Multirotor helicopter dynamics modelling and control

B(E)3M33MRS — Aerial Multi-Robot Systems

Ing. Tomáš Báča, Ph.D.

Multi-Robot Systems group, Faculty of Electrical Engineering
Czech Technical University in Prague

FACULTY OF ELECTRICAL ENGINEERING CTU IN PRAGUE | MRS MULTI-ROBOT SYSTEMS GROUP

Lecture 2: UAV Control

2024-09-29

Multirotor helicopter dynamics modelling and control
B(E)3M33MRS — Aerial Multi-Robot Systems
Ing. Tomáš Báča, Ph.D.
Multi-Robot Systems group, Faculty of Electrical Engineering
Czech Technical University in Prague

Multirotor helicopters

Under-actuated fixed-tilt multirotors

Figure 1: Commercial fixed-tilt quadrotor: DJI Mavic Pro.

- **Controllable DOFs**: 4 (3 position, 1 rotation).
- Common; commercially viable (since 2010).
- Balance between capabilities and mechanical simplicity.
- Differentially-flat system

Fully actuated multirotors [1]

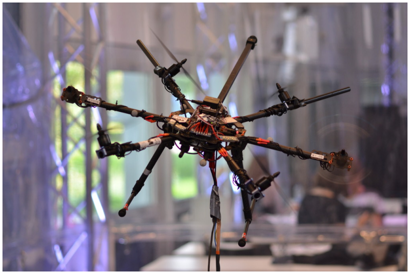
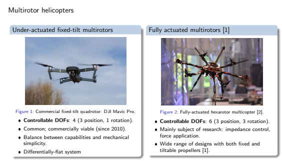


Figure 2: Fully-actuated hexarotor multicopter [2].

- **Controllable DOFs**: 6 (3 position, 3 rotation).
- Mainly subject of research: impedance control, force application.
- Wide range of designs with both fixed and tiltable propellers [1].



- We will focus on the underactuated platforms for their *mainstream* spread.

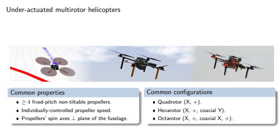
Common properties

- ≥ 4 fixed-pitch non-tiltable propellers.
- Individually-controlled propeller speed.
- Propellers' spin axes $\perp$ plane of the fuselage.

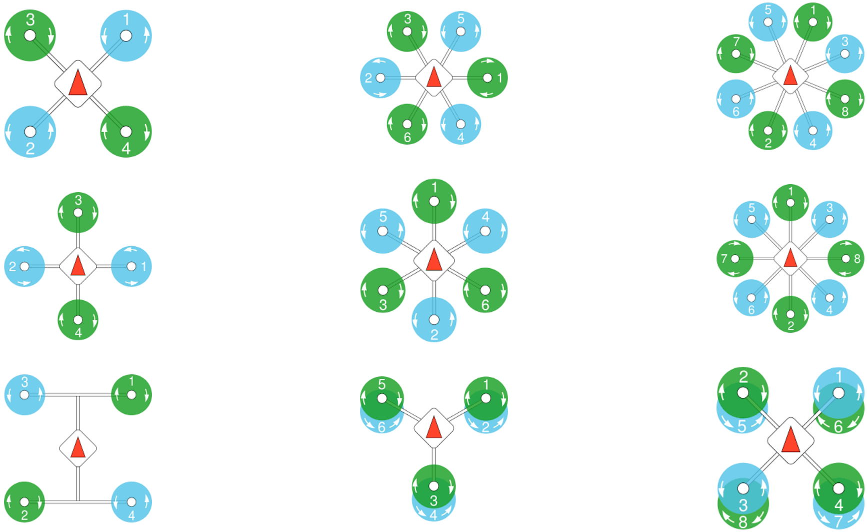
Common configurations

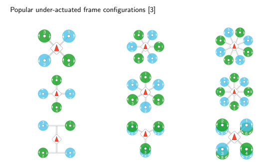
- Quadrotor (X, $+$).
- Hexarotor (X, $+$, coaxial Y).
- Octarotor (X, $+$, coaxial X, $+$).

2024-09-29
Lecture 2: UAV Control
Dynamics model
Input mapping
Under-actuated multirotor helicopters



Popular under-actuated frame configurations [3]





- Common frame configuration, as define by the PX4 flight controller [3].
- Other flight controller systems' definition might differ!
- We will focus on modelling the Quad-X configuration.
- The systems become robust to motor/propeller outage at ≥ 8 propellers, without scarifying controllable DOFs.
- With special control scheme and while scarifying output DOFs, UAVs can fly even with just 1 propeller.

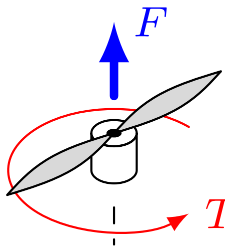
Single propeller model

Propeller thrust model (simplified)

Thrust is produced due to the propeller's lift.

$$F \approx k\omega^2 \tag{1}$$

- F: thrust force [N].
- k: linear coefficient [-].
- ω: propeller angular velocity [rad s^{-1}].



Propeller torque model (simplified)

Torque is produced due to the propeller's drag.

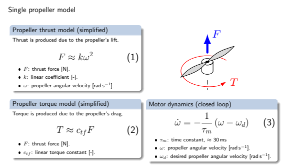
$$T \approx c_{tf}F \tag{2}$$

- F: thrust force [N].
- c_{tf}: linear torque constant [-].

Motor dynamics (closed loop)

$$\dot{\omega} = -\frac{1}{\tau_m}\left(\omega - \omega_d\right) \tag{3}$$

- τ_m: time constant, $\approx$ 30 ms
- ω: propeller angular velocity [rad s^{-1}].
- ω_d: desired propeller angular velocity [rad s^{-1}].



- This thrust model is extremely simplified, however, works quite well for slowly-flying multirotor UAVs.
- k and c_{tf} can be identified experimentally (test stand) or using numerical simulations.
- k and c_{tf} depend on the propeller's blade shape:
  - The propeller's thrust (k) is grows with propeller's pitch angle.
  - The propeller's drag (c_{tf}) diminishes with more aero-dynamic wing-shaped propellers.
- More accurate models: Blade Momentum Theory.

- The thrust curve and other parameters can be measured on a stand.
- However, the final properties can be different when mounted on the UAV frame.
- Rely on previous work of others, e.g., MRS motor tests at `https://ctu-mrs.github.io/docs/hardware/motor_tests.html`.

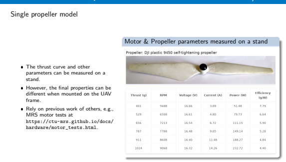
Motor & Propeller parameters measured on a stand

Propeller: DJI plastic 9450 self-tightening propeller

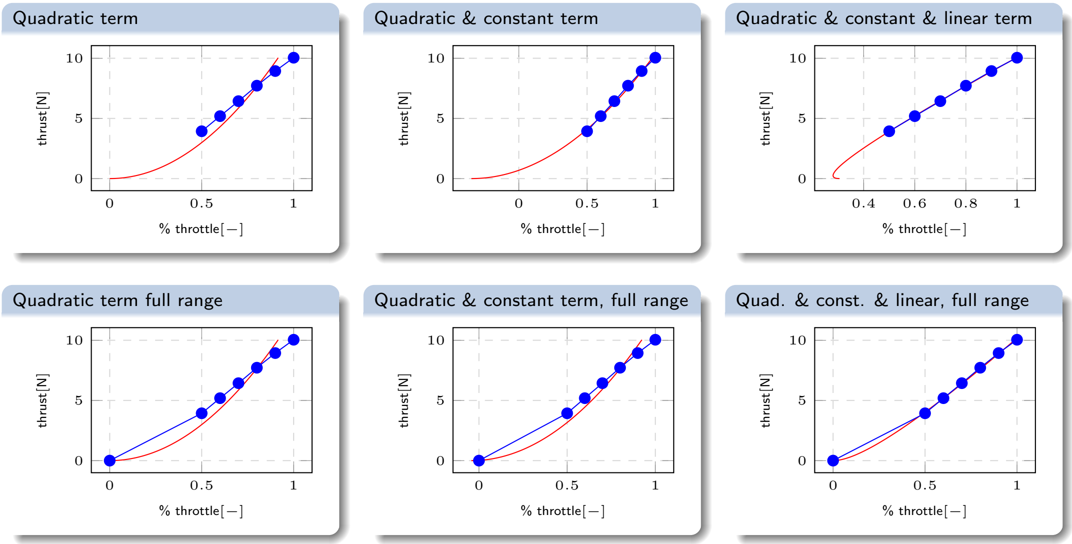
Thrust (g)	RPM	Voltage (V)	Current (A)	Power (W)	Efficiency (g/W)
401	5688	16.66	3.09	51.48	7.79
529	6508	16.61	4.80	79.73	6.64
656	7213	16.54	6.72	111.15	5.90
787	7788	16.48	9.05	149.14	5.28
911	8608	16.40	11.48	188.27	4.84
1024	9068	16.32	14.26	232.72	4.40

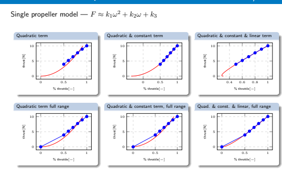
Lecture 2: UAV Control
└─Dynamics model
  └─Input mapping
    └─Single propeller model

2024-09-29



Single propeller model — $F \approx k_1\omega^2 + k_2\omega + k_3$



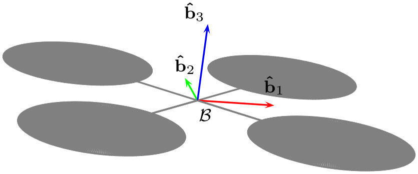


- Sometimes, more precise thrust model than $F \approx k\omega^2$ is needed.
- Use the best model as you can for agile flight with high relying on feedforward.
- Use appropriate function for an appropriate range of values: the operation range.
- - More parameters requires better estimation and fitting.
- - More parameter requires more complex function, which is more difficult to work with.
- - Beware of overfitting.

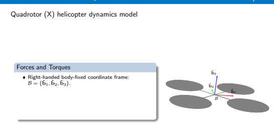
Quadrotor (X) helicopter dynamics model

Forces and Torques

- Right-handed body-fixed coordinate frame: $\mathcal{B} = \{\hat{\mathbf{b}}_1, \hat{\mathbf{b}}_2, \hat{\mathbf{b}}_3\}$.

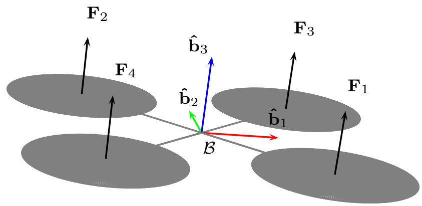


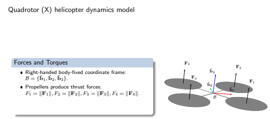
Lecture 2: UAV Control
Dynamics model
Input mapping
Quadrotor (X) helicopter dynamics model
2024-09-29



Forces and Torques

- Right-handed body-fixed coordinate frame: $\mathcal{B} = \{\hat{\mathbf{b}}_1, \hat{\mathbf{b}}_2, \hat{\mathbf{b}}_3\}$.
- Propellers produce thrust forces: $F_1 = \|\mathbf{F}_1\|, F_2 = \|\mathbf{F}_2\|, F_3 = \|\mathbf{F}_3\|, F_4 = \|\mathbf{F}_4\|$.

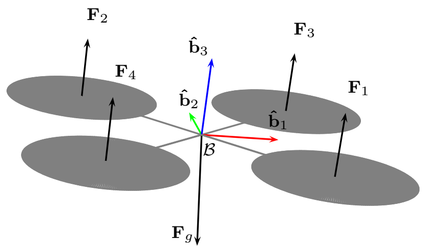


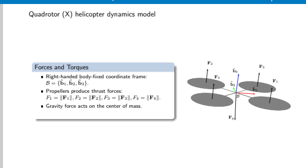


- The propeller forces $\mathbf{F}_1, \mathbf{F}_2, \mathbf{F}_3, \mathbf{F}_4$ add up to the total force vector $\mathbf{F}_t$. The direction of the force vector along $\mathbf{b}_3$ thanks to the planar configuration of the frame. This creates our **1st degree of freedom**: the magnitude of the collective thrust vector.

Forces and Torques

- Right-handed body-fixed coordinate frame: $\mathcal{B} = \{\hat{\mathbf{b}}_1, \hat{\mathbf{b}}_2, \hat{\mathbf{b}}_3\}$.
- Propellers produce thrust forces: $F_1 = \|\mathbf{F}_1\|, F_2 = \|\mathbf{F}_2\|, F_3 = \|\mathbf{F}_3\|, F_4 = \|\mathbf{F}_4\|$.
- Gravity force acts on the center of mass.



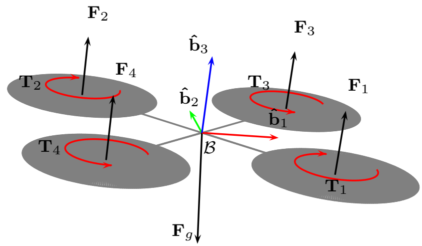


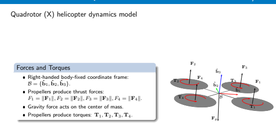
- The propeller forces $\mathbf{F}_1, \mathbf{F}_2, \mathbf{F}_3, \mathbf{F}_4$ add up to the total force vector $\mathbf{F}_t$. The direction of the force vector along $\mathbf{b}_3$ thanks to the planar configuration of the frame. This creates our **1st degree of freedom**: the magnitude of the collective thrust vector.

Quadrotor (X) helicopter dynamics model

Forces and Torques

- Right-handed body-fixed coordinate frame: $\mathcal{B} = \{\hat{\mathbf{b}}_1, \hat{\mathbf{b}}_2, \hat{\mathbf{b}}_3\}$.
- Propellers produce thrust forces: $F_1 = \|\mathbf{F}_1\|, F_2 = \|\mathbf{F}_2\|, F_3 = \|\mathbf{F}_3\|, F_4 = \|\mathbf{F}_4\|$.
- Gravity force acts on the center of mass.
- Propellers produce torques: $\mathbf{T}_1, \mathbf{T}_2, \mathbf{T}_3, \mathbf{T}_4$.



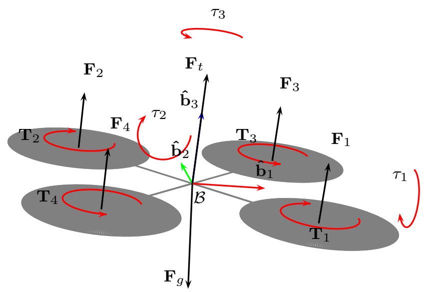


- The propeller forces $\mathbf{F}_1, \mathbf{F}_2, \mathbf{F}_3, \mathbf{F}_4$ add up to the total force vector $\mathbf{F}_t$. The direction of the force vector along $\mathbf{b}_3$ thanks to the planar configuration of the frame. This creates our **1st degree of freedom**: the magnitude of the collective thrust vector.
- The differential thrust creates torques τ_1, τ_2. These are our **2nd and 3rd** degree of freedom, which can be used to point the thrust vector in 3D.
- The propeller torques $\mathbf{T}_1, \mathbf{T}_2, \mathbf{T}_3, \mathbf{T}_4$ add up to one torque τ_3 due to the planar configuration of the propellers. This creates our last and **4th degree of freedom**, the rotation of the body around the $\mathbf{b}_3$ axis.
- The τ_3 has the worst control authority of all the DOFs. This is because it produced solely by the *parasitic* drag of the propellers. This DOF can become uncontrollable in extreme cases, e.g., with highly-efficient propellers.

Quadrotor (X) helicopter dynamics model

Forces and Torques

- Right-handed body-fixed coordinate frame: $\mathcal{B} = \{\hat{\mathbf{b}}_1, \hat{\mathbf{b}}_2, \hat{\mathbf{b}}_3\}$.
- Propellers produce thrust forces: $F_1 = \|\mathbf{F}_1\|, F_2 = \|\mathbf{F}_2\|, F_3 = \|\mathbf{F}_3\|, F_4 = \|\mathbf{F}_4\|$.
- Gravity force acts on the center of mass.
- Propellers produce torques: $\mathbf{T}_1, \mathbf{T}_2, \mathbf{T}_3, \mathbf{T}_4$.



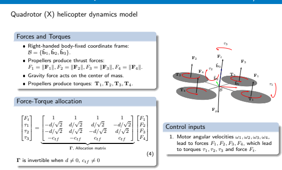
Force-Torque allocation

$$\begin{bmatrix} F_t \\ \tau_1 \\ \tau_2 \\ \tau_3 \end{bmatrix} = \underbrace{\begin{bmatrix} 1 & 1 & 1 & 1 \\ -d/\sqrt{2} & d/\sqrt{2} & d/\sqrt{2} & -d/\sqrt{2} \\ -d/\sqrt{2} & d/\sqrt{2} & -d/\sqrt{2} & d/\sqrt{2} \\ -c_{tf} & -c_{tf} & c_{tf} & c_{tf} \end{bmatrix}}_{\mathbf{\Gamma}\text{, Allocation matrix}} \begin{bmatrix} F_1 \\ F_2 \\ F_3 \\ F_4 \end{bmatrix} \tag{4}$$

$\mathbf{\Gamma}$ is invertible when $d \neq 0$, $c_{tf} \neq 0$

Control inputs

1. Motor angular velocities $\omega_1, \omega_2, \omega_3, \omega_4$, lead to forces F_1, F_2, F_3, F_4, which lead to torques τ_1, τ_2, τ_3 and force F_t.



- The propeller forces $\mathbf{F}_1, \mathbf{F}_2, \mathbf{F}_3, \mathbf{F}_4$ add up to the total force vector $\mathbf{F}_t$. The direction of the force vector along $\mathbf{b}_3$ thanks to the planar configuration of the frame. This creates our **1st degree of freedom**: the magnitude of the collective thrust vector.
- The differential thrust creates torques τ_1, τ_2. These are our **2nd and 3rd** degree of freedom, which can be used to point the thrust vector in 3D.
- The propeller torques $\mathbf{T}_1, \mathbf{T}_2, \mathbf{T}_3, \mathbf{T}_4$ add up to one torque τ_3 due to the planar configuration of the propellers. This creates our last and **4th degree of freedom**, the rotation of the body around the $\mathbf{b}_3$ axis.
- The τ_3 has the worst control authority of all the DOFs. This is because it produced solely by the *parasitic* drag of the propellers. This DOF can become uncontrollable in extreme cases, e.g., with highly-efficient propellers.

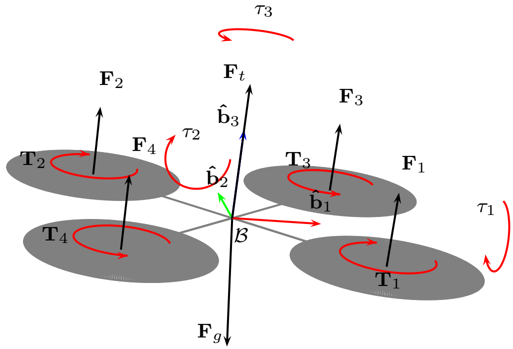
Quadrotor (X) helicopter dynamics model

Control input mapping

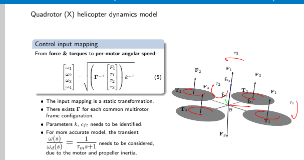
From **force & torques** to **per-motor angular speed**:

$$\begin{bmatrix}\omega_1\\ \omega_2\\ \omega_3\\ \omega_4\end{bmatrix} = \sqrt{\left(\mathbf{\Gamma}^{-1}\begin{bmatrix}F_t\\ \tau_1\\ \tau_2\\ \tau_3\end{bmatrix}\right)k^{-1}} \tag{5}$$

- The input mapping is a static transformation.
- There exists $\mathbf{\Gamma}$ for each common multirotor frame configuration.
- Parameters k, c_{ft} needs to be identified.
- For more accurate model, the transient $\dfrac{\omega(s)}{\omega_d(s)} = \dfrac{1}{\tau_m s+1}$ needs to be considered, due to the motor and propeller inertia.



2024-09-29



- Now we substitute the forces in (4) with motor angular velocities using (1), we can then express the motor angular velocities as a function of the collective force F_t and the torques τ_1, τ_2, τ_3.

Intermezzo — Representing rigid body rotation in 3D

Rigid body rotation: two meanings

1. How to rotate vectors from $\mathcal{W}$ to $\mathcal{B}$.
2. How to rotate vectors from $\mathcal{B}$ to $\mathcal{W}$.

Rotation matrix

$$\mathbf{R} \in \mathbb{R}^{3\times 3}, \tag{6}$$

such that

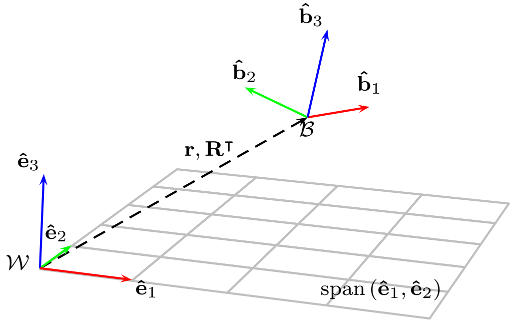
$$\mathbf{R}^\intercal\mathbf{R} = \mathbf{R}\mathbf{R}^\intercal = \mathbf{I} \tag{7}$$

- rotation matrices are **orthogonal**,
- such matrices $\in$ *O(3)* group,

and, such that,

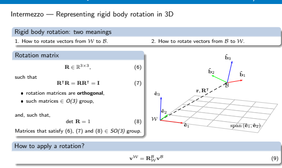
$$\det\ \mathbf{R} = 1 \tag{8}$$

Matrices that satisfy (6), (7) and (8) $\in$ *SO(3)* group.



How to apply a rotation?

$$\mathbf{v}^{\mathcal{W}} = \mathbf{R}_{\mathcal{W}}^{\mathcal{B}}\mathbf{v}^{\mathcal{B}} \tag{9}$$



- Orthogonal matrices have orthonormal column vectors.

Intermezzo — Representing rigid body rotation in 3D

Rigid body rotation: two meanings

1. How to rotate vectors from $\mathcal{W}$ to $\mathcal{B}$.
2. **How to rotate vectors from $\mathcal{B}$ to $\mathcal{W}$.**

Rotation matrix

$$\mathbf{R} \in \mathbb{R}^{3\times3}, \tag{6}$$

such that

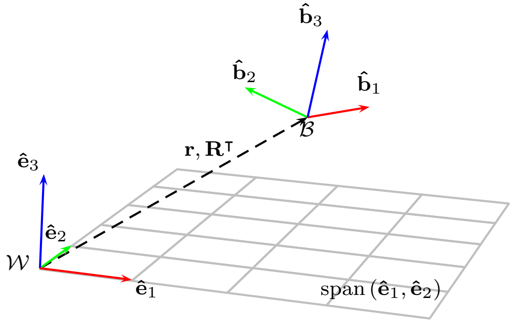
$$\mathbf{R}^\intercal\mathbf{R} = \mathbf{R}\mathbf{R}^\intercal = \mathbf{I} \tag{7}$$

- rotation matrices are **orthogonal**,
- such matrices $\in$ *O(3)* group,

and, such that,

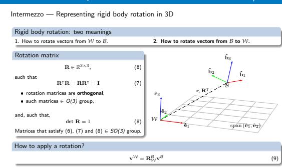
$$\det\,\mathbf{R} = 1 \tag{8}$$

Matrices that satisfy (6), (7) and (8) $\in$ *SO(3)* group.



How to apply a rotation?

$$\mathbf{v}^{\mathcal{W}} = \mathbf{R}_{\mathcal{W}}^{\mathcal{B}}\mathbf{v}^{\mathcal{B}} \tag{9}$$



- Orthogonal matrices have orthonormal column vectors.
- We choose to have rotation matrix $\mathbf{R}$ to transform vectors from the frame $\mathcal{B}$ to the frame $\mathcal{W}$.
- Rotation matrix $\mathbf{R}$ in the figure would have the orthonormal basis $\{\hat{\mathbf{b}}_1, \hat{\mathbf{b}}_2, \hat{\mathbf{b}}_3\}$ in its columns.

Intermezzo — Representing rigid body rotation in 3D

Rigid body rotation: two meanings

1. How to rotate vectors from $\mathcal{W}$ to $\mathcal{B}$.
2. **How to rotate vectors from $\mathcal{B}$ to $\mathcal{W}$.**

Rotation matrix

$$\mathbf{R} \in \mathbb{R}^{3\times3}, \tag{6}$$

such that

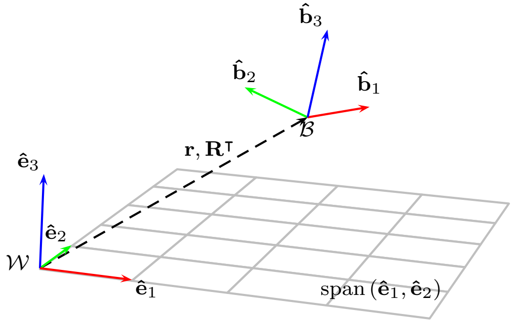
$$\mathbf{R}^\intercal\mathbf{R} = \mathbf{R}\mathbf{R}^\intercal = \mathbf{I} \tag{7}$$

- rotation matrices are **orthogonal**,
- such matrices $\in$ *O(3)* group,

and, such that,

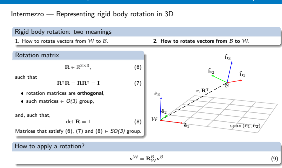
$$\det \mathbf{R} = 1 \tag{8}$$

Matrices that satisfy (6), (7) and (8) $\in$ *SO(3)* group.



How to apply a rotation?

$$\mathbf{v}^{\mathcal{W}} = \mathbf{R}^{\mathcal{B}}_{\mathcal{W}}\mathbf{v}^{\mathcal{B}} \tag{9}$$



- Orthogonal matrices have orthonormal column vectors.
- We choose to have rotation matrix $\mathbf{R}$ to transform vectors from the frame $\mathcal{B}$ to the frame $\mathcal{W}$.
- Rotation matrix $\mathbf{R}$ in the figure would have the orthonormal basis $\{\hat{\mathbf{b}}_1, \hat{\mathbf{b}}_2, \hat{\mathbf{b}}_3\}$ in its columns.

Intermezzo — Representing rigid body rotation in 3D

Lecture 2: UAV Control
Tomáš Báča
Dynamics model
Input mapping
Rotations in 3D
Rotational dynamics
Translational dynamics
UAV control
Motor control
NMPC control
Angular rate control
Attitude control
Force control
Translation control
Lab task
Conclusion

Rigid body rotation: two meanings

1. How to rotate vectors from $\mathcal{W}$ to $\mathcal{B}$.
2. **How to rotate vectors from $\mathcal{B}$ to $\mathcal{W}$.**

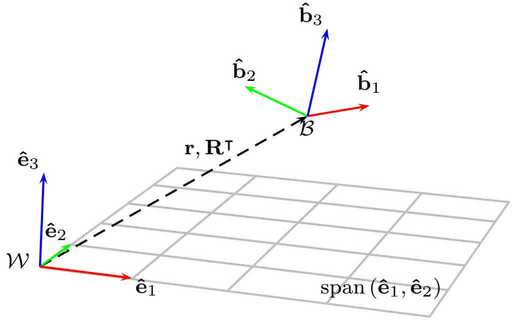
Unit quaternions

- Complex numbers with imaginary parts i, j, k

$$q = a + b\mathrm{i} + c\mathrm{j} + d\mathrm{k}, \tag{10}$$

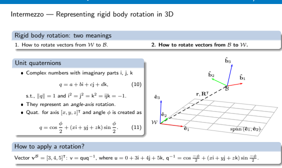
  s.t., $\|q\| = 1$ and $\mathrm{i}^2 = \mathrm{j}^2 = \mathrm{k}^2 = \mathrm{ijk} = -1$.
- They represent an *angle-axis* rotation.
- Quat. for axis $[x, y, z]^\intercal$ and angle ϕ is created as

$$q = \cos\frac{\phi}{2} + (x\mathrm{i} + y\mathrm{j} + z\mathrm{k})\sin\frac{\phi}{2}. \tag{11}$$



How to apply a rotation?

Vector $\mathbf{v}^{\mathcal{B}} = [3, 4, 5]^\intercal$: $\mathsf{v} = \mathsf{quq}^{-1}$, where $\mathsf{u} = 0 + 3\mathrm{i} + 4\mathrm{j} + 5\mathrm{k}$, $\mathsf{q}^{-1} = \cos\frac{-\phi}{2} + (x\mathrm{i} + y\mathrm{j} + z\mathrm{k})\sin\frac{-\phi}{2}$.



- Great video on the topic by 3Blue1Brown: `https://www.youtube.com/watch?v=d4EgbgTm0Bg`.
- Quaternions are rarely used "directly", but rather serve as a way to store and transmit rotations.
- Quaternions are very difficult to visualize intuitively, just by looking at their coefficients.
- Quaternions are very useful for interpolating rotations.
- The important drawback of quaternions is their **ambiguity**: There are **two** quaternions for each unique rotation.

Intermezzo — Representing rigid body rotation in 3D

Rigid body rotation: two meanings

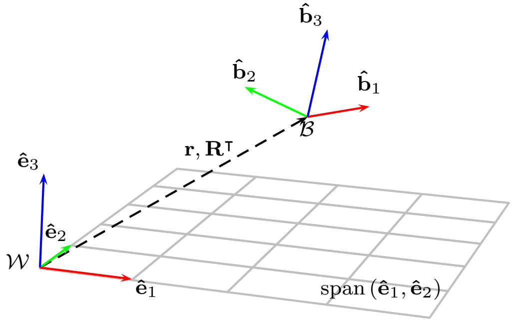
1. How to rotate vectors from $\mathcal{W}$ to $\mathcal{B}$.
2. **How to rotate vectors from $\mathcal{B}$ to $\mathcal{W}$.**

Euler Angles

- Decomposition of $\mathbf{R}$ to three sub-rotations

$$\mathbf{R}(\phi, \theta, \psi) = \mathbf{R}_3(\psi)\mathbf{R}_2(\theta)\mathbf{R}_1(\phi), \tag{12}$$

  each being applied around different axis.
- The first and the last rotation around the same axis (proper angles).
- 6 pos.: y-x-y, z-x-z, x-y-x, z-y-z, x-z-x, y-z-y.
- **Extrinsic**: around the axes of the original system.
- **Intrinsic**: around the axes of the new system.

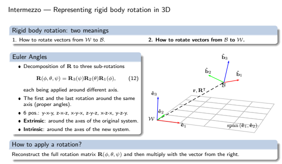


How to apply a rotation?

Reconstruct the full rotation matrix $\mathbf{R}(\phi, \theta, \psi)$ and then multiply with the vector from the right.

2024-09-29

Lecture 2: UAV Control
└─Dynamics model
  └─Rotations in 3D
    └─Intermezzo — Representing rigid body rotation in 3D


Intermezzo — Representing rigid body rotation in 3D
Rigid body rotation: two meanings
1. How to rotate vectors from $\mathcal{W}$ to $\mathcal{B}$. 2. How to rotate vectors from $\mathcal{B}$ to $\mathcal{W}$.
Euler Angles
Decomposition of $\mathbf{R}$ to three sub-rotations
$\mathbf{R}(\phi, \theta, \psi) = \mathbf{R}_3(\psi)\mathbf{R}_2(\theta)\mathbf{R}_1(\phi)$, (12)
each being applied around different axis.
The first and the last rotation around the same axis (proper angles).
6 pos.: y-x-y, z-x-z, x-y-x, z-y-z, x-z-x, y-z-y.
Extrinsic: around the axes of the original system.
Intrinsic: around the axes of the new system.
How to apply a rotation?
Reconstruct the full rotation matrix $\mathbf{R}(\phi, \theta, \psi)$ and then multiply with the vector from the right.

Intermezzo — Representing rigid body rotation in 3D

Rigid body rotation: two meanings

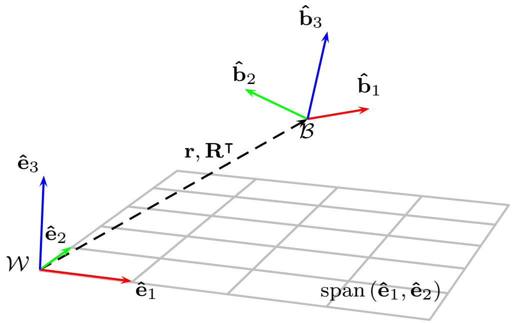
1. How to rotate vectors $\mathcal{W}$ to $\mathcal{B}$.
2. **How to rotate vectors from $\mathcal{B}$ to $\mathcal{W}$.**

Tait-Bryan angles

- Decomposition of $\mathbf{R}$ to three sub-rotations

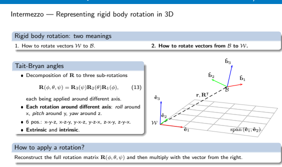
$$\mathbf{R}(\phi, \theta, \psi) = \mathbf{R}_3(\psi)\mathbf{R}_2(\theta)\mathbf{R}_1(\phi), \tag{13}$$

  each being applied around different axis.
- **Each rotation around different axis**: *roll* around x, *pitch* around y, *yaw* around z.
- 6 pos.: x-y-z, x-z-y, y-x-z, y-z-x, z-x-y, z-y-x.
- **Extrinsic** and **intrinsic**.



How to apply a rotation?

Reconstruct the full rotation matrix $\mathbf{R}(\phi, \theta, \psi)$ and then multiply with the vector from the right.

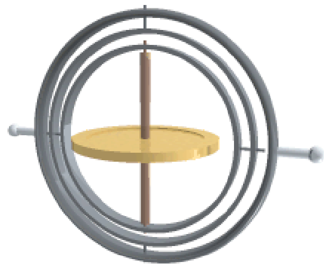


- Tait-Bryan/Euler angles are often mistaken for the *Tilt angles* (see the end of the presentation).
- They are applied in sequence.
- They are all interdependent to create the 3D rotation.
- They are often used when the model is linearized, e.g., around the *hover point*.

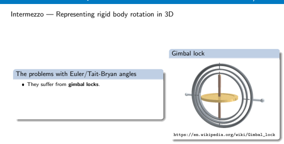
The problems with Euler/Tait-Bryan angles

- They suffer from **gimbal locks**.

Gimbal lock



https://en.wikipedia.org/wiki/Gimbal_lock

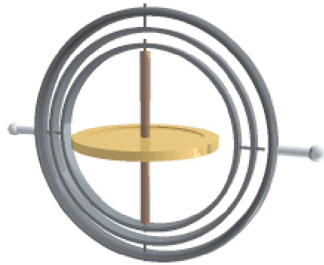


- Gimbal locks lead to the loss of DOFs at some configurations.
- Ambiguities lead to two combination of angles for each 3D rotation.
- The fixed ordering of operations leads to poor intuitive control over the final rotation.
- The **yaw** is often mistaken for the **heading**.

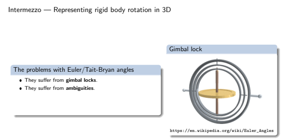
The problems with Euler/Tait-Bryan angles

- They suffer from **gimbal locks**.
- They suffer from **ambiguities**.

Gimbal lock



https://en.wikipedia.org/wiki/Euler_Angles

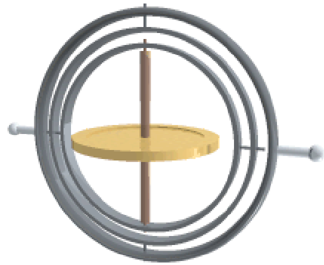


- Gimbal locks lead to the loss of DOFs at some configurations.
- Ambiguities lead to two combination of angles for each 3D rotation.
- The fixed ordering of operations leads to poor intuitive control over the final rotation.
- The **yaw** is often mistaken for the **heading**.

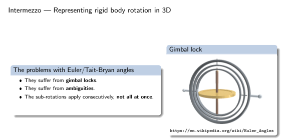
The problems with Euler/Tait-Bryan angles

- They suffer from **gimbal locks**.
- They suffer from **ambiguities**.
- The sub-rotations apply consecutively, **not all at once**.

Gimbal lock



https://en.wikipedia.org/wiki/Euler_Angles



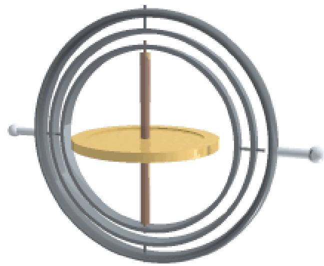
- Gimbal locks lead to the loss of DOFs at some configurations.
- Ambiguities lead to two combination of angles for each 3D rotation.
- The fixed ordering of operations leads to poor intuitive control over the final rotation.
- The **yaw** is often mistaken for the **heading**.

Intermezzo — Representing rigid body rotation in 3D

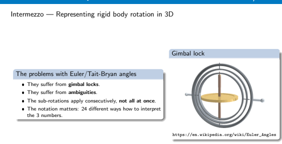
The problems with Euler/Tait-Bryan angles

- They suffer from **gimbal locks**.
- They suffer from **ambiguities**.
- The sub-rotations apply consecutively, **not all at once**.
- The notation matters: 24 different ways how to interpret the 3 numbers.

Gimbal lock

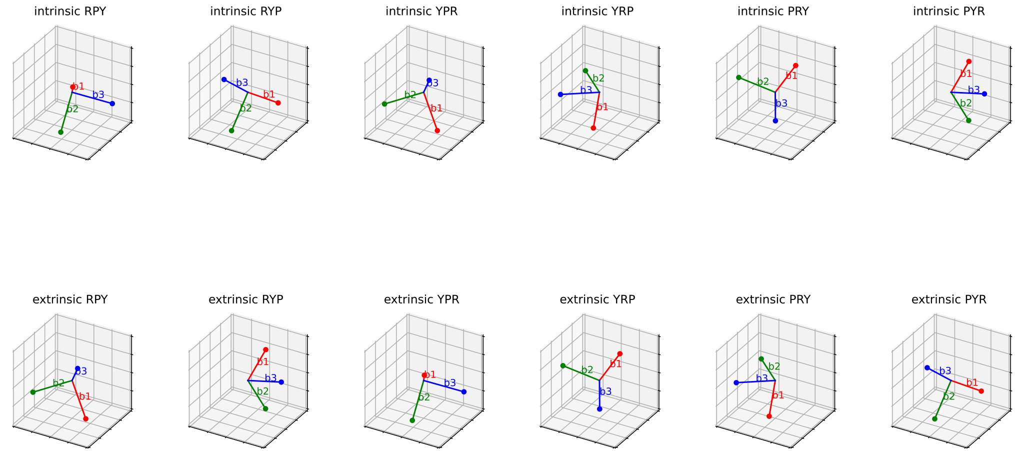


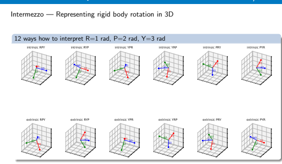
https://en.wikipedia.org/wiki/Euler_Angles


Intermezzo — Representing rigid body rotation in 3D
The problems with Euler/Tait-Bryan angles
They suffer from gimbal locks.
They suffer from ambiguities.
The sub-rotations apply consecutively, not all at once.
The notation matters: 24 different ways how to interpret the 3 numbers.
Gimbal lock
https://en.wikipedia.org/wiki/Euler_Angles

- Gimbal locks lead to the loss of DOFs at some configurations.
- Ambiguities lead to two combination of angles for each 3D rotation.
- The fixed ordering of operations leads to poor intuitive control over the final rotation.
- The **yaw** is often mistaken for the **heading**.

12 ways how to interpret R=1 rad, P=2 rad, Y=3 rad





- All 12 configurations of 3D rotaion, if **yaw**, **pitch**, **roll** is given, without specifying the notation.
- This causes serious problem in practical use.
- Note that: Extrinsic ABC = Intrinsic CBA.

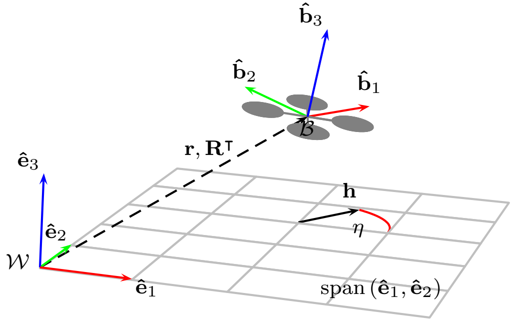
Intermezzo — Representing rigid body rotation in 3D

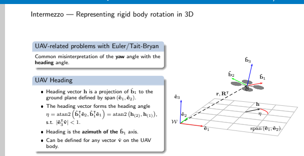
UAV-related problems with Euler/Tait-Bryan

Common misinterpretation of the **yaw** angle with the **heading** angle.

UAV Heading

- Heading vector $\mathbf{h}$ is a projection of $\hat{\mathbf{b}}_1$ to the ground plane defined by span $(\hat{\mathbf{e}}_1, \hat{\mathbf{e}}_2)$.
- The heading vector forms the heading angle $\eta = \operatorname{atan2}\left(\hat{\mathbf{b}}_1^\intercal \hat{\mathbf{e}}_2, \hat{\mathbf{b}}_1^\intercal \hat{\mathbf{e}}_1\right) = \operatorname{atan2}\left(\mathbf{h}_{(2)}, \mathbf{h}_{(1)}\right)$, s.t. $|\hat{\mathbf{e}}_3^\intercal \hat{\mathbf{v}}| < 1$.
- Heading is the **azimuth of the** $\hat{\mathbf{b}}_1$ axis.
- Can be defined for any vector $\hat{\mathbf{v}}$ on the UAV body.





- The **yaw** angle is often mistaken for the *heading* angle (the azimuth of the $\mathbf{b}_1$ axis).
- One can not specify just the **yaw** (e.g., to point the front of the UAV somewhere) without taking the value of **roll** and **pitch** into account.

Control input mapping

From **force & torques** to **per-motor angular speed**:

$$\begin{bmatrix}\omega_1\\ \omega_2\\ \omega_3\\ \omega_4\end{bmatrix} = \sqrt{\left(\mathbf{\Gamma}^{-1}\begin{bmatrix}F_t\\ \tau_1\\ \tau_2\\ \tau_3\end{bmatrix}\right)k^{-1}} \tag{14}$$

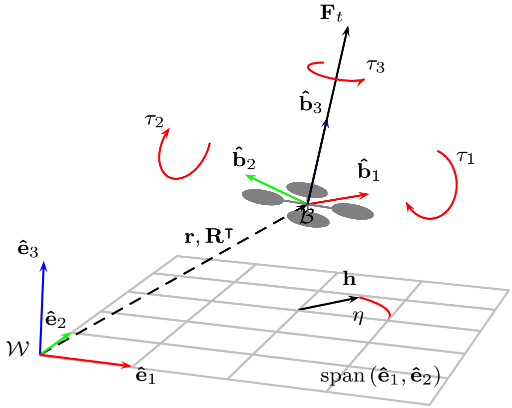
Control input: $\boldsymbol{\tau} = [\tau_1, \tau_2, \tau_3]^\intercal$, F_t.

Physical properties of the UAV

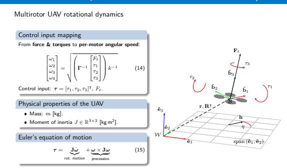
- Mass: m [kg].
- Moment of inertia $J \in \mathbb{R}^{3\times 3}$ [kg m^2].

Euler's equation of motion

$$\boldsymbol{\tau} = \underbrace{\mathbf{J}\dot{\boldsymbol{\omega}}}_{\text{rot. motion}} + \underbrace{\boldsymbol{\omega}\times\mathbf{J}\boldsymbol{\omega}}_{\text{precession}} \tag{15}$$



2024-09-29

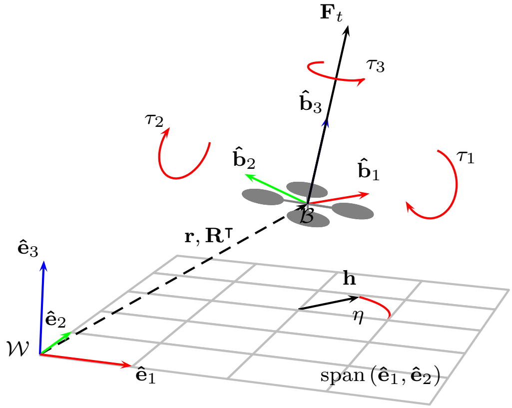


- $\mathbf{J}\boldsymbol{\omega}$ is the angular momentum

Euler's equation of motion

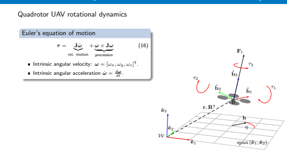
$$\boldsymbol{\tau} = \underbrace{\mathbf{J}\dot{\boldsymbol{\omega}}}_{\text{rot. motion}} + \underbrace{\boldsymbol{\omega} \times \mathbf{J}\boldsymbol{\omega}}_{\text{precession}} \tag{16}$$

- Intrinsic angular velocity: $\boldsymbol{\omega} = [\omega_x, \omega_y, \omega_z]^\intercal$.
- Intrinsic angular acceleration $\dot{\boldsymbol{\omega}} = \frac{d\boldsymbol{\omega}}{dt}$.



2024-09-29

Lecture 2: UAV Control
└─Dynamics model
└─Rotational dynamics
└─Quadrotor UAV rotational dynamics



- $\mathbf{J}\boldsymbol{\omega}$ is the angular momentum

Euler's equation of motion

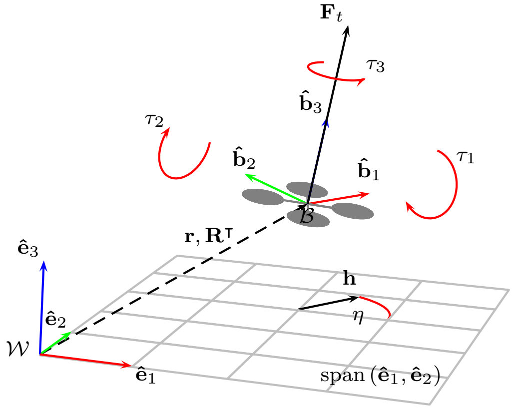
$$\boldsymbol{\tau} = \underbrace{\mathbf{J}\dot{\boldsymbol{\omega}}}_{\text{rot. motion}} + \underbrace{\boldsymbol{\omega} \times \mathbf{J}\boldsymbol{\omega}}_{\text{precession}} \tag{17}$$

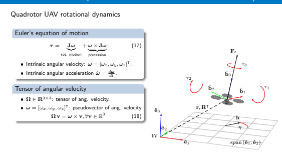
- Intrinsic angular velocity: $\boldsymbol{\omega} = [\omega_x, \omega_y, \omega_z]^\intercal$.
- Intrinsic angular acceleration $\dot{\boldsymbol{\omega}} = \frac{d\boldsymbol{\omega}}{dt}$.

Tensor of angular velocity

- $\boldsymbol{\Omega} \in \mathbf{R}^{3\times 3}$: tensor of ang. velocity.
- $\boldsymbol{\omega} = [\omega_x, \omega_y, \omega_z]^\intercal$: pseudovector of ang. velocity

$$\boldsymbol{\Omega}\,\mathbf{v} = \boldsymbol{\omega} \times \mathbf{v}, \forall \mathbf{v} \in \mathbb{R}^3 \tag{18}$$





- $\mathbf{J}\boldsymbol{\omega}$ is the angular momentum
- Tensor of angular velocity is a skew-symmetric matrix:

$$\boldsymbol{\Omega} = \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix} \tag{16}$$

Quadrotor UAV rotational dynamics

Euler's equation of motion

$$\boldsymbol{\tau} = \underbrace{\mathbf{J}\dot{\boldsymbol{\omega}}}_{\text{rot. motion}} + \underbrace{\boldsymbol{\omega} \times \mathbf{J}\boldsymbol{\omega}}_{\text{precession}} \tag{17}$$

- Intrinsic angular velocity: $\boldsymbol{\omega} = [\omega_x, \omega_y, \omega_z]^\intercal$.
- Intrinsic angular acceleration $\dot{\boldsymbol{\omega}} = \frac{d\boldsymbol{\omega}}{dt}$.

Tensor of angular velocity

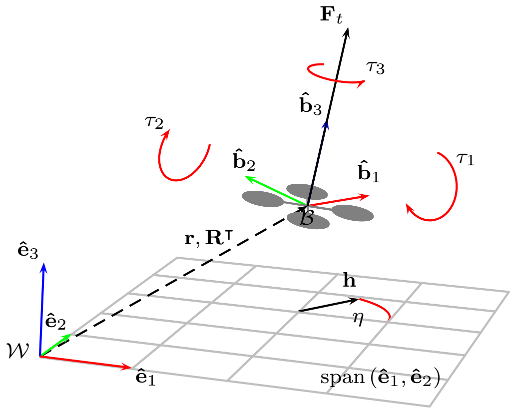
- $\mathbf{\Omega} \in \mathbf{R}^{3\times 3}$: tensor of ang. velocity.
- $\boldsymbol{\omega} = [\omega_x, \omega_y, \omega_z]^\intercal$: pseudovector of ang. velocity

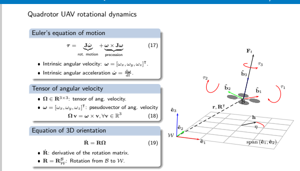
$$\mathbf{\Omega}\,\mathbf{v} = \boldsymbol{\omega} \times \mathbf{v}, \forall \mathbf{v} \in \mathbb{R}^3 \tag{18}$$

Equation of 3D orientation

$$\dot{\mathbf{R}} = \mathbf{R}\mathbf{\Omega} \tag{19}$$

- $\dot{\mathbf{R}}$: derivative of the rotation matrix.
- $\mathbf{R} = \mathbf{R}_{\mathcal{W}}^{\mathcal{B}}$: Rotation from $\mathcal{B}$ to $\mathcal{W}$.





- $\mathbf{J}\boldsymbol{\omega}$ is the angular momentum
- Tensor of angular velocity is a skew-symmetric matrix:

$$\mathbf{\Omega} = \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix} \tag{20}$$

Multirotor UAV translational dynamics

Rotational Dynamics

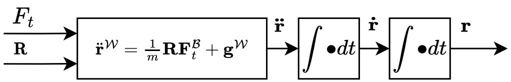
$$\boldsymbol{\tau} = \mathbf{J}\dot{\boldsymbol{\omega}} + \boldsymbol{\omega} \times \mathbf{J}\boldsymbol{\omega} \tag{21}$$

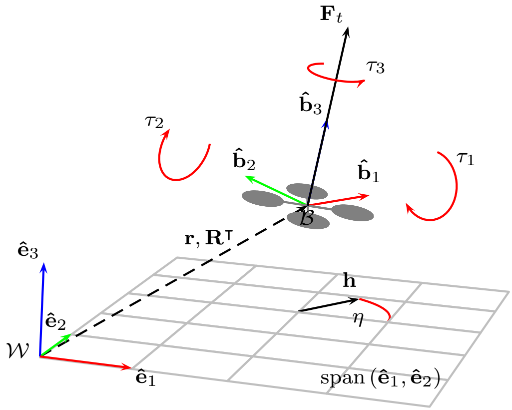
$$\dot{\mathbf{R}} = \mathbf{R}\boldsymbol{\Omega} \tag{22}$$

Translational dynamics: 2nd Newton's law

$$\ddot{\mathbf{r}}^{\mathcal{W}} = \frac{1}{m}\mathbf{R}\begin{bmatrix}0\\0\\F_t\end{bmatrix}^{\mathcal{B}} + \mathbf{g}^{\mathcal{W}} \tag{23}$$

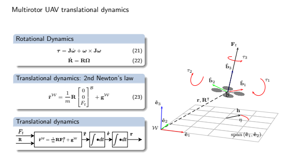
Translational dynamics





2024-09-29

Lecture 2: UAV Control
└─Dynamics model
└─Translational dynamics
└─Multirotor UAV translational dynamics



- The multirotor UAV body dynamics can be split to two: **rotational dynamic** and **translational dynamics**.

Full dynamics model

$$\begin{bmatrix}\dot{\omega}_1\\ \dot{\omega}_2\\ \dot{\omega}_3\\ \dot{\omega}_4\end{bmatrix} = -\frac{1}{\tau_m}\left(\begin{bmatrix}\omega_1\\ \omega_2\\ \omega_3\\ \omega_4\end{bmatrix} - \begin{bmatrix}\omega_{1d}\\ \omega_{2d}\\ \omega_{3d}\\ \omega_{4d}\end{bmatrix}\right) \tag{24}$$

$$\begin{bmatrix}F_t\\ \tau_1\\ \tau_2\\ \tau_3\end{bmatrix} = \mathbf{\Gamma} k \begin{bmatrix}\omega_1^2\\ \omega_2^2\\ \omega_3^2\\ \omega_4^2\end{bmatrix} \tag{25}$$

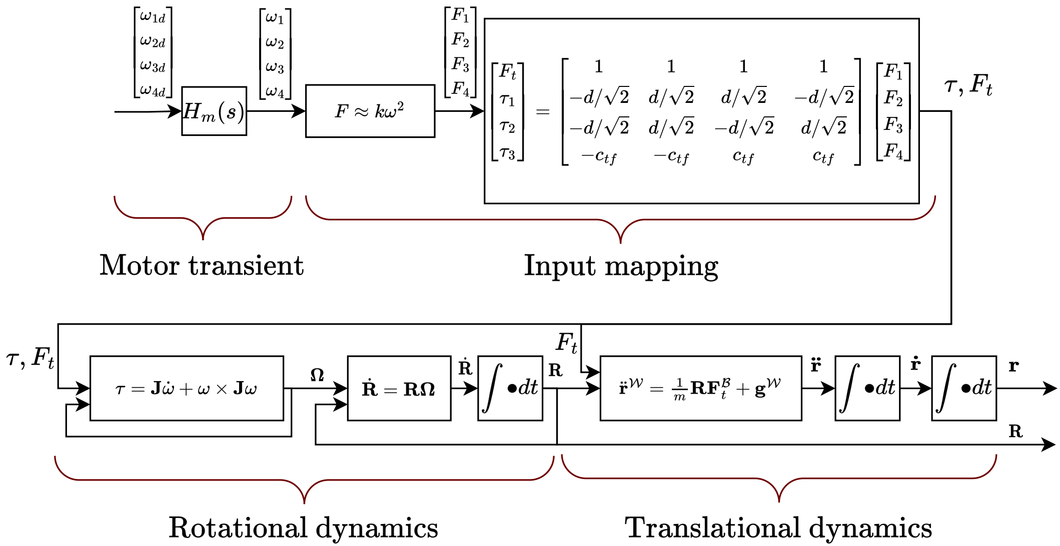
$$\boldsymbol{\tau} = \mathbf{J}\dot{\boldsymbol{\omega}} + \boldsymbol{\omega} \times \mathbf{J}\boldsymbol{\omega} \tag{26}$$

$$\dot{\mathbf{R}} = \mathbf{R}\mathbf{\Omega} \tag{27}$$

$$\ddot{\mathbf{r}}^{\mathcal{W}} = \frac{1}{m}\mathbf{R}\begin{bmatrix}0\\ 0\\ F_t\end{bmatrix}^{\mathcal{B}} + \mathbf{g}^{\mathcal{W}} \tag{28}$$

Multirotor UAV full dynamics
Full dynamics model
(24) (25) (26) (27) (28)

- The Full dynamics then include the **motor transients**, **force-torque allocation**, **rotational dynamics** and **translational dynamics**.
- But this is still the *simplest* full dynamics model. We neglected many phenomena such as **air drag**, **propeller aerodynamics**, **blade flapping**.



Lecture 2: UAV Control
Dynamics model
Translational dynamics
Multirotor UAV full dynamics

2024-09-29

Multirotor UAV full dynamics

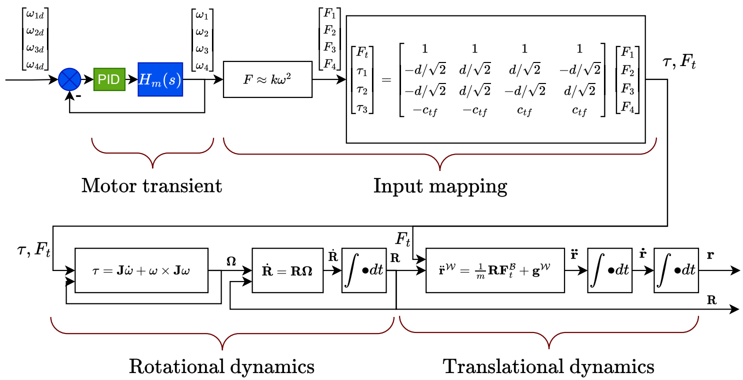
- The Full dynamics then include the **motor transients**, **force-torque allocation**, **rotational dynamics** and **translational dynamics**.
- But this is still the *simplest* full dynamics model. We neglected many phenomena such as **air drag**, **propeller aerodynamics**, **blade flapping**.

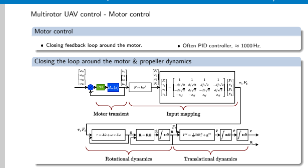
Multirotor UAV control - Motor control

Motor control

- Closing feedback loop around the motor.
- Often PID controller, $\approx$ 1000 Hz.

Closing the loop around the motor & propeller dynamics





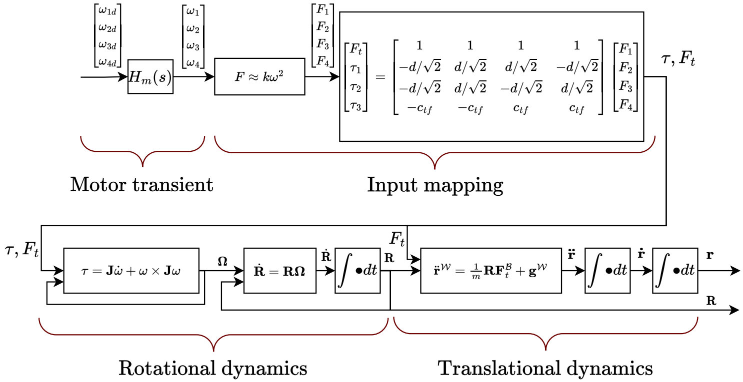
- The motor controller should be tuned such that the resulting motor & propeller & motor controller closed loop system exhibits critically-dampened first order transition on desired motor angular velocity.
- The time constant of the closed loop transient can be anywhere from 1 – 50 ms, depending on the size of the Unmanned Aerial Vehicle (UAV).

Multirotor UAV control - Motor control

Motor control

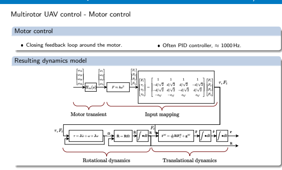
- Closing feedback loop around the motor.
- Often PID controller, $\approx 1000\,\text{Hz}$.

Resulting dynamics model



Lecture 2: UAV Control
└─UAV control
  └─Motor control
    └─Multirotor UAV control - Motor control

2024-09-29

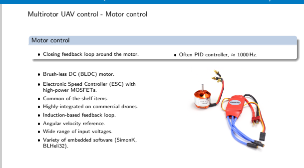


- The motor controller should be tuned such that the resulting motor & propeller & motor controller closed loop system exhibits critically-dampened first order transition on desired motor angular velocity.
- The time constant of the closed loop transient can be anywhere from 1 – 50 ms, depending on the size of the UAV.

Motor control

- Closing feedback loop around the motor.
- Often PID controller, ≈ 1000 Hz.

- Brush-less DC (BLDC) motor.
- Electronic Speed Controller (ESC) with high-power MOSFETs.
- Common of-the-shelf items.
- Highly-integrated on commercial drones.
- Induction-based feedback loop.
- Angular velocity reference.
- Wide range of input voltages.
- Variety of embedded software (SimonK, BLHeli32).



- The motor controller should be tuned such that the resulting motor & propeller & motor controller closed loop system exhibits critically-dampened first order transition on desired motor angular velocity.
- The time constant of the closed loop transient can be anywhere from 1 – 50 ms, depending on the size of the UAV.

Multirotor UAV control - ESC configuration

- Lots of parameters to tune.
- Some can cause hardware damage.
- Use caution, ESC is the "heavy lifter" of the UAV.

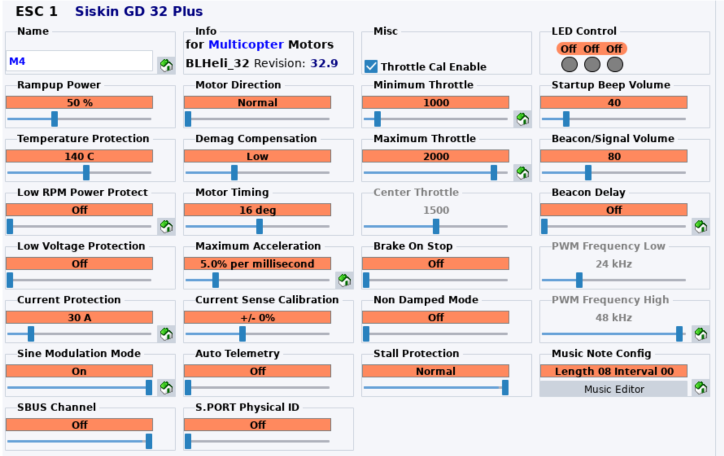
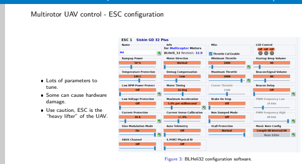


Figure 3: BLHeli32 configuration software.

Lecture 2: UAV Control
UAV control
Motor control
Multirotor UAV control - ESC configuration

2024-09-29



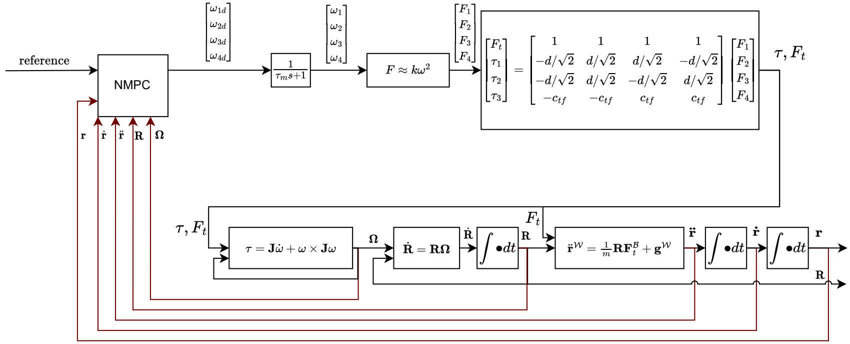
Multirotor UAV control – NMPC

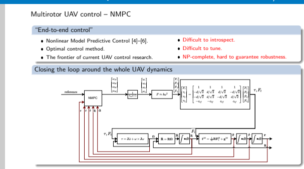
"End-to-end control"

- Nonlinear Model Predictive Control [4]–[6].
- Optimal control method.
- The frontier of current UAV control research.

- Difficult to introspect.
- Difficult to tune.
- NP-complete, hard to guarantee robustness.

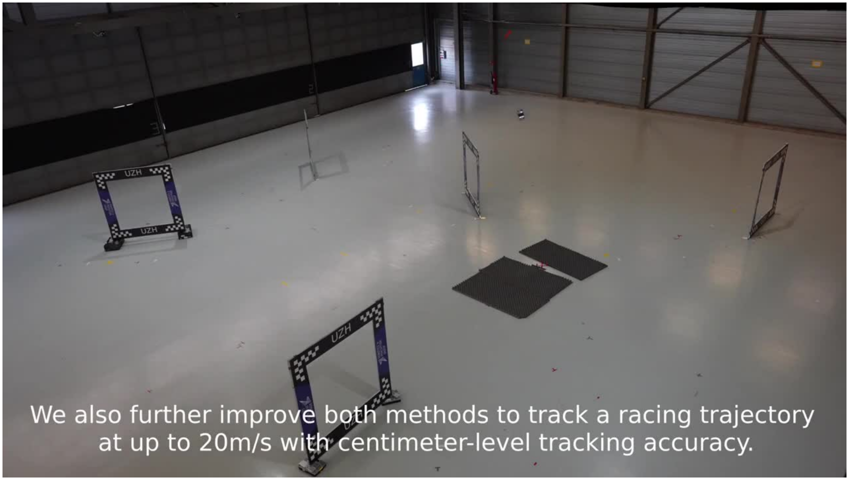
Closing the loop around the whole UAV dynamics



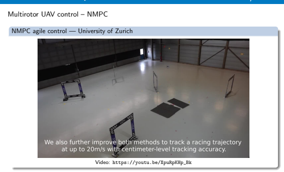


- It is still a nested loop with the motor ESC being already there. Motor control is better to leave for ESC for most applications. The exception is the study of **motor failure**, and the study of **fully-actuated** aerial vehicles, where the behaviour and performance of motor control needs to be adjusted directly by the UAV controller.

NMPC agile control — University of Zurich



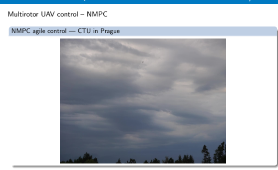
Video: `https://youtu.be/XpuRpKHp_Bk`



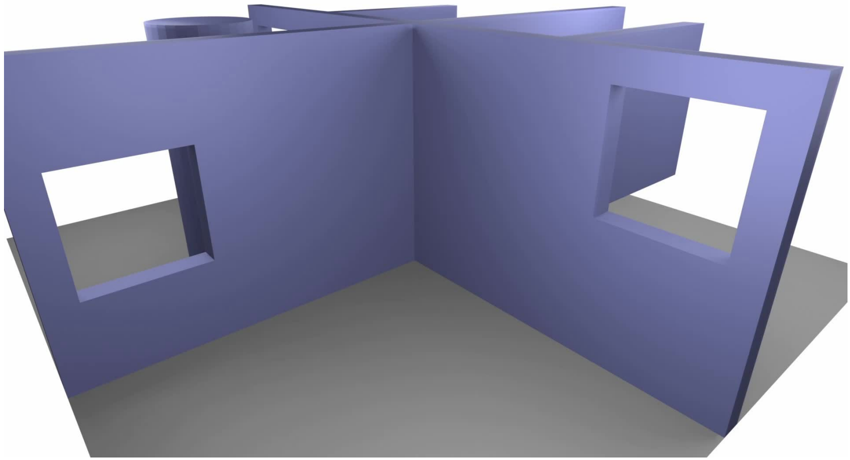
NMPC agile control — CTU in Prague

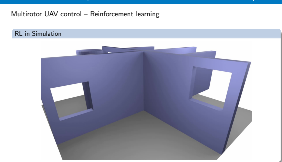
2024-09-29

Lecture 2: UAV Control
└─UAV control
 └─NMPC control
  └─Multirotor UAV control – NMPC



RL in Simulation





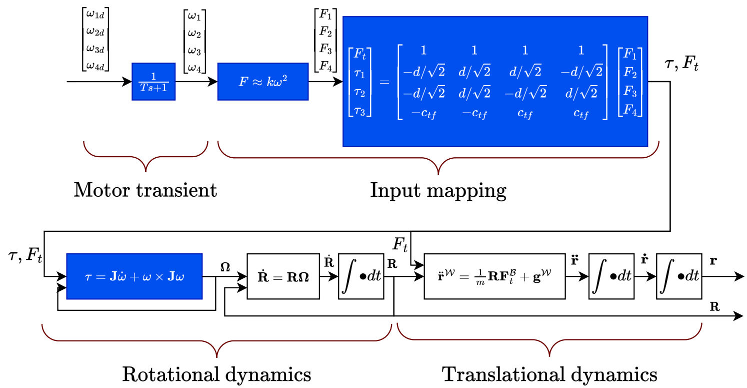
- Forefront of AI research in UAV control.
- Not very practical yet.

Multirotor UAV control — Nested control

Angular rate control

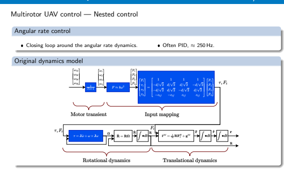
- Closing loop around the angular rate dynamics.
- Often PID, ≈ 250 Hz.

Original dynamics model



Lecture 2: UAV Control
└UAV control
└Angular rate control
└Multirotor UAV control — Nested control

2024-09-29

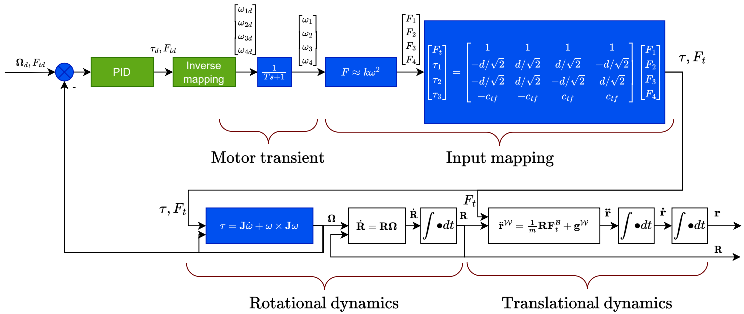


Multirotor UAV control — Nested control

Angular rate control

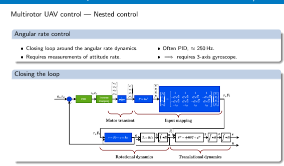
- Closing loop around the angular rate dynamics.
- Requires measurements of attitude rate.
- Often PID, ≈ 250 Hz.
- $\implies$ requires 3-axis gyroscope.

Closing the loop



Lecture 2: UAV Control
└─UAV control
└─Angular rate control
└─Multirotor UAV control — Nested control

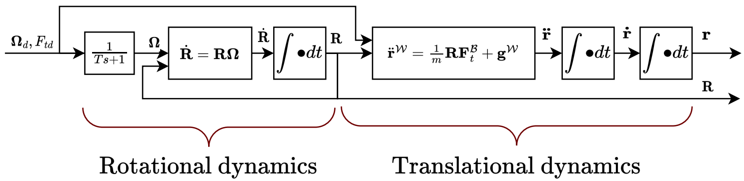
2024-09-29



Angular rate control

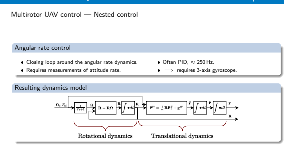
- Closing loop around the angular rate dynamics.
- Requires measurements of attitude rate.
- Often PID, ≈ 250 Hz.
- $\implies$ requires 3-axis gyroscope.

Resulting dynamics model



Lecture 2: UAV Control
└UAV control
 └Angular rate control
  └Multirotor UAV control — Nested control

2024-09-29



Multirotor UAV control — Nested control

Angular rate control

- Closing loop around the angular rate dynamics.
- Often PID, $\approx 250\,\mathrm{Hz}$.

- Flight controllers are common of-the-shelf part.
- Highly-integrated in commercial platforms.
- Implement the attitude rate PID controller.
- Embedded Micro Electro Mechanical Systems (MEMS) 3-axis gyroscope and accelerometer.
- Implement filters for the attitude rate.
- Interfaces for remote radio controllers.
- Send signals directly to the ESCs.
- Often capable of using GPS receiver, Barometer, Height range finder.

Embedded flight controllers

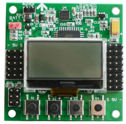
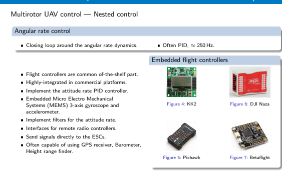


Figure 4: KK2

Figure 6: DJI Naza

Figure 5: Pixhawk

Figure 7: Betaflight

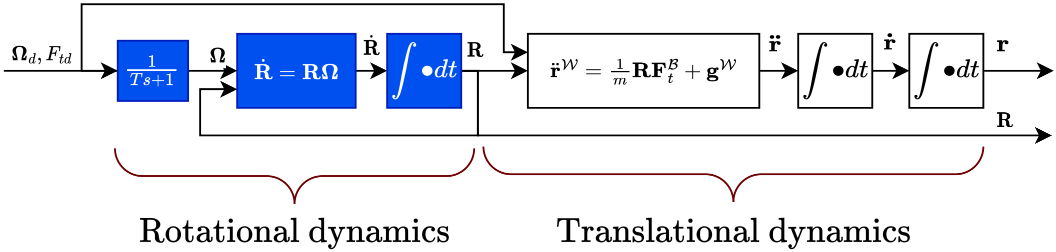


Multirotor UAV control — Nested control

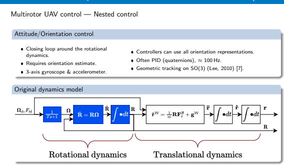
Attitude/Orientation control

- Closing loop around the rotational dynamics.
- Requires orientation estimate.
- 3-axis gyroscope & accelerometer.
- Controllers can use all orientation representations.
- Often PID (quaternions), $\approx 100\,\text{Hz}$.
- Geometric tracking on SO(3) (Lee, 2010) [7].

Original dynamics model



2024-09-29

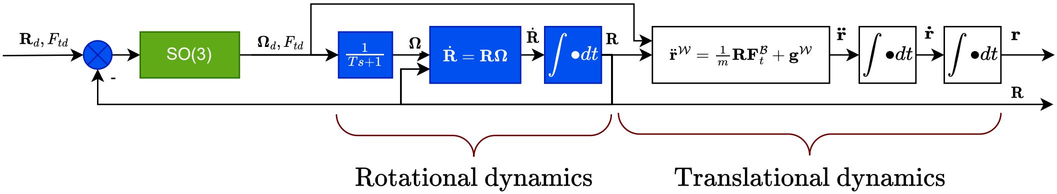


Multirotor UAV control — Nested control

Attitude/Orientation control

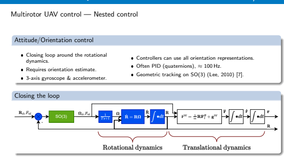
- Closing loop around the rotational dynamics.
- Requires orientation estimate.
- 3-axis gyroscope & accelerometer.
- Controllers can use all orientation representations.
- Often PID (quaternions), $\approx 100\,\text{Hz}$.
- Geometric tracking on SO(3) (Lee, 2010) [7].

Closing the loop



2024-09-29

Lecture 2: UAV Control
└─UAV control
 └─Attitude control
  └─Multirotor UAV control — Nested control

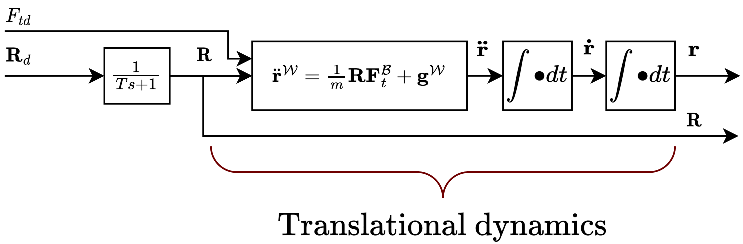


Multirotor UAV control — Nested control

Attitude/Orientation control

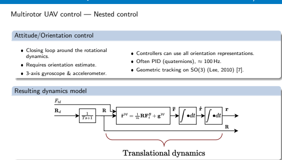
- Closing loop around the rotational dynamics.
- Requires orientation estimate.
- 3-axis gyroscope & accelerometer.
- Controllers can use all orientation representations.
- Often PID (quaternions), $\approx 100\,\text{Hz}$.
- Geometric tracking on SO(3) (Lee, 2010) [7].

Resulting dynamics model



Lecture 2: UAV Control
└─UAV control
  └─Attitude control
    └─Multirotor UAV control — Nested control

2024-09-29



Attitude/Orientation control

- Closing loop around the rotational dynamics.
- Requires orientation estimate.
- 3-axis gyroscope & accelerometer.
- Controllers can use all orientation representations.
- Often PID (quaternions), $\approx 100\,\text{Hz}$.
- Geometric tracking on SO(3) (Lee, 2010) [7].

Geometric tracking on SO(3) (simplified) [7]

- Orientation error: $\mathbf{E} = \frac{1}{2}\left(\mathbf{R}_d^\intercal \mathbf{R} - \mathbf{R}_d \mathbf{R}^\intercal\right)$.
- Orientation error vector: $\mathbf{e}_R = [\mathbf{E}_{2,3}, \mathbf{E}_{3,1}, \mathbf{E}_{1,2}]^\intercal$.
- Proportional orientation feedback:

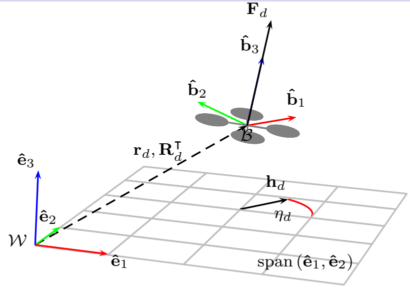
$$\boldsymbol{\omega}_d = k_R \mathbf{e}_R. \qquad (24)$$

2024-09-29

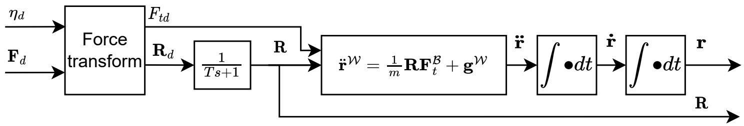
Multirotor UAV control — Nested control
Attitude/Orientation control
- Closing loop around the rotational dynamics.
- Requires orientation estimate.
- 3-axis gyroscope & accelerometer.
- Controllers can use all orientation representations.
- Often PID (quaternions), ≈ 100 Hz.
- Geometric tracking on SO(3) (Lee, 2010) [7].
Geometric tracking on SO(3) (simplified) [7]
- Orientation error: $\mathbf{E} = \frac{1}{2}\left(\mathbf{R}_d^\intercal \mathbf{R} - \mathbf{R}_d \mathbf{R}^\intercal\right)$.
- Orientation error vector: $\mathbf{e}_R = [\mathbf{E}_{2,3}, \mathbf{E}_{3,1}, \mathbf{E}_{1,2}]^\intercal$.
- Proportional orientation feedback:
$\boldsymbol{\omega}_d = k_R \mathbf{e}_R$. (24)

Force control

- Transforming inputs from desired 3D force and heading $\mathbf{F}_d, \eta_d$, to desired orientation and scalar thrust force $\mathbf{R}_d, F_{td}$.
- $\mathbf{F}_d, \eta_d$ and $\mathbf{R}_d, F_{td}$ are functionally equivalent, both offer control over the 4 DOFs.
- $\mathbf{F}_d, \eta_d$ is more intuitive to work with.

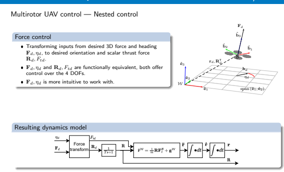


Resulting dynamics model



2024-09-29

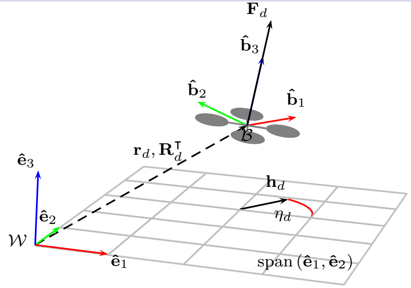
Lecture 2: UAV Control
└─UAV control
  └─Force control
    └─Multirotor UAV control — Nested control



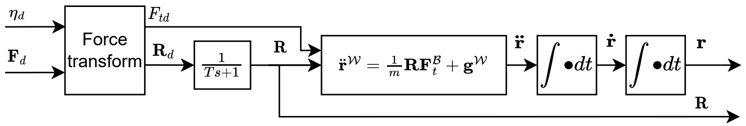
Multirotor UAV control — Nested control

Force control

1. We get $\mathbf{F}_d, \eta_d$.

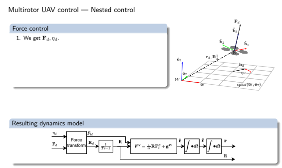


Resulting dynamics model



Lecture 2: UAV Control
└─UAV control
  └─Force control
    └─Multirotor UAV control — Nested control

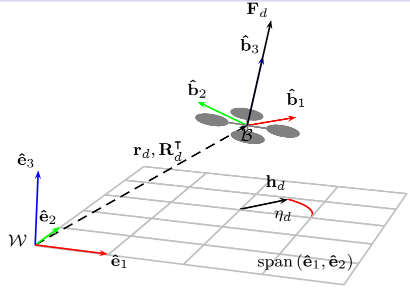
2024-09-29



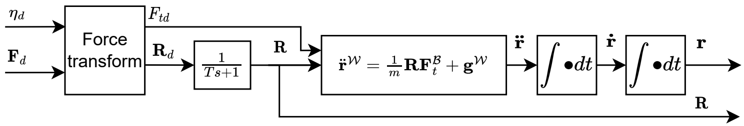
Multirotor UAV control — Nested control

Force control

1. We get $\mathbf{F}_d, \eta_d$.
2. We begin to construct $\mathbf{R}_d$ column by column as

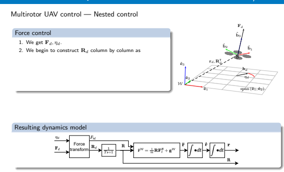


Resulting dynamics model



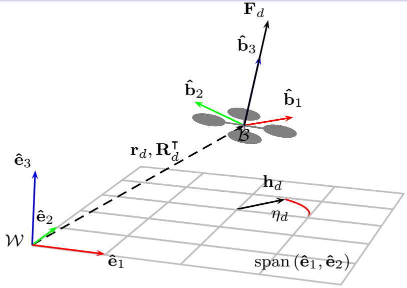
2024-09-29

Lecture 2: UAV Control
└─UAV control
└─Force control
└─Multirotor UAV control — Nested control

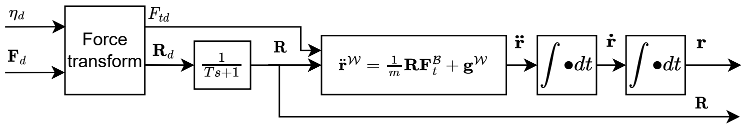


Force control

1. We get $\mathbf{F}_d, \eta_d$.
2. We begin to construct $\mathbf{R}_d$ column by column as
   - $\mathbf{R}_{d,[,3]} = \hat{\mathbf{b}}_{3d} = \mathbf{F}_d / \|\mathbf{F}_d\|$,

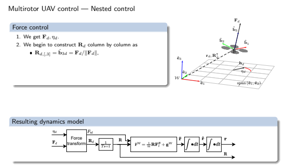


Resulting dynamics model



Lecture 2: UAV Control
└─UAV control
  └─Force control
    └─Multirotor UAV control — Nested control

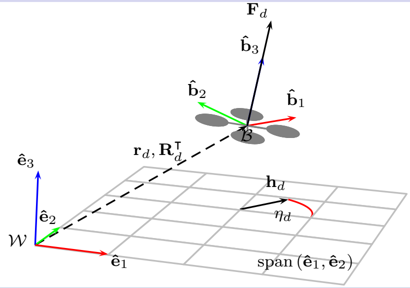
2024-09-29



Multirotor UAV control — Nested control

Force control

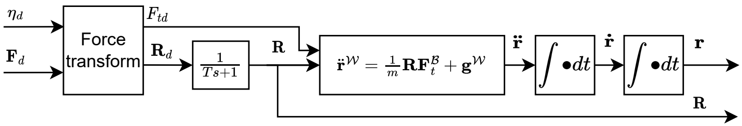
1. We get $\mathbf{F}_d, \eta_d$.
2. We begin to construct $\mathbf{R}_d$ column by column as
   - $\mathbf{R}_{d,[,3]} = \hat{\mathbf{b}}_{3d} = \mathbf{F}_d / \|\mathbf{F}_d\|$,
   - $\mathbf{R}_{d,[,1]} = \hat{\mathbf{b}}_{1d} =$ projection of $\mathbf{h}_d$ to span $\left(\hat{\mathbf{b}}_{3d}\right)^{\perp}$,



Projection of the heading vector

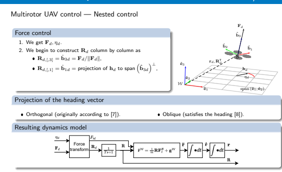
- Orthogonal (originally according to [7]).
- Oblique (satisfies the heading [8]).

Resulting dynamics model



Lecture 2: UAV Control
└─UAV control
  └─Force control
    └─Multirotor UAV control — Nested control

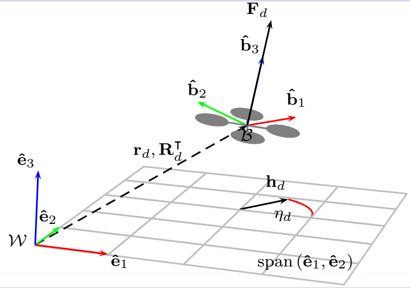
2024-09-29



Multirotor UAV control — Nested control

Force control

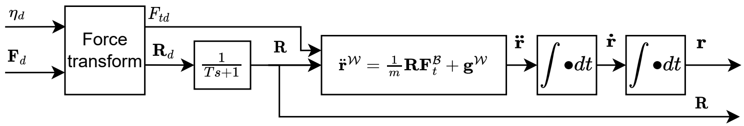
1. We get $\mathbf{F}_d, \eta_d$.
2. We begin to construct $\mathbf{R}_d$ column by column as
   - $\mathbf{R}_{d,[,3]} = \hat{\mathbf{b}}_{3d} = \mathbf{F}_d / \|\mathbf{F}_d\|$,
   - $\mathbf{R}_{d,[,1]} = \hat{\mathbf{b}}_{1d} =$ projection of $\mathbf{h}_d$ to span $\left(\hat{\mathbf{b}}_{3d}\right)^{\perp}$,
   - $\mathbf{R}_{d,[,2]} = \hat{\mathbf{b}}_{2d} = \hat{\mathbf{b}}_{3d} \times \hat{\mathbf{b}}_{1d}$.



Projection of the heading vector

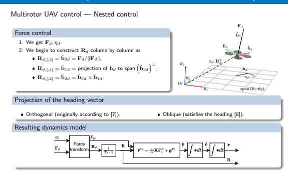
- Orthogonal (originally according to [7]).
- Oblique (satisfies the heading [8]).

Resulting dynamics model



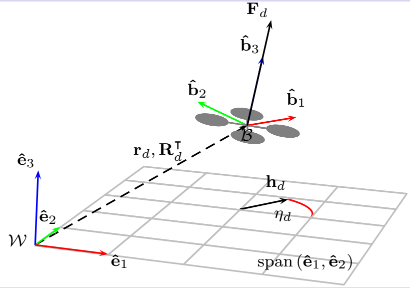
Lecture 2: UAV Control
└─UAV control
  └─Force control
    └─Multirotor UAV control — Nested control

2024-09-29



Force control

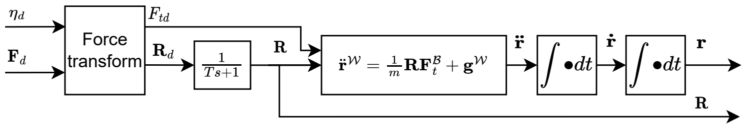
1. We get $\mathbf{F}_d, \eta_d$.
2. We begin to construct $\mathbf{R}_d$ column by column as
   - $\mathbf{R}_{d,[,3]} = \hat{\mathbf{b}}_{3d} = \mathbf{F}_d / \|\mathbf{F}_d\|$,
   - $\mathbf{R}_{d,[,1]} = \hat{\mathbf{b}}_{1d} =$ projection of $\mathbf{h}_d$ to span $\left(\hat{\mathbf{b}}_{3d}\right)^{\perp}$,
   - $\mathbf{R}_{d,[,2]} = \hat{\mathbf{b}}_{2d} = \hat{\mathbf{b}}_{3d} \times \hat{\mathbf{b}}_{1d}$.
3. Desired total thrust: $F_{td} = \|\mathbf{F}_d\|$.



Projection of the heading vector

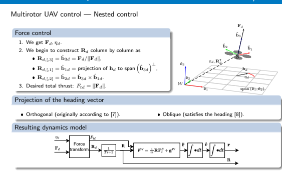
- Orthogonal (originally according to [7]).
- Oblique (satisfies the heading [8]).

Resulting dynamics model



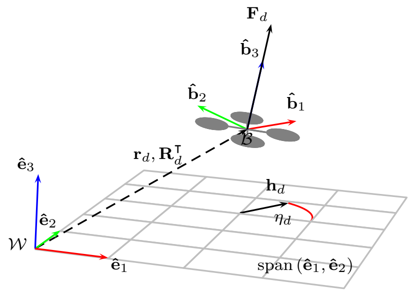
2024-09-29

Lecture 2: UAV Control
└─UAV control
  └─Force control
    └─Multirotor UAV control — Nested control

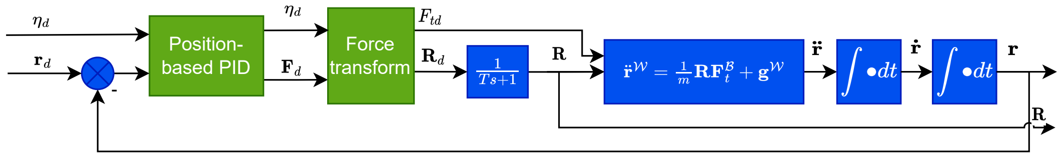


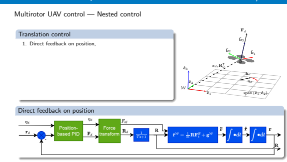
Translation control

1. Direct feedback on position,



Direct feedback on position

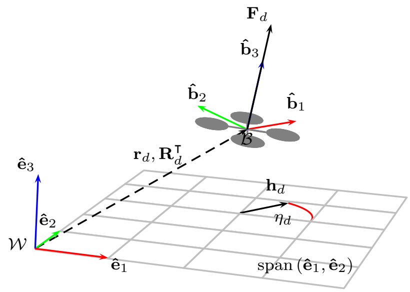




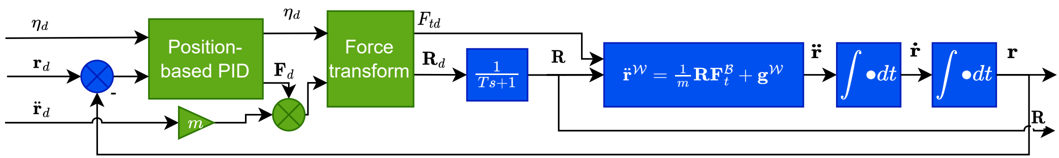
- Direct position PID control loop can run at $\approx 30\,\mathrm{Hz}$.

Translation control

1. Direct feedback on position,
2. + acceleration feedforward.

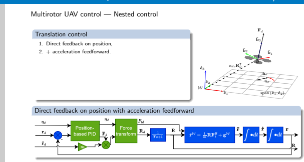


Direct feedback on position with acceleration feedforward



2024-09-29

Lecture 2: UAV Control
└─UAV control
└─Translation control
└─Multirotor UAV control — Nested control

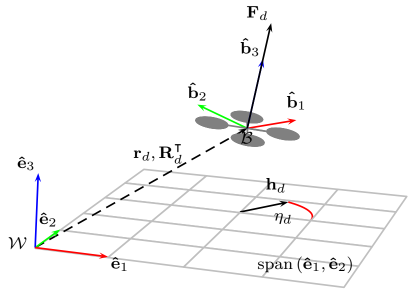


- Direct position PID control loop can run at $\approx 30\,\mathrm{Hz}$.

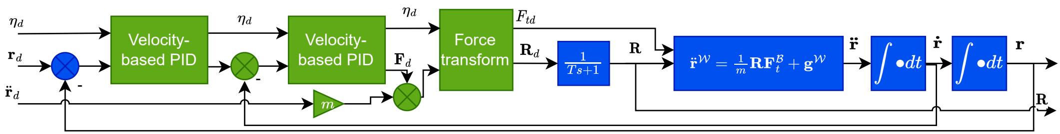
Multirotor UAV control — Nested control

Translation control

1. Direct feedback on position,
2. + acceleration feedforward.
3. Cascade velocity + position feedback with feedforward.

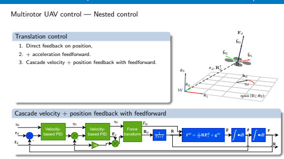


Cascade velocity + position feedback with feedforward



Lecture 2: UAV Control
└─UAV control
 └─Translation control
  └─Multirotor UAV control — Nested control

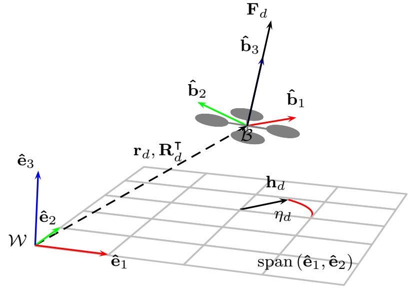
2024-09-29



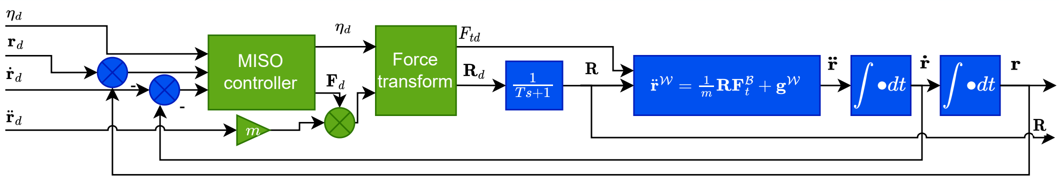
- Direct position PID control loop can run at $\approx 30\,\text{Hz}$.
- The outer position PID loop can run $\approx 10\,\text{Hz}$.

Translation control

1. Direct feedback on position,
2. + acceleration feedforward.
3. Cascade velocity + position feedback with feedforward.
4. MISO control with feedforward:
   - state feedback (e.g., pole placement, LQR),
   - Model Predictive Control (MPC),
   - multiple P-feedbacks,
   - geometric tracking on SE(3) [7].

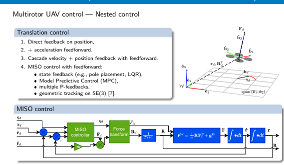


MISO control



2024-09-29

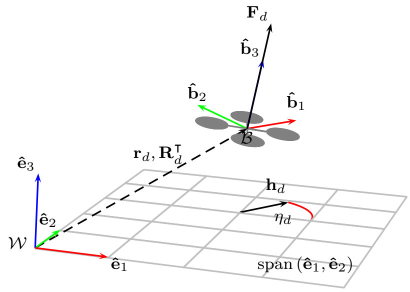
Lecture 2: UAV Control
└─UAV control
 └─Translation control
  └─Multirotor UAV control — Nested control



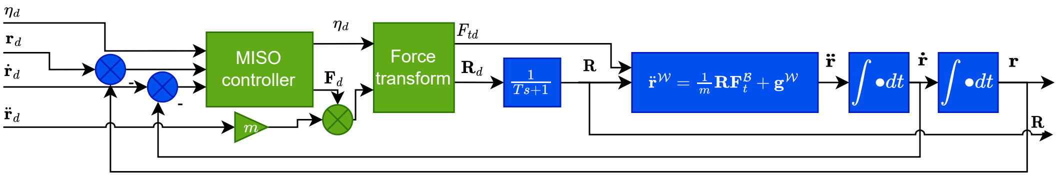
- Direct position PID control loop can run at ≈ 30 Hz.
- The outer position PID loop can run ≈ 10 Hz.

MISO control by MRS [8]

$$\mathbf{F}_d = \overbrace{\mathbf{k}_p \circ \mathbf{e}_p}^{\text{position feedback}} + \overbrace{\mathbf{k}_v \circ \mathbf{e}_v}^{\text{velocity feedback}} + \overbrace{m\ddot{\mathbf{r}}_d}^{\text{reference feedforward}} + \underbrace{mg\hat{\mathbf{e}}_3}_{\text{gravity compensation}} + \underbrace{-\mathbf{d}_w \circ \begin{bmatrix}1\\1\\0\end{bmatrix}}_{\text{world disturbance compensation}} + \underbrace{-\mathbf{d}_b \circ \begin{bmatrix}1\\1\\0\end{bmatrix}}_{\text{body disturbance compensation}} \quad (25)$$

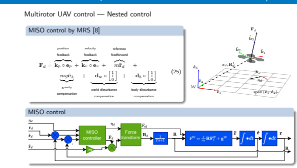


MISO control



Lecture 2: UAV Control
└─UAV control
└─Translation control
└─Multirotor UAV control — Nested control

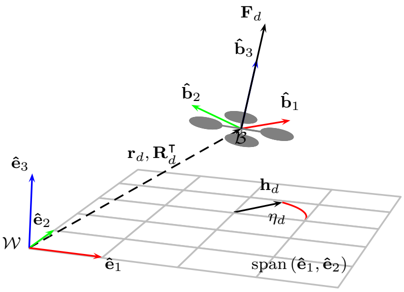
2024-09-29



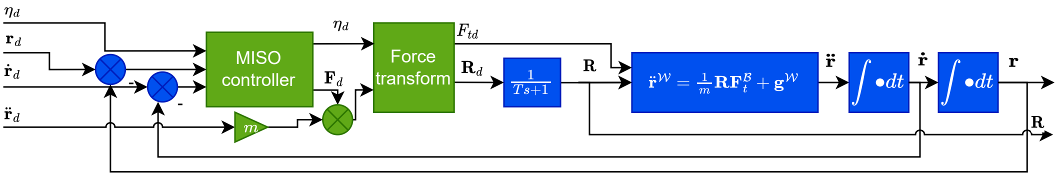
Multirotor UAV control — Nested control

MPC force feedback control by MRS [8]

$$\mathbf{F}_d = \overbrace{m\ddot{\mathbf{r}}_d}^{\text{reference feedforward}} + \overbrace{m\mathbf{c}_d}^{\text{MPC feedforward}} + \overbrace{mg\hat{\mathbf{e}}_3}^{\text{gravity compensation}} + \underbrace{-\mathbf{d}_w \circ \begin{bmatrix}1\\1\\0\end{bmatrix}}_{\text{world disturbance compensation}} + \underbrace{-\mathbf{d}_b \circ \begin{bmatrix}1\\1\\0\end{bmatrix}}_{\text{body disturbance compensation}} \quad (25)$$

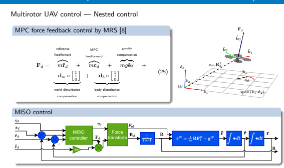


MISO control



2024-09-29

Lecture 2: UAV Control
└─UAV control
└─Translation control
└─Multirotor UAV control — Nested control



More information in publications:

[8] T. Baca, M. Petrlik, M. Vrba, V. Spurny, R. Penicka, D. Hert, *et al.*, "The MRS UAV System: Pushing the Frontiers of Reproducible Research, Real-world Deployment, and Education with Autonomous Unmanned Aerial Vehicles," *Journal of Intelligent & Robotic Systems*, vol. 102, no. 26, pp. 1–28, 1 May 2021
`https://arxiv.org/abs/2008.08050`

[9] T. Baca, D. Hert, G. Loianno, M. Saska, and V. Kumar, "Model Predictive Trajectory Tracking and Collision Avoidance for Reliable Outdoor Deployment of Unmanned Aerial Vehicles," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems*, IEEE, 2018, pp. 1–8
`http://mrs.felk.cvut.cz/data/papers/iros_2018_mpc.pdf`

Lecture 2: UAV Control
└─UAV control
  └─Translation control
    └─MRS UAV System

2024-09-29

MRS UAV System

More information in publications:

[8] T. Baca, M. Petrlik, M. Vrba, V. Spurny, R. Penicka, D. Hert, *et al.*, "The MRS UAV System: Pushing the Frontiers of Reproducible Research, Real-world Deployment, and Education with Autonomous Unmanned Aerial Vehicles," *Journal of Intelligent & Robotic Systems*, vol. 102, no. 26, pp. 1–28, 1 May 2021
https://arxiv.org/abs/2008.08050

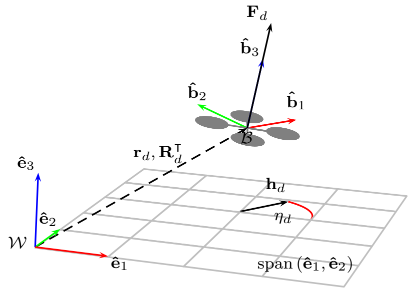
[9] T. Baca, D. Hert, G. Loianno, M. Saska, and V. Kumar, "Model Predictive Trajectory Tracking and Collision Avoidance for Reliable Outdoor Deployment of Unmanned Aerial Vehicles," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems*, IEEE, 2018, pp. 1–8
http://mrs.felk.cvut.cz/data/papers/iros_2018_mpc.pdf

Tilt control (more in the first lab task)

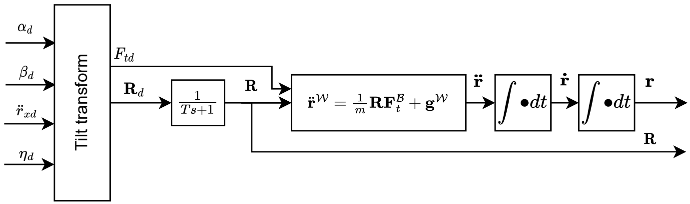
- Decoupling the UAV acceleration

$$
\begin{aligned}
\ddot{r}_x^{\mathcal{W}} &= \frac{\sin\alpha}{m}\|\mathbf{F}_{xz}^{\mathcal{W}}\|,\\
\ddot{r}_y^{\mathcal{W}} &= \frac{\sin\beta}{m}\|\mathbf{F}_{yz}^{\mathcal{W}}\|,\\
\ddot{r}_z^{\mathcal{W}} &= \frac{1}{m}F_z^{\mathcal{W}} - g.
\end{aligned} \tag{26}
$$

- Let's make α, β, and $\ddot{r}_z$ the inputs.

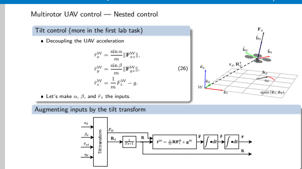


Augmenting inputs by the tilt transform



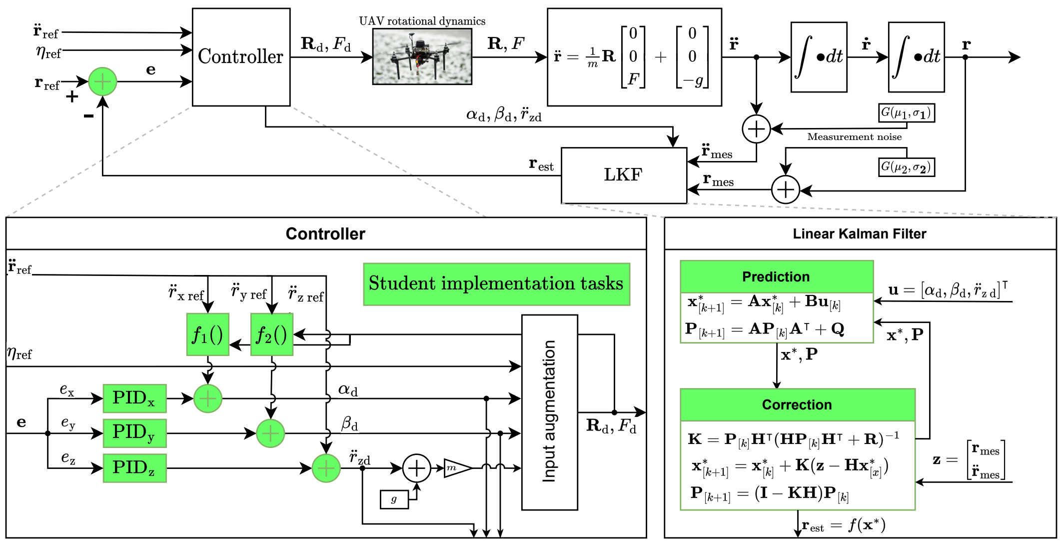
2024-09-29

Lecture 2: UAV Control
└─UAV control
  └─Translation control
    └─Multirotor UAV control — Nested control



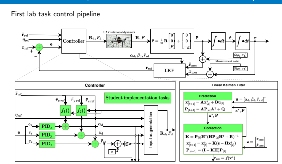
- This input mapping is rarely used in practice. We present it here due to it being used in the lab task.
- α and β are typically the control inputs over the hobby radio controller (after "undoing" the heading).

First lab task control pipeline



2024-09-29

Lecture 2: UAV Control
└ UAV control
  └ Lab task
    └ First lab task control pipeline



How to obtain states of the UAV

- We need estimates of $\mathbf{r}$, $\dot{\mathbf{r}}$, $\ddot{\mathbf{r}}$, $\mathbf{R}$, $\dot{\mathbf{R}}$, m, for control.
- Focus of **lecture 3**.

What did not fit into the lecture?
How to obtain states of the UAV
- We need estimates of $\mathbf{r}$, $\dot{\mathbf{r}}$, $\ddot{\mathbf{r}}$, $\mathbf{R}$, $\dot{\mathbf{R}}$, m, for control.
- Focus of **lecture 3**.

What did not fit into the lecture?

How to obtain states of the UAV

- We need estimates of $\mathbf{r}$, $\dot{\mathbf{r}}$, $\ddot{\mathbf{r}}$, $\mathbf{R}$, $\dot{\mathbf{R}}$, m, for control.
- Focus of **lecture 3**.

How to handle disturbances and uncertainties

- The subject of Robust control.
- How to estimate external disturbances and changes in the model?

What did not fit into the lecture?
How to obtain states of the UAV
- We need estimates of $\mathbf{r}$, $\dot{\mathbf{r}}$, $\ddot{\mathbf{r}}$, $\mathbf{R}$, $\dot{\mathbf{R}}$, m, for control.
- Focus of **lecture 3**.
How to handle disturbances and uncertainties
- The subject of Robust control.
- How to estimate external disturbances and changes in the model?

What did not fit into the lecture?

How to obtain states of the UAV

- We need estimates of $\mathbf{r}$, $\dot{\mathbf{r}}$, $\ddot{\mathbf{r}}$, $\mathbf{R}$, $\dot{\mathbf{R}}$, m, for control.
- Focus of **lecture 3**.

How to handle disturbances and uncertainties

- The subject of Robust control.
- How to estimate external disturbances and changes in the model?

How to obtain feasible feedforward?

- Pre-computed geometry-based methods, e.g., [10].
- In-real time, e.g., using Model Predictive Control [9].

Lecture 2: UAV Control
Conclusion
What did not fit into the lecture?
2024-09-29

What did not fit into the lecture?
How to obtain states of the UAV
We need estimates of $\mathbf{r}$, $\dot{\mathbf{r}}$, $\ddot{\mathbf{r}}$, $\mathbf{R}$, $\dot{\mathbf{R}}$, m, for control.
Focus of lecture 3.
How to handle disturbances and uncertainties
The subject of Robust control.
How to estimate external disturbances and changes in the model?
How to obtain feasible feedforward?
Pre-computed geometry-based methods, e.g., [10].
In-real time, e.g., using Model Predictive Control [9].

What did not fit into the lecture?

How to obtain states of the UAV

- We need estimates of $\mathbf{r}$, $\dot{\mathbf{r}}$, $\ddot{\mathbf{r}}$, $\mathbf{R}$, $\dot{\mathbf{R}}$, m, for control.
- Focus of **lecture 3**.

How to handle disturbances and uncertainties

- The subject of Robust control.
- How to estimate external disturbances and changes in the model?

How to obtain feasible feedforward?

- Pre-computed geometry-based methods, e.g., [10].
- In-real time, e.g., using Model Predictive Control [9].

How to implement and test the controllers?

- Basics are the **focus of the lab tasks**.
- In practice: code generation from toolboxes.

What did not fit into the lecture?
How to obtain states of the UAV
- We need estimates of $\mathbf{r}$, $\dot{\mathbf{r}}$, $\ddot{\mathbf{r}}$, $\mathbf{R}$, $\dot{\mathbf{R}}$, m, for control.
- Focus of **lecture 3**.
How to handle disturbances and uncertainties
- The subject of Robust control.
- How to estimate external disturbances and changes in the model?
How to obtain feasible feedforward?
- Pre-computed geometry-based methods, e.g., [10].
- In-real time, e.g., using Model Predictive Control [9].
How to implement and test the controllers?
- Basics are the **focus of the lab tasks**.
- In practice: code generation from toolboxes.

References I

[1] R. Rashad, J. Goerres, R. Aarts, J. B. Engelen, and S. Stramigioli, "Fully actuated multirotor uavs: A literature review," *IEEE Robotics & Automation Magazine*, vol. 27, no. 3, pp. 97–107, 2020.

[2] R. Rashad, D. Bicego, R. Jiao, S. Sanchez-Escalonilla, and S. Stramigioli, "Towards vision-based impedance control for the contact inspection of unknown generically-shaped surfaces with a fully-actuated UAV," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, 2020, pp. 1605–1612.

[3] L. Meier, D. Honegger, and M. Pollefeys, "PX4: A node-based multithreaded open source robotics framework for deeply embedded platforms," in *2015 IEEE International Conference on Robotics and Automation*, IEEE, 2015, pp. 6235–6240.

[4] S. Sun, A. Romero, P. Foehn, E. Kaufmann, and D. Scaramuzza, "A comparative study of nonlinear MPC and differential-flatness-based control for quadrotor agile flight," *IEEE Transactions on Robotics*, 2022.

[5] D. Bicego, J. Mazzetto, R. Carli, M. Farina, and A. Franchi, "Nonlinear model predictive control with enhanced actuator model for multi-rotor aerial vehicles with generic designs," *Journal of Intelligent & Robotic Systems*, vol. 100, no. 3, pp. 1213–1247, 2020.

[6] J. C. Pereira, V. J. Leite, and G. V. Raffo, "Nonlinear Model Predictive Control on SE(3) for Quadrotor Trajectory Tracking and Obstacle Avoidance," in *2019 19th International Conference on Advanced Robotics (ICAR)*, IEEE, 2019, pp. 155–160.

[7] T. Lee, M. Leoky, and N. H. McClamroch, "Geometric tracking control of a quadrotor UAV on SE(3)," in *2010 IEEE Conference on Decision and Control*, IEEE, 2010, pp. 5420–5425.

[8] T. Baca, M. Petrlik, M. Vrba, V. Spurny, R. Penicka, D. Hert, and M. Saska, "The MRS UAV System: Pushing the Frontiers of Reproducible Research, Real-world Deployment, and Education with Autonomous Unmanned Aerial Vehicles," *Journal of Intelligent & Robotic Systems*, vol. 102, no. 26, pp. 1–28, 1 May 2021.

[9] T. Baca, D. Hert, G. Loianno, M. Saska, and V. Kumar, "Model Predictive Trajectory Tracking and Collision Avoidance for Reliable Outdoor Deployment of Unmanned Aerial Vehicles," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems*, IEEE, 2018, pp. 1–8.

[10] D. Mellinger and V. Kumar, "Minimum snap trajectory generation and control for quadrotors," in *2011 IEEE International Conference on Robotics and Automation*, IEEE, 2011, pp. 2520–2525.

Thanks for listening.

Lecture 2: UAV Control
└─Conclusion
└─Conclusion

2024-09-29

Conclusion
Thanks for listening.