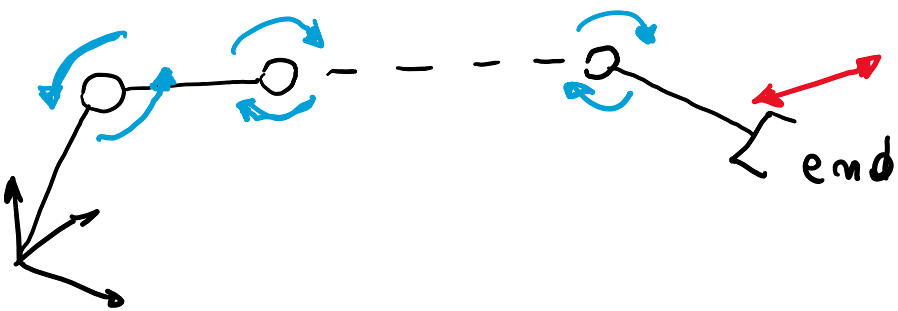
Slides have been created by Prof. Michele Focchi

webpage: https://mfocchi.github.io/Teaching/

E1-Direct Kinematics of a Manipulator

KINEMATICS describes the motion without considering the forces that cause it



by moving its joints the robot changes the position/orientation of the end-effector (... and also of all the other points of the robot)

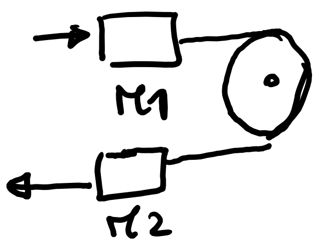
DIRECT KINEMATICS: compute position/orientation of a frame (e.g. at end-effector) as a function of joint variables q

KINEMATIC SPACES

To have a motion of the joints we need to actuate the actuators

actuation space JOINT SPACE TASK SPACE

Joint space and actuation space can have different dimensions:



e.g. Antagonistic actuators

Two motors move 1 joint

We assume 1 motor per joint (dim JOINT SPACE = dim actuation space) and describe the joint space with GENERALIZED COORDINATES

GENERALIZED COORDINATES

- Suppose we have a system of N rigid bodies (e.g. links) free to move in space.

- The motion of the system is described by 6N coords. x

- If we limit the motion of the links inserting joints they eliminate 5 out of 6 relative degrees of fredom between any 2 links $\Rightarrow$ a constraint exists

$R(x) = 0$ $\rightarrow$ holonomic (depends on position not on velocity)

$\rightarrow$ 5 scalar equations

$\Rightarrow$ we can eliminate s coordinates remaining with $n = 6N - s$ coordinates that are called generalized coordinates

- represent motion of system implicitly taking into account constraints

- n : number of degrees of freedom of the mechanical system

- They minimal and unambiguous

- a vector of generalized coordinates can contain both linear and angular quantities (depending on joint nature)

KINEMATICS : PARAMETRIZATIONS

Joint space — DIRECT $r = f(q)$ — TASK SPACE

$q = (q_1, \dots, q_n)$ — INVERSE $q = f^{-1}(r)$ — $r = (r_1, \dots, r_m)$

JOINT SPACE: space of the joints defined by the vector of joint variables:

$$q = (q_1, \dots, q_n)$$

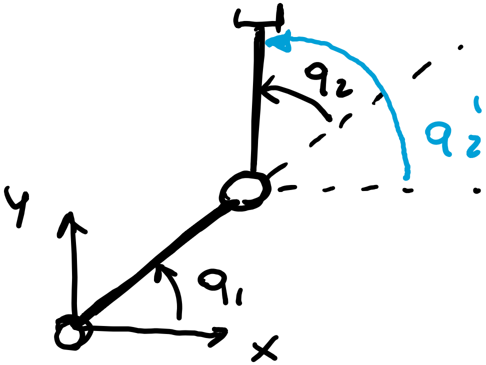
choice of parametrization $q \in \mathbb{R}^n$

Generalized coordinates: minimum and unambiguous number of coordinates that represent the motion taking into account kinematic constraints (joints)

$$\boxed{n = \#\ \text{Degrees of Freedom (DoFs)} = \#\ \text{robot joints}}$$

EXAMPLES OF PARAMETRIZATION CHOICE

2 DOFs PLANAR

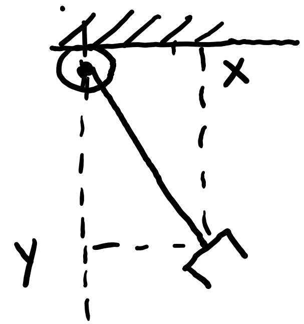


q_2 is relative

q_2' is absolute

ok

1 DOF PLANAR



x, y is minimal? NO

need to add constraint

$x^2 + y^2 = L^2$

→ need 1 VARIABLE

TASK / OPERATIONAL SPACE

- space where a task is defined and trajectories are planned
  e.g. move the end-effector on a plane (2D task)
  drill a hole ⊥ to a surface (1D task)

⇒ Direct kinematics is defined <u>for</u> <u>each</u> task

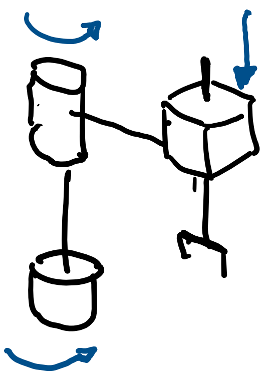
EXAMPLE OF TASKS

- move position of the end-effector
- move position AND orientation of end-effector
- move camera such that some features in the image plane are tracking some desired ones
- Tasks are not limited to cartesian space!

choice of parametrization for $r \in \mathbb{R}^m$

- compact description of the position / orientation variables of interest for the required task (unambiguous and minimal)
- usually $m < 6$

EXAMPLE : SCARA ROBOT



$n = 3$

$m = 6$

from q can I compute r ?

- YES $\longrightarrow$ 3 out of 6 variables are FIXED
- I cannot compute inverse kinematics
- if $m \leq n$ $\Rightarrow$ We are guaranteed that we have enough DoFs to place the task variables arbitrarily

TYPICAL INDUSTRIAL CASE

$m = n = 6$

if $m = 6$ $n = 7$ $\Rightarrow$ redundancy

- for a serial manipulators:

$q \rightarrow \boxed{\text{DIRECT KIN.}} \rightarrow r$

Always has a solution, easy

$$\boxed{r = f_r(q)}$$

$r \rightarrow \boxed{\text{INVERSE KIN.}} \rightarrow q$

hard

(A) multiple / ∞ solutions
(B) no solution
(C) needs FK to be computed

- for a parallel manipulator:

DIRECT KIN. : hard   INVERSE KIN. : easy

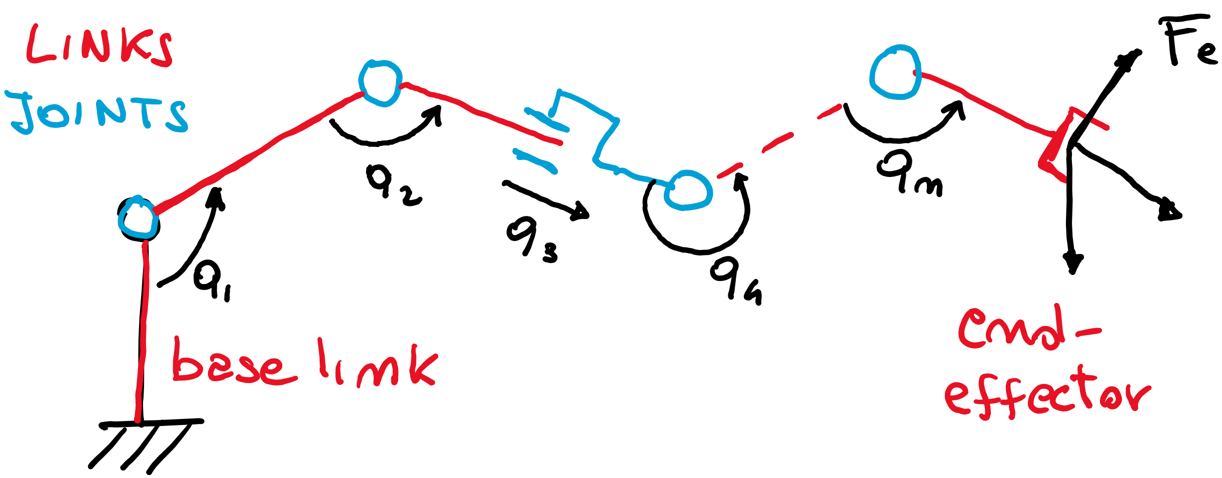
• how to compute direct kinematic map $r = f_r(q)$?

in a manipulator:

A) A ref. frame is attached to each link / joint

B) links are numbered from fixed base (link $\emptyset$) to end-effector (link n)

C) joints connect links and allow their relative motion



$\boxed{q : \text{joint variables}}$

$\boxed{r = (r_1, \ldots, r_m) \text{ Task variables:}}$

example: position/orientation of the frame Fe

IDEA: we can define transforms between each frame and the next and concatenate them

HOMOGENEOUS TRANSFORMS

- Unique ROTO-TRANSLATION operator that combines the representation of position ⊕ orientation (POSE) of a rigid body w.r.t another one

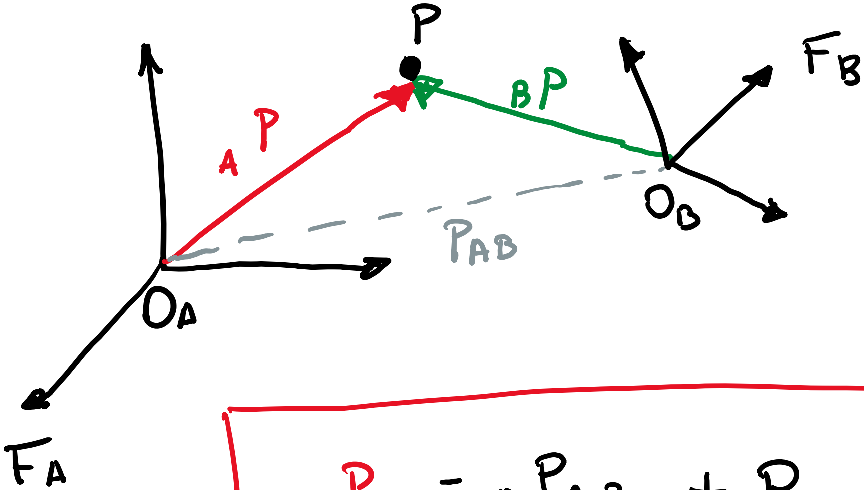
POSE

$\boxed{\text{in } \mathbb{R}^3}$: 3 position parameters
3 orientation params

$\boxed{\text{in } \mathbb{R}^2}$: 2 position params
1 orientation param

- basic tool to define direct kinematics of a manipulator

- 4 x 4 matrix that allows to compose roto-translations as we did for rotations



frame B has different orientation and its origin is displaced w.r.t F_A

$$\boxed{{}_A P = {}_A P_{AB} + {}_A R_B \, {}_B P}$$

express point p in frame ∅

→ affine operation : mutiply + sum

→ vectors should be summed in the **same** frame (i.e. A frame)

if we rewrite in homogeneous coordinates (append a scale factor):

orient. of F_B wrt F_A

$$ {}_A P_R = \begin{bmatrix} {}_A P \\ \cdots \\ 1 \end{bmatrix} = \left[\begin{array}{c|c} {}_A R_B & {}_A P_{AB} \\ \hline 0 \;\; 0 \;\; 0 & 1 \end{array}\right] \begin{bmatrix} {}_B P \\ \cdots \\ 1 \end{bmatrix} = \boxed{{}_A T_B} \, {}_B P_R $$

is a linear operator!

vector represented in homogeneous coords with scale factor 1

homogeneous Transform (4×4)

origin of B w.r.t A expressed in A

INTEPRETATIONS OF T

① describe pose (position/orientation) of a frame wrt another

② Tranform the representation of a geometric vector from a frame to another

- ${}_AT_B$ Transforms a geometric vector (from origin O_B To P) into another geometric vector (from origin O_A to P)
- ${}_AT_B$ is a linear relationship $\rightarrow$ I can compose several T

COMPOSITION OF HOMOGENEOUS TRANSFORMS

$$
{}_A T_B \; {}_B T_C = \begin{bmatrix} {}_A R_B & {}_A P_{AB} \\ 0^T & 1 \end{bmatrix} \begin{bmatrix} {}_B R_C & {}_B P_{BC} \\ 0^T & 1 \end{bmatrix} =
$$

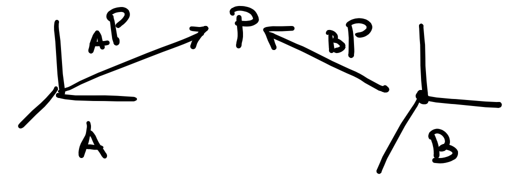
$$
= \left[\begin{array}{c|c} {}_A R_B \, {}_B R_C & {}_A R_B \, {}_B P_{BC} + {}_A P_{AB} \\ \hline 0^T & 1 \end{array}\right] = \begin{bmatrix} {}_A R_C & {}_A P_{AC} \\ 0^T & 1 \end{bmatrix} = {}_A T_C
$$

$$
\boxed{{}_A T_{\cancel{B}} \; {}_{\cancel{B}} T_C = {}_A T_C}
$$

generalization of what we did for rotations but frames have different origins

$\rightarrow$ do not commute!

INVERSE OF A HOMOGENEOUS TRANSFORM



$({}_AT_B)^{-1} = {}_BT_A$ always invertible

Proof ${}_BT_A = {}_AT_B^{-1}$

$${}_AT_B = \begin{bmatrix} {}_AR_B & {}_Ap_{AB} \\ 0^T & 1 \end{bmatrix} \Rightarrow {}_AP = {}_Ap_{AB} + {}_AR_B\, {}_BP$$

$${}_BT_A = \begin{bmatrix} {}_BR_A & {}_Bp_{BA} \\ 0^T & 1 \end{bmatrix}$$

$$\Downarrow$$

$${}_BP = {}_Bp_{BA} + {}_BR_A\, {}_AP = -{}_BR_A\, {}_Ap_{AB} + {}_BR_A\, {}_AP =$$

$$= -{}_AR_B^T\, {}_Ap_{AB} + {}_AR_B^T\, {}_AP$$

$$\Downarrow$$

- check ${}_AT_B^{-1}\, {}_AT_B = \begin{bmatrix} I & 0 \\ 0^T & 1 \end{bmatrix}$
- explicit expression for the inverse

$$\left[\begin{array}{c|c} {}_AR_B^T & -{}_AR_B^T\, {}_Ap_{AB} \\ \hline 0^T & 1 \end{array}\right] = ({}_AT_B)^{-1}$$

BASIC TRANSFORMS

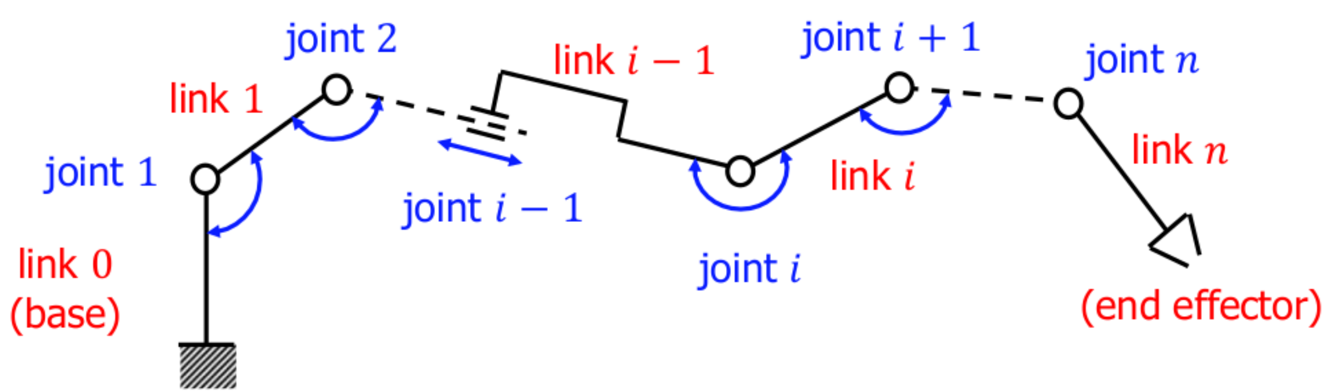
pure rotation: $T(R,0) = \begin{bmatrix} R & 0 \\ 0^T & 1 \end{bmatrix}$

pure translation: $T(0,t) = \begin{bmatrix} I & t \\ 0^T & 1 \end{bmatrix}$

FRAME PLACEMENT

- Frames can be arbitrarily chosen as long as they are rigidly attached to the links
- Denavit - Hartenberg convention is a systematic way to define the link frames that requires only 4 parameters.
  - ⊕ faster computation
  - ⊖ you cannot choose any placement
  - ⊖ no longer used (we use URDF model of the robot that allows generic frame placement)

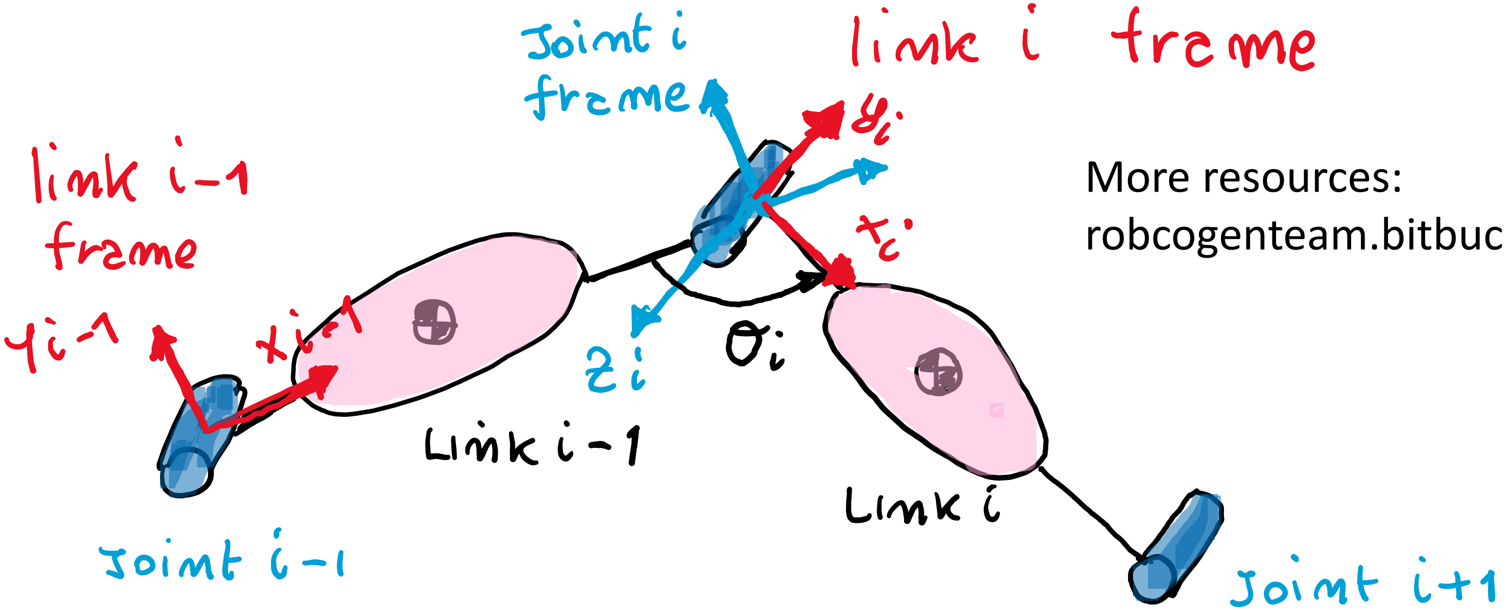
Numbering of joints and links



each joint has 1 axis:

revolute

prismatic



More resources:
robcogenteam.bitbucket.io/rmodel.html

link i frame: moves with link i and has origin coincident with the supporting joint i

- x axis along link main direction

joint i frame:

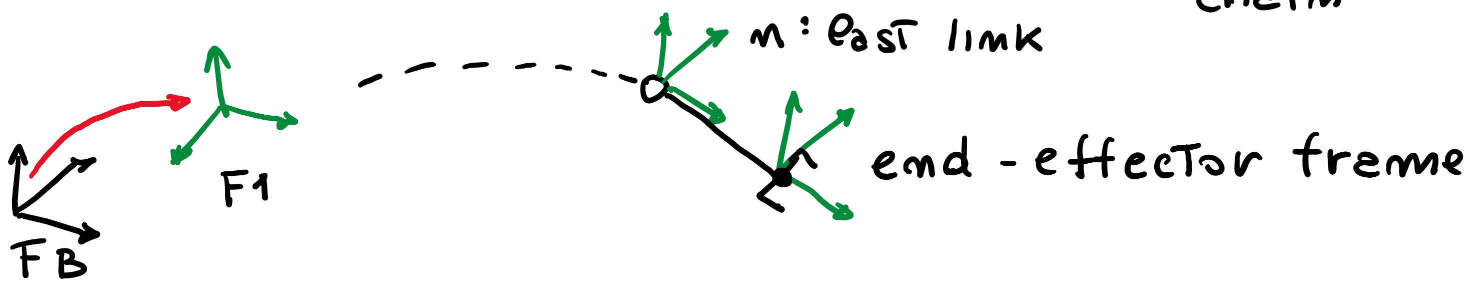
- z axis as rotational axis (revolute joint)
- pose specified as a rigid Tranform w.r.t The predesessor link i-1

z_i ↺ θ_i : angle between x_{i-1} and x_i

z_i ↑ d_i : distance between x_{i-1} and x_i

DIRECT KINEMATICS OF A MANIPULATOR

since we have a frame attached to each link we can compose the Transforms from base to end-effector to obtain the DIRECT KINEMATICS (for serial manipulators) chain



$$\boxed{{}_B T_e = {}_0T_1(q_1)\ {}_1T_2(q_2) \cdots {}_{n-1}T_n(q_n)\ {}_nT_e}$$

$\llcorner$ ${}_nT_e$: fixed Transform

${}_{i-1}T_i(q_i)$: each elementary Tranformation is a direct function of only The correspondent Joint That drives The link

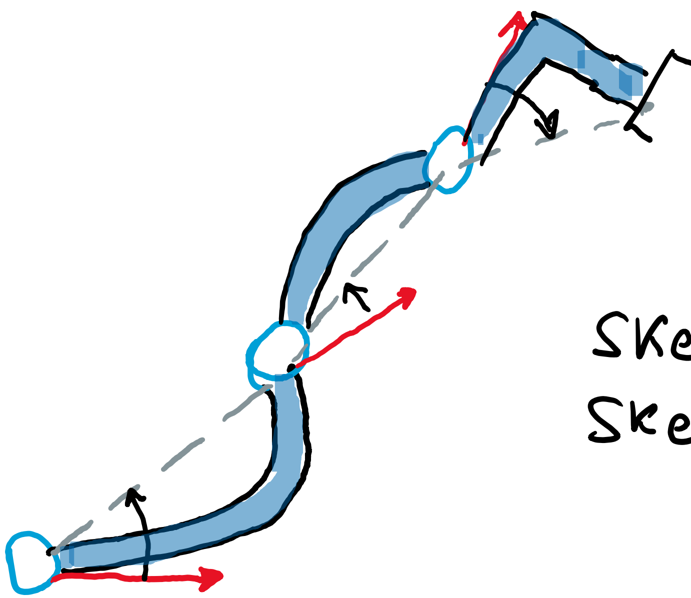
$${}_BT_n = \begin{bmatrix} {}_BR_e & {}_BP_{ee} \\ 0^T & 1 \end{bmatrix} \qquad \boxed{r = \begin{bmatrix} x \\ \phi \end{bmatrix} = f(q)}$$

${}_BR_e \to \phi$: map To Euler

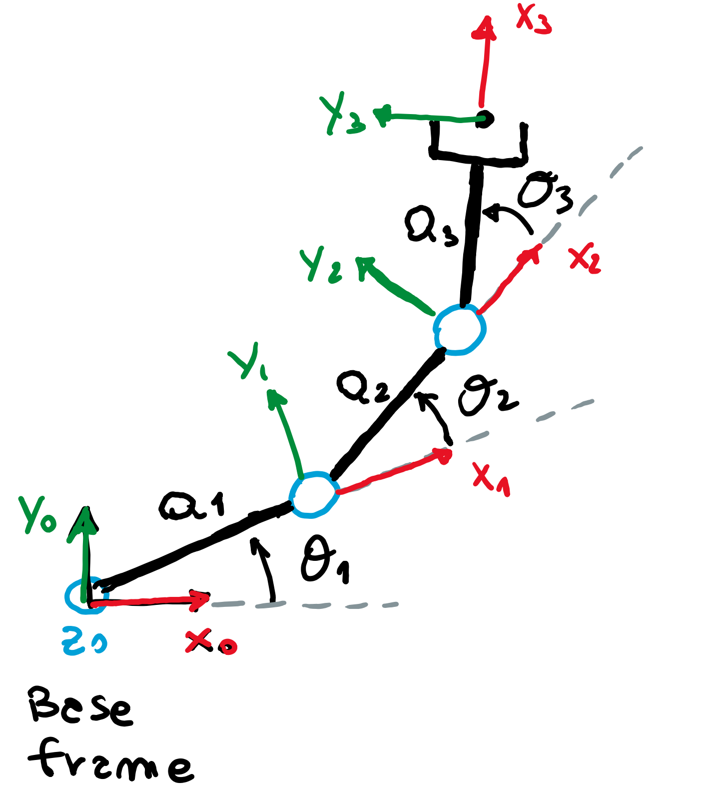
alternative form of of direct kinematics as a non-linear operator

EXAMPLE : 3 DOFs PLANAR ROBOT (USE INTUITION)

- Typical manipulator structure often encountered in industrial robots
- direct. kin. can be easily obtained with some geometric intuition



⇒ Skeleton Sketch

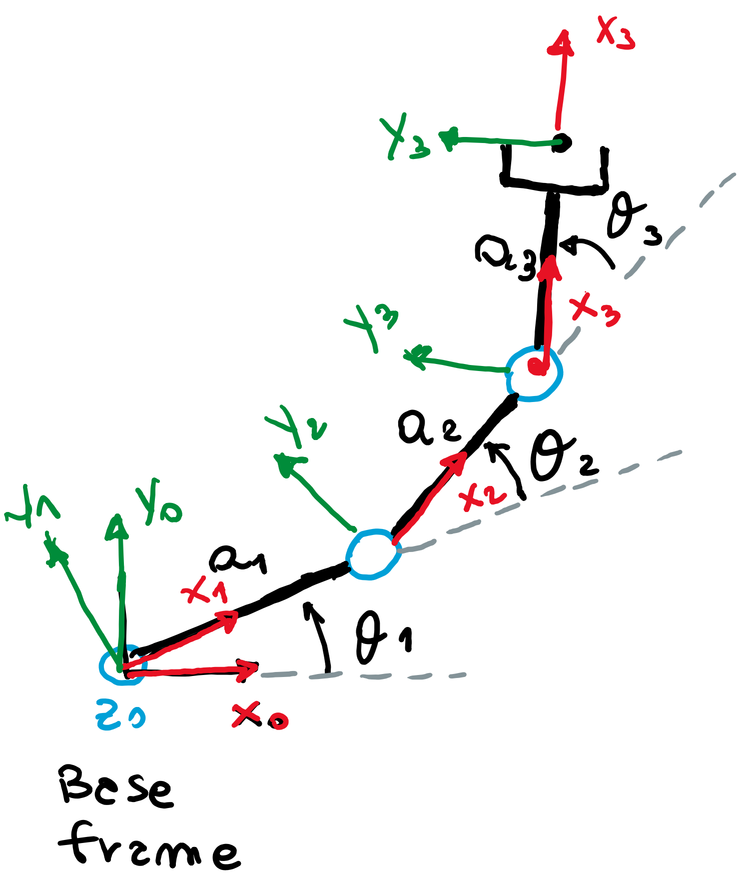


Task : position / orientation of end-effector

\# DOFs = 3 $\qquad q = (\theta_1, \theta_2, \theta_3)$

OBSERVATIONS

- revolute joints' axes are parallel
- links are on a plane x_0, y_0
- x_i are along the link directions



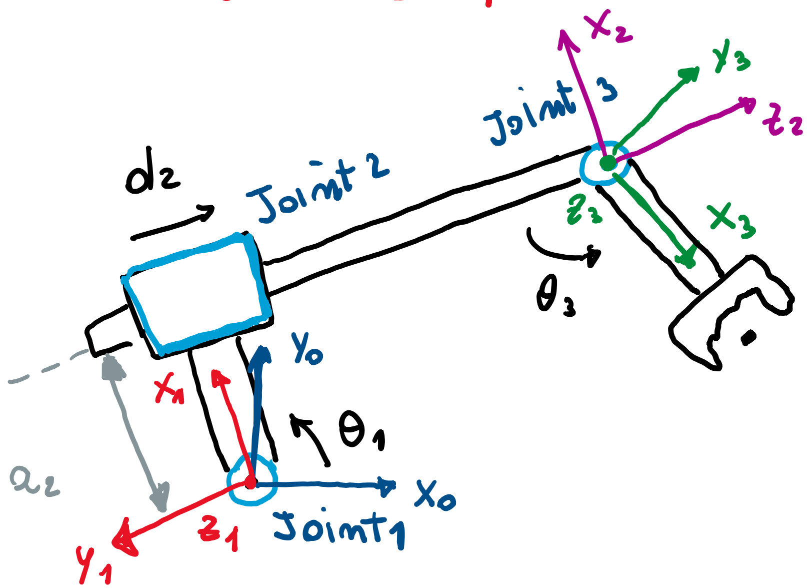
$$
{}_0P_3 = \begin{bmatrix} a_1 c_1 + a_2 c_{12} + a_3 c_{123} \\ a_1 s_1 + a_2 s_{12} + a_3 s_{123} \\ 0 \end{bmatrix}
$$

not a function of q (planar)

$$
{}_0R_3 = R_z(q_1+q_2+q_3) = \begin{bmatrix} c_{123} & -s_{123} & 0 \\ s_{123} & c_{123} & 0 \\ 0 & 0 & 1 \end{bmatrix}
$$

$$
\Rightarrow {}_0T_3 = \begin{bmatrix} {}_0R_3 & {}_0P_3 \\ 0^T & 1 \end{bmatrix}
$$

EXERCISE 4: RPR ROBOT (USE HOM. TRANSFORMS)



links' frames placement

z_i ↻ θ_i : angle between x_{i-1} and x_i

$q_i = \theta_i$ revolute joint

z_i ↑ d_i : distance between x_{i-1} and x_i

$q_i = d_i$ prismatic joint

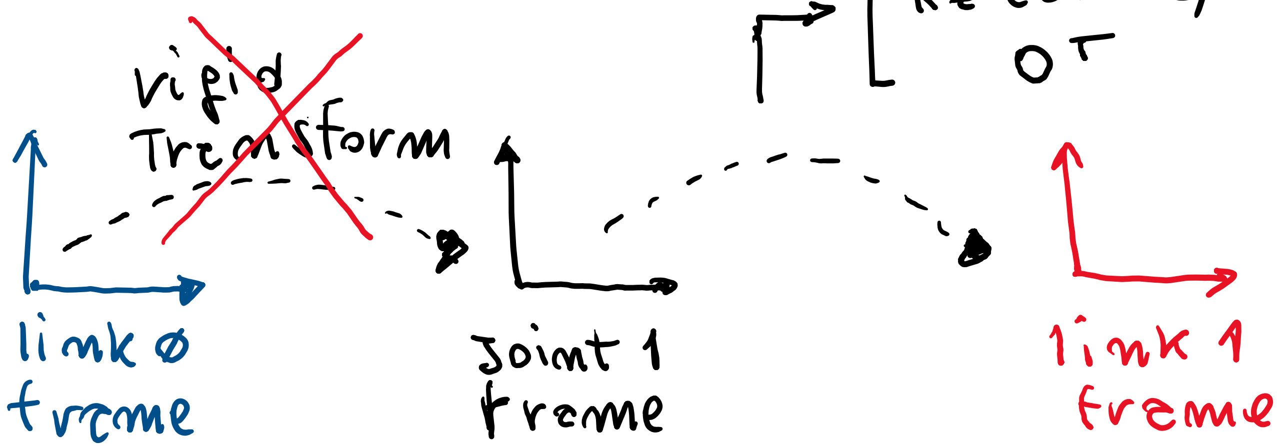
HOMOGENEOUS TRANSFORMS (BETWEEN LINKS)

$$
{}^0T_1 = \begin{bmatrix} c_1 & -s_1 & 0 & 0 \\ s_1 & c_1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}
$$

rotation around z_0

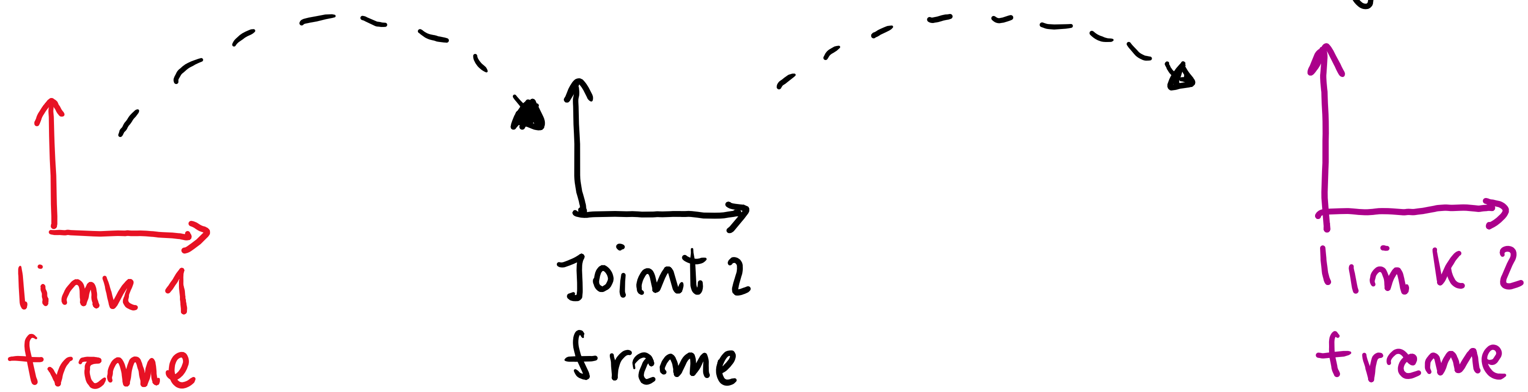
coincident origins

$$
\rightarrow \begin{bmatrix} R_z(z_1 = z_0) & 0 \\ 0^T & 1 \end{bmatrix}
$$



rigid Transform

$\begin{bmatrix} I & t \\ 0 & 1 \end{bmatrix}$ pure Translation along z_2 ($= x_1$)



- we merge together rigid Transform and Traslational Transform due To prismatic joint

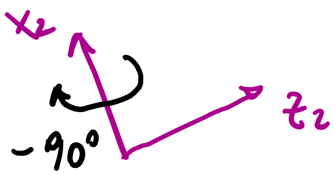
axes of frame 2 seen from 1 (expressed in 1)

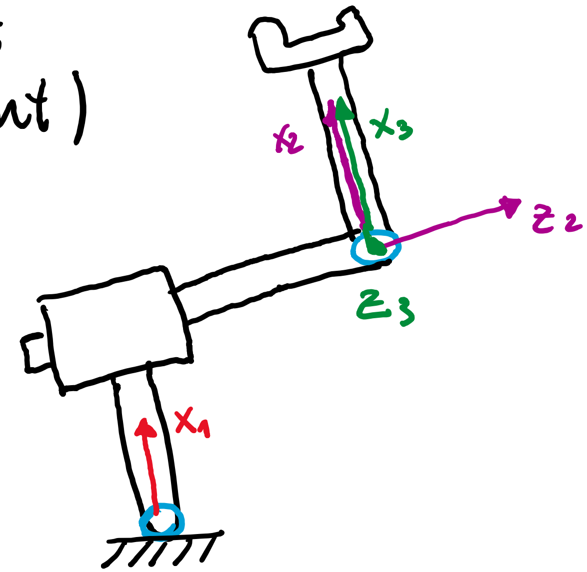
$$ {}_1T_2 = \begin{bmatrix} 1 & 0 & 0 & a_2 \\ 0 & 0 & -1 & -d_2 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

$$\underbrace{\underbrace{\begin{matrix} x_2 = x_1 \\ y_2 = z_1 \\ z_2 = -y_1 \end{matrix}}_{\text{rotation}} \qquad \underbrace{\begin{matrix} a_2 \text{ along } x_1 \\ 0 \text{ along } y_1 \\ 0 \text{ along } z_1 \end{matrix}}_{\text{Translation}}}_{\text{rigid Tranform}} \quad \oplus \quad \begin{matrix} -d_2 \text{ along } y_1 \\ \text{Pure Joint} \\ \text{Translation} \end{matrix}$$

From frame 2 To frame 3 ? first set

$q_3 = 0$ (x_2, x_3 axes become coincident)





① $-90°$ around x_2 (first apply The rigid Tramform)

$$R_x(-90°) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & -1 & 0 \end{bmatrix}$$

② θ_3 around z_3

$$R_z(q_3) = \begin{bmatrix} c_3 & -s_3 & 0 \\ s_3 & c_3 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$${}_2R_3 = R_x(-90°)\, R_z(q_3) = \begin{bmatrix} c_3 & -s_3 & 0 \\ 0 & 0 & 1 \\ -s_3 & -c_3 & 0 \end{bmatrix}$$

Tranlation : no $\longrightarrow$

$${}_2T_3 = \begin{bmatrix} c_3 & -s_3 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -s_3 & -c_3 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$ {}_0T_2 = {}_0T_1\, {}_1T_2 = \begin{bmatrix} c_1 & 0 & s_1 & a_2 c_1 + d_2 s_1 \\ s_1 & 0 & -c_1 & a_2 s_1 - d_2 c_1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

$$ -(s_1 c_3 + c_1 s_3) \xleftarrow{B} $$

$$ c_1 c_3 - s_1 s_3 \xleftarrow{A} $$

$$ {}_0T_3 = {}_0T_2\, {}_2T_3 = \begin{bmatrix} c_1 c_3 - s_1 s_3 & -c_1 s_3 - s_1 c_3 & 0 & a_2 c_1 + d_2 s_1 \\ s_1 c_3 + c_1 s_3 & -s_1 s_3 + c_1 c_3 & 0 & a_2 s_1 - d_2 c_1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

(+) trigon. relationships:

(A) $c_\alpha c_\beta - s_\alpha s_\beta = c(\alpha + \beta)$

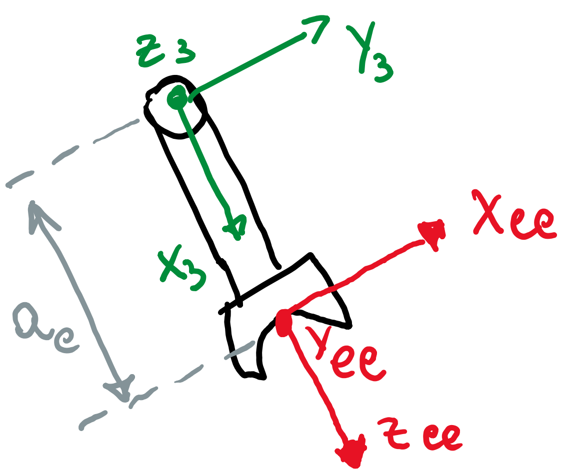
(B) $s_\alpha c_\beta + c_\alpha s_\beta = s(\alpha + \beta)$

$$ {}_0T_3 = \begin{bmatrix} c_{13} & -s_{13} & 0 & a_2 c_1 + d_2 s_1 \\ s_{13} & c_{13} & 0 & a_2 s_1 - d_2 c_1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

frame 3 is obtained with a rotation around z_0

position of end-effector in F_0 (base frame)

To obtain the trasform up To The end effector we need To define e frame for That and compute The rigid Tranform:

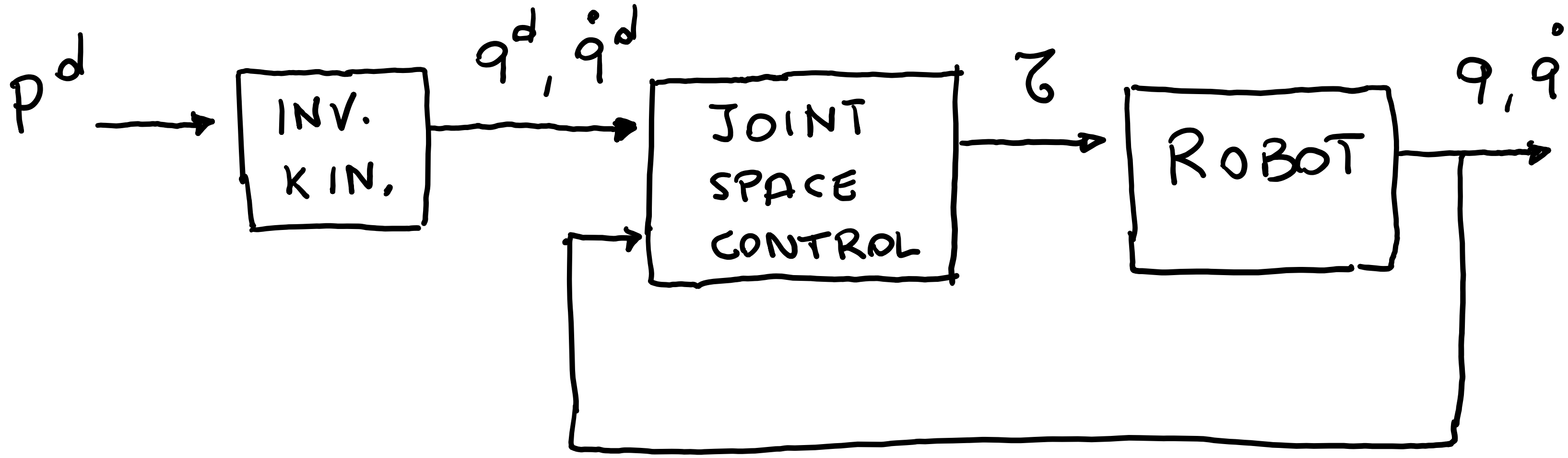


$$
{}_3T_{ee} = \begin{bmatrix} 0 & 0 & 1 & a_e \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}
$$

$$
{}_0T_{ee} = {}_0T_3 \, {}_3T_{ee}
$$

E2-Inverse Kinematics
(A.K.A Inverse geometry)

Motivation



2 APPROACHES:

① symbolic IK (This lecture)
- write down equations of FWD. KIN
- Try To solve Them explicitally

② numerical IK (lecture E4)
- no need of analytical expression of FWD. KIN.
- find q by evaluating FK several times
- formulated as an optimization problem

Problem: given a point on the robot (eg end effector pose) determine the joint variables that will realize it

inputs:

$${}_0T_n = \begin{bmatrix} R(q) & P(q) \\ 0^T & 1 \end{bmatrix} : \text{IK for a given end-effector pose } T$$

$$r = f(q) \quad : \text{IK for a given task } r \text{ (eg elbow position)}$$

- equations are tipically non-linear and $m < n$
  $\Rightarrow$ m equations in n variables

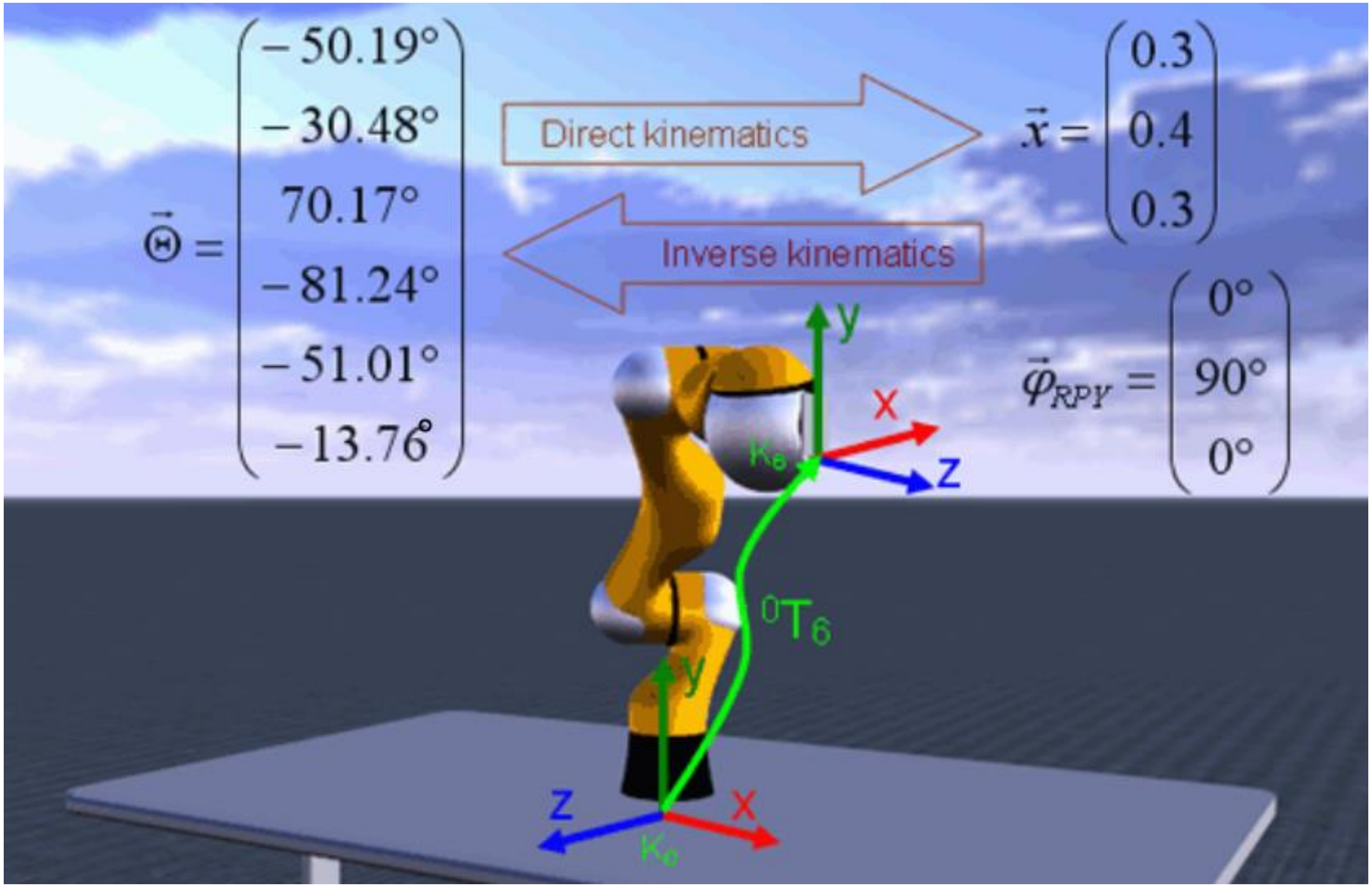
Q1: a solution exists? (workspace definition)

Q2: if a solution exists, is unique or $\exists$ multiple solutions? (ie. multiple joint config. $\Leftrightarrow$ same pose)

Q3: can I write them in closed form?

$$T/r \Rightarrow \boxed{IK} \Rightarrow q$$

Example KUKA Robot (6R)



direct kinematics is always unique; inverse kinematics not…multiple configurations can give the same pose of the end effector

Q1- SOLVABILITY: DO ANY SOLUTION EXIST?

A solution exists if p is within the reachable work space

REACHABLE WORKSPACE WS_1: positions of end-effector that can be reached with at least one orientation

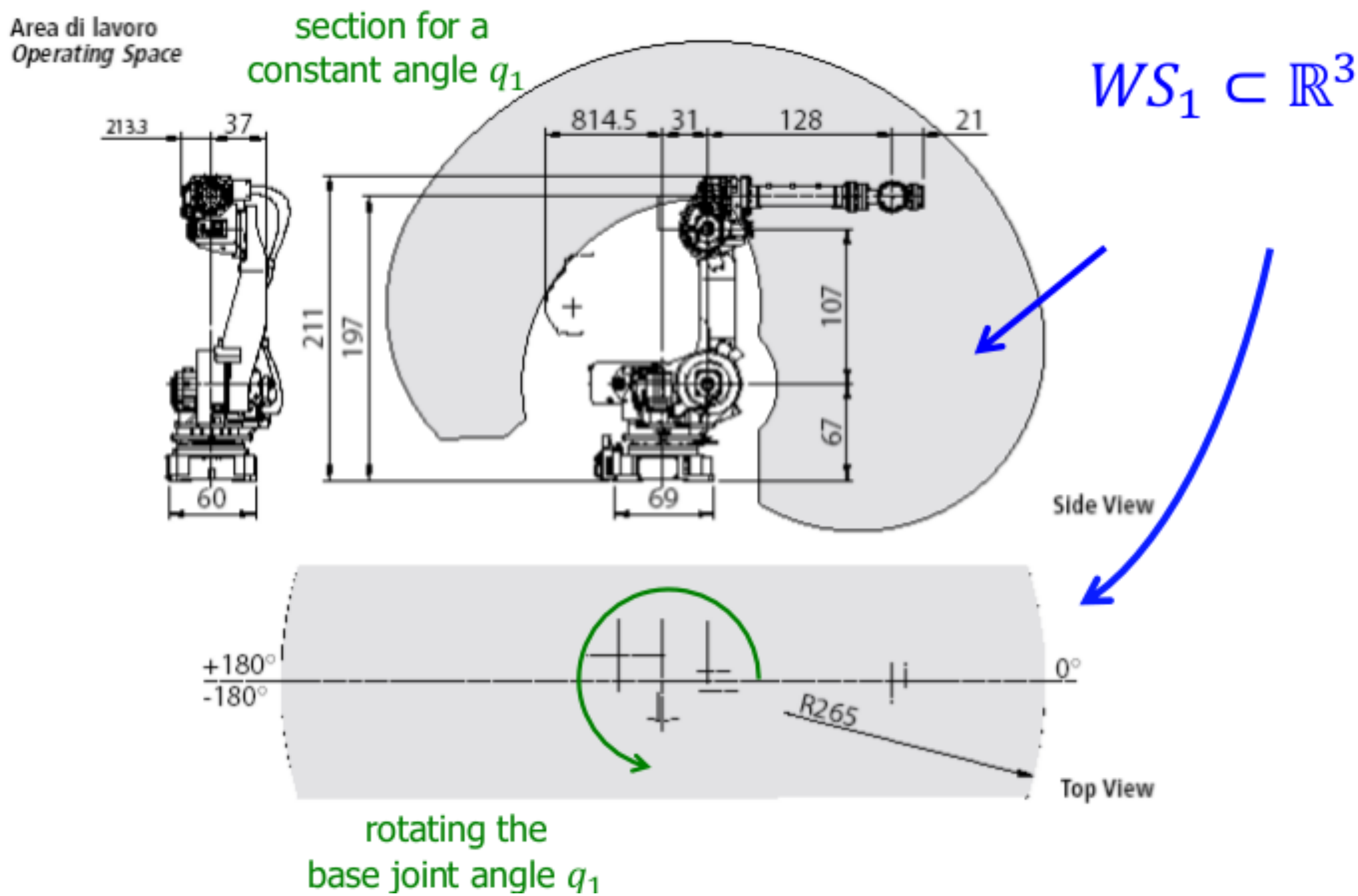
- if $q \notin WS1 \Rightarrow$ no solution to IK
- if $q \in WS1 \Rightarrow \exists$ solution but I cannot realize any orientation

DEXTEROUS WORKSPACE WS2 positions of the end-effector for which any orientation can be realized (among the ones we can realize from direct kinematics)

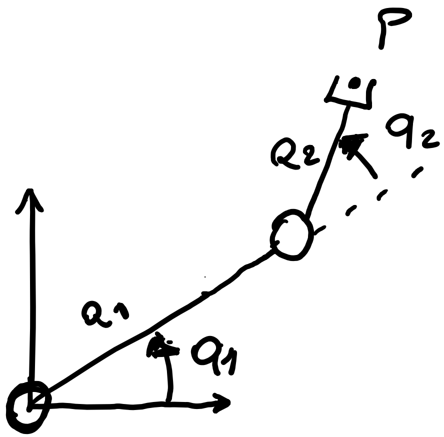
$\Rightarrow WS2 \subseteq WS1$

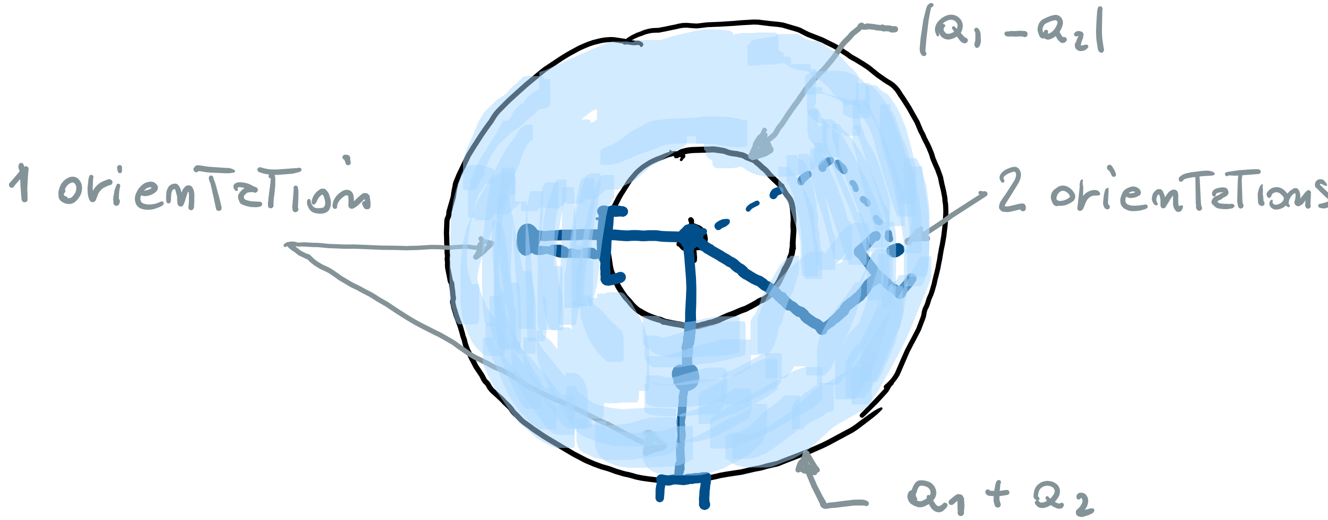
$\Rightarrow \exists$ solution for any orientation

Workspace of Fanuc robot



WORKSPACE OF A PLANAR RR-ARM





- CASE $a_1 \neq a_2$

$$WS_1 = \{ P \in \mathbb{R}^2 : |a_1 - a_2| \le \|P\| \le a_1 + a_2 \}$$

$WS_2 = \emptyset$ empty

- CASE $a_1 = a_2 = a$

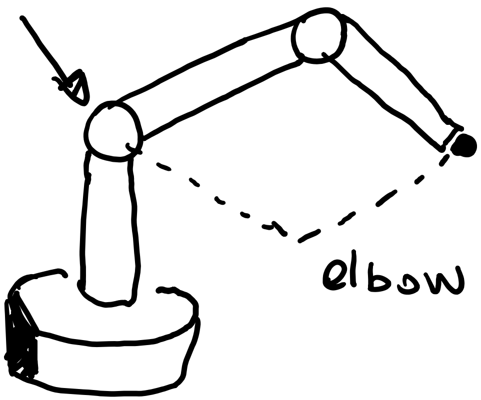
$$WS_1 = \{ P \in R : \|P\| \le 2a \}$$

$WS_2 = \{ P = 0 \}$ at origin all orientations in plane are possible!

IK SOLUTIONS FOR ANTROPOMORPHIC MANIPULATOR

Task : position EE (no orientation) 3 R - robot

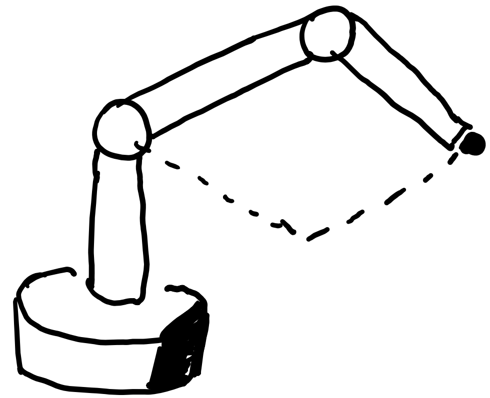
offset elbow up



elbow down

left

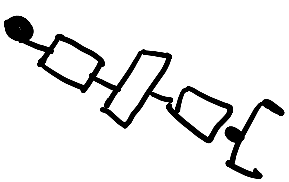
$\boxed{\text{4 inverse solutions}}$



right

considering also the wrist ... 6 R - robot

Task : position EE + orientation

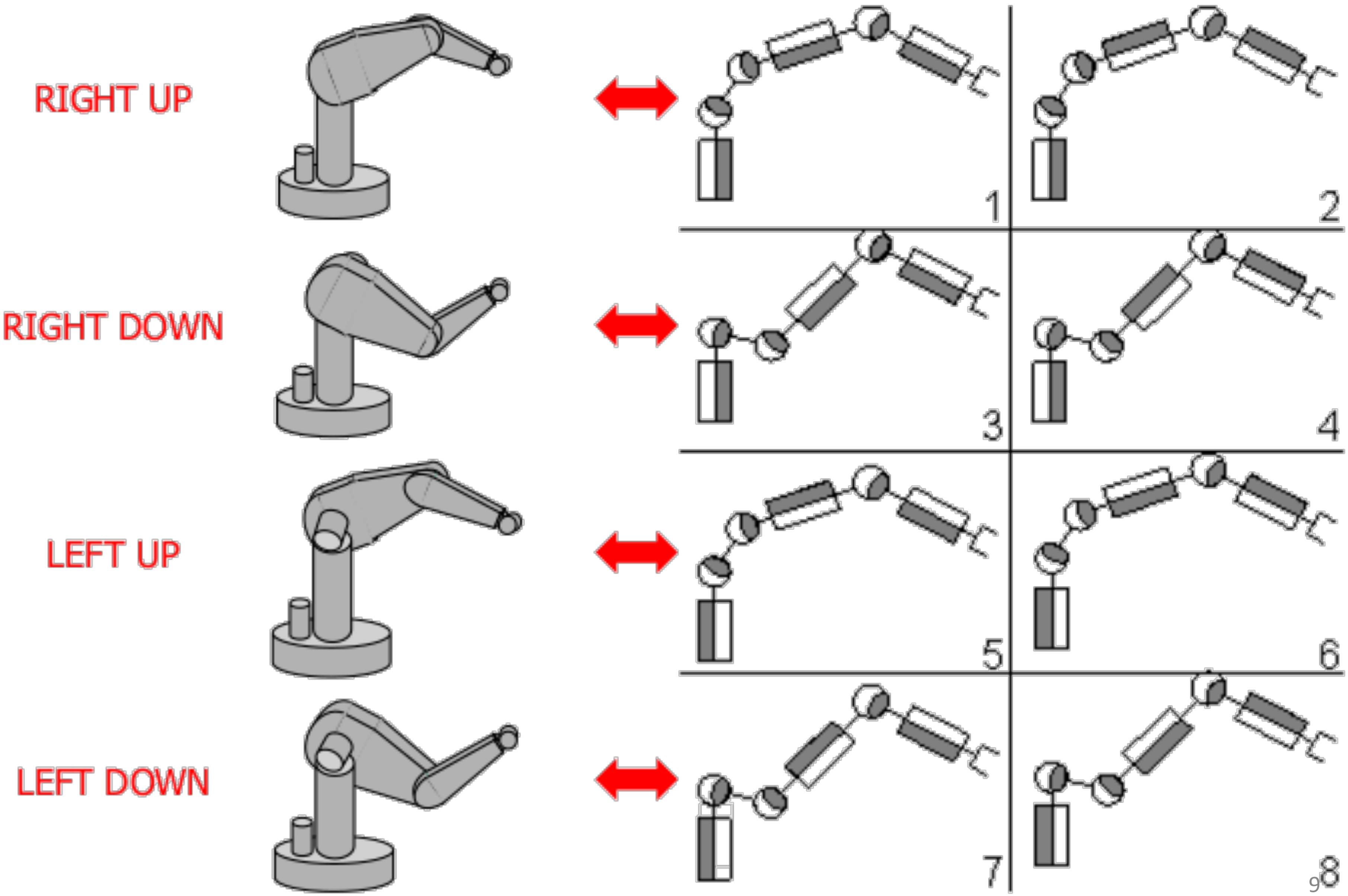


q_4, q_5, q_6 euler angles

→ 2 solutions for euler angles

→ 8 inverse solutions

VISUALIZING THE 8 SOLUTIONS



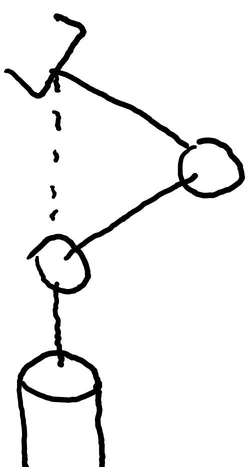
Q2: MULTIPLE SOLUTIONS ?

E-E positioning of Planar 2R arm

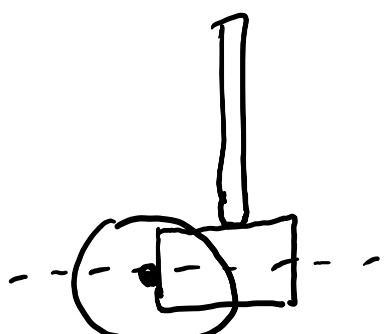
- 2 regular solutions in WS_1
- 1 solution on ∂WS_1 $\Rightarrow$ avoid complete arm extension

E-E positioning of 3R arm

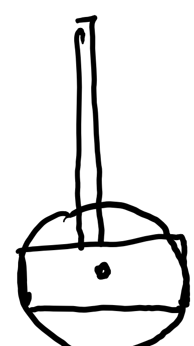
- 4 regular solutions
- 2 solutions on ∂WS_1



OFFset



no offet



∞ solutions

Q2: MULTIPLE SOLUTIONS?

n = # DOFs

m = Task dimension

- let's not consider singular solutions only regular:

① $n \le m$: we can have multiple (discrete) solutions

② $n > m$: robot is redundant for the kinematic Task

∞^{n-m} solutions

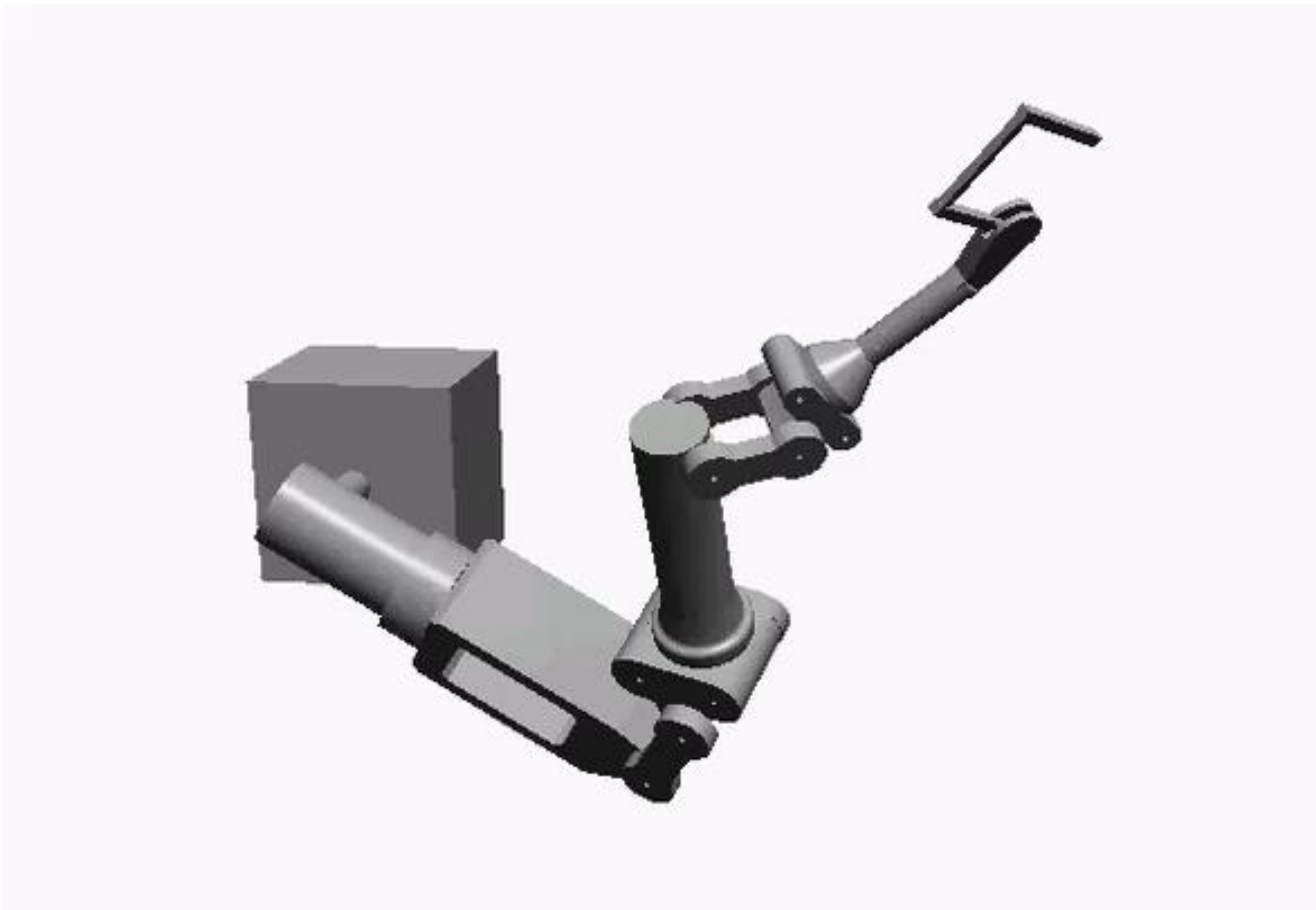
Dexter arm 8R

$m = 6$

$n = 8$

$\Rightarrow \infty^{8-6} = \infty^2$ IK solutions

$\hookrightarrow$ redundancy degree

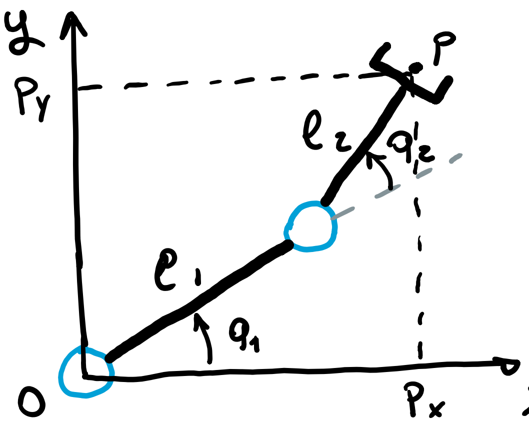


Q3: can I write solutions in closed form?

- only in some special cases
- require geometric intuition
- require an expression of the direct kinematics

INVERSE KINEMATICS FOR PLANAR 2R-ARM

- closed form method based on geometric intuition



- first compute forward kinematics analitically:

$$p_x = \ell_1 c_1 + \ell_2 c_{12}$$

$$p_y = \ell_1 s_1 + \ell_2 s_{12}$$

↑ input data $\qquad q_1, q_2$ unknowns

1 intuition: The distance between P and O depends only on q_2

$$p_x^2 + p_y^2 = \ell_1^2 c_1^2 + \ell_2^2 c_{12}^2 + \ell_1^2 s_1^2 + \ell_2^2 s_{12}^2 + 2\,\ell_1 \ell_2 \underbrace{(c_1 c_{12} + s_1 s_{12})}_{\cos(q_1+q_2-q_1) = c_2}$$

$\ell_1^2 \qquad \ell_2^2$

isolate c_2:

$$c_2 = \frac{p_x^2 + p_y^2 - \ell_1^2 - \ell_2^2}{2\,\ell_1 \ell_2} \qquad s_2 = \pm\sqrt{1 - c_2^2}$$

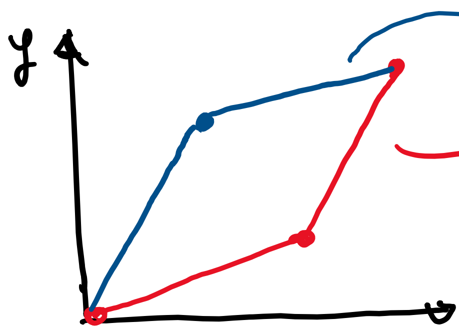
$+$ elbow down

$-$ elbow up

$$\Rightarrow \boxed{q_2 = \operatorname{atan2}(s_2, c_2)}$$

(arccos is defined only in 2 quadrants)

Elbow up and elbow down solutions



elbow up solution ($\sin q_2 < 0$)

elbow down solution ($\sin q_2 > 0$)

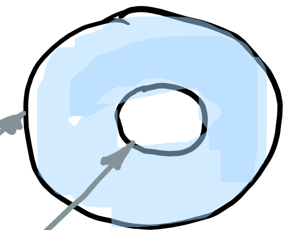
c_2 returns a value $\in [-1, 1]$

$$\Downarrow$$

$$-1 \leq \frac{P_x^2 + P_y^2 - \ell_1^2 - \ell_2^2}{2\ell_1\ell_2} \leq 1$$

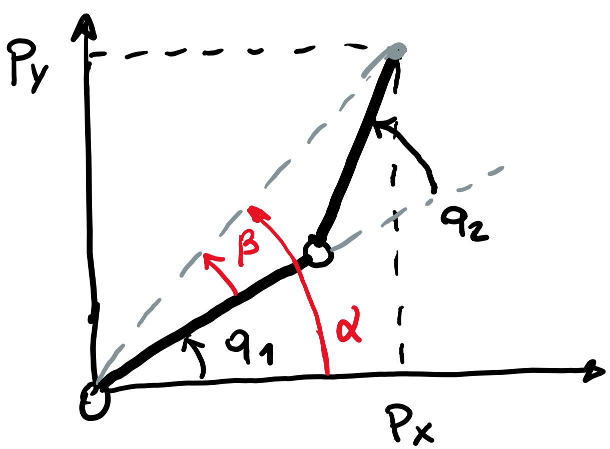
$$P_x^2 + P_y^2 \leq \ell_1^2 + \ell_2^2 + 2\ell_1\ell_2 = (\ell_1 + \ell_2)^2$$

$$(\ell_1 - \ell_2)^2 = \ell_1^2 + \ell_2^2 - 2\ell_1\ell_2 \leq P_x^2 + P_y^2$$



$\Rightarrow$ we get for free a check that the solution is inside WS1.

2° intuition (geometric solution for q_1)



$$q_1 = \alpha - \beta$$

$$q_1 = \underbrace{\mathrm{atan2}(P_y, P_x)}_{\alpha} - \underbrace{\mathrm{atan2}(\overbrace{\ell_2 s_2}^{\sin\beta}, \overbrace{\ell_1 + \ell_2 c_2}^{\cos\beta})}_{\beta}$$

NB difference of atan2 needs to be remapped in $(-\pi, \pi)$

Algebraic solution for q_1

$$\left.\begin{aligned} p_x &= \ell_1 c_1 + \ell_2 c_{12} = \ell_1 c_1 + \ell_2 (c_1 c_2 - s_1 s_2) \\ p_y &= \ell_1 s_1 + \ell_2 s_{12} = \ell_1 s_1 + \ell_2 (s_1 c_2 + c_1 s_2) \end{aligned}\right\} \text{linear system in } c_1, s_1$$

$$\underbrace{\begin{bmatrix} \ell_1 + \ell_2 c_2 & -\ell_2 s_2 \\ \ell_2 s_2 & \ell_1 + \ell_2 c_2 \end{bmatrix}}_{A} \underbrace{\begin{bmatrix} c_1 \\ s_1 \end{bmatrix}}_{x} = \underbrace{\begin{bmatrix} p_x \\ p_y \end{bmatrix}}_{b} \Rightarrow x = A^{-1} b$$

$$\det = \ell_1^2 + \ell_2^2 + 2 \ell_1 \ell_2 c_2 > 0$$

except when $\ell_1 = \ell_2$ and $q_2 = -\pi$

$$\boxed{q_1 = \operatorname{atan2}(s_1, c_1)}$$

∞ q_1 (singularity)

Summary of IK for 2R arm

We found in close analytic form for all cases:

1) we checked if we are inside the WS1 and then computed the 2 solutions for q_2 (regular case)

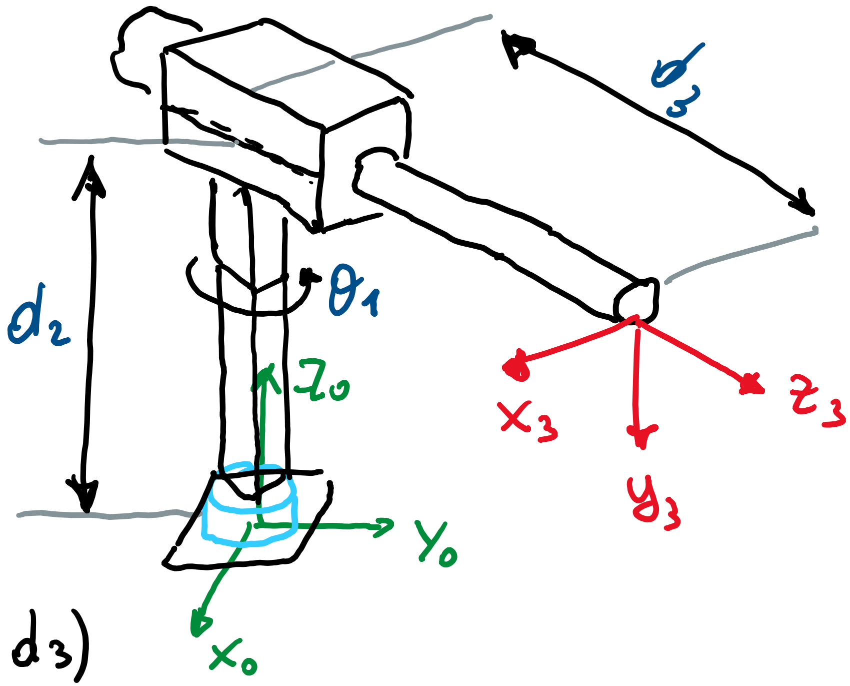
2) Elbow up/elbow down solutions collapse into 1 solution when the arm is straight

3) We computed the solution for q_1 checking the only case q_1 is not defined (singularity)

EXAMPLE : RPP

FWD KIN: find ${}_0T_3$

$$
{}_0T_3 = \begin{bmatrix} c_1 & 0 & -s_1 & -d_3 s_1 \\ s_1 & 0 & c_1 & d_3 c_1 \\ 0 & -1 & 0 & d_2 \\ 0 & 0 & 0 & 1 \end{bmatrix}
$$



INV. KIN: solve for (θ_1, d_2, d_3)

$$
\begin{bmatrix} c_1 & 0 & -s_1 & -d_3 s_1 \\ s_1 & 0 & c_1 & d_3 c_1 \\ 0 & -1 & 0 & d_2 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \underbrace{\begin{bmatrix} r_{11} & r_{12} & r_{13} & P_x \\ r_{21} & r_{22} & r_{23} & P_y \\ r_{31} & r_{23} & r_{33} & P_z \\ 0 & 0 & 0 & 1 \end{bmatrix}}_{\text{known}}
$$

$$\boxed{d_2 = P_z}$$

$$\boxed{\theta_1 = \operatorname{atan2}(r_{21}, r_{11})}$$

$$-P_x S_1 + P_y C_1 = S_1^2 d_3 + C_1^2 d_3 = d_3$$

$$\Rightarrow \boxed{d_3 = -P_x S_1 + P_y C_1}$$

SOLUTION METHODS

ANALYTICAL

- closed form
- requires intuition (geometric)
- need to handle cases of multiple solutions with "IF - else" constructs
- algebraic methods
- in specific cases (i.e. 3 consecutive // axes revolute joints or spherical wrist

$\Rightarrow$ systematic way to generate a reduced set of (non linear) equations

[Pieper Thesis, 1968]

NUMERICAL

- mandatory if $n > m$ (∞ solutions)
- slower but easier to setup
- use analytic Jacobian of direct kinematics

$$J_R(q) = \frac{\partial f(q)}{\partial q}$$

- Newton method / gradient method