Feb. 17, 2020

Recap: Forward Kinematics

$$T_n^0(q) = \begin{bmatrix} R_n^0(q) & O_n^0(q) \\ 0 & 1 \end{bmatrix} = A_1^0(a_1,\alpha_1,d_1,\theta_1)A_2^1(a_2,\alpha_2,d_2,\theta_2)\cdots A_n^{n-1}(a_n,\alpha_n,d_n,\theta_n)$$

Pose of n^{th} frame relative to the 0^{th} frame.

R_n^0 : Orientation of n^{th} frame relative to the 0^{th} frame

O_n^0 : Position of the n^{th} frame relative to the 0^{th} frame.

Note: 0^{th} frame is the fixed frame at the base of the robot

A_i^{i-1} : Link Transformation

$a_i, \alpha_i, d_i, \theta_i$: DH-Parameters.

$$q_i = \begin{cases} d_i & \text{for prismatic joint} \\ \theta_i & \text{for revolute joint} \end{cases}$$

→ generalized coordinate "joint variable"

Note
e.e.: end effector

Note:
$\tilde{\square}$ denotes numeric (i.e. non-variable) quantities

Forward Kinematics: Given some robot structure and joint positions $\tilde{q}$, calculate the e.e. pose $T_n^0(\tilde{q})$.

Inverse Kinematics: Given some robot structure and e.e. pose $\tilde{T}_n^0$, find joint positions $\tilde{q}$ that satisfy $T_n^0(\tilde{q}) = \tilde{T}_n^0$

Today: Velocity Kinematics

We want a relationship of the form:

linear velocity

angular velocity

joint speeds

$$\underbrace{\begin{bmatrix} v \\ \omega \end{bmatrix}}_{6\times 1} = \underbrace{\begin{bmatrix} J \end{bmatrix}}_{6\times n} \underbrace{\begin{bmatrix} \dot{q} \end{bmatrix}}_{n\times 1}$$

Let's write this in partitioned form:

$$\underbrace{\begin{bmatrix} v \\ \omega \end{bmatrix}}_{6\times 1} = \underbrace{\begin{bmatrix} | & | & & | \\ J_1(q_1,\dots,q_n) & J_2(q_1,\dots,q_n) & \cdots & J_n(q_1,\dots,q_n) \\ | & | & & | \end{bmatrix}}_{6\times n} \underbrace{\begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \vdots \\ \dot{q}_n \end{bmatrix}}_{n\times 1}$$

Partition one more time:

$$\begin{bmatrix} v \\ \omega \end{bmatrix} = \begin{bmatrix} J_{v_1} & J_{v_2} & \cdots & J_{v_n} \\ J_{\omega_1} & J_{\omega_2} & & J_{\omega_n} \end{bmatrix} \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \vdots \\ \dot{q}_n \end{bmatrix}$$

Note: J_{v_i} & J_{ω_i} are still functions of q. We omit for compactness.

J_{v_i} says: What linear velocity will the e.e. experience if I wiggle the i^{th} joint by $\dot{q}_i$?

J_{ω_i} says: What angular velocity will the e.e. experience if I wiggle the i^{th} joint by $\dot{q}_i$?

Now the game is: how to calculate J_{v_i} & J_{ω_i}?

Linear velocity

By the chain rule:

$$v = \sum_{i=1}^{n} \frac{\partial o_n^0}{\partial q_i} \frac{\partial q_i}{\partial t}$$

$$\Rightarrow J_{v_i} = \frac{\partial o_n^0}{\partial q_i}$$

Case I: Prismatic joint.

Prismatic joint generates e.e. linear velocity along its joint axis so:

$$J_{v_i} = z_{i-1}^0$$

Case II. Revolute joint

The linear velocity generated by a pure rotation is given by:

$$v = \omega \times r$$
$$= (z_{i-1}^0 \dot{q}_i) \times (o_n^0 - o_{i-1}^0)$$
$$J_{v_i} = z_{i-1}^0 \times (o_n^0 - o_{i-1}^0)$$

Recall:

$$T_{i-1}^0 = A_1^0 A_2^1 \cdots A_{i-1}^{i-2} = \begin{bmatrix} | & | & | & | \\ x_{i-1}^0 & y_{i-1}^0 & z_{i-1}^0 & o_{i-1}^0 \\ | & | & | & | \\ - & 0 & - & 1 \end{bmatrix}$$

z_{i-1}^0 : axis of i^{th} joint in {0} frame coordinates

o_{i-1}^0 : center of i^{th} joint in {0} frame coordinates.

Angular velocity

Case I: Prismatic joints. No angular velocity.

So $J_{\omega_i} = 0$

Case II. Revolute joints.

Note: Sec 4.4 in Spong

$$J_{\omega_i} = z_{i-1}^0$$

$\Downarrow$

We can add angular velocities as long as they are all expressed in same ~~frame~~ coordinates

$$\omega_{0,n}^0 = z_0^0 \dot{q}_1 + z_1^0 \dot{q}_2 + \ldots + z_{n-1}^0 \dot{q}_n$$

Now:

$$J(q) = \begin{bmatrix} J_v \\ J_\omega \end{bmatrix} = \begin{bmatrix} J_{v_1} & J_{v_2} & \dots & J_{v_n} \\ J_{\omega_1} & J_{\omega_2} & & J_{\omega_n} \end{bmatrix}$$

and

$$\begin{bmatrix} v \\ \omega \end{bmatrix} = J(q)\,\dot{q}$$

(1) $J(q)$ is called the "Geometric Jacobian". A.k.a "standard" Jacobian, "hybrid"

v is linear velocity of e.e. relative to robot base, expressed in $\{0\}$ coordinates.

ω is angular " " " " " " " " " " .

(2) $J_b(q)$ is called the "Body Jacobian"

It yields:

v as the linear velocity of e.e relative to robot base, expressed in e.e. coordinates

ω as the angular " " " " " " " " " " "

$$J_b(q) = \begin{bmatrix} R_n^0(q)^T & 0 \\ 0 & R_n^0(q)^T \end{bmatrix} J(q)$$

(3) $J_a(q)$ is called the "Analytic Jacobian"

Use when E.E. ~~pose~~ orientation is represented by Euler angles

E.E. pose is then given by:

$$X(q) = \begin{bmatrix} o_n^0(q) \\ \phi(q) \\ \theta(q) \\ \psi(q) \end{bmatrix} \in \mathbb{R}^6 \qquad \text{Let } \alpha(q) = \begin{bmatrix} \phi(q) \\ \theta(q) \\ \psi(q) \end{bmatrix}$$

$$J_a(q) = \frac{\partial X(q)}{\partial q}$$ (Traditional Jacobian)

(Derivative of one vector w.r.t. other)

where

$$\begin{bmatrix} v \\ \dot{\alpha} \end{bmatrix} = J_a(q)\,\dot{q} \implies J_a(q) = \begin{bmatrix} I & 0 \\ 0 & B^{-1}(\alpha) \end{bmatrix} J(q)$$

You can show

Euler angles α for which B is non-invertible are called representational singularities.

(4) Screw-based Jacobian $J_s(q)$ "Spatial Jacobian" in MLS

ω is angular velocity of e.e. relative to the robot base, expressed in $\{0\}$ coordinates.

V is linear velocity of a point on a rigid body attached to the e.e. that is coincident with the base. (The point is instantaneously coincident to the base)

$$J_s(q) = \begin{bmatrix} R_n^0 & \hat{o}_n^0 R_n^0 \\ 0 & R_n^0 \end{bmatrix} J_b(q)$$

Note: R_n^0, o_n^0 are functions of q.

$\hat{\square}$ is the "hat" operator

$$\hat{\omega} = \begin{bmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{bmatrix}$$

$$\hat{a} b = a \times b$$

Note: Deriving $B(\alpha)$. Start with definition of spatial angular velocity:

$\hat{\omega} = \dot{R}R^{-1}$. Now use chain rule:

$$\hat{\omega} = \sum_{i=1}^{3} \left(\frac{\partial R}{\partial \alpha_i} \dot{\alpha}_i \right) R^{-1}$$

$\dot{\alpha}_i$ is a scalar so:

$$= \sum_{i=1}^{3} \left(\frac{\partial R}{\partial \alpha_i} R^{-1} \right) \dot{\alpha}_i$$

The result follows.

OR, judiciously say: (for a specific case: ZYZ angles)

$$\omega = \underbrace{\omega_z}_{\text{unit vector in } z} \dot{\psi} + R_{z,\psi} \underbrace{\omega_y}_{\text{unit vector in } y} \dot{\theta} + R_{z,\psi} R_{y,\theta} \underbrace{\omega_z}_{\text{unit vector in } z} \dot{\phi}$$

$$\omega_z = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \qquad \omega_y = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}$$

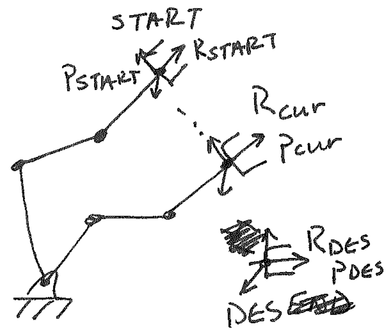
Similar to above, we are expressing ~~for example~~ all of the angular velocity terms in a single set of coordinates. Here, we are using $\{0\}$ coordinates.

s.t.

$$\omega = B(\alpha)\dot{\alpha} = \underbrace{\begin{bmatrix} | & | & | \\ R_{z,\psi} R_{y,\theta} \omega_z & R_{z,\psi} \omega_y & \omega_z \\ | & | & | \end{bmatrix}}_{B(\alpha)} \underbrace{\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix}}_{\dot{\alpha}}$$

Let's do something useful with the Jacobian:

Resolved Rates Control: or, sidestepping inverse kinematics.



Given some initial configuration $q_{START} \Rightarrow P_{START}, R_{START}$

Find the joint configuration $q_{\cancel{END}\,\cancel{DES}\,END}$ that yields robot ~~to~~ pose $R_{\cancel{DES}\,END}$, $P_{\cancel{DES}\,END}$ that lies within some radius of convergence of desired pose: R_{DES}, P_{DES}.

The idea: define position error as:

$$P_{error} = P_{DES} - P_{cur}$$

ε_p : radius of convergence for position

define rotation error as:

$$R_{DES} = R_{error} R_{cur}$$

$$\Rightarrow R_{error} = R_{DES} R_{cur}^{-1}$$

$\hookrightarrow \omega_{error}, \theta_{error}$ (axis angle rep)

ε_ω : radius of convergence for rotation

At each time step, choose linear and angular velocities to reduce those errors:

while $\|P_{error}\| > \varepsilon_p$ ~~&~~ OR $\|\cancel{\omega_{DES}}\ \theta_{error}\| > \varepsilon_\omega$

get R_{cur}, P_{cur} from $T_n^0(q_{cur})$. Calculate P_{error}, R_{error}.

choose $V_{DES} = \dfrac{P_{error}}{\|P_{error}\|} \cancel{speed}\ V_{speed}$ ~ linear speed.

$\omega_{DES} = \dfrac{\omega_{error}}{\|\omega_{error}\|} \dot{\theta}_{speed}$ ~ angular speed.

choose speed based on distance from goal state.

set:

$$q_{cur} = q_{PREVIOUS} + \dot{q}_{DES} \Delta t$$

where $\dot{q}_{des} = J^{-1} \begin{bmatrix} V_{des} \\ \omega_{des} \end{bmatrix}$

end

$\Downarrow$ Use "singularity-robust weighted pseudo inverse"

$$J^{+} = W^{-1} J^{T} \left(\alpha^2 I + J W^{-1} J^{T} \right)^{-1}$$

where α is some small number (5)
W is diag weighting matrix