2.2_PoseIn3D

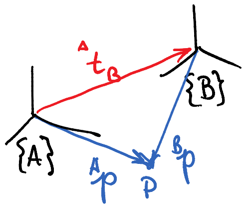
Monday, May 28, 2012
1:22 PM

- 2.2. Pose in 3D
  - Eq.(2.11)
  - 2.2.1. Representing orientation
    - Euler's thm - any orientation can be achieved by a sequence of 3 rotations different coordinate axes
    - Introduce SO(3). It has 3 dof
    - Rotations do not commute, i.e., operator (+) is not commutative.
    - Orientation representations: R matrix, Euler and Cardan angles, axis-angle, and unit quat
  - 2.2.2. Combining translation and rotation
    - Introduce SE(3)
    - 4x4 Htform

Pose of a 3-d rigid body is an element of $SE(3)$

$$SE(3) = SO(3) \times \mathbb{R}^3$$

$${}^A T_B = \begin{bmatrix} {}^A R_B & {}^A t_B \\ 0\,0\,0 & 1 \end{bmatrix}_{(4\times4)} \quad , \quad t \in \mathbb{R}^3 \ , \ {}^A R_B \in SO(3)$$



Note: R is (3×3).

$${}^A\tilde{p}_B = {}^A T_B \, {}^B\tilde{p} \ , \quad \text{where}$$

$$\tilde{p} = \begin{bmatrix} p \\ 1 \end{bmatrix}_{(4\times1)}$$

The product & inverse formulas for T remain unchanged.

$$\ominus {}^A\xi_B \sim \left({}^A T_B\right)^{-1} = \begin{bmatrix} {}^A R_B & {}^A t_B \\ 0\,0 & 1 \end{bmatrix}^{-1} = \begin{bmatrix} {}^A R_B^T & -{}^B R_A^T \, {}^A t_B \\ 0\,0 & 1 \end{bmatrix}$$

$${}^A\xi_B \oplus {}^B\xi_C \sim {}^A T_B \, {}^B T_C = {}^A T_C = \left[\begin{array}{c:c} {}^A R_B \, {}^B R_C & {}^A t_B + {}^A R_B \, {}^B t_C \\ \hdashline 0\,0 & 1 \end{array}\right]_{(4\times4)}$$

However:

① rotations do <u>NOT</u> commute

② mapping from R to parametric representations can have singularities.

If $A \in SO(3)$, then $A^{-1} = A^T$ and $\det(A) = 1$.

$$\therefore \quad a_1^T a_1 = 1 \qquad a_1^T a_2 = 0$$
$$a_2^T a_2 = 1 \qquad a_2^T a_3 = 0$$

$$a_3^T a_3 = 1 \qquad a_3^T a_1 = 0$$

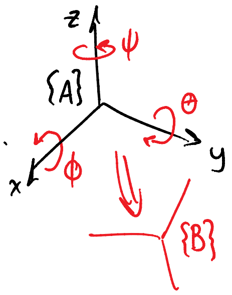
A has 9 elements and 6 constraints

$\therefore$ A has 3 degrees of freedom

Euler's theorem:

$R \in SO(3)$ can be constructed from a sequence of three rotations about coordinate axes.

eg. $R_z(\psi) \cdot R_y(\theta) \cdot R_x(\phi)$

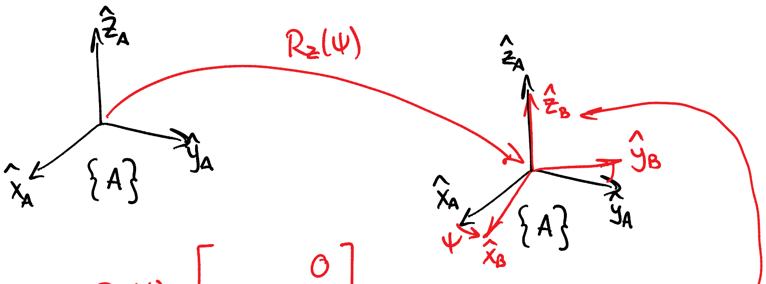


Note: a rotation about a given axis is a planar rotation.

$$R_x(\theta) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & c_\theta & -s_\theta \\ 0 & s_\theta & c_\theta \end{bmatrix} \quad \text{where} \quad \begin{aligned} s_\theta &= \sin(\theta) \\ c_\theta &= \cos(\theta) \end{aligned}$$

$$R_y(\theta) = \begin{bmatrix} c_\theta & 0 & s_\theta \\ 0 & 1 & 0 \\ -s_\theta & 0 & c_\theta \end{bmatrix} \qquad R_z(\theta) = \begin{bmatrix} c_\theta & -s_\theta & 0 \\ s_\theta & c_\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

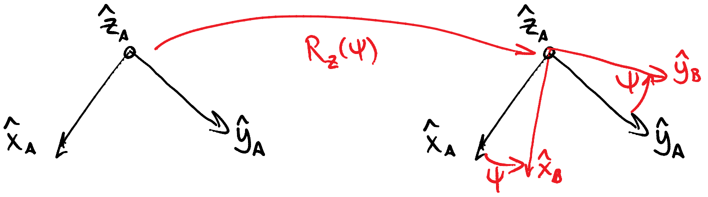
Graphical interpretation of rotation matrix: ${}^A R_B = \left[{}^A\hat{x}_B \;\; {}^A\hat{y}_B \;\; {}^A\hat{z}_B \right]$



$$R_z(\psi) = \begin{bmatrix} & & 0 \\ & & 0 \\ 0 & 0 & 1 \end{bmatrix} \underbrace{}_{{}^A\hat{z}_B}$$

$${}^A\hat{z}_B = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \Rightarrow {}^A\hat{z}_B = {}^A\hat{z}_A$$

$$ {}^A\hat{x}_B = \begin{bmatrix} \cos(\psi) \\ \sin(\psi) \\ 0 \end{bmatrix} \qquad {}^A\hat{y}_B = \begin{bmatrix} -\sin(\psi) \\ \cos(\psi) \\ 0 \end{bmatrix} $$



Euler's theorem says that given $R = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix}$, a solution

to the following equation exists:

$$ R = R_z(\psi) R_y(\theta) R_x(\phi) $$

However, soln is not necessarily unique!

2.2.1.2. Three-Angle Representations

Euler angles: Rotate about one axis, then another, then the first. XYX, ZXZ, ...

Cardan angles: Use all three axes: XYZ, ...

The toolbox uses ZYZ Euler angles:

$$ \Rightarrow \quad R = \text{rotz}(0.1) + \text{roty}(0.2) * \text{rotz}(0.3) $$

$$ = \text{eul2r}(0.1, 0.2, 0.3) $$

$$ \Gamma = \text{tr2eul}(R) = (0.1, 0.2, 0.3) $$

Solution is not unique!

The toolbox always returns the solution with $\theta > 0$.

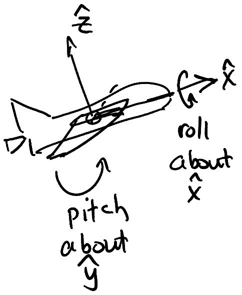
$$ \text{tr2eul}(\text{eul2r}(0.1, -0.2, 0.3)) = (-\pi + 0.1, 0.2, -\pi + 0.3) $$

The toolbox also uses roll-pitch-yaw angles

$$R = \mathrm{rotx}(\theta_r) * \mathrm{roty}(\theta_p) * \mathrm{rotz}(\theta_y)$$

$$= \mathrm{rpy2r}(\theta_r, \theta_p, \theta_y)$$

$$T = \mathrm{tr2rpy}(R)$$



Again 2 solutions exist in generic cases.

Inverse kinematics solution for Z-Y-Z Euler angles

$$R = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} = R_z(\phi)\, R_y(\theta)\, R_z(\psi)$$

$\Rightarrow$ rotate about original z by ϕ
then " " new y by θ
" " " new z by ψ

Expand the product of rotation matrices

$$R = \left[\begin{array}{c:c:c} C_\phi C_\theta C_\psi - S_\phi S_\psi & -C_\phi C_\theta S_\psi - S_\phi C_\psi & C_\phi S_\theta \\ S_\phi C_\theta C_\psi + C_\phi S_\psi & -S_\phi C_\theta S_\psi + C_\phi C_\psi & S_\phi S_\theta \\ -S_\theta C_\psi & S_\theta S_\psi & C_\theta \end{array}\right]$$

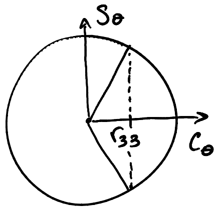
Generic solution:

$\theta = \cos^{-1}(r_{33}) \Longrightarrow$ two solutions, θ_1 and $\theta_2 = -\theta_1$

$$\Rightarrow \boxed{\begin{aligned} \theta_1 &= \cos^{-1}(r_{33}) \\ \theta_2 &= -\theta_1 \end{aligned}}$$

Note that: $\sin\theta_1 = +\sqrt{1 - r_{33}^2}$

$\sin\theta_2 = -\sqrt{1 - r_{33}^2}$



$$\left.\begin{aligned} r_{13} &= C_\phi S_{\theta_i} \\ r_{23} &= S_\phi S_{\theta_i} \end{aligned}\right\} \Longrightarrow \left.\begin{aligned} C_\phi &= r_{13}/S_{\theta_i} \\ S_\phi &= r_{23}/S_{\theta_i} \end{aligned}\right\} \Longrightarrow \underset{i=\{1,2\}}{\phi = \mathrm{atan2}(r_{23}/S_{\theta_i}, r_{13}/S_{\theta_i})}$$

$$\Rightarrow \boxed{\begin{aligned} \phi_1 &= \mathrm{atan2}(r_{23}, r_{13}) \\ \phi_2 &= \mathrm{atan2}(-r_{23}, -r_{13}) = \phi_1 + \pi \end{aligned}}$$

$r_{31} = -s_\theta c_\psi$

$r_{32} = s_\theta s_\psi$

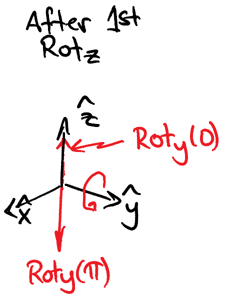
$$\Rightarrow \boxed{\begin{aligned} \psi_1 &= \text{atan2}(r_{32}, -r_{31}) \\ \psi_2 &= \text{atan2}(-r_{32}, r_{31}) = \psi_1 + \pi \end{aligned}}$$

2.2.1.3. Singular Configurations & Gimbal Lock

All three-parameter representations have singular configurations.

Example ZYZ Euler Angles

if rotation about $\hat{y} = 0$ or π, then $\hat{z}$ after y-rotation is about $\hat{z}$ or $-\hat{z}$ after y-rotation.



In singular cases, rotz(φ) * roty(θ) * rotz(ψ) reduces:

$$\theta = 0 \Rightarrow \text{rotz}(\phi) * I_{(3\times3)} * \text{rotz}(\psi) = \text{rotz}(\phi+\psi)$$

$$\theta = \pi \Rightarrow \text{rotz}(\phi) * \begin{bmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -1 \end{bmatrix} * \text{rotz}(\psi) = \text{rotz}(\phi-\psi)$$

In singular cases:

$$R = \begin{bmatrix} \pm c_\phi c_\psi - s_\phi s_\psi & \mp c_\phi s_\psi - s_\phi c_\psi & 0 \\ \pm s_\phi c_\psi + c_\phi s_\psi & \mp s_\phi s_\psi + c_\phi c_\psi & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Apply angle sum and difference formulas:

$$\left.\begin{aligned} \sin(\phi \pm \psi) &= s_\phi c_\psi \pm c_\phi s_\psi \\ \cos(\phi \pm \psi) &= c_\phi c_\psi \mp s_\phi s_\psi \end{aligned}\right\} \Rightarrow$$

$$R = \begin{bmatrix} \cos(\phi \pm \psi) & -\sin(\phi \pm \psi) & 0 \\ \sin(\phi \pm \psi) & \cos(\phi \pm \psi) & 0 \\ 0 & 0 & \pm 1 \end{bmatrix}$$

$\Rightarrow$ if $r_{33} \simeq 1$

$$\theta = 0$$
$$\phi + \psi = \mathrm{atan2}(r_{21}, r_{11})$$

else $r_{33} \simeq -1$

$$\theta = \pi$$
$$\phi - \psi = \mathrm{atan2}(r_{21}, r_{11})$$

There is an infinite number of solutions:

θ is unique and $\phi+\psi$ (or $\phi-\psi$) is unique,

but ϕ & ψ are not unique.

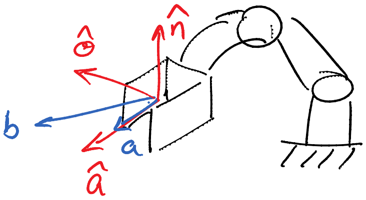
In singular configs, tr2rpy sets $\phi = 0$ by default.

Singular configurations lose degrees of freedom. (see Fig 2.13)

When two axes of a gimbal align, rotation about an axis perpendicular to the remaining two is impossible!

2.2.1.4 Two Vector Representations

Perhaps there are features on the end effector to define orientation.



Any two non-parallel vectors can define an orientation:

Given a & o:

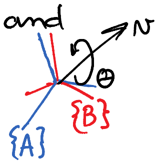
$$\left.\begin{array}{l} a = \text{approach} \\ b = \text{another} \end{array}\right\} \Rightarrow \left.\begin{array}{c} \hat{a} = \dfrac{a}{\|a\|} \\ \dfrac{b \times \hat{a}}{\|b \times \hat{a}\|} = \hat{n} \end{array}\right\} \Rightarrow \hat{a} \times \hat{n} = \hat{o}$$

toolbox function: ao2tr(b, a) $\Rightarrow$ $R = \begin{bmatrix} \hat{n} & \hat{o} & \hat{a} \end{bmatrix}$

~~tr2ao(~~) ← not implemented. not needed

2.2.1.5. Rotation about an arbitrary vector

Any rotation R can be represented by an axis and an angle of rotation about that axis, (θ, ν)



Rodrigues' Formula

$$^{A}R_{B} = I_{(3\times3)} + \sin(\theta)\, S(\nu) + (1-\cos(\theta))\left(\overset{(3\times3)}{\nu\nu^{T}} - I_{(3\times3)}\right)$$

where $S(\nu) = \begin{bmatrix} 0 & -\nu_z & \nu_y \\ \nu_z & 0 & -\nu_x \\ -\nu_y & \nu_x & 0 \end{bmatrix}$ and $\|\nu\| = 1$.

The # of parameters is 4: $\theta, \nu_x, \nu_y, \nu_z$,

but $\|\nu\| = 1$, is a constraint, so still R has only three degrees of freedom!

Toolbox function: angvec2tr(θ, [ν_1 ν_2 ν_3]) $\Rightarrow$ T or R
tr2angvec((T or R)) $\Rightarrow$ θ, ν

2.2.1.6. Unit Quaternions, a.k.a. Euler Parameters.

A quaternion, $\mathring{q}$, is 4 elements interpreted as a scalar and a hyper-complex number

$$\begin{aligned} \mathring{q} &= s + \nu_1 i + \nu_2 j + \nu_3 k \\ &= s + \nu \\ &= s\langle \nu_1, \nu_2, \nu_3 \rangle \end{aligned}$$

where $i^2 = j^2 = k^2 = -1$
$ijk = -1$

We use unit quaternions to represent rotations:
i.e. $s^2 + \|\nu\| = 1$

Again we have 4 parameters and 1 constraint,
$\therefore$ unit quaternions provide 3 degrees of freedom.

Angle vector interpretation:

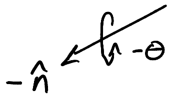
Angle vector interpretation:

$$s = \frac{\cos(\theta)}{2} \quad v = \frac{\sin(\theta)}{2}\hat{n}$$

where rotation is relative to $\hat{n}$

Note:

$\mathring{q} = -\mathring{q}$



Quaternion composition is efficient

$$\mathring{q} \oplus \mathring{q}' = \begin{bmatrix} s & v_1 & v_2 & v_3 \\ -v_1 & s & -v_3 & v_2 \\ -v_2 & v_3 & s & -v_1 \\ -v_3 & -v_2 & v_1 & s \end{bmatrix} \begin{bmatrix} s' \\ v_1' \\ v_2' \\ v_3' \end{bmatrix} \Rightarrow \begin{cases} 16 \text{ multiplications} \\ 12 \text{ additions} \end{cases}$$

However $RR' \Rightarrow \begin{cases} 27 \text{ mult.} \\ 18 \text{ add.} \end{cases}$

How was the above matrix-vector multiplication rule derived? Use the usual rules for multinomial multiplication and apply the identities relating i, j, & k

$$(s + \underbrace{v_1 i + v_2 j + v_3 k}_{v})(\underbrace{t}_{s'} + \underbrace{u_1 i + u_2 j + u_3 k}_{v'})$$

renamed these in the mat-vec operation above.

$st + su + tv + \ldots$

$\ldots - v_1u_1 - v_2u_2 - v_3u_3 + v_1u_2k + v_2u_3i + v_3u_1j + \ldots$

$\ldots - v_2u_1k - v_3u_2i - v_1u_3j$

Collect the scalar, i, j, and k components to construct the matrix and vector above.

The best thing about unit quaternions is that there are <u>no</u> singular configurations!

Let $\mathring{q} = a\langle b,c,d\rangle$ (temporarily replace $s\langle v_1, v_2, v_3\rangle$)

Mapping $\mathring{q}$ to R

$$R(\mathring{q}) = \begin{bmatrix} 2(a^2+b^2)-1 & 2(bc-ad) & 2(bd+ac) \\ 2(bc+ad) & 2(a^2+c^2)-1 & 2(cd-ab) \\ 2(bd-ac) & 2(cd+ab) & 2(a^2+d^2)-1 \end{bmatrix}$$

Mapping $R \rightarrow \mathring{q}$

Let $R = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix}$ and

replacing 1 in $R(\mathring{q})$ with $a^2+b^2+c^2+d^2$ yields:

$$a^2 = \frac{1}{4}(1 + r_{11} + r_{22} + r_{33}) \qquad (1)$$

$$b^2 = \frac{1}{4}(1 + r_{11} - r_{22} - r_{33}) \qquad (2)$$

$$c^2 = \frac{1}{4}(1 - r_{11} + r_{22} - r_{33}) \qquad (3)$$

$$d^2 = \frac{1}{4}(1 - r_{11} - r_{22} + r_{33}) \qquad (4)$$

These equations imply 16 solutions, but only 2 exist.

Off-diagonal terms give:

(5) $ab = \frac{1}{4}(r_{32} - r_{23})$ $\qquad$ $bc = \frac{1}{4}(r_{12} + r_{21})$ (8)

(6) $ac = \frac{1}{4}(r_{13} - r_{31})$ $\qquad$ $bd = \frac{1}{4}(r_{13} + r_{31})$ (9)

(7) $ad = \frac{1}{4}(r_{21} - r_{12})$ $\qquad$ $cd = \frac{1}{4}(r_{23} + r_{32})$ (10)

Soln approach:

① Use equation (1), (2), (3), or (4) with largest value to get 2 values of a, b, c, or d.

② Use 3 eqs from (5-10) to solve for other 3 values

③ Result is 2 solutions satisfying $\mathring{q}_1 = -\mathring{q}_2$

Alg.

```
If trace(R) > 0
    solve eq (1) for    $a = \frac{1}{2}\sqrt{tr(R)+1}$
    solve for b,c,d using equations (5-7)
        eq(5) yields $\Rightarrow$  $b = \frac{r_{32} - r_{23}}{4a}$
        eq(6)    $\Rightarrow$   similar
        eq(7)    $\Rightarrow$   similar

else if   $r_{11} = \max(r_{11}, r_{22}, r_{33})$
    solve for b using eq (2)
    solve for a, c, d using eqs (5,8,9)

else if   $r_{22} = \max(r_{11}, r_{22}, r_{33})$
    solve for c using eq (3)
    solve for a, b, d using eqs (6,8,10)
else
    solve for d using eq (4)
    solve for a, b, c using eqs (7,9,10)
end

finally  $\mathring{q}_1 = (a,b,c,d)$ & $\mathring{q}_2 = -\mathring{q}_1$
    are the two solutions.
```

Toolbox functions for quaternions

q = Quaternion(R)

q.R	q. norm
q.plot()	q.inv()

To see all the methods, type "methods(Quaternion)"

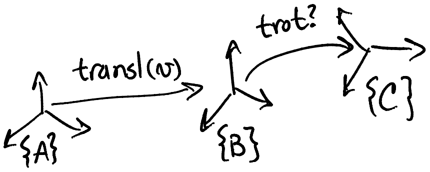
Toolbox functions for SE(3)

transl (x,y,z)

trotx (θ) , troty(θ) , trotz(θ)

trplot(T)

t2r(T) = R



example : $\,^{A}T_{c}$ = transl(v) * trotz(θ)

See table 2.1 & figure 2.15 for a complete list.