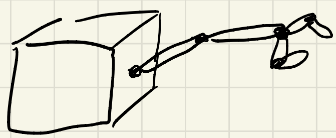
Last Time:

- Euler's Equation from Least Action
- Intro to floating-base systems

Today:

- Floating base Kinematics + Jacobian
- Floating-base dynamics in joint/generalized
- " " " in Maximal coordinates

Floating-Base Kinematics:



- Using joint coordinates, we write the configuration as:

$$q = \begin{bmatrix} r_0 \\ Q_0 \\ q_1 \\ \vdots \\ q_n \end{bmatrix} \begin{matrix} \} \text{ Floating base position + attitude} \\ \\ \} \text{ Joint angles} \end{matrix}$$

- Assume we have a Kinematics function:

$$\begin{bmatrix} r_0 \\ Q_0 \\ r_1 \\ Q_1 \\ \vdots \\ r_n \\ Q_n \end{bmatrix} = K(q)$$

full SE(3) pose of each link

- We want a kinematic Jacobian that maps:

$$\begin{bmatrix} \dot{r}_0 \\ \omega_0 \\ \dot{r}_1 \\ \omega_1 \\ \vdots \\ \dot{r}_n \\ \omega_n \end{bmatrix} = K(q) \underbrace{\begin{bmatrix} \dot{r}_0 \\ \omega_0 \\ \dot{q}_1 \\ \vdots \\ \dot{q}_n \end{bmatrix}}_{v}$$

- The naive Jacobian of $k(q)$ maps:

$$\begin{bmatrix} \dot{r}_0 \\ \dot{Q}_0 \\ \dot{r}_1 \\ \dot{Q}_1 \\ \vdots \\ \dot{r}_n \\ \dot{Q}_n \end{bmatrix} = \frac{\partial K}{\partial q} \underbrace{\begin{bmatrix} \dot{r}_0 \\ \dot{Q}_0 \\ \dot{q}_1 \\ \vdots \\ \dot{q}_n \end{bmatrix}}_{\dot{q}}$$

note $v \neq \dot{q}$

- Remember quaternion kinematics:

$$\dot{Q} = \frac{1}{2} L(Q) H \omega \Rightarrow \omega = 2 H^T L^T(Q) \dot{Q}$$

$$= \frac{1}{2} G(Q) \omega \qquad\qquad = 2 G^T(Q) \dot{Q}$$

- We can get the Jacobian we want by sandwiching $\frac{\partial k}{\partial q}$:

$$\begin{bmatrix} \dot{r}_0 \\ \omega_0 \\ \vdots \\ \dot{r}_n \\ \omega_n \end{bmatrix} = \underbrace{\begin{bmatrix} I & & & \\ & 2G^T(Q_0) & & \\ & & \ddots & \\ & & & I \\ & & & & 2G^T(Q_n) \end{bmatrix} \frac{\partial K}{\partial q} \begin{bmatrix} I & & & \\ & \frac{1}{2}G(Q_0) & & \\ & & \ddots & \\ & & & I \\ & & & & \ddots \end{bmatrix}}_{K(q)} V$$

- Note factors of 2 from quaternion kinematics
- From here, we can get the mass matrix:

$$M(q) = K(q)^T \bar{M} K(q), \quad \bar{M} = \begin{bmatrix} m_0 I & & & & 0 \\ & J_0 & & & \\ & & \ddots & & \\ & & & m_n I & \\ 0 & & & & J_n \end{bmatrix}$$

$m_i \in \mathbb{R}$ (mass of each link)

$J_i \in \mathbb{R}^{3\times3} > 0$ (inertia matrix of each link)

Floating-Base Dynamics:

- We want an "Euler-Lagrange Equation" for floating-base systems.

- Lagrangian has the usual form:

$$L(q,v) = \frac{1}{2} v^T M(q) v - V(q)$$

- Introduce explicit quaternion kinematics constraint:

$$S = \int_{t_0}^{t_f} L(q,v) + \underbrace{\ell^T}_{\text{Lagrang multiplier}} [\underbrace{2H^T(Q_0^+ * \dot{Q}_0) - \omega_0}_{\text{Quaternion kinematics}}] \, dt$$

- We need to set derivatives w.r.t. $q(t)$, $v(t)$, $\ell(t) = 0$

- Derivatives that don't involve the quaternion look like standard EL equation:

$$\delta_{r_0} S = \int_{t_0}^{t_f} \frac{\partial L}{\partial r_0} \delta r_0 + \frac{\partial L}{\partial \dot{r}_0} \delta \dot{r}_0 \, dt = 0$$

$$= \int_{t_0}^{t_f} \frac{\partial L}{\partial r_0} \delta r_0 - \frac{d}{dt}\left(\frac{\partial L}{\partial \dot{r}_0}\right) \delta r_0 \, dt = 0$$

$$\Rightarrow \boxed{\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{r}_0}\right) - \frac{\partial L}{\partial r_0} = 0} \quad (1)$$

- Same for joint angles:

$$\boxed{\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}_{1:n}}\right) - \frac{\partial L}{\partial q_{1:n}} = 0} \quad (2)$$

$$\delta_{\omega_0} S = \int_{t_0}^{t_f} \left[\frac{\partial L}{\partial \omega_0} - l^T\right] \delta \omega_0 \, dt$$

$$\Rightarrow \boxed{l = \nabla_{\omega_0} L(q,v)} \quad (3)$$

- The derivative w.r.t. Q_0 is messy but essentially the same as what we did for Euler's eqn. (see previos notes)

$$\Rightarrow \dot{l} + \omega \times l - \tfrac{1}{2} G^T(Q_0) \nabla_{Q_0} L(q,v) = 0$$

$$\Rightarrow \boxed{\frac{d}{dt}\left(\nabla_{\omega_0} L(q,v)\right) + \omega \times \left(\nabla_{\omega_0} L(q,v)\right) - \tfrac{1}{2} G^T(Q_0) \nabla_{Q_0} L(q,v) = 0} \quad (4)$$

- Equations (1), (2), and (4) give the dynamics of a floating-base system in joint coordinates.

- This is a special case of Hamel's equations and Kane's equations.

- We can easily write this in manipulator form:

$$M(q)\dot{v} + C(q,v) + \underbrace{G(q)}_{\text{(not attitude jacobian)}} = B(q)u$$

- This is the standard setup in most robotics simulators (e.g. Bullet, MuJoCo, etc.)