Last Time:

- Stability of Spinning Bodies
- Numerical Simulation w/ 3D Rotations

Today:

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

Newton-Euler Dynamics

- Full translation + rotation dynamics combine $F=ma$ and Euler's equation:

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

- Still some options. Two common choices of state:

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

or

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

- For simulation, really doesn't matter but for controllers and estimators the 2nd is often preferred.

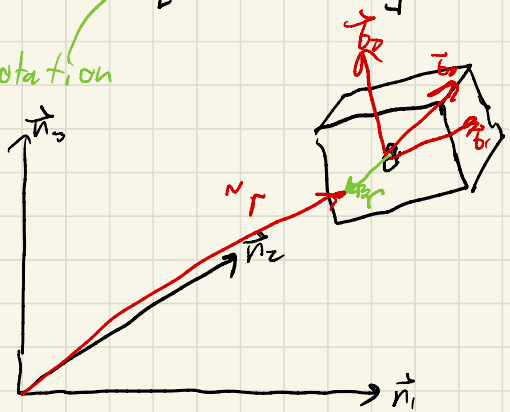
* Special Euclidean Group $SE(3)$:

- Matrix representation:

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

← translation of CoM

rotation



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

- $SE(3)$ Kinematics / Velocity:

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

$$= \begin{bmatrix} \underbrace{Q\,\Delta Q}_{\text{same as } SO(3)} & {}^{N}Q^{B}\Delta d + {}^{N}d \\ 0 & \end{bmatrix}$$

Δd is in B frame!

- Taylor expand to 1st order:

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

$$\Delta d \approx h\,{}^{B}v$$

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

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

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

$$T + hTW$$

- Take $\lim h \to 0$

$$\Rightarrow \dot{T} = TW = T\hat{\mathcal{W}}$$

$\hat{\mathcal{W}}$: hat map for $se(3)$

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

- As with $SO(3)$, for constant $\mathcal{W}$ we have:

$$T(t) = T_0 \exp(\hat{\mathcal{W}} t)$$

Full Newton-Euler Dynamics on $SE(3)$

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

$${}^NV = Q\,{}^BV \Rightarrow {}^N\dot{V} = \dot{Q}\,{}^BV + Q\,{}^B\dot{V} = Q\,{}^B\hat{\omega}\,{}^BV + Q\,{}^B\dot{V} = \frac{1}{m}{}^NF$$

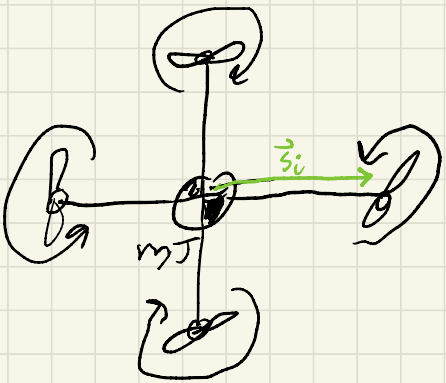
$$\Rightarrow \boxed{{}^B\dot{V} = \frac{1}{m}Q^T\,{}^NF - {}^B\omega \times {}^BV = \frac{1}{m}{}^BF - {}^B\omega \times {}^BV}$$

- Similarly, with ${}^NV$ instead of ${}^BV$:

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

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

Quadrotor:



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

$$F_i = k_T u_i, \quad \tau_i = k_m u_i$$

"Thrust coefficient" — "moment coefficient"

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

$$^B F = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ k_T & k_T & k_T & k_T \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \end{bmatrix} + G^T \begin{bmatrix} 0 \\ 0 \\ -mg \end{bmatrix}$$

gravity in body frame

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

$$s_1 = \begin{bmatrix} s \\ 0 \\ 0 \end{bmatrix}, \quad s_2 = \begin{bmatrix} 0 \\ s \\ 0 \end{bmatrix}, \quad s_3 = \begin{bmatrix} -s \\ 0 \\ 0 \end{bmatrix}, \quad s_4 = \begin{bmatrix} 0 \\ -s \\ 0 \end{bmatrix}$$

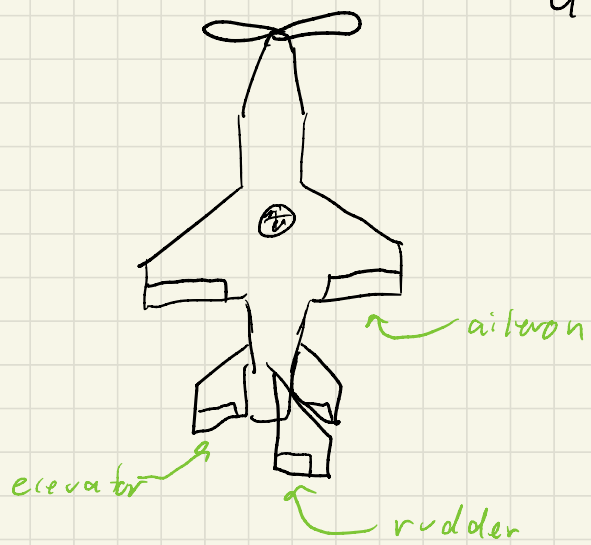
$s = \|\vec{s}_i\|$

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

- Now just plug into Newton-Euler dynamics

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

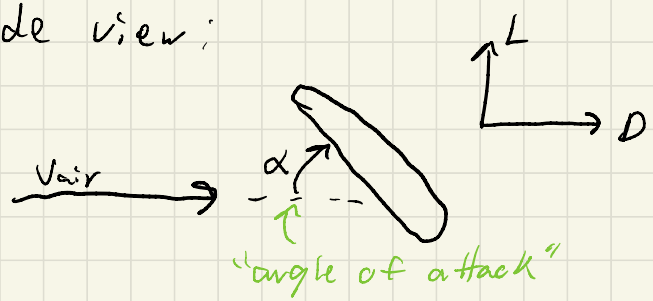
* Airplane:

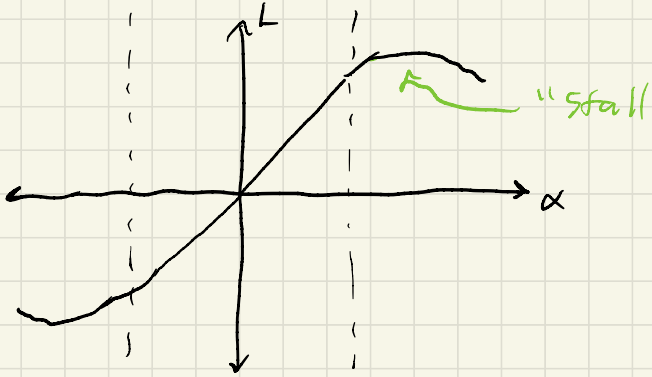


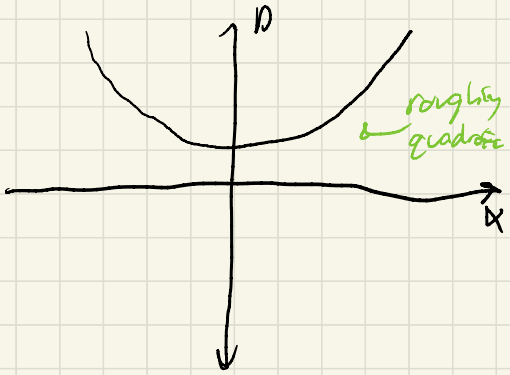
$$u = \begin{bmatrix} \text{throttle} \\ \text{ailerons} \\ \text{elevator} \\ \text{rudder} \end{bmatrix} \in \mathbb{R}^4$$

- Propeller model is same as quadrotor
- Each lifting surface (wing, horizontal + vertical stabilizers) generates lift and drag:

side view:







- For most applications, we can use linear approximations of L and D as functions of α

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

(C_{L_α}: lift coefficient; $C_{D_0} + C_{D_2}\alpha^2$: drag coefficient)

(ρ: air density; A: wing area; V: velocity)

- Flaps just change effective angle of attack:

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

(u_i: control input = flap deflection angle; ε_i: "flap effectiveness")

- Now just add up forces + torques from all lifting surfaces + prop

- Model breaks at high α (e.g. aerobatics) doesn't includ prop wash.

* Other Common (Single) Rigid-Body Systems

- Underwater robots / submarines
- Spacecraft / rockets
- Wheeled vehicles / cars
- Legged robots (centroidal dynamics)