

Last Time:

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

Today:

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

## Newton-Euler Dynamics

- Full translation + rotation dynamics combine  $\ddot{r} = \dot{\omega} \times r$  and Euler's equation:

$$\begin{aligned} m^N \ddot{v} &= {}^N F \\ J^B \ddot{\omega} + {}^B \omega \times J^B \omega &= {}^B \tau \end{aligned}$$

- Still some options. Two common choices of state:

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

OR

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

- 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

- $SE(3)$  Kinematics / Velocity:

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

same as  $SO(3)$

$\Delta d$  is in B frame!

- Taylor expand to 1<sup>st</sup> order:

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

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

$$\begin{aligned}
 \Rightarrow T' &\approx \begin{bmatrix} Q & d \\ 0 & 1 \end{bmatrix} \begin{bmatrix} (I + h\hat{\omega}) & h^B v \\ 0 & 1 \end{bmatrix} \\
 &= \begin{bmatrix} Q + hQ^B\hat{\omega} & hQ^B v + "d" \\ 0 & 1 \end{bmatrix} \\
 &= \underbrace{\begin{bmatrix} Q & "d" \\ 0 & 1 \end{bmatrix}}_{T} + h \underbrace{\begin{bmatrix} Q & "d" \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \hat{\omega} & B_v \\ 0 & 0 \end{bmatrix}}_w
 \end{aligned}$$

- Take  $\lim h \rightarrow 0$

$$\Rightarrow \dot{T} = TW = T \underbrace{\hat{w}}_{\text{That map for } se(3)}$$

$w = \begin{bmatrix} B_v \\ B_w \end{bmatrix} \in \mathbb{R}^6$  is the Lie algebra  $se(3)$

- As with  $SO(3)$ , for constant  $w$  we have:

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

# \* Full Newton-Euler Dynamics on SE(3)

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

${}^N V = Q^B V \Rightarrow {}^N \dot{V} = \dot{Q}^B V + Q^B \dot{V} = Q^B \hat{w} {}^B V + Q^B \dot{V} = \frac{1}{m} {}^N F$

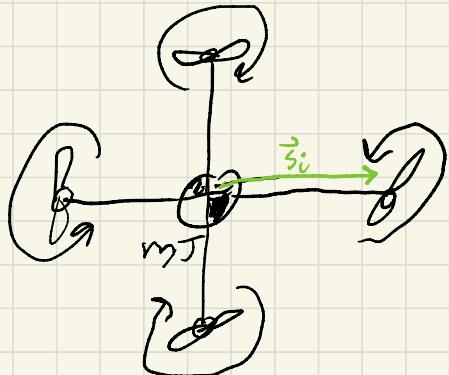
$\Rightarrow {}^B \dot{V} = \frac{1}{m} Q^T {}^N F - {}^B \omega \times {}^B V = \frac{1}{m} {}^B F - {}^B \omega \times {}^B V$

- Similarly, with  ${}^N V$  instead of  ${}^B V$ :

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

- Now we just need to compute  $F$  and  $\tau$  to simulate any rigid body system.

Quadrotor:



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

$$F_i = K_T U_i, \quad \tau_i = K_m U_i$$

$y$        $y$   
"Thrust"      "moment"  
"coefficient"      "coefficient"

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

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

gravity in body frame

- Remember we also get  ${}^B \tau_i \times F$  torque due to props being offset from  ${}^B M$ :

$${}^B \tau = \begin{bmatrix} 0 & 0 & 0 & C \\ 0 & 0 & 0 & 0 \\ km & -km & km & -km \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \end{bmatrix} + \sum_i {}^B \tau_i \times \begin{bmatrix} 0 \\ 0 \\ u_T \end{bmatrix} u_i$$

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

$$\tau_1 = \begin{bmatrix} s \\ 0 \\ 0 \end{bmatrix}, \quad \tau_2 = \begin{bmatrix} 0 \\ s \\ 0 \end{bmatrix}, \quad \tau_3 = \begin{bmatrix} 0 \\ 0 \\ s \end{bmatrix}, \quad \tau_4 = \begin{bmatrix} 0 \\ -s \\ 0 \end{bmatrix}$$

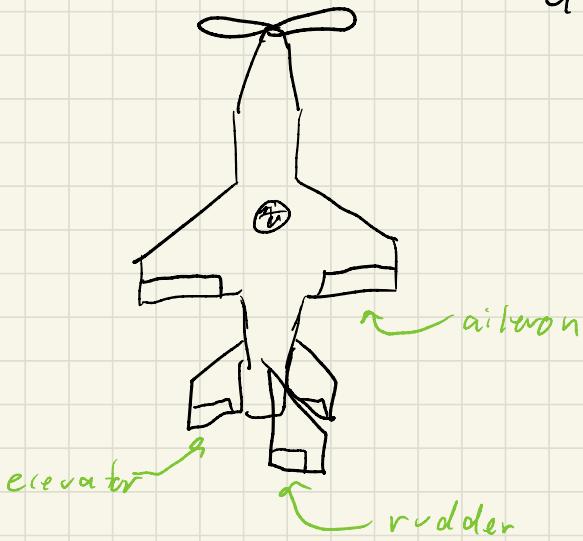
$s = ||\vec{\tau}_i||$

$$\Rightarrow {}^B \tau = \begin{bmatrix} 0 & SK_T & C & -SK_T \\ -SK_T & 0 & SK_T & 0 \\ km & -km & km & -km \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \end{bmatrix}$$

- 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 aggressive flying or fly near walls/floor.

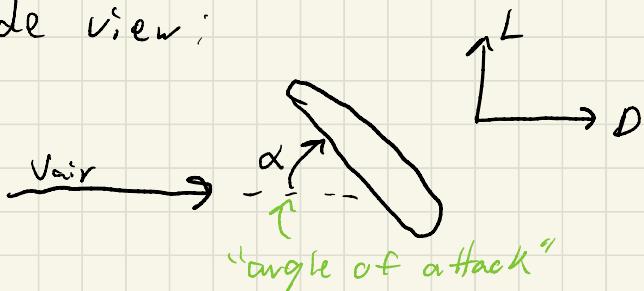
\* Airplane:

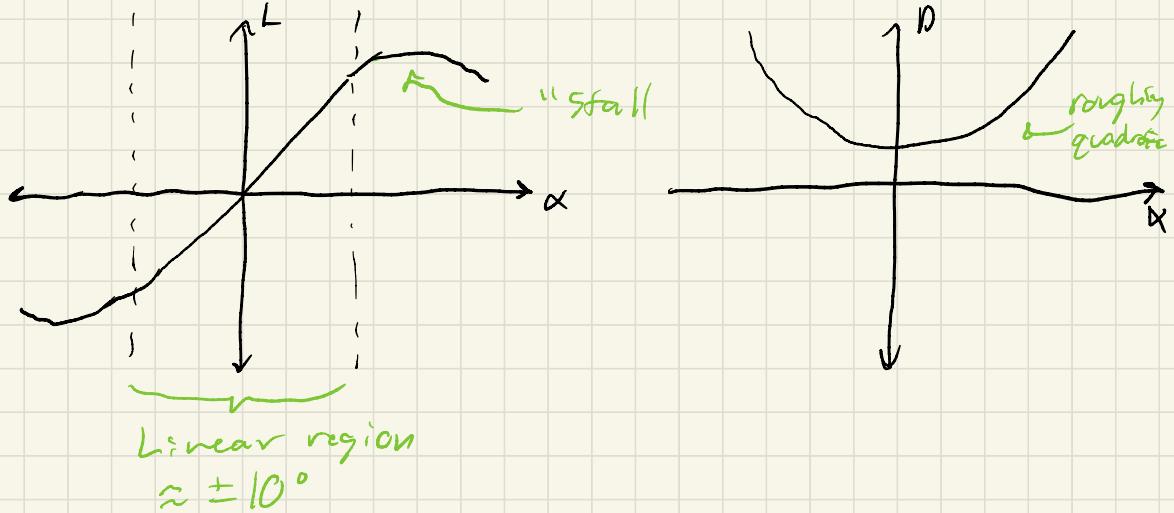


$$U = \begin{bmatrix} \text{throttle} \\ \text{aileron} \\ \text{elevator} \\ \text{rudder} \end{bmatrix} \in \mathbb{R}^4$$

- 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  $\alpha$

$$L \approx \frac{1}{2} \rho A V^2 C_{L0} \alpha, \quad D = \frac{1}{2} \rho A V^2 [C_{D0} + C_{D\alpha} \alpha]$$

air density  $\nearrow$  wing area  $\nearrow$  velocity  $\nearrow$   
 Lift coefficient  $\nearrow$   
 Drag coefficient  $\nearrow$

- Flaps just change effective angle of attack!

$$\alpha_{\text{eff}} = \alpha + \sum_i \epsilon_i u_i \quad \leftarrow \begin{array}{l} \text{control input = flap deflection} \\ \text{angle} \end{array}$$

$\nearrow$  "flap effectiveness"

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

- Model breaks at high  $\alpha$  (e.g. aerobatics) doesn't include prop wash -

## \* Other Common (Single) Rigid Body Systems

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