Last Time:

- Fixed-Base Manipulators
- Forward kinematics in 2D + 3D
- Differentiating Quaternions

Today:

- Floating-Base Manipulators

Floating-Base Systems

- We want to handle robots not bolted to the ground, e.g. mobile/legged/aerial/underwater/space
- Essentially want to combine manipulator/Lagrangian stuff with $SE(3)$ stuff for base.
- To ease into this, lets first look at a single rigid body from the Lagrangian perspective
- Assuming no forces, the Lagrangian is:

$$L = \tfrac{1}{2} m v^T v + \tfrac{1}{2} \omega^T J \omega$$

- If we naively apply EL eqn. to ω:

$$\cancel{\frac{d}{dt}\left(\frac{\partial L}{\partial \omega}\right) = J\dot{\omega} = 0}$$

- We know the right answer (Euler's eqn.)
- The standard EL eqn. doesn't work because $\omega \neq \dot{q}$

- We can still use least-action with a constraint:

$$\min_{\substack{Q(t)\\ \omega(t)}} \int_{t_0}^{t_f} \omega^T J \omega \, dt$$

$$\text{s.t.} \quad \underbrace{\dot{Q} = \tfrac{1}{2} Q * (H\omega)}$$ } Quaternion Kinematics

$$\Downarrow$$

$$\omega = 2H^T(Q^+ * \dot{Q})$$

- FON Conditions:

$$\frac{\partial}{\partial \begin{bmatrix} Q \\ \omega \\ \ell \end{bmatrix}} \left[\int_{t_0}^{t_f} \tfrac{1}{2}\omega^T J \omega + \underbrace{\ell^T}_{\text{Lagrange multiplier}} [2H^T(Q^+ * \dot{Q}) - \omega] \, dt \right]$$

- The derivative w.r.t. ω is easy:

$$\int_{t_0}^{t_f} \omega^T J - \ell^T \, dt = 0 \Rightarrow \underbrace{\ell}_{\text{angular momentum}} = J\omega$$

- For the next part we need some quaternion identities:

$$Q^+ * \dot{Q} = L(Q^+)\dot{Q} = L^T(Q)\dot{Q} = L^{-1}(Q)\dot{Q}$$

$$= R(\dot{Q})Q^+ = R(\dot{Q})TQ, \quad T = \begin{bmatrix} 1 & 0 \\ 0 & -I \end{bmatrix}$$

- We also need to differentiate these quaternions:

$$\delta Q = G(Q)\,\delta\phi = L(Q)H\,\delta\phi$$

$$\Rightarrow \delta Q^{+} = T\,\delta Q = T\,G(Q)\,\delta\phi = T\,L(Q)H\,\delta\phi$$

$$\Rightarrow \delta \dot{Q} = G(Q)\,\delta\dot{\phi} + G(\dot{Q})\,\delta\phi$$

$$= L(Q)H\,\delta\dot{\phi} + L(\dot{Q})H\,\delta\phi$$

- Putting this together:

$$\frac{\partial}{\partial Q(t)}\left[\int_{t_0}^{t_f} \tfrac{1}{2}\omega^T J \omega + \lambda^T\left[2H^T(Q^{+} * \dot{Q}) - \omega\right] dt\right]$$

$$= \int_{t_0}^{t_f} \lambda^T\left[2H^T\left(\delta Q^{+} * \dot{Q} + Q^{+} * \delta\dot{Q}\right)\right] dt$$

$$= \int_{t_0}^{t_f} \lambda^T\Big[2H^T\Big(\underbrace{R(\dot{Q})TL(Q)H\delta\phi}_{(Q * H\delta\phi)^{+} * \dot{Q}}$$

$$(Q * H\delta\phi)^{+} * \dot{Q} = -H\delta\phi * Q^{+} * \dot{Q} = -(H\delta\phi) * (\tfrac{1}{2}H\omega)$$

$$+ \underbrace{L^T(Q)\big(L(Q)}_{I}H\,\delta\dot{\phi} + \underbrace{L(\dot{Q})H\delta\phi}_{\frac{1}{2}(H\omega) * (H\delta\phi)}\big)\Big)\Big]dt$$

- Recall:

$$Q_1 * Q_2 = \begin{bmatrix} s_1 s_2 - v_1^T v_2 \\ s_1 v_2 + s_2 v_1 + v_1 \times v_2 \end{bmatrix}$$

$\Rightarrow$ we can turn all the quaternion products above into vector cross products

$$= \int_{t_0}^{t_f} 2 l^T (\omega \times \delta\phi + \delta\dot{\phi})\, dt$$

$$= \int_{t_0}^{t_f} 2 l^T (\omega \times \delta\phi) - 2\dot{l}^T \delta\phi \, dt$$

Recall:

$$\frac{d}{dt}\left(a(t) \cdot b(t)\right) = \dot{a}\, b + a\dot{b}$$

$$\Downarrow$$

$$\int_{t_0}^{t_f} \frac{d}{dt}\left(a(t) \cdot b(t)\right) dt = \left(a(t) \cdot b(t)\right)\Big|_{t_0}^{t_f} = \int_{t_0}^{t_f} \dot{a} b \, dt + \int_{t_0}^{t_f} a \dot{b}\, dt$$

$$\Rightarrow \int_{t_0}^{t_f} a \dot{b}\, dt = -\int_{t_0}^{t_f} \dot{a} b \, dt + (ab)\Big|_{t_0}^{t_f}$$

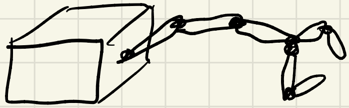
$$= \int_{t_0}^{t_f} 2\Big[l^T \underbrace{\hat{\omega}}_{\text{skew-symmetric cross-product matrix}} - \dot{l}^T\Big] \delta\phi \, dt = 0$$

$$\Rightarrow \dot{l} + \omega \times l = 0$$

$$\Rightarrow \boxed{J\dot{\omega} + \omega \times J\omega = 0}$$

- This is a special case of the Euler-Poincaré equation, which generalizes the EL equation to Lie groups.

- Now let's look at adding more links + joints!



- Forward kinematics don't really change, we just have a position + orientation for the base now:

$$ {}^{N}R_{0}^{B} = H^T L(Q_0) R^T(Q_0) H $$

$$ {}^{N}r_{0} = r_0 $$

These are in your state vector

$$ R_n = R_{n-1} S_n(q_n) $$

$$ r_n = r_{n-1} + R_{n-1} b_{n-1} + R_n a_n $$

- We can define a kinematics function:

$$ \begin{bmatrix} r_0 \\ Q_0 \\ r_1 \\ Q_1 \\ \vdots \\ r_n \\ Q_n \end{bmatrix} = K(q), \quad q = \begin{bmatrix} r_0 \\ Q_0 \\ q_1 \\ \vdots \\ q_n \end{bmatrix} $$

- We can get the kinematic Jacobian like before using the attitude Jacobian in the right places!

$$\begin{bmatrix} \dot{r}_0 \\ \omega_0 \\ \dot{r}_1 \\ \omega_1 \\ \vdots \\ \dot{r}_n \\ \omega_n \end{bmatrix} = \underbrace{\begin{bmatrix} I & & & & & & \\ & G^T(Q_0) & & & & 0 & \\ & & I & & & & \\ & & & G^T(Q_1) & & & \\ & 0 & & & \ddots & & \\ & & & & & I & \\ & & & & & & G^T(Q_n) \end{bmatrix} \frac{\partial k}{\partial q} \begin{bmatrix} I & & 0 \\ & G(Q_0) & \\ 0 & & I \\ & & \end{bmatrix}}_{K(q)} \begin{bmatrix} \dot{r}_0 \\ \omega_0 \\ \dot{q}_1 \\ \vdots \\ \dot{q}_n \end{bmatrix}$$

- Note $v \neq \dot{q}$

- We can get $M(q)$ like before:

$$T = \sum_{n=1}^{n} \frac{1}{2} \begin{bmatrix} \dot{r}_n \\ \omega_n \end{bmatrix}^T \begin{bmatrix} mI & 0 \\ 0 & J \end{bmatrix} \begin{bmatrix} \dot{r}_n \\ \omega_n \end{bmatrix} = \frac{1}{2} v^T \underbrace{K(q)^T \bar{M} K(q)}_{M(q)} v$$

$$\bar{M} = \begin{bmatrix} m_0 I & & & & \\ & J_0 & & 0 & \\ & & \ddots & & \\ & 0 & & m_n I & \\ & & & & J_n \end{bmatrix}$$