23 JAN 2020 KINEMATICS [lec 4]

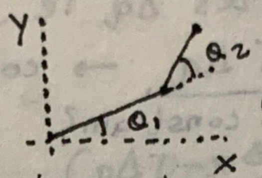
* Forward kinematics ⇒ joint angle → co-ordinates
(q) R, T
Homogeneous transformation orientation position

- Column vectors of transf matrix are its axes in new frame
- $\boxed{\text{URDF Files}}$ Translation and Roll-Pitch-Yaw

* Inverse kinematics - optimize
1] Jacobian transponse $E(q) = \frac{1}{2} \Delta_x^T \Delta x \Rightarrow \Delta q = -\alpha \left[\frac{\partial E(q_0)}{\partial q_0}\right]^T$
- For small Δx
- doesn't minimize movement of joints $= \alpha J^T \Delta x$

2] Pseudo inverse Jacoban
- reduces q
3] Damped least squard

* Collision checking



2 link arm
$Q = [2 \times 1] \Rightarrow$ Jacobian $= [2 \times 2]$

First column of Jacobian gives change due to Q_1 } and so on
second column due to Q_2 }

$\therefore$ Let $J = [J_1\ J_2] \Rightarrow V_e = J_1 \Delta Q_1 + J_2 \Delta Q_2$

* Forward kinematics → Takes in robot joint angles (q vector) and
$f(q)$ gives position(T) and orientation (R) of effector

Homogeneous Transformation
Change of origin

$$\begin{bmatrix} x_1^B \\ x_2^B \\ x_3^B \\ 1 \end{bmatrix} = \begin{bmatrix} R_{11} & R_{12} & R_{13} & T_1 \\ R_{21} & R_{22} & R_{23} & T_2 \\ R_{31} & R_{32} & R_{33} & T_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1^A \\ x_2^A \\ x_3^A \\ 1 \end{bmatrix}$$

Here $[R_{11}\ R_{21}\ R_{31}]$ is the new x axis
$[R_{12}\ R_{22}\ R_{32}]$ is Y axis
$[R_{13}\ R_{23}\ R_{33}]$ is x axis
$[T_1\ T_2\ T_3]$ translation for x, y, z

$$\Rightarrow x^B = \begin{bmatrix} R & t \\ 0 & 1 \end{bmatrix} x^A = \begin{pmatrix} R x^A + t \\ 1 \end{pmatrix}$$

$R \in \mathbb{R}^{3\times3}$ | R = Rotation matrix
$T \in \mathbb{R}^{3\times1}$ | T = Translation matrix

- Homogeneous Transformation can be chained by multiplication

* Jacobian → For non-linear transformation, transformation at a point can be approximated to linear.
- Jacobian matrix is locally linear transformation matrix at a given point.

Let transformation be $\begin{bmatrix} x_{new} \\ y_{new} \end{bmatrix} = \begin{bmatrix} f_1(x,y) \\ f_2(x,y) \end{bmatrix} = \begin{bmatrix} x + \sin y \\ y + \sin x \end{bmatrix}$

$\therefore$ Jacobian $= \begin{bmatrix} \partial f_1/\partial x & \partial f_1/\partial y \\ \partial f_2/\partial x & \partial f_2/\partial y \end{bmatrix}$ at some point

Position of effector $x = f(q)$

* Velocities $= \frac{\partial x}{\partial q} = \frac{\partial f(q)}{\partial q} = J(q)$

* Inverse Kinematics → orientation (R), Position (T) is known, Find Joint angles (J)

$q = f^{-1}(x)$ → multiple/no solutions

- Hence iterative optimization process instead of analytical

1] Jacobian Transpose $\quad x = f(q)$ $\quad x^*$ is new position

$$E(q) = \frac{1}{2}\Delta x^T \Delta x = \frac{1}{2}[x^* - f(q)]^T[x^* - f(q)] \rightarrow \text{cost}$$

$A = I$ How does this transpose come? As $\frac{\partial X^T A X}{\partial X} = -2AX$

$$\frac{\partial E(q)}{\partial q} = \frac{\partial E(q)}{\partial x}\cdot\frac{\partial x}{\partial q} \Rightarrow -\Delta x^T \cdot \frac{\partial f(q)}{\partial q}$$

why take transpose?

$$\text{Now } \Delta q = -\alpha\left[\frac{\partial E(q_0)}{\partial q_0}\right]^T = \alpha\left[(x^* - f(q))^T J(q_0)\right]^T = \alpha J(q_0)^T \Delta x$$

minimises Δx

2] Pseudo-Inverse → minimizes Δq ie joint movements

$\min \Delta q^T \Delta q$ for $\underbrace{\Delta x = J\Delta q}$ → cost

why this constrain? $\rightarrow J = \frac{\partial f(q)}{\partial q} = \frac{\partial x}{\partial q} \Rightarrow \boxed{\frac{\partial x}{\partial q} \times \Delta q \approx \Delta x}$ ensures small step

$$\Rightarrow E(q) = \frac{1}{2}\Delta q^T \Delta q + \lambda^T(\Delta x - J\Delta q)$$

$$\Rightarrow \frac{\partial E(q)}{\partial q} = \Delta q^T - \lambda^T J = 0 \quad \cancel{\Rightarrow \Delta q^T = \lambda^T J} \Rightarrow \Delta q = J^T\lambda$$

$$\Rightarrow J\Delta q = JJ^T\lambda \Rightarrow \lambda = (JJ^T)^{-1}\underbrace{J\Delta q}_{\Delta x} \Rightarrow \lambda = (JJ^T)^{-1}\Delta x$$

why multiply both sides by J → To sst with Δx

$$\text{But } \Delta q = J^T\lambda = \underbrace{J^T(JJ^T)^{-1}}_{J^\#}\Delta x$$

3] Damped Least Square → weights to minimise Δx and Δq combination of both

$$E(q) = \frac{1}{2}|f(q) - x^*|^2_C + \frac{1}{2}|q - q_0|^2_W$$

where $f(q) = f(q_0) + J(q - q_0) = x_0 + J(q - q_0)$

$$\Rightarrow E(q) = \frac{1}{2}[x_0 - x^* + J(q-q_0)]^T C [x_0 - x^* + J(q-q_0)] + \frac{1}{2}(q-q_0)^T W (q-q_0)$$

$$\frac{\partial E(q)}{\partial q} = [x_0 - x^* + J(q-q_0)]^T CJ + (q-q_0)^T W$$

$$\Rightarrow \Delta q = \underbrace{W^{-1}J^T(JW^{-1}J^T + C^{-1})^{-1}}_{J^\#}\Delta x$$

when $C = \infty$, $W = I$ → we get pseudo inverse case

With any method $q_{t+1} = q_t + J^\#[x^*_{t+1} - f(q_t)]$

---

* Collision Tracking

Separation Axis Theorem → For convex solids

- Check surface normal → 3+3 ⇒ 6
- cross product between edges → 3×3 ⇒ 9

} Check till we don't find seperation.