02.11.22 Minimal representations of orientation - Euler & RPY angles

Homogeneous transformations

Euler, RPY

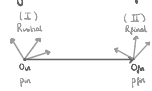
From 3x3 rot. matrices we move to a 4-D representation, but we introduce a number of minimal rep of orientation. This minimal number is 3 for a generic orientation in space. (3 angles → Euler / RPY)

"Minimal" representations

- rotation matrices in SO(3): 9 elements
  - 3 orthogonality relationships
  - 3 unitary relationships (unitary vectors) } orthonormality (but rot matrices also have det = +1)
  - = 3 independent variables → I can define orientation w/ just 3 parameters, for instance 3 angles.

direct problem ↑ ↓ inverse problem

Why am I not satisfied w/ rotation matrices?



1st reason:
I go from (I) to (II). As for the position, a path $p(s) = p_{in} + \Delta s$ where $\Delta = p_{fin} - p_{in}$, $s \in [0,1]$ does the job.
Can I say the same for the orientation?
$R(s) = R_{in} + (R_{fin} - R_{in})s$?
It's okay for initial & final state but it's WRONG cause for a generic s, R(s) is not a rot. matrix! (except for special cases like a rot. around a main axis)
Rotation matrices as such aren't okay to continuously describe a change of orientation.

2nd reason: If I'm doing this motion and have an error during the change of orientation how do I define the orientation error? I can't treat it like a position error, for the same reason as before (generic 3x3 matrix, not rot matrix).
→ I need a more compact way. error on the 3 angles.
⚠ even if I will often arrange those angles as an array, they are not a vector! They are a SEQUENCE of angles

- sequence of 3 rotations w.r.t. independent axes
  - by angles $\alpha_i, i = 1,2,3$, around fixed (a_i) or moving/current (a_i') axes
    - generically called Roll-Pitch-Yaw (fixed axes) or Euler (moving axes) angles
  - 12 + 12 possible different sequences (e.g., XYX)
    - without contiguous repetitions of axes (e.g., no XXZ nor YZ'Z')
  - actually, only 12 sequences are different since we shall see that

$$\{(a_1,\alpha_1),(a_2,\alpha_2),(a_3,\alpha_3)\} \equiv \{(a_3',\alpha_3),(a_2',\alpha_2),(a_1',\alpha_1)\}$$

Redundancy: a robot needs 6 dof in order to position & orient the e-e as desired within the workspace. (3 dof x positioning & 3 for orientation).
→ 7 dof: I may have redundancy. (∞^1 sol.)

→ I define a sequence of 3 rot. around indip. axes: the coordinate axes of the rotation frames.

→ either fixed (RPY) or moving (Euler) axes. → 1st rot. around Z moves X & Y axes to X' and Y' → 2nd rot. around one of those moved / current axes.

12 sequences:

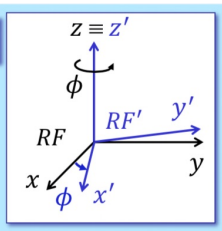
xyz''	xzy''	xyx''	xzx''
yxz''	yzx''	yxy''	yzy''
zxy''	zyx''	zxz''	zyz''

- rot. around fixed axes = rot. around moving axes but in the reversed order.

In a limited domain I want to pick the sequence that doesn't have singularities.

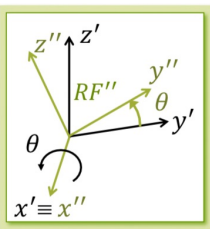
ZX'Z'' Euler angles

1



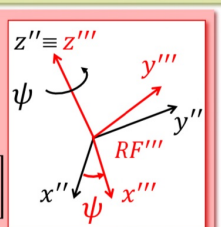
$$R_z(\phi) = \begin{bmatrix} \cos\phi & -\sin\phi & 0 \\ \sin\phi & \cos\phi & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

2



$$R_{x'}(\theta) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta \\ 0 & \sin\theta & \cos\theta \end{bmatrix}$$

3



$$R_{z''}(\psi) = \begin{bmatrix} \cos\psi & -\sin\psi & 0 \\ \sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

<elementary rotations!> ♡

- **direct problem:** given ϕ, θ, ψ, find R

$$R_{ZX'Z''}(\phi,\theta,\psi) = R_Z(\phi)R_{X'}(\theta)R_{Z''}(\psi) \quad \text{CONCATENATION}$$

order of definition in concatenation

$$= \begin{bmatrix} c\phi c\psi - s\phi c\theta s\psi & -c\phi s\psi - s\phi c\theta c\psi & s\phi s\theta \\ s\phi c\psi + c\phi c\theta s\psi & -s\phi s\psi + c\phi c\theta c\psi & -c\phi s\theta \\ s\theta s\psi & s\theta c\psi & c\theta \end{bmatrix}$$

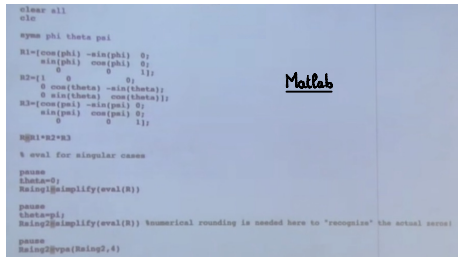
- given a vector $v''' = (x''', y''', z''')$ expressed in RF''', its expression in the coordinates of RF is

$$v = R_{ZX'Z''}(\phi,\theta,\psi)v'''$$

- the orientation of RF''' is the same that would be obtained with the sequence of rotations

ψ around z, θ around x (fixed), ϕ around z (fixed)

NB: if I write the final matrix by hand I can check for errors by squaring & summing the comps. of the columns ($\to 1$); scalar products between columns $\to 0$; det: $+1$.



```
clear all
clc

syms phi theta psi

R1=[cos(phi) -sin(phi)  0;
    sin(phi)  cos(phi)  0;
          0         0     1];
R2=[1          0          0;
    0  cos(theta) -sin(theta);
    0  sin(theta)  cos(theta)];
R3=[cos(psi) -sin(psi) 0;
    sin(psi)  cos(psi) 0;
          0         0    1];

R@R1*R2*R3

% eval for singular cases

pause
theta=0;
Reing1@simplify(eval(R))

pause
theta=pi;
Reing2@simplify(eval(R)) %numerical rounding is needed here to "recognize" the actual zeros!

pause
Reing2@vpa(Reing2,4)
```

Matlab

- **inverse problem:** given $R = \{r_{ij}\}$, find ϕ, θ, ψ

$$\begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} = \begin{bmatrix} c\phi c\psi - s\phi c\theta s\psi & -c\phi s\psi - s\phi c\theta c\psi & s\phi s\theta \\ s\phi c\psi + c\phi c\theta s\psi & -s\phi s\psi + c\phi c\theta c\psi & -c\phi s\theta \\ s\theta s\psi & s\theta c\psi & c\theta \end{bmatrix}$$

$\leftrightarrow$ 9 non linear eq in 3 unknowns, but I know I have 2 rot. matrices $\to$ I try and single out the 3 angles in an easy way.

- $r_{13}^2 + r_{23}^2 = s^2\theta$, $r_{33} = c\theta$ $\Rightarrow$ $\theta = \text{atan2}\left\{\pm\sqrt{r_{13}^2 + r_{23}^2},\ r_{33}\right\}$

  two values differing just for the sign

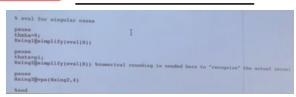
- if $r_{13}^2 + r_{23}^2 \neq 0$ (i.e., $s\theta \neq 0$)

  $r_{31}/s\theta = s\psi$, $r_{32}/s\theta = c\psi$ $\Rightarrow$ $\psi = \text{atan2}\{r_{31}/s\theta, r_{32}/s\theta\}$

- similarly... $\phi = \text{atan2}\{r_{13}/s\theta, -r_{23}/s\theta\}$

- there is always a pair of solutions in the regular case
- there are always singularities (here $\theta = 0$ or $\pm\pi$) $\Rightarrow$ only the sum $\phi + \psi$ or the difference $\phi - \psi$ can be determined

SINGULARITIES: $\theta = 0 \to$ two rot. around z



```
% eval for singular cases

pause
theta=0;
Reing1@simplify(eval(R))

pause
theta=pi;
Reing2@simplify(eval(R)) %numerical rounding is needed here to "recognize" the actual zeros!

pause
Reing2@vpa(Reing2,4)

Reing1 =
[ cos(phi + psi), -sin(phi + psi), 0]
[ sin(phi + psi),  cos(phi + psi), 0]
[              0,               0, 1]
```

$R = \begin{bmatrix} a & b & \cdot \\ c & d & \cdot \\ \cdot & \cdot & \cdot \end{bmatrix}$ the rot. is associated to a singularity because $s\theta = 0$ $\to \phi + \psi$

$\leftrightarrow$ rot of $\phi + \psi$ around z $\to \infty$ nb of col. 'cause I can only characterize the sum.

$\theta = \pi$

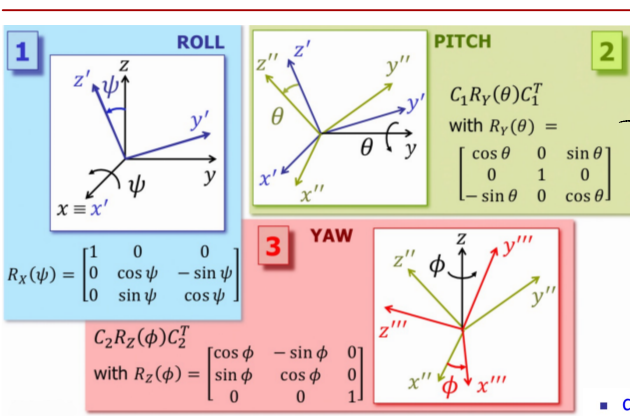
```
Reing2 =
[ cos(2*pi - phi),  -sin(2*pi - phi),  0, 1.220e-16*sin(phi)]
[ sin(2*pi - phi),  cos(2*pi - phi),  0, -1.220e-16*cos(phi)]
[ 1.220e-16*sin(phi),  1.220e-16*cos(phi),  -1, 0]
```

zero to $\emptyset$

$\leftrightarrow \begin{bmatrix} \cos(\psi - \phi) & -\sin(\psi - \phi) & 0 \\ -\sin(\psi - \phi) & -\cos(\psi - \phi) & 0 \\ 0 & 0 & -1 \end{bmatrix}$ I can only solve for the difference.

basically $\emptyset \to 0$ (not exactly because π is approximated)

w/ θ I have flipped over the z-axis.



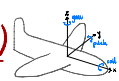
$$R_X(\psi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\psi & -\sin\psi \\ 0 & \sin\psi & \cos\psi \end{bmatrix}$$

$$C_2 R_Z(\phi) C_2^T \quad \text{with } R_Z(\phi) = \begin{bmatrix} \cos\phi & -\sin\phi & 0 \\ \sin\phi & \cos\phi & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$C_1 R_Y(\theta) C_1^T \quad \text{with } R_Y(\theta) = \begin{bmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{bmatrix}$$

singularity in $\cos\theta = 0$

Roll-Pitch-Yaw angles (fixed XYZ)



$\to$ rot. around the original Y-axis $\to$ all the (') angles will change orientation and go on (''). y is a generic r vector in the frame of y' $\to C_1 R_Y(\theta) C_1^T$

- **direct problem:** given ψ, θ, ϕ, find R

$$R_{RPY}(\psi,\theta,\phi) = R_Z(\phi)R_Y(\theta)R_X(\psi) \quad \Leftarrow \text{note the order of products!}$$

order of definition

$$= \begin{bmatrix} c\phi c\theta & c\phi s\theta s\psi - s\phi c\psi & c\phi s\theta c\psi + s\phi s\psi \\ s\phi c\theta & s\phi s\theta s\psi + c\phi c\psi & s\phi s\theta c\psi - c\phi s\psi \\ -s\theta & c\theta s\psi & c\theta c\psi \end{bmatrix}$$

- **inverse problem:** given $R = \{r_{ij}\}$, find ψ, θ, ϕ
- $r_{32}^2 + r_{33}^2 = c^2\theta$, $r_{31} = -s\theta$ $\Rightarrow$ $\theta = \text{atan2}\left\{-r_{31}, \pm\sqrt{r_{32}^2 + r_{33}^2}\right\}$

  for $r_{31} < 0$, two symmetric values w.r.t. $\pi/2$

- if $r_{32}^2 + r_{33}^2 \neq 0$ (i.e., $c\theta \neq 0$)

  $r_{32}/c\theta = s\psi$, $r_{33}/c\theta = c\psi$ $\Rightarrow$ $\psi = \text{atan2}\{r_{32}/c\theta, r_{33}/c\theta\}$

- similarly ... $\phi = \text{atan2}\{r_{21}/c\theta, r_{11}/c\theta\}$
- singularities for $\theta = \pm\pi/2 \Rightarrow$ only $\phi + \psi$ or $\phi - \psi$ are defined

...why this order in the product?

$$R_{RPY}(\psi,\theta,\phi) = R_Z(\phi)R_Y(\theta)R_X(\psi)$$

order of definition → "reverse" order in the product (pre-multiplication...)

X-rot

Y-rot : I reverse the 1st op., do the rot., & re-apply the 1st rot.

Z-rot : I reverse the 2nd rot, reverse the 1st, do the third, & reapply 2nd & 1st.

- need to refer each rotation in the sequence to one of the original fixed axes
  - use the angle/axis technique for each rotation in the sequence: $C\,R(\alpha)\,C^T$, with C being the rotation matrix reverting the previously made rotations (= "go back" to the original axes)

concatenating three rotations: [ ] [ ] [ ] (post-multiplication...)

$$\begin{aligned} R_{RPY}(\psi,\theta,\phi) &= [R_X(\psi)]\,[R_X^T(\psi)\,R_Y(\theta)\,R_X(\psi)] \\ &\quad [R_X^T(\psi)R_Y^T(\theta)\,R_Z(\phi)\,R_Y(\theta)R_X(\psi)] \\ &= R_Z(\phi)R_Y(\theta)R_X(\psi) \end{aligned}$$

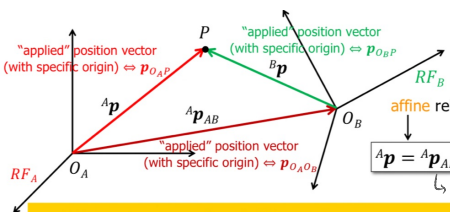
$R_x\,R_{y'}\quad R_z$:

$R_x\,R_{y'}\left[R_{y'}^T\,R_x^T\,R_z\,R_x\,R_{y'}\right]$

ORDER IS IMPORTANT!

Homogeneous

Homogeneous transformations → position & orientation together



affine relationship

$${}^A\boldsymbol{p} = {}^A\boldsymbol{p}_{AB} + {}^A R_B\,{}^B\boldsymbol{p}$$

↳ ~offset

$${}^A\boldsymbol{p}_{\text{hom}} = \begin{bmatrix} {}^A\boldsymbol{p} \\ \hline 1 \end{bmatrix} = \left[\begin{array}{c|c} {}^A R_B & {}^A\boldsymbol{p}_{AB} \\ \hline \boldsymbol{0}^T & 1 \end{array}\right] \begin{bmatrix} {}^B\boldsymbol{p} \\ \hline 1 \end{bmatrix} = {}^A T_B\,{}^B\boldsymbol{p}_{\text{hom}}$$

scaling factor

vector in homogeneous coordinates

4 × 4 matrix of homogeneous transformation

← linear relationship

reference frame RF_A. I want to define pose of RF_B.

If I know the expression of the position of the point P in frame B : ${}^B p$

I translate it in frame A.

I'm summing vectors expressed in the same frame.

" affine operation" → displacement : I can't really concatenate like in a matrix product.

→ not convenient.

⇒ HOMOGENEOUS TRANSF. MATRICES to get change of position & orientation in the same matrix.

first 3 rows multiplied by column : ${}^A R_B\,{}^B p + {}^A p_{AB}\cdot 1$: same formula!

NB: "applied" vector 'cause it's important where it starts from.

If I had a 3rd frame I could concatenate : ${}^A T_B\,{}^B T_C = {}^A T_C = \left[\begin{array}{c|c} {}^A R_B & {}^A p_{AB} \\ \hline 0^T & 1 \end{array}\right]\left[\begin{array}{c|c} {}^B R_C & {}^B p_{BC} \\ \hline 0^T & 1 \end{array}\right]$

blockwise $= \left[\begin{array}{c|c} {}^A R_B\,{}^B R_C + p_{AB}\cdot 0 & {}^A R_B\,{}^B p_{BC} + {}^A p_{AB} \\ \hline 0 & 1 \\ 0^T\,{}^B R_C + 1\cdot 0^T & 0^T\,{}^B p_{BC} + 1\cdot 1 \end{array}\right]$

$= \left[\begin{array}{c|c} {}^A R_C & {}^A p_{AC} \\ \hline 0 & 1 \end{array}\right] = {}^A T_C$

Properties:

- describes the relation between two reference frames (relative pose = position & orientation)
- transforms the representation of a position vector (applied vector starting from the origin of the frame) from one frame to another frame (changing coords)
- it is a roto-translation operator on vectors in the three-dimensional space
- it is always invertible $({}^A T_B)^{-1} = {}^B T_A$ inverse is not the transpose!!
- can be composed, i.e., ${}^A T_B\,{}^B T_C = {}^A T_C$ ← note: it does not commute in general! $\neq {}^B T_C\,{}^A T_B$

Inverse of a homogeneous transformation

exchange $A \rightleftarrows B$ ... with the original vectors/matrices ...

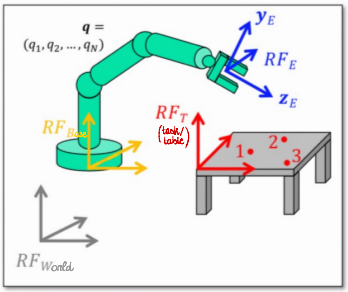
$${}^A\boldsymbol{p} = {}^A\boldsymbol{p}_{AB} + {}^A R_B\,{}^B\boldsymbol{p} \qquad {}^B\boldsymbol{p} = {}^B\boldsymbol{p}_{BA} + {}^B R_A\,{}^A\boldsymbol{p} = -\,{}^A R_B^T\,{}^A\boldsymbol{p}_{AB} + {}^A R_B^T\,{}^A\boldsymbol{p}$$

A→B instead of B→A

explicit form for the inverse

$$\underbrace{\left[\begin{array}{c|c} {}^A R_B & {}^A\boldsymbol{p}_{AB} \\ \hline 0\;0\;0 & 1 \end{array}\right]}_{{}^A T_B} \qquad \underbrace{\left[\begin{array}{c|c} {}^B R_A & {}^B\boldsymbol{p}_{BA} \\ \hline 0\;0\;0 & 1 \end{array}\right]}_{{}^B T_A} = \underbrace{\left[\begin{array}{c|c} {}^A R_B^T & -\,{}^A R_B^T\,{}^A\boldsymbol{p}_{AB} \\ \hline 0\;0\;0 & 1 \end{array}\right]}_{({}^A T_B)^{-1}}$$

Defining a robotic task



absolute definition of task

task definition relative to the robot end-effector

$$\underbrace{{}^{W}T_T}_{\text{Known}} = \underbrace{{}^{W}T_B}_{\text{known}}\ \underbrace{{}^{B}T_E}_{\text{depends on robot config}}\ \underbrace{{}^{E}T_T}_{\text{related to the specific task (given by task)}}$$

from W → T passing through the manipulator

known, once the robot is placed

direct kinematics of the robot arm (function of q)

→ ${}^{B}T_E$ is the only thing I can use

I isolate it

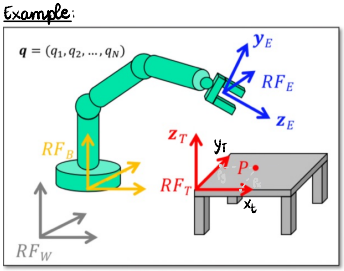
solve for q (inverse kinematics)

$$\boxed{{}^{B}T_E(q) = {}^{W}T_B^{-1}\ {}^{W}T_T\ {}^{E}T_T^{-1} = \text{constant}}$$

I want to find the q-s that satisfy the relation

I need also to define the direct kinematics.

Example:



- the robot carries a depth camera (e.g., a Kinect) on the end-effector
- the end-effector should go to a pose above the point P on the table, pointing its approach axis z_E downward and being aligned with the table sides

must be orthogonal to the others & det = +1.

$${}^{E}R_T = \left[{}^{E}x_T \quad {}^{E}y_T \quad {}^{E}z_T\right] = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{bmatrix}$$

- point P is known in the table frame RF_T

$${}^{T}p = \begin{bmatrix} p_x \\ p_y \\ 0 \end{bmatrix}$$

- the robot proceeds by centering point P in its camera image until it senses a depth h from the table (in RF_E) (TASK)

$${}^{E}p = \begin{bmatrix} 0 \\ 0 \\ h \end{bmatrix}$$

Q: where is the EE frame w.r.t. the table frame?

$${}^{T}T_E = \begin{bmatrix} {}^{T}R_E & {}^{T}p_{TE} \\ 0^T & 1 \end{bmatrix} = {}^{E}T_T^{-1}$$

with ${}^{T}R_E = \left({}^{E}R_T\right)^T = {}^{E}R_T$

$${}^{T}p_{TE} = {}^{T}p - {}^{T}R_E\ {}^{E}p = \begin{bmatrix} p_x \\ p_y \\ h \end{bmatrix}$$

$${}^{E}T_T = \begin{bmatrix} \overbrace{\begin{matrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{matrix}}^{{}^{E}R_T} & \begin{matrix} \Delta x \\ \Delta y \\ \Delta z \end{matrix} \\ 0\ 0\ 0 & 1 \end{bmatrix}$$

origin of table wrt origin of E.E.

<Task: move sensor until ${}^{E}p_H = \begin{bmatrix} 0 \\ 0 \\ h \\ 1 \end{bmatrix}$ >

$$\rightarrow {}^{T}p_H = \begin{bmatrix} p_x \\ p_y \\ 0 \\ 1 \end{bmatrix} = {}^{T}T_E\ {}^{E}p_H = {}^{E}T_T^{-1}\ {}^{E}p_H$$

Remarks on homogeneous matrices

- the main tool used for computing the direct kinematics of robot manipulators
- relevant in many other applications (in robotics and beyond)
  - in positioning/orienting a vision camera (matrix ${}^{b}T_c$ with extrinsic parameters of the camera pose)
  - in computer graphics, for the real-time visualization of 3D solid objects when changing the observation point

$${}^{A}T_B = \begin{bmatrix} & {}^{A}R_B & & {}^{A}p_{AB} \\ \alpha_x & \alpha_y & \alpha_z & \sigma \end{bmatrix}$$

all zero in robotics

coefficients of perspective deformation

scaling coefficient

always unitary in robotics

<(lunghezze unitarie)>

A robotic problem with T matrices

Task: 2R planar robot **A** should hand over an object at a given location to 3R planar robot **B**

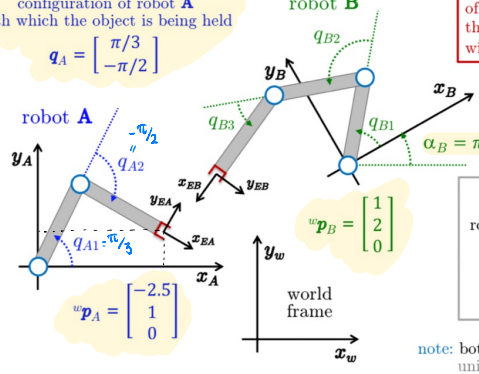
configuration of robot **A** with which the object is being held

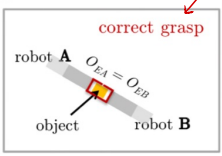
$$\boldsymbol{q}_A = \begin{bmatrix} \pi/3 \\ -\pi/2 \end{bmatrix}$$

Q: Find a configuration $\boldsymbol{q}_B$ of robot **B** so as to grasp the object held by robot **A** with the right orientation

q_{B1}, q_{B2}, q_{B3}

Ex #3, Robotics 1 exam of Sep 11, 2020





note: both robots have unitary link lengths

$$1)\quad {}^wT_A = \left(\begin{array}{c|c} {}^wR_A & {}^wp_A \\ \hline 0^T & 1 \end{array}\right) = \left(\begin{array}{c|c} I_{3\times3} & \begin{matrix} -2.5 \\ 1 \\ 0 \end{matrix} \\ \hline 0^T & 1 \end{array}\right)$$

$$2)\quad {}^wT_B = \left(\begin{array}{c|c} {}^wR_B & \begin{matrix} 1 \\ 2 \\ 0 \end{matrix} \\ \hline 0^T & 1 \end{array}\right) \qquad {}^wR_B = \begin{bmatrix} \cos\frac{\pi}{6} & -\sin\frac{\pi}{6} & 0 \\ \sin\frac{\pi}{6} & \cos\frac{\pi}{6} & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \sqrt{3}/2 & -1/2 & 0 \\ 1/2 & \sqrt{3}/2 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

3) EE(A) wrt base frame of A

$${}^AT_{EA} = \left(\begin{array}{c|c} {}^AR_{EA} & {}^Ap_{EA} \\ \hline 0^T & 1 \end{array}\right) \qquad {}^AR_{EA} = \begin{bmatrix} c(q_{A1}+q_{A2}) & -s(q_{A1}+q_{A2}) & 0 \\ s(q_{A1}+q_{A2}) & c(q_{A1}+q_{A2}) & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \sqrt{3}/2 & 1/2 & 0 \\ -1/2 & \sqrt{3}/2 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$${}^Ap_{EA} = \begin{pmatrix} c\,q_{A1} + c(q_{A1}+q_{A2}) \\ s\,q_{A1} + s(q_{A1}+q_{A2}) \\ 0 \\ 1 \end{pmatrix}$$

4) EE(B) wrt EE(A)

$$5)\quad {}^wT_A\,{}^AT_{EA}\,{}^{EA}T_{EB} = {}^wT_B\,{}^BT_{EB}$$

$$6)\quad {}^BT_{EB}\ \text{desired} = \left(\begin{array}{c|c} {}^BR_{EB,d} & {}^Bp_{EB,d} \\ \hline 0^T & 1 \end{array}\right) = \left({}^wT_B\right)^{-1}\,{}^wT_A\,{}^AT_{EA}\,{}^{EA}T_{EB}$$

$${}^w\boldsymbol{T}_A = \begin{pmatrix} {}^w\boldsymbol{R}_A & {}^w\boldsymbol{p}_A \\ \boldsymbol{0}^T & 1 \end{pmatrix} = \begin{pmatrix} \boldsymbol{I}_{3\times3} & \begin{matrix} -2.5 \\ 1 \\ 0 \end{matrix} \\ \boldsymbol{0}^T & 1 \end{pmatrix}$$

base frame of robot **A** w.r.t. world

$${}^w\boldsymbol{T}_B = \begin{pmatrix} {}^w\boldsymbol{R}_B & {}^w\boldsymbol{p}_B \\ \boldsymbol{0}^T & 1 \end{pmatrix} = \begin{pmatrix} \cos\alpha_B & -\sin\alpha_B & 0 & 1 \\ \sin\alpha_B & \cos\alpha_B & 0 & 2 \\ 0 & 0 & 1 & 0 \\ & \boldsymbol{0}^T & & 1 \end{pmatrix} = \begin{pmatrix} 0.8660 & -0.5 & 0 & 1 \\ 0.5 & 0.8660 & 0 & 2 \\ 0 & 0 & 1 & 0 \\ & \boldsymbol{0}^T & & 1 \end{pmatrix}$$

base frame of robot **B** w.r.t. world

$${}^A\boldsymbol{T}_{EA} = \begin{pmatrix} {}^A\boldsymbol{R}_{EA} & {}^A\boldsymbol{p}_{EA} \\ \boldsymbol{0}^T & 1 \end{pmatrix} = \begin{pmatrix} \cos(q_{A1}+q_{A2}) & -\sin(q_{A1}+q_{A2}) & 0 & \cos q_{A1} + \cos(q_{A1}+q_{A2}) \\ \sin(q_{A1}+q_{A2}) & \cos(q_{A1}+q_{A2}) & 0 & \sin q_{A1} + \sin(q_{A1}+q_{A2}) \\ 0 & 0 & 1 & 0 \\ & \boldsymbol{0}^T & & 1 \end{pmatrix} = \begin{pmatrix} 0.8660 & 0.5 & 0 & 1.3660 \\ -0.5 & 0.8660 & 0 & 0.3660 \\ 0 & 0 & 1 & 0 \\ & \boldsymbol{0}^T & & 1 \end{pmatrix}$$

end-effector frame of robot **A** w.r.t. its base frame (uses $\boldsymbol{q}_A$) = direct kinematics of robot **A**!

$${}^{EA}\boldsymbol{T}_{EB} = \begin{pmatrix} {}^{EA}\boldsymbol{R}_{EB} & {}^{EA}\boldsymbol{p}_{EB} \\ \boldsymbol{0}^T & 1 \end{pmatrix} = \begin{pmatrix} -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ & \boldsymbol{0}^T & & 1 \end{pmatrix}$$

z axis in common, x & y reversed

same origin

end-effector frame of robot **B** w.r.t. end-effector frame of robot **A** to realize the right grasp for correct handover

Solution procedure

$$\boxed{{}^w\boldsymbol{T}_A\,{}^A\boldsymbol{T}_{EA}\,{}^{EA}\boldsymbol{T}_{EB} = {}^w\boldsymbol{T}_B\,{}^B\boldsymbol{T}_{EB}}$$

kinematic equation defining the task

end-effector frame of robot **B** w.r.t. world passing via the given configuration of robot **A**	=	end-effector frame of robot **B** w.r.t. world passing via its base frame

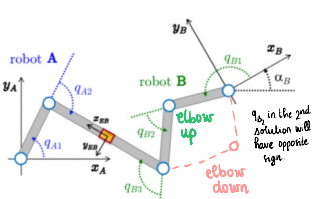
$${}^B\boldsymbol{T}_{EB,d} = \begin{pmatrix} {}^B\boldsymbol{R}_{EB,d} & {}^B\boldsymbol{p}_{EB,d} \\ \boldsymbol{0}^T & 1 \end{pmatrix} = ({}^w\boldsymbol{T}_B)^{-1}\,{}^w\boldsymbol{T}_A\,{}^A\boldsymbol{T}_{EA}\,{}^{EA}\boldsymbol{T}_{EB}$$

$$= \begin{pmatrix} -0.5 & -0.8660 & 0 & -2.1651 \\ 0.8660 & -0.5 & 0 & 0.5179 \\ 0 & 0 & 1 & 0 \\ & \boldsymbol{0}^T & & 1 \end{pmatrix} =$$

desired end-effector frame of robot **B** w.r.t. its base = input for the inverse kinematics of robot **B**!

one solution $\boldsymbol{q}_B$ (out of 2) of the inverse kinematics of robot **B**

$$\boldsymbol{q}_B = \begin{bmatrix} q_{B1} \\ q_{B2} \\ q_{B3} \end{bmatrix} = \begin{bmatrix} 2.7939 \\ 1.1076 \\ -1.8071 \end{bmatrix} [\text{rad}] = \begin{bmatrix} 160.08° \\ 63.46° \\ -103.54° \end{bmatrix}$$



Direct Kinematics

Kinematics of robot manipulators

study of geometric & timing aspects of robot motion, w\ reference to the causes producing it

what happens when I move a link?... ↳ linking cause & effect : dynamic model

robot seen as an (open) kinematic chain of rigid bodies interconnected by R or P joints.

↳ structure seen as skeleton. ↳ links ↳ 1 dof / joint

transmissions : perfectly stiff.

Motivations : functional aspects : definition of robot workspace ; calibration → starting from nominal values of a parameter → improves accuracy

→ important to solve the inverse pb.

- primary (reachable) workspace
- secondary workspace

operational aspects :

task execution (actuation by motors) ⇄ (inverse / direct) task definition and performance

two different "spaces" related by kinematic (and dynamic) maps

- trajectory planning
- programming
- motion control

↳ based on error we correct plan & motion

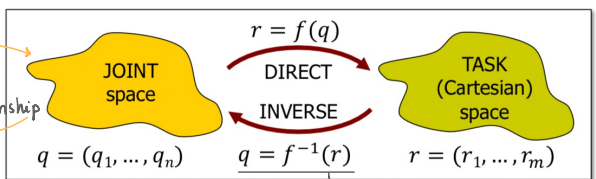
I apply commands in the joint space.

I need to evaluate performance (cartesian space - generally) ↔ 'TASK SPACE'

Formulation & parametrizations

There is actually a 3rd space : Important in dynamics.

ACTUAT-ORS SPACE — 1:1 relationship



↳ optimistically (it's not always that easy).

- choice of parameterization q

somewhat arbitrary, but needs to be :

- unambiguous and minimal characterization of robot configuration
- n = # degrees of freedom (dof) = # robot joints (rotational or translational)

- choice of parameterization r

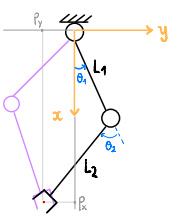
m : dimension of the task

e.g. : planar → 2 coords. for position + 1 angle : m = 3

- compact description of position and/or orientation (pose) variables of interest to the required task
- usually, $m \leq n$ and $m = 6$ (but none of these is strictly necessary)

↳ cartesian pose (3D space)

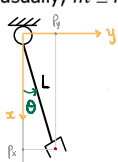
if m<n → n-m redundant dof.



⚠ P_x & P_y may also represent ↳ ambiguous

same problem for $\theta_2 = 0$ →

$$q = \begin{pmatrix} \theta_1 \\ \theta_2 \end{pmatrix}$$



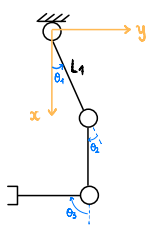
$$q = \begin{pmatrix} P_x \\ P_y \end{pmatrix}$$

: unambiguous but NOT MINIMAL

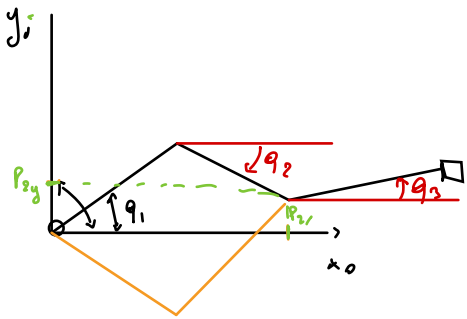
$P_x^2 + P_y^2 = L^2$: constraint between the two → I can eliminate one

I will take θ as only variable $\theta \in [-\pi, \pi)$

$q = \theta$



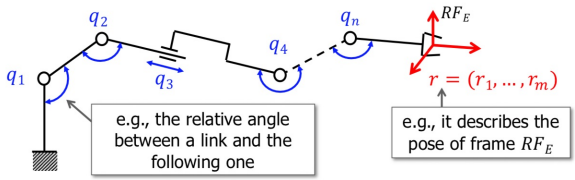
$$q = \begin{pmatrix} \theta_1 \\ \theta_2 \\ \theta_3 \end{pmatrix}$$



$$P_x^2 + P_y^2 = L^2$$

$$q' = \begin{pmatrix} P_{2x} \\ P_{2y} \\ q_3 \end{pmatrix}$$

Open kinematic chains

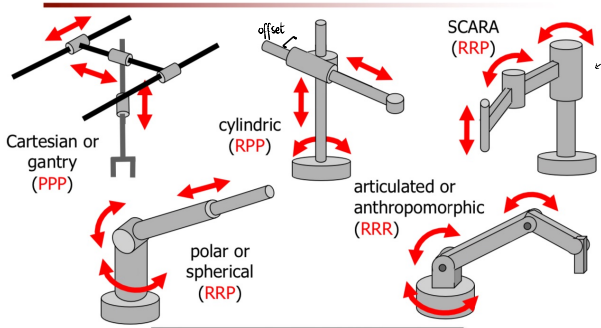


- $m = 2$ → I need only 2 variables for:
  - pointing in space (orientation)
  - or positioning in the plane
- $m = 3$
  - orientation in space
  - or positioning and orientation in the plane
- $m = 5$
  - positioning (3) and pointing (2) in space (like for spot welding) (6 dof robot would be redundant)
- $m = 6$
  - positioning and orientation in space
  - or positioning of two points in space (e.g., end-effector and elbow)

avoid collision

Classification by kinematic type

first 3 dofs only



reduced representation (usually RRPR : 4 dof - w/gripper)

(in direct kinematics, I assume I can go everywhere, & then I limit the workspace)

R = 1-dof rotational (revolute) joint
P = 1-dof translational (prismatic) joint

Direct kinematic map

- the structure of the direct kinematics function depends on the chosen r

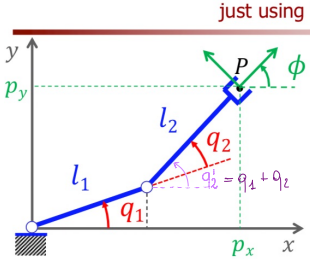
$$r = f_r(q)$$

If I change the task the function will change.

- methods for computing $f_r(q)$
  - geometric/by inspection
  - systematic: assigning frames attached to the robot links and using homogeneous transformation matrices

Direct kinematics of 2R planar robot

just using inspection...



$$q = \begin{bmatrix} q_1 \\ q_2 \end{bmatrix} \qquad n = 2$$

$$r = \begin{bmatrix} p_x \\ p_y \\ \phi \end{bmatrix} \qquad m = 3$$

$$\boxed{\begin{aligned} p_x &= l_1 \cos q_1 + l_2 \cos(\overbrace{q_1 + q_2}^{q_2'}) \\ p_y &= l_1 \sin q_1 + l_2 \sin(\underbrace{q_1 + q_2}_{q_2'}) \\ \phi &= q_1 + q_2 = q_2' \end{aligned}}$$

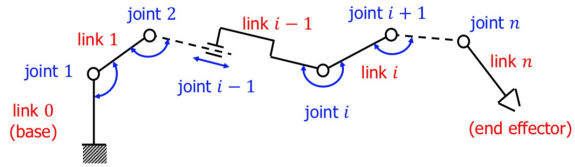
→ $n < m$ → I get rid of the 3rd eq.

I could have used $q' = \begin{pmatrix} q_1' \\ q_2' \end{pmatrix}$ ↳ absolute wrt base axis x.

so $l_1 = l_2 \Rightarrow$ ↳ $2l \cos q_1 + l \cos q_2 = l(2\cos q_1 + \cos q_2)$

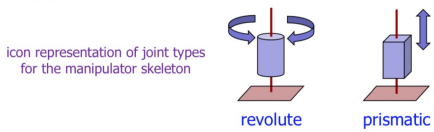
for more general cases, we need a 'method'! ⟶ DH convention

Numbering links & joints

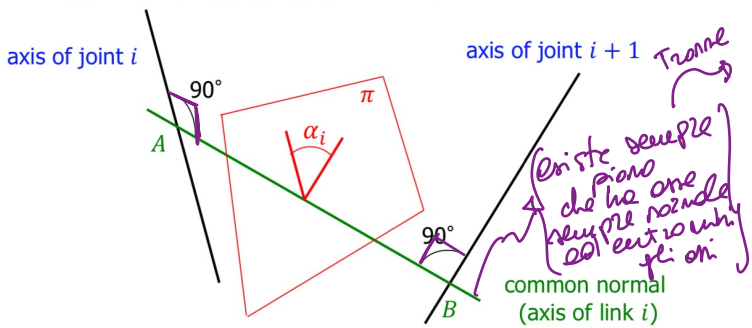


$i = 1, \dots, n$

$n+1$ bodies (links), n joints



Spatial relation between joint axes



Trovare

(esiste sempre che ha una sempre normale ad entrambi gli assi)

2 skew axes have 1 common normals.

special cases:

$\parallel$: ∞ common normal

$\perp$: 1 common normal, lenght $\emptyset$

$\times$

$|$: coincident : ∞^2 common normals w/ lenght $\emptyset$

a_i = **displacement** AB between joint axes (always well defined)

α_i = **twist angle** between joint axes

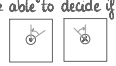
— projected on a plane π orthogonal to the link axis

always constant!

with sign (pos/neg)!

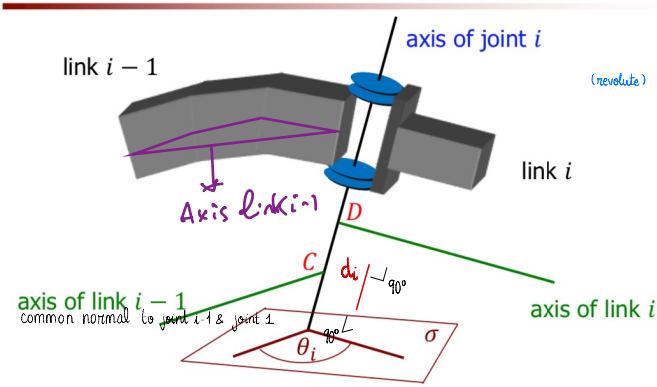
- depending on orientation
- depending on wrt which axis I'm doing the rotation

↳ depending on where the z-axis is oriented to be able to decide if it's a ⊕ or ⊖ rot.



↳ from // axes (∞ normals), when I twist (→ 1 normal), I define a twist angle.

Spatial relation between link axes



d_i = **displacement** CD (a variable if joint i is prismatic)

θ_i = **angle between link axes** (a variable if joint i is revolute)

— projected on a plane σ orthogonal to the joint axis

constant if joint is R

with sign (pos/neg)!

constant if joint is P