ME 221 lecture 12 11/8/22

Last time: $\dot{x} = J(q)\,\dot{q}$ (Jacobian)

$J(q)$ provides <u>important information</u>

① Singularities $\text{rank}(J(q)) <$ # of dof

② Manipulability $A = JJ^T \Rightarrow A$ tells us:

① $J(q) = \dot{T}(q)\,T^{-1}(q)$

② $J_R = \dot{R}R^T$

$J_D = \left[\frac{\partial}{\partial q_1}\vec{d} \;\cdots\cdots\; \frac{\partial}{\partial q_n}\vec{d}\right]$

① from the eigenvectors the principle semi-axes directions

② from the eigenvalues the length of each semi axes

Summary of Geometric Methods for the Jacobian.

① Jacobian generating vectors

$$\begin{bmatrix} J_R \\ J_D \end{bmatrix} = \left[\underbrace{\begin{matrix} {}^0\hat{k}_0 \\ {}^0\hat{k}_0 \times {}^0_0\vec{d}_n \end{matrix}}_{c_i} \;\middle|\; \cdots\cdots \;\middle|\; \begin{matrix} {}^0\hat{k}_{n-1} \\ {}^0\hat{k}_{n-1} \times {}^0_{n-1}\vec{d} \end{matrix} \right]$$

$$c_i = \begin{bmatrix} {}^0\hat{k}_{i-1} \\ {}^0\hat{k}_{i-1} \times {}^0_{i-1}\vec{d}_n \end{bmatrix}$$

${}^0_{i-1}\vec{d}_n$: origin of the end-effector frame with respect to the $i-1$ frame, written in the Global frame

${}^0\hat{k}_{i-1}$: unit vector in the direction of the joint axis i, expressed in global frame

Revolute Joint

$$c_i = \begin{bmatrix} {}^{0}\hat{k}_{i-1} \\ {}^{0}\hat{k}_{i-1} \times {}^{0}_{i-1}\vec{d}_n \end{bmatrix}$$

Prismatic Joint

$$c_i = \begin{bmatrix} 0 \\ {}^{0}\hat{k}_{i-1} \end{bmatrix}$$

$${}^{0}\hat{k}_{i-1} = {}^{0}R_{i-1}\,{}^{i-1}\hat{k}_{i-1}$$

$${}^{0}\hat{k}_{i-1} \times {}^{0}_{i-1}\vec{d}_n = {}^{0}R_{i-1}\left({}^{i-1}\hat{k}_{i-1} \times {}^{i-1}\vec{d}_n\right)$$

Step by step: Jazar Ex. 269 pg. 519.

(2.) PoE method

if we have $T = \begin{bmatrix} R & d \\ 0 & 1 \end{bmatrix}$

define $[Ad_T] = \begin{bmatrix} R & 0 \\ \tilde{d}R & R \end{bmatrix}$ 6x6 adjoint map

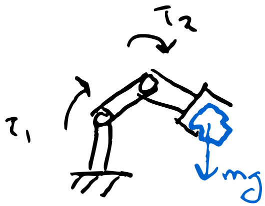
if you have $T(q) = e^{\tilde{S}_1 q_1} \cdots e^{\tilde{S}_n q_n} M$

$$J_i = Ad_{e^{\tilde{S}_1 q_1} \cdots e^{\tilde{S}_{i-1} q_{i-1}}} S_i$$

with $J_1 = S_1$

I will not ask for you to compute Jacobian unless it's simple like 2R

Statics



Conservation of Energy

$$P_{in} = P_{out}$$

$$P_{in} = \sum_i^n p_i = \sum \dot{q}_i \tau_i = \dot{q}^T \tau$$

$$\dot{q} = \begin{Bmatrix} \dot{q}_1 \\ \vdots \\ \dot{q}_n \end{Bmatrix}$$

$$\tau = \begin{Bmatrix} \tau_1 \\ \vdots \\ \tau_n \end{Bmatrix}$$

$$P_{out} = v^T \mathcal{F}$$

$$\mathcal{F} = \begin{Bmatrix} F_x \\ F_y \\ F_z \\ \tau_x \\ \tau_y \\ \tau_z \end{Bmatrix}$$

$$\therefore \dot{q}^T \tau = v^T \mathcal{F}$$

but we know $v = J(q)\,\dot{q}$

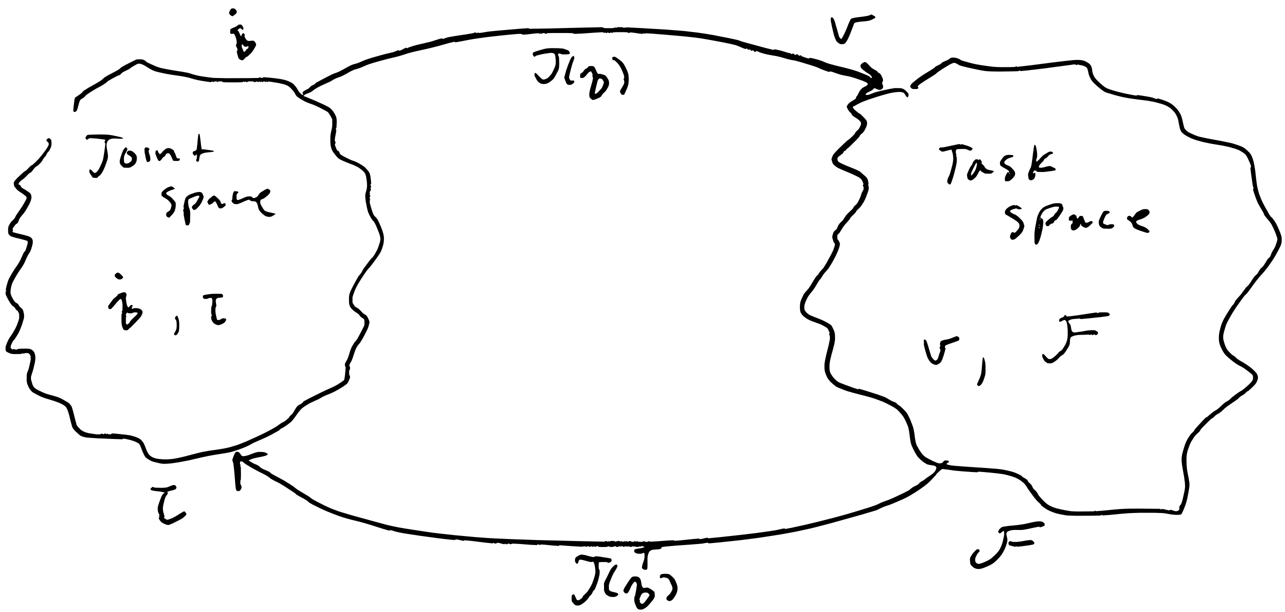
$$\dot{q}^T \tau = (J(q)\,\dot{q})^T \mathcal{F}$$

$$\underbrace{\dot{q}^T \tau = \dot{q}^T J(q)^T \mathcal{F}}$$

equal!!

$$\boxed{\tau = J(q)^T \mathcal{F}}$$

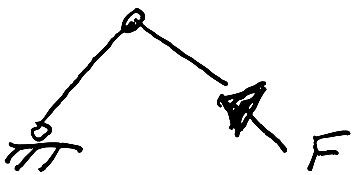
Duality between velocity & force



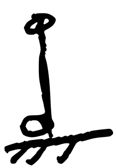
There is symmetry between Force/velocity.

Null Space

$$\text{Null}\{ J(q)^T \} = \{ F \mid J^T(q) F = 0 \}$$



vs.

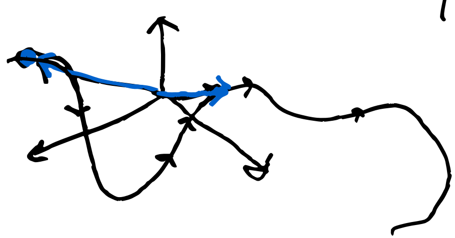


Robot Dynamics

Lynch & Park Ch. 8
Murray, Li, Sastry Ch. 4

Dynamical system: a system in which a function describes the time dependence of a point in space.

if we think about $\vec{x} = \begin{Bmatrix} x \\ y \\ z \end{Bmatrix}$



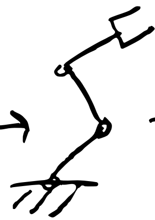
$$\dot{x}(t) = f(\bar{x}(t))$$

Equations of motion !!

For Robot Dynamics we can represent the EoM in Joint space or Task space.

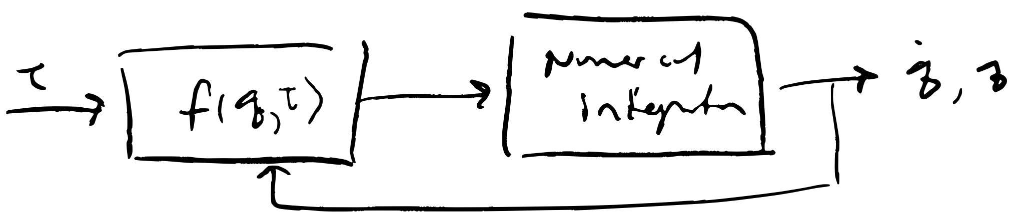
Forward Dynamics

$\tau \rightarrow \ddot{q}$ (second order because we inertia)

$\tau = \begin{bmatrix} \tau_1 \\ \vdots \\ \tau_n \end{bmatrix} \rightarrow$  $\rightarrow \ddot{q}$

$\tau \rightarrow \boxed{\text{Dynamics}} \rightarrow \ddot{q}$

We use forward dynamics for simulations ; it's very important for control.



Inverse Dynamics : $q_d, \dot{q}_d, \ddot{q}_d \longrightarrow \tau$

usually used in control algorithms, and provides the required torque give a trajectory.

$$q_d, \dot{q}_d, \ddot{q}_d \longrightarrow \boxed{\text{Inverse}} \longrightarrow \tau$$

Two Main Methods

Euler - Langrange

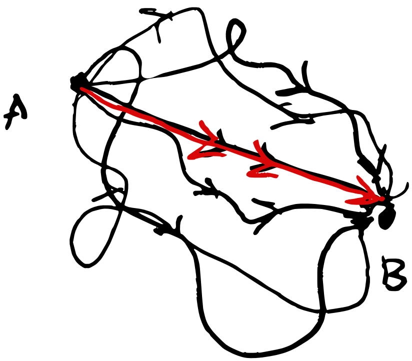
- energy based method
- very general
- find Kinectic & Potental energy.

$T(q, \dot{q}) \quad V(q)$

Newton - Euler

- force balance
- Newton's Laws
  $F = ma$
- Euler
  $\tau = I\ddot{\theta}$
- recursive algotim

Principle of Least Action



this framework is used:

- thermodynamics
- fluid mechanics
- relativity
- Quantum Mechanics

⋱

Which path would a particle take?

$$\mathcal{S} = \overbrace{\int_a^b \mathcal{L}(x, f(x), f'(x))\,dx}^{\text{action.}}$$

$f^*(x) \rightarrow$ the path that minimizes the action functional

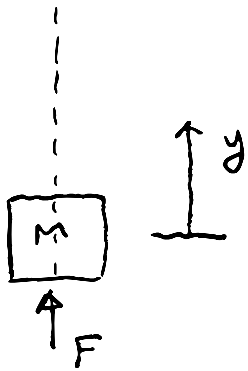
$$\mathcal{L} = \underset{\text{Kinetic energy}}{T(q, \dot{q})} - \underset{\text{Potential energy.}}{V(q)}$$

then,

$$\frac{d}{dt}\frac{\partial \mathcal{L}}{\partial \dot{q}} - \frac{\partial \mathcal{L}}{\partial q} = F_{ext}$$

the Equation of Motion just fall out.

Example: Vertical Mass



$$\mathcal{L} = T - V, \quad \frac{d}{dt}\frac{\partial \mathcal{L}}{\partial \dot{q}} - \frac{\partial \mathcal{L}}{\partial q} = F$$

$$T(q, \dot{q}) = \frac{1}{2} m \dot{y}^2$$

$$V(q) = m g y$$

$$q = y$$

$$\mathcal{L} = \frac{1}{2} m \dot{y}^2 - m g y$$

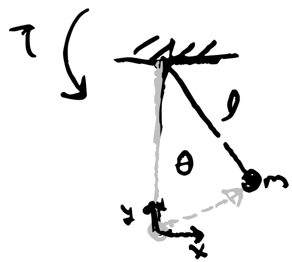
$$\frac{\partial \mathcal{L}}{\partial \dot{y}} = m \dot{y} \Rightarrow \frac{d}{dt}\frac{\partial \mathcal{L}}{\partial \dot{y}} = m \ddot{y}$$

$$\frac{\partial \mathcal{L}}{\partial y} = -mg$$

$$m\ddot{y} + mg = F$$

$$\boxed{m\ddot{y} = F - mg}$$

Example: pendulum.



$$\mathcal{L} = T - V \,, \quad \frac{d}{dt}\frac{\partial \mathcal{L}}{\partial \dot{q}} - \frac{\partial \mathcal{L}}{\partial q} = \tau$$

$x = l\sin\theta \qquad v_x = l\cos\theta\,\dot\theta \qquad \Longrightarrow \quad T = \frac{1}{2}m\left(v_x^2 + v_y^2\right)$

$y = l - l\cos\theta \qquad v_y = l\sin\theta\,\dot\theta \qquad\qquad = \frac{1}{2}m l^2 \dot\theta^2$

$\triangle$ $\frac{d}{dt}(x) = \frac{d}{dt}\left(l\sin\theta\right) = l\,\frac{d}{d\theta}\sin\theta\,\dot\theta = l\cos\theta\,\dot\theta$

$V = mg(l - l\cos\theta) \qquad \mathcal{L} = \frac{1}{2}m l^2\dot\theta^2 - mg(l - l\cos\theta)$

$\frac{\partial \mathcal{L}}{\partial \dot\theta} = m l^2 \dot\theta \,, \quad \frac{d}{dt}\frac{\partial \mathcal{L}}{\partial \dot\theta} = m l^2 \ddot\theta$

$\frac{\partial \mathcal{L}}{\partial \theta} = -mgl\sin\theta$

$$\boxed{\tau = m l^2 \ddot\theta + mgl\sin\theta}$$

$$\boxed{\tau = ml^2\ddot{\theta} + mgl\sin\theta}$$

How to simulate in matlab?

$$\ddot{\theta} = \left(\frac{1}{ml^2}\right)\tau - \frac{g}{l}\sin\theta$$

$$\dot{x} = f(x, u)$$

$$\underbrace{\frac{d}{dt}\begin{bmatrix}\theta \\ \dot{\theta}\end{bmatrix}}_{\substack{\text{state} \\ \text{or} \\ \text{generalized coordinates}}} = \underbrace{\begin{bmatrix}\dot{\theta} \\ -g/l\sin\theta + \frac{1}{ml^2}\overset{u}{\tau}\end{bmatrix}}_{\substack{\text{dynamics} \\ \text{of} \\ \text{Equation of Motion.}}}$$