Robot Motion Planning & Control (ME47035)
Lecture 4: Dynamics and Control – Aerial Robots

Javier Alonso-Mora

Cognitive Robotics, TU Delft

Project

Groups are fixed now!

Description and tips for project -→ Brightspace

Template report → Brightspace

Deadline report robot model & eq. motion [feedback only] [Mo. 02.03 EOD]

→ Submit a printed version in my mailbox (J. Alonso Mora) at the end of F-2 or give it to me during the class on Tuesday 03.03. - no reports will be accepted afterwards

Feedback preliminary report [10.03]

Deadline project report [Fr. 03.04 EOD]

Last lecture

Ground robots with wheels

- Kinematics (models)
- Control

Coordinate frames

Body Frame

- Frame fixed to the robot – moves with the robot

Inertial/World frame

- Fixed frame – does not move

For computing the wheel constraints we consider an **instantaneous** Robot frame, that does **not** move with the robot. It is aligned with the robot at this instant. Instantaneously it is equal to the Body frame. But it is **static**.

$$p_R = R(\theta) p_I \quad \rightarrow \quad \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix}_R = \begin{bmatrix} R(\theta) & \mathbf{0} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix}_I$$

Note 1: With an abuse of notation we denote both matrices with $R(\theta)$ in lecture 3..

Note 2: If two frames are moving, then the conversion of velocities is more complex.

This lecture

State space (Dynamics) modelling

“Modeling and Control of Aerial Robots“ Ch. 52.1-3 Handbook of Robotics, 2nd Edition

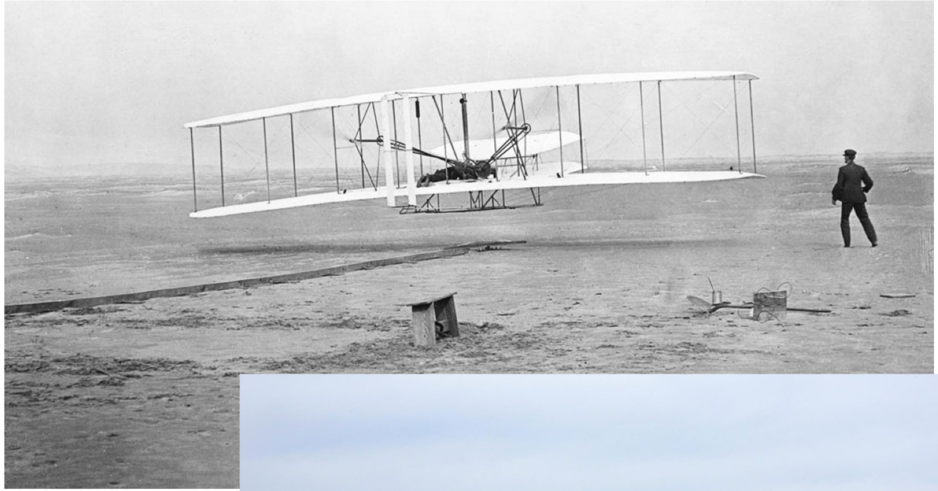
Only the subchapters for Quadrotor control!

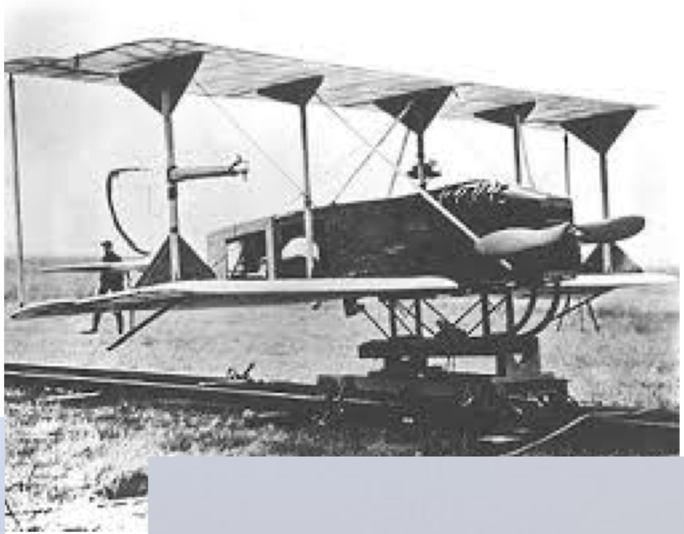
Fix wing (airplane) → optional reading, not included in the exam

Additional Examples

- System Modeling
  http://ctms.engin.umich.edu/CTMS/index.php?example=Introduction&section=SystemModeling
- State-Space Control
  http://ctms.engin.umich.edu/CTMS/index.php?example=Introduction&section=ControlStateSpace

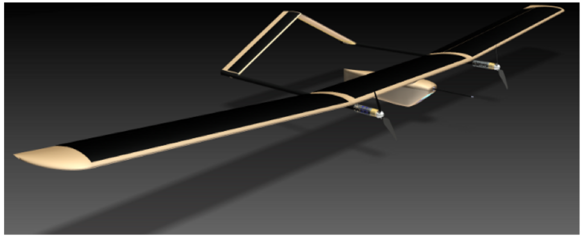
Aerial vehicles





Aerial vehicles

Fixed wing



Multi-rotor

Flapping wing

Blimp

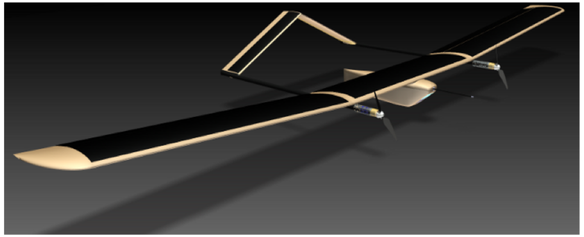
Helicopter

Skye

M. Burri, [. . .], J. Alonso-Mora, R. Siegwart, P. Beardsley, "Design and Control of a Spherical Omnidirectional Blimp", in Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS), Nov. 2013. Video: https://youtu.be/qXvl3anK3w0

Aerial vehicles

Fixed wing



Multi-rotor

Flapping wing

Blimp

Helicopter

This lecture

Newton-Euler equations of motion

- Equations of motion for the quadrotor

State-space system modeling & stability

- State-space model for the quadrotor

Low level control

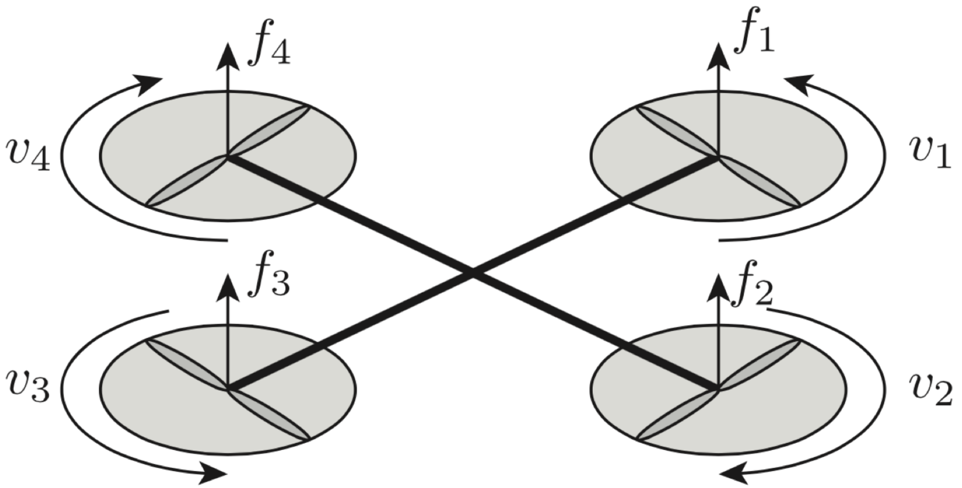
- Attitude and position control for the quadrotor

Formation control

J. Alonso-Mora, E. Montijano, M. Schwager, D. Rus, "Distributed Multi-Robot Navigation in Formation among Obstacles: A Geometric and Optimization Approach with Consensus", in Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (ICRA), May 2016. Video: https://youtu.be/khzM54Qk1QQ

Dynamics and control of quadrotors

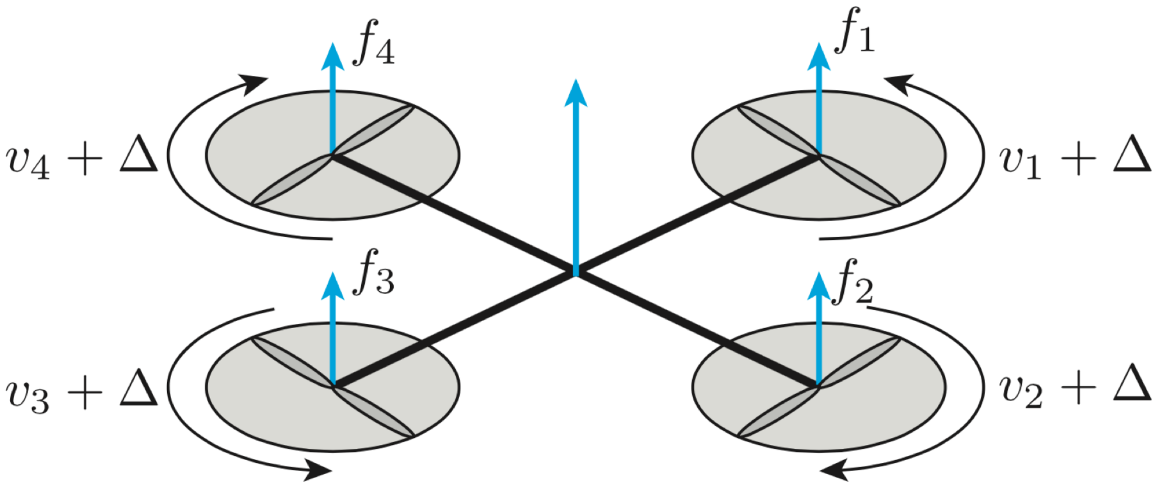
How does it work?



Hovering. All propeller velocities are the same. Force balancing.

Dynamics and control of quadrotors

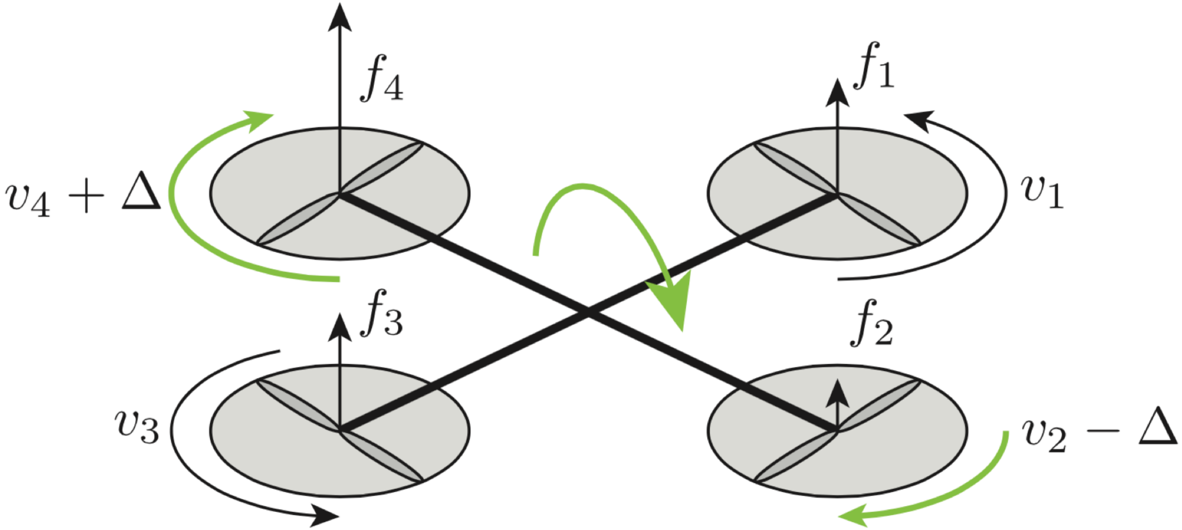
How does it work?



Upwards motion. All propeller velocities are the same. Upwards acceleration.

Dynamics and control of quadrotors

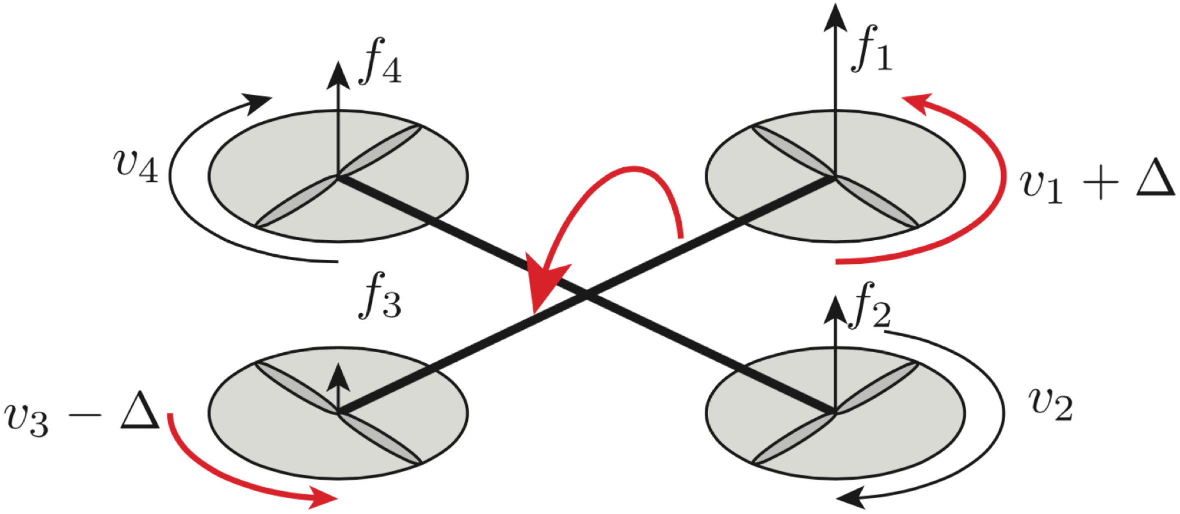
How does it work?



Roll movement. Propeller velocity offsets in v_2 and v_4.

Dynamics and control of quadrotors

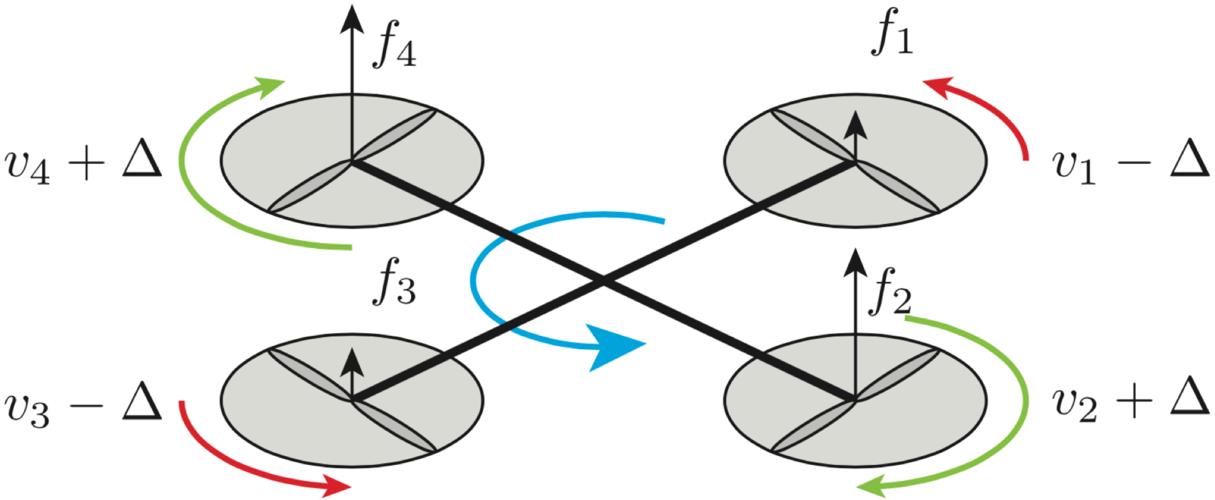
How does it work?



Pitch movement. Propeller velocity offsets in v_1 and v_3.

Dynamics and control of quadrotors

How does it work?



Yaw movement. All propeller velocities are offset.

1. Modelling

17

1.1 Newton-Euler Equations of Motion

18

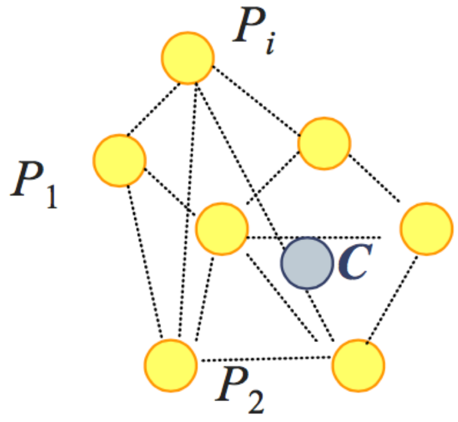
Newton's Second Law for a System of Particles

Recall: For a single particle, $\mathbf{F} = m\mathbf{a}$

The ***center of mass*** for a system of *N* particles accelerates in an inertial frame *A* as if it was a single particle of mass *m* (equal to the total mass of the system) acted upon by a force equal to the net external force. ALSO HOLDS FOR RIGID BODY.

Center of mass: $$\mathbf{r}_c = \frac{1}{m}\sum_{i=1}^{N} m_i \mathbf{p}_i$$

Then: $$\mathbf{F} = \sum_{i=1}^{N} \mathbf{F}_i = \underbrace{m \frac{{}^{A}d\,{}^{A}\mathbf{v}^{c}}{dt}}_{m\mathbf{a}}$$



Rotational Equations of Motion for a Rigid Body

The rate of change of **angular momentum** of the rigid body *S* relative to point *C* in an inertial frame *A* is equal to the resultant **moment** of all external forces acting on the system relative to *C*.

$$\frac{{}^{A}d\,{}^{A}\mathbf{H}_C^B}{dt} = \mathbf{M}_C^B$$

Net moment on body B around point C from all external forces and torques

$${}^{A}\mathbf{H}_C^B = \mathbf{I}_C \cdot {}^{A}\omega^{B}$$

Angular momentum of body B about point C in frame A

inertia tensor

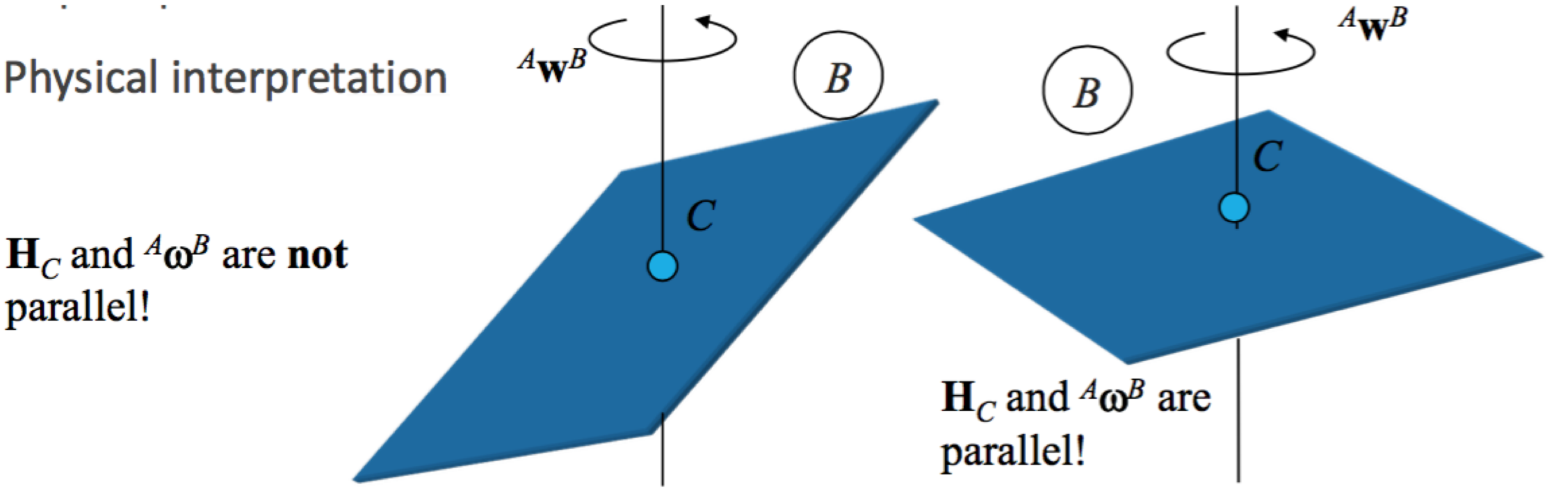
Principal Axes of Inertia

Principal axis of inertia

- $\mathbf{u}$ is a unit vector along a principal axis if $\mathbf{Iu}$ is parallel to $\mathbf{u}$
- There are 3 independent principal axes!

Principal moment of inertia

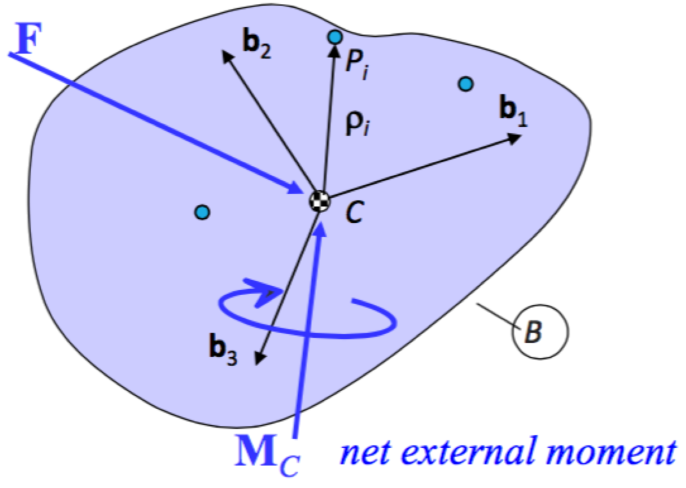
- The moment of inertia with respect to a principal axis, $\mathbf{u}^{\mathrm{T}}\mathbf{I}\mathbf{u}$, is called a principal moment of inertia



Euler's Equations

Let $\mathbf{b}_1$, $\mathbf{b}_2$, $\mathbf{b}_3$, be **along principal axes** and $\quad {}^A\boldsymbol{\omega}^B = \omega_1\,\mathbf{b}_1 + \omega_2\,\mathbf{b}_2 + \omega_3\mathbf{b}_3$

$$\mathbf{M}_c = \frac{{}^A d\mathbf{H}_C^B}{dt} = \frac{{}^B d\mathbf{H}_C^B}{dt} + \underbrace{{}^A\omega^B \times \mathbf{H}_C^B}_{\text{Differentiating in a moving frame}}$$



$$\begin{bmatrix} I_{11} & 0 & 0 \\ 0 & I_{22} & 0 \\ 0 & 0 & I_{33} \end{bmatrix}\begin{bmatrix} \dot\omega_1 \\ \dot\omega_2 \\ \dot\omega_3 \end{bmatrix} + \underbrace{\begin{bmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{bmatrix}}_{\widehat{\omega}}\begin{bmatrix} I_{11} & 0 & 0 \\ 0 & I_{22} & 0 \\ 0 & 0 & I_{33} \end{bmatrix}\begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \end{bmatrix} = \begin{bmatrix} M_{C,1} \\ M_{C,2} \\ M_{C,3} \end{bmatrix}$$

Angular Velocity

Relate the angular velocities to the Euler angles

Recall that $\widehat{\omega}^b = R^T \dot{R}$

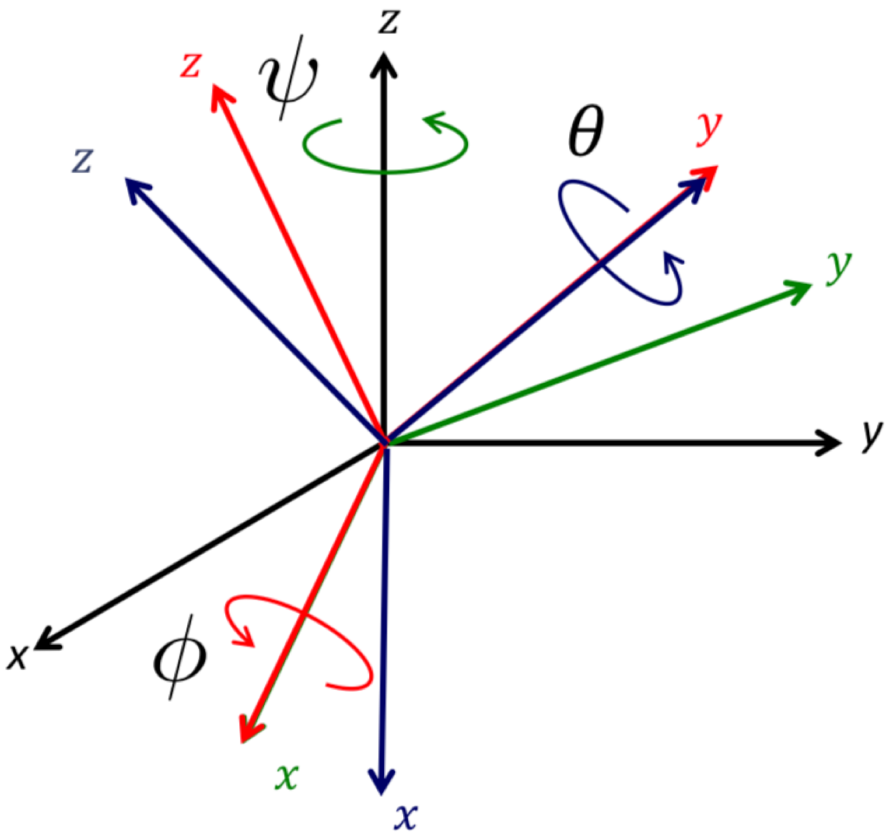
For the ZXY Euler angles (ψ, ϕ, θ) this yields the vector:

$$\omega^b = \begin{bmatrix} p \\ q \\ r \end{bmatrix} = \begin{bmatrix} c_\theta & 0 & -c_\phi s_\theta \\ 0 & 1 & s_\phi \\ s_\theta & 0 & c_\phi c_\theta \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix}$$

Roll rate
Pitch rate
Yaw rate

Angular velocity components in B

ZXY Euler Angles



Sequence of three rotations about **body-fixed** axes

- $R_{z,\psi}$
- $R_{x,\phi}$
- $R_{y,\theta}$

$$R = R_{z,\psi} R_{x,\phi} R_{y,\theta}$$

Yaw, roll, pitch

Note: there are multiple conventions for the order or rotations

1.2 Application to quadrotors

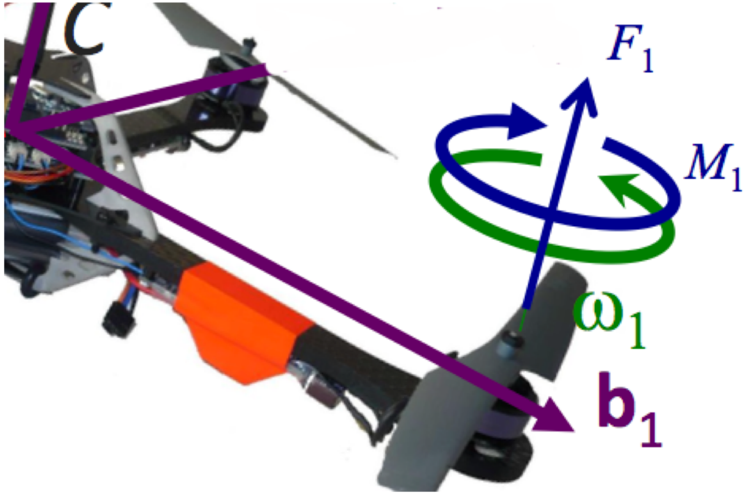
25

Model of a Rotor

Each rotor rotates with angular velocity w and generates a force F and a moment M

$$F_i = k_F \omega_i^2$$

$$M_i = k_M \omega_i^2$$



$$k_F = k_T \rho \, D^4$$

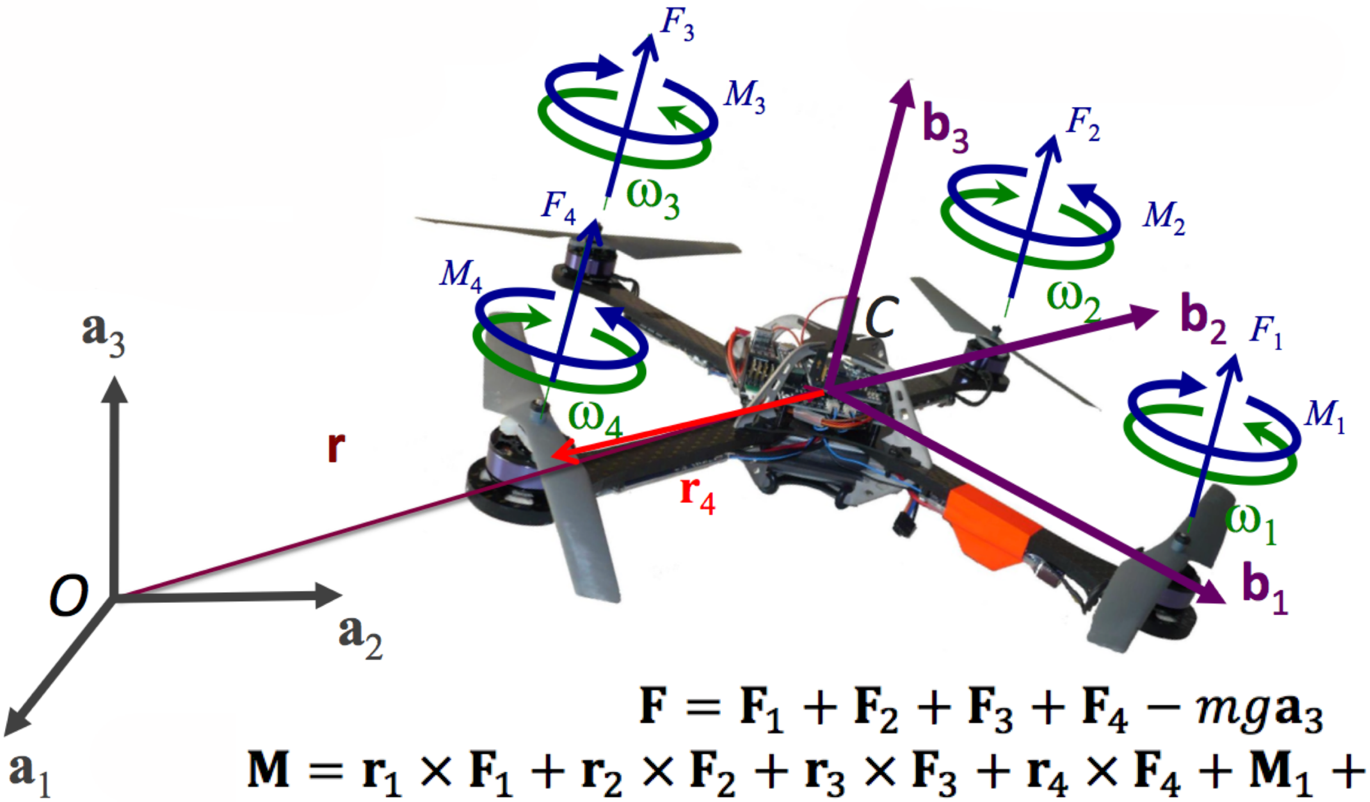
$$k_M = k_Q \rho \, D^5$$

With:

k_T	thrust coefficient
k_Q	torque coefficient
ρ	fluid density
D	diameter of the propeller

Equations of Motion Quadrotor

Total thrust and moment is the sum of individual ones



$$\mathbf{F} = \mathbf{F}_1 + \mathbf{F}_2 + \mathbf{F}_3 + \mathbf{F}_4 - mg\mathbf{a}_3$$

$$\mathbf{M} = \mathbf{r}_1 \times \mathbf{F}_1 + \mathbf{r}_2 \times \mathbf{F}_2 + \mathbf{r}_3 \times \mathbf{F}_3 + \mathbf{r}_4 \times \mathbf{F}_4 + \mathbf{M}_1 + \mathbf{M}_2 + \mathbf{M}_3 + \mathbf{M}_4$$

Newton-Euler Equations for a Quadrotor

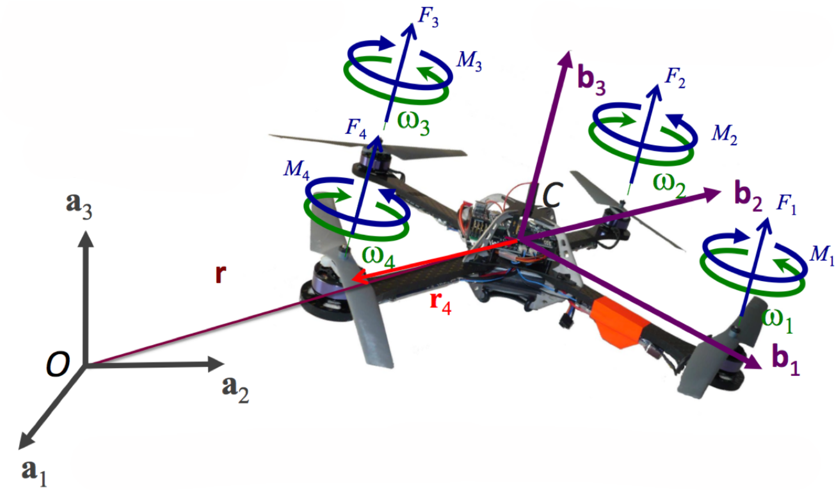
$$^{A}\omega^{B} = p\,\mathbf{b}_1 + q\,\mathbf{b}_2 + r\,\mathbf{b}_3$$

$$m\ddot{\mathbf{r}} = \begin{bmatrix} 0 \\ 0 \\ -mg \end{bmatrix} + R \begin{bmatrix} 0 \\ 0 \\ \underbrace{\boxed{F_1 + F_2 + F_3 + F_4}}_{u_1} \end{bmatrix}$$

In inertial frame

Thrust $T = u_1$

Newton-Euler Equations for a Quadrotor



Angular rates

$$I \begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} = \boxed{\begin{bmatrix} L(F_2 - F_4) \\ L(F_3 - F_1) \\ -M_1 + M_2 - M_3 + M_4 \end{bmatrix}} - \begin{bmatrix} p \\ q \\ r \end{bmatrix} \times I \begin{bmatrix} p \\ q \\ r \end{bmatrix}$$

In body frame

$\mathbf{u}_2$ Angular moments

Newton-Euler Equations for a Quadrotor

Recall that $\mathbf{F}_i = k_F \omega_i^2$ and $\mathbf{M}_i = k_M \omega_i^2$

Let $\gamma = \frac{k_M}{k_F} = \frac{\mathbf{M}_i}{\mathbf{F}_i} \Leftrightarrow \mathbf{M}_i = \gamma \mathbf{F}_i$

$$I \begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} = \begin{bmatrix} L(F_2 - F_4) \\ L(F_3 - F_1) \\ -M_1 + M_2 - M_3 + M_4 \end{bmatrix} - \begin{bmatrix} p \\ q \\ r \end{bmatrix} \times I \begin{bmatrix} p \\ q \\ r \end{bmatrix}$$

$$I \begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} = \underbrace{\boxed{\begin{bmatrix} 0 & L & 0 & -L \\ -L & 0 & L & 0 \\ -\gamma & \gamma & -\gamma & \gamma \end{bmatrix} \begin{bmatrix} F_1 \\ F_2 \\ F_3 \\ F_4 \end{bmatrix}}}_{\mathbf{u}_2} - \begin{bmatrix} p \\ q \\ r \end{bmatrix} \times I \begin{bmatrix} p \\ q \\ r \end{bmatrix}$$

Inputs

Joint equations

$$\mathbf{u} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 0 & L & 0 & -L \\ -L & 0 & L & 0 \\ -\gamma & \gamma & -\gamma & \gamma \end{bmatrix} \begin{bmatrix} F_1 \\ F_2 \\ F_3 \\ F_4 \end{bmatrix}$$

$$= \begin{bmatrix} \text{thrust} \\ \text{moment about } x \\ \text{moment about } y \\ \text{moment about } z \end{bmatrix}$$

$$\Rightarrow$$

$$\begin{bmatrix} T \\ \tau_1 \\ \tau_2 \\ \tau_3 \end{bmatrix} = \begin{bmatrix} k_F & k_F & k_F & k_F \\ 0 & Lk_F & 0 & -Lk_F \\ -Lk_F & 0 & Lk_F & 0 \\ -k_M & k_M & -k_M & k_M \end{bmatrix} \begin{bmatrix} \varpi_1^2 \\ \varpi_2^2 \\ \varpi_3^2 \\ \varpi_4^2 \end{bmatrix}$$

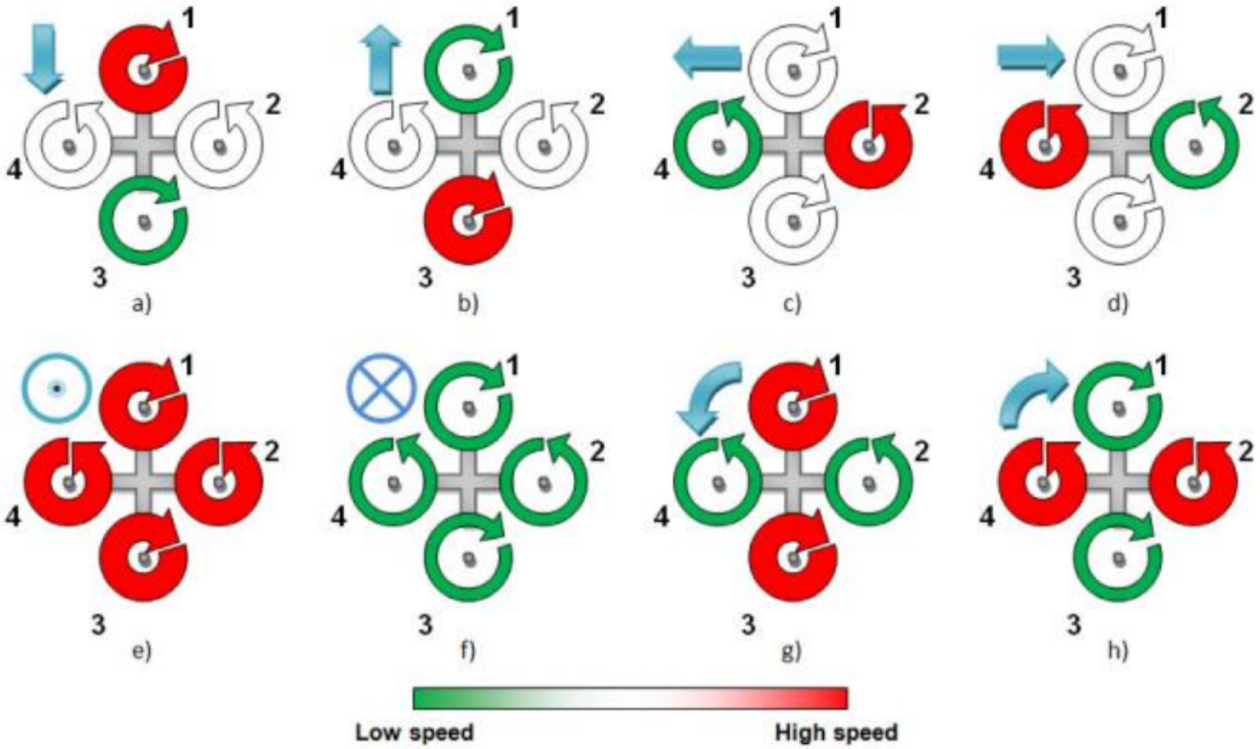
Motor speeds

Note: $\mathbf{u} = \begin{bmatrix} u_1 \\ \boldsymbol{u}_2 \end{bmatrix} = \begin{bmatrix} T \\ \tau \end{bmatrix} = \begin{bmatrix} T \\ \tau_1 \\ \tau_2 \\ \tau_3 \end{bmatrix}$

All quantities in body frame

Example

Different maneuvers depending on the speed and direction of the blades' rotation



Equations quadrotor

Motor speeds to thrust and moments

$$\begin{bmatrix} T \\ \tau_1 \\ \tau_2 \\ \tau_3 \end{bmatrix} = \begin{bmatrix} k_F & k_F & k_F & k_F \\ 0 & Lk_F & 0 & -Lk_F \\ -Lk_F & 0 & Lk_F & 0 \\ -k_M & k_M & -k_M & k_M \end{bmatrix} \begin{bmatrix} \varpi_1{}^2 \\ \varpi_2{}^2 \\ \varpi_3{}^2 \\ \varpi_4{}^2 \end{bmatrix}$$

Equations of motion

- Linear dynamics

F = [T,0,0] (body frame)

$$m\dot{v} = -mge_3 + RF \qquad \dot{\xi} = v$$

- Angular dynamics

$\Omega = [p,q,r]^T$

$$\mathbf{I}\dot{\Omega} = -\Omega \times \mathbf{I}\Omega + \tau \qquad \dot{R} = R\Omega_\times$$

2. State Space and System Modelling

34

State Space for Dynamical Systems

State: $\mathbf{x} = [x_1 \; x_2 \cdots x_n]^\mathsf{T}$

Robotic systems: The state typically includes the configuration (position) $\mathbf{q}$ and its derivatives $\dot{q}$, $\ddot{q}$ ..., that is:

$$\mathbf{q} = \begin{bmatrix} q_1 \\ q_2 \\ \vdots \\ q_j \end{bmatrix}, \mathbf{x} = \begin{bmatrix} \mathbf{q} \\ \dot{\mathbf{q}} \end{bmatrix}$$

The evolution of system's state over time is governed by a set of ordinary differential equations (ODEs).

ODEs are often expressed in their equivalent state-space form: $\dot{\mathbf{x}} = f(\mathbf{x}, \mathbf{u})$

Main Steps to Build a State- Space Model

Given an ODE (for now of a single variable, $y(t)$)

- Isolate the n^{th} highest derivative, $y^{(n)} = g(y, \dot{y}, \dots, y^{(n-1)}, \mathbf{u})$
- Set $x_1 = y(t), x_2 = \dot{y}(t), \dots, x_n = y^{(n-1)}(t)$
- Create state vector $\mathbf{x} = [x_1 \;\; x_2 \;\; \cdots \;\; x_n]^{\mathrm{T}} = \left[y \;\; \dot{y} \;\; \cdots \;\; y^{(n-1)}\right]^{\mathrm{T}}$
- Rewrite into a system of coupled first-order differential equations

$$\dot{x}_1 = \dot{y} = x_2$$

$$\dot{x}_2 = \ddot{y} = x_3$$

$$\dots$$

$$\dot{x}_n = y^{(n)} = g\left(y, \dot{y}, \dots, y^{(n-1)}, \mathbf{u}\right) = g(x_1, x_2, \dots, x_n, \mathbf{u})$$

Main Steps to Build a State- Space Model

- Rewrite in matrix form

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \vdots \\ \dot{x}_n \end{bmatrix} = \begin{bmatrix} x_2 \\ x_3 \\ \vdots \\ g(x_1, x_2, \ldots, x_n, \mathbf{u}) \end{bmatrix}$$

$$\dot{\mathbf{x}} = f(\mathbf{x}, \mathbf{u})$$

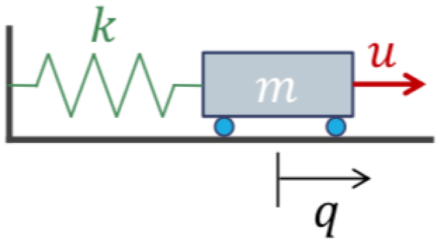
- Note: A system is linear time-invariant (LTI) when

$$\dot{\mathbf{x}} = A\mathbf{x} + B\mathbf{u},$$

where $\mathbf{x} \in \mathbb{R}^n$, $\mathbf{u} \in \mathbb{R}^m$, A is an $n \times n$ constant matrix, and B is an $n \times m$ constant matrix.

Example: Spring-Mass System

$$m\ddot{q}(t) + kq(t) = u(t)$$



Order of the system = 2

Rewrite $\ddot{q}(t) = \frac{u(t) - kq(t)}{m}$

State vector $\mathbf{x} = [x_1 \ x_2]^{\mathrm{T}} = [q \ \ \dot{q}]^{\mathrm{T}}$

Coupled equations $\dot{x}_1 = x_2, \dot{x}_2 = \frac{u - kx_1}{m}$

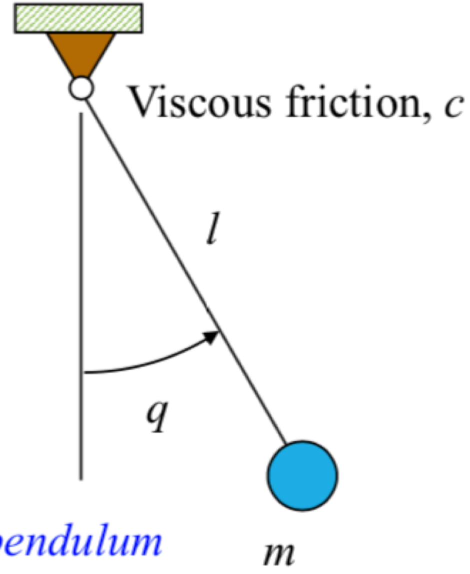
Then $\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} x_2 \\ \frac{u - kx_1}{m} \end{bmatrix}$, or $\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -\frac{k}{m} & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{1}{m} \end{bmatrix} u$

- Linear system

Example: Damped Pendulum

Equation of motion

$$\ddot{q} + \frac{c}{ml^2}\dot{q} + \frac{g}{l}\sin q = 0$$



State space representation

$$x = \begin{bmatrix} q \\ \dot{q} \end{bmatrix}, \dot{x} = \begin{bmatrix} x_2 \\ -\frac{g}{l}\sin x_1 - \frac{c}{ml^2}x_2 \end{bmatrix}$$

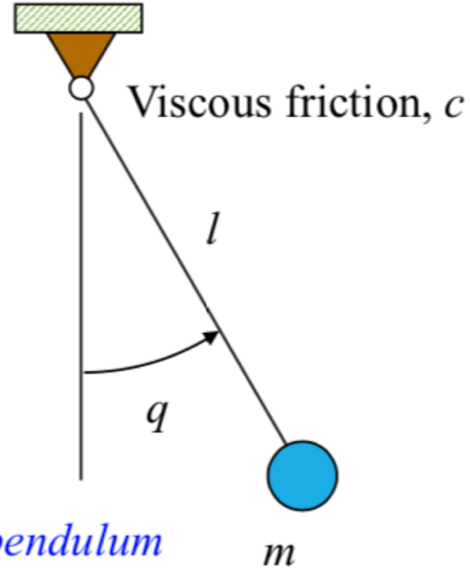
simple pendulum

- **Nonlinear** $\Longrightarrow$ <u>Linearize</u> around equilibria $\quad \dot{\mathbf{x}} = \boxed{f(\mathbf{x}) \equiv 0}$

Example: Damped Pendulum

Equation of motion

$$\ddot{q} + \frac{c}{ml^2}\dot{q} + \frac{g}{l}\sin q = 0$$



simple pendulum

State space representation

$$x = \begin{bmatrix} q \\ \dot{q} \end{bmatrix}, \dot{x} = \begin{bmatrix} x_2 \\ -\frac{g}{l}\sin x_1 - \frac{c}{ml^2}x_2 \end{bmatrix}$$

Equilibrium point(s)

$$x_{e,1} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, x_{e,2} = \begin{bmatrix} \pi \\ 0 \end{bmatrix}$$

Consider points near equilibria

$$\tilde{x} = \left(x - x_{e,i}\right)$$

Equilibria

Consider a system with n degrees of freedom

Let q_e be a configuration at static equilibrium ($\dot{\mathbf{x}} = f(\mathbf{x}) \equiv 0$)

$$x(t_0) = \begin{bmatrix} q_e \\ \hdashline 0 \end{bmatrix} \Rightarrow x(t > t_0) = \begin{bmatrix} q_e \\ \hdashline 0 \end{bmatrix}$$

- An equilibrium point can be
  - Stable
  - Unstable
  - Critically stable (or neutrally stable)
- We are interested in the behavior of the system around equilibrium points.
- We linearize the system around equilibria!

Linearization

- Given a nonlinear system $\dot{\mathbf{x}} = f(\mathbf{x}, \mathbf{u})$, $\mathbf{x}, f \in \mathbb{R}^n, \mathbf{u} \in \mathbb{R}^m$, derive an approximate linear system $\dot{\mathbf{x}} = A\mathbf{x} + B\mathbf{u}$ about an equilibrium point $(\mathbf{x}_e, \mathbf{u}_e)$.
- Taylor series expansion around equilibrium point:

$$f(\mathbf{x}_e + \Delta\mathbf{x}, \mathbf{u}_e + \Delta\mathbf{u}) = f(\mathbf{x}_e, \mathbf{u}_e) + \left[\frac{\partial f}{\partial \mathbf{x}}\right]_{(\mathbf{x}_e, \mathbf{u}_e)} \Delta\mathbf{x} + \left[\frac{\partial f}{\partial \mathbf{u}}\right]_{(\mathbf{x}_e, \mathbf{u}_e)} \Delta\mathbf{u} + \cancel{\mathrm{H.O.T.}}$$

$$\cancel{\dot{\mathbf{x}}} + \Delta\dot{\mathbf{x}} \approx f(\cancel{\mathbf{x}_e}, \mathbf{u}_e) + \left[\frac{\partial f}{\partial \mathbf{x}}\right]_{(\mathbf{x}_e, \mathbf{u}_e)} \Delta\mathbf{x} + \left[\frac{\partial f}{\partial \mathbf{u}}\right]_{(\mathbf{x}_e, \mathbf{u}_e)} \Delta\mathbf{u} \qquad \Longrightarrow \Delta\dot{\mathbf{x}} = A\Delta\mathbf{x} + B\Delta\mathbf{u}$$

Re-defining $\Delta\mathbf{x} \triangleq \mathbf{x}$, and $\Delta\mathbf{u} \triangleq \mathbf{u}$ yields $\dot{\mathbf{x}} = A\mathbf{x} + B\mathbf{u}$ with

$$A_{n\times n} = \left[\frac{\partial f}{\partial \mathbf{x}}\right]_{(\mathbf{x}_e, \mathbf{u}_e)} = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_n}{\partial x_1} & \cdots & \frac{\partial f_n}{\partial x_n} \end{bmatrix}_{(\mathbf{x}_e, \mathbf{u}_e)}, \quad B_{n\times m} = \left[\frac{\partial f}{\partial \mathbf{u}}\right]_{(\mathbf{x}_e, \mathbf{u}_e)} = \begin{bmatrix} \frac{\partial f_1}{\partial u_1} & \cdots & \frac{\partial f_1}{\partial u_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_n}{\partial u_1} & \cdots & \frac{\partial f_n}{\partial u_n} \end{bmatrix}_{(\mathbf{x}_e, \mathbf{u}_e)}$$

Back to the Pendulum

$$\dot{\mathbf{x}} = \begin{bmatrix} x_2 \\ -\frac{g}{l}\sin x_1 - \frac{c}{ml^2}x_2 \end{bmatrix} = f(\mathbf{x})$$

Equilibrium point 1

$$\mathbf{x}_{e,1} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

$$\dot{\mathbf{x}} \approx \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} \end{bmatrix}_{\mathbf{x}_{e,1}} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

$$= \begin{bmatrix} 0 & 1 \\ -\frac{g}{l} & -\frac{c}{ml^2} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

Equilibrium point 2

$$\mathbf{x}_{e,2} = \begin{bmatrix} \pi \\ 0 \end{bmatrix}$$

$$\dot{\mathbf{x}} \approx \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} \end{bmatrix}_{\mathbf{x}_{e,2}} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

$$= \begin{bmatrix} 0 & 1 \\ \frac{g}{l} & -\frac{c}{ml^2} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

Back to the Pendulum

Linear systems are stable iff the real parts of the eigenvalues are negative

Equilibrium point 1

$$\mathbf{x}_{e,1} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

$$\dot{\mathbf{x}} \approx \begin{bmatrix} 0 & 1 \\ -\frac{g}{l} & -\frac{c}{ml^2} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

$$\lambda = -\frac{c}{2ml^2} \pm \sqrt{\left(\frac{c}{2ml^2}\right)^2 - \frac{g}{l}}$$

Stable

Marginally stable if $c = 0$

Equilibrium point 2

$$\mathbf{x}_{e,2} = \begin{bmatrix} \pi \\ 0 \end{bmatrix}$$

$$\dot{\mathbf{x}} \approx \begin{bmatrix} 0 & 1 \\ \frac{g}{l} & -\frac{c}{ml^2} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

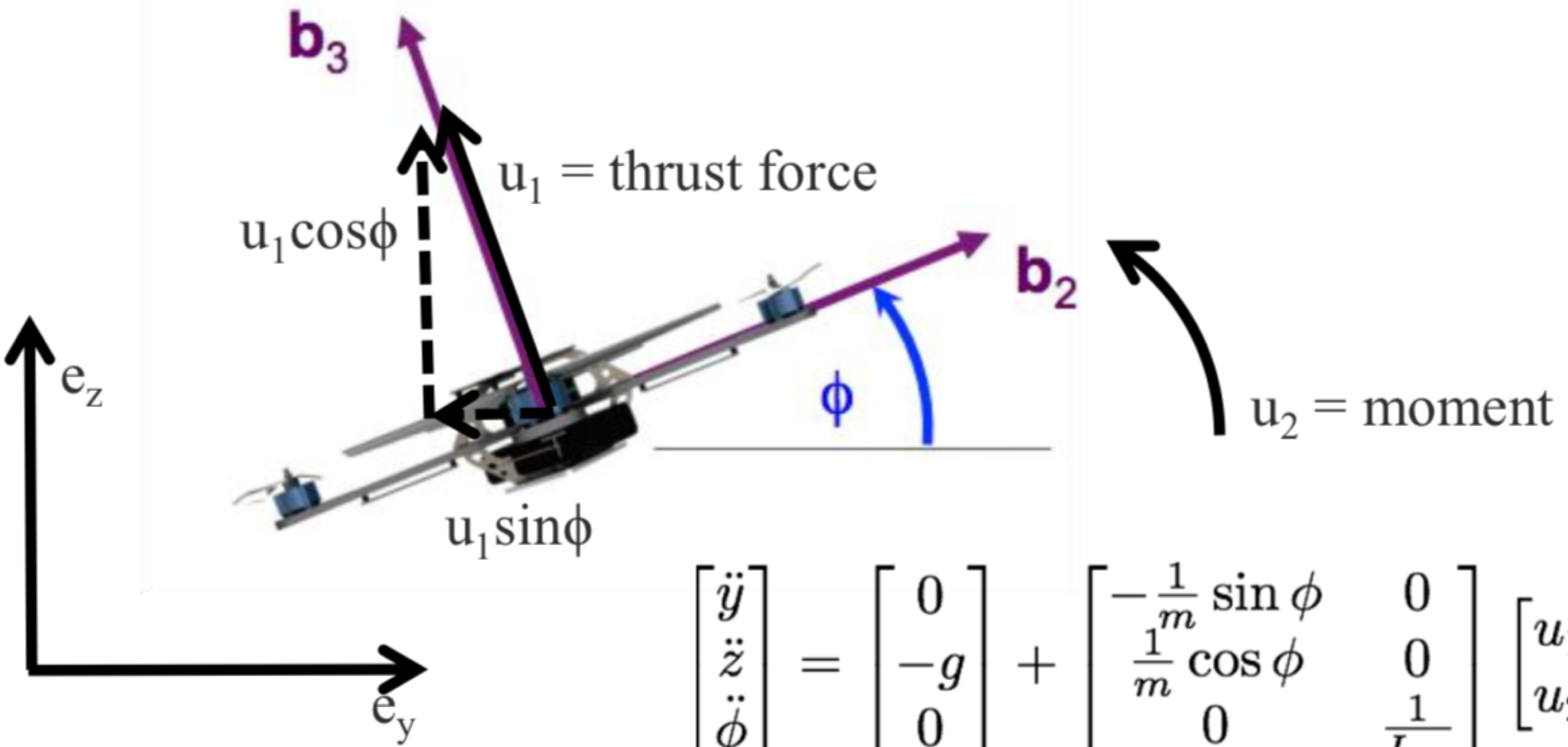
$$\lambda = -\frac{c}{2ml^2} \pm \sqrt{\left(\frac{c}{2ml^2}\right)^2 + \frac{g}{l}}$$

Unstable

3. Planar quadrotor

45

Planar Quadrotor

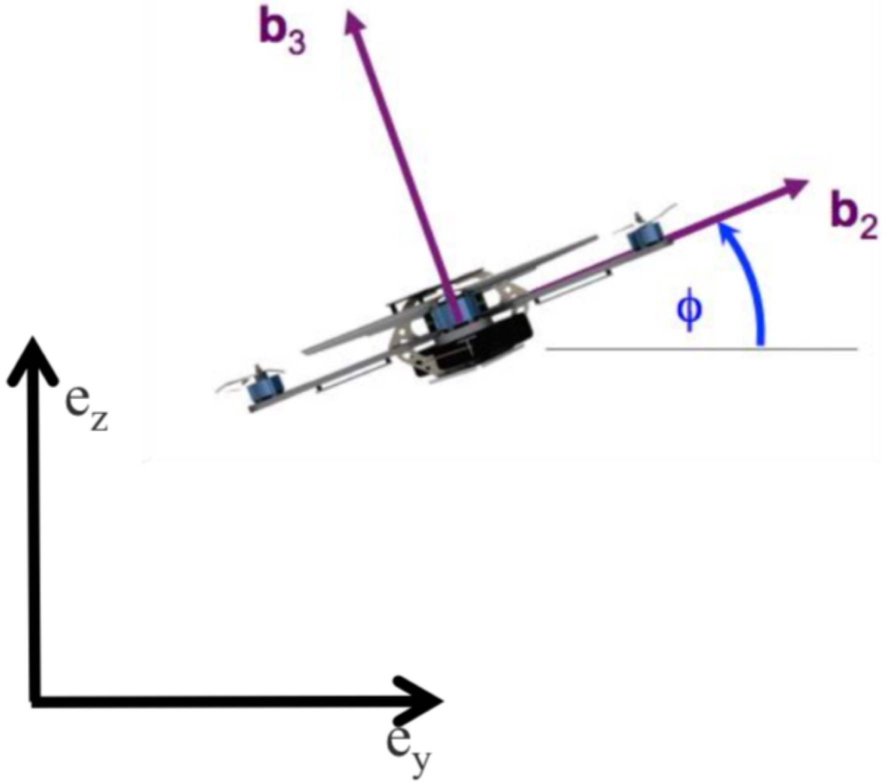


$$\begin{bmatrix} \ddot{y} \\ \ddot{z} \\ \ddot{\phi} \end{bmatrix} = \begin{bmatrix} 0 \\ -g \\ 0 \end{bmatrix} + \begin{bmatrix} -\frac{1}{m}\sin\phi & 0 \\ \frac{1}{m}\cos\phi & 0 \\ 0 & \frac{1}{I_{xx}} \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}$$

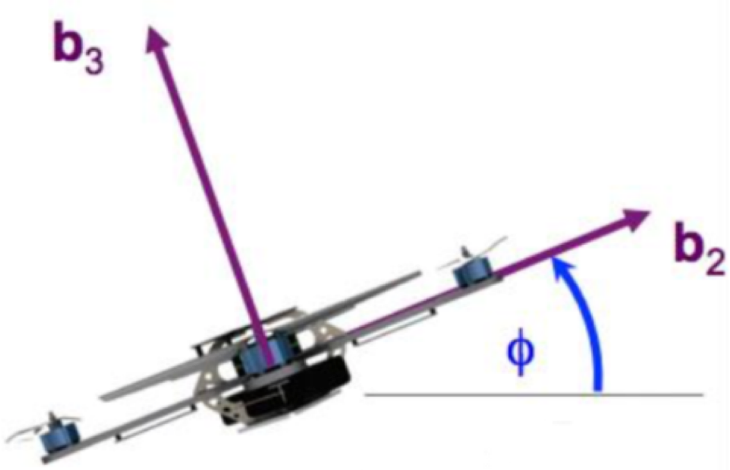
State Space

State vector

$$\mathbf{q} = \begin{bmatrix} y \\ z \\ \phi \end{bmatrix}, \mathbf{x} = \begin{bmatrix} \mathbf{q} \\ \dot{\mathbf{q}} \end{bmatrix}$$



Planar Quadrotor Model



$$\begin{bmatrix} \ddot{y} \\ \ddot{z} \\ \ddot{\phi} \end{bmatrix} = \begin{bmatrix} 0 \\ -g \\ 0 \end{bmatrix} + \begin{bmatrix} -\frac{1}{m}\sin\phi & 0 \\ \frac{1}{m}\cos\phi & 0 \\ 0 & \frac{1}{I_{xx}} \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}$$

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} y \\ z \\ \phi \\ \dot{y} \\ \dot{z} \\ \dot{\phi} \end{bmatrix}$$

$$\dot{\mathbf{x}} = \begin{bmatrix} x_4 \\ x_5 \\ x_6 \\ 0 \\ -g \\ 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ -m^{-1}\sin x_3 & 0 \\ m^{-1}\cos x_3 & 0 \\ 0 & I_{xx}^{-1} \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}$$

Linearized Dynamics

Nonlinear dynamics

$$\ddot{y} = -\frac{u_1}{m}\sin(\phi)$$

$$\ddot{z} = -g + \frac{u_1}{m}\cos(\phi)$$

$$\ddot{\phi} = \frac{u_2}{I_{xx}}$$

Equilibrium configuration

$$\mathbf{q}_e = \begin{bmatrix} y_0 \\ z_0 \\ 0 \end{bmatrix}, \mathbf{x}_e = \begin{bmatrix} \mathbf{q}_e \\ \mathbf{0} \end{bmatrix}$$

Equilibrium hover condition

$$y_0, z_0$$

$$\phi_0 = 0$$

$$u_{1,0} = mg,\ u_{2,0} = 0$$

Linearized model

$$\ddot{y} = -g\phi$$

$$\ddot{z} = -g + \frac{u_1}{m}$$

$$\ddot{\phi} = \frac{u_2}{I_{xx}}$$

4. 3D Quadrotor

Linearization and hover control

50

Control around hovering

Linearization

Pitch Roll Yaw

$$(u_1 \sim mg, \theta \sim 0, \phi \sim 0, \psi \sim \psi_0)$$

$$\cos(\theta) \sim 1, \cos(\phi) \sim 1$$

$$\sin(\theta) \sim \theta, \sin(\phi) \sim \phi$$

$$m\ddot{\mathbf{r}} = \begin{bmatrix} 0 \\ 0 \\ -mg \end{bmatrix} + R \begin{bmatrix} 0 \\ 0 \\ u_1 \end{bmatrix}$$

Thrust

$$\ddot{r}_1 = \ddot{x} = g(\theta \cos\psi + \phi \sin\psi)$$

$$\ddot{r}_2 = \ddot{y} = g(\theta \sin\psi - \phi \cos\psi)$$

Control around hovering

Linearization

$$p = q = r = 0$$

$$I\begin{bmatrix}\dot{p}\\ \dot{q}\\ \dot{r}\end{bmatrix} = \begin{bmatrix}u_2\\ u_3\\ u_4\end{bmatrix} - \begin{bmatrix}p\\ q\\ r\end{bmatrix} \times I\begin{bmatrix}p\\ q\\ r\end{bmatrix}$$

$$\begin{bmatrix}\dot{p}\\ \dot{q}\\ \dot{r}\end{bmatrix} = I^{-1}\begin{bmatrix}u_2\\ u_3\\ u_4\end{bmatrix}$$

$$u_1 = mg, u_2 = u_3 = u_4 = 0$$
$$\Rightarrow F_1 = F_2 = F_3 = F_4 = \frac{mg}{4}$$

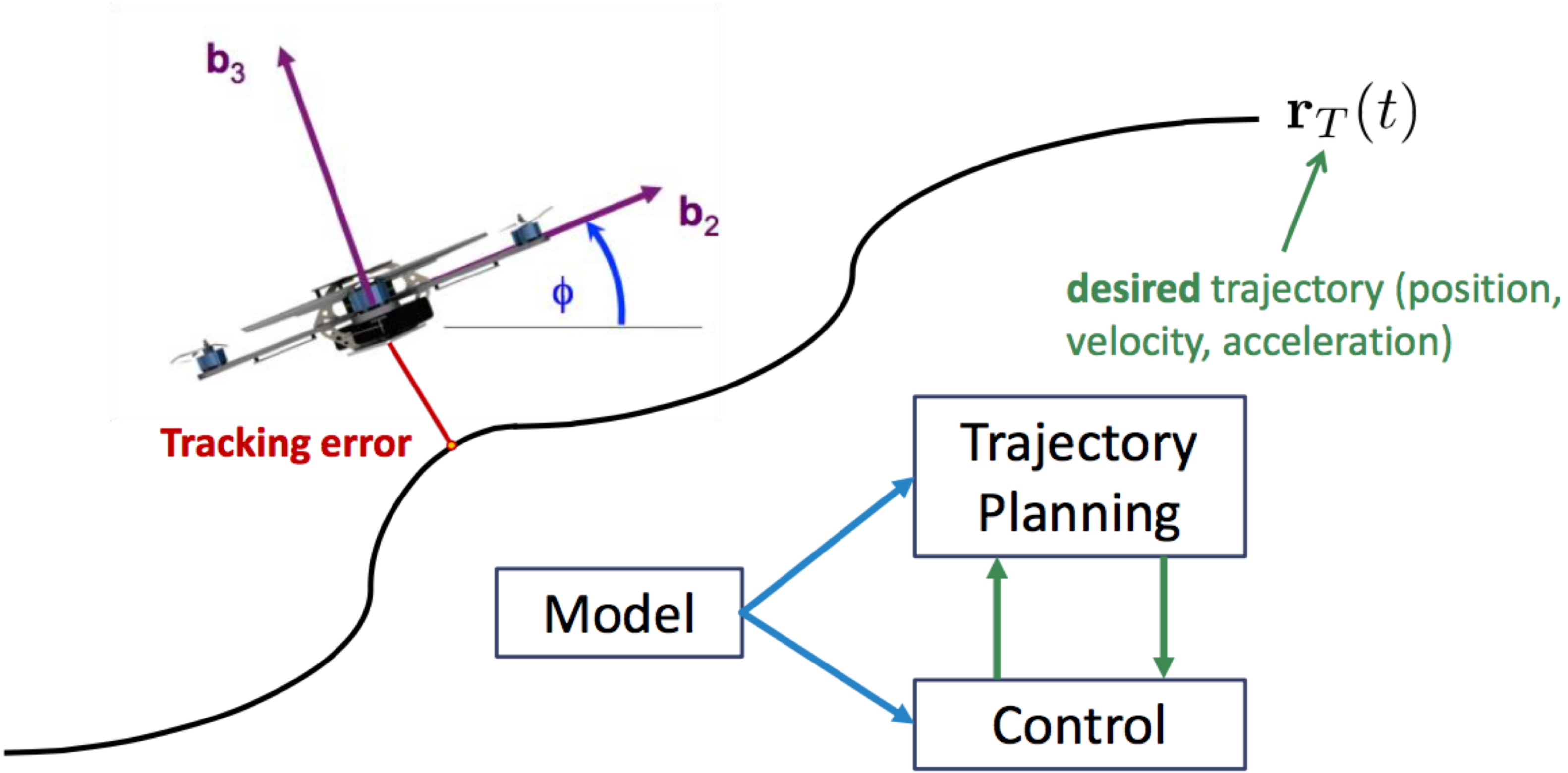
Control around hovering

Linearized quadrotor model around hovering

$$\ddot{r}_1 = \ddot{x} = g(\theta \cos\psi + \phi \sin\psi)$$
$$\ddot{r}_2 = \ddot{y} = g(\theta \sin\psi - \phi \cos\psi)$$
$$\ddot{r}_3 = \ddot{z} = -g + \frac{u_1}{m}$$

$$\begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} = I^{-1} \begin{bmatrix} u_2 \\ u_3 \\ u_4 \end{bmatrix} = I^{-1}\mathbf{u}_2$$

High-level picture



Controlling the quadrotor

Linear approach

- Linearise
- Design PID, LQR, MPC…

Nonlinear approach

- Geometric control
- Feedback linearisation
- Sliding mode control
- Nonlinear MPC
- ....

5. Linear control & tracking

56

Control of a Second Order System

Problem

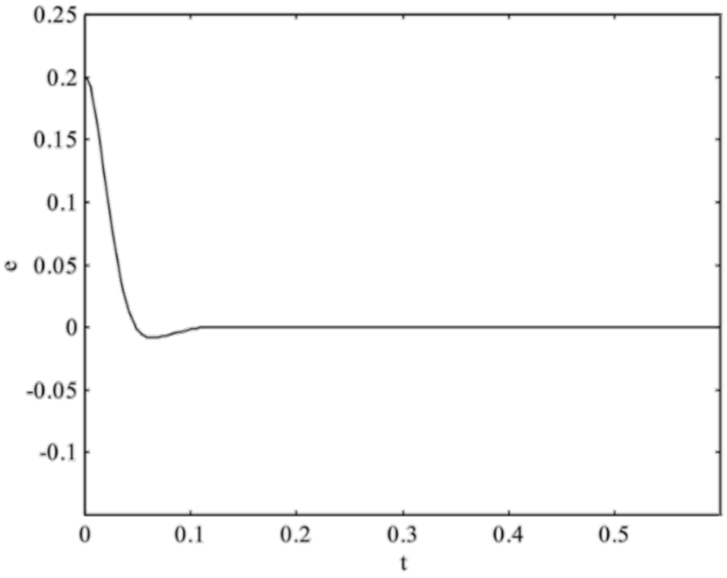
- State $\mathbf{x}$ and input $\mathbf{u}$
- Kinematic model $\ddot{\mathbf{x}} = \mathbf{u}$
- Want to follow trajectory $\mathbf{x}^{\text{des}}(t)$

General approach

- Define error $\mathbf{e}(t) = \mathbf{x}^{\text{des}}(t) - \mathbf{x}(t)$
- Want $\mathbf{e}(t)$ to converge exponentially to 0

Strategy

- Find $\mathbf{u}$ such that $\ddot{\mathbf{e}} + K_d\dot{\mathbf{e}} + K_p\mathbf{e} = 0$
- Pick some $K_p, K_d > 0$
- $\mathbf{u}(t) = \ddot{\mathbf{x}}^{\text{des}}(t) + K_d\dot{\mathbf{e}}(t) + K_p\mathbf{e}(t)$



Control for Trajectory Tracking

PD Control

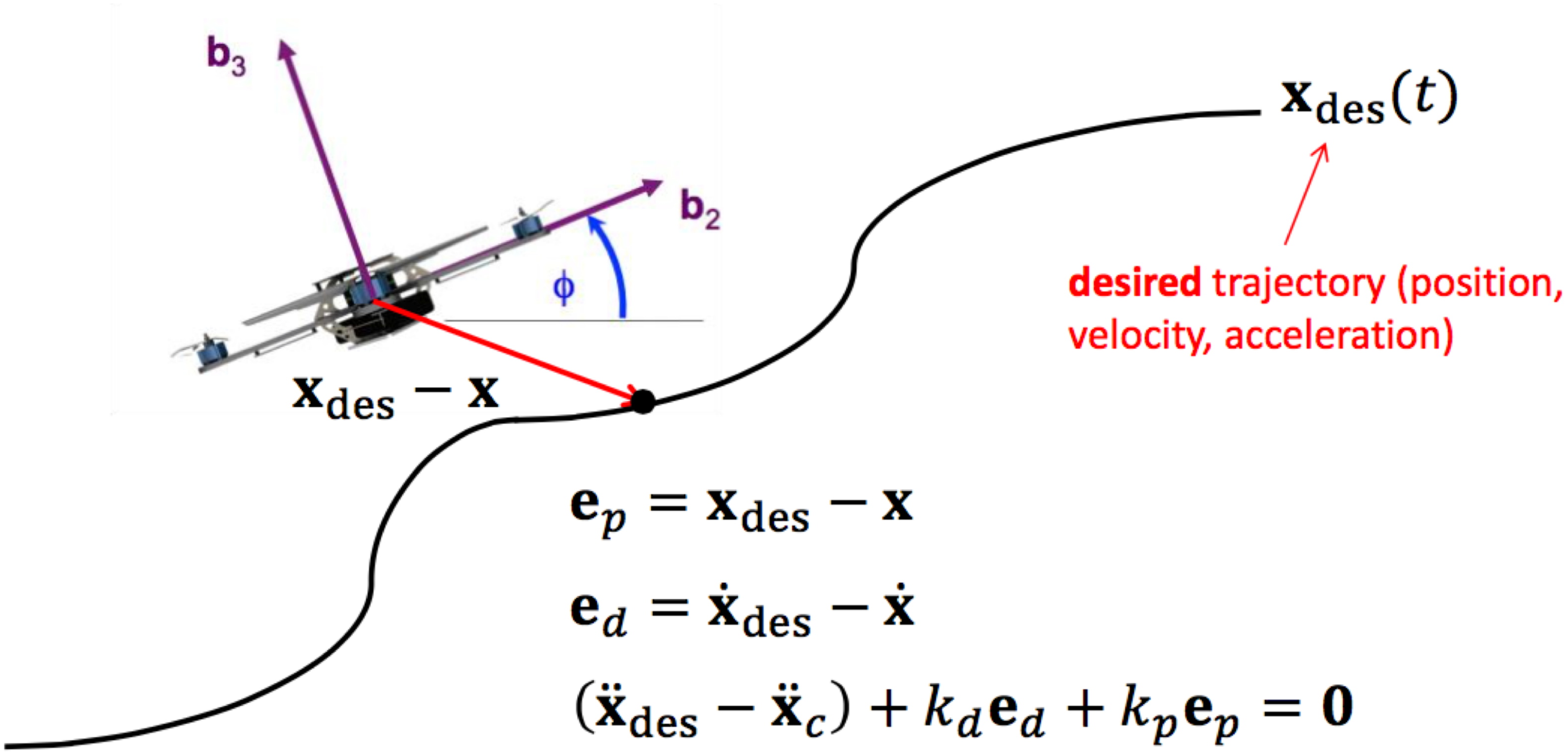
- $\mathbf{u}(t) = \ddot{\mathbf{x}}^{\text{des}}(t) + K_d \dot{\mathbf{e}}(t) + K_p \mathbf{e}(t)$
- Proportional term (K_p) has a spring (capacitance) response
- Derivative term (K_d) has a dashpot (resistance) response

PID Control

- $\mathbf{u}(t) = \ddot{\mathbf{x}}^{\text{des}}(t) + K_d \dot{\mathbf{e}}(t) + K_p \mathbf{e}(t) + K_I \int_0^t \mathbf{e}(\tau) d\tau$
- Integral term (K_I) makes steady state error go to 0
  - Accounts for model error or disturbances
- PID control generates a third-order closed-loop system

Trajectory tracking

Follow trajectory "exactly"



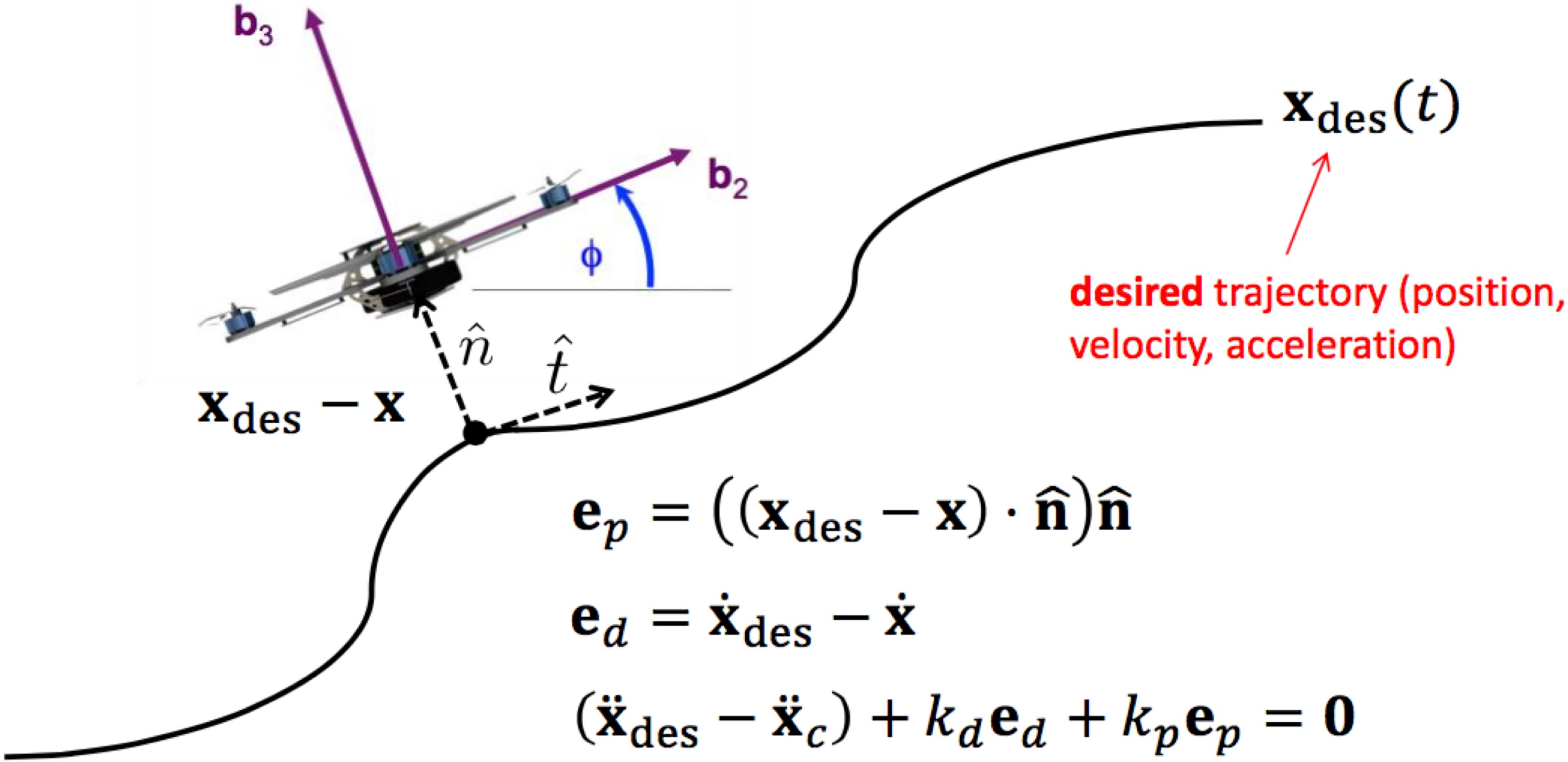
$$\mathbf{e}_p = \mathbf{x}_{\mathrm{des}} - \mathbf{x}$$

$$\mathbf{e}_d = \dot{\mathbf{x}}_{\mathrm{des}} - \dot{\mathbf{x}}$$

$$(\ddot{\mathbf{x}}_{\mathrm{des}} - \ddot{\mathbf{x}}_c) + k_d \mathbf{e}_d + k_p \mathbf{e}_p = \mathbf{0}$$

Trajectory tracking

Follow closest point

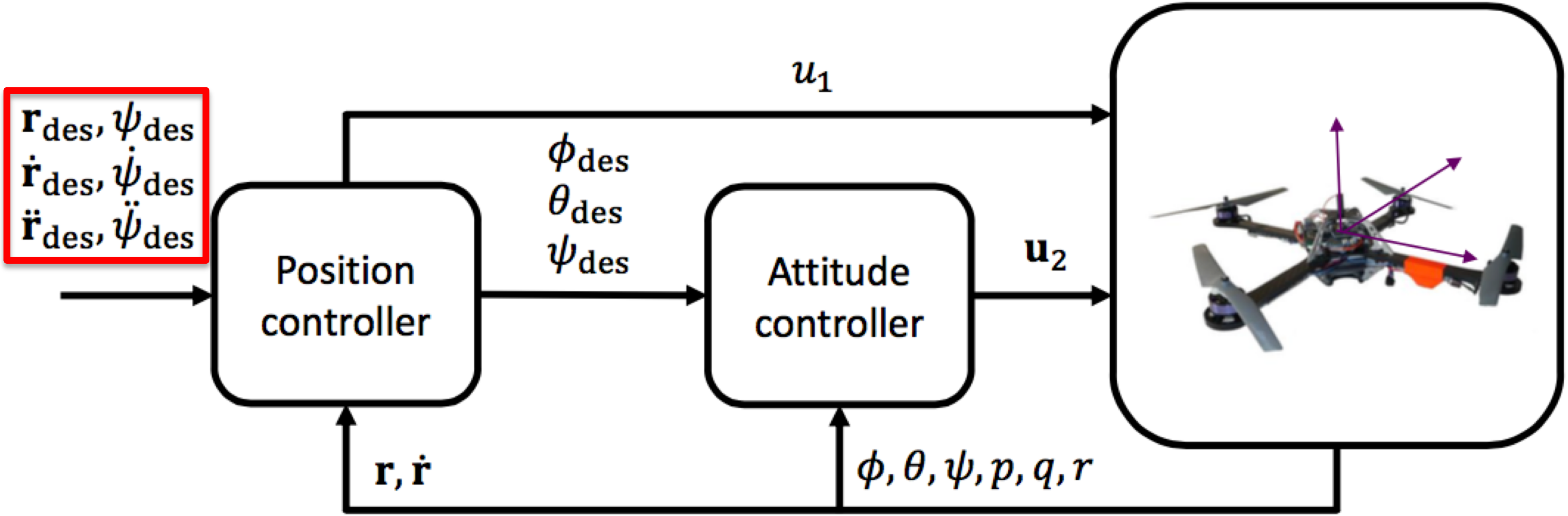


$$\mathbf{e}_p = \left(\left(\mathbf{x}_{\mathrm{des}} - \mathbf{x}\right) \cdot \hat{\mathbf{n}}\right)\hat{\mathbf{n}}$$

$$\mathbf{e}_d = \dot{\mathbf{x}}_{\mathrm{des}} - \dot{\mathbf{x}}$$

$$(\ddot{\mathbf{x}}_{\mathrm{des}} - \ddot{\mathbf{x}}_c) + k_d \mathbf{e}_d + k_p \mathbf{e}_p = \mathbf{0}$$

Hierarchical control



3D controller (linearized hovering model)

Proportional + Derivative error terms

commanded *actual* (feedback)

$$(\ddot{r}_{i,des} - \ddot{r}_{i,c}) + k_{d,i}(\dot{r}_{i,des} - \dot{r}_i) + k_{p,i}(r_{i,des} - r_i) = 0$$

specified

Linearized model:

$$\ddot{r}_1 = \ddot{x} = g(\theta \cos\psi + \phi \sin\psi)$$
$$\ddot{r}_2 = \ddot{y} = g(\theta \sin\psi - \phi \cos\psi)$$
$$\ddot{r}_3 = \ddot{z} = -g + \frac{u_1}{m}$$

$$\ddot{r}_{i,c}$$

3D controller (linearized hovering model)

From substitution:

$$\ddot{r}_1 = \ddot{x} = g(\theta \cos\psi + \phi \sin\psi)$$
$$\ddot{r}_2 = \ddot{y} = g(\theta \sin\psi - \phi \cos\psi)$$
$$\ddot{r}_3 = \ddot{z} = -g + \frac{u_1}{m}$$

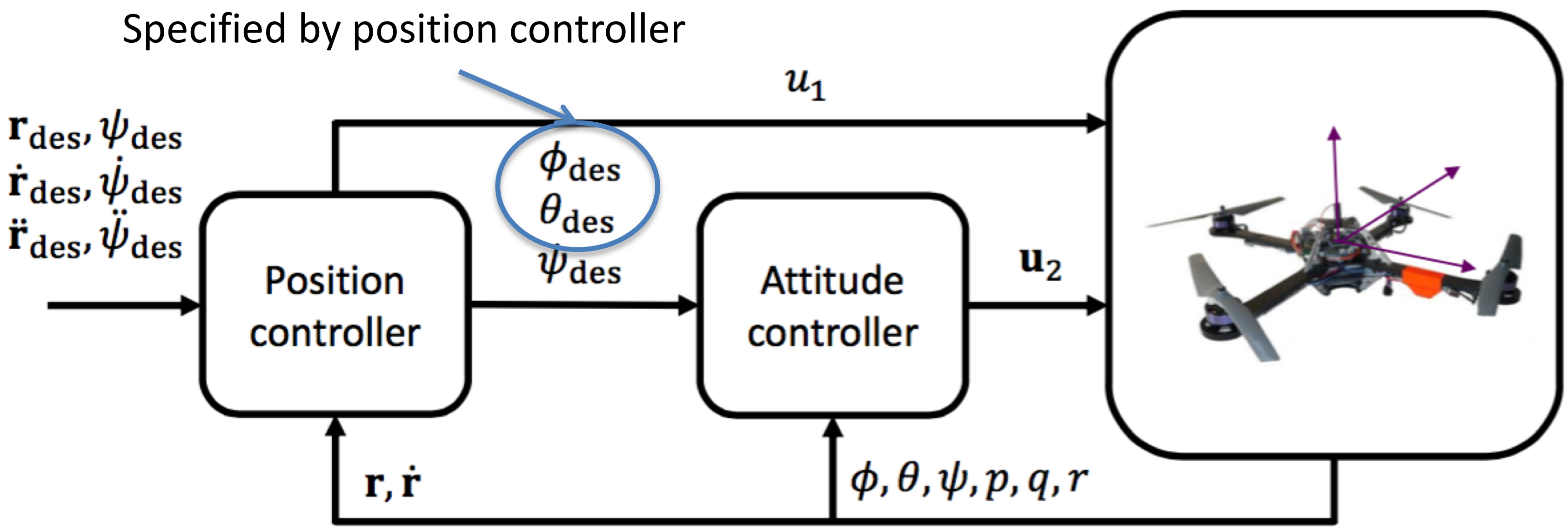
$$\leftarrow \quad \ddot{r}_{i,c}$$

$$u_1 = m(g + \ddot{r}_{3,c})$$

$$\phi_c = \frac{1}{g}(\ddot{r}_{1,c} \sin\psi_{des} - \ddot{r}_{2,c} \cos\psi_{des})$$

$$\theta_c = \frac{1}{g}(\ddot{r}_{1,c} \cos\psi_{des} + \ddot{r}_{2,c} \sin\psi_{des})$$

Hierarchical control



3D controller (linearized hovering model)

$$\phi_c = \frac{1}{g}(\ddot{r}_{1,c}\sin\psi_{des} - \ddot{r}_{2,c}\cos\psi_{des})$$

$$\theta_c = \frac{1}{g}(\ddot{r}_{1,c}\cos\psi_{des} + \ddot{r}_{2,c}\sin\psi_{des})$$

$$\mathbf{u}_2 = I\begin{bmatrix} k_{d,\phi}(p_{\text{des}} - p) + k_{p,\phi}(\phi_{\text{des}} - \phi) \\ k_{d,\phi}(q_{\text{des}} - q) + k_{p,\phi}(\theta_{\text{des}} - \theta) \\ k_{d,\phi}(r_{\text{des}} - r) + k_{p,\phi}(\psi_{\text{des}} - \psi) \end{bmatrix}$$

$$\begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} = I^{-1}\begin{bmatrix} u_2 \\ u_3 \\ u_4 \end{bmatrix} = I^{-1}\mathbf{u}_2$$

Summary

Model and equations of motion of a quadrotor

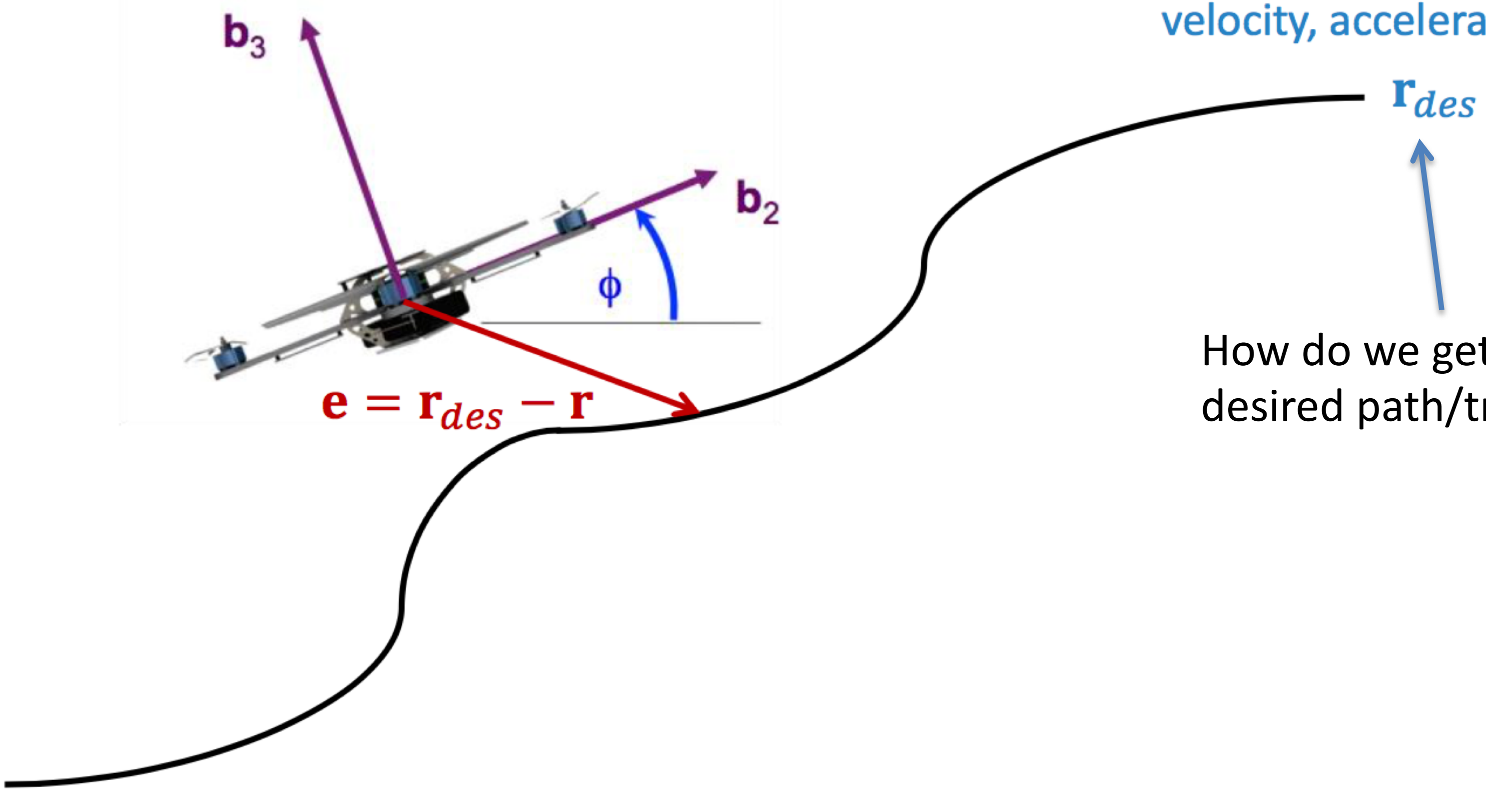
These controllers and equations can be used for:

- Attitude control (3D orientation of the drone)
- Position control (3D position and speed of the drone)
- Path and trajectory following

Last lecture: (idem) for ground wheeled robots

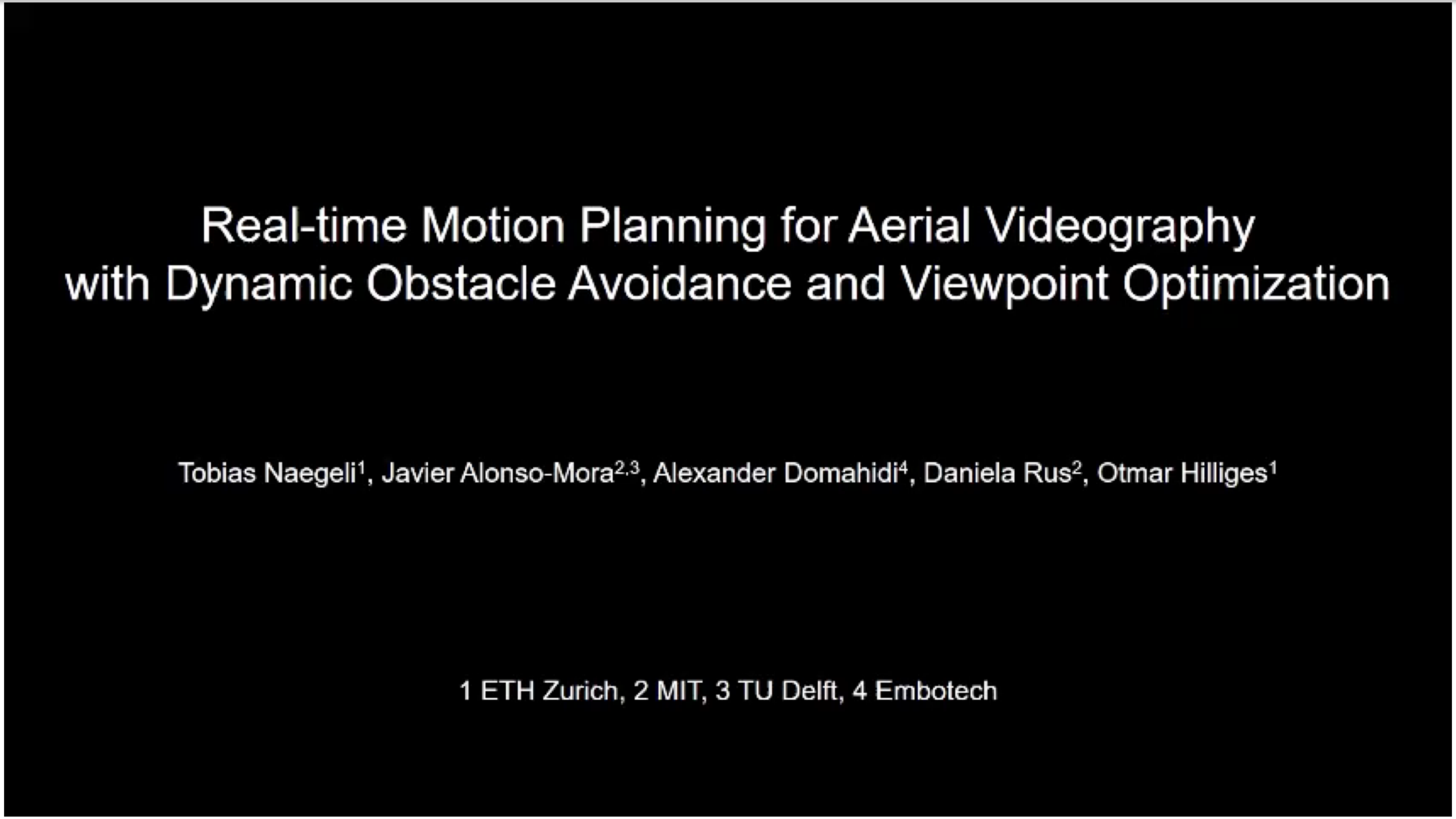
Next lectures: Path and trajectory generation

desired trajectory (position, velocity, acceleration)



How do we get this desired path/trajectory?

Trajectory optimization



T. Naegeli, J. Alonso-Mora, A. Domahidi, D. Rus, O. Hilliges, "Real-time Motion Planning for Aerial Videography with Dynamic Obstacle Avoidance and Viewpoint Optimization", in IEEE Robotics and Automation Letters, January 2017. Video: https://youtu.be/dh7yOEHBc_w

Exercises

Exercise 1:

- Due date on Monday!

Exercise 2:

- Now online

Deadline exercise 1: Kinematics and closed loop control [Mo. 24.02 EOD]

Project

Monday in a week and a half is the deadline for the intermediate report of the project (which will not be graded, but you will receive feedback).

Our recommendation is that the intermediate report (follow the given template in brightspace) contains at least:

- Introduction defining the project, task to be achieved and chosen motion planner (and why).
- Kinematics/Dynamic model of the robot
- Equations of motion
- Chosen planner / controller
- Planned simulation environment
- Planned scenarios for evaluation of the planner

The intermediate project reports should be submitted in print in J. Alonso Mora's mailbox at the end of the F-2 corridor or handed in to me during the lecture on 11th March.

Questions?