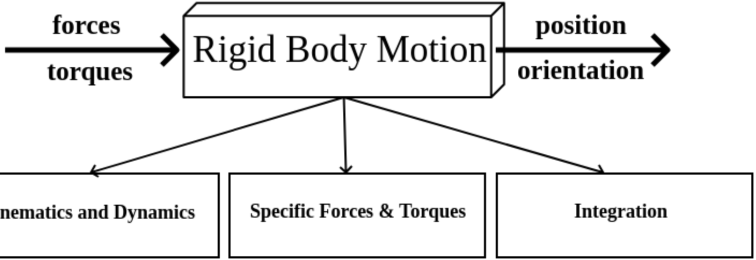
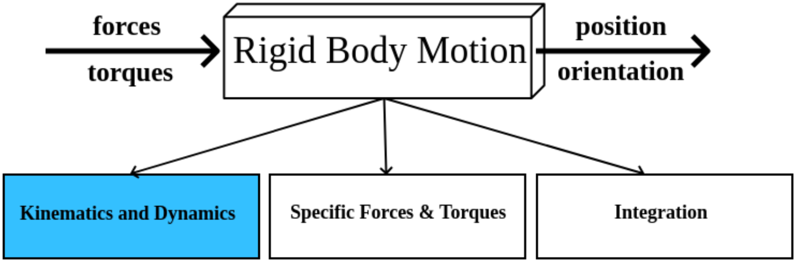
# Rigid Body Motion




- Kinematics: how linear and angular velocity transforms into motion (position and orientation change)
- Dynamics: how forces and torques produce linear acceleration and angular acceleration
- Specific forces and torques for our system (drone)
- Integration: how to solve differential equations

# 3D Kinematics and Dynamics





# 3D Kinematics and Dynamics

## Differential Eqs. of Motion with the Rotation Matrix

$$\dot{\boldsymbol{p}}^e = \boldsymbol{v}^e \tag{1a}$$

$$\dot{\boldsymbol{v}}^e = \frac{1}{m}\boldsymbol{f}^e_{total,ext} = \frac{1}{m}\mathbf{R}^e_b \boldsymbol{f}^b_{total,ext} \tag{1b}$$

$$\dot{\mathbf{R}}^e_b = \mathbf{R}^e_b \left[\boldsymbol{\omega}^b\right]_\times \tag{1c}$$

$$\dot{\boldsymbol{\omega}}^b = \left(\mathbf{J}^b\right)^{-1}\left(-\left[\boldsymbol{\omega}^b\right]_\times \mathbf{J}^b \boldsymbol{\omega}^b + \boldsymbol{\tau}^b\right) \tag{1d}$$

# 3D Kinematics and Dynamics

## Differential Eqs. of Motion with the Unit Quaternion 1/2



$$\dot{\boldsymbol{p}}^e = \boldsymbol{v}^e \tag{2a}$$

$$\dot{\boldsymbol{v}}^e = \frac{1}{m}\boldsymbol{f}^e_{total,ext} = \frac{1}{m}\mathbf{R}^e_b(\boldsymbol{q})\boldsymbol{f}^b_{total,ext} \tag{2b}$$

$$\dot{\boldsymbol{q}} = \begin{bmatrix} \dot{s} \\ \dot{\mathbf{v}} \end{bmatrix} = \frac{1}{2}\begin{bmatrix} -\boldsymbol{v}^T \\ s\mathbf{I}_3 + [\mathbf{v}]_\times \end{bmatrix}\boldsymbol{\omega}^b = \frac{1}{2}\begin{bmatrix} -v_1\omega^b_x - v_2\omega^b_y - v_3\omega^b_z \\ s\omega^b_x - v_3\omega^b_y + v_2\omega^b_z \\ v_3\omega^b_x + s\omega^b_y - v_1\omega^b_z \\ -v_2\omega^b_x + v_1\omega^b_y + s\omega^b_z \end{bmatrix} \tag{2c}$$

$$\dot{\boldsymbol{\omega}}^b = \left(\mathbf{J}^b\right)^{-1}\left(-\left[\omega^b\right]_\times \mathbf{J}^b\boldsymbol{\omega}^b + \boldsymbol{\tau}^b\right) \tag{2d}$$

# 3D Kinematics and Dynamics

## Differential Eqs. of Motion with the Unit Quaternion 2/2



$$
\begin{aligned}
\mathbf{R}_b^e(\boldsymbol{q}) &= \begin{bmatrix} -\boldsymbol{v} & s\mathbf{I}_3 + [\mathbf{v}]_\times \end{bmatrix} \begin{bmatrix} -\boldsymbol{v}^T \\ s\mathbf{I}_3 + [\mathbf{v}]_\times \end{bmatrix} \\
&= 2 \begin{bmatrix} s^2 + v_1^2 - 0.5 & v_1 v_2 - s v_3 & v_1 v_3 + s v_2 \\ v_1 v_2 + v_3 s & s^2 + v_2^2 - 0.5 & -s v_1 + v_2 v_3 \\ v_1 v_3 - s v_2 & v_2 v_3 + v_1 s & s^2 + v_3^2 - 0.5 \end{bmatrix}
\end{aligned}
$$

# 3D Kinematics and Dynamics

## Differential Eqs. of Motion with Euler Angles 1/2



$$\dot{\boldsymbol{p}}^e = \boldsymbol{v}^e \tag{3a}$$

$$\dot{\boldsymbol{v}}^e = \frac{1}{m}\boldsymbol{f}^e_{total,ext} = \frac{1}{m}\mathbf{R}^e_b(\boldsymbol{e})\boldsymbol{f}^b_{total,ext} \tag{3b}$$

$$\dot{\boldsymbol{e}} = \begin{bmatrix} r \\ p \\ y \end{bmatrix} = \frac{1}{\cos p}\begin{bmatrix} \cos p & \sin r \cdot \sin p & \cos r \cdot \sin p \\ 0 & \cos r \cdot \cos p & -\sin r \cdot \cos p \\ 0 & \sin r & \cos r \end{bmatrix} \omega^b \tag{3c}$$

$$\dot{\omega}^b = \left(\mathbf{J}^b\right)^{-1}\left(-\left[\omega^b\right]_\times \mathbf{J}^b \omega^b + \tau^b_c\right) \tag{3d}$$

# 3D Kinematics and Dynamics

## Differential Eqs. of Motion with Euler Angles 2/2



where:

$$\mathbf{R}_b^e(\boldsymbol{e}) = \mathbf{R}_z(y)\mathbf{R}_y(p)\mathbf{R}_x(r)$$

$$\mathbf{R}_z(y) = \begin{bmatrix} \cos y & -\sin y & 0 \\ \sin y & \cos y & 0 \\ 0 & 0 & 1 \end{bmatrix} \mathbf{R}_y(p) = \begin{bmatrix} \cos p & 0 & \sin p \\ 0 & 1 & 0 \\ -\sin p & 0 & \cos p \end{bmatrix} \mathbf{R}_x(r) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos r & -\sin r \\ 0 & \sin r & \cos r \end{bmatrix}$$

# 3D Kinematics and Dynamics

## Note



Special care must be taken in the calculations for numerical errors not to affect the structure of the rotation representations:

- The rotation matrix to be kept orthonormal;
- The quaternion to be kept unitary;

The Euler Angles do not suffer from this problem, although it is a good idea to keep them to within a constant $2\pi$ interval.

- See [Diebel,2006] for a compendium of attitude representations and equivalences

# 3D Kinematics and Dynamics

## Params, Inputs, Initial Conditions of the Differential Eqs. of Motion



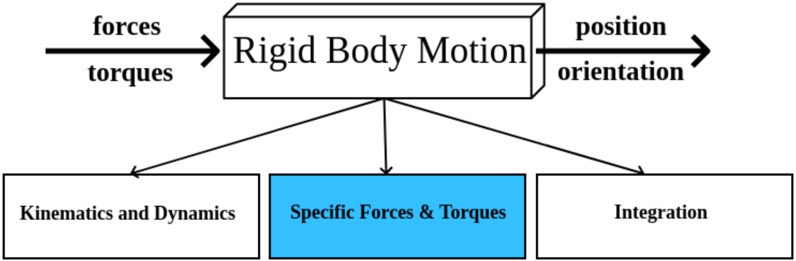
- Parameters: $m$ - the mass of the objects, $\mathbf{J}^b$ - the local/body inertia matrix
- Inputs: forces $\boldsymbol{f}_{total,ext}$ and torques about the center of mass $\tau_C$
- Example of initial conditions: quadrotor object at the origin of the e-frame, b-frame aligned with the e-frame, and the quadrotor is at rest:

$$\boldsymbol{p}^e = \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}, \quad \boldsymbol{v}^e = \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}, \quad \boldsymbol{\omega}^b = \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}$$

$$\mathbf{R}_b^e = \mathbf{I}_3 \text{ and } \boldsymbol{q} = \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix}$$

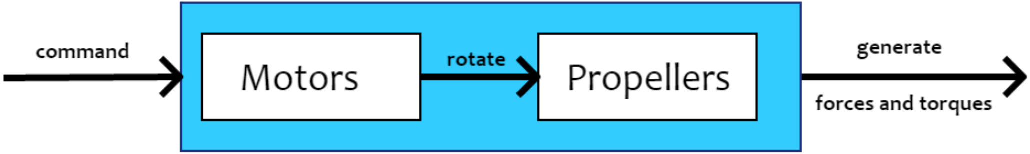
# Forces and Torques





# Forces and Torques





# Forces and Torques

## Overview



- The code on the electronics (firmware/embedded code) sends a digital command to the motors, that is then transformed to an analog PWM (Pulse Width Modulation signal) that commands the coreless motors of the Crazyflie
- The digital command consists of four 16-bit integer numbers (from 0 to 65535), one for each motor.

# Forces and Torques

## Overview



- Given command from 0 to 65535, how fast does a propeller rotate ?
- How much thrust does a rotating propeller produce ?
- How much torque does a rotating propeller produce ?
- What are the aerodynamic forces on the quadrotor in flight ?

# Forces and Torques

## Overview

- Given command from 0 to 65535, we assume that the motors and propellers respond instantaneously that is we do not look at the transitory response
- The aerodynamic forces are due mainly the drag
- We are going to use results from Julian Förster bachelor thesis at ETH Zurich, which is uploaded on the lecture's resources

# Forces and Torques

## Physical Parameters



- Mass of the quadrotor: $m = 0.028$ kg
- Radius of the quadrotor (length from center of mass to the propeller center) $r = 0.045$ m
- Inertia matrix

$$\mathbf{I}^b = \begin{bmatrix} 16.571710 & -0.830806 & -0.718277 \\ -0.830806 & 16.655602 & -1.800197 \\ -0.718277 & -1.800197 & 29.261652 \end{bmatrix} 10^{-6} \cdot \mathrm{kg} \cdot \mathrm{m}^2$$

# Forces and Torques

## Command to Motor Outputs



- Mapping from input command to the motor/propeller angular velocity:

$$\omega_{r,i} = 0.04076521 \cdot \mathrm{cmd}_i + 380.8359 \text{ [rad/s]} \tag{5}$$

- Mapping from input command to the motor/propeller thrust:

$$\begin{aligned} f_{\mathrm{thrust},i} =& 2.130295 \cdot 10^{-11} \cdot \mathrm{cmd}_i^2 \\ &+ 1.03263310^{-6} \cdot \mathrm{cmd}_i + 5.484560 \cdot 10^{-4} \text{ [N]} \end{aligned} \tag{6}$$

- Mapping from motor thrust to motor torque

$$\tau_i = 0.005964552 \cdot f_{\mathrm{thrust},i} + 1.563383 \cdot 10^{-5} \text{ [N·m]} \tag{7}$$

# Forces and Torques
## Drag model



$$\boldsymbol{f}_a^b = K_{\text{aero}} \left( \sum_{i=1}^{4} \omega_{r,i} \right) \boldsymbol{v}^b \tag{8}$$

$$\text{where } K_{\text{aero}} = \begin{bmatrix} -10.2506 & -0.3177 & -0.4332 \\ -0.3177 & -10.2506 & -0.4332 \\ -7.7050 & -7.7050 & -7.5530 \end{bmatrix} 10^{-7}$$

# Forces and Torques
## Final Model



- Force in body frame

$$\boldsymbol{f}^b = \left[0, \quad 0, \quad \textstyle\sum_{i=1}^{4} f_{\text{thrust},i}(\text{cmd}_i)\right]^T + \boldsymbol{f}_a^b \tag{9}$$

- Torque in body frame, for the plus configuration

$$\boldsymbol{\tau}^b = \left[\left(f_{\text{thrust},2} - f_{\text{thrust},4}\right) \cdot r, \quad \left(f_{\text{thrust},3} - f_{\text{thrust},1}\right) \cdot r, \quad -\tau_1 - \tau_3 + \tau_2 + \tau_4\right]^T \tag{10}$$

- Torque in body frame, for the cross-configuration

$$\boldsymbol{\tau}^b = [(f_{\text{thrust},2} + f_{\text{thrust},3} - f_{\text{thrust},1} - f_{\text{thrust},4})\frac{\sqrt{2}}{2} \cdot r, \tag{11}$$

$$(f_{\text{thrust},3} + f_{\text{thrust},4} - f_{\text{thrust},2} - f_{\text{thrust},1})\frac{\sqrt{2}}{2} \cdot r, -\tau_1 - \tau_3 + \tau_2 + \tau_4]^T \tag{12}$$

# Forces and Torque

## Simplified Command-to-Thrust Model



A simplified model where we consider the command to thrust and torque models to be quadratic in the rotor angular velocity

$$f_{\text{thrust},i} = c_T \omega_{r,i}^2,\ c_T = 1.903 \cdot 10^{-8} \tag{13}$$

$$\tau_i = c_Q \omega_{r,i}^2,\ c_Q = c_T \cdot 0.005964552 = 1.246 \cdot 10^{-10} \tag{14}$$

And we ignore the aerodynamic draf force. We use this model for inverse command calculations.

# Forces and Torque

## Simplified Model, Plus Configuration

$$\begin{bmatrix} f_z^b \\ \tau_x \\ \tau_y \\ \tau_z \end{bmatrix} = \underbrace{\begin{bmatrix} c_T & c_T & c_T & c_T \\ 0 & r \cdot c_T & 0 & -r \cdot c_T \\ -r \cdot c_T & 0 & r \cdot c_T & 0 \\ -c_Q & c_Q & -c_Q & c_Q \end{bmatrix}}_{\mathbf{\Gamma}} \begin{bmatrix} \omega_{r,1}^2 \\ \omega_{r,2}^2 \\ \omega_{r,3}^2 \\ \omega_{r,4}^2 \end{bmatrix} \tag{15}$$

# Forces and Torques

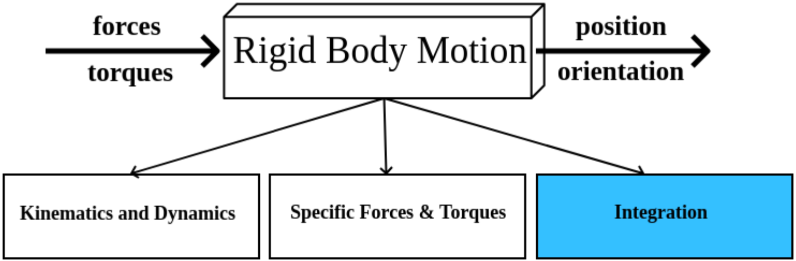
## Simplified Model, Cross Configuration



$$\begin{bmatrix} f_z^b \\ \tau_x \\ \tau_y \\ \tau_z \end{bmatrix} = \underbrace{\begin{bmatrix} c_T & c_T & c_T & c_T \\ -r\frac{\sqrt{2}}{2}c_T & r\frac{\sqrt{2}}{2}c_T & r\frac{\sqrt{2}}{2}c_T & -r\frac{\sqrt{2}}{2}c_T \\ -r\frac{\sqrt{2}}{2}c_T & -r\frac{\sqrt{2}}{2}c_T & r\frac{\sqrt{2}}{2}c_T & r\frac{\sqrt{2}}{2}c_T \\ -c_Q & c_Q & -c_Q & c_Q \end{bmatrix}}_{\mathbf{\Gamma}} \begin{bmatrix} \omega_{r,1}^2 \\ \omega_{r,2}^2 \\ \omega_{r,3}^2 \\ \omega_{r,4}^2 \end{bmatrix} \tag{16}$$

# Integration of Rigid Body Motion Eqs





# Integration

## Simple Euler



- Given a differential equation $\dot{y}(t) = f(y, u, t)$, where $y(t)$ is unknown, $u(t)$ is a known input function, and the initial condition $y(0) = y_0$ is known also, we can obtain an approximation of $y(k\Delta t)$, $\tilde{y}(k\Delta t)$ as

$$\tilde{y}((k+1)\Delta t) = \tilde{y}(k\Delta t) + \dot{y}(k\Delta t)\Delta t$$

$$\boxed{\tilde{y}((k+1)\Delta t) = \tilde{y}(k\Delta t) + f(y(k\Delta t), u(k), k\Delta t)\Delta t}$$

- The smaller $\Delta t$, the better the solution approximation (and more steps to advance in time)