
Figure 1: Quadrotor with coordinate system and rotor forces.

4.2   Variables and Parameters

Here we define some variables and parameters which we will use later

- m Mass of the quadrotor.

- l Length of a quadrotor arm.

- f_i Thrust produced by propeller i.

- κ Torque constant relating rotor thrust to its drag torque

- $\mathbf{c} = \begin{bmatrix} 0 \\ 0 \\ c \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \frac{1}{m}\Sigma f_i \end{bmatrix}$ Mass normalized, collective thrust of all the propellers.

- M_i Drag torque produced by propeller i.

- $\boldsymbol{\eta} = \begin{bmatrix} \frac{\sqrt{2}}{2}l(f_1 - f_2 - f_3 + f_4) \\ \frac{\sqrt{2}}{2}l(-f_1 - f_2 + f_3 + f_4) \\ \kappa(f_1 - f_2 + f_3 - f_4) \end{bmatrix}$ Torques acting on the body caused by the rotor thrusts.

- $_\mathrm{W}\mathbf{g} = \begin{bmatrix} 0 \\ 0 \\ -g \end{bmatrix}$ Gravitation vector in the world frame W, where $g = 9.81 \mathrm{m/s^2}$.

- $\mathbf{J} = \begin{bmatrix} J_{xx} & 0 & 0 \\ 0 & J_{yy} & 0 \\ 0 & 0 & J_{zz} \end{bmatrix}$ Moment of inertia matrix for rotations around the center of mass.

- $_\mathrm{W}\mathbf{p}_\mathrm{WB} = \begin{bmatrix} x \\ y \\ z \end{bmatrix}$ Position of the quadrotor in the world frame W.

- $_\mathrm{W}\mathbf{v}_\mathrm{WB} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix}$ Velocity of the quadrotor in the world frame W.

- $_\mathrm{W}\boldsymbol{\omega}_\mathrm{WB} = {}_\mathrm{B}\boldsymbol{\Omega} = \begin{bmatrix} p \\ q \\ r \end{bmatrix}$ Angular velocity of the quadrotor in body coordinates B. Note that the B system is attached to the quadrotor body frame, hence $_\mathrm{W}\boldsymbol{\omega}_\mathrm{WB} = {}_\mathrm{B}\boldsymbol{\Omega}$.

4.3 Using Euler Angles

In this section we use Euler angles and the corresponding rotation matrices as defined in Section 3.1, where the rotation matrix $\mathbf{R}_{\mathrm{WB}}(\phi, \theta, \psi)$ is a function of the euler angles.

We can evaluate (3) in the world frame as

$$
\begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ -g \end{bmatrix} + \mathbf{R}_{\mathrm{WB}}(\phi, \theta, \psi) \begin{bmatrix} 0 \\ 0 \\ c \end{bmatrix}, \tag{62}
$$

or in vectorized form

$$
\ddot{\mathbf{p}} = {}_{\mathrm{W}}\mathbf{g} + \mathbf{R}_{\mathrm{WB}}(\phi, \theta, \psi) \cdot \mathbf{c}. \tag{63}
$$

Next, we evaluate (8) in the body coordinate system B

$$
\begin{bmatrix} J_{xx} & 0 & 0 \\ 0 & J_{yy} & 0 \\ 0 & 0 & J_{zz} \end{bmatrix} \begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} + \begin{bmatrix} p \\ q \\ r \end{bmatrix} \times \begin{bmatrix} J_{xx} & 0 & 0 \\ 0 & J_{yy} & 0 \\ 0 & 0 & J_{zz} \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix} = \begin{bmatrix} \frac{\sqrt{2}}{2} l(f_1 - f_2 - f_3 + f_4) \\ \frac{\sqrt{2}}{2} l(-f_1 - f_2 + f_3 + f_4) \\ \kappa(f_1 - f_2 + f_3 - f_4) \end{bmatrix}, \tag{64}
$$

$$
\begin{bmatrix} J_{xx}\dot{p} + q \cdot r \left(J_{zz} - J_{yy} \right) \\ J_{yy}\dot{q} + p \cdot r \left(J_{xx} - J_{zz} \right) \\ J_{zz}\dot{r} + p \cdot q \left(J_{yy} - J_{xx} \right) \end{bmatrix} = \begin{bmatrix} \frac{\sqrt{2}}{2} l(f_1 - f_2 - f_3 + f_4) \\ \frac{\sqrt{2}}{2} l(-f_1 - f_2 + f_3 + f_4) \\ \kappa(f_1 - f_2 + f_3 - f_4) \end{bmatrix}, \tag{65}
$$

or in vectorized form

$$
\mathbf{J} \cdot {}_{\mathrm{B}}\dot{\boldsymbol{\omega}}_{\mathrm{WB}} + {}_{\mathrm{W}}\boldsymbol{\omega}_{\mathrm{WB}} \times \mathbf{J} \cdot {}_{\mathrm{W}}\boldsymbol{\omega}_{\mathrm{WB}} = \boldsymbol{\eta}. \tag{66}
$$

Now, we want to write the quadrotor dynamics in state space form. First, we chose the state vector as

$$
\mathbf{s} = \begin{bmatrix} x & y & z & \dot{x} & \dot{y} & \dot{z} & \phi & \theta & \psi & p & q & r \end{bmatrix}^\top, \tag{67}
$$

$$
= \begin{bmatrix} {}_{\mathrm{W}}\mathbf{p}_{\mathrm{WB}} & {}_{\mathrm{W}}\mathbf{v}_{\mathrm{WB}} & \phi & \theta & \psi & {}_{\mathrm{W}}\boldsymbol{\omega}_{\mathrm{WB}} \end{bmatrix}^\top. \tag{68}
$$

In the following we rewrite (62) and (65) in order to fit the state space form.

$$
\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix}, \tag{69}
$$

$$
\begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ -g \end{bmatrix} + \mathbf{R}_{\mathrm{WB}}(\phi, \theta, \psi) \begin{bmatrix} 0 \\ 0 \\ c \end{bmatrix}, \tag{70}
$$

$$
\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 1 & \mathrm{t}(\theta)\,\mathrm{s}(\phi) & \mathrm{t}(\theta)\,\mathrm{c}(\phi) \\ 0 & \mathrm{c}(\phi) & -\mathrm{s}(\phi) \\ 0 & \frac{\mathrm{s}(\phi)}{\mathrm{c}(\theta)} & \frac{\mathrm{c}(\phi)}{\mathrm{c}(\theta)} \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix}, \tag{71}
$$

$$
\begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} = \begin{bmatrix} \frac{1}{J_{xx}} \left(\frac{\sqrt{2}}{2} l(f_1 - f_2 - f_3 + f_4) - q \cdot r \left(J_{zz} - J_{yy} \right) \right) \\ \frac{1}{J_{yy}} \left(\frac{\sqrt{2}}{2} l(-f_1 - f_2 + f_3 + f_4) - p \cdot r \left(J_{xx} - J_{zz} \right) \right) \\ \frac{1}{J_{zz}} \left(\kappa(f_1 - f_2 + f_3 - f_4) - q \cdot p \left(J_{yy} - J_{xx} \right) \right) \end{bmatrix}, \tag{72}
$$

or in vectorized form

$$
{}_{\mathrm{W}}\dot{\mathbf{p}}_{\mathrm{WB}} = {}_{\mathrm{W}}\mathbf{v}_{\mathrm{WB}}, \tag{73}
$$

$$
{}_{\mathrm{W}}\dot{\mathbf{v}}_{\mathrm{WB}} = {}_{\mathrm{W}}\mathbf{g} + \mathbf{R}_{\mathrm{WB}}(\phi, \theta, \psi) \cdot \mathbf{c}, \tag{74}
$$

$$
\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 1 & \mathrm{t}(\theta)\,\mathrm{s}(\phi) & \mathrm{t}(\theta)\,\mathrm{c}(\phi) \\ 0 & \mathrm{c}(\phi) & -\mathrm{s}(\phi) \\ 0 & \frac{\mathrm{s}(\phi)}{\mathrm{c}(\theta)} & \frac{\mathrm{c}(\phi)}{\mathrm{c}(\theta)} \end{bmatrix} {}_{\mathrm{W}}\boldsymbol{\omega}_{\mathrm{WB}}, \tag{75}
$$

$$
{}_{\mathrm{B}}\dot{\boldsymbol{\omega}}_{\mathrm{WB}} = \mathbf{J}^{-1} \cdot \left(\boldsymbol{\eta} - {}_{\mathrm{W}}\boldsymbol{\omega}_{\mathrm{WB}} \times \mathbf{J} \cdot {}_{\mathrm{B}}\boldsymbol{\omega}_{\mathrm{WB}} \right). \tag{76}
$$

Finally, the derivation of the Euler angle derivatives in (71) and (75) is shown here. They are a function of the Euler angles and the body rates:

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} \dot{\phi}(\phi, \theta, \psi, p, q, r) \\ \dot{\theta}(\phi, \theta, \psi, p, q, r) \\ \dot{\psi}(\phi, \theta, \psi, p, q, r) \end{bmatrix}. \tag{77}$$

For this we use (25):

$$\begin{bmatrix} p \\ q \\ r \end{bmatrix} = \begin{bmatrix} 1 & 0 & -\mathrm{s}(\theta) \\ 0 & \mathrm{c}(\phi) & \mathrm{s}(\phi)\,\mathrm{c}(\theta) \\ 0 & -\mathrm{s}(\phi) & \mathrm{c}(\phi)\,\mathrm{c}(\theta) \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix}. \tag{78}$$

Inverting (78) finally leads to

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 1 & \mathrm{t}(\theta)\,\mathrm{s}(\phi) & \mathrm{t}(\theta)\,\mathrm{c}(\phi) \\ 0 & \mathrm{c}(\phi) & -\mathrm{s}(\phi) \\ 0 & \frac{\mathrm{s}(\phi)}{\mathrm{c}(\theta)} & \frac{\mathrm{c}(\phi)}{\mathrm{c}(\theta)} \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix}. \tag{79}$$

4.4 Using Rotation Matrices

We define $\mathbf{R}_{\mathrm{WB}}$ to be the rotation matrix which transforms vectors from B to W, but also describes the orientation of the quadrotor. In this section we will use the vectorized forms of the equations only.

We can evaluate (3) in the world frame as

$$\ddot{\mathbf{p}} = {}_{\mathrm{W}}\mathbf{g} + \mathbf{R}_{\mathrm{WB}} \cdot \mathbf{c}. \tag{80}$$

Since we use the same representation of the body rates as in Section 4.3, the evaluation of (8) remains the same as in (66).

We chose the state vector as

$$\mathbf{s} = \begin{bmatrix} {}_{\mathrm{W}}\mathbf{p}_{\mathrm{WB}} & {}_{\mathrm{W}}\mathbf{v}_{\mathrm{WB}} & \mathbf{R}_{\mathrm{WB}} & {}_{\mathrm{W}}\boldsymbol{\omega}_{\mathrm{WB}} \end{bmatrix}^{\top}. \tag{81}$$

In the following, we rewrite (80) and (66) in order to fit the state space form.

$${}_{\mathrm{W}}\dot{\mathbf{p}}_{\mathrm{WB}} = {}_{\mathrm{W}}\mathbf{v}_{\mathrm{WB}}, \tag{82}$$

$${}_{\mathrm{W}}\dot{\mathbf{v}}_{\mathrm{WB}} = {}_{\mathrm{W}}\mathbf{g} + \mathbf{R}_{\mathrm{WB}} \cdot \mathbf{c}, \tag{83}$$

$$\dot{\mathbf{R}}_{\mathrm{WB}} = \mathbf{R}_{\mathrm{WB}} \cdot {}_{\mathrm{B}}\hat{\boldsymbol{\omega}}_{\mathrm{WB}}, \tag{84}$$

$${}_{\mathrm{B}}\dot{\boldsymbol{\omega}}_{\mathrm{WB}} = \mathbf{J}^{-1} \cdot \left(\boldsymbol{\eta} - {}_{\mathrm{W}}\boldsymbol{\omega}_{\mathrm{WB}} \times \mathbf{J} \cdot {}_{\mathrm{W}}\boldsymbol{\omega}_{\mathrm{WB}} \right). \tag{85}$$

4.5 Using Quaternions

In this section, we use a quaternion $\mathbf{q}_{\mathrm{WB}}$, as defined in Section 3.2, to represent the orientation of the quadrotor. We will again use the vectorized forms of the equations only.

We can evaluate (3) in the world frame as

$$\ddot{\mathbf{p}} = {}_{\mathrm{W}}\mathbf{g} + \mathbf{q}_{\mathrm{WB}} \odot \mathbf{c}. \tag{86}$$

Since we use the same representation of the body rates as in Section 4.3, the evaluation of (8) remains the same as in (66).

We chose the state vector as

$$\mathbf{s} = \begin{bmatrix} {}_{\mathrm{W}}\mathbf{p}_{\mathrm{WB}} & {}_{\mathrm{W}}\mathbf{v}_{\mathrm{WB}} & \mathbf{q}_{\mathrm{WB}} & {}_{\mathrm{W}}\boldsymbol{\omega}_{\mathrm{WB}} \end{bmatrix}^{\top}. \tag{87}$$

In the following, we rewrite (86) and (66) in order to fit the state space form.

$$_\mathrm{W}\dot{\mathbf{p}}_\mathrm{WB} = {}_\mathrm{W}\mathbf{v}_\mathrm{WB}, \tag{88}$$

$$_\mathrm{W}\dot{\mathbf{v}}_\mathrm{WB} = {}_\mathrm{W}\mathbf{g} + \mathbf{q}_\mathrm{WB} \odot \mathbf{c}, \tag{89}$$

$$\dot{\mathbf{q}}_\mathrm{WB} = \mathbf{\Lambda}({}_\mathrm{W}\boldsymbol{\omega}_\mathrm{WB}) \cdot \mathbf{q}_\mathrm{WB}, \tag{90}$$

$$_\mathrm{B}\dot{\boldsymbol{\omega}}_\mathrm{WB} = \mathbf{J}^{-1} \cdot \left(\boldsymbol{\eta} - {}_\mathrm{W}\boldsymbol{\omega}_\mathrm{WB} \times \mathbf{J} \cdot {}_\mathrm{W}\boldsymbol{\omega}_\mathrm{WB}\right), \tag{91}$$

where

$$\mathbf{\Lambda}({}_\mathrm{W}\boldsymbol{\omega}_\mathrm{WB}) = \frac{1}{2}\begin{bmatrix} 0 & -p & -q & -r \\ p & 0 & r & -q \\ q & -r & 0 & p \\ r & q & -p & 0 \end{bmatrix} = \frac{1}{2}\bar{\mathbf{Q}}\left(\begin{bmatrix} 0 \\ {}_\mathrm{W}\boldsymbol{\omega}_\mathrm{WB} \end{bmatrix}\right) = \frac{1}{2}\bar{\mathbf{Q}}\left(\begin{bmatrix} 0 \\ p \\ q \\ r \end{bmatrix}\right). \tag{92}$$

We find (90) by using (41)

$$\dot{\mathbf{q}}_\mathrm{WB} = \frac{1}{2}\mathbf{q}_\mathrm{WB} \cdot \begin{bmatrix} 0 \\ {}_\mathrm{W}\boldsymbol{\omega}_\mathrm{WB} \end{bmatrix}, \tag{93}$$

$$= \frac{1}{2}\bar{\mathbf{Q}}\left(\begin{bmatrix} 0 \\ {}_\mathrm{W}\boldsymbol{\omega}_\mathrm{WB} \end{bmatrix}\right)\mathbf{q}_\mathrm{WB}. \tag{94}$$

Equation (90) can be integrated as described in Appendix A.

5   Control

TODO: Adapt according to [2] and [4].

This section describes our control algorithms that are used for OptiTrack based as well as for vision-based flying with our quadrotors. The controller is split into a high-level part and a low-level part. The high-level controller enables the quadrotor to track desired positions and velocities, whereas the low-level controller enables it to track desired attitudes or body rates.

5.1   Control Structure Overview

We can fly our quadrotors in two different set-ups namely when flying based only on on-board sensors (a camera and an IMU) and when flying in an OptiTrack motion capture system. The schematics of the control architecture for vision based flight are illustrated in Fig. **??**. In vision based flight, all the state estimation and high-level control runs on the Odroid which sends attitude commands to the PX4. In this setting, the user laptop is only used for feedback and some very high level inputs.

The schematics of the control architecture for OptiTrack based flight are illustrated in Fig. **??**. In OptiTrack based flight, the Odroid only serves as a relay for feedback from the PX4 via the WiFi link. In this setting, the user laptop receives pose measurements from the OptiTrack and uses them to compute a state estimate. To compensate for the round trip delay from when a pose was measured until the corresponding control command is received on the quadrotor, we also predict the state forward in time to compensate for this delay. This predicted state is then used on the laptop to compute body rate commands which are sent through Laird modules directly to the PX4.

5.2   Nomenclature

When describing the control and state estimation, we make use of some notation that we introduce here for clarity. We use a hat (e.g. $\hat{v}$) and a tilde (e.g. $\tilde{v}$) to denote an estimated and measured value, respectively. To describe the multiplication of two quaternions $\mathbf{q}_1$ and $\mathbf{q}_2$ we write $\mathbf{q}_1 \otimes \mathbf{q}_2$, and we write $\mathbf{q} \odot \mathbf{v}$ for the rotation of a vector $\mathbf{v}$ by the quaternion $\mathbf{q}$. Furthermore, we express the basis vectors of a coordinate system as e.g. $\mathbf{x}_\mathrm{B}$ which denotes the x-basis vector of the body coordinate system, as illustrated in Fig. 1. A prescript (e.g. $_B\mathbf{v}$) indicates that the vector $\mathbf{v}$ is

expressed in the body coordinate system. In this section, for simplicity, vectors that are expressed in world coordinates do not have prescripts with the exception of the body rates $\boldsymbol{\omega}$, which are always expressed in body coordinates.

5.3 Summary of the used Dynamical Model

For state estimation and control, we make use of the following dynamical model (as derived in Section 4.5) for our quadrotor:

$$\dot{\mathbf{R}} = \mathbf{v}, \tag{95}$$
$$\dot{\mathbf{v}} = \mathbf{g} + \mathbf{q} \odot \mathbf{c}, \tag{96}$$
$$\dot{\mathbf{q}} = \boldsymbol{\Lambda}(\boldsymbol{\omega}) \cdot \mathbf{q}, \tag{97}$$
$$\dot{\boldsymbol{\omega}} = \mathbf{J}^{-1} \cdot \left(\boldsymbol{\eta} - \boldsymbol{\omega} \times \mathbf{J} \cdot \boldsymbol{\omega} \right), \tag{98}$$

where $\mathbf{R} = [x\ y\ z]^\top$ and $\mathbf{v} = [v_x\ v_y\ v_z]^\top$ are the position and velocity in world coordinates, $\mathbf{q} = [q_w\ q_x\ q_y\ q_z]^\top$ is the orientation of the quadrotor's body coordinates with respect to the world coordinates, and $\boldsymbol{\omega} = [p\ q\ r]^\top$ denotes the body rates (roll, pitch and yaw, respectively) expressed in body coordinates. The skew-symmetric matrix $\boldsymbol{\Lambda}(\boldsymbol{\omega})$ is defined as

$$\boldsymbol{\Lambda}(\boldsymbol{\omega}) = \frac{1}{2} \begin{bmatrix} 0 & -p & -q & -r \\ p & 0 & r & -q \\ q & -r & 0 & p \\ r & q & -p & 0 \end{bmatrix}. \tag{99}$$

We define the gravity vector as $\mathbf{g} = [0\ 0\ -g]^\top$ with $g = 9.81\,\mathrm{m\,s^{-2}}$ and the second-order moment-of-inertia matrix of the quadrotor as $\mathbf{J} = \mathrm{diag}\,(J_{xx},\ J_{yy},\ J_{zz})$. The mass-normalized thrust vector is $\mathbf{c} = [0\ 0\ c]^\top$, with

$$mc = f_1 + f_2 + f_3 + f_4, \tag{100}$$

where m is the mass of the quadrotor and f_i are the four motor thrusts as illustrated in Fig. 1. The torque inputs $\boldsymbol{\eta}$ are composed of the single-rotor thrusts as

$$\boldsymbol{\eta} = \begin{bmatrix} \frac{\sqrt{2}}{2} l(f_1 - f_2 - f_3 + f_4) \\ \frac{\sqrt{2}}{2} l(-f_1 - f_2 + f_3 + f_4) \\ \kappa(f_1 - f_2 + f_3 - f_4) \end{bmatrix}, \tag{101}$$

where l is the quadrotor arm length and κ is the rotor-torque coefficient. Our coordinate-system conventions and rotor numbering are illustrated in Fig. 1.

5.4 High-Level Control

TODO: Control from [4], reference to TR [3].

    The high-level controller takes a reference state as input and computes the desired attitude or desired body rates, which are then sent to the low-level controller. A reference state consists of a reference position $\mathbf{p}_{ref}$, a reference velocity $\mathbf{v}_{ref}$, a reference acceleration $\mathbf{a}_{ref}$, and a reference heading ψ_{ref}. First, we describe the position controller, followed by the attitude controller. The two high-level control loops are synchronized and run at $50\,\mathrm{Hz}$.

5.4.1 Position Controller

To track a reference trajectory, we implemented a PD controller with feed-forward terms on the reference acceleration from the sampled trajectory and gravity:

$$\mathbf{a}_{des} = \mathbf{P}_{pos} \cdot (\mathbf{p}_{ref} - \hat{\mathbf{p}}) + \mathbf{D}_{pos} \cdot (\mathbf{v}_{ref} - \hat{\mathbf{v}}) + \mathbf{a}_{ref} - \mathbf{g}, \tag{102}$$

with gain matrices $\mathbf{P}_{pos} = \mathrm{diag}\,(p_{xy}, p_{xy}, p_z)$ and $\mathbf{D}_{pos} = \mathrm{diag}\,(d_{xy}, d_{xy}, d_z)$. To compute the desired normalized thrust c_{des}, we project the desired acceleration onto the current $_W\mathbf{e}_z^B$ axis

$$c_{des} = \mathbf{a}_{des} \cdot \mathbf{e}_z^B. \tag{103}$$

The output of the position controller is the desired accelerations $\mathbf{a}_{des}$. The desired acceleration, together with the reference heading ψ_{ref}, encodes the desired orientation as well as a mass normalized thrust c_{des}.

5.4.2 Attitude Controller

Since a quadrotor can only accelerate in its body z direction, $\mathbf{a}_{des}$ enforces two degrees of freedom of the desired attitude. The third degree of freedom is enforced by reference heading ψ_{ref}. Note that the rotation around the body z axis has no influence on the translational behaviour of the quadrotor. Therefore, we want to align the body z axis with the desired acceleration $\mathbf{a}_{des}$ by rotating around the body x and y axes and use rotations around the body z axis only to control the heading. Our quadrotors have much more attitude control authority on the x and y axes than on the z axis because there, they can make use of thrust differences as opposed to differences in rotor drag torques. The moment due to the maximum-possible thrust difference is much larger than the maximum possible difference of rotor drag torques. For this reason, we separate the attitude control into two parts as described in the following.

Desired Roll and Pitch Rates

From the current attitude estimate and the desired acceleration, we can compute the current and the desired body z axis, respectively, as

$$\hat{\mathbf{e}}_z^B = \hat{\mathbf{q}} \odot [0\ 0\ 1]^\top, \tag{104}$$

$$\mathbf{e}_{z,des}^B = \frac{\mathbf{a}_{des}}{\|\mathbf{a}_{des}\|}. \tag{105}$$

Now, we design an error quaternion that describes the necessary rotation to align these two vectors. To do so, we compute the angle α between the two vectors and a normal vector $\mathbf{n}$ to both of them:

$$\alpha = \arccos(\hat{\mathbf{e}}_z^B \cdot \mathbf{e}_{z,des}^B), \tag{106}$$

$$\mathbf{n} = \frac{\hat{\mathbf{e}}_z^B \times \mathbf{e}_{z,des}^B}{\left\|\hat{\mathbf{e}}_z^B \times \mathbf{e}_{z,des}^B\right\|}, \tag{107}$$

where $\mathbf{n}$, α are the rotation axis and angle of the error quaternion. Since we want to apply this rotation with respect to the current body orientation, we have to transform the rotation axis $\mathbf{n}$ into body coordinates using the current attitude estimate $\hat{\mathbf{q}}$

$$_B\mathbf{n} = \hat{\mathbf{q}}^{-1} \odot \mathbf{n}. \tag{108}$$

The error quaternion can then be constructed as

$$\mathbf{q}_{e,rp} = \begin{bmatrix} \cos(\frac{\alpha}{2}) \\ _B\mathbf{n}\sin(\frac{\alpha}{2}) \end{bmatrix}. \tag{109}$$

Note that if $\alpha = 0$, the rotation axis is undetermined and we set the error quaternion $\mathbf{q}_{e,rp}$ to be the identity directly. Also note that by construction, the z component of $\mathbf{q}_{e,rp}$ is always zero, which assures that no rotation around the body z axis is necessary to align $\hat{\mathbf{e}}_z^B$ with $\mathbf{e}_{z,des}^B$. From the error quaternion $\mathbf{q}_{e,rp}$, we can then compute the desired roll and pitch rates with the following control law:

$$\begin{bmatrix} p \\ q \end{bmatrix}_{des} = \begin{cases} 2 \cdot p_{rp} \cdot \mathbf{q}_{e,rp}^{(x,y)} & \text{if } \mathbf{q}_{e,rp}^{(w)} \geq 0 \\ -2 \cdot p_{rp} \cdot \mathbf{q}_{e,rp}^{(x,y)} & \text{if } \mathbf{q}_{e,rp}^{(w)} < 0 \end{cases}. \tag{110}$$

It can be shown that this control law is globally asymptotically stable and its discrete implementation is robust to measurement noise [1, 5].
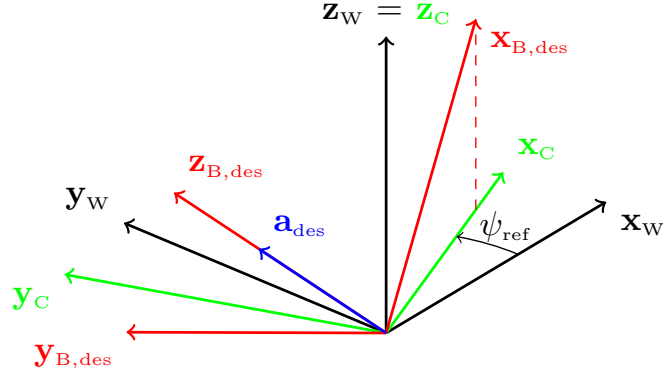
Figure 2: Coordinate frames used in the attitude controller. We make use of the coordinate frame C, which is obtained by rotating the world frame (W) by the reference heading ψ_{ref} to construct, together with the desired acceleration $\mathbf{a}_{des}$, the full desired orientation of the body frame (B).

Desired Yaw Rate

To compute the desired yaw rate r_{des}, we look at the heading error that remains after aligning $\hat{\mathbf{e}}_z^B$ with $\mathbf{e}_{z,des}^B$ with the above control law. To do so, we first compute the full desired attitude. For this, we make use of an intermediate coordinate system C, which is the world frame rotated around its z axis by the desired heading ψ_{ref} as illustrated in Fig. 2. The x and y axes of the coordinate system C are defined as

$$\mathbf{e}_x^C = [\cos(\psi_{ref}) \ \sin(\psi_{ref}) \ 0]^\top, \tag{111}$$

$$\mathbf{e}_y^C = [-\sin(\psi_{ref}) \ \cos(\psi_{ref}) \ 0]^\top. \tag{112}$$

The goal of rotating around the quadrotor's z axis is to align the projection of its x axis onto the world $x - y$ plane with $\mathbf{e}_x^C$. This forces the desired body x axis $\mathbf{e}_{x,des}^B$ to lie in a plane spanned by $\mathbf{e}_x^C$ and $\mathbf{e}_x^W$, which is fulfilled by

$$\mathbf{e}_{x,des}^B = \frac{\mathbf{e}_y^C \times \mathbf{e}_{z,des}^B}{\left\| \mathbf{e}_y^C \times \mathbf{e}_{z,des}^B \right\|}. \tag{113}$$

Note that if this cross product is zero, there are infinitely many rotations around the desired body z axis that achieve the desired heading. Therefore, in that case, we apply a desired yaw rate $r = 0$.

If the desired body z axis has a negative z component (i.e. $\mathbf{e}_{z,des}^B$ is pointing downwards), the projection of the computed desired body x axis into the horizontal plane will point in the opposite direction of $\mathbf{e}_x^C$, therefore we use the negation of it. From $\mathbf{e}_{x,des}^B$ and $\mathbf{e}_{z,des}^B$, we can then compute $\mathbf{e}_{y,des}^B$ as

$$\mathbf{e}_{y,des}^B = \frac{\mathbf{e}_{z,des}^B \times \mathbf{e}_{x,des}^B}{\left\| \mathbf{e}_{z,des}^B \times \mathbf{e}_{x,des}^B \right\|}. \tag{114}$$

Now, the full desired attitude $\mathbf{q}_{des}$ can be built from the three desired body axes $\mathbf{e}_{x,des}^B$, $\mathbf{e}_{y,des}^B$ and $\mathbf{e}_{z,des}^B$. Our definition of the heading has the advantage of being meaningful for any orientation of the quadrotor, which is not the case, for instance, when using a definition based on Euler angles. From this, we can then compute an error quaternion that describes the necessary rotation to achieve the full desired attitude after rotating by $\mathbf{q}_{e,rp}$ as

$$\mathbf{q}_{e,y} = (\hat{\mathbf{q}} \otimes \mathbf{q}_{e,rp})^{-1} \otimes \mathbf{q}_{des}. \tag{115}$$

Note that the x and y components of $\mathbf{q}_{e,y}$ are always zero. Similarly to the desired roll and pitch rate, we can now compute the desired yaw rate from $\mathbf{q}_{e,y}^{(z)}$ with a gain p_{yaw} as

$$
r_{des} = \begin{cases} 2 \cdot p_{yaw} \cdot \mathbf{q}_{e,y}^{(z)} & \text{if } \mathbf{q}_{e,y}^{(w)} \geq 0 \\ -2 \cdot p_{yaw} \cdot \mathbf{q}_{e,y}^{(z)} & \text{if } \mathbf{q}_{e,y}^{(w)} < 0 \end{cases}.
\tag{116}
$$

Splitting the attitude control into these two parts allows us to have different gains for p_{rp} and p_{yaw}, which is desirable due to different control limits.

5.5   Low-Level Control

TODO: Update to [2].

The commands sent to the low-level control on the PX4 are the desired body rates $\boldsymbol{\omega}_{des}$ and the desired mass-normalized thrust c_{des}. From the desired body rates $\boldsymbol{\omega}_{des}$ and the measured body rates $\tilde{\boldsymbol{\omega}}$, we can compute the desired torques $\boldsymbol{\eta}_{des}$ with a feedback linearizing control scheme:

$$
\boldsymbol{\eta}_{des} = \mathbf{J} \cdot \mathbf{P}_{att} \cdot (\boldsymbol{\omega}_{des} - \hat{\boldsymbol{\omega}}) + \hat{\boldsymbol{\omega}} \times \mathbf{J} \cdot \hat{\boldsymbol{\omega}},
\tag{117}
$$

where $\mathbf{P}_{att} = \text{diag}\,(p_{pq}, p_{pq}, p_r)$. Then, we can substitute $\boldsymbol{\eta}_{des}$ and c_{des} into (101) and (100) and solve them for the desired nominal rotor thrusts (see Section **??**) that must be applied:

$$
\begin{bmatrix} \check{f}_1 \\ \check{f}_2 \\ \check{f}_3 \\ \check{f}_4 \end{bmatrix} = \begin{bmatrix} \frac{1}{4\lambda_1} \cdot \left(m \cdot c_{des} + \frac{1}{\kappa} \cdot \boldsymbol{\eta}_{des}^y - \frac{\sqrt{2}}{l} \cdot \boldsymbol{\eta}_{des}^p + \frac{\sqrt{2}}{l} \cdot \boldsymbol{\eta}_{des}^r \right) \\ \frac{1}{4\lambda_2} \cdot \left(m \cdot c_{des} - \frac{1}{\kappa} \cdot \boldsymbol{\eta}_{des}^y - \frac{\sqrt{2}}{l} \cdot \boldsymbol{\eta}_{des}^p - \frac{\sqrt{2}}{l} \cdot \boldsymbol{\eta}_{des}^r \right) \\ \frac{1}{4\lambda_3} \cdot \left(m \cdot c_{des} + \frac{1}{\kappa} \cdot \boldsymbol{\eta}_{des}^y + \frac{\sqrt{2}}{l} \cdot \boldsymbol{\eta}_{des}^p - \frac{\sqrt{2}}{l} \cdot \boldsymbol{\eta}_{des}^r \right) \\ \frac{1}{4\lambda_4} \cdot \left(m \cdot c_{des} - \frac{1}{\kappa} \cdot \boldsymbol{\eta}_{des}^y + \frac{\sqrt{2}}{l} \cdot \boldsymbol{\eta}_{des}^p + \frac{\sqrt{2}}{l} \cdot \boldsymbol{\eta}_{des}^r \right) \end{bmatrix},
\tag{118}
$$

with κ being the rotor-torque coefficient and λ_i being the rotor fitness coefficients as described in Section **??**.

5.5.1   Rotor Thrusts Saturation

TODO: Describe how we saturate the single rotor thrusts according to [2].

5.6   Delay Compensation

TODO: about state predictor

6   Trajectory Generation

6.1   Computing Relevant Maxima from a Reference State

When designing a trajectory, it is often desired to enforce some limits which need to be computed from a desired state. This section shows how to compute to most relevant values for quadrotors, namely the speed, the collective thrust, and the norm of the roll and pitch rates (especially relevant for vision-based quadrotors). We assume that from a sample point on a given trajectory, we can get the reference state $\mathbf{s}_{des}$ containing the position, velocity, acceleration, and jerk.

$$
\mathbf{s}_{des} = \begin{bmatrix} \mathbf{R}_{des} & \mathbf{v}_{des} & \mathbf{a}_{des} & \mathbf{j}_{des} \end{bmatrix}
\tag{119}
$$

Speed and Collective Thrust

The speed can simply be obtained by taking the norm of the reference velocity

$$
v = \|\mathbf{v}_{des}\|,
\tag{120}
$$

and the collective thrust can be computed as

$$
c = \|\mathbf{a}_{des} - \mathbf{g}\|,
\tag{121}
$$

with $\mathbf{g} = [0\ 0\ -g]^\top$.

Roll and Pitch Rate Norm

We introduce the desired acceleration vector in world coordinates as

$$\mathbf{f} = \mathbf{a}_{des} - \mathbf{g} = \mathbf{R}_{\mathrm{WB}} \cdot \begin{bmatrix} 0 \\ 0 \\ c \end{bmatrix} = \mathbf{R}_{\mathrm{WB}} \cdot \mathbf{c}, \tag{122}$$

where $\mathbf{g}$ is defined as above. Note that $\|\mathbf{f}\| = c$ since the rotation of a vector does not change is length. Therefore we can say

$$\mathbf{R}_{\mathrm{WB}} \cdot \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = \frac{\mathbf{f}}{\|\mathbf{f}\|} = \bar{\mathbf{f}}. \tag{123}$$

Taking the derivative leads to

$$\mathbf{R}_{\mathrm{WB}} \cdot \hat{\boldsymbol{\omega}} \cdot \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = \dot{\bar{\mathbf{f}}}, \tag{124}$$

$$\hat{\boldsymbol{\omega}} \cdot \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = \mathbf{R}_{\mathrm{WB}}^{\top} \cdot \dot{\bar{\mathbf{f}}}, \tag{125}$$

$$\begin{bmatrix} \omega_y \\ -\omega_x \\ 0 \end{bmatrix} = \mathbf{R}_{\mathrm{WB}}^{\top} \cdot \dot{\bar{\mathbf{f}}}, \tag{126}$$

where $\dot{\bar{\mathbf{f}}}$ is computed as

$$\dot{\bar{\mathbf{f}}} = \frac{d}{dt} \left(\frac{\mathbf{f}}{\|\mathbf{f}\|} \right), \tag{127}$$

$$= \frac{\dot{\mathbf{f}} \cdot \|\mathbf{f}\| - \mathbf{f} \cdot \frac{\mathbf{f}^{\top} \cdot \dot{\mathbf{f}}}{\|\mathbf{f}\|}}{\|\mathbf{f}\|^2}, \tag{128}$$

$$= \frac{\dot{\mathbf{f}}}{\|\mathbf{f}\|} - \frac{\mathbf{f} \cdot \mathbf{f}^{\top} \cdot \dot{\mathbf{f}}}{\|\mathbf{f}\|^3}, \tag{129}$$

$$= \frac{\mathbf{j}}{\|\mathbf{f}\|} - \frac{\mathbf{f} \cdot \mathbf{f}^{\top} \cdot \mathbf{j}}{\|\mathbf{f}\|^3}, \tag{130}$$

using the fact that $\dot{\mathbf{f}} = \mathbf{j}$. With all this we can write

$$\begin{bmatrix} \omega_y \\ -\omega_x \\ 0 \end{bmatrix} = \mathbf{R}_{\mathrm{WB}}^{\top} \cdot \left(\frac{\mathbf{j}}{\|\mathbf{f}\|} - \frac{\mathbf{f} \cdot \mathbf{f}^{\top} \cdot \mathbf{j}}{\|\mathbf{f}\|^3} \right), \tag{131}$$

$$= \mathbf{R}_{\mathrm{WB}}^{\top} \cdot \frac{\mathbf{j}}{c} - \mathbf{R}_{\mathrm{WB}}^{\top} \frac{\mathbf{R}_{\mathrm{WB}} \cdot \mathbf{c} \cdot \mathbf{c}^{\top} \cdot \mathbf{R}_{\mathrm{WB}}^{\top} \cdot \mathbf{j}}{c^3}, \tag{132}$$

$$= \mathbf{R}_{\mathrm{WB}}^{\top} \cdot \frac{\mathbf{j}}{c} - \frac{\mathbf{c} \cdot \mathbf{c}^{\top} \cdot \mathbf{R}_{\mathrm{WB}}^{\top} \cdot \mathbf{j}}{c^3}. \tag{133}$$

Noting that

$$\mathbf{c} \cdot \mathbf{c}^{\top} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & c^2 \end{bmatrix}, \tag{134}$$

and using (18), we get

$$\begin{bmatrix} \omega_y \\ -\omega_x \\ 0 \end{bmatrix} = \begin{bmatrix} (\mathbf{e}_x^B)^{\top} \cdot \frac{\mathbf{j}}{c} \\ (\mathbf{e}_y^B)^{\top} \cdot \frac{\mathbf{j}}{c} \\ (\mathbf{e}_z^B)^{\top} \cdot \frac{\mathbf{j}}{c} \end{bmatrix} - \begin{bmatrix} 0 \\ 0 \\ (\mathbf{e}_z^B)^{\top} \cdot \frac{\mathbf{j}}{c} \end{bmatrix}. \tag{135}$$

Note that $\left\| \mathbf{R}_{\mathrm{WB}}^{\top} \cdot \frac{\mathbf{j}}{c} \right\| = \left\| \frac{\mathbf{j}}{c} \right\|$ and that $\mathbf{e}_z^B$ can be retrieved from the desired acceleration as

$$\mathbf{e}_z^B = \frac{\mathbf{a}_{des} - \mathbf{g}}{\|\mathbf{a}_{des} - \mathbf{g}\|}. \tag{136}$$

Therefore, we can compute the norm of the roll and pitch body rates as

$$\|\omega_{x,y}\| = \left\| \begin{bmatrix} \omega_y \\ -\omega_x \\ 0 \end{bmatrix} \right\| = \sqrt{ \left\| \frac{\mathbf{j}}{c} \right\|^2 - \left((\mathbf{e}_z^B)^{\top} \cdot \frac{\mathbf{j}}{c} \right)^2 } \tag{137}$$

Appendices

A   Zeroth Order Integration of a Unit Quaternion

Simply integrating a quaternion derivative component-wise does not guarantee to result in a unit quaternion and requires normalization. Under the assumption of a discrete time step in the interval $[t, t+\Delta t]$ with constant body rates $_\mathrm{W}\boldsymbol{\omega}_\mathrm{WB}$, a nice solution for the integration exists (from [6]) which does not require to normalize the quaternion. First we restate (90)

$$\dot{\mathbf{q}}_\mathrm{WB} = \boldsymbol{\Lambda}(_\mathrm{W}\boldsymbol{\omega}_\mathrm{WB}) \cdot \mathbf{q}_\mathrm{WB}. \tag{138}$$

In literature we find the solution to this differential equation with starting point $\mathbf{q}_\mathrm{WB}(t)$ to be

$$\mathbf{q}_\mathrm{WB}(t + \Delta t) = e^{(\boldsymbol{\Lambda}(_\mathrm{W}\boldsymbol{\omega}_\mathrm{WB})\Delta t)} \cdot \mathbf{q}_\mathrm{WB}(t). \tag{139}$$

The matrix exponential is defined as

$$e^{(\boldsymbol{\Lambda}(_\mathrm{W}\boldsymbol{\omega}_\mathrm{WB})\Delta t)} = \mathbf{I}_4 + \boldsymbol{\Lambda}(_\mathrm{W}\boldsymbol{\omega}_\mathrm{WB}) \cdot \Delta t + \frac{1}{2!}\left(\boldsymbol{\Lambda}(_\mathrm{W}\boldsymbol{\omega}_\mathrm{WB}) \cdot \Delta t\right)^2 + \frac{1}{3!}\left(\boldsymbol{\Lambda}(_\mathrm{W}\boldsymbol{\omega}_\mathrm{WB}) \cdot \Delta t\right)^3 + \ldots \tag{140}$$

It can easily be verified that

$$\left(\boldsymbol{\Lambda}(_\mathrm{W}\boldsymbol{\omega}_\mathrm{WB})\right)^2 = -\frac{1}{4}\|_\mathrm{W}\boldsymbol{\omega}_\mathrm{WB}\|^2 \cdot \mathbf{I}_4, \tag{141}$$

where

$$\|_\mathrm{W}\boldsymbol{\omega}_\mathrm{WB}\| = \sqrt{p^2 + q^2 + r^2}. \tag{142}$$

Using (141) and rearranging the terms, (140) can be written as

$$e^{(\boldsymbol{\Lambda}(_\mathrm{W}\boldsymbol{\omega}_\mathrm{WB})\Delta t)} = \mathbf{I}_4 \cdot \left(1 - \frac{\left(\|_\mathrm{W}\boldsymbol{\omega}_\mathrm{WB}\|\frac{\Delta t}{2}\right)^2}{2!} + \frac{\left(\|_\mathrm{W}\boldsymbol{\omega}_\mathrm{WB}\|\frac{\Delta t}{2}\right)^4}{4!} - \ldots\right), \tag{143}$$

$$+ \frac{2}{\|_\mathrm{W}\boldsymbol{\omega}_\mathrm{WB}\|}\boldsymbol{\Lambda}(_\mathrm{W}\boldsymbol{\omega}_\mathrm{WB}) \cdot \left(\frac{\left(\|_\mathrm{W}\boldsymbol{\omega}_\mathrm{WB}\|\frac{\Delta t}{2}\right)}{1!} - \frac{\left(\|_\mathrm{W}\boldsymbol{\omega}_\mathrm{WB}\|\frac{\Delta t}{2}\right)^3}{3!} + \ldots\right),$$

$$= \mathbf{I}_4 \cdot \cos\left(\frac{\|_\mathrm{W}\boldsymbol{\omega}_\mathrm{WB}\|\Delta t}{2}\right) + \frac{2}{\|_\mathrm{W}\boldsymbol{\omega}_\mathrm{WB}\|} \cdot \boldsymbol{\Lambda}(_\mathrm{W}\boldsymbol{\omega}_\mathrm{WB}) \cdot \sin\left(\frac{\|_\mathrm{W}\boldsymbol{\omega}_\mathrm{WB}\|\Delta t}{2}\right). \tag{144}$$

So the state update of the quaternion becomes

$$\mathbf{q}_\mathrm{WB}(t + \Delta t) = \left(\mathbf{I}_4 \cdot \cos\left(\frac{\|_\mathrm{W}\boldsymbol{\omega}_\mathrm{WB}\|\Delta t}{2}\right) + \frac{2}{\|_\mathrm{W}\boldsymbol{\omega}_\mathrm{WB}\|} \cdot \boldsymbol{\Lambda}(_\mathrm{W}\boldsymbol{\omega}_\mathrm{WB}) \cdot \sin\left(\frac{\|_\mathrm{W}\boldsymbol{\omega}_\mathrm{WB}\|\Delta t}{2}\right)\right) \cdot \mathbf{q}_\mathrm{WB}(t). \tag{145}$$

B   Differentiating the Angular Momentum

This section shows how to differentiate $\mathbf{L}_\mathrm{O}$. First, we restate (5) for the definition of the angular momentum

$$\mathbf{L}_\mathrm{O} = \mathbf{R}_\mathrm{OS} \times \mathbf{p} + \mathbf{I}_\mathrm{S} \cdot \boldsymbol{\Omega}. \tag{146}$$

Later we will use Euler's differentiation rule, therefore we restate it here. Euler's differentiation rule is used to differentiate in a body fixed coordinate system B

$$_\mathrm{B}\dot{\mathbf{c}} = \left(_\mathrm{B}\mathbf{c}\right)' + _\mathrm{W}\boldsymbol{\omega}_\mathrm{WB} \times {}_\mathrm{B}\mathbf{c}. \tag{147}$$

Note the special notation for $\left(_\mathrm{B}\mathbf{c}\right)'$ which is just the differentiation over time of $_\mathrm{B}\mathbf{c}$ which is different than the differentiation expressed in the B system $_\mathrm{B}\dot{\mathbf{c}}$. Now, lets start to calculate $\dot{\mathbf{L}}_\mathrm{O}$. For better

18

readability we differentiate the two terms in (146) separately:

$$(\mathbf{R}_{\text{OS}} \times \mathbf{p})^{\cdot} = \dot{\mathbf{R}}_{\text{OS}} \times \mathbf{p} + \mathbf{R}_{\text{OS}} \times \dot{\mathbf{p}}, \tag{148}$$

$$= v_S \times \mathbf{p} + \mathbf{R}_{\text{OS}} \times \dot{\mathbf{p}}, \tag{149}$$

$$= v_S \times m \cdot v_S + \mathbf{R}_{\text{OS}} \times \dot{\mathbf{p}}, \tag{150}$$

$$= \mathbf{R}_{\text{OS}} \times \dot{\mathbf{p}}. \tag{151}$$

To differentiate $\mathbf{I}_{\text{S}} \cdot \mathbf{\Omega}$ we evaluate it in the body fixed coordinate system B. We do this because the second moment of inertia matrix $_{\text{B}}\mathbf{I}_{\text{S}}$ is constant in a body fixed frame

$$_B(\mathbf{I}_{\text{S}} \cdot \mathbf{\Omega})^{\cdot} = (_{\text{B}}\mathbf{I}_{\text{SB}}\mathbf{\Omega})' + _{\text{B}}\mathbf{\Omega} \times _{\text{B}}\mathbf{I}_{\text{S}} \cdot _{\text{B}}\mathbf{\Omega}, \tag{152}$$

$$= (_{\text{B}}\mathbf{I}_{\text{S}})'^{\,0}_{\text{B}}\mathbf{\Omega} + _{\text{B}}\mathbf{I}_{\text{S}} (_{\text{B}}\mathbf{\Omega})' + _{\text{B}}\mathbf{\Omega} \times _{\text{B}}\mathbf{I}_{\text{S}} \cdot _{\text{B}}\mathbf{\Omega}, \tag{153}$$

$$= _{\text{B}}\mathbf{I}_{\text{S}} (_{\text{B}}\mathbf{\Omega})' + _{\text{B}}\mathbf{\Omega} \times _{\text{B}}\mathbf{I}_{\text{S}} \cdot _{\text{B}}\mathbf{\Omega}. \tag{154}$$

To rewrite $(_{\text{B}}\mathbf{\Omega})'$ we differentiate the angular velocity of the body using euler's formula (147)

$$_{\text{B}}\dot{\mathbf{\Omega}} = (_{\text{B}}\mathbf{\Omega})' + _{\text{W}}\boldsymbol{\omega}_{\text{WB}} \times _{\text{B}}\mathbf{\Omega}, \tag{155}$$

$$= (_{\text{B}}\mathbf{\Omega})' + _{\text{B}}\mathbf{\Omega} \times _{\text{B}}\mathbf{\Omega}^{\,0}, \tag{156}$$

$$= (_{\text{B}}\mathbf{\Omega})'. \tag{157}$$

Inserting (157) in (154) we can continue with the differentiation of the second term

$$_B(\mathbf{I}_{\text{S}} \cdot \mathbf{\Omega})^{\cdot} = _{\text{B}}\mathbf{I}_{\text{SB}}\dot{\mathbf{\Omega}} + _{\text{B}}\mathbf{\Omega} \times _{\text{B}}\mathbf{I}_{\text{S}} \cdot _{\text{B}}\mathbf{\Omega}. \tag{158}$$

This result is valid in any coordinate system not only the body fixed, therefore we can write (158) as

$$(\mathbf{I}_{\text{S}} \cdot \mathbf{\Omega})^{\cdot} = \mathbf{I}_{\text{S}}\dot{\mathbf{\Omega}} + \mathbf{\Omega} \times \mathbf{I}_{\text{S}} \cdot \mathbf{\Omega}. \tag{159}$$

Combining the results (151) and (159) we finally find

$$\dot{\mathbf{L}}_O = \mathbf{R}_{\text{OS}} \times \dot{\mathbf{p}} + \mathbf{I}_{\text{S}}\dot{\mathbf{\Omega}} + \mathbf{\Omega} \times \mathbf{I}_{\text{S}} \cdot \mathbf{\Omega}, \tag{160}$$

$$= \mathbf{R}_{\text{OS}} \times \dot{\mathbf{p}} + \mathbf{I}_{\text{S}}\mathbf{\Psi} + \mathbf{\Omega} \times \mathbf{I}_{\text{S}} \cdot \mathbf{\Omega}. \tag{161}$$

Again this result is valid in any coordinate system.

C   Proof of not Measuring Gravity in Flight

When a quadrotor is flying, its accelerometer is not measuring gravity but all the other forces acting on the quadrotor (Thrust, Disturbances, ...). In fact, the accelerometer is never measuring gravity. Just by definition, the magnitude of the measured accelerations in hover, or when standing on the ground, is equal to the gravitational acceleration. But note that in these cases we measure $\tilde{\mathbf{a}} = -\mathbf{g}$ which is due to the upwards force acting on the quadrotor in order to have it standing still in a world frame. We demonstrate this using a simple model for an accelerometer. Fig. 3 shows the schematics of a model of an accelerometer in 2D. We model the accelerometer as a mass m_M connected by springs in all three axes to its casing which is rigidly attached to the quadrotor with mass m_q. The springs are modelled to be linear with spring constant k_s. The output of the accelerometer is the estimated acceleration of the quadrotor $\tilde{\mathbf{a}}_q$, which is measured by the deflection $\mathbf{d}$ of the mass m_M. The force of the spring due to its deflection is

$$\mathbf{f}_s = k_s \cdot \mathbf{d} = m_M \cdot \tilde{\mathbf{a}}_q. \tag{162}$$

By establishing the equilibrium of forces for the mass m_M and the quadrotor we get

$$\mathbf{a}_M \cdot m_M = m_M \cdot \mathbf{g} + \mathbf{f}_s, \tag{163}$$

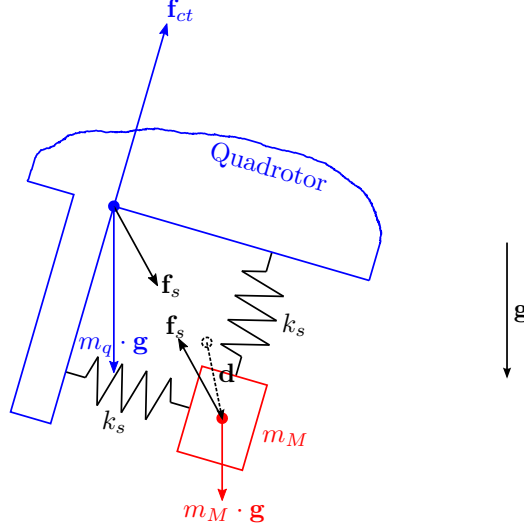$$\mathbf{a}_q \cdot m_q = \mathbf{f}_{ct} + \mathbf{f}_d + m_q \cdot \mathbf{g} - \mathbf{f}_s, \tag{164}$$

Figure 3: Schematics of the accelerometer model. The mass m_M is connected to the quadrotor by linear springs with spring constant k_s. No dynamical effects of the mass m_M and its deflection $\mathbf{d}$ are considered.

where $\mathbf{a}_M$ and $\mathbf{a}_q$ are the actual accelerations of the mass m_M and the quadrotor represented in the world frame, respectively. The vector $\mathbf{g} = [0\ 0\ -g]^\top$ denotes the gravitational acceleration, $\mathbf{f}_{ct} = m_q \cdot \mathbf{c}$ is the force due to the applied collective thrust and $\mathbf{f}_d$ are the forces due to external disturbances. By substituting (162) into (163) and (164), we get

$$\mathbf{a}_M \cdot m_M = m_M \cdot \mathbf{g} + m_M \cdot \tilde{\mathbf{a}}_q, \tag{165}$$

$$\mathbf{a}_q \cdot m_q = \mathbf{f}_{ct} + \mathbf{f}_d + m_q \cdot \mathbf{g} - m_M \cdot \tilde{\mathbf{a}}_q. \tag{166}$$

We neglect any dynamics of the accelerometer mass and assume that for a given acceleration the corresponding deflection is reached instantly. Therefore, we can set $\mathbf{a}_M \equiv \mathbf{a}_q$. Using this and substituting (165) into (166), we get

$$(\tilde{\mathbf{a}}_q + \mathbf{g}) \cdot m_q = \mathbf{f}_{ct} + \mathbf{f}_d + m_q \cdot \mathbf{g} - m_M \cdot \tilde{\mathbf{a}}_q. \tag{167}$$

Since the mass m_M is very tiny, we can assume that $m_M \ll m_q$. Therefore, we can solve (167) for the measured acceleration of the quadrotor $\tilde{\mathbf{a}}_q$ as

$$\tilde{\mathbf{a}}_q = \frac{1}{m_q} \cdot (\mathbf{f}_{ct} + \mathbf{f}_d). \tag{168}$$

This shows that the accelerometer only measures accelerations due to the collective thrust applied on the quadrotor. Because of this fact, it is not possible to estimate the attitude of a quadrotor in flight without drift (even in roll and pitch) by only using IMU measurements. Nonetheless, when the quadrotor has contact to a static object, and therefore is constrained in position, roll and pitch can be estimated without drift by only using IMU measurements.

References

[1] Dario Brescianini, Markus Hehn, and Raffaello D'Andrea. Nonlinear quadrocopter attitude control. Technical report, Department of Mechanical and Process Engineering, ETHZ, October 2013.

[2] Matthias Faessler, Davide Falanga, and Davide Scaramuzza. Thrust mixing, saturation, and body-rate control for accurate aggressive quadrotor flight. *IEEE Robot. Autom. Lett.*, 2(2):476–482, April 2017.

[3] Matthias Faessler, Antonio Franchi, and Davide Scaramuzza. Detailed derivations of 'Differential flatness of quadrotor dynamics subject to rotor drag for accurate tracking of high-speed trajectories'. *arXiv e-prints*, December 2017.

[4] Matthias Faessler, Antonio Franchi, and Davide Scaramuzza. Differential flatness of quadrotor dynamics subject to rotor drag for accurate tracking of high-speed trajectories. *IEEE Robot. Autom. Lett.*, 3(2):620–626, April 2018.

[5] Christopher G. Mayhew, Ricardo G. Sanfelice, and Andrew R. Teel. Quaternion-based hybrid control for robust global attitude tracking. *IEEE Trans. Autom. Control*, 56(11):2555–2566, November 2011.

[6] N. Trawny and S. I. Roumeliotis. Indirect Kalman filter for 3D attitude estimation. Technical Report 2005-002, University of Minnesota, Dept. of Comp. Sci. & Eng., March 2005.