Last Time:

- Stability of spinning Bodies
- Numerical simulation w/ 3D Rotation

Today:

- Newton-Euler dynamics
- SE(3) group
- Quadrotors
- Airplanes

Newton-Euler Dynamics

- Full translation + rotation dynamics for a rigid body
- Combine F=ma and Euler's equation

$$m{}^N\dot{v} = {}^N F$$
$$J{}^B\dot{\omega} + {}^B\omega \times J{}^B\omega = {}^B\tau$$

- Still some options. Two common choices of state:

$$x = \begin{bmatrix} {}^N r \\ {}^N q^B \\ {}^N v \\ {}^B \omega \end{bmatrix} \begin{matrix} \leftarrow \text{position (N frame)} \\ \leftarrow \text{attitude } (B \to N) \\ \leftarrow \text{velocity (N frame)} \\ \leftarrow \text{angular velocity (B-frame)} \end{matrix}$$

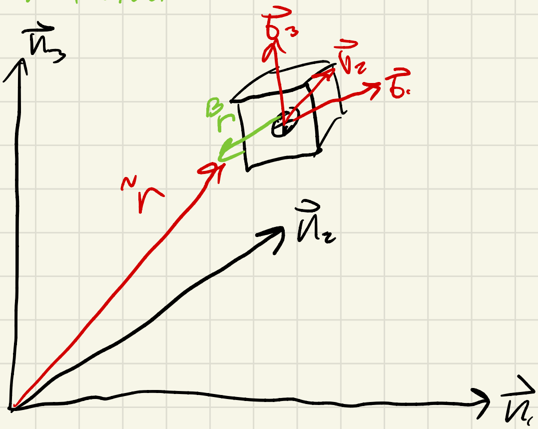
$$x = \begin{bmatrix} {}^N r \\ {}^N q^B \\ {}^B v \\ {}^B \omega \end{bmatrix} \begin{matrix} \text{"} \\ \text{"} \\ \leftarrow \text{velocity (B-frame)} \\ \text{"} \end{matrix}$$

- For simulation really doesn't matter, but for control + estimation the 2nd one is often preferred.

Special Euclidean Group SE(3)

- Matrix Representation

$$
{}^N T^B = \begin{bmatrix} \underbrace{{}^N Q^B}_{\text{rotation}} & \overbrace{{}^N d}^{\text{CoM translation}} \\ 0 & 1 \end{bmatrix}
$$



$$
\begin{bmatrix} {}^N Q^B & {}^N d \\ 0 & 1 \end{bmatrix} \begin{bmatrix} {}^B r \\ 1 \end{bmatrix} = \begin{bmatrix} Qr + d \\ 1 \end{bmatrix}
$$

- SE(3) Kinematics / Velocity:

$$
T' = T \Delta T = \begin{bmatrix} {}^N Q^B & {}^N d \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \Delta Q & \Delta d \\ 0 & 1 \end{bmatrix}
$$

$$
= \begin{bmatrix} \overbrace{Q \Delta Q}^{SO(3)} & \underbrace{{}^N Q^B \Delta d}_{\Delta d \text{ is in B frame!}} + {}^N d \\ 0 & 1 \end{bmatrix}
$$

- Taylor expand to 1st order:

$$\delta Q = \exp(\hat{\omega} h) \approx I + h\hat{\omega}$$

$$\delta d \approx h\,{}^{B}V$$

$$\Rightarrow T' \approx \begin{bmatrix} Q & d \\ 0 & 1 \end{bmatrix} \begin{bmatrix} (I + h\hat{\omega}) & h\,{}^{B}V \\ 0 & 1 \end{bmatrix}$$

$$\begin{bmatrix} Q + hQ\,{}^{B}\hat{\omega} & hQ\,{}^{B}V + d \\ 0 & 1 \end{bmatrix}$$

$$= \underbrace{\begin{bmatrix} Q & d \\ 0 & 1 \end{bmatrix}}_{T} + h \underbrace{\begin{bmatrix} Q & d \\ 0 & 1 \end{bmatrix}}_{T} \underbrace{\begin{bmatrix} {}^{B}\hat{\omega} & {}^{B}V \\ 0 & 1 \end{bmatrix}}_{\hat{W}}$$

- Take limit $h \to 0$

$$\frac{T' - T}{h} \approx \dot{T} = T\,\hat{w}$$

hat map for se(3)

$$w = \begin{bmatrix} {}^{B}V \\ {}^{B}\omega \end{bmatrix} \in \mathbb{R}^6 \text{ is the Lie algebra } se(3)$$

- As with so(3), for constant w, we have

$$T(t) = T_0 \exp(t\,\hat{w})$$

* Full Newton-Euler Dynamics on SE(3)

$$\dot{x} = \begin{bmatrix} {}^N\dot{r} \\ \dot{q} \\ {}^B\dot{v} \\ {}^B\dot{\omega} \end{bmatrix} = \begin{bmatrix} Q\,{}^Bv \\ \frac{1}{2} L(q)\, H\,{}^B\omega \\ \frac{1}{m}\,{}^BF - {}^B\omega \times {}^Bv \\ J^{-1}\left({}^B\tau - {}^B\omega \times J\,{}^B\omega\right) \end{bmatrix}$$

$${}^Nv = Q\,{}^Bv \Rightarrow {}^N\dot{v} = \dot{Q}\,{}^Bv + Q\,{}^B\dot{v} = Q\,{}^B\hat{\omega}\,{}^Bv + Q\,{}^B\dot{v}$$
$$= \frac{1}{m}\,{}^NF$$

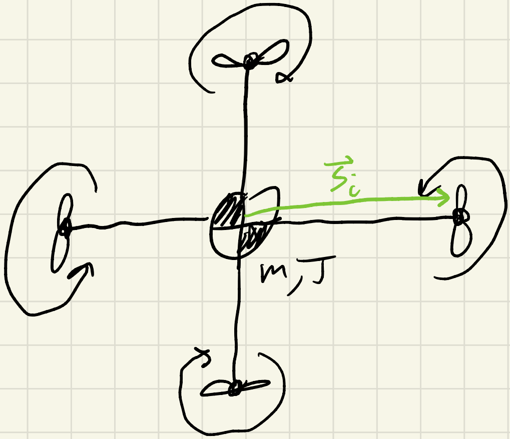
$$\Rightarrow \boxed{{}^B\dot{v} = \frac{1}{m} Q^T\,{}^NF - {}^B\omega \times {}^Bv = \frac{1}{m}\,{}^BF - {}^B\omega \times {}^Bv}$$

- Similarly, ${}^Nv$ instead of ${}^Bv$:

$$\dot{x} = \begin{bmatrix} {}^N\dot{r} \\ \dot{q} \\ {}^N\dot{v} \\ {}^B\dot{\omega} \end{bmatrix} = \begin{bmatrix} {}^Nv \\ \frac{1}{2} L(q)\, H\,{}^B\omega \\ \frac{1}{m}\,{}^NF \\ J^{-1}\left({}^B\tau - {}^B\omega \times J\,{}^B\omega\right) \end{bmatrix}$$

- Now we just need to compute F and τ to simulate any rigid-body system.

Quadrotors:



- Each prop exerts a force (thrust) in $\vec{b}_3$ direction and torque in the $\pm\vec{b}_3$ direction due to drag

$$F_i = \underbrace{K_T}_{\text{"Thrust Coefficient"}} u_i, \quad \tau_i = \underbrace{K_m}_{\text{"moment Coefficient"}} u_i$$

- Since F and τ are linear in $u \in \mathbb{R}^4$ we can write this as a matrix:

$$^B F = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ K_T & K_T & K_T & K_T \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \end{bmatrix} + \underbrace{Q^T \begin{bmatrix} 0 \\ 0 \\ -mg \end{bmatrix}}_{\text{gravity in body frame}}$$

- Remember we also get $s_i \times F$ torque due to props being offset from CoM:

$$^B\tau = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ K_m & -K_m & K_m & -K_m \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \end{bmatrix} + \sum_i {}^B s_i \times \begin{bmatrix} 0 \\ 0 \\ K_T \end{bmatrix} u_i$$

- Assuming ${}^B s_i$ are aligned with $\vec{b}_i$ axes:

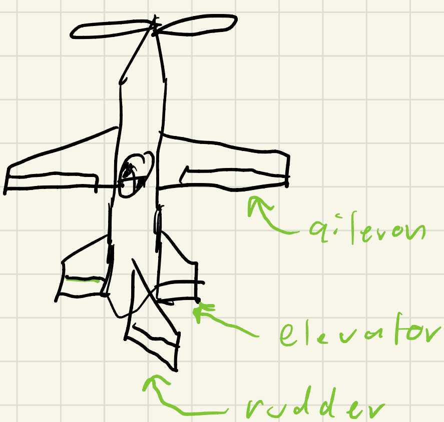
$$s_1 = \begin{bmatrix} s_1 \\ 0 \\ 0 \end{bmatrix} \quad \|s_1\| \qquad s_2 = \begin{bmatrix} 0 \\ s_2 \\ 0 \end{bmatrix} \qquad s_3 = \begin{bmatrix} -s_3 \\ 0 \\ 0 \end{bmatrix} \qquad s_4 = \begin{bmatrix} 0 \\ -s_4 \\ 0 \end{bmatrix}$$

$$\Rightarrow {}^B\tilde{\tau} = \begin{bmatrix} 0 & s k_T & 0 & -s k_T \\ -s k_T & 0 & s k_T & 0 \\ k_m & -k_m & k_m & -k_m \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \end{bmatrix}$$

- Now just plug into Newton-Euler dynamics.

- Note we ignored lots of aerodynamics: body drag, wake/boundary effects. These start to matter if you go fast and/or fly near walls/floor.

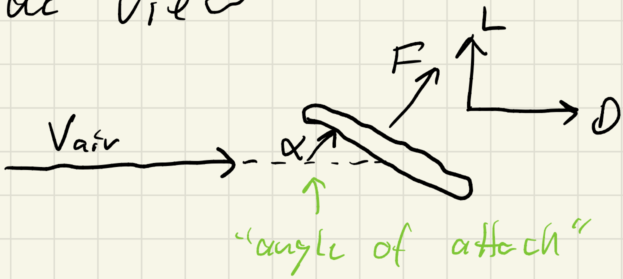
* Airplane:

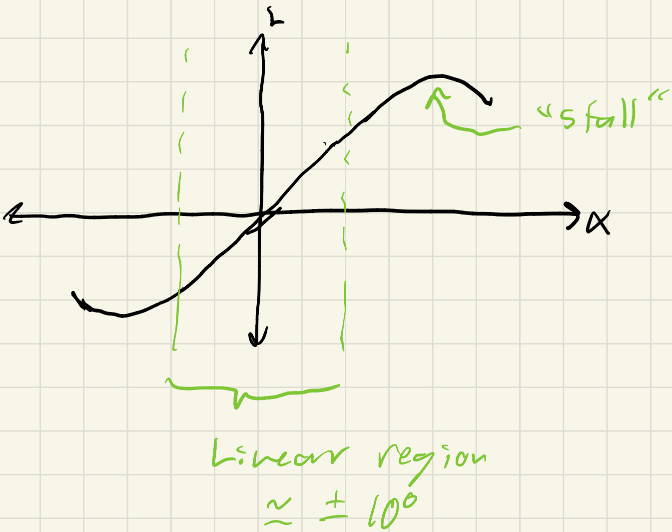


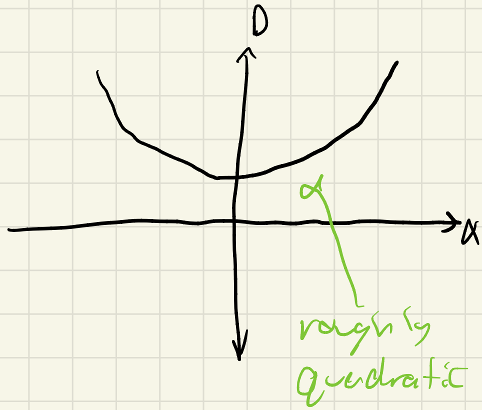
$$u = \begin{bmatrix} \text{throttle} \\ \text{ailerons} \\ \text{elevator} \\ \text{rudder} \end{bmatrix} \in \mathbb{R}^4$$

- Prop model is same as quadrotor
- Each lifting surface (wings, horizontal + vertical stabilizers) generate lift + drag:

side view







- For most applications, we can use linear approximations of $L(\alpha)$ and quadratic approximations of $D(\alpha)$

$$L \approx \frac{1}{2} \rho A v^2 C_{L_0} \alpha$$

air density, wing area, velocity, Lift Coefficient

$$D \approx \frac{1}{2}\rho A V^2 \left[C_{D_0} + C_{D_2} \alpha^2 \right]$$

Drag Coefficient

- Flaps just change effective α:

$$\alpha_{eff} = \alpha + \varepsilon_i u_i$$

control input = flap deflection angle

"flap effectiveness"

- Now just add up forces + torques from all lifting surfaces + prop
- Model breaks at high α (e.g. aerobatics)
- Also need to include prop wash

* Other common (single) rigid-body systems

  - Underwater robots
  - Spacecraft
  - Wheeled vehicles / cars
  - Legged robots