Theory and Math Behind RPG Quadrotor Control

Matthias Faessler and Flavio Fontana

25 сентября 2018 г.

Содержание

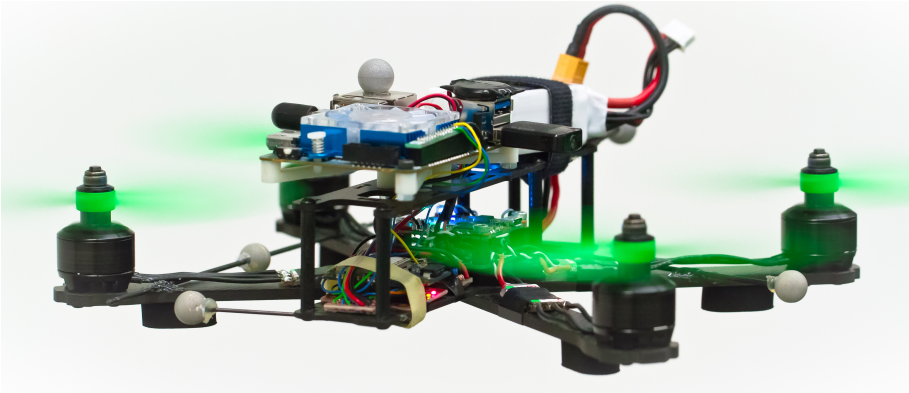

Рис. 1: First-person-view racing inspired quadrotor platform used for the experiments presented in [1].

1 Введение

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`. The control algorithms presented here are published in [1], for which we developed the quadrotor shown in Fig. 1, and in [2].

2 Nomenclature

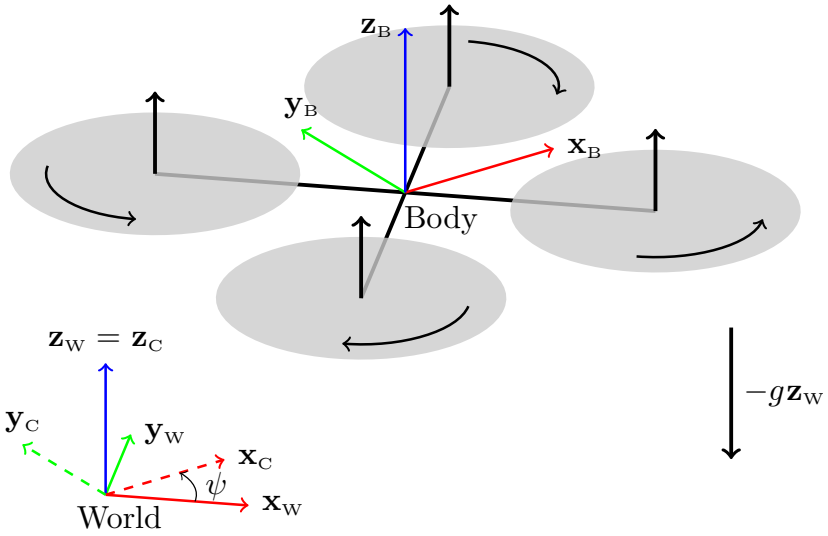

Рис. 2: Schematics of the considered quadrotor model with the used coordinate systems and rotor forces.

In this work, we make use of a world frame W with orthonormal basis $\{\mathbf{x}_\mathrm{W}, \mathbf{y}_\mathrm{W}, \mathbf{z}_\mathrm{W}\}$ represented in world coordinates and a body frame B with orthonormal basis $\{\mathbf{x}_\mathrm{B}, \mathbf{y}_\mathrm{B}, \mathbf{z}_\mathrm{B}\}$

also represented in world coordinates. The body frame is fixed to the quadrotor with an origin coinciding with its center of mass as depicted in Fig. 2. The rotor numbering and rotation directions can also be seen in Fig. 2. The quadrotor is subject to a gravitational acceleration g in negative $\mathbf{z}_\mathrm{W}$ direction. We denote the position of the quadrotor's center of mass as $\mathbf{p}$, and its derivatives, velocity, acceleration, jerk, and snap as $\mathbf{v}$, $\mathbf{a}$, $\mathbf{j}$, and $\mathbf{s}$, respectively. We represent the quadrotor's orientation as a rotation matrix $\mathbf{R} = \begin{bmatrix}\mathbf{x}_\mathrm{B} & \mathbf{y}_\mathrm{B} & \mathbf{z}_\mathrm{B}\end{bmatrix}$ and its body rates (i.e., the angular velocity) as $\boldsymbol{\omega}$ represented in body coordinates. Finally, we denote quantities that can be computed from a reference trajectory as *reference values* and quantities that are computed by an outer loop feedback control law and passed to an inner loop controller as *desired values*. A reference trajectory consists of the reference position $\mathbf{p}_\mathrm{ref}$, the reference velocity $\mathbf{v}_\mathrm{ref}$, the reference acceleration $\mathbf{a}_\mathrm{ref}$, the reference jerk $\mathbf{j}_\mathrm{ref}$, and the reference snap $\mathbf{s}_\mathrm{ref}$, as well as a reference heading ψ_ref, a reference heading rate $\dot{\psi}_\mathrm{ref}$, and a reference heading acceleration $\ddot{\psi}_\mathrm{ref}$.

3 Динамическая модель

Рассмотрим квадрокоптер, который моделируется как жесткое тело, контролируемое тягами четырёх двигателей f_i, как изображено на Рис. 2. By changing these four single rotor thrusts, a three axis torque $\boldsymbol{\eta}$ and a mass normalized collective thrust c can be applied on the quadrotor's body. The relation of the single rotor thrusts to the collective thrust and the body torques can be formulated using the coordinate system of Fig. 2 as

$$\boldsymbol{\eta} = \begin{bmatrix} \frac{\sqrt{2}}{2} l (f_1 - f_2 - f_3 + f_4) \\ \frac{\sqrt{2}}{2} l (-f_1 - f_2 + f_3 + f_4) \\ \kappa_1 f_1 - \kappa_2 f_2 + \kappa_3 f_3 - \kappa_4 f_4 \end{bmatrix}, \tag{1}$$

$$mc = f_1 + f_2 + f_3 + f_4, \tag{2}$$

where l is the quadrotor's arm length, $\kappa_i = \kappa(f_i)$ is a coefficient relating the drag torque and the thrust of a single rotor, and m is the quadrotor's mass. Note that unlike in [3] and [4], we consider the rotor drag torque coefficient κ to be a function of the rotor thrust (cf. Fig 4) and *not a constant*.

We model the single rotor thrust f and drag torque τ as quadratic polynomials of the motor input u as

$$f(u) = k_2^f u^2 + k_1^f u + k_0^f, \tag{3}$$

$$\tau(u) = k_2^\tau u^2 + k_1^\tau u + k_0^\tau, \tag{4}$$

where the coefficients k_j^f and k_j^τ are identified by running a single motor with a propeller on a load cell and measuring the resulting forces and moments. The motor input u corresponds to the command we can send to our electronic speed controllers in the range $[-1,\ 1]$. We chose to have three coefficients since it approximates the measured values better than modeling the rotor thrust and drag torque with only a quadratic term, as proposed in e.g. [5]. Fig. 3 compares the two methods for fitting the thrust mapping. From (3) and (4), we can compute the rotor drag torque

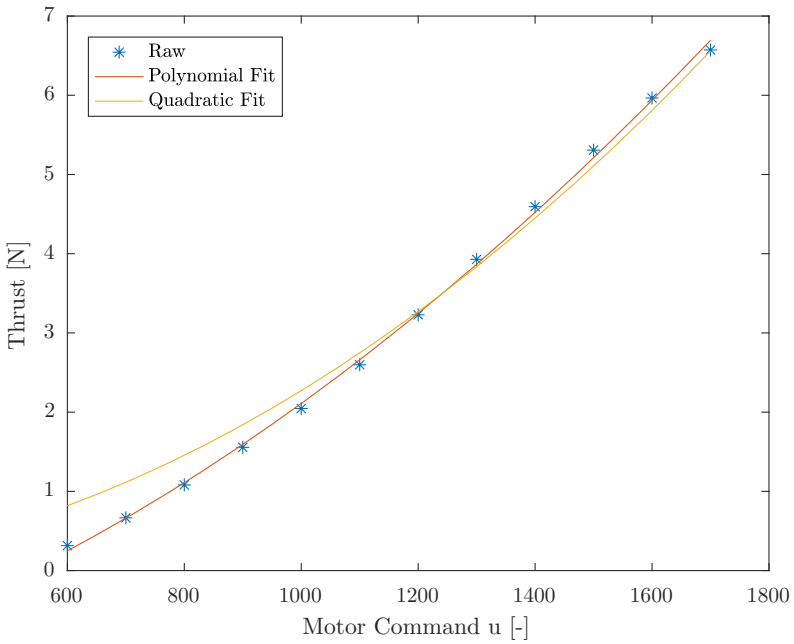

Рис. 3: To approximate the thrust mapping for a single motor with propeller, we fit a second order polynomial into raw thrust measurements obtained by running the motor on a load cell. The polynomial fit approximates the measurements much better than a purely quadratic fit of the form $f(u) = k_2^f u^2$. Data is captured with stiff 6x4.5 inch propellers, Cobra CM-2208/20 2000Kv motors, and DYS XSD20A electronic speed controllers which are commanded through the Dshot300 protocol (Setup shown in Fig. 1).

coefficient as

$$\kappa(f) = \frac{\tau\left(u = \frac{-k_1^f + \sqrt{\left(k_1^f\right)^2 - 4k_2^f\left(k_0^f - f\right)}}{2k_2^f}\right)}{f}. \tag{5}$$

Fig. 4 shows the identified values for the rotor drag torque coefficient and how it varies by about 10 % over the entire range of available motor inputs. We consider the dynamical model of a quadrotor with rotor drag developed in [6] with no wind, stiff propellers, and no dependence of the rotor drag on the thrust. Additionally, as in [2], we model the dynamics of the single rotor thrusts as first order systems

$$\dot{f} = \frac{1}{\alpha_{\text{mot}}}\left(f_{des} - f\right) \tag{6}$$

where the time constant α_{mot} is identified from applying step inputs to a single motor with propeller on a load cell as illustrated in Fig. 5. From these load-cell experiments we found that for modern ESCs that actively break the motor when spinning down, the time constant is almost identical when spinning a motor up or down and we therefore do not need to separate those cases. Because of that, also the bod torques have first order dynamics with the same time constant as the single rotor thrusts. According to these models in [6] and [2], the dynamics of the position $\mathbf{p}$, velocity $\mathbf{v}$,

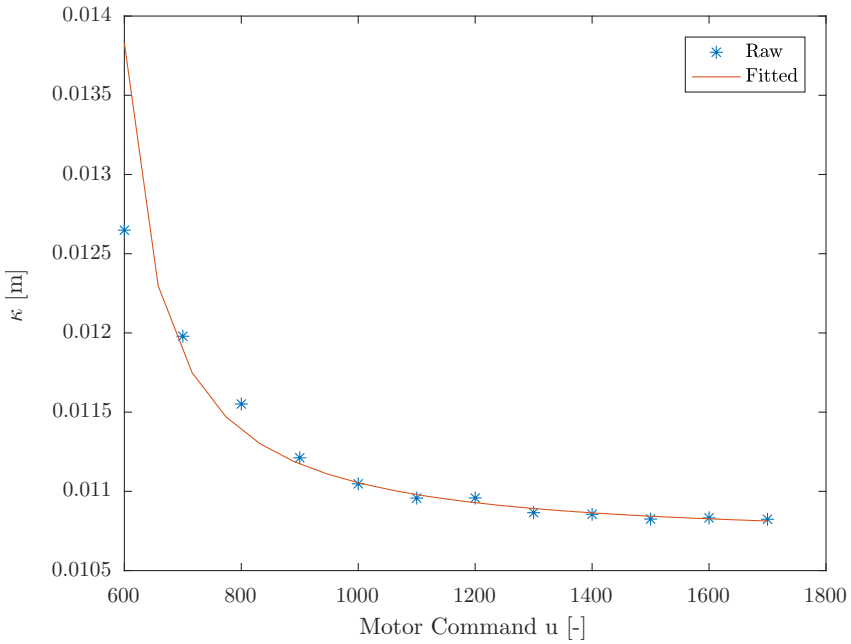

Рис. 4: Values for κ estimated from load cell data obtained by dividing drag torque and thrust values for a motor and propeller. The fitted curve is the ratio of the fitted quadratic functions for measured thrust and drag torque in (3) and (4). Data is captured with stiff 6x4.5 inch propellers, Cobra CM-2208/20 2000Kv motors, and DYS XSD20A electronic speed controllers which are commanded through the Dshot300 protocol (Setup shown in Fig. 1).

orientation $\mathbf{R}$, body rates $\boldsymbol{\omega}$, and body torques $\boldsymbol{\eta}$ can be written as

$$\dot{\mathbf{p}} = \mathbf{v} \tag{7}$$

$$\dot{\mathbf{v}} = -g\mathbf{z}_{\mathrm{W}} + c\mathbf{z}_{\mathrm{B}} - \mathbf{R}\mathbf{D}\mathbf{R}^{\top}\mathbf{v} \tag{8}$$

$$\dot{\mathbf{R}} = \mathbf{R}\hat{\boldsymbol{\omega}} \tag{9}$$

$$\dot{\boldsymbol{\omega}} = \mathbf{J}^{-1}\left(\boldsymbol{\eta} - \boldsymbol{\omega} \times \mathbf{J}\boldsymbol{\omega} - \boldsymbol{\tau}_{\mathrm{g}} - \mathbf{A}\mathbf{R}^{\top}\mathbf{v} - \mathbf{B}\boldsymbol{\omega}\right) \tag{10}$$

$$\dot{\boldsymbol{\eta}} = \frac{1}{\alpha_{\mathrm{mot}}}\left(\boldsymbol{\eta}_{\mathrm{des}} - \boldsymbol{\eta}\right) \tag{11}$$

where c is the mass-normalized collective thrust, $\mathbf{D} = \mathrm{diag}\left(d_{\mathrm{x}}, d_{\mathrm{y}}, d_{\mathrm{z}}\right)$ is a constant diagonal matrix formed by the rotor-drag coefficients, $\hat{\boldsymbol{\omega}}$ is a skew-symmetric matrix formed from $\boldsymbol{\omega}$, $\mathbf{J}$ is the quadrotor's inertia matrix, $\boldsymbol{\eta}$ are the torque inputs, $\boldsymbol{\tau}_{\mathrm{g}}$ are gyroscopic torques from the propellers, and $\mathbf{A}$ and $\mathbf{B}$ are constant matrices of the form

$$\mathbf{A} = \begin{bmatrix} 0 & * & 0 \\ * & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \text{и} \quad \mathbf{B} = \begin{bmatrix} * & 0 & 0 \\ 0 & * & 0 \\ 0 & 0 & * \end{bmatrix}. \tag{12}$$

Выводы и подробности по этим членам можно найти в [6]. В этой модели, мы рассматриваем динамику двигателей for the rotational states но пренебрегаем ей

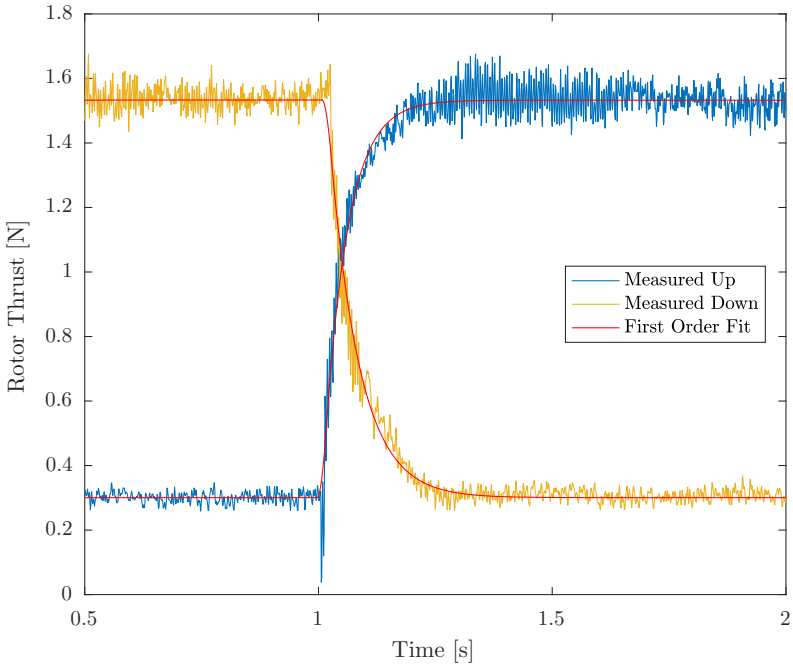

Рис. 5: Thrust transients of a single motor with propeller on a load cell for a step up (blue) and a step down (red). A first order system fit for both is overlaid in red. Data is captured with stiff 6x4.5 inch propellers, Cobra CM-2208/20 2000Kv motors, and DYS XSD20A electronic speed controllers which are commanded through the Dshot300 protocol (Setup shown in Fig. 1).

для полной тяги c. Это сделано, поскольку полная тяга вычисляется в высокоуровневом контроллере позиции, который обычно работает на частоте, сравнимой с динамикой двигателей, и поэтому не имеет достаточного быстродействия для учёта этой динамики в методе управления. В данной работе, мы приняли модель тяги предложенную в [7]

$$c = c_{\text{cmd}} + k_h v_h^2 \tag{13}$$

где c_{cmd} is the commanded collective thrust input, k_h is a constant, and $v_h = \mathbf{v}^\top(\mathbf{x}_\text{B} + \mathbf{y}_\text{B})$. The term $k_h v_h^2$ acts as a quadratic velocity-dependent input disturbance which adds up to the input c_{cmd}. The additional linear velocity-dependent disturbance in the $\mathbf{z}_\text{B}$ direction of the thrust model in [7] is lumped by d_z directly in (8) by neglecting its dependency on the rotor speeds. Заметим, что эта динамическая модель квадрокоптера является обобщением типичной модели, применённой, к примеру, в [8], где компоненты линейного сопротивления ротора опущены, то есть, матрицы $\mathbf{D}$, $\mathbf{A}$ и $\mathbf{B}$ предполагаются нулевыми.

4 Управление

В этой секции описана схема управления квадрокоптером, которая разделена на контроллер верхнего уровня для управления позицией и контроллер нижнего

уровня для управления ориентацией. Контроллер позиции расчитывает требуемую ориентацию, суммарный газ, угловые скорости, и угловые ускорения. Эти значения выдерживаются контроллером положения, который напрямую управляет четырьмя двигателями. Данный контроллер реализован в виде открытого ПО https://github.com/uzh-rpg/rpg_quadrotor_control

4.1 Контроллер верхнего уровня

4.1.1 Управляющие входные сигналы

Для точного следования заданной траектории, контроллеру позиции требуются входные сигналы, которые он использует как опережающие члены. В данном разделе, мы покажем, что требуемые ориентация, тяга, угловые скорости и угловые ускорения могут быть рассчитаны из заданной траектории через свойство дифференциальной плоскости динамики квадрокоптера при учёте сопротивления ротора, как доказано в [1]. Для краткости, опустим индекс *ref* для значений, вычисляемых в этом разделе, поскольку все они являются уставками.

Расчёты этого раздела реализованы в коде

```
control/position_controller/src/position_controller.cpp:200
PositionController::computeAeroCompensatedReferenceInputs(
        reference_state, state_estimate,
        config,
        reference_inputs, //< Result
        drag_accelerations)
```

Во-первых, рассчитаем требуемую ориентацию $\mathbf{R}$, которая определяется по (8) и курс ψ. Из (8), мы можем вывести ограничения

$$\mathbf{x}_\mathrm{B}^\top \boldsymbol{\alpha} = 0, \text{ где } \quad \boldsymbol{\alpha} = \mathbf{a} + g\mathbf{z}_\mathrm{W} + d_\mathrm{x}\mathbf{v} \tag{14}$$

$$\mathbf{y}_\mathrm{B}^\top \boldsymbol{\beta} = 0, \text{ где } \quad \boldsymbol{\beta} = \mathbf{a} + g\mathbf{z}_\mathrm{W} + d_\mathrm{y}\mathbf{v}. \tag{15}$$

To enforce a reference heading ψ, we furthermore constrain the projection of the $\mathbf{x}_\mathrm{B}$ axis into the $\mathbf{x}_\mathrm{W} - \mathbf{y}_\mathrm{W}$ plane to be collinear with $\mathbf{x}_\mathrm{C}$ (cf. Fig.2), where

$$\mathbf{x}_\mathrm{C} = \begin{bmatrix}\cos(\psi) & \sin(\psi) & 0\end{bmatrix}^\top \tag{16}$$

$$\mathbf{y}_\mathrm{C} = \begin{bmatrix}-\sin(\psi) & \cos(\psi) & 0\end{bmatrix}^\top. \tag{17}$$

Из этого, (14) и (15), и ограничения, что $\mathbf{x}_\mathrm{B}$, $\mathbf{y}_\mathrm{B}$, и $\mathbf{z}_\mathrm{B}$ должны быть перпендикулярны друг другу и иметь единичную длину, мы можем построить $\mathbf{R}$ из

$$\mathbf{x}_\mathrm{B} = \frac{\mathbf{y}_\mathrm{C} \times \boldsymbol{\alpha}}{\|\mathbf{y}_\mathrm{C} \times \boldsymbol{\alpha}\|} \tag{18}$$

$$\mathbf{y}_\mathrm{B} = \frac{\boldsymbol{\beta} \times \mathbf{x}_\mathrm{B}}{\|\boldsymbol{\beta} \times \mathbf{x}_\mathrm{B}\|} \tag{19}$$

$$\mathbf{z}_\mathrm{B} = \mathbf{x}_\mathrm{B} \times \mathbf{y}_\mathrm{B} \tag{20}$$

как

$$\mathbf{R} = \begin{bmatrix}\mathbf{x}_\mathrm{B} & \mathbf{y}_\mathrm{B} & \mathbf{z}_\mathrm{B}\end{bmatrix}. \tag{21}$$

Можно проверить, что эти вектора имеют единичную длину, перпендикулярны друг другу, и удовлетворяют ограничениям (14) - (17). Если $\mathbf{y}_\mathrm{C}$ is collinear to $\boldsymbol{\alpha}$ or $\boldsymbol{\beta}$ is collinear to $\mathbf{x}_\mathrm{B}$, the computation of the orientation is affected by a singularity which we tackle as described in Приложении A.

The collective reference thrust can then be computed as

$$c = \mathbf{z}_\mathrm{B}^\top \left(\mathbf{a} + g\mathbf{z}_\mathrm{W} + d_\mathrm{z}\mathbf{v}\right). \tag{22}$$

Тогда, полная тяга может быть рассчитана как функция c, $\mathbf{R}$, и the flat outputs, как

$$c_\mathrm{cmd} = c - k_h(\mathbf{v}^\top(\mathbf{x}_\mathrm{B} + \mathbf{y}_\mathrm{B}))^2. \tag{23}$$

By following [1], мы можем расчитать желаемые скорости путём решения следующей системы линейных уравнений

$$\begin{aligned}
&\omega_y \left(c - \left(d_\mathrm{z} - d_\mathrm{x}\right)\left(\mathbf{z}_\mathrm{B}^\top\mathbf{v}\right)\right) - \omega_z \left(d_\mathrm{x} - d_\mathrm{y}\right)\left(\mathbf{y}_\mathrm{B}^\top\mathbf{v}\right) \\
&\quad = \mathbf{x}_\mathrm{B}^\top\mathbf{j} + d_\mathrm{x}\mathbf{x}_\mathrm{B}^\top\mathbf{a} && (24)\\
&\omega_x \left(c + \left(d_\mathrm{y} - d_\mathrm{z}\right)\left(\mathbf{z}_\mathrm{B}^\top\mathbf{v}\right)\right) + \omega_z \left(d_\mathrm{x} - d_\mathrm{y}\right)\left(\mathbf{x}_\mathrm{B}^\top\mathbf{v}\right) \\
&\quad = -\mathbf{y}_\mathrm{B}^\top\mathbf{j} - d_\mathrm{y}\mathbf{y}_\mathrm{B}^\top\mathbf{a} && (25)\\
&\omega_z = \frac{1}{\|\mathbf{y}_\mathrm{C} \times \mathbf{z}_\mathrm{B}\|}\left(\dot{\psi}\mathbf{x}_\mathrm{C}^\top\mathbf{x}_\mathrm{B} + \omega_y\mathbf{y}_\mathrm{C}^\top\mathbf{z}_\mathrm{B}\right) && (26)
\end{aligned}$$

for ω_x, ω_y, and ω_z as

$$\omega_x = \frac{-\mathcal{B}_1\mathcal{C}_2\mathcal{D}_3 + \mathcal{B}_1\mathcal{C}_3\mathcal{D}_2 - \mathcal{B}_3\mathcal{C}_1\mathcal{D}_2 + \mathcal{B}_3\mathcal{C}_2\mathcal{D}_1}{\mathcal{A}_2\left(\mathcal{B}_1\mathcal{C}_3 - \mathcal{B}_3\mathcal{C}_1\right)} \tag{27}$$

$$\omega_y = \frac{-\mathcal{C}_1\mathcal{D}_3 + \mathcal{C}_3\mathcal{D}_1}{\mathcal{B}_1\mathcal{C}_3 - \mathcal{B}_3\mathcal{C}_1} \tag{28}$$

$$\omega_z = \frac{\mathcal{B}_1\mathcal{D}_3 - \mathcal{B}_3\mathcal{D}_1}{\mathcal{B}_1\mathcal{C}_3 - \mathcal{B}_3\mathcal{C}_1} \tag{29}$$

где

$$\begin{aligned}
\mathcal{B}_1 &= c - \left(d_\mathrm{z} - d_\mathrm{x}\right)\left(\mathbf{z}_\mathrm{B}^\top\mathbf{v}\right) && (30)\\
\mathcal{C}_1 &= -\left(d_\mathrm{x} - d_\mathrm{y}\right)\left(\mathbf{y}_\mathrm{B}^\top\mathbf{v}\right) && (31)\\
\mathcal{D}_1 &= \mathbf{x}_\mathrm{B}^\top\mathbf{j} + d_\mathrm{x}\mathbf{x}_\mathrm{B}^\top\mathbf{a} && (32)\\
\mathcal{A}_2 &= c + \left(d_\mathrm{y} - d_\mathrm{z}\right)\left(\mathbf{z}_\mathrm{B}^\top\mathbf{v}\right) && (33)\\
\mathcal{C}_2 &= \left(d_\mathrm{x} - d_\mathrm{y}\right)\left(\mathbf{x}_\mathrm{B}^\top\mathbf{v}\right) && (34)\\
\mathcal{D}_2 &= -\mathbf{y}_\mathrm{B}^\top\mathbf{j} - d_\mathrm{y}\mathbf{y}_\mathrm{B}^\top\mathbf{a} && (35)\\
\mathcal{B}_3 &= -\mathbf{y}_\mathrm{C}^\top\mathbf{z}_\mathrm{B} && (36)\\
\mathcal{C}_3 &= \|\mathbf{y}_\mathrm{C} \times \mathbf{z}_\mathrm{B}\| && (37)\\
\mathcal{D}_3 &= \dot{\psi}\mathbf{x}_\mathrm{C}^\top\mathbf{x}_\mathrm{B}. && (38)
\end{aligned}$$

Для расчёта требуемых угловых ускорений, мы берём производную системы

линейных уравнений (24)-(26) и решаем её для $\dot{\omega}_x$, $\dot{\omega}_y$, и $\dot{\omega}_z$ как

$$\dot{\omega}_x = \frac{-\mathcal{B}_1\mathcal{C}_2\mathcal{E}_3 + \mathcal{B}_1\mathcal{C}_3\mathcal{E}_2 - \mathcal{B}_3\mathcal{C}_1\mathcal{E}_2 + \mathcal{B}_3\mathcal{C}_2\mathcal{E}_1}{\mathcal{A}_2\left(\mathcal{B}_1\mathcal{C}_3 - \mathcal{B}_3\mathcal{C}_1\right)} \tag{39}$$

$$\dot{\omega}_y = \frac{-\mathcal{C}_1\mathcal{E}_3 + \mathcal{C}_3\mathcal{E}_1}{\mathcal{B}_1\mathcal{C}_3 - \mathcal{B}_3\mathcal{C}_1} \tag{40}$$

$$\dot{\omega}_z = \frac{\mathcal{B}_1\mathcal{E}_3 - \mathcal{B}_3\mathcal{E}_1}{\mathcal{B}_1\mathcal{C}_3 - \mathcal{B}_3\mathcal{C}_1} \tag{41}$$

где

$$\mathcal{E}_1 = \mathbf{x}_{\mathrm{B}}^{\top}\mathbf{s} - 2\dot{c}\omega_y - c\omega_x\omega_z + \mathbf{x}_{\mathrm{B}}^{\top}\boldsymbol{\xi} \tag{42}$$

$$\mathcal{E}_2 = -\mathbf{y}_{\mathrm{B}}^{\top}\mathbf{s} - 2\dot{c}\omega_x + c\omega_y\omega_z - \mathbf{y}_{\mathrm{B}}^{\top}\boldsymbol{\xi} \tag{43}$$

$$\begin{aligned}\mathcal{E}_3 = {} & \ddot{\psi}\mathbf{x}_{\mathrm{C}}^{\top}\mathbf{x}_{\mathrm{B}} + 2\dot{\psi}\omega_z\mathbf{x}_{\mathrm{C}}^{\top}\mathbf{y}_{\mathrm{B}} - 2\dot{\psi}\omega_y\mathbf{x}_{\mathrm{C}}^{\top}\mathbf{z}_{\mathrm{B}} \\ & - \omega_x\omega_y\mathbf{y}_{\mathrm{C}}^{\top}\mathbf{y}_{\mathrm{B}} - \omega_x\omega_z\mathbf{y}_{\mathrm{C}}^{\top}\mathbf{z}_{\mathrm{B}}\end{aligned} \tag{44}$$

$$\begin{aligned}\dot{c} = {} & \mathbf{z}_{\mathrm{B}}^{\top}\mathbf{j} + \omega_x\left(d_{\mathrm{y}} - d_{\mathrm{z}}\right)\left(\mathbf{y}_{\mathrm{B}}^{\top}\mathbf{v}\right) \\ & + \omega_y\left(d_{\mathrm{z}} - d_{\mathrm{x}}\right)\left(\mathbf{x}_{\mathrm{B}}^{\top}\mathbf{v}\right) + d_{\mathrm{z}}\mathbf{z}_{\mathrm{B}}^{\top}\mathbf{a}\end{aligned} \tag{45}$$

$$\begin{aligned}\boldsymbol{\xi} = {} & \mathbf{R}(\hat{\boldsymbol{\omega}}^2\mathbf{D} + \mathbf{D}\hat{\boldsymbol{\omega}}^2 + 2\hat{\boldsymbol{\omega}}\mathbf{D}\hat{\boldsymbol{\omega}}^{\top})\,\mathbf{R}^{\top}\mathbf{v} \\ & + 2\mathbf{R}(\hat{\boldsymbol{\omega}}\mathbf{D} + \mathbf{D}\hat{\boldsymbol{\omega}}^{\top})\,\mathbf{R}^{\top}\mathbf{a} + \mathbf{R}\mathbf{D}\mathbf{R}^{\top}\mathbf{j}.\end{aligned} \tag{46}$$

Расчёт угловых скоростей и угловых ускорений может приводить к сингулярным значениям — эта проблема рассмотрена в Приложении А. Подробнее об этих выводах можно прочитать в техническом докладе [9].

4.1.2 Высокоуровневый закон управления

В нашем контроллере позиции высокого уровня, адаптированном из [10], мы рассчитываем желаемую ориентацию $\mathbf{R}_{\mathrm{des}}$, суммарную тягу c_{cmd}, угловые скорости $\boldsymbol{\omega}_{\mathrm{des}}$, угловые ускорения $\dot{\boldsymbol{\omega}}_{\mathrm{des}}$, и угловой jerk $\ddot{\boldsymbol{\omega}}_{\mathrm{des}}$, которые затем выдерживаются контроллером нижнего уровня описанным далее. На первом шаге в контроллере позиции, мы вычисляем желаемое ускорение квадрокоптера как

$$\mathbf{a}_{\mathrm{des}} = \mathbf{a}_{\mathrm{fb}} + \mathbf{a}_{\mathrm{ref}} - \mathbf{a}_{\mathrm{rd}} + g\mathbf{z}_{\mathrm{W}} \tag{47}$$

где $\mathbf{a}_{\mathrm{fb}}$ — выход пропорционально-дифференциального контроллера[1], обрабатывающего невязки позиции и скорости как

$$\mathbf{a}_{\mathrm{fb}} = -\mathbf{K}_{\mathrm{pos}}\left(\mathbf{p} - \mathbf{p}_{\mathrm{ref}}\right) - \mathbf{K}_{\mathrm{vel}}\left(\mathbf{v} - \mathbf{v}_{\mathrm{ref}}\right) \tag{48}$$

где $\mathbf{K}_{\mathrm{pos}}$ и $\mathbf{K}_{\mathrm{vel}}$ являются постоянными диагональными матрицами. Заданное ускорение $\mathbf{a}_{\mathrm{ref}}$ берётся прямо из траектории, а ускорение из-за торможения ротора $\mathbf{a}_{\mathrm{rd}}$ рассчитывается[2] в соответствии с (8) как

$$\mathbf{a}_{\mathrm{rd}} = -\mathbf{R}_{\mathrm{ref}}\mathbf{D}\mathbf{R}_{\mathrm{ref}}^{\top}\mathbf{v}_{\mathrm{ref}}. \tag{49}$$

[1] PositionController::computePIDErrorAcc()

[2] PositionController::computeAeroCompensatedReferenceInputs()

Желаемая ориентация $\mathbf{R}_{\text{des}}$ вычисляется так, что $\mathbf{z}_{\text{B,des}} = \mathbf{a}_{\text{des}} / \|\mathbf{a}_{\text{des}}\|$ и с учётом заданного курса ψ_{ref}. Эти условия выполняются при

$$\mathbf{z}_{\text{B,des}} = \frac{\mathbf{a}_{\text{des}}}{\|\mathbf{a}_{\text{des}}\|} \tag{50}$$

$$\mathbf{x}_{\text{B,des}} = \frac{\mathbf{y}_{\text{C}} \times \mathbf{z}_{\text{B,des}}}{\|\mathbf{y}_{\text{C}} \times \mathbf{z}_{\text{B,des}}\|} \tag{51}$$

$$\mathbf{y}_{\text{B,des}} = \mathbf{z}_{\text{B,des}} \times \mathbf{x}_{\text{B,des}} \tag{52}$$

как

$$\mathbf{R}_{\text{des}} = \begin{bmatrix} \mathbf{x}_{\text{B,des}} & \mathbf{y}_{\text{B,des}} & \mathbf{z}_{\text{B,des}} \end{bmatrix}. \tag{53}$$

Проецируя желательные ускорения на ось z квадрокоптера и рассматривая модель тяги (13), мы можем вычислить уставку суммарной тяги как

$$c_{\text{cmd}} = \mathbf{a}_{\text{des}}^{\top} \mathbf{z}_{\text{B}} - k_h (\mathbf{v}^{\top} (\mathbf{x}_{\text{B}} + \mathbf{y}_{\text{B}}))^2. \tag{54}$$

Наконец, мы получим желательные угловые скорости, угловые ускорения, и угловой jerk, преобразовав заданные угловые скорости в локальную систему координат и продифференцировав их как

$$\boldsymbol{\omega}_{\text{des}} = \mathbf{R}^{\top} \mathbf{R}_{\text{ref}} \boldsymbol{\omega}_{\text{ref}} \tag{55}$$

$$\dot{\boldsymbol{\omega}}_{\text{des}} = \mathbf{R}^{\top} \mathbf{R}_{\text{ref}} \dot{\boldsymbol{\omega}}_{\text{ref}} - \hat{\boldsymbol{\omega}} \mathbf{R}^{\top} \mathbf{R}_{\text{ref}} \boldsymbol{\omega}_{\text{ref}} \tag{56}$$

4.2 Низкоуровневый контроллер

Контроллер нижнего уровня предназначен для поддержания требуемых ориентации, угловых скоростей и угловых ускорений, расчитанных контроллером верхнего уровня. This can be achieved by controllers such as presented in [11] or [8] but they are often not practical due to the unavailability of an attitude estimate in the low-level control loop. Therefore, we split the low-level controller into an attitude part that runs with the high-level controller and a separate body-rate controller which does not require an attitude estimate. В этом разделе описывается низкоуровневое управление, которое реализовано в rpg_rotors_interface, за исключением контроллера ориентации, который реализован в position_controller.

4.2.1 Управление ориентацией

Контроллер ориентации основан на кватернионном представлении и похож на [12]. Можно показать, что этот метод управления глобально асимптотически устойчив и его цифровая реализация устойчива к измерительным шумам [12, 13]. Этот контроллер реализован в position_controller.

```
control/position_controller/src/position_controller.cpp:521
PositionController::computeFeedBackControlBodyrates(
        desired_attitude, attitude_estimate, config)
```

Сначала определим ошибку ориентации в виде кватерниона невязки как

$$\mathbf{q}_{\mathrm{e}} = \mathbf{q}^{-1} \otimes \mathbf{q}_{\mathrm{des}}. \tag{57}$$

Из элементов кватерниона невязки можно напрямую рассчитать ответные угловые скорости как

$$\boldsymbol{\omega}_{\mathrm{fb}} = \begin{cases} 2 \cdot \mathbf{K}_{\mathrm{att}} \cdot \mathbf{q}_{\mathrm{e}} & \text{if } \mathbf{q}_{\mathrm{e,w}} \geq 0 \\ -2 \cdot \mathbf{K}_{\mathrm{att}} \cdot \mathbf{q}_{\mathrm{e}} & \text{if } \mathbf{q}_{\mathrm{e,w}} < 0 \end{cases} \tag{58}$$

где кватернион невязки $\mathbf{q}_{\mathrm{e}}$ определён в нотации (126) и $\mathbf{q}_{\mathrm{e,w}}$ — его вещественная часть, и

$$\mathbf{K}_{\mathrm{att}} = \begin{bmatrix} 0 & k_{rp} & 0 & 0 \\ 0 & 0 & k_{rp} & 0 \\ 0 & 0 & 0 & k_y \end{bmatrix} \tag{59}$$

где k_{rp} и k_y — коэффициенты контроллера ориентации.

4.2.2 Управление угловыми скоростями

In this section, we present a body-rate controller that provides good tracking and disturbance rejection performance by considering the dynamics of the body rates and body torques (10) and (11) as suggested in our work [2]. We achieve this by designing an LQR controller for a dynamical system containing the body rates and body torques as state. The inputs to this controller are the desired body rates $\boldsymbol{\omega}_{\mathrm{des}}$, the feedback body rates $\boldsymbol{\omega}_{\mathrm{fb}}$, and the desired mass normalized collective thrust c_{cmd}, which are given from the high-level position and attitude controller.

Линеаризуя (10) и (11) при $\boldsymbol{\omega} = \mathbf{0}$ и $\boldsymbol{\eta} = \mathbf{0}$ получаем систему

$$\begin{bmatrix} \dot{\boldsymbol{\omega}} \\ \dot{\boldsymbol{\eta}} \end{bmatrix} = \begin{bmatrix} \mathbf{0} & \mathbf{J}^{-1} \\ \mathbf{0} & -\frac{1}{\alpha_{\mathrm{mot}}}\mathbf{I}_3 \end{bmatrix} \begin{bmatrix} \boldsymbol{\omega} \\ \boldsymbol{\eta} \end{bmatrix} + \begin{bmatrix} \mathbf{0} \\ \frac{1}{\alpha_{\mathrm{mot}}}\mathbf{I}_3 \end{bmatrix} \boldsymbol{\eta}_{\mathrm{des}} \tag{60}$$

which we can use to design an infinite-horizon LQR control law $\mathbf{u} = -\mathbf{K}_{lqr}\mathbf{s}$ that minimizes the cost function

$$\int \mathbf{s}^\top \mathbf{Q}\mathbf{s} + \mathbf{u}^\top \mathbf{R}\mathbf{u}\, dt \tag{61}$$

где $\mathbf{Q}$ — диагональная матрица весов и $\mathbf{R}$ — единичная матрица. The solution to the formulated LQR problem is a gain matrix of the form

$$\mathbf{K}_{lqr} = \begin{bmatrix} k_{\omega_{xy}} & 0 & 0 & k_{\eta_{xy}} & 0 & 0 \\ 0 & k_{\omega_{xy}} & 0 & 0 & k_{\eta_{xy}} & 0 \\ 0 & 0 & k_{\omega_z} & 0 & 0 & k_{\eta_z} \end{bmatrix} \tag{62}$$

которая соответствует пропорционально-дифференциальному контроллеру угловых скоростей. Additionally, мы добавляем feed forward terms такие, что $\boldsymbol{\omega}_{\mathrm{des}}$ достигаается при $\dot{\boldsymbol{\omega}} = \dot{\boldsymbol{\omega}}_{\mathrm{des}}$, resulting in the control policy

$$\boldsymbol{\eta}_{\mathrm{des}} = \mathbf{K}_{lqr} \begin{bmatrix} \boldsymbol{\omega}_{\mathrm{des}} + \boldsymbol{\omega}_{\mathrm{fb}} - \boldsymbol{\omega} \\ \boldsymbol{\eta}_{\mathrm{ref}} - \boldsymbol{\eta} \end{bmatrix} + \boldsymbol{\omega} \times \mathbf{J}\boldsymbol{\omega} + \mathbf{J}\dot{\boldsymbol{\omega}}_{\mathrm{des}}, \tag{63}$$

with $\boldsymbol{\eta}_{\text{ref}} = \boldsymbol{\omega}_{\text{des}} \times \mathbf{J}\boldsymbol{\omega}_{\text{des}} + \mathbf{J}\dot{\boldsymbol{\omega}}_{\text{des}}$ computed from (10). The vector $\boldsymbol{\omega}$ are the estimated body rates measured by the onboard gyroscopes, and $\boldsymbol{\eta}$ are the estimated body torques obtained by estimating the single rotor thrusts with (6) and using (1) to transform them into body torques. Also note that this estimation can be improved if feedback of the rotor speeds is available. In the controller (63), the term $\boldsymbol{\omega} \times \mathbf{J}\boldsymbol{\omega}$ provides feedback linearization, compensating for the coupling terms in the body-rate dynamics, and $\mathbf{J}\dot{\boldsymbol{\omega}}_{\text{des}}$ is a feed-forward term on desired angular accelerations.

4.2.3 Итеративный подбор тяги

Для расчёта тяги каждого двигателяTo compute the single rotor thrusts that achieve the desired body torques $\boldsymbol{\eta}_{\text{des}}$ and collective thrust c_{cmd}, which we denote as *mixer inputs*, необходимо решить (1) и (2) для f_i. Поскольку мы рассматриваем коэффициенты крутящего момента ротора как be a function of the rotor thrust, we cannot solve this system of equations directly, but we can do so iteratively, as proposed in our previous work [2].

В начале, мы задаём тяги двигателей равными, так, чтобы в сумме они выдавали требуемую тягу:

$$f_i = \frac{m c_{\text{cmd}}}{4}. \tag{64}$$

Заметим, что эти значения используются только для расчёта коэффициентов крутящего момента роторов в первой итерации. Затем, мы производим несколько итераций, состоящих из двух шагов: i) решить (5) для получения κ_i, и ii) решить (1) и (2) с $\boldsymbol{\eta}_{\text{des}}$ и c_{cmd} для получения f_i.

4.2.4 Насыщение с различными приоритетами входных сигналов

Once we have computed the desired single rotor thrusts, we have to make sure that they lie within the feasible range $[f_{min}, f_{max}]$ for each single motor, which we do as proposed in our previous work [2]. Naively, feasible inputs can be achieved by clipping each rotor thrust if its desired value is outside this range. This is a simple and fast procedure with the drawback that none of the desired mixer inputs $\boldsymbol{\eta}_{\text{des}}$ and c_{cmd} is achieved exactly if one of the rotor thrusts is clipped. Nonetheless, not all these mixer inputs are equally important in terms of the quadrotor's ability to stabilize and track a trajectory. Since a quadrotor can only produce a collective thrust in its body upwards direction, it has to be aligned with the desired acceleration for following a trajectory in 3D space. The rotation around the thrust direction is irrelevant for the translational motion of the quadrotor. Therefore, we want to give least priority to achieve the desired yaw torque in case of an input saturation. On the other hand, the quadrotor uses roll and pitch torques to change its thrust direction which enables stabilization and therefore makes them the most important inputs. Furthermore, state of the art control methods for quadrotors (e.g. [8], [3], [10]) are based on the assumption that the orientation of the thrust vector can be changed quickly. For these reasons, in case of an input saturation, we want to give highest priority to applying the desired roll and pitch torques, second highest priority to applying the desired collective thrust, and lowest priority to applying the desired yaw torque.

Algorithm 1 Rotor Thrust Saturation

Compute f_i as detailed in Section 4.2.3
Perform yaw-torque saturation:
if Motor saturated **AND** $|\eta_{z,des}| > \eta_{z,assured}$ **then**
  Find rotor j that violates thrust limits the most
  $f_j \leftarrow f_{limit}$
  Solve (1) and (2) for $f_{i \setminus j}$ and η_z
  if $\text{sign}(\eta_{z,des}) \cdot \eta_z < \eta_{z,assured}$ **then**
    $\eta_z \leftarrow \text{sign}(\eta_{z,des}) \cdot \eta_{z,assured}$
    Solve (1) and (2) for f_i
  end if
end if
Perform collective-thrust saturation:
if Motor saturated **then**
  if NOT(upper **AND** lower saturation reached) **then**
    Find rotor j that violates thrust limits the most
    Shift f_i equally s.t. $f_j = f_{limit}$
  end if
end if
Enforce single rotor thrust limits by thrust clipping

Yaw-Torque Saturation We achieve this prioritization by a saturation scheme as summarized in Algorithm 1. First, the single rotor thrusts are computed according to Section 4.2.3. If one of the single rotor thrusts exceeds its limits, we try to change the applied yaw torque to avoid saturation, given that the desired yaw torque is above a certain minimum $\eta_{z,assured}$, which we can optionally impose. Such an assured yaw torque might be desired for applications where we want to guarantee that we always have some control on the heading of a quadrotor. In case of saturation, we do not apply the iterative mixer in order to save time since we are unable to apply the desired yaw torque anyways. To do the yaw-torque saturation, we find the rotor that violates the input limit the most, set it to the corresponding limit and then solve (1) and (2) for the remaining rotor thrusts and the yaw torque. If the resulting yaw torque is still above the value we want to assure, we successfully enforced all the rotor-thrust limits by only changing the applied yaw torque. In other words, in this case, Algorithm 1 guarantees that the quadrotor applies the desired roll and pitch torques and the desired collective thrust, but *not* the desired yaw torque. If the resulting yaw torque is below the value we want to assure, we set it to the assured value $\eta_{z,assured}$ and recompute the rotor thrusts.

Collective-Thrust Saturation If one of the rotor-thrust limits is still violated, we try to change the applied collective thrust to avoid saturation. This is only possible if two rotors do not violate the upper and the lower limit simultaneously, in which case it is impossible to achieve the desired roll and pitch torques by changing the applied collective thrust. If only one limit is violated, we find the rotor that violates the input limit the most, set it to its limit and shift the remaining rotor

thrusts by the same amount. In this case, Algorithm 1 ensures that the quadrotor applies the desired roll and pitch torques but *not* the desired collective thrust and *not* the desired yaw torque.

Thrust Clipping At this point, if a rotor still violates its input limits, we have to apply thrust clipping and can therefore not achieve any of the desired mixer inputs precisely.

4.3 Компенсация задержки

TODO: about state predictor

4.4 Компенсация изменения напряжения батареи

Since we use electronic speed controllers that control the percentage of input voltage, the resulting rotor thrust for a given command depends on the battery voltage. To improve the trajectory tracking performance of our quadrotors, we compensate for this voltage dependency. By performing thrust mapping identifications with different voltages which are controlled by a power supply unit, in Fig. 6, we found that they are related by a constant factor to each other. Furthermore, by letting a quadrotor hover for an entire battery charge while applying the thrust mapping identified at 12 V, we found that this factor, which depicts the ratio of the commanded thrust to the actually produced thrust, is a linear function of the battery voltage (see Fig. 7). The collective thrust command that compensates for the varying battery voltage can therefore be obtained as

$$c_{\text{cmd,comp}} = (k_1^v v_{bat} + k_0^v)\, c_{\text{cmd}} \tag{65}$$

where v_{bat} is the measured battery voltage and k_1^v and k_0^v are the identified coefficients of the linear function describing the voltage dependent thrust ratio. Compared to [14], our method only needs to measure the battery voltage not the state of charge of the battery. Furthermore, since other influences on the thrust, such as the air density, are also multiplicative, we can identify all of them lumped together with this procedure. The benefits of this procedure are illustrated in Fig. 8.

5 Расчёт траектории

5.1 Расчёт соответствующих максимумов из заданного состояния

Часто, при разработке траектории, требуется соблюдать определённые ограничения, которые должны быть рассчитаны из требуемого состояния. В этом разделе показано как вычислить наиболее важные для квадрокоптеров значения, а именно: скорость, коллективную тягу, а также норму скоростей вращения по крену и тангажу (особенно важно для квадрокоптеров с визуальной навигацией) как это реализовано в библиотеке polynomial_trajectories. Мы предполагаем, что из выбранной точки на заданной траектории, можно получить

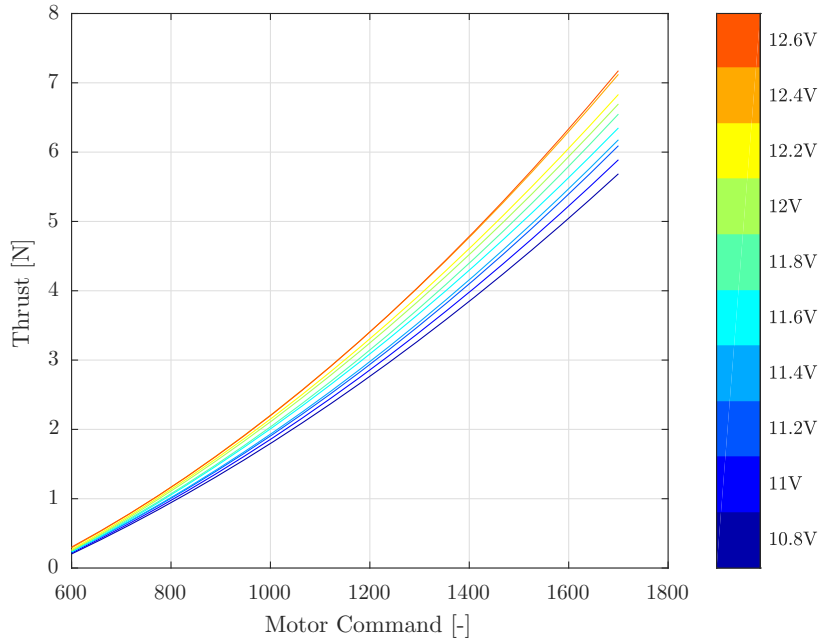

Рис. 6: Second order polynomial thrust mapping (c.f. (3)) for different voltages between 10.8 V and 12.6 V identified on an ATI Mini40 load cell.

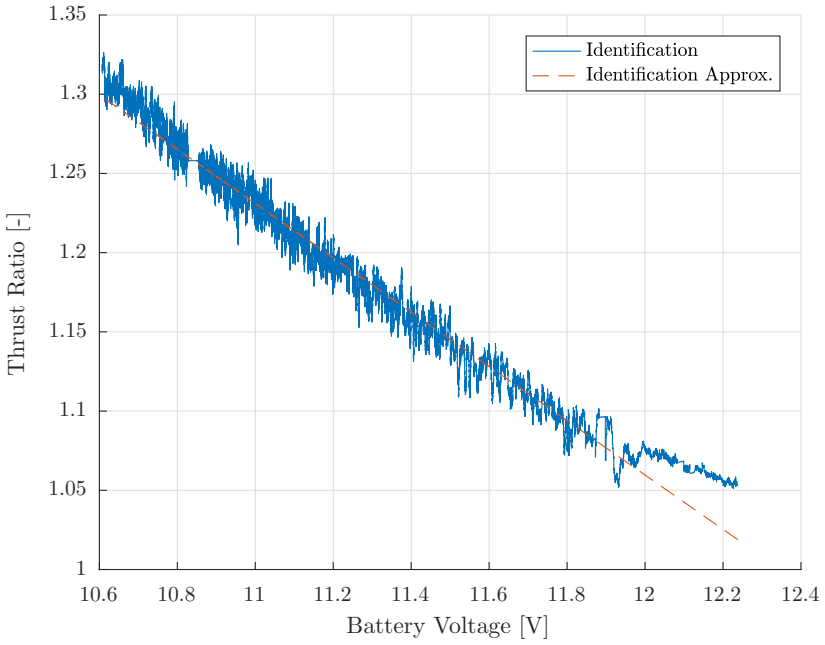

Рис. 7: Оценка коэффициента корреции тяги.

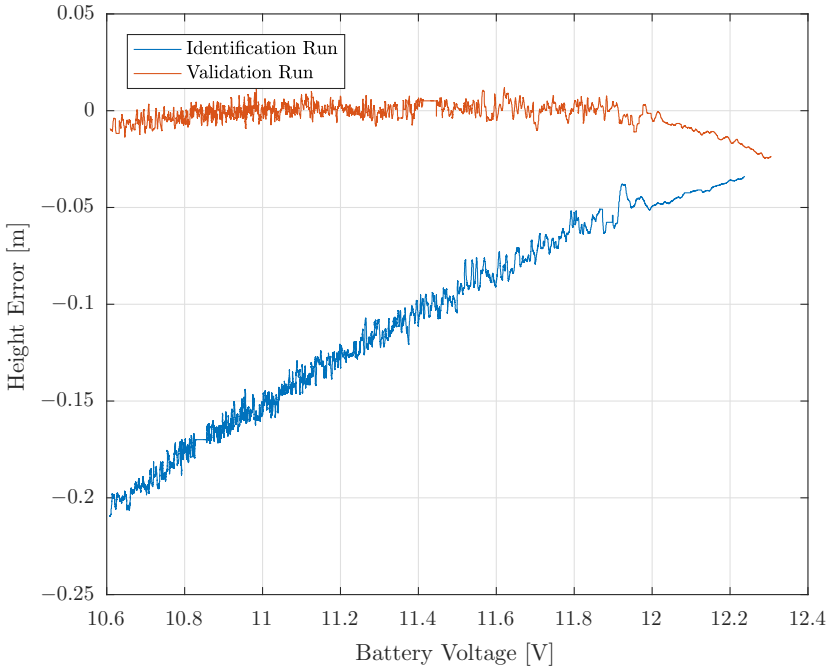

Рис. 8: Проверка компенсации напряжения.

требуемое состояние $\mathbf{s}_{des}$, содержащее позицию, скорость, ускорение, и jerk.

$$\mathbf{s}_{des} = \begin{bmatrix} \mathbf{R}_{\text{des}} & \mathbf{v}_{des} & \mathbf{a}_{des} & \mathbf{j}_{des} \end{bmatrix} \tag{66}$$

Скорость и полная тяга

Скорость может быть легко получена через взятие нормы заданной скорости

$$v = \|\mathbf{v}_{des}\|, \tag{67}$$

и полная тяга может быть рассчитана как

$$c = \|\mathbf{a}_{des} - \mathbf{g}\|, \tag{68}$$

где $\mathbf{g} = [0\ 0\ -g]^\top$.

Норма скорости вращения по крену и тангажу

Введём вектор желаемых ускорений в глобальных координатах как

$$\mathbf{f} = \mathbf{a}_{des} - \mathbf{g} = \mathbf{R}_{\text{WB}} \cdot \begin{bmatrix} 0 \\ 0 \\ c \end{bmatrix} = \mathbf{R}_{\text{WB}} \cdot \mathbf{c}, \tag{69}$$

где $\mathbf{g}$ определено как выше. Заметим, что $\|\mathbf{f}\| = c$, поскольку вращение вектора не изменяет его длину. Следовательно, можно записать

$$\mathbf{R}_{\mathrm{WB}} \cdot \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = \frac{\mathbf{f}}{\|\mathbf{f}\|} = \bar{\mathbf{f}}. \tag{70}$$

Взятие производных приводит к

$$\mathbf{R}_{\mathrm{WB}} \cdot \hat{\boldsymbol{\omega}} \cdot \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = \dot{\bar{\mathbf{f}}}, \tag{71}$$

$$\hat{\boldsymbol{\omega}} \cdot \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = \mathbf{R}_{\mathrm{WB}}^{\top} \cdot \dot{\bar{\mathbf{f}}}, \tag{72}$$

$$\begin{bmatrix} \omega_y \\ -\omega_x \\ 0 \end{bmatrix} = \mathbf{R}_{\mathrm{WB}}^{\top} \cdot \dot{\bar{\mathbf{f}}}, \tag{73}$$

где $\dot{\bar{\mathbf{f}}}$ рассчитывается как

$$\dot{\bar{\mathbf{f}}} = \frac{d}{dt}\left(\frac{\mathbf{f}}{\|\mathbf{f}\|}\right), \tag{74}$$

$$= \frac{\dot{\mathbf{f}} \cdot \|\mathbf{f}\| - \mathbf{f} \cdot \frac{\mathbf{f}^{\top} \cdot \dot{\mathbf{f}}}{\|\mathbf{f}\|}}{\|\mathbf{f}\|^2}, \tag{75}$$

$$= \frac{\dot{\mathbf{f}}}{\|\mathbf{f}\|} - \frac{\mathbf{f} \cdot \mathbf{f}^{\top} \cdot \dot{\mathbf{f}}}{\|\mathbf{f}\|^3}, \tag{76}$$

$$= \frac{\mathbf{j}}{\|\mathbf{f}\|} - \frac{\mathbf{f} \cdot \mathbf{f}^{\top} \cdot \mathbf{j}}{\|\mathbf{f}\|^3}, \tag{77}$$

учитывая факт, что $\dot{\mathbf{f}} = \mathbf{j}$. With all this we can write

$$\begin{bmatrix} \omega_y \\ -\omega_x \\ 0 \end{bmatrix} = \mathbf{R}_{\mathrm{WB}}^{\top} \cdot \left(\frac{\mathbf{j}}{\|\mathbf{f}\|} - \frac{\mathbf{f} \cdot \mathbf{f}^{\top} \cdot \mathbf{j}}{\|\mathbf{f}\|^3}\right), \tag{78}$$

$$= \mathbf{R}_{\mathrm{WB}}^{\top} \cdot \frac{\mathbf{j}}{c} - \mathbf{R}_{\mathrm{WB}}^{\top} \frac{\mathbf{R}_{\mathrm{WB}} \cdot \mathbf{c} \cdot \mathbf{c}^{\top} \cdot \mathbf{R}_{\mathrm{WB}}^{\top} \cdot \mathbf{j}}{c^3}, \tag{79}$$

$$= \mathbf{R}_{\mathrm{WB}}^{\top} \cdot \frac{\mathbf{j}}{c} - \frac{\mathbf{c} \cdot \mathbf{c}^{\top} \cdot \mathbf{R}_{\mathrm{WB}}^{\top} \cdot \mathbf{j}}{c^3}. \tag{80}$$

Заметим, что

$$\mathbf{c} \cdot \mathbf{c}^{\top} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & c^2 \end{bmatrix}, \tag{81}$$

и используя (118), получим

$$\begin{bmatrix} \omega_y \\ -\omega_x \\ 0 \end{bmatrix} = \begin{bmatrix} (\mathbf{e}_x^B)^\top \cdot \frac{\mathbf{j}}{c} \\ (\mathbf{e}_y^B)^\top \cdot \frac{\mathbf{j}}{c} \\ (\mathbf{e}_z^B)^\top \cdot \frac{\mathbf{j}}{c} \end{bmatrix} - \begin{bmatrix} 0 \\ 0 \\ (\mathbf{e}_z^B)^\top \cdot \frac{\mathbf{j}}{c} \end{bmatrix}. \tag{82}$$

Заметим, что $\left\|\mathbf{R}_{\text{WB}}^\top \cdot \frac{\mathbf{j}}{c}\right\| = \left\|\frac{\mathbf{j}}{c}\right\|$ и что $\mathbf{e}_z^B$ может быть получено из заданного ускорения как

$$\mathbf{e}_z^B = \frac{\mathbf{a}_{des} - \mathbf{g}}{\|\mathbf{a}_{des} - \mathbf{g}\|}. \tag{83}$$

Следовательно, мы можем вычислить норму скоростей вращения по крену и тангажу как

$$\|\omega_{x,y}\| = \left\| \begin{bmatrix} \omega_y \\ -\omega_x \\ 0 \end{bmatrix} \right\| = \sqrt{\left\|\frac{\mathbf{j}}{c}\right\|^2 - \left((\mathbf{e}_z^B)^\top \cdot \frac{\mathbf{j}}{c}\right)^2} \tag{84}$$

6 Структура исходного кода

6.1 Планирование траектории

6.2 Контроллер позиции

7 Математика

7.1 Механика

7.1.1 Euler's first law

Момент импульса жёсткого тела, $\mathbf{p}$ определён как

$$\mathbf{p} := m \cdot \mathbf{v}_S, \tag{85}$$

где m — масса тела, $\mathbf{v}_S$ — скорость центра масс. Euler's first law states that the change of linear momentum is equal to the sum of all external forces $\mathbf{F}$:

$$\dot{\mathbf{p}} = \mathbf{F}, \tag{86}$$

Дифференцируя (85) и вставляя в (86) получаем

$$\dot{\mathbf{p}} = m \cdot \mathbf{a}_S = \mathbf{F}, \tag{87}$$

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

7.1.2 Euler's second law

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

$$\mathbf{L}_{\mathrm{O}} := \mathbf{R}_{\mathrm{OP}} \times \mathbf{p} + \mathbf{I}_{\mathrm{P}} \cdot \boldsymbol{\Omega} + m \cdot \mathbf{R}_{\mathrm{PS}} \times \mathbf{v}_{\mathrm{P}}, \tag{88}$$

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 (88) not for an arbitrary P but at the center of mass $P = S$ we find

$$\mathbf{L}_{\mathrm{O}} = \mathbf{R}_{\mathrm{OS}} \times \mathbf{p} + \mathbf{I}_{\mathrm{S}} \cdot \boldsymbol{\Omega}. \tag{89}$$

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:

$$\dot{\mathbf{L}}_{\mathrm{O}} = \mathbf{M}_{\mathrm{O}}. \tag{90}$$

Differentiating (89) and inserting in (90) gives

$$\dot{\mathbf{L}}_{\mathrm{O}} = \mathbf{R}_{\mathrm{OS}} \times \dot{\mathbf{p}} + \mathbf{I}_{\mathrm{S}} \cdot \boldsymbol{\Psi} + \boldsymbol{\Omega} \times \mathbf{I}_{\mathrm{S}} \cdot \boldsymbol{\Omega} = \mathbf{M}_{\mathrm{O}}, \tag{91}$$

where $\boldsymbol{\Psi}$ is the angular acceleration of the body. We can further rewrite (91) as

$$\begin{aligned} \dot{\mathbf{L}}_{\mathrm{O}} - \mathbf{R}_{\mathrm{OS}} \times \dot{\mathbf{p}} = \mathbf{I}_{\mathrm{S}} \cdot \boldsymbol{\Psi} + \boldsymbol{\Omega} \times \mathbf{I}_{\mathrm{S}} \cdot \boldsymbol{\Omega} &= \mathbf{M}_{\mathrm{O}} - \mathbf{R}_{\mathrm{OS}} \times \dot{\mathbf{p}}, \\ &= \mathbf{M}_{\mathrm{O}} + \mathbf{R}_{\mathrm{SO}} \times \dot{\mathbf{p}}, \\ &= \mathbf{M}_{\mathrm{O}} + \mathbf{R}_{\mathrm{SO}} \times \mathbf{F}, \\ \mathbf{I}_{\mathrm{S}} \cdot \boldsymbol{\Psi} + \boldsymbol{\Omega} \times \mathbf{I}_{\mathrm{S}} \cdot \boldsymbol{\Omega} &= \mathbf{M}_{\mathrm{S}}. \end{aligned} \tag{92}$$

7.1.3 Differentiating the Angular Momentum

This section shows how to differentiate $\mathbf{L}_{\mathrm{O}}$. First, we restate (89) for the definition of the angular momentum

$$\mathbf{L}_{\mathrm{O}} = \mathbf{R}_{\mathrm{OS}} \times \mathbf{p} + \mathbf{I}_{\mathrm{S}} \cdot \boldsymbol{\Omega}. \tag{93}$$

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

$${}_{\mathrm{B}}\dot{\mathbf{c}} = ({}_{\mathrm{B}}\mathbf{c})' + {}_{\mathrm{W}}\boldsymbol{\omega}_{\mathrm{WB}} \times {}_{\mathrm{B}}\mathbf{c}. \tag{94}$$

Note the special notation for $({}_{\mathrm{B}}\mathbf{c})'$ 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 readability we differentiate the two terms in (93) separately:

$$\begin{aligned} \dot{(\mathbf{R}_{\mathrm{OS}} \times \mathbf{p})} &= \dot{\mathbf{R}}_{\mathrm{OS}} \times \mathbf{p} + \mathbf{R}_{\mathrm{OS}} \times \dot{\mathbf{p}}, && (95) \\ &= v_S \times \mathbf{p} + \mathbf{R}_{\mathrm{OS}} \times \dot{\mathbf{p}}, && (96) \\ &= v_S \times m \cdot v_S + \mathbf{R}_{\mathrm{OS}} \times \dot{\mathbf{p}}, && (97) \\ &= \mathbf{R}_{\mathrm{OS}} \times \dot{\mathbf{p}}. && (98) \end{aligned}$$

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

$$
\begin{aligned}
{}_B(\mathbf{I}_S \dot{\cdot} \boldsymbol{\Omega}) &= ({}_B\mathbf{I}_{SB}\boldsymbol{\Omega})' + {}_B\boldsymbol{\Omega} \times {}_B\mathbf{I}_S \cdot {}_B\boldsymbol{\Omega}, && (99)\\
&= \cancelto{0}{({}_B\mathbf{I}_S)'}\,{}_B\boldsymbol{\Omega} + {}_B\mathbf{I}_S\,({}_B\boldsymbol{\Omega})' + {}_B\boldsymbol{\Omega} \times {}_B\mathbf{I}_S \cdot {}_B\boldsymbol{\Omega}, && (100)\\
&= {}_B\mathbf{I}_S\,({}_B\boldsymbol{\Omega})' + {}_B\boldsymbol{\Omega} \times {}_B\mathbf{I}_S \cdot {}_B\boldsymbol{\Omega}. && (101)
\end{aligned}
$$

To rewrite $({}_B\boldsymbol{\Omega})'$ we differentiate the angular velocity of the body using euler's formula (94)

$$
\begin{aligned}
{}_B\dot{\boldsymbol{\Omega}} &= ({}_B\boldsymbol{\Omega})' + {}_W\boldsymbol{\omega}_{WB} \times {}_B\boldsymbol{\Omega}, && (102)\\
&= ({}_B\boldsymbol{\Omega})' + \cancelto{0}{{}_B\boldsymbol{\Omega} \times {}_B\boldsymbol{\Omega}}, && (103)\\
&= ({}_B\boldsymbol{\Omega})'. && (104)
\end{aligned}
$$

Inserting (104) in (101) we can continue with the differentiation of the second term

$${}_B(\mathbf{I}_S \dot{\cdot} \boldsymbol{\Omega}) = {}_B\mathbf{I}_{SB}\dot{\boldsymbol{\Omega}} + {}_B\boldsymbol{\Omega} \times {}_B\mathbf{I}_S \cdot {}_B\boldsymbol{\Omega}. \tag{105}$$

This result is valid in any coordinate system not only the body fixed, therefore we can write (105) as

$$(\mathbf{I}_S \dot{\cdot} \boldsymbol{\Omega}) = \mathbf{I}_S\dot{\boldsymbol{\Omega}} + \boldsymbol{\Omega} \times \mathbf{I}_S \cdot \boldsymbol{\Omega}. \tag{106}$$

Combining the results (98) and (106) we finally find

$$
\begin{aligned}
\dot{\mathbf{L}}_O &= \mathbf{R}_{OS} \times \dot{\mathbf{p}} + \mathbf{I}_S\dot{\boldsymbol{\Omega}} + \boldsymbol{\Omega} \times \mathbf{I}_S \cdot \boldsymbol{\Omega}, && (107)\\
&= \mathbf{R}_{OS} \times \dot{\mathbf{p}} + \mathbf{I}_S\boldsymbol{\Psi} + \boldsymbol{\Omega} \times \mathbf{I}_S \cdot \boldsymbol{\Omega}. && (108)
\end{aligned}
$$

Again this result is valid in any coordinate system.

7.1.4 Summary

Note that we can not only evaluate (87) and (92) in the world frame W but in any rotating coordinate systems A and C.

$$
\begin{aligned}
m \cdot {}_A\mathbf{a}_S &= {}_A\mathbf{F}, && (109)\\
{}_C\mathbf{I}_S \cdot {}_C\boldsymbol{\Psi} + {}_C\boldsymbol{\Omega} \times {}_C\mathbf{I}_S \cdot {}_C\boldsymbol{\Omega} &= {}_C\mathbf{M}_S. && (110)
\end{aligned}
$$

7.2 Представления ориентации

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.

7.2.1 Матрицы вращения и углы Эйлера

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

$$ {}_{\mathrm{W}}\mathbf{c} = \mathbf{R}_{\mathrm{WB}} \cdot {}_{\mathrm{B}}\mathbf{c}. \tag{111} $$

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

$$ {}_{\mathrm{W}}\mathbf{p} = \mathbf{R}_{\mathrm{WB}} \cdot {}_{\mathrm{B}}\mathbf{p} \ + \ {}_{\mathrm{W}}\mathbf{t}_{\mathrm{WB}}. \tag{112} $$

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

$$ \mathbf{R}_{\mathrm{WB}} = \mathbf{R}_{\mathrm{z}}(\psi)\mathbf{R}_{\mathrm{y}}(\theta)\mathbf{R}_{\mathrm{x}}(\phi), \tag{113} $$

where

$$ \mathbf{R}_{\mathrm{z}}(\psi) = \begin{bmatrix} \cos(\psi) & -\sin(\psi) & 0 \\ \sin(\psi) & \cos(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix}, \tag{114} $$

$$ \mathbf{R}_{\mathrm{y}}(\theta) = \begin{bmatrix} \cos(\theta) & 0 & \sin(\theta) \\ 0 & 1 & 0 \\ -\sin(\theta) & 0 & \cos(\theta) \end{bmatrix}, \tag{115} $$

$$ \mathbf{R}_{\mathrm{x}}(\phi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\phi) & -\sin(\phi) \\ 0 & \sin(\phi) & \cos(\phi) \end{bmatrix}. \tag{116} $$

Putting it all together we get

$$ \mathbf{R}_{\mathrm{WB}} = \begin{bmatrix} \mathrm{c}(\psi)\,\mathrm{c}(\theta) & \mathrm{c}(\psi)\,\mathrm{s}(\theta)\,\mathrm{s}(\phi) - \mathrm{s}(\psi)\,\mathrm{c}(\phi) & \mathrm{c}(\psi)\,\mathrm{s}(\theta)\,\mathrm{c}(\phi) + \mathrm{s}(\psi)\,\mathrm{s}(\phi) \\ \mathrm{s}(\psi)\,\mathrm{c}(\theta) & \mathrm{s}(\psi)\,\mathrm{s}(\theta)\,\mathrm{s}(\phi) + \mathrm{c}(\psi)\,\mathrm{c}(\phi) & \mathrm{s}(\psi)\,\mathrm{s}(\theta)\,\mathrm{c}(\phi) - \mathrm{c}(\psi)\,\mathrm{s}(\phi) \\ -\mathrm{s}(\theta) & \mathrm{c}(\theta)\,\mathrm{s}(\phi) & \mathrm{c}(\theta)\,\mathrm{c}(\phi) \end{bmatrix}, \tag{117} $$

$$ = \begin{bmatrix} \mathbf{x}_{\mathrm{B}} & \mathbf{y}_{\mathrm{B}} & \mathbf{z}_{\mathrm{B}} \end{bmatrix}, \tag{118} $$

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. 2. The angular velocity

of coordinate system B with respect to the world frame W is denoted as $\boldsymbol{\omega}_{\text{WB}}$. For the next step, we define the skew symmetric matrix

$$\hat{\boldsymbol{\omega}}_{\text{WB}} = \begin{bmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{bmatrix}. \tag{119}$$

The angular velocity is defined as

$${}_{\text{W}}\hat{\boldsymbol{\omega}}_{\text{WB}} := \dot{\mathbf{R}}_{\text{WB}} \cdot \mathbf{R}_{\text{WB}}^T, \tag{120}$$

or in body frame B

$$\begin{aligned} {}_{\text{B}}\hat{\boldsymbol{\omega}}_{\text{WB}} &= \mathbf{R}_{\text{BW}} \cdot {}_{\text{W}}\hat{\boldsymbol{\omega}}_{\text{WB}} \cdot \mathbf{R}_{\text{BW}}^T, && (121)\\ &= \mathbf{R}_{\text{BW}} \cdot \dot{\mathbf{R}}_{\text{WB}} \cdot \mathbf{R}_{\text{WB}}^T \cdot \mathbf{R}_{\text{BW}}^T, && (122)\\ &= \mathbf{R}_{\text{WB}}^\top \cdot \dot{\mathbf{R}}_{\text{WB}}. && (123) \end{aligned}$$

Evaluating (120) and (123) for $\boldsymbol{\omega}_{\text{WB}}$ we get

$${}_{\text{W}}\boldsymbol{\omega}_{\text{WB}} = \begin{bmatrix} \dot{\phi}\,\text{c}(\theta)\,\text{c}(\psi) - \dot{\theta}\,\text{s}(\psi) \\ \dot{\phi}\,\text{c}(\theta)\,\text{s}(\psi) + \dot{\theta}\,\text{c}(\psi) \\ -\dot{\phi}\,\text{s}(\theta) + \dot{\psi} \end{bmatrix} = \begin{bmatrix} \text{c}(\theta)\,\text{c}(\psi) & -\text{s}(\psi) & 0 \\ \text{c}(\theta)\,\text{s}(\psi) & \text{c}(\psi) & 0 \\ -\text{s}(\theta) & 0 & 1 \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix}, \tag{124}$$

and

$${}_{\text{B}}\boldsymbol{\omega}_{\text{WB}} = \begin{bmatrix} \dot{\phi} - \text{s}(\theta)\dot{\psi} \\ \text{c}(\phi)\dot{\theta} + \text{s}(\phi)\,\text{c}(\theta)\dot{\psi} \\ -\text{s}(\phi)\dot{\theta} + \text{c}(\phi)\,\text{c}(\theta)\dot{\psi} \end{bmatrix} = \begin{bmatrix} 1 & 0 & -\text{s}(\theta) \\ 0 & \text{c}(\phi) & \text{s}(\phi)\,\text{c}(\theta) \\ 0 & -\text{s}(\phi) & \text{c}(\phi)\,\text{c}(\theta) \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix}. \tag{125}$$

7.2.2 Quaternions

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

$$\mathbf{q}_{\text{WB}} = \begin{bmatrix} q_w & q_x & q_y & q_z \end{bmatrix}^\top \tag{126}$$

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 $\mathbf{q}_{\text{WB}}$ is what you get from optitrack. The adjoint, norm, and inverse of the quaternion, $\mathbf{q}$, are

$$\begin{aligned} \bar{\mathbf{q}} &= \begin{bmatrix} q_w & -q_x & -q_y & -q_z \end{bmatrix}^\top, && (127)\\ \|\mathbf{q}\| &= \sqrt{q_w^2 + q_x^2 + q_y^2 + q_z^2}, && (128)\\ \mathbf{q}^{-1} &= \frac{\bar{\mathbf{q}}}{\|\mathbf{q}\|}. && (129) \end{aligned}$$

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

$$\mathbf{q} \otimes \mathbf{p} = \mathbf{Q}(\mathbf{q}) \cdot \mathbf{p} = \bar{\mathbf{Q}}(\mathbf{p}) \cdot \mathbf{q}, \tag{130}$$
$$\mathbf{p} \otimes \mathbf{q} = \mathbf{Q}(\mathbf{p}) \cdot \mathbf{q} = \bar{\mathbf{Q}}(\mathbf{q}) \cdot \mathbf{p}, \tag{131}$$
$$\tag{132}$$

where

$$\mathbf{Q}(\mathbf{q}) = \begin{bmatrix} q_w & -q_x & -q_y & -q_z \\ q_x & q_w & -q_z & q_y \\ q_y & q_z & q_w & -q_x \\ q_z & -q_y & q_x & q_w \end{bmatrix}, \tag{133}$$

$$\bar{\mathbf{Q}}(\mathbf{q}) = \begin{bmatrix} q_w & -q_x & -q_y & -q_z \\ q_x & q_w & q_z & -q_y \\ q_y & -q_z & q_w & q_x \\ q_z & q_y & -q_x & q_w \end{bmatrix}. \tag{134}$$

We find that

$$\mathbf{Q}(\bar{\mathbf{q}}) = \mathbf{Q}(\mathbf{q})^\top, \tag{135}$$
$$\bar{\mathbf{Q}}(\bar{\mathbf{q}}) = \bar{\mathbf{Q}}(\mathbf{q})^\top. \tag{136}$$

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

$$\begin{bmatrix} 0 \\ {}_W\mathbf{v} \end{bmatrix} = \begin{bmatrix} 0 \\ \mathbf{q}_{WB} \odot {}_B\mathbf{v} \end{bmatrix} = \bar{\mathbf{Q}}^\top(\mathbf{q}_{WB}) \cdot \mathbf{Q}(\mathbf{q}_{WB}) \cdot \begin{bmatrix} 0 \\ {}_B\mathbf{v} \end{bmatrix}. \tag{137}$$

This equation will be more clear by looking at the derivations for transforming a quaternion into a rotation matrix as described in Section 7.2.3.

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 (120).

$$\begin{bmatrix} 0 \\ {}_W\boldsymbol{\omega}_{WB} \end{bmatrix} = 2\dot{\mathbf{q}}_{WB} \otimes \bar{\mathbf{q}}_{WB}, \tag{138}$$
$$= 2\mathbf{Q}(\dot{\mathbf{q}}_{WB})\,\bar{\mathbf{q}}_{WB}, \tag{139}$$
$$= 2\bar{\mathbf{Q}}(\bar{\mathbf{q}}_{WB})\,\dot{\mathbf{q}}_{WB}, \tag{140}$$

or in body fixed frame B

$$\begin{bmatrix} 0 \\ {}_W\boldsymbol{\omega}_{WB} \end{bmatrix} = 2\bar{\mathbf{q}}_{WB} \otimes \dot{\mathbf{q}}_{WB}. \tag{141}$$

More compactly, we can write

$${}_W\boldsymbol{\omega}_{WB} = 2\mathbf{W}(\mathbf{q}_{WB})\,\dot{\mathbf{q}}_{WB}, \tag{142}$$
$${}_W\boldsymbol{\omega}_{WB} = 2\mathbf{W}'(\mathbf{q}_{WB})\,\dot{\mathbf{q}}_{WB}, \tag{143}$$

with $\mathbf{W}(\mathbf{q}_{WB})$ and $\mathbf{W}'(\mathbf{q}_{WB})$ as defined in Section 7.2.3.

7.2.3 Changing Transformation Representation

This section lists the common conversions between different orientation representations.

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

$$\phi = \operatorname{atan2}(R_{32}, R_{33}) \tag{144}$$

$$\theta = \operatorname{atan2}(R_{31}, \sqrt{R_{32}^2 + R_{33}^2}) \tag{145}$$

$$\psi = \operatorname{atan2}(R_{21}, R_{11}) \tag{146}$$

Euler Angles to Rotation Matrix

$$\mathbf{R}_{\text{WB}} = \begin{bmatrix} \mathrm{c}(\psi)\,\mathrm{c}(\theta) & \mathrm{c}(\psi)\,\mathrm{s}(\theta)\,\mathrm{s}(\phi) - \mathrm{s}(\psi)\,\mathrm{c}(\phi) & \mathrm{c}(\psi)\,\mathrm{s}(\theta)\,\mathrm{c}(\phi) + \mathrm{s}(\psi)\,\mathrm{s}(\phi) \\ \mathrm{s}(\psi)\,\mathrm{c}(\theta) & \mathrm{s}(\psi)\,\mathrm{s}(\theta)\,\mathrm{s}(\phi) + \mathrm{c}(\psi)\,\mathrm{c}(\phi) & \mathrm{s}(\psi)\,\mathrm{s}(\theta)\,\mathrm{c}(\phi) - \mathrm{c}(\psi)\,\mathrm{s}(\phi) \\ -\mathrm{s}(\theta) & \mathrm{c}(\theta)\,\mathrm{s}(\phi) & \mathrm{c}(\theta)\,\mathrm{c}(\phi) \end{bmatrix} \tag{147}$$

Quaternion to Euler Angles

$$\phi = \operatorname{atan2}(2q_w q_x + 2q_y q_z,\ q_w^2 - q_x^2 - q_y^2 + q_z^2) \tag{148}$$

$$\theta = -\operatorname{asin}(2q_x q_z - 2q_w q_y) \tag{149}$$

$$\psi = \operatorname{atan2}(2q_w q_z + 2q_x q_y,\ q_w^2 + q_x^2 - q_y^2 - q_z^2) \tag{150}$$

Euler Angles to Quaternion

$$\mathbf{q}_{\text{WB}} = \begin{bmatrix} q_w \\ q_x \\ q_y \\ q_z \end{bmatrix} = \begin{bmatrix} \cos\left(\frac{\phi}{2}\right)\cos\left(\frac{\theta}{2}\right)\cos\left(\frac{\psi}{2}\right) + \sin\left(\frac{\phi}{2}\right)\sin\left(\frac{\theta}{2}\right)\sin\left(\frac{\psi}{2}\right) \\ \sin\left(\frac{\phi}{2}\right)\cos\left(\frac{\theta}{2}\right)\cos\left(\frac{\psi}{2}\right) - \cos\left(\frac{\phi}{2}\right)\sin\left(\frac{\theta}{2}\right)\sin\left(\frac{\psi}{2}\right) \\ \cos\left(\frac{\phi}{2}\right)\sin\left(\frac{\theta}{2}\right)\cos\left(\frac{\psi}{2}\right) + \sin\left(\frac{\phi}{2}\right)\cos\left(\frac{\theta}{2}\right)\sin\left(\frac{\psi}{2}\right) \\ \cos\left(\frac{\phi}{2}\right)\cos\left(\frac{\theta}{2}\right)\sin\left(\frac{\psi}{2}\right) - \sin\left(\frac{\phi}{2}\right)\sin\left(\frac{\theta}{2}\right)\cos\left(\frac{\psi}{2}\right) \end{bmatrix} \tag{151}$$

Quaternion to Rotation Matrix Unit quaternions are quaternions with unity norm. Throughout this section, we assume that

$$\|\mathbf{q}\| = 1. \tag{152}$$

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

$$\begin{bmatrix} 0 \\ {}_{\text{W}}\mathbf{c} \end{bmatrix} = \mathbf{q} \cdot \begin{bmatrix} 0 \\ {}_{\text{B}}\mathbf{c} \end{bmatrix} \cdot \mathbf{q}^{-1}, \tag{153}$$

$$= \mathbf{q} \cdot \begin{bmatrix} 0 \\ {}_{\text{B}}\mathbf{c} \end{bmatrix} \cdot \bar{\mathbf{q}}, \tag{154}$$

$$= \bar{\mathbf{Q}}\left(\mathbf{q}\right)^{\top} \mathbf{Q}\left(\mathbf{q}\right) \begin{bmatrix} 0 \\ {}_{\text{B}}\mathbf{c} \end{bmatrix}, \tag{155}$$

$$= \begin{bmatrix} 1 & \mathbf{0}^{\top} \\ \mathbf{0} & \mathbf{R}_{\text{WB}} \end{bmatrix} \begin{bmatrix} 0 \\ {}_{\text{B}}\mathbf{c} \end{bmatrix}. \tag{156}$$

From (155) and (156), it can be seen that the rotation matrix can be as

$$\mathbf{R}_{\mathrm{WB}} = \begin{bmatrix} q_w^2 + q_x^2 - q_y^2 - q_z^2, & 2q_xq_y - 2q_wq_z, & 2q_wq_y + 2q_xq_z \\ 2q_wq_z + 2q_xq_y, & q_w^2 - q_x^2 + q_y^2 - q_z^2, & 2q_yq_z - 2q_wq_x \\ 2q_xq_z - 2q_wq_y, & 2q_wq_x + 2q_yq_z, & q_w^2 - q_x^2 - q_y^2 + q_z^2 \end{bmatrix}. \tag{157}$$

To rewrite (155) and (156), we define

$$\mathbf{W}(\mathbf{q}_{\mathrm{WB}}) = \begin{bmatrix} -q_x & q_w & -q_z & q_y \\ -q_y & q_z & q_w & -q_x \\ -q_z & -q_y & q_x & q_w \end{bmatrix}, \tag{158}$$

$$\mathbf{W}'(\mathbf{q}_{\mathrm{WB}}) = \begin{bmatrix} -q_x & q_w & q_z & -q_y \\ -q_y & -q_z & q_w & q_x \\ -q_z & q_y & -q_x & q_w \end{bmatrix}, \tag{159}$$

which allows us to write (157) as

$$\mathbf{R}_{\mathrm{WB}} = \mathbf{W}(\mathbf{q}_{\mathrm{WB}})\mathbf{W}'(\mathbf{q}_{\mathrm{WB}})^{\top}. \tag{160}$$

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

$$\mathbf{R}(\mathbf{q} \cdot \mathbf{p}) = \mathbf{R}(\mathbf{q})\,\mathbf{R}(\mathbf{p})\,. \tag{161}$$

Getting a quaternion directly from a rotation matrix is not straight forward but is also not used very often in our applications and if, we let Eigen do it.

7.3 Classical Modeling of a Quadrotor

7.3.1 Using Euler Angles

In this section we use Euler angles and the corresponding rotation matrices as defined in Section 7.2.1, where the rotation matrix $\mathbf{R}_{\mathrm{WB}}(\phi, \theta, \psi)$ is a function of the euler angles.

We can evaluate (87) in the world frame as

$$\begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ -g \end{bmatrix} + \mathbf{R}_{\mathrm{WB}}(\phi, \theta, \psi) \begin{bmatrix} 0 \\ 0 \\ c \end{bmatrix}, \tag{162}$$

or in vectorized form

$$\ddot{\mathbf{p}} = {}_{\mathrm{W}}\mathbf{g} + \mathbf{R}_{\mathrm{WB}}(\phi, \theta, \psi) \cdot \mathbf{c}. \tag{163}$$

Next, we evaluate (92) in the body coordinate system B

$$\begin{bmatrix} J_{xx} & 0 & 0 \\ 0 & J_{yy} & 0 \\ 0 & 0 & J_{zz} \end{bmatrix} \begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} + \begin{bmatrix} p \\ q \\ r \end{bmatrix} \times \begin{bmatrix} J_{xx} & 0 & 0 \\ 0 & J_{yy} & 0 \\ 0 & 0 & J_{zz} \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix} = \begin{bmatrix} \frac{\sqrt{2}}{2} l(f_1 - f_2 - f_3 + f_4) \\ \frac{\sqrt{2}}{2} l(-f_1 - f_2 + f_3 + f_4) \\ \kappa(f_1 - f_2 + f_3 - f_4) \end{bmatrix}, \tag{164}$$

$$\begin{bmatrix} J_{xx}\dot{p} + q \cdot r\,(J_{zz} - J_{yy}) \\ J_{yy}\dot{q} + p \cdot r\,(J_{xx} - J_{zz}) \\ J_{zz}\dot{r} + p \cdot q\,(J_{yy} - J_{xx}) \end{bmatrix} = \begin{bmatrix} \frac{\sqrt{2}}{2} l(f_1 - f_2 - f_3 + f_4) \\ \frac{\sqrt{2}}{2} l(-f_1 - f_2 + f_3 + f_4) \\ \kappa(f_1 - f_2 + f_3 - f_4) \end{bmatrix}, \tag{165}$$

or in vectorized form

$$\mathbf{J} \cdot {}_{\mathrm{B}}\dot{\boldsymbol{\omega}}_{\mathrm{WB}} + {}_{\mathrm{W}}\boldsymbol{\omega}_{\mathrm{WB}} \times \mathbf{J} \cdot {}_{\mathrm{W}}\boldsymbol{\omega}_{\mathrm{WB}} = \boldsymbol{\eta}. \tag{166}$$

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

$$\mathbf{s} = \begin{bmatrix} x & y & z & \dot{x} & \dot{y} & \dot{z} & \phi & \theta & \psi & p & q & r \end{bmatrix}^\top, \tag{167}$$

$$= \begin{bmatrix} {}_{\mathrm{W}}\mathbf{p}_{\mathrm{WB}} & {}_{\mathrm{W}}\mathbf{v}_{\mathrm{WB}} & \phi & \theta & \psi & {}_{\mathrm{W}}\boldsymbol{\omega}_{\mathrm{WB}} \end{bmatrix}^\top. \tag{168}$$

In the following we rewrite (162) and (165) in order to fit the state space form.

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix}, \tag{169}$$

$$\begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ -g \end{bmatrix} + \mathbf{R}_{\mathrm{WB}}(\phi, \theta, \psi) \begin{bmatrix} 0 \\ 0 \\ c \end{bmatrix}, \tag{170}$$

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 1 & \mathrm{t}(\theta)\,\mathrm{s}(\phi) & \mathrm{t}(\theta)\,\mathrm{c}(\phi) \\ 0 & \mathrm{c}(\phi) & -\mathrm{s}(\phi) \\ 0 & \frac{\mathrm{s}(\phi)}{\mathrm{c}(\theta)} & \frac{\mathrm{c}(\phi)}{\mathrm{c}(\theta)} \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix}, \tag{171}$$

$$\begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} = \begin{bmatrix} \frac{1}{J_{xx}} \left(\frac{\sqrt{2}}{2} l(f_1 - f_2 - f_3 + f_4) - q \cdot r\,(J_{zz} - J_{yy}) \right) \\ \frac{1}{J_{yy}} \left(\frac{\sqrt{2}}{2} l(-f_1 - f_2 + f_3 + f_4) - p \cdot r\,(J_{xx} - J_{zz}) \right) \\ \frac{1}{J_{zz}} \left(\kappa(f_1 - f_2 + f_3 - f_4) - q \cdot p\,(J_{yy} - J_{xx}) \right) \end{bmatrix}, \tag{172}$$

or in vectorized form

$${}_{\mathrm{W}}\dot{\mathbf{p}}_{\mathrm{WB}} = {}_{\mathrm{W}}\mathbf{v}_{\mathrm{WB}}, \tag{173}$$

$${}_{\mathrm{W}}\dot{\mathbf{v}}_{\mathrm{WB}} = {}_{\mathrm{W}}\mathbf{g} + \mathbf{R}_{\mathrm{WB}}(\phi, \theta, \psi) \cdot \mathbf{c}, \tag{174}$$

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 1 & \mathrm{t}(\theta)\,\mathrm{s}(\phi) & \mathrm{t}(\theta)\,\mathrm{c}(\phi) \\ 0 & \mathrm{c}(\phi) & -\mathrm{s}(\phi) \\ 0 & \frac{\mathrm{s}(\phi)}{\mathrm{c}(\theta)} & \frac{\mathrm{c}(\phi)}{\mathrm{c}(\theta)} \end{bmatrix} {}_{\mathrm{W}}\boldsymbol{\omega}_{\mathrm{WB}}, \tag{175}$$

$${}_{\mathrm{B}}\dot{\boldsymbol{\omega}}_{\mathrm{WB}} = \mathbf{J}^{-1} \cdot \left(\boldsymbol{\eta} - {}_{\mathrm{W}}\boldsymbol{\omega}_{\mathrm{WB}} \times \mathbf{J} \cdot {}_{\mathrm{B}}\boldsymbol{\omega}_{\mathrm{WB}} \right). \tag{176}$$

Finally, the derivation of the Euler angle derivatives in (171) and (175) is shown here. They are a function of the Euler angles and the body rates:

$$\begin{bmatrix}\dot{\phi}\\ \dot{\theta}\\ \dot{\psi}\end{bmatrix} = \begin{bmatrix}\dot{\phi}(\phi,\theta,\psi,p,q,r)\\ \dot{\theta}(\phi,\theta,\psi,p,q,r)\\ \dot{\psi}(\phi,\theta,\psi,p,q,r)\end{bmatrix}. \tag{177}$$

For this we use (125):

$$\begin{bmatrix}p\\ q\\ r\end{bmatrix} = \begin{bmatrix}1 & 0 & -\mathrm{s}(\theta)\\ 0 & \mathrm{c}(\phi) & \mathrm{s}(\phi)\,\mathrm{c}(\theta)\\ 0 & -\mathrm{s}(\phi) & \mathrm{c}(\phi)\,\mathrm{c}(\theta)\end{bmatrix}\begin{bmatrix}\dot{\phi}\\ \dot{\theta}\\ \dot{\psi}\end{bmatrix}. \tag{178}$$

Inverting (178) finally leads to

$$\begin{bmatrix}\dot{\phi}\\ \dot{\theta}\\ \dot{\psi}\end{bmatrix} = \begin{bmatrix}1 & \mathrm{t}(\theta)\,\mathrm{s}(\phi) & \mathrm{t}(\theta)\,\mathrm{c}(\phi)\\ 0 & \mathrm{c}(\phi) & -\mathrm{s}(\phi)\\ 0 & \frac{\mathrm{s}(\phi)}{\mathrm{c}(\theta)} & \frac{\mathrm{c}(\phi)}{\mathrm{c}(\theta)}\end{bmatrix}\begin{bmatrix}p\\ q\\ r\end{bmatrix}. \tag{179}$$

7.3.2 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 (87) in the world frame as

$$\ddot{\mathbf{p}} = {}_{\mathrm{W}}\mathbf{g} + \mathbf{R}_{\mathrm{WB}} \cdot \mathbf{c}. \tag{180}$$

Since we use the same representation of the body rates as in Section 7.3.1, the evaluation of (92) remains the same as in (166).

We chose the state vector as

$$\mathbf{s} = \begin{bmatrix}{}_{\mathrm{W}}\mathbf{p}_{\mathrm{WB}} & {}_{\mathrm{W}}\mathbf{v}_{\mathrm{WB}} & \mathbf{R}_{\mathrm{WB}} & {}_{\mathrm{W}}\boldsymbol{\omega}_{\mathrm{WB}}\end{bmatrix}^{\top}. \tag{181}$$

In the following, we rewrite (180) and (166) in order to fit the state space form.

$$\begin{aligned}
{}_{\mathrm{W}}\dot{\mathbf{p}}_{\mathrm{WB}} &= {}_{\mathrm{W}}\mathbf{v}_{\mathrm{WB}}, && (182)\\
{}_{\mathrm{W}}\dot{\mathbf{v}}_{\mathrm{WB}} &= {}_{\mathrm{W}}\mathbf{g} + \mathbf{R}_{\mathrm{WB}} \cdot \mathbf{c}, && (183)\\
\dot{\mathbf{R}}_{\mathrm{WB}} &= \mathbf{R}_{\mathrm{WB}} \cdot {}_{\mathrm{B}}\hat{\boldsymbol{\omega}}_{\mathrm{WB}}, && (184)\\
{}_{\mathrm{B}}\dot{\boldsymbol{\omega}}_{\mathrm{WB}} &= \mathbf{J}^{-1} \cdot \left(\boldsymbol{\eta} - {}_{\mathrm{W}}\boldsymbol{\omega}_{\mathrm{WB}} \times \mathbf{J} \cdot {}_{\mathrm{W}}\boldsymbol{\omega}_{\mathrm{WB}}\right). && (185)
\end{aligned}$$

7.3.3 Using Quaternions

In this section, we use a quaternion $\mathbf{q}_{\mathrm{WB}}$, as defined in Section 7.2.2, to represent the orientation of the quadrotor. We will again use the vectorized forms of the equations only.

We can evaluate (87) in the world frame as

$$\ddot{\mathbf{p}} = {}_{\mathrm{W}}\mathbf{g} + \mathbf{q}_{\mathrm{WB}} \odot \mathbf{c}. \tag{186}$$

Since we use the same representation of the body rates as in Section 7.3.1, the evaluation of (92) remains the same as in (166).

We chose the state vector as

$$\mathbf{s} = \begin{bmatrix} {}_{\mathrm{W}}\mathbf{p}_{\mathrm{WB}} & {}_{\mathrm{W}}\mathbf{v}_{\mathrm{WB}} & \mathbf{q}_{\mathrm{WB}} & {}_{\mathrm{W}}\boldsymbol{\omega}_{\mathrm{WB}} \end{bmatrix}^{\top}. \tag{187}$$

In the following, we rewrite (186) and (166) in order to fit the state space form.

$$ {}_{\mathrm{W}}\dot{\mathbf{p}}_{\mathrm{WB}} = {}_{\mathrm{W}}\mathbf{v}_{\mathrm{WB}}, \tag{188}$$
$$ {}_{\mathrm{W}}\dot{\mathbf{v}}_{\mathrm{WB}} = {}_{\mathrm{W}}\mathbf{g} + \mathbf{q}_{\mathrm{WB}} \odot \mathbf{c}, \tag{189}$$
$$ \dot{\mathbf{q}}_{\mathrm{WB}} = \boldsymbol{\Lambda}({}_{\mathrm{W}}\boldsymbol{\omega}_{\mathrm{WB}}) \cdot \mathbf{q}_{\mathrm{WB}}, \tag{190}$$
$$ {}_{\mathrm{B}}\dot{\boldsymbol{\omega}}_{\mathrm{WB}} = \mathbf{J}^{-1} \cdot \left(\boldsymbol{\eta} - {}_{\mathrm{W}}\boldsymbol{\omega}_{\mathrm{WB}} \times \mathbf{J} \cdot {}_{\mathrm{W}}\boldsymbol{\omega}_{\mathrm{WB}}\right), \tag{191}$$

where

$$\boldsymbol{\Lambda}({}_{\mathrm{W}}\boldsymbol{\omega}_{\mathrm{WB}}) = \frac{1}{2}\begin{bmatrix} 0 & -p & -q & -r \\ p & 0 & r & -q \\ q & -r & 0 & p \\ r & q & -p & 0 \end{bmatrix} = \frac{1}{2}\bar{\mathbf{Q}}\left(\begin{bmatrix} 0 \\ {}_{\mathrm{W}}\boldsymbol{\omega}_{\mathrm{WB}} \end{bmatrix}\right) = \frac{1}{2}\bar{\mathbf{Q}}\left(\begin{bmatrix} 0 \\ p \\ q \\ r \end{bmatrix}\right). \tag{192}$$

We find (190) by using (141)

$$\dot{\mathbf{q}}_{\mathrm{WB}} = \frac{1}{2}\mathbf{q}_{\mathrm{WB}} \cdot \begin{bmatrix} 0 \\ {}_{\mathrm{W}}\boldsymbol{\omega}_{\mathrm{WB}} \end{bmatrix}, \tag{193}$$
$$= \frac{1}{2}\bar{\mathbf{Q}}\left(\begin{bmatrix} 0 \\ {}_{\mathrm{W}}\boldsymbol{\omega}_{\mathrm{WB}} \end{bmatrix}\right)\mathbf{q}_{\mathrm{WB}}. \tag{194}$$

Equation (190) can be integrated as described in Appendix B.

Appendices

A Сингулярности в управляющих сигналах

В этом разделе описаны практические рекомендации для некоторых специальных случаев, когда представленная математика не корректна. Как правило, эти рекомендации работают только если мы можем предположить, что такие события длятся достаточно короткий промежуток времени.

Случай — $\mathbf{y}_\mathrm{C} \times \boldsymbol{\alpha} = \mathbf{0}$ This case occurs if either $\mathbf{y}_\mathrm{C}$ is aligned with $\boldsymbol{\alpha}$ (defined in (14)) or $\boldsymbol{\alpha} = \mathbf{0}$. In this case, any $\mathbf{x}_\mathrm{B}$ that is perpendicular to $\mathbf{y}_\mathrm{C}$ satisfies the constraint (14). To overcome this ambiguity, we compute $\mathbf{x}_\mathrm{B}$ by projecting the estimated body x-axis into the $\mathbf{x}_\mathrm{C} - \mathbf{z}_\mathrm{C}$ plane and normalizing it as

$$\mathbf{x}_\mathrm{B} = \frac{\mathbf{x}_{\mathrm{B,est}} - \left(\mathbf{x}_{\mathrm{B,est}}^\top \mathbf{y}_\mathrm{C}\right) \mathbf{y}_\mathrm{C}}{\left\| \mathbf{x}_{\mathrm{B,est}} - \left(\mathbf{x}_{\mathrm{B,est}}^\top \mathbf{y}_\mathrm{C}\right) \mathbf{y}_\mathrm{C} \right\|}. \tag{195}$$

If the obtained $\left\| \mathbf{x}_{\mathrm{B,est}} - \left(\mathbf{x}_{\mathrm{B,est}}^\top \mathbf{y}_\mathrm{C}\right) \mathbf{y}_\mathrm{C} \right\| = \mathbf{0}$, we set $\mathbf{x}_\mathrm{B} = \mathbf{x}_\mathrm{C}$. Note that this might lead to jumps in the desired orientation. By assuming that this special case only occurs for a short duration it might be better to just remember the last desired orientation that was computed before the special case occurred.

Case - $\boldsymbol{\beta} \times \mathbf{x}_\mathrm{B} = \mathbf{0}$ This case occurs if either $\mathbf{x}_\mathrm{B}$ is aligned with $\boldsymbol{\beta}$ (defined in (15)) or $\boldsymbol{\beta} = \mathbf{0}$. In this case, any $\mathbf{y}_\mathrm{B}$ that is perpendicular to $\mathbf{x}_\mathrm{B}$ satisfies the constraint (15). To overcome this ambiguity, we compute $\mathbf{y}_\mathrm{B}$ by the cross product of the estimated body z-axis $\mathbf{z}_{\mathrm{B,est}}$ and $\mathbf{x}_\mathrm{B}$ and normalizing it as

$$\mathbf{y}_\mathrm{B} = \frac{\mathbf{z}_{\mathrm{B,est}} \times \mathbf{x}_\mathrm{B}}{\left\| \mathbf{z}_{\mathrm{B,est}} \times \mathbf{x}_\mathrm{B} \right\|}. \tag{196}$$

If the obtained $\left\| \mathbf{z}_{\mathrm{B,est}} \times \mathbf{x}_\mathrm{B} \right\| = \mathbf{0}$, we set $\mathbf{y}_\mathrm{B} = \mathbf{y}_\mathrm{C}$. Note that this might lead to jumps in the desired orientation. By assuming that this special case only occurs for a short duration it might be better to just remember the last desired orientation that was computed before the special case occurred.

Case - Inverted Flight where $\mathbf{z}_\mathrm{W}^\top \boldsymbol{\alpha} < 0$ In the case where $\mathbf{z}_\mathrm{W}^\top \boldsymbol{\alpha} < 0$, (18) leads to a body x-axis $\mathbf{x}_\mathrm{B}$ which has a projection into the $\mathbf{x}_\mathrm{W} - \mathbf{y}_\mathrm{W}$ plane that points into the $-\mathbf{x}_\mathrm{C}$ direction. It is still collinear to $\mathbf{x}_\mathrm{C}$ but this causes the actual heading to be off by 180° from the reference heading ψ_ref. Enforcing the reference heading during inverted flight by changing the sign of the desired $\mathbf{x}_\mathrm{B}$ axis might lead to a jump in the desired orientation. However, to the best of our knowledge, it is not possible to prevent jumps in the orientation for any trajectory. For example, when performing a vertical loop where parts of it are flown upside down, we end up with a continuous orientation of the quadrotor for a reference heading $\psi_\mathrm{ref} = 0°$ when allowing the quadrotor to have its actual heading 180° off as long as it flies upside down. When doing the same with a reference heading $\psi_\mathrm{ref} = 90°$, we do not end up

with a continuous orientation of the quadrotor when applying the same method. In this particular case, changing the sign of $\mathbf{x}_{\text{B}}$ when the quadrotor is inverted, would lead to a continuous orientation over the entire loop. In summary, the presented computation of the desired orientation is not well suited for inverted flights, which require special considerations.

Случай - $\mathcal{A}_2 = 0$ **или** $(\mathcal{B}_1\mathcal{C}_3 - \mathcal{B}_3\mathcal{C}_1) = 0$ Решения of the body rates (27)-(29) and angular accelerations (39)-(41) are obtained by divisions where the denominator is composed of coefficients as defined in (30)-(38). These denominators can become zero which makes the body rates and angular accelerations undefined. Similarly to $\boldsymbol{\alpha} = \mathbf{0}$ and $\boldsymbol{\beta} = \mathbf{0}$, this is the case where the quadrotor is executing ballistic trajectories. In such cases we set $\boldsymbol{\omega} = \mathbf{0}$ and $\dot{\boldsymbol{\omega}} = \mathbf{0}$.

B Интегрирование нулевого порядка единичного кватерниона

Простое покомпонентное интегрирование производной кватерниона не гарантирует, что результат останется единичным кватернионом, и требует нормализации. Under the assumption of a discrete time step in the interval $[t, t + \Delta t]$ with constant body rates ${}_{\text{W}}\boldsymbol{\omega}_{\text{WB}}$, a nice solution for the integration exists (from [15]) which does not require to normalize the quaternion. First we restate (190)

$$\dot{\mathbf{q}}_{\text{WB}} = \boldsymbol{\Lambda}({}_{\text{W}}\boldsymbol{\omega}_{\text{WB}}) \cdot \mathbf{q}_{\text{WB}}. \tag{197}$$

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

$$\mathbf{q}_{\text{WB}}(t + \Delta t) = e^{(\boldsymbol{\Lambda}({}_{\text{W}}\boldsymbol{\omega}_{\text{WB}})\Delta t)} \cdot \mathbf{q}_{\text{WB}}(t). \tag{198}$$

The matrix exponential is defined as

$$e^{(\boldsymbol{\Lambda}({}_{\text{W}}\boldsymbol{\omega}_{\text{WB}})\Delta t)} = \mathbf{I}_4 + \boldsymbol{\Lambda}({}_{\text{W}}\boldsymbol{\omega}_{\text{WB}})\cdot\Delta t + \frac{1}{2!}\left(\boldsymbol{\Lambda}({}_{\text{W}}\boldsymbol{\omega}_{\text{WB}}) \cdot \Delta t\right)^2 + \frac{1}{3!}\left(\boldsymbol{\Lambda}({}_{\text{W}}\boldsymbol{\omega}_{\text{WB}}) \cdot \Delta t\right)^3 + \dots \tag{199}$$

It can easily be verified that

$$\left(\boldsymbol{\Lambda}({}_{\text{W}}\boldsymbol{\omega}_{\text{WB}})\right)^2 = -\frac{1}{4}\|{}_{\text{W}}\boldsymbol{\omega}_{\text{WB}}\|^2 \cdot \mathbf{I}_4, \tag{200}$$

where

$$\|{}_{\text{W}}\boldsymbol{\omega}_{\text{WB}}\| = \sqrt{p^2 + q^2 + r^2}. \tag{201}$$

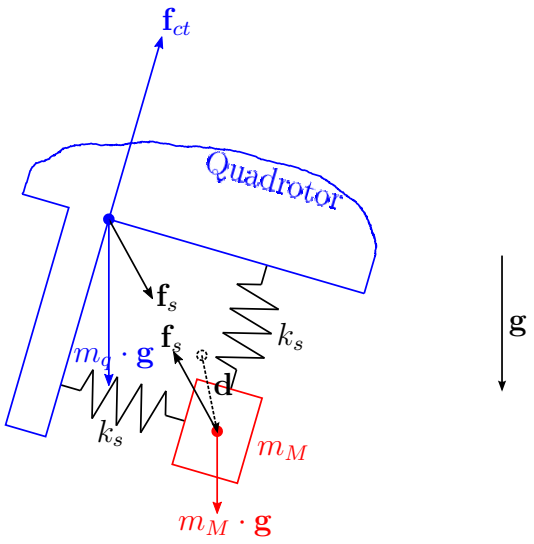

Рис. 9: Schematics of the accelerometer model. The mass m_M is connected to the quadrotor by linear springs with spring constant k_s. No dynamical effects of the mass m_M and its deflection $\mathbf{d}$ are considered.

Using (200) and rearranging the terms, (199) can be written as

$$e^{(\mathbf{\Lambda}({}_{\mathrm{W}}\boldsymbol{\omega}_{\mathrm{WB}})\Delta t)} = \mathbf{I}_4 \cdot \left(1 - \frac{\left(\|{}_{\mathrm{W}}\boldsymbol{\omega}_{\mathrm{WB}}\|\frac{\Delta t}{2}\right)^2}{2!} + \frac{\left(\|{}_{\mathrm{W}}\boldsymbol{\omega}_{\mathrm{WB}}\|\frac{\Delta t}{2}\right)^4}{4!} - \dots\right), \tag{202}$$

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

$$= \mathbf{I}_4 \cdot \cos\left(\frac{\|{}_{\mathrm{W}}\boldsymbol{\omega}_{\mathrm{WB}}\|\Delta t}{2}\right) + \frac{2}{\|{}_{\mathrm{W}}\boldsymbol{\omega}_{\mathrm{WB}}\|} \cdot \mathbf{\Lambda}({}_{\mathrm{W}}\boldsymbol{\omega}_{\mathrm{WB}}) \cdot \sin\left(\frac{\|{}_{\mathrm{W}}\boldsymbol{\omega}_{\mathrm{WB}}\|\Delta t}{2}\right). \tag{203}$$

So the state update of the quaternion becomes

$$\mathbf{q}_{\mathrm{WB}}(t+\Delta t) = \left(\mathbf{I}_4 \cdot \cos\left(\frac{\|{}_{\mathrm{W}}\boldsymbol{\omega}_{\mathrm{WB}}\|\Delta t}{2}\right) + \frac{2}{\|{}_{\mathrm{W}}\boldsymbol{\omega}_{\mathrm{WB}}\|} \cdot \mathbf{\Lambda}({}_{\mathrm{W}}\boldsymbol{\omega}_{\mathrm{WB}}) \cdot \sin\left(\frac{\|{}_{\mathrm{W}}\boldsymbol{\omega}_{\mathrm{WB}}\|\Delta t}{2}\right)\right) \cdot \mathbf{q}_{\mathrm{WB}}(t). \tag{204}$$

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. 9 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

$$\mathbf{f}_s = k_s \cdot \mathbf{d} = m_M \cdot \tilde{\mathbf{a}}_q. \tag{205}$$

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

$$\mathbf{a}_M \cdot m_M = m_M \cdot \mathbf{g} + \mathbf{f}_s, \tag{206}$$
$$\mathbf{a}_q \cdot m_q = \mathbf{f}_{ct} + \mathbf{f}_d + m_q \cdot \mathbf{g} - \mathbf{f}_s, \tag{207}$$

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 (205) into (206) and (207), we get

$$\mathbf{a}_M \cdot m_M = m_M \cdot \mathbf{g} + m_M \cdot \tilde{\mathbf{a}}_q, \tag{208}$$
$$\mathbf{a}_q \cdot m_q = \mathbf{f}_{ct} + \mathbf{f}_d + m_q \cdot \mathbf{g} - m_M \cdot \tilde{\mathbf{a}}_q. \tag{209}$$

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 (208) into (209), we get

$$(\tilde{\mathbf{a}}_q + \mathbf{g}) \cdot m_q = \mathbf{f}_{ct} + \mathbf{f}_d + m_q \cdot \mathbf{g} - m_M \cdot \tilde{\mathbf{a}}_q. \tag{210}$$

Since the mass m_M is very tiny, we can assume that $m_M \ll m_q$. Therefore, we can solve (210) for the measured acceleration of the quadrotor $\tilde{\mathbf{a}}_q$ as

$$\tilde{\mathbf{a}}_q = \frac{1}{m_q} \cdot (\mathbf{f}_{ct} + \mathbf{f}_d)\,. \tag{211}$$

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.

Список литературы

[1] M. Faessler, A. Franchi, and D. Scaramuzza, "Differential flatness of quadrotor dynamics subject to rotor drag for accurate tracking of high-speed trajectories," *IEEE Robot. Autom. Lett.*, vol. 3, pp. 620–626, Apr. 2018.

[2] M. Faessler, D. Falanga, and D. Scaramuzza, "Thrust mixing, saturation, and body-rate control for accurate aggressive quadrotor flight," *IEEE Robot. Autom. Lett.*, vol. 2, pp. 476–482, Apr. 2017.

[3] S. Lupashin, M. Hehn, M. W. Mueller, A. P. Schoellig, M. Sherback, and R. D'Andrea, "A platform for aerial robotics research and demonstration: The Flying Machine Arena," *J. Mechatronics*, vol. 24, pp. 41–54, Feb. 2014.

[4] M. Faessler, F. Fontana, C. Forster, E. Mueggler, M. Pizzoli, and D. Scaramuzza, "Autonomous, vision-based flight and live dense 3D mapping with a quadrotor MAV," *J. Field Robot.*, vol. 33, no. 4, pp. 431–450, 2016.

[5] N. Michael, D. Mellinger, Q. Lindsey, and V. Kumar, "The GRASP multiple micro UAV testbed," *IEEE Robot. Autom. Mag.*, vol. 17, pp. 56–65, Sept. 2010.

[6] J.-M. Kai, G. Allibert, M.-D. Hua, and T. Hamel, "Nonlinear feedback control of quadrotors exploiting first-order drag effects," in *IFAC World Congress*, vol. 50, pp. 8189–8195, July 2017.

[7] J. Svacha, K. Mohta, and V. Kumar, "Improving quadrotor trajectory tracking by compensating for aerodynamic effects," in *IEEE Int. Conf. Unmanned Aircraft Syst. (ICUAS)*, pp. 860–866, June 2017.

[8] D. Mellinger and V. Kumar, "Minimum snap trajectory generation and control for quadrotors," in *IEEE Int. Conf. Robot. Autom. (ICRA)*, pp. 2520–2525, May 2011.

[9] M. Faessler, A. Franchi, and D. Scaramuzza, "Detailed derivations of 'Differential flatness of quadrotor dynamics subject to rotor drag for accurate tracking of high-speed trajectories'," *arXiv e-prints*, Dec. 2017.

[10] M. Faessler, F. Fontana, C. Forster, and D. Scaramuzza, "Automatic re-initialization and failure recovery for aggressive flight with a monocular vision-based quadrotor," in *IEEE Int. Conf. Robot. Autom. (ICRA)*, pp. 1722–1729, 2015.

[11] T. Lee, M. Leoky, and N. McClamroch, "Geometric tracking control of a quadrotor uav on se(3)," in *IEEE Conf. Decision Control (CDC)*, pp. 5420–5425, Dec. 2010.

[12] D. Brescianini, M. Hehn, and R. D'Andrea, "Nonlinear quadrocopter attitude control," tech. rep., Department of Mechanical and Process Engineering, ETHZ, Oct. 2013.

[13] C. G. Mayhew, R. G. Sanfelice, and A. R. Teel, "Quaternion-based hybrid control for robust global attitude tracking," *IEEE Trans. Autom. Control*, vol. 56, pp. 2555–2566, Nov. 2011.

[14] M. Podhradsky, J. Bone, C. Coopmans, and A. Jensen, "Battery model-based thrust controller for a small, low cost multirotor unmanned aerial vehicles," in *IEEE Int. Conf. Unmanned Aircraft Syst. (ICUAS)*, pp. 105–113, May 2013.

[15] N. Trawny and S. I. Roumeliotis, "Indirect Kalman filter for 3D attitude estimation," Tech. Rep. 2005-002, University of Minnesota, Dept. of Comp. Sci. & Eng., Mar. 2005.