Robotics 1

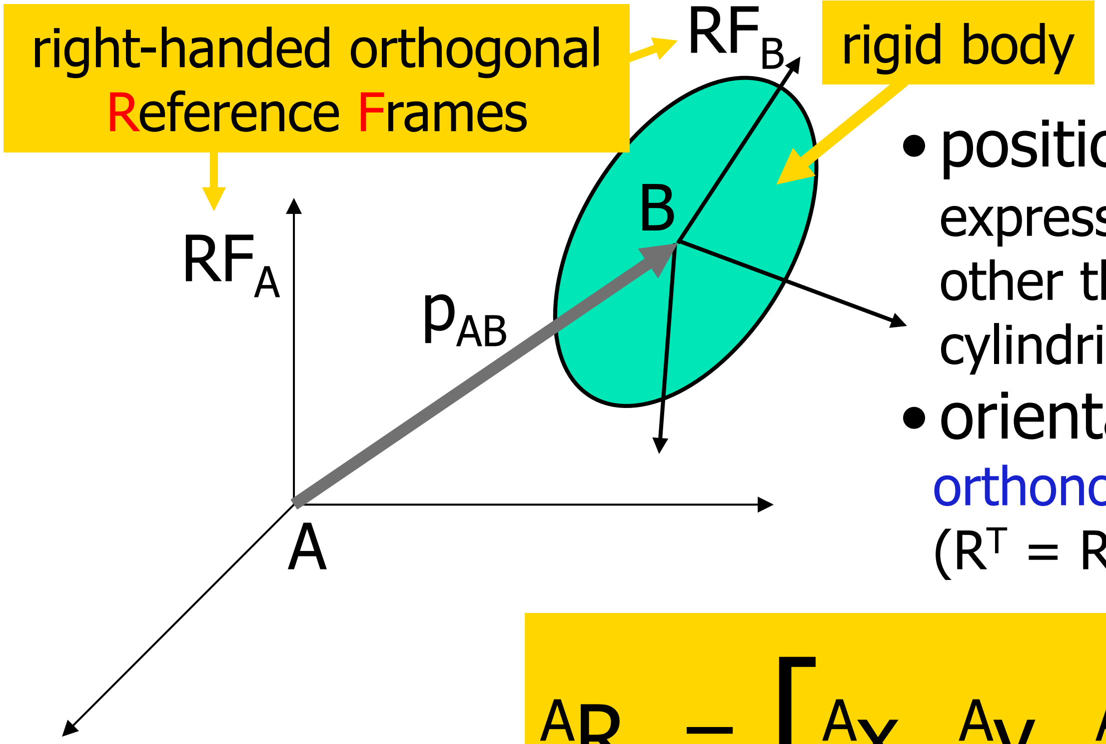
Position and orientation of rigid bodies

Prof. Alessandro De Luca

Dipartimento di Informatica e Sistemistica Antonio Ruberti

Sapienza Università di Roma

Position and orientation



- position: $^Ap_{AB}$ (vector $\in R^3$), expressed in RF_A (use of coordinates other than Cartesian is possible, e.g. cylindrical or spherical)
- orientation: orthonormal 3x3 matrix ($R^T = R^{-1} \Rightarrow$ $^AR_B\,^BR_A = I$), with det = +1

$$^AR_B = \left[\,^Ax_B\;^Ay_B\;^Az_B\right]$$

- $x_A\, y_A\, z_A$ ($x_B\, y_B\, z_B$) are unit vectors (with unitary norm) of frame RF_A (RF_B)
- components in ^AR_B are the direction cosines of the axes of RF_B with respect to (w.r.t.) RF_A

Rotation matrix

$$
{}^{A}R_{B} = \begin{bmatrix} x_A^T x_B & x_A^T y_B & x_A^T z_B \\ y_A^T x_B & y_A^T y_B & y_A^T z_B \\ z_A^T x_B & z_A^T y_B & z_A^T z_B \end{bmatrix}
$$

direction cosine of z_B w.r.t. x_A

orthonormal, with det = +1

algebraic structure of a group SO(3) (neutral element = I; inverse element = R^T)

chain rule property

$$
\boxed{{}^{k}R_{i} \cdot {}^{i}R_{j} = {}^{k}R_{j}}
$$

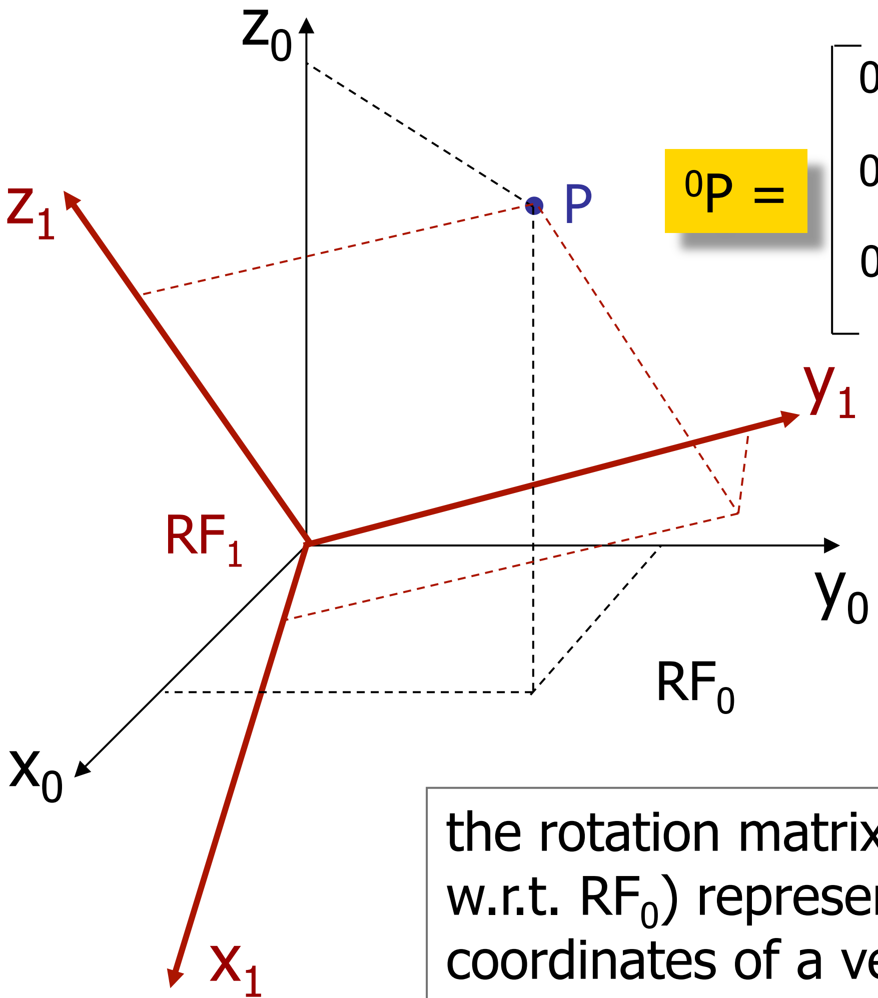
orientation of RF_i w.r.t. RF_k

orientation of RF_j w.r.t. RF_i

orientation of RF_j w.r.t. RF_k

NOTE: in general, the product of rotation matrices does not commute!

Change of coordinates



$$
{}^0P = \begin{bmatrix} {}^0p_x \\ {}^0p_y \\ {}^0p_z \end{bmatrix} = {}^1p_x\,{}^0\mathbf{x}_1 + {}^1p_y\,{}^0\mathbf{y}_1 + {}^1p_z\,{}^0\mathbf{z}_1
$$

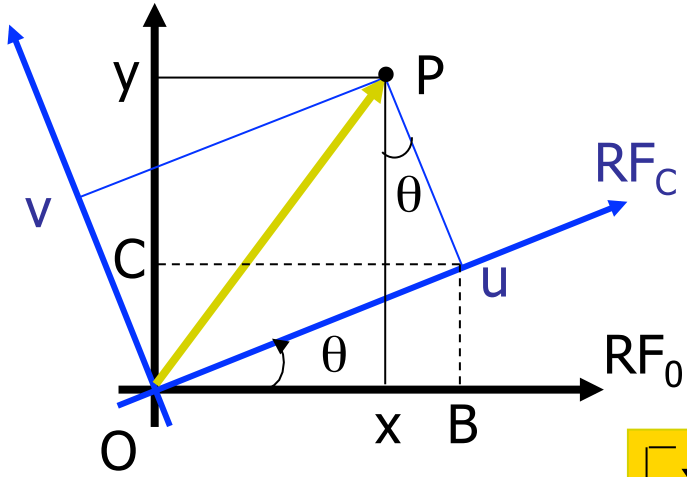
$$
= \begin{bmatrix} {}^0\mathbf{x}_1 & {}^0\mathbf{y}_1 & {}^0\mathbf{z}_1 \end{bmatrix} \begin{bmatrix} {}^1p_x \\ {}^1p_y \\ {}^1p_z \end{bmatrix}
$$

$$
= {}^0R_1\,{}^1P
$$

the rotation matrix ${}^0R_1$ (i.e., the orientation of RF_1 w.r.t. RF_0) represents also the change of coordinates of a vector from RF_1 to RF_0

Ex: Orientation of frames in a plane

(elementary rotation around z-axis)



$$x = OB - xB = u \cos\theta - v \sin\theta$$
$$y = OC + Cy = u \sin\theta + v \cos\theta$$
$$z = w$$

or...

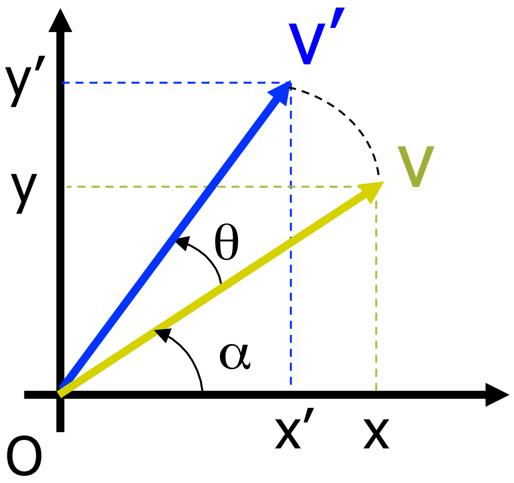
$$
\begin{array}{c}
\\
{}^0OP \rightarrow
\end{array}
\begin{bmatrix} x \\ y \\ z \end{bmatrix}
=
\begin{array}{c}
\begin{array}{ccc} {}^0x_C & {}^0y_C & {}^0z_C \end{array} \\
\begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}
\end{array}
\begin{bmatrix} u \\ v \\ w \end{bmatrix}
= R_z(\theta)
\begin{array}{c}
{}^C OP \\
\begin{bmatrix} u \\ v \\ w \end{bmatrix}
\end{array}
$$

$$\boxed{R_z(-\theta) = R_z^T(\theta)}$$

similarly:

$$R_x(\theta) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta \\ 0 & \sin\theta & \cos\theta \end{bmatrix} \qquad R_y(\theta) = \begin{bmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{bmatrix}$$

Ex: Rotation of a vector around z



$$x = |v| \cos \alpha$$
$$y = |v| \sin \alpha$$

$$x' = |v| \cos(\alpha + \theta) = |v| (\cos \alpha \cos \theta - \sin \alpha \sin \theta)$$
$$= x \cos \theta - y \sin \theta$$

$$y' = |v| \sin(\alpha + \theta) = |v| (\sin \alpha \cos \theta + \cos \alpha \sin \theta)$$
$$= x \sin \theta + y \cos \theta$$

$$z' = z$$

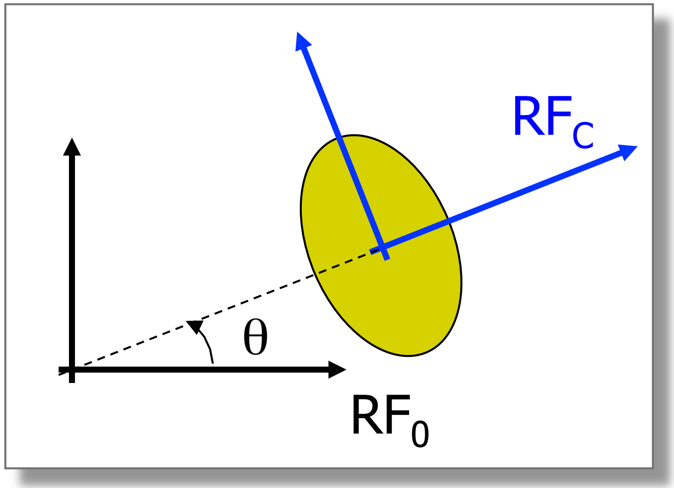
or...

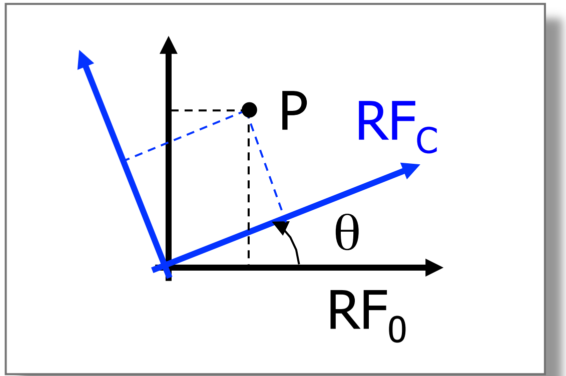
$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = R_z(\theta) \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

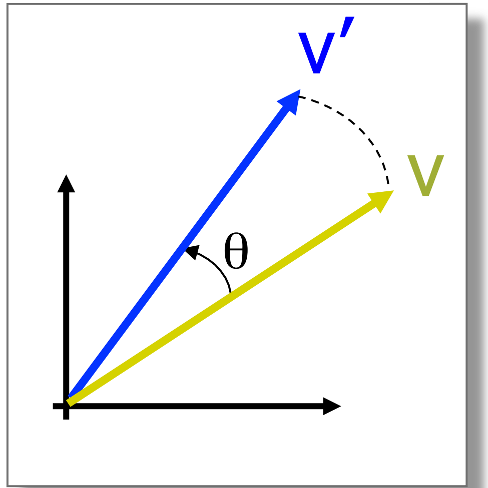
...as before!

Equivalent interpretations of a rotation matrix

the same rotation matrix, e.g., $R_z(\theta)$, may represent:







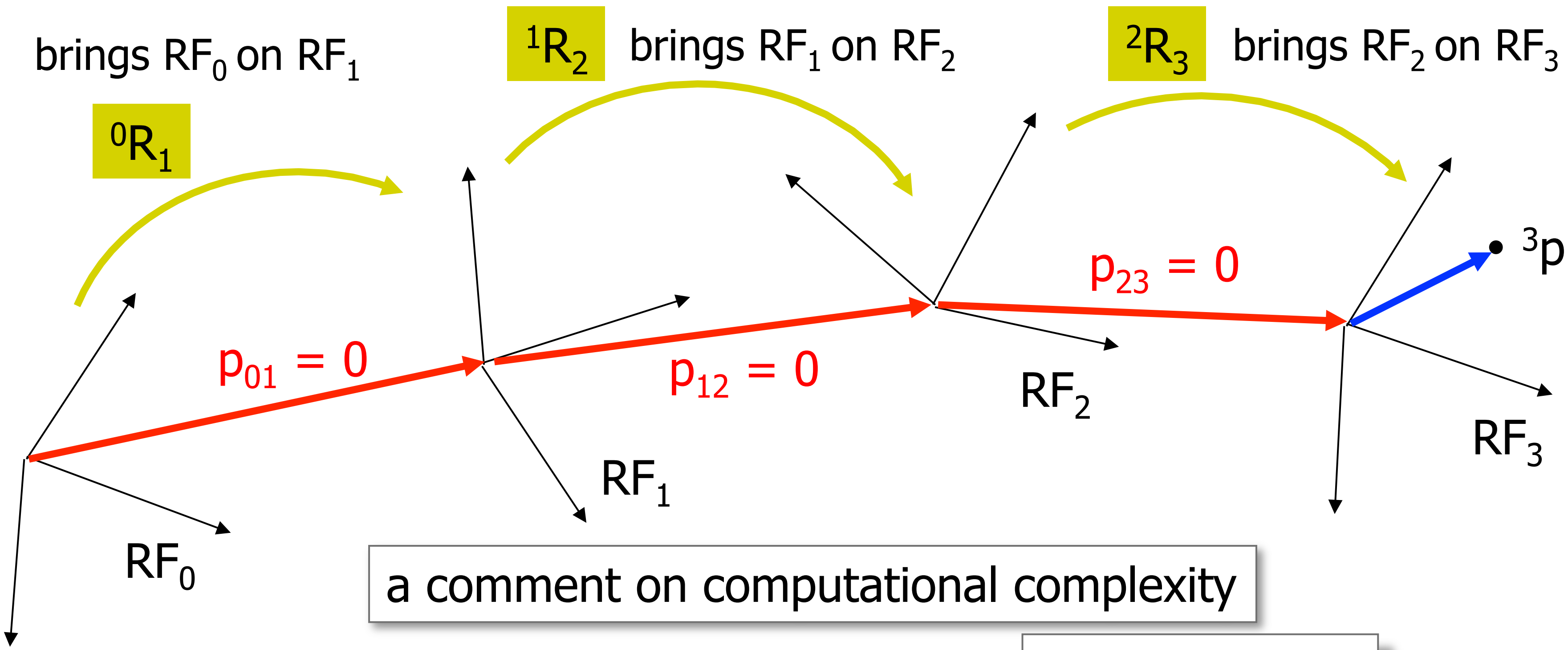
the orientation of a rigid body with respect to a reference frame RF_0
ex: $[{}^0x_c \; {}^0y_c \; {}^0z_c] = R_z(\theta)$

the change of coordinates from RF_C to RF_0
ex: ${}^0P = R_z(\theta)\, {}^CP$

the vector rotation operator
ex: $v' = R_z(\theta)\, v$

the rotation matrix ${}^0R_C$ is an operator superposing frame RF_0 to frame RF_C

Composition of rotations



a comment on computational complexity

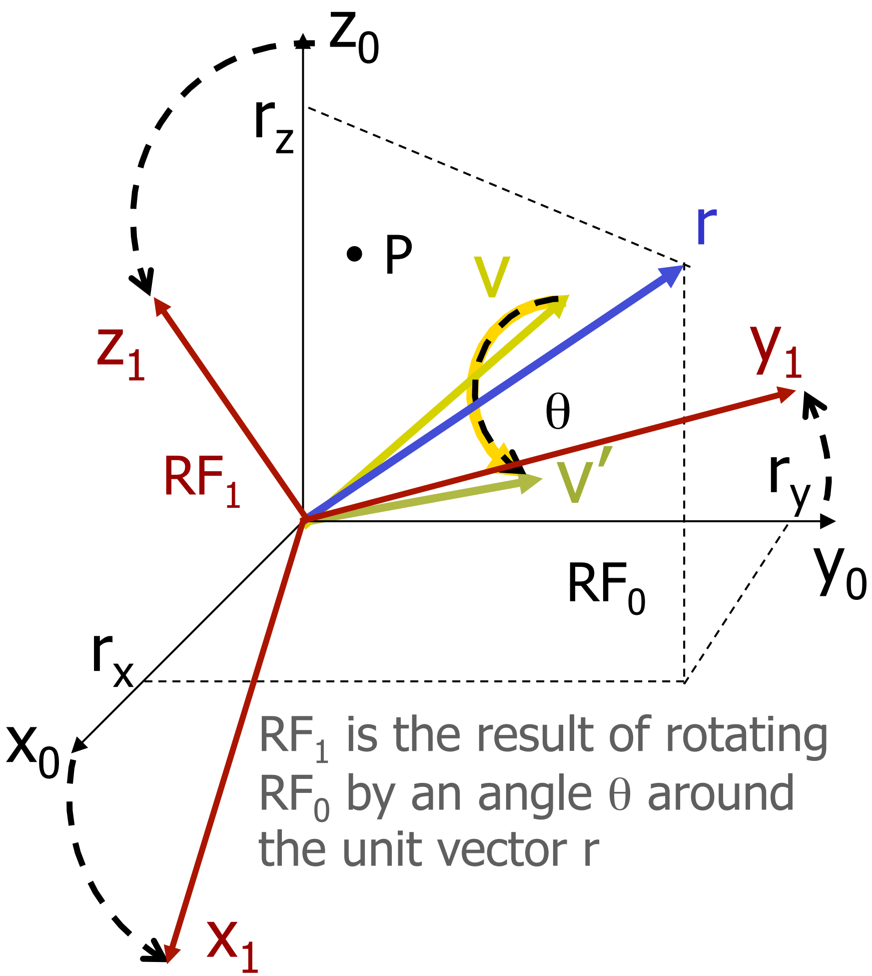
$$ {}^0p = ({}^0R_1\,{}^1R_2\,{}^2R_3)\,{}^3p = {}^0R_3\,{}^3p $$

63 products
42 summations

$$ {}^0p = {}^0R_1\,\underbrace{({}^1R_2\,\underbrace{({}^2R_3\,{}^3p)}_{{}^2p})}_{{}^1p} $$

27 products
18 summations

Axis/angle representation



RF_1 is the result of rotating RF_0 by an angle θ around the unit vector r

DATA

- unit vector r ($\|r\| = 1$)
- θ (positive if counterclockwise, as seen from an "observer" placed like r)

DIRECT PROBLEM

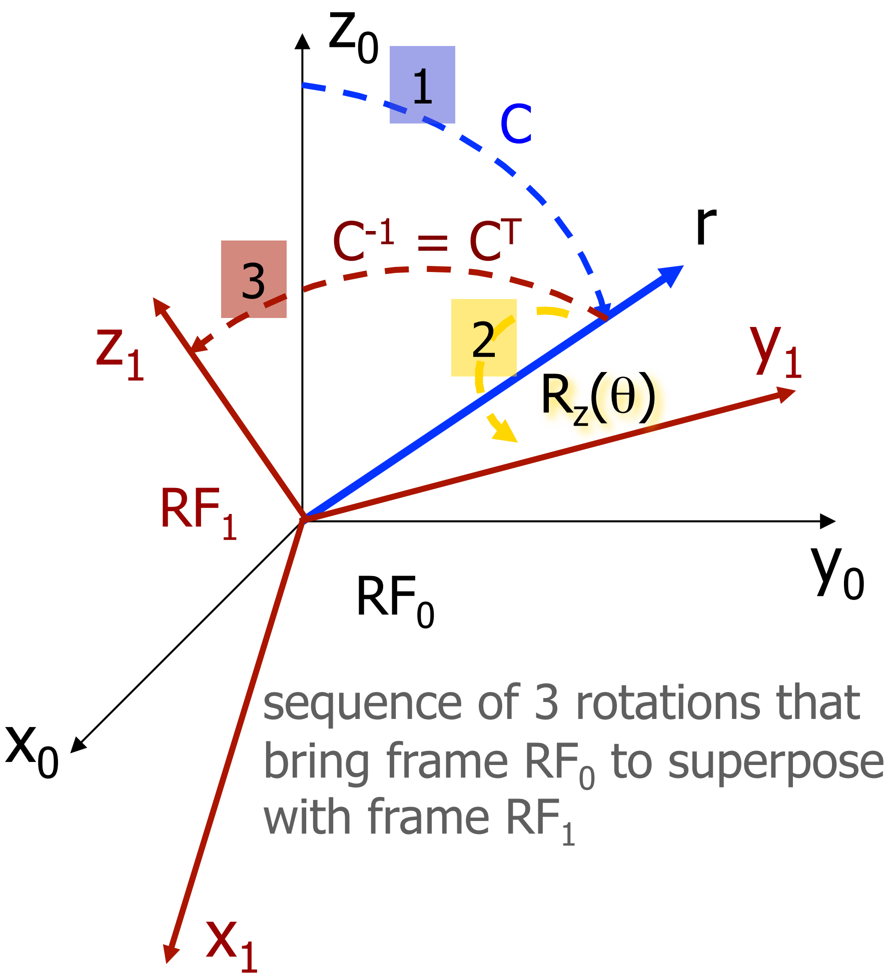
find

$$R(\theta,r) = [{}^0x_1 \; {}^0y_1 \; {}^0z_1]$$

such that

$${}^0P = R(\theta,r)\,{}^1P \qquad {}^0v' = R(\theta,r)\,{}^0v$$

Axis/angle: Direct problem



sequence of 3 rotations that bring frame RF_0 to superpose with frame RF_1

$$R(\theta,r) = C\, R_z(\theta)\, C^T$$

concatenation of three rotations

$$C = \begin{bmatrix} n & s & r \end{bmatrix}$$

after the first rotation the z-axis coincides with r

n and s are orthogonal unit vectors such that $n \times s = r$, or

$$n_y s_z - s_y n_z = r_x$$

$$n_z s_x - s_z n_x = r_y$$

$$n_x s_y - s_x n_y = r_z$$

Axis/angle: Direct problem
solution

$$\boxed{R(\theta,r) = C\,R_z(\theta)\,C^T}$$

$$R(\theta,r) = \begin{bmatrix} n & s & r \end{bmatrix} \begin{bmatrix} c\theta & -s\theta & 0 \\ s\theta & c\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} n^T \\ s^T \\ r^T \end{bmatrix}$$

$$= r\,r^T + (n\,n^T + s\,s^T)\,c\theta + (s\,n^T - n\,s^T)\,s\theta$$

taking into account that

$C\,C^T = n\,n^T + s\,s^T + r\,r^T = I$, and that

$$s\,n^T - n\,s^T = \begin{bmatrix} 0 & -r_z & r_y \\ & 0 & -r_x \\ \text{skew-symm} & & 0 \end{bmatrix} = S(r)$$

skew-symmetric(r):

$r \times v = S(r)v = -S(v)r$

depends only on r and θ !!

$$\boxed{R(\theta,r) = r\,r^T + (I - r\,r^T)\,c\theta + S(r)\,s\theta} = \boxed{R^T(-\theta,r) = R(-\theta,-r)}$$

Rodriguez formula

$$v' = R(\theta,r)\, v$$

$$v' = v \cos\theta + (r \times v) \sin\theta + (1 - \cos\theta)(r^T v)\, r$$

proof:

$$R(\theta,r)\, v = (r\, r^T + (I - r\, r^T) \cos\theta + S(r) \sin\theta) v$$

$$= r\, r^T v\, (1 - \cos\theta) + v \cos\theta + (r \times v) \sin\theta$$

q.e.d.

Unit quaternion

- to eliminate undetermined and singular cases arising in the axis/angle representation, one can use the *unit quaternion* representation

$$\mathcal{Q} = \{\underset{\text{a scalar}}{\eta},\ \underset{\text{3-dim vector}}{\boldsymbol{\varepsilon}}\} = \{\cos(\theta/2),\ \sin(\theta/2)\,\mathbf{r}\}$$

- $\eta^2 + \|\boldsymbol{\varepsilon}\|^2 = 1$ (thus, "unit ...")
- $(\theta, \mathbf{r})$ and $(-\theta, -\mathbf{r})$ gives the same quaternion $\mathcal{Q}$
- the absence of rotation is associated to $\mathcal{Q} = \{1, \mathbf{0}\}$
- unit quaternions can be composed with special rules (in a similar way as in the product of rotation matrices)

$$\mathcal{Q}_1 * \mathcal{Q}_2 = \{\eta_1\eta_2 - \boldsymbol{\varepsilon}_1^T\boldsymbol{\varepsilon}_2,\ \eta_1\boldsymbol{\varepsilon}_2 + \eta_2\boldsymbol{\varepsilon}_1 + \boldsymbol{\varepsilon}_1 \times \boldsymbol{\varepsilon}_2\}$$

Robotics 1

Minimal representations of orientation
(Euler and roll-pitch-yaw angles)
Homogeneous transformations

Prof. Alessandro De Luca

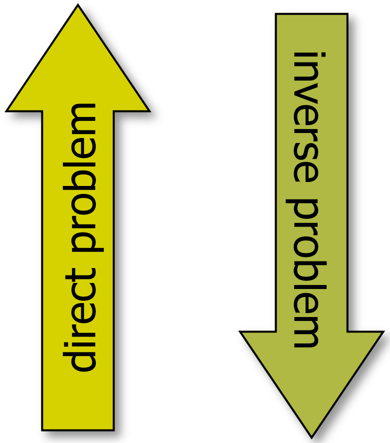
Dipartimento di Informatica e Sistemistica Antonio Ruberti

Sapienza Università di Roma

“Minimal” representations

- rotation matrices:

	9 elements
-	3 orthogonality relationships
-	3 unitary relationships
=	3 independent variables

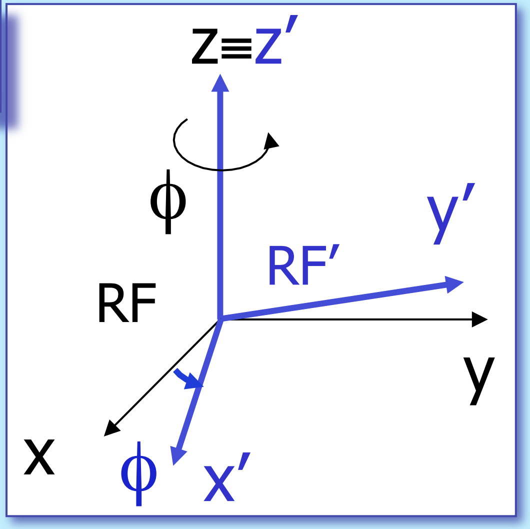


- sequence of 3 rotations around independent axes
  - fixed (a_i) or moving/current (a'_i) axes
  - 12 + 12 possible different sequences (e.g., XYX)
  - actually, only 12 since

$$\{(a_1\ \alpha_1), (a_2\ \alpha_2), (a_3\ \alpha_3)\} \equiv \{\ (a'_3\ \alpha_3)\ , (a'_2\ \alpha_2), (a'_1\ \alpha_1)\}$$

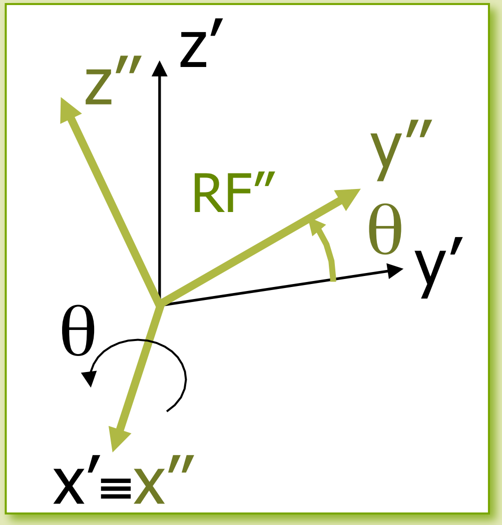
ZX′Z″ Euler angles

1



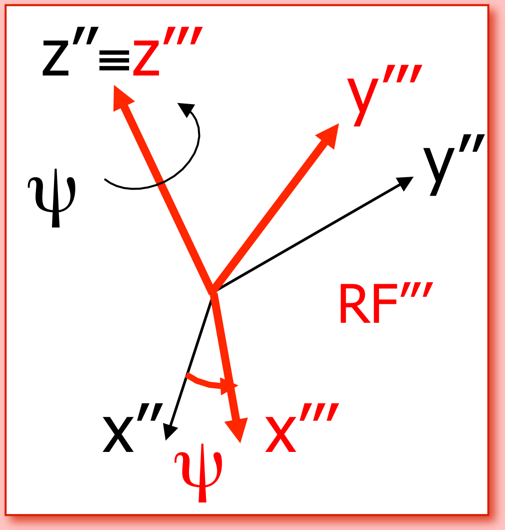
$$R_z(\phi) = \begin{bmatrix} \cos\phi & -\sin\phi & 0 \\ \sin\phi & \cos\phi & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

2



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

3



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

ZX'Z'' Euler angles

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

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

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

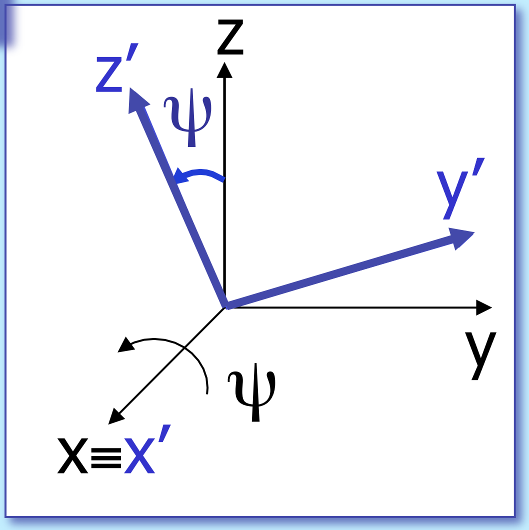
$$v = R_{ZX'Z''}(\phi,\ \theta,\ \psi)\ v'''$$

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

$$\boxed{\psi \text{ around z, } \theta \text{ around x (fixed), } \phi \text{ around z (fixed)}}$$

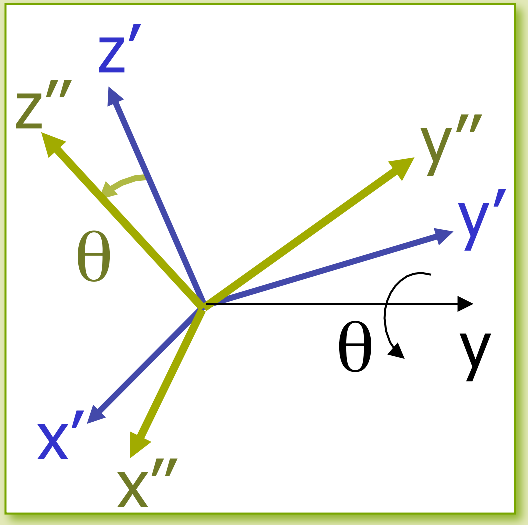
Roll-Pitch-Yaw angles

1 **ROLL**



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

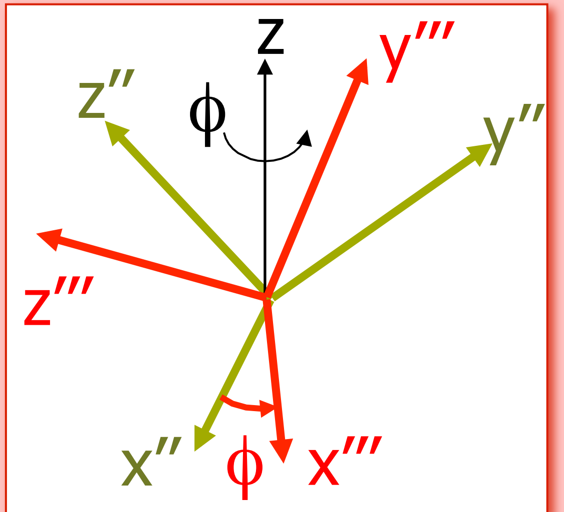
PITCH **2**



$C_1 R_Y(\theta) C_1^T$

with $R_Y(\theta) = \begin{bmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{bmatrix}$

3 **YAW**



$C_2 R_Z(\phi) C_2^T$

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

Roll-Pitch-Yaw angles (fixed XYZ)

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

$R_{RPY}(\psi, \theta, \phi) = R_Z(\phi)\, R_Y(\theta)\, R_X(\psi)$ ⇐ 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
- $r_{32}^2 + r_{33}^2 = c^2\theta$, $r_{31} = -s\theta$ $\Rightarrow$ $\boxed{\theta = \text{ATAN2}\{-r_{31}, \pm\sqrt{r_{32}^2 + r_{33}^2}\}}$

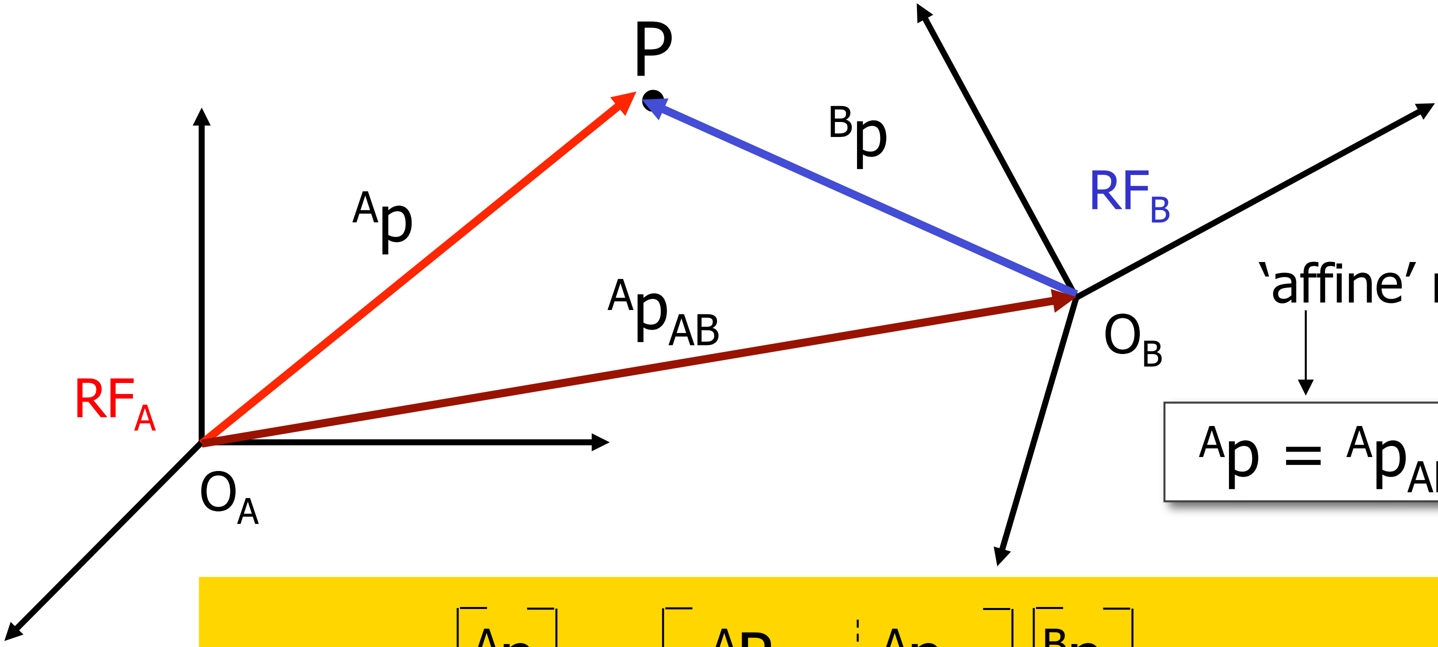
two symmetric values w.r.t. $\pi/2$

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

$r_{32}/c\theta = s\psi$, $r_{33}/c\theta = c\psi$ $\Rightarrow$ $\boxed{\psi = \text{ATAN2}\{r_{32}/c\theta, r_{33}/c\theta\}}$

- similarly... $\boxed{\phi = \text{ATAN2}\{r_{21}/c\theta, r_{11}/c\theta\}}$
- singularities for $\theta = \pm\pi/2$

Homogeneous transformations



'affine' relationship

$$\boxed{{}^Ap = {}^Ap_{AB} + {}^AR_B\, {}^Bp}$$

$${}^Ap_{hom} = \begin{bmatrix} {}^Ap \\ \hdashline 1 \end{bmatrix} = \left[\begin{array}{c:c} {}^AR_B & {}^Ap_{AB} \\ \hdashline 0 \;\; 0 \;\; 0 & 1 \end{array}\right] \begin{bmatrix} {}^Bp \\ \hdashline 1 \end{bmatrix} = {}^AT_B\, {}^Bp_{hom}$$

linear relationship

vector in homogeneous coordinates

4x4 matrix of homogeneous transformation

Properties of T matrix

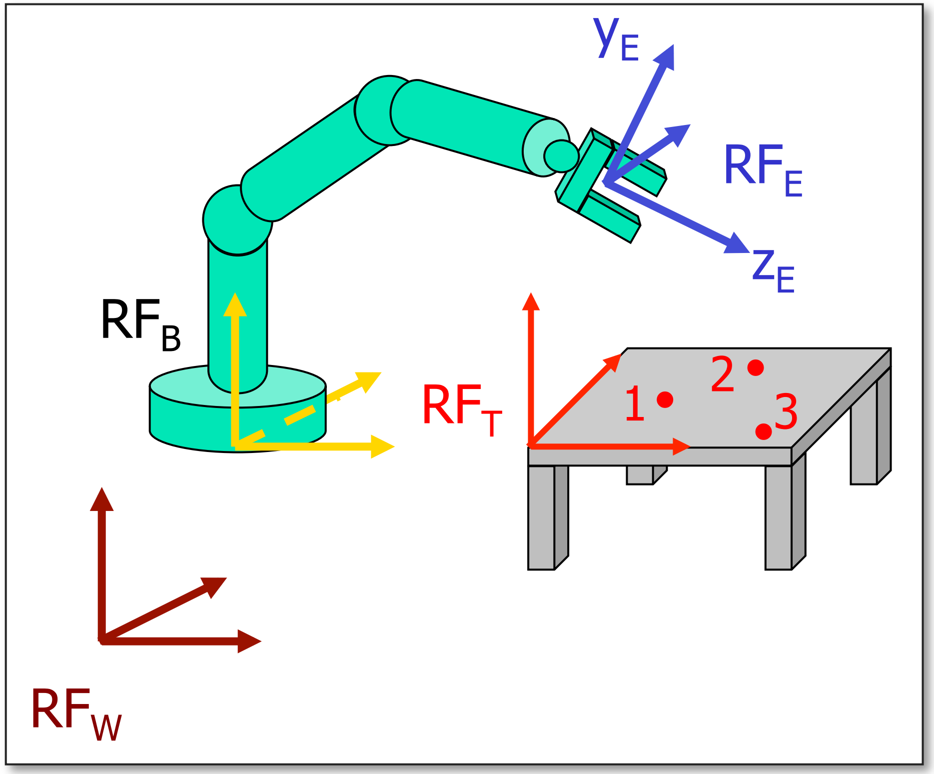
- describes the relation between reference frames (relative pose = position & orientation)
- transforms the representation of a position vector (applied vector from the origin of the frame) from a given frame to another frame
- it is a roto-translation operator on vectors in the three-dimensional space
- it is always invertible $({}^{A}T_{B})^{-1} = {}^{B}T_{A}$
- can be composed, i.e., ${}^{A}T_{C} = {}^{A}T_{B}\,{}^{B}T_{C}$ $\leftarrow$ note: it does not commute!

Inverse of a homogeneous transformation

$$ {}^{A}p = {}^{A}p_{AB} + {}^{A}R_{B}\,{}^{B}p \qquad {}^{B}p = {}^{B}p_{BA} + {}^{B}R_{A}\,{}^{A}p = -\,{}^{A}R_{B}^{T}\,{}^{A}p_{AB} + {}^{A}R_{B}^{T}\,{}^{A}p $$

$$ {}^{A}T_{B} = \left[\begin{array}{ccc|c} & {}^{A}R_{B} & & {}^{A}p_{AB} \\ \hline 0 & 0 & 0 & 1 \end{array}\right] \qquad {}^{B}T_{A} = \left[\begin{array}{ccc|c} & {}^{B}R_{A} & & {}^{B}p_{BA} \\ \hline 0 & 0 & 0 & 1 \end{array}\right] = \left[\begin{array}{ccc|c} & {}^{A}R_{B}^{T} & & -\,{}^{A}R_{B}^{T}\,{}^{A}p_{AB} \\ \hline 0 & 0 & 0 & 1 \end{array}\right] = ({}^{A}T_{B})^{-1} $$

Defining a robot task



absolute definition of task

task definition relative to the robot end-effector

$$ {}^{W}T_{T} = {}^{W}T_{B}\, {}^{B}T_{E}\, {}^{E}T_{T} $$

known, once the robot is placed

direct kinematics of the robot arm (function of q)

$$ {}^{B}T_{E}(q) = {}^{W}T_{B}^{\ -1}\, {}^{W}T_{T}\, {}^{E}T_{T}^{\ -1} = \text{cost} $$

Final comments on T matrices

- they are the main tool for computing the direct kinematics of robot manipulators
- they are used in many application areas (in robotics and beyond)
  - in the positioning of a vision camera (matrix ${}^{b}T_{c}$ with the extrinsic parameters of the camera posture)
  - in computer graphics, for the real-time visualization of 3D solid objects when changing the observation point

$$
{}^{A}T_{B} = \left[\begin{array}{ccc|c} & {}^{A}R_{B} & & {}^{A}p_{AB} \\ \hline \alpha_x & \alpha_y & \alpha_z & \sigma \end{array}\right]
$$

all zero in robotics

coefficients of perspective deformation

scaling coefficient

always unitary in robotics

Robotics 1

Direct kinematics

Prof. Alessandro De Luca

Dipartimento di Ingegneria Informatica Automatica e Gestionale Antonio Ruberti

Sapienza Università di Roma

Kinematics of robot manipulators

- "study of geometric and time properties of the motion of robotic structures, without reference to the causes producing it"

- robot seen as

  "(open) kinematic chain of rigid bodies interconnected by (revolute or prismatic) joints"

Motivations

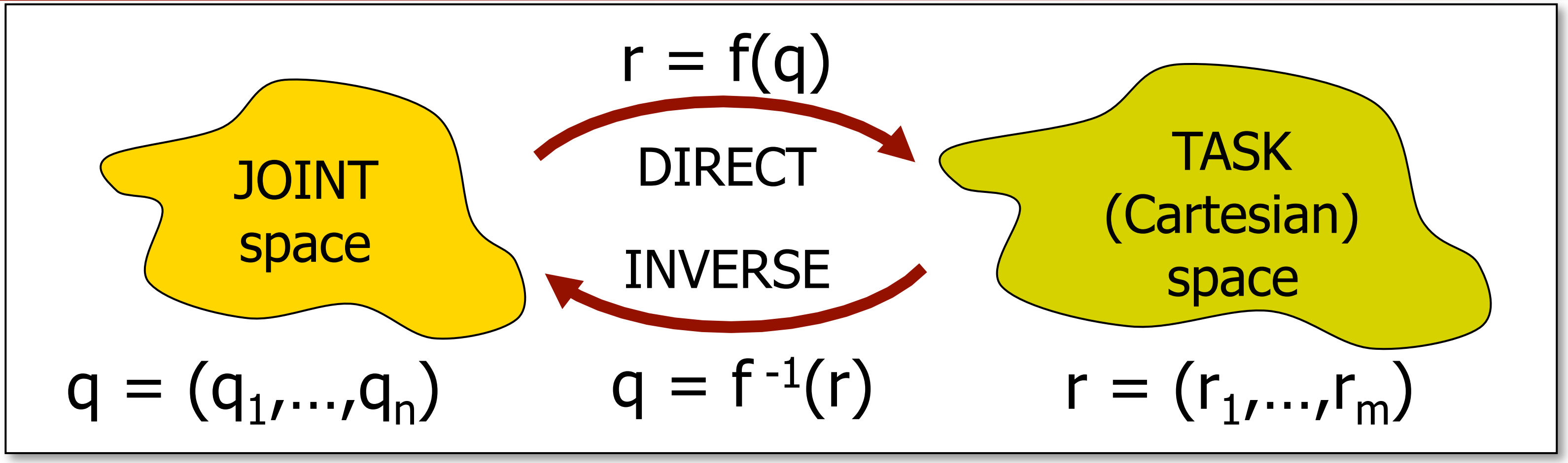
- functional aspects
  - definition of robot workspace
  - calibration
- operational aspects

task execution (actuation by motors) ⇄ task definition and performance

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

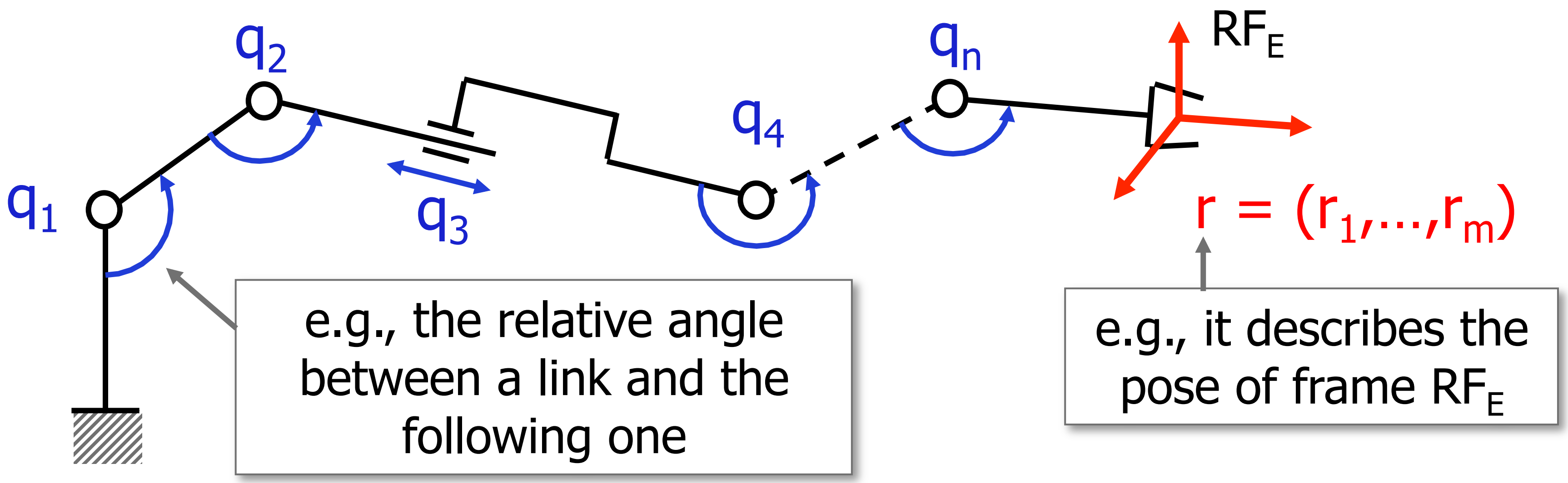
- trajectory planning
- programming
- motion control

Kinematics
formulation and parameterizations



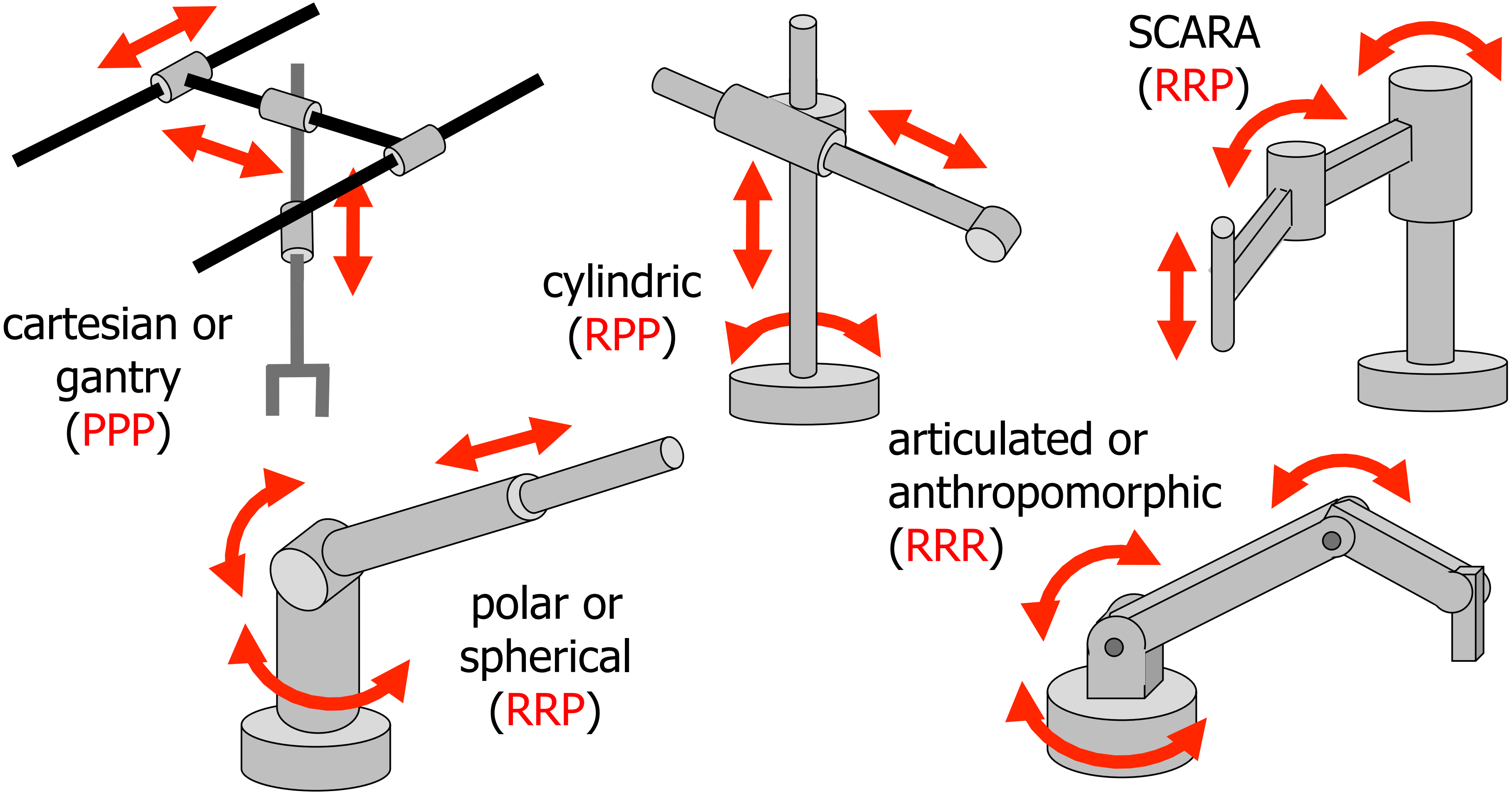
- choice of parameterization q
  - unambiguous and minimal characterization of the robot configuration
  - n = # degrees of freedom (dof) = # robot joints (rotational or translational)
- choice of parameterization r
  - compact description of positional and/or orientation (pose) components of interest to the required task
  - $m \le 6$, and usually $m \le n$ (but this is not strictly needed)

Open kinematic chains



- m = 2
  - pointing in space
  - positioning in the plane
- m = 3
  - orientation in space
  - positioning and orientation in the plane

Classification by kinematic type
(first 3 dofs)



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

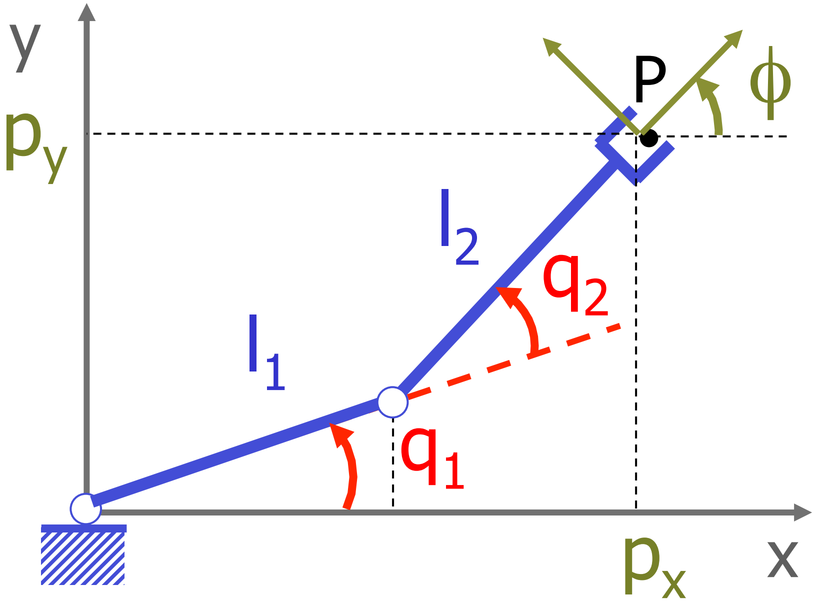
Direct kinematic map

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

$$r = f_r(q)$$

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

Example: direct kinematics of 2R arm



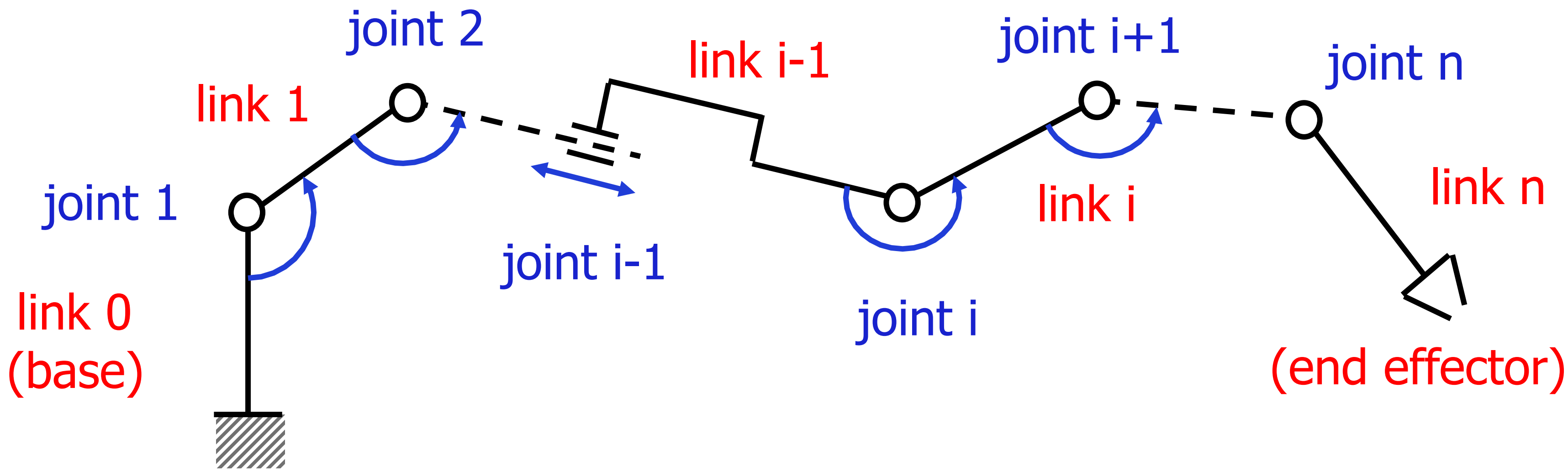
$$q = \begin{bmatrix} q_1 \\ q_2 \end{bmatrix} \qquad n = 2$$

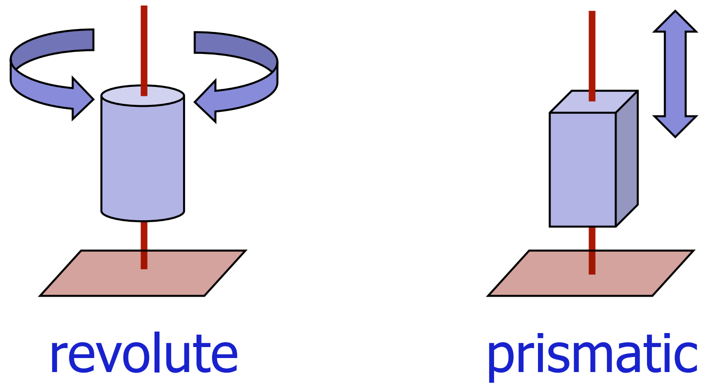
$$r = \begin{bmatrix} p_x \\ p_y \\ \phi \end{bmatrix} \qquad m = 3$$

$$p_x = l_1 \cos q_1 + l_2 \cos(q_1+q_2)$$
$$p_y = l_1 \sin q_1 + l_2 \sin(q_1+q_2)$$
$$\phi = q_1 + q_2$$

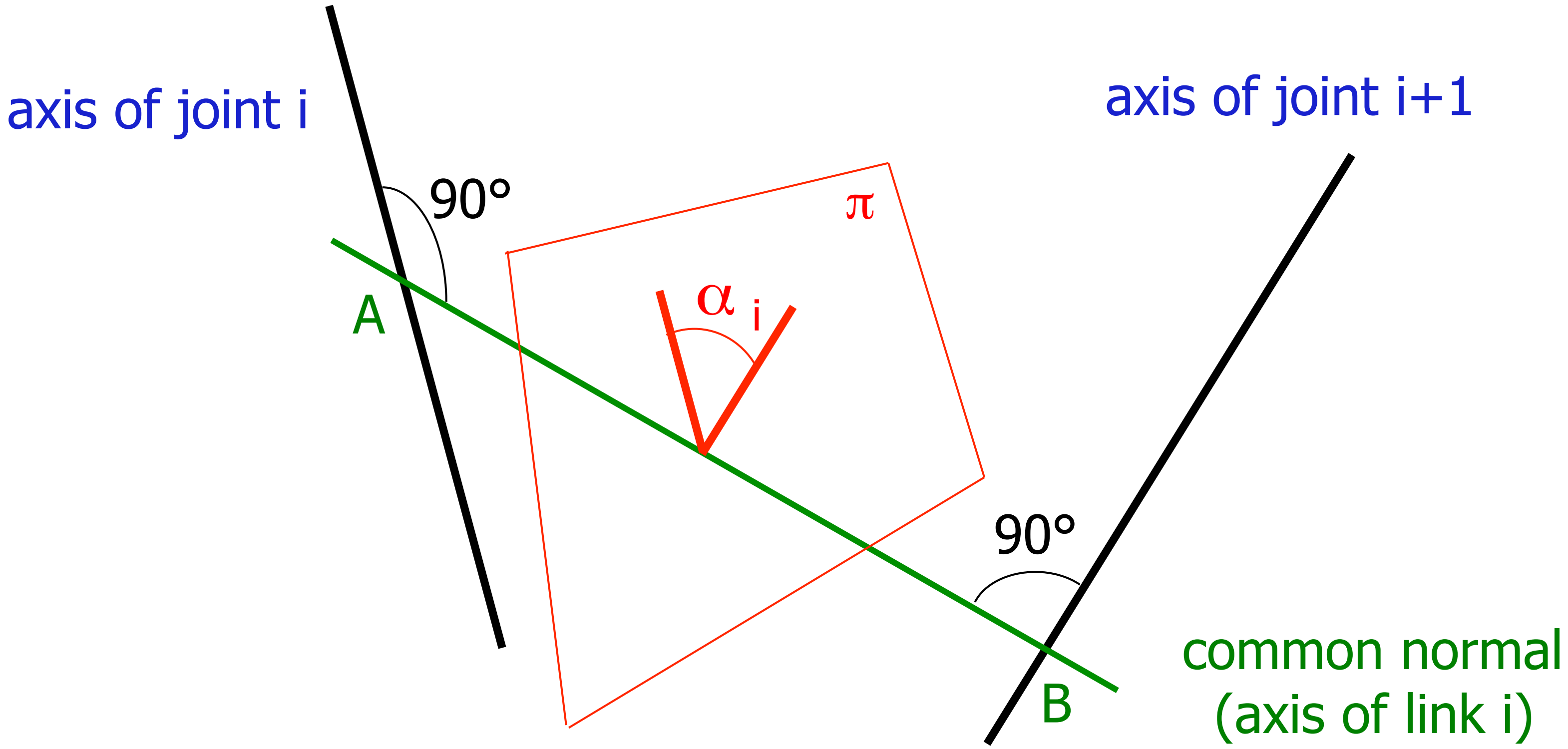
for more general cases we need a "method"!

Numbering links and joints





Relation between joint axes

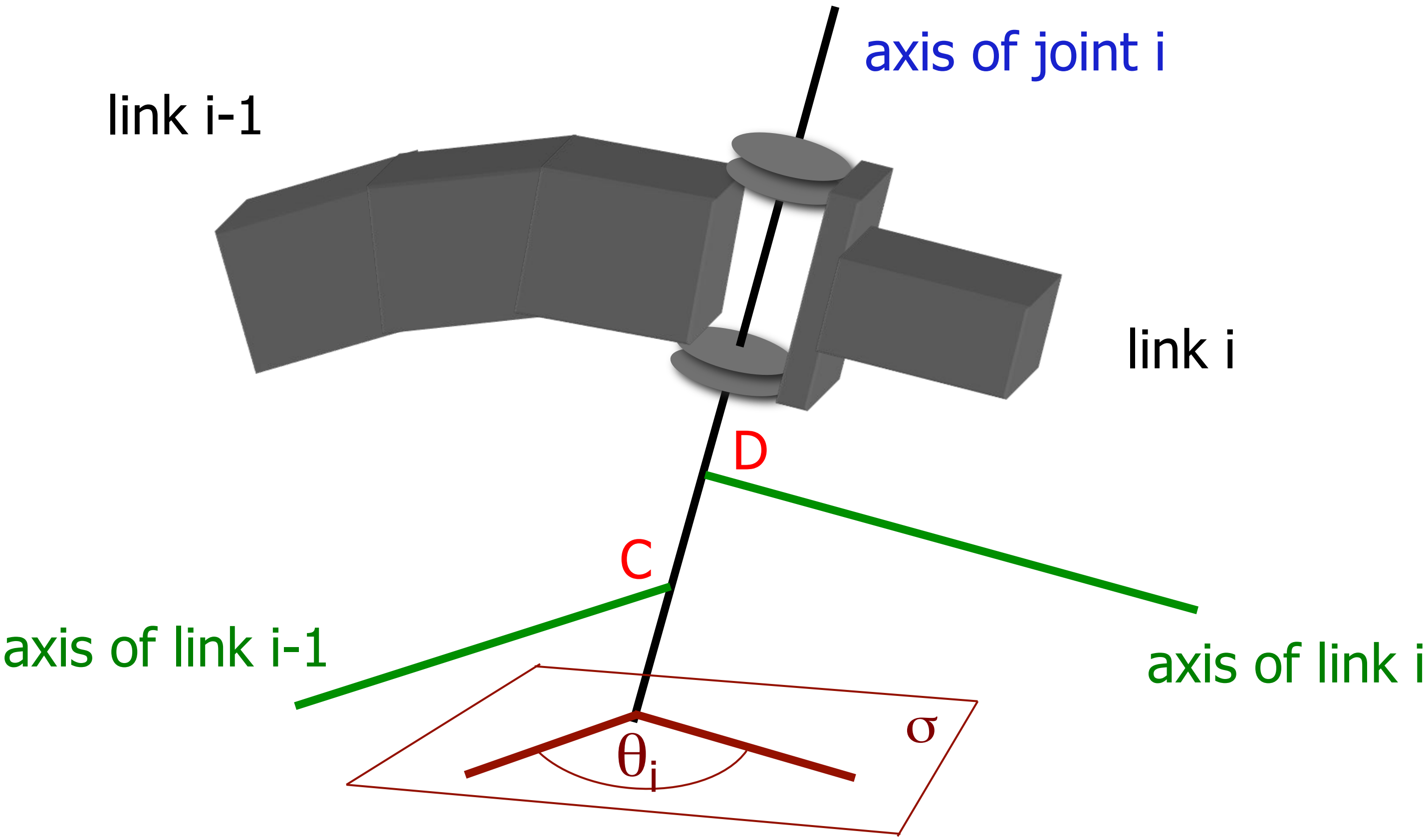


a_i = distance AB between joint axes (always well defined)

α_i = twist angle between joint axes
[projected on a plane π orthogonal to the link axis]

with sign (pos/neg)!

Relation between link axes

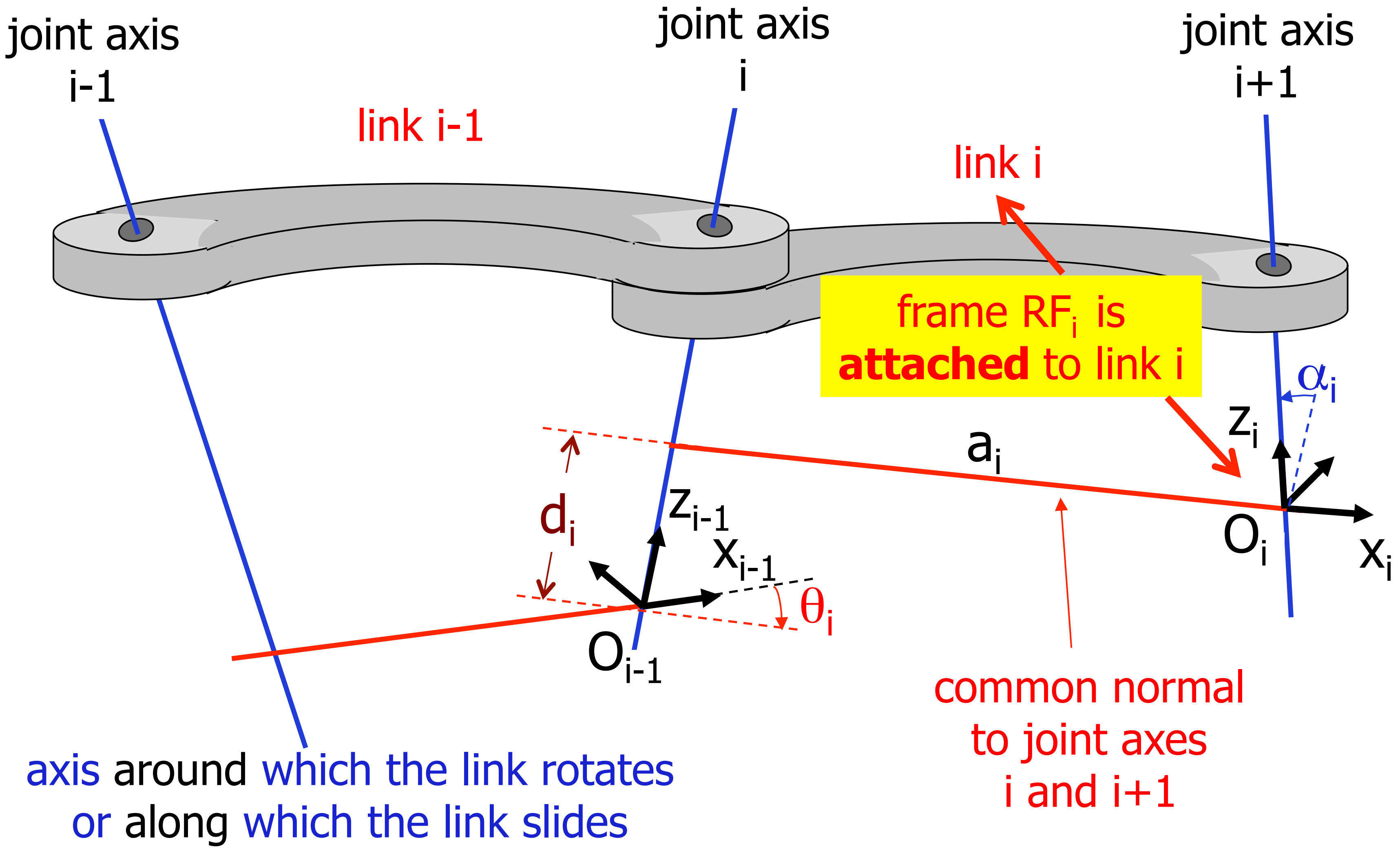


d_i = distance CD (a variable if joint i is prismatic)

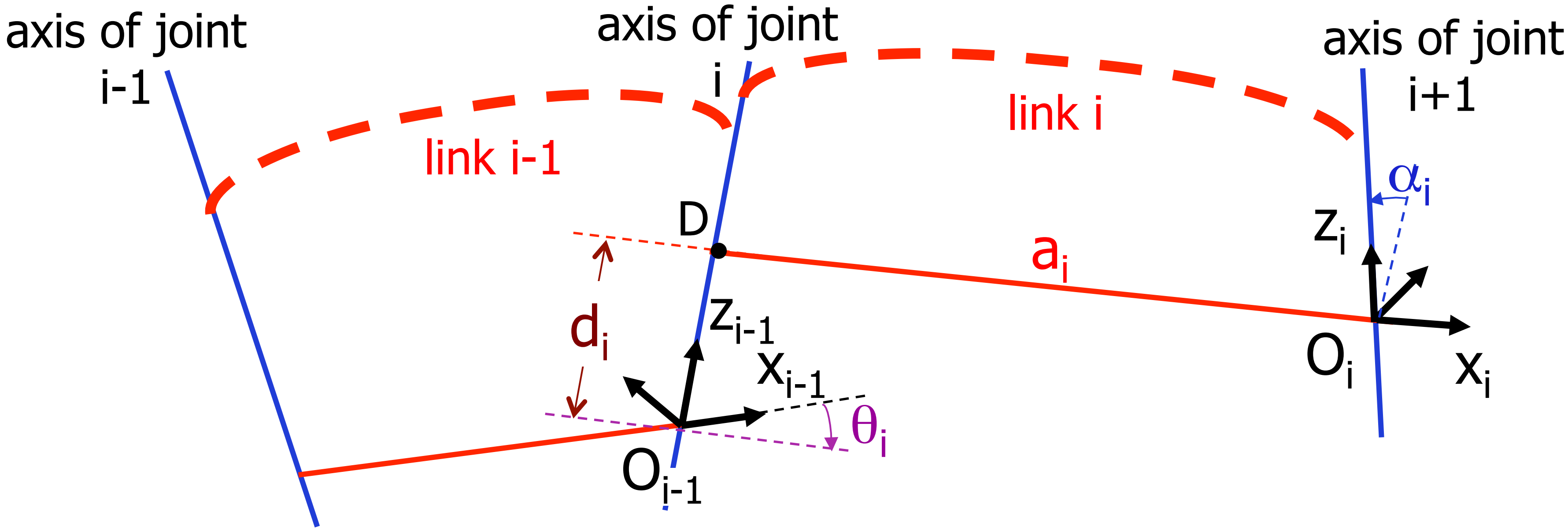
θ_i = angle (a variable if joint i is revolute) between link axes [projected on a plane σ orthogonal to the joint axis]

} with sign (pos/neg)!

Frame assignment by Denavit-Hartenberg (DH)



Denavit-Hartenberg parameters



- unit vector z_i along axis of joint i+1
- unit vector x_i along the common normal to joint i and i+1 axes (i → i+1)
- a_i = distance DO_i – positive if oriented as x_i (constant = "length" of link i)
- d_i = distance $O_{i-1}D$ – positive if oriented as z_{i-1} (variable if joint i is PRISMATIC)
- α_i = twist angle between z_{i-1} and z_i around x_i (constant)
- θ_i = angle between x_{i-1} and x_i around z_{i-i} (variable if joint i is REVOLUTE)

Homogeneous transformation

between DH frames (from frame$_{i-1}$ to frame$_i$)

- roto-translation around and along z_{i-1}

$$
{}^{i-1}A_{i'}(q_i) = \left[\begin{array}{ccc|c} c\theta_i & -s\theta_i & 0 & 0 \\ s\theta_i & c\theta_i & 0 & 0 \\ 0 & 0 & 1 & 0 \\ \hline 0 & 0 & 0 & 1 \end{array}\right] \left[\begin{array}{ccc|c} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_i \\ \hline 0 & 0 & 0 & 1 \end{array}\right] = \left[\begin{array}{ccc|c} c\theta_i & -s\theta_i & 0 & 0 \\ s\theta_i & c\theta_i & 0 & 0 \\ 0 & 0 & 1 & d_i \\ \hline 0 & 0 & 0 & 1 \end{array}\right]
$$

rotational joint $\Rightarrow$ $q_i = \theta_i$

prismatic joint $\Rightarrow$ $q_i = d_i$

- roto-translation around and along x_i

$$
{}^{i'}A_i = \left[\begin{array}{ccc|c} 1 & 0 & 0 & a_i \\ 0 & c\alpha_i & -s\alpha_i & 0 \\ 0 & s\alpha_i & c\alpha_i & 0 \\ \hline 0 & 0 & 0 & 1 \end{array}\right]
$$

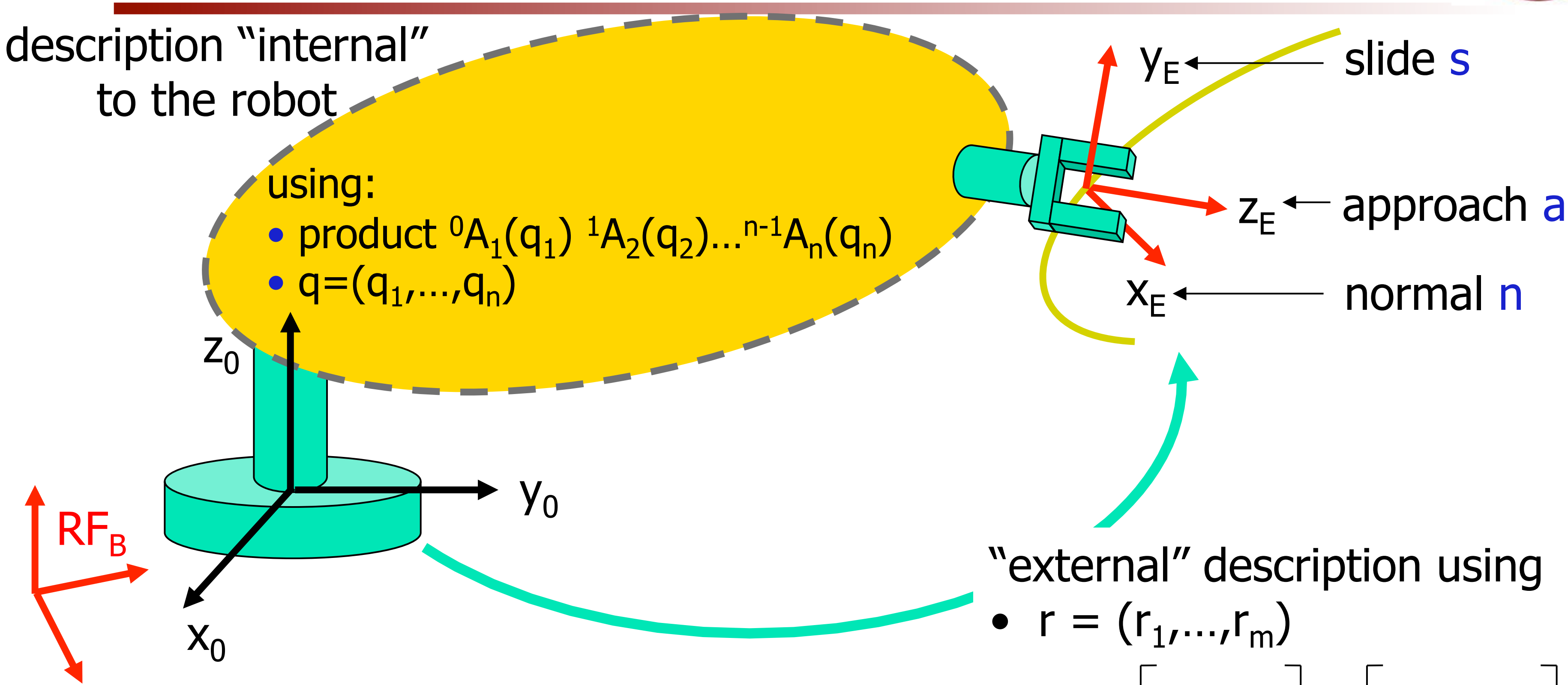
always a constant matrix

Denavit-Hartenberg matrix

$$
{}^{i-1}A_i\,(q_i) = {}^{i-1}A_{i'}\,(q_i)\;{}^{i'}A_i = \left[\begin{array}{ccc|c} c\theta_i & -c\alpha_i\, s\theta_i & s\alpha_i\, s\theta_i & a_i c\theta_i \\ s\theta_i & c\alpha_i\, c\theta_i & -s\alpha_i\, c\theta_i & a_i s\theta_i \\ 0 & s\alpha_i & c\alpha_i & d_i \\ \hline 0 & 0 & 0 & 1 \end{array}\right]
$$

compact notation: c = cos, s = sin

Direct kinematics of manipulators



"external" description using

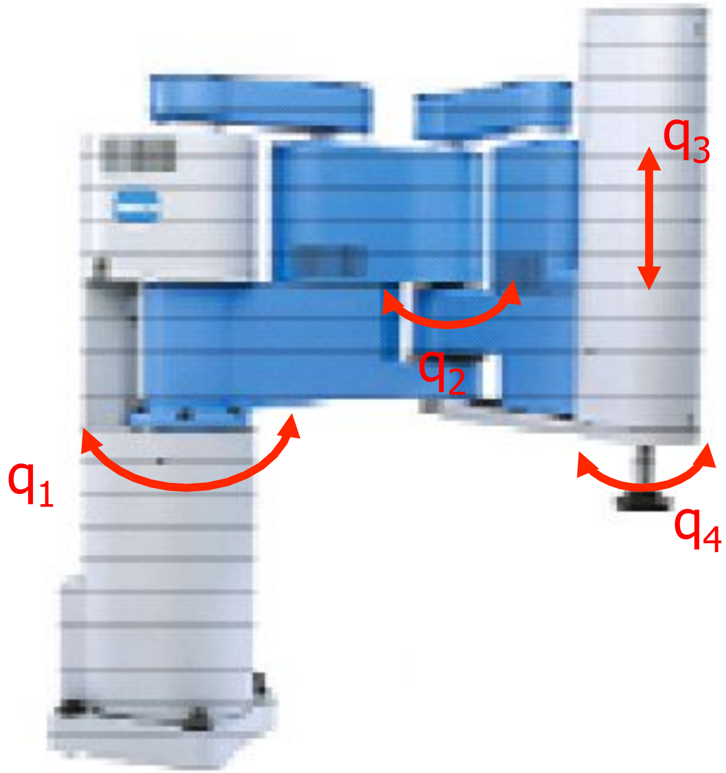
- $r = (r_1,\ldots,r_m)$
- $${}^BT_E = \left[\begin{array}{c:c} R & p \\ \hdashline 000 & 1 \end{array}\right] = \left[\begin{array}{ccc:c} n & s & a & p \\ \hdashline 0 & 0 & 0 & 1 \end{array}\right]$$

$${}^BT_E = {}^BT_0\ {}^0A_1(q_1)\ {}^1A_2(q_2)\ \ldots{}^{n-1}A_n(q_n)\ {}^nT_E$$

$$r = f_r(q)$$

alternative descriptions of robot direct kinematics

Example: SCARA robot

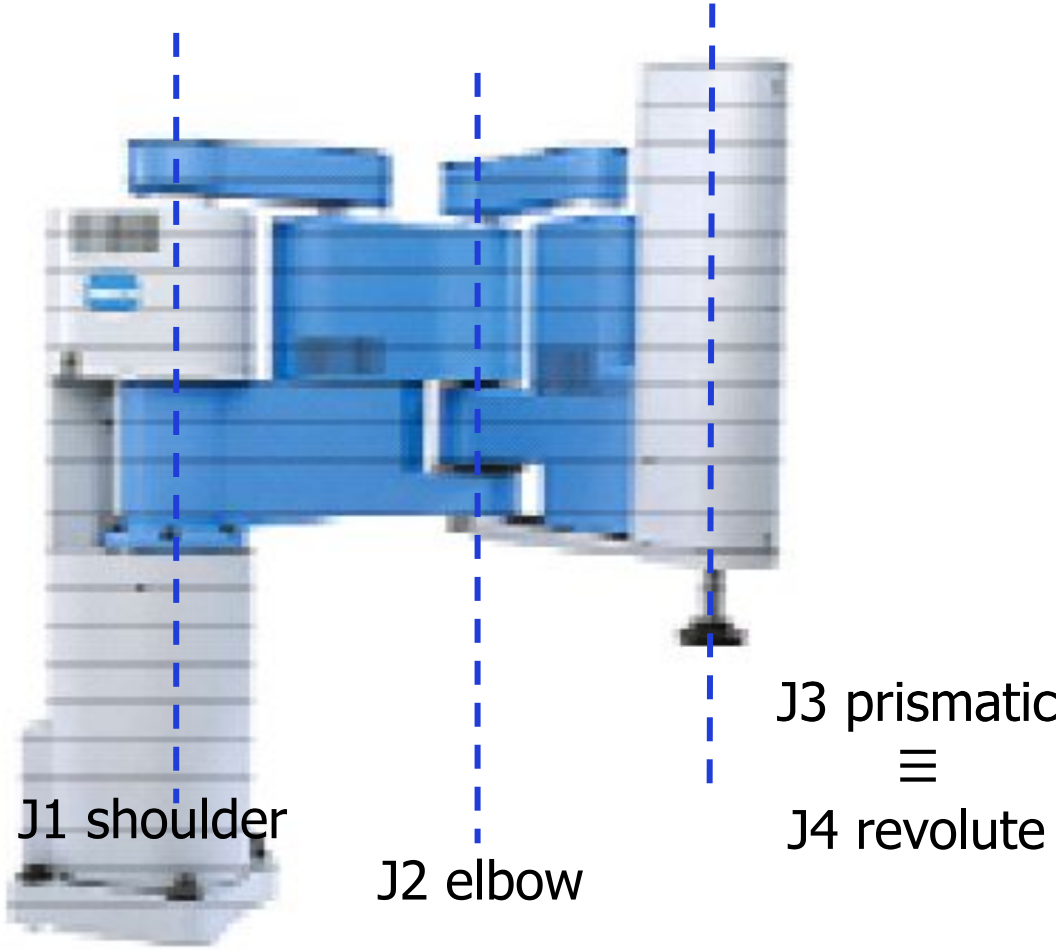


Step 1: joint axes

all parallel
(or coincident)

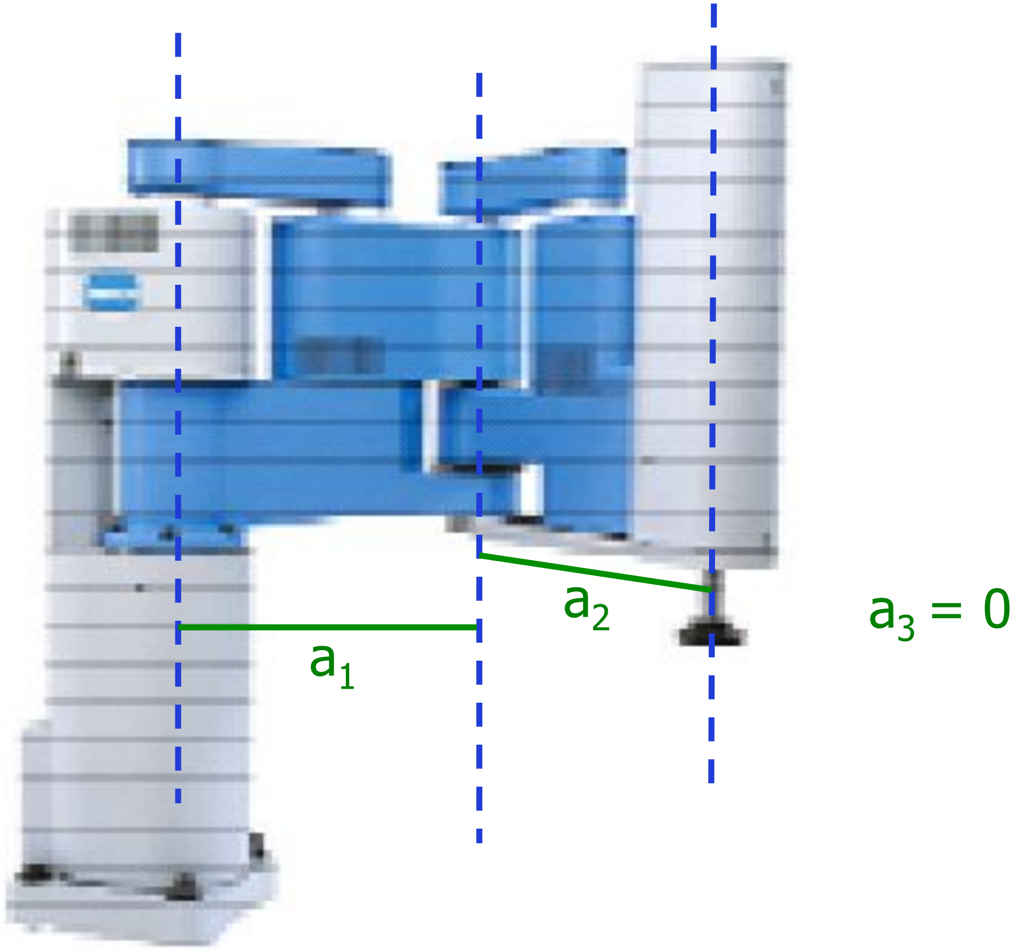
⬇

twists $\alpha_i = 0$
or π



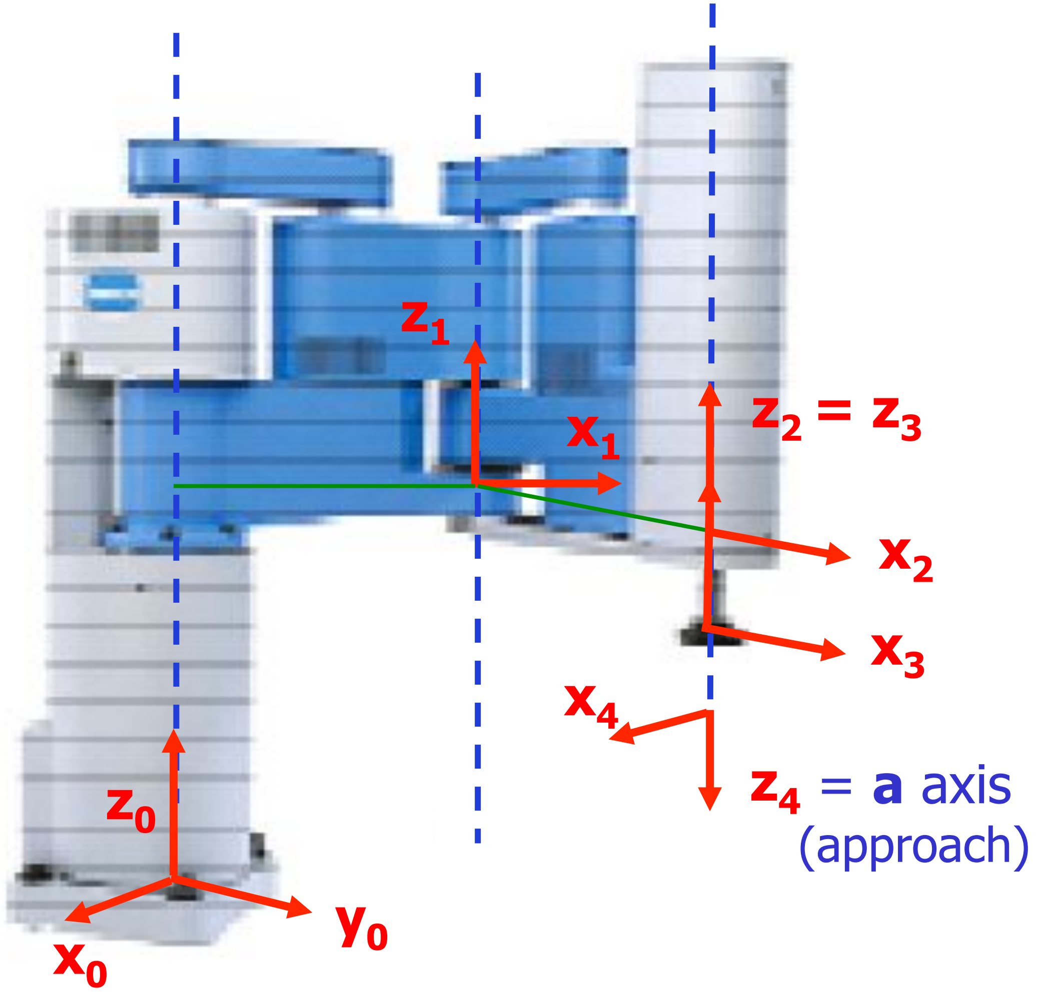
Step 2: link axes

the vertical "heights" of the link axes are arbitrary (for the time being)

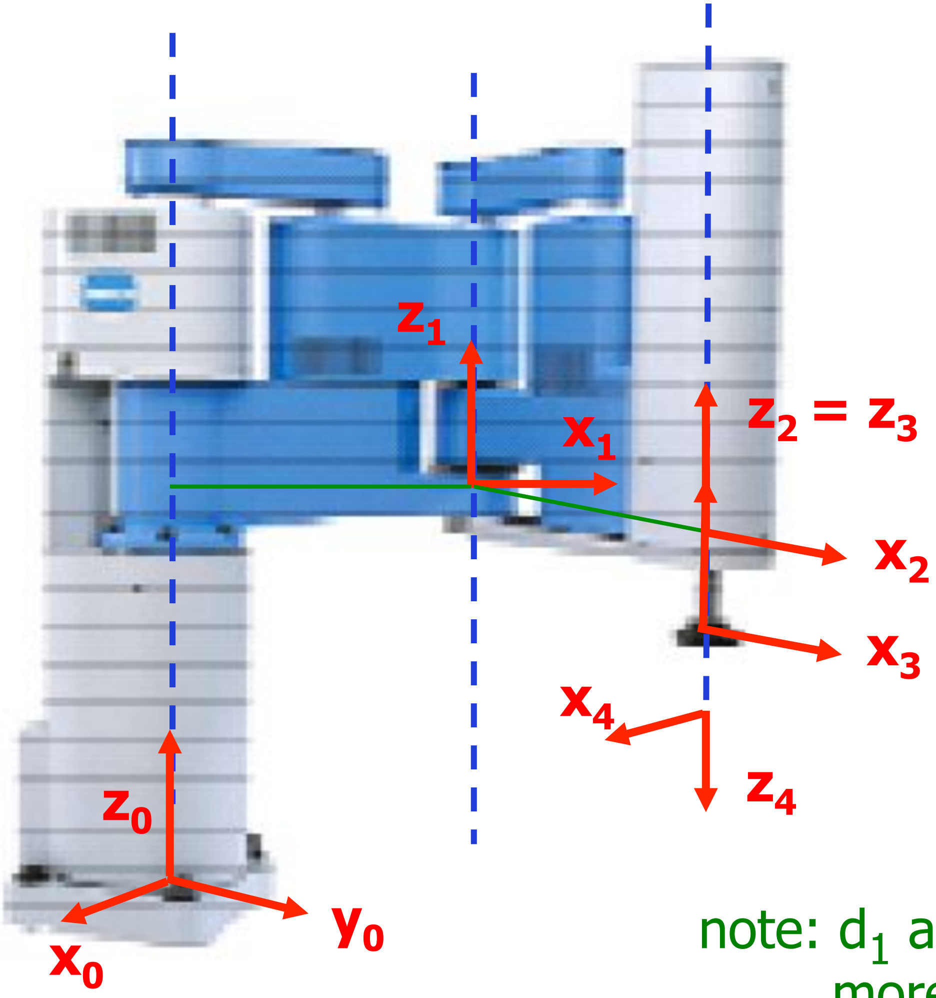


Step 3: frames

y_i axes for i > 0
are not shown
(and not needed;
they form
right-handed frames)



Step 4: DH parameters table



i	α_i	a_i	d_i	θ_i
1	0	a_1	d_1	q_1
2	0	a_2	0	q_2
3	0	0	q_3	0
4	π	0	d_4	q_4

note: d_1 and d_4 could have been chosen = 0 !
moreover, here it is $d_4 < 0$!!

Step 5: transformation matrices

$$
{}^0A_1(q_1) = \begin{bmatrix} c\theta_1 & -s\theta_1 & 0 & a_1 c\theta_1 \\ s\theta_1 & c\theta_1 & 0 & a_1 s\theta_1 \\ 0 & 0 & 1 & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}
$$

$$
{}^1A_2(q_2) = \begin{bmatrix} c\theta_2 & -s\theta_2 & 0 & a_2 c\theta_2 \\ s\theta_2 & c\theta_2 & 0 & a_2 s\theta_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}
$$

$$
{}^2A_3(q_3) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_3 \\ 0 & 0 & 0 & 1 \end{bmatrix}
$$

$$
{}^3A_4(q_4) = \begin{bmatrix} c\theta_4 & s\theta_4 & 0 & 0 \\ s\theta_4 & -c\theta_4 & 0 & 0 \\ 0 & 0 & -1 & d_4 \\ 0 & 0 & 0 & 1 \end{bmatrix}
$$

$$
\begin{aligned} q &= (q_1, q_2, q_3, q_4) \\ &= (\theta_1, \theta_2, d_3, \theta_4) \end{aligned}
$$

Step 6: direct kinematics

$$ {}^0A_3(q_1,q_2,q_3)=\begin{bmatrix} c_{12} & -s_{12} & 0 & a_1c_1+a_2c_{12} \\ s_{12} & c_{12} & 0 & a_1s_1+a_2s_{12} \\ 0 & 0 & 1 & d_1+q_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

$$ {}^3A_4(q_4)=\begin{bmatrix} c_4 & s_4 & 0 & 0 \\ s_4 & -c_4 & 0 & 0 \\ 0 & 0 & -1 & d_4 \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

$R(q_1,q_2,q_4)=[\ n\ s\ a\ ]$

$$ {}^0A_4(q_1,q_2,q_3,q_4)=\begin{bmatrix} c_{124} & s_{124} & 0 & a_1c_1+a_2c_{12} \\ s_{124} & -c_{124} & 0 & a_1s_1+a_2s_{12} \\ 0 & 0 & -1 & d_1+q_3+d_4 \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

$p = p(q_1,q_2,q_3)$

Robotics 1

Differential kinematics

Prof. Alessandro De Luca

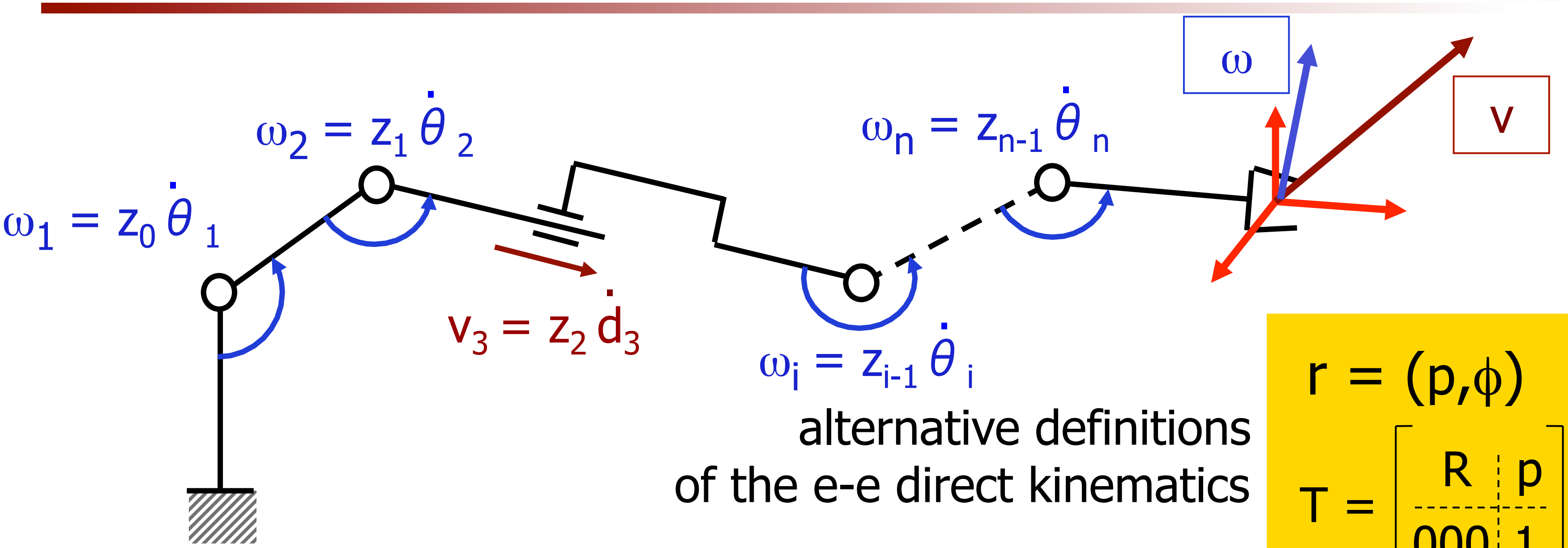
Dipartimento di Ingegneria Informatica
Automatica e Gestionale Antonio Ruberti

Sapienza
Università di Roma

Differential kinematics

- "relationship between motion (velocity) in the joint space and motion (linear and angular velocity) in the task (Cartesian) space"
- instantaneous velocity mappings can be obtained through time derivation of the direct kinematics function or geometrically at the differential level
  - different treatments arise for rotational quantities
  - establish the link between angular velocity and
    - time derivative of a rotation matrix
    - time derivative of the angles in a minimal representation of orientation

Linear and angular velocity of the robot end-effector



alternative definitions of the e-e direct kinematics

$$r = (p, \phi)$$

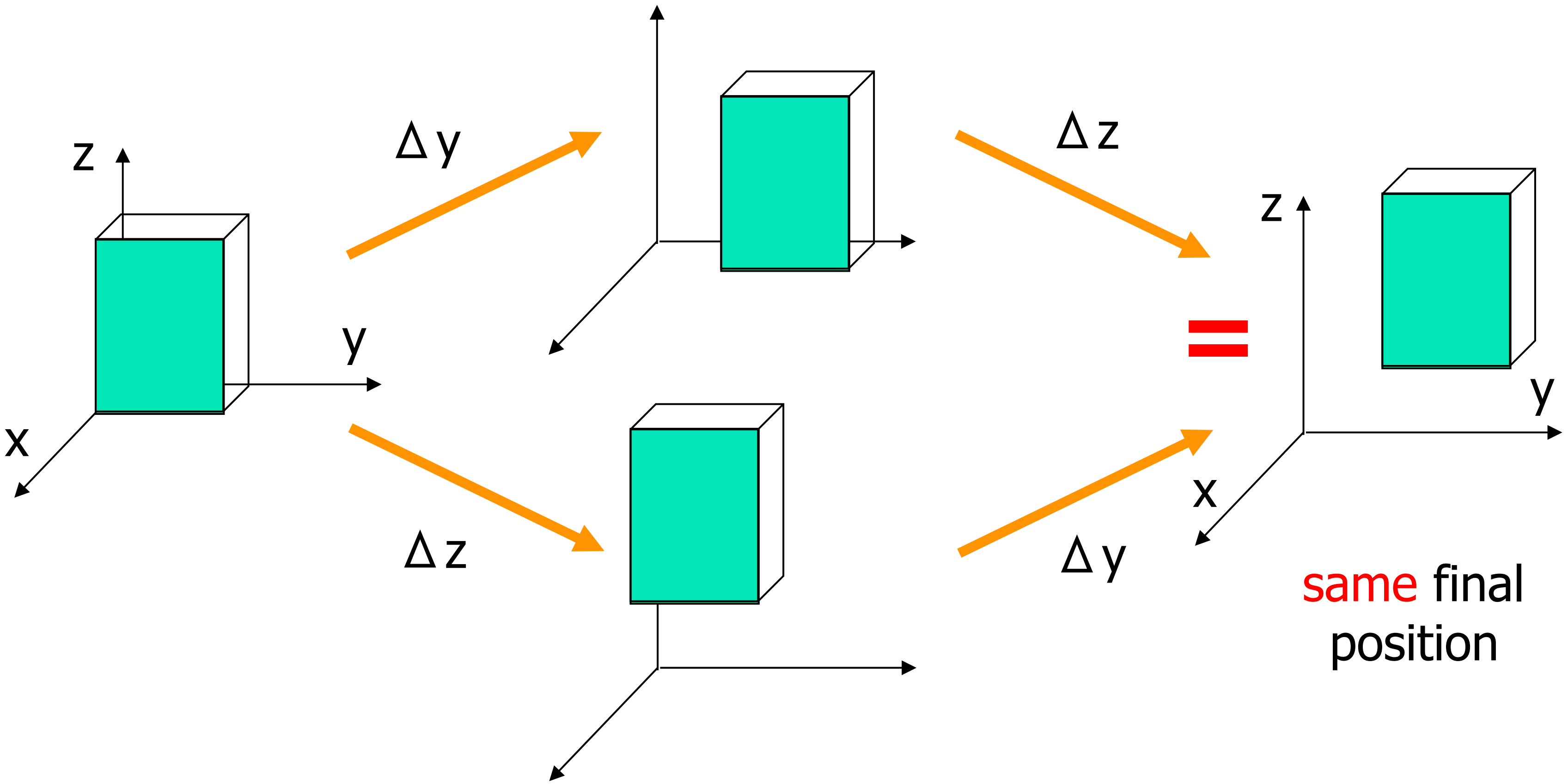
$$T = \left[\begin{array}{c|c} R & p \\ \hline 000 & 1 \end{array}\right]$$

- v and ω are "vectors", namely elements of vector spaces: they can be obtained as the sum of contributions of the joint velocities (in any order)
- on the other hand, ϕ (and $d\phi/dt$) is not an element of a vector space: a minimal representation of a sequence of rotations is not obtained by summing the corresponding minimal representations (angles ϕ)

in general, $\omega \neq d\phi/dt$

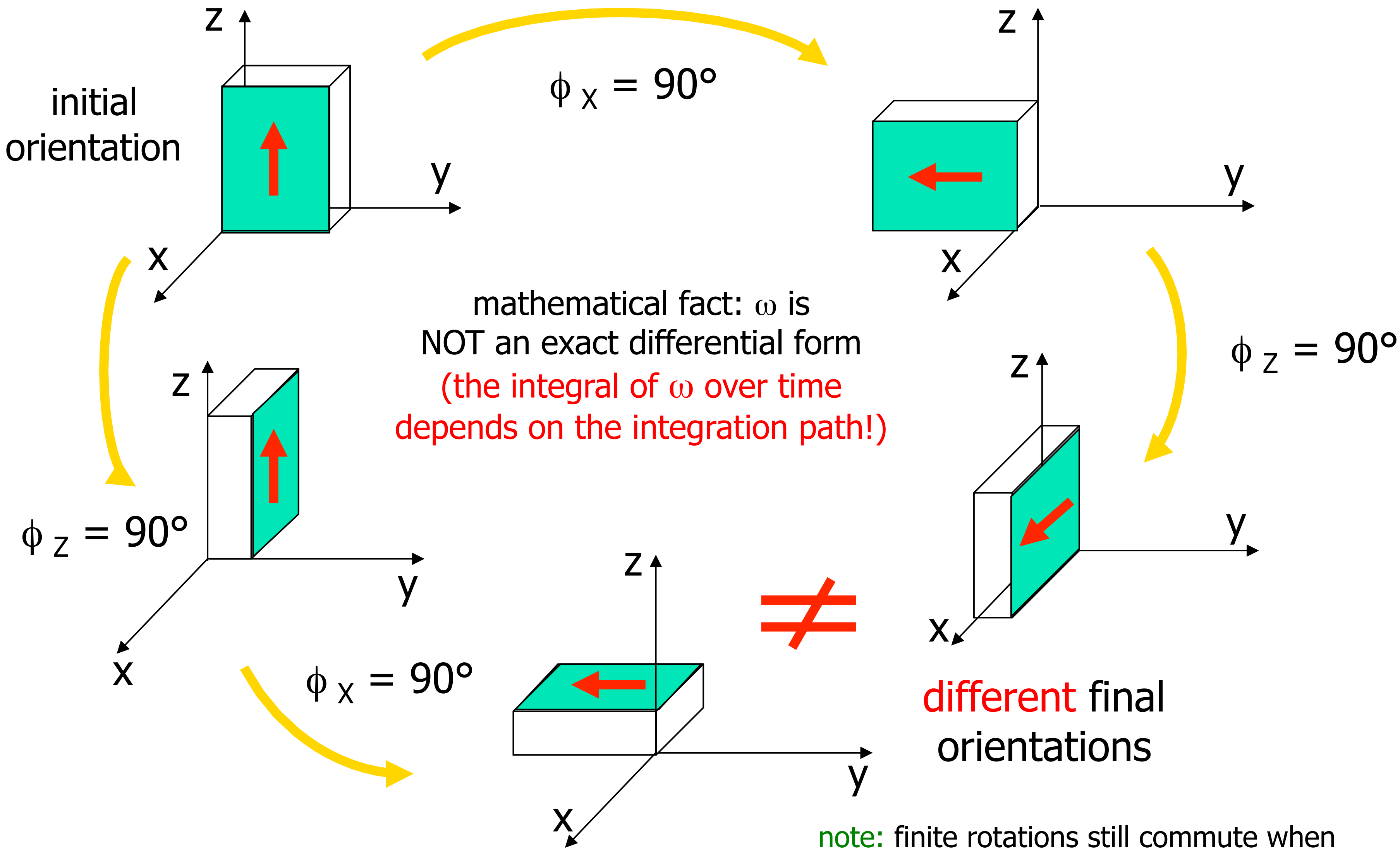
Finite and infinitesimal translations

- finite Δx, Δy, Δz or infinitesimal dx, dy, dz translations (linear displacements) always commute



Finite rotations do not commute

example



mathematical fact: ω is NOT an exact differential form
(the integral of ω over time depends on the integration path!)

different final orientations

note: finite rotations still commute when made around the same fixed axis

Infinitesimal rotations commute!

- infinitesimal rotations $d\phi_X$, $d\phi_Y$, $d\phi_Z$ around x,y,z axes

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

$$R_Y(\phi_Y) = \begin{bmatrix} \cos\phi_Y & 0 & \sin\phi_Y \\ 0 & 1 & 0 \\ -\sin\phi_Y & 0 & \cos\phi_Y \end{bmatrix} \Rightarrow R_Y(d\phi_Y) = \begin{bmatrix} 1 & 0 & d\phi_Y \\ 0 & 1 & 0 \\ -d\phi_Y & 0 & 1 \end{bmatrix}$$

$$R_Z(\phi_Z) = \begin{bmatrix} \cos\phi_Z & -\sin\phi_Z & 0 \\ \sin\phi_Z & \cos\phi_Z & 0 \\ 0 & 0 & 1 \end{bmatrix} \Rightarrow R_Z(d\phi_Z) = \begin{bmatrix} 1 & -d\phi_Z & 0 \\ d\phi_Z & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

- $$R(d\phi) = \underset{\text{in any sequence}}{R(d\phi_X, d\phi_Y, d\phi_Z)} = \begin{bmatrix} 1 & -d\phi_Z & d\phi_Y \\ d\phi_Z & 1 & -d\phi_X \\ -d\phi_Y & d\phi_X & 1 \end{bmatrix} \leftarrow \text{neglecting second- and third-order (infinitesimal) terms}$$

$$= I + S(d\phi)$$

Time derivative of a rotation matrix

- let $R = R(t)$ be a rotation matrix, given as a function of time
- since $I = R(t)R^T(t)$, taking the time derivative of both sides yields

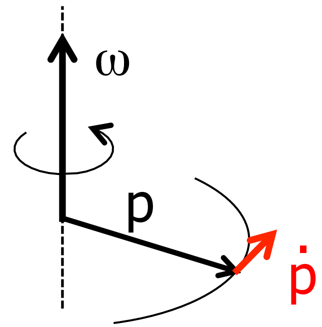
$$0 = d[R(t)R^T(t)]/dt = dR(t)/dt\, R^T(t) + R(t)\, dR^T(t)/dt$$
$$= dR(t)/dt\, R^T(t) + [dR(t)/dt\, R^T(t)]^T$$

thus $dR(t)/dt\, R^T(t) = S(t)$ is a skew-symmetric matrix

- let $p(t) = R(t)p'$ a vector (with constant norm) rotated over time
- comparing

$$dp(t)/dt = dR(t)/dt\, p' = S(t)R(t)\, p' = S(t)\, p(t)$$
$$dp(t)/dt = \omega(t) \times p(t) = S(\omega(t))\, p(t)$$

we get $S = S(\omega)$



$$\boxed{\dot{R} = S(\omega)\, R} \quad \Longleftrightarrow \quad \boxed{S(\omega) = \dot{R}\, R^T}$$

Robot Jacobian matrices

- analytical Jacobian (obtained by time differentiation)

$$r = \begin{pmatrix} p \\ \phi \end{pmatrix} = f_r(q) \quad \Rightarrow \quad \dot{r} = \begin{pmatrix} \dot{p} \\ \dot{\phi} \end{pmatrix} = \frac{\partial f_r(q)}{\partial q}\,\dot{q} = J_r(q)\,\dot{q}$$

- geometric Jacobian (no derivatives)

$$\begin{pmatrix} v \\ \omega \end{pmatrix} = \begin{pmatrix} \dot{p} \\ \omega \end{pmatrix} = \begin{pmatrix} J_L(q) \\ J_A(q) \end{pmatrix}\dot{q} = J(q)\,\dot{q}$$

Geometric Jacobian

always a 6 x n matrix

end-effector instantaneous velocity
$$\begin{pmatrix} v_E \\ \omega_E \end{pmatrix} = \begin{pmatrix} J_L(q) \\ J_A(q) \end{pmatrix} \dot{q} = \begin{pmatrix} J_{L1}(q) & \cdots & J_{Ln}(q) \\ J_{A1}(q) & \cdots & J_{An}(q) \end{pmatrix} \begin{pmatrix} \dot{q}_1 \\ \vdots \\ \dot{q}_n \end{pmatrix}$$

superposition of effects

$$v_E = \underbrace{J_{L1}(q)\, \dot{q}_1}_{} + \ldots + J_{Ln}(q)\, \dot{q}_n \qquad \omega_E = \underbrace{J_{A1}(q)\, \dot{q}_1}_{} + \ldots + J_{An}(q)\, \dot{q}_n$$

contribution to the linear e-e velocity due to $\dot{q}_1$

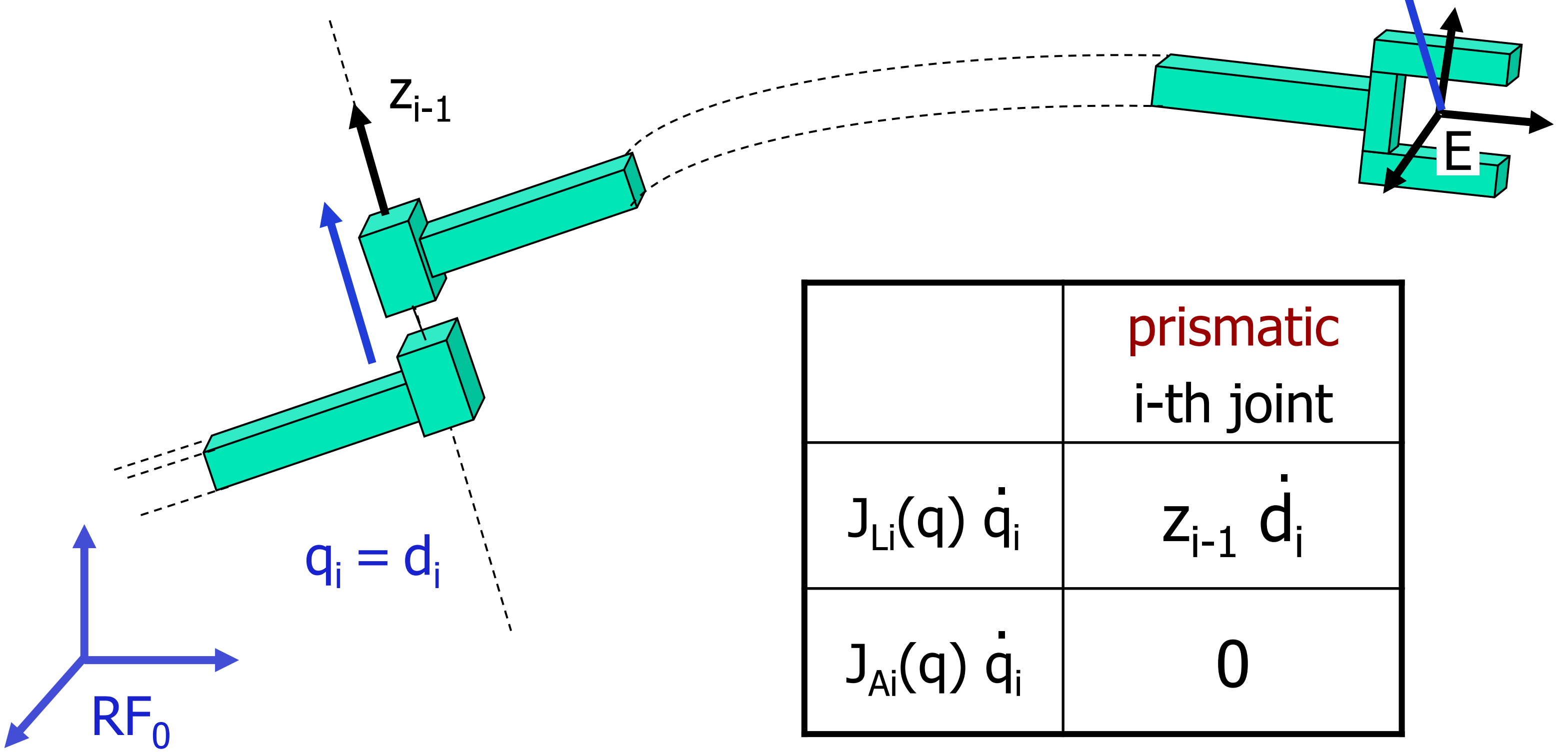
contribution to the angular e-e velocity due to $\dot{q}_1$

linear and angular velocity belong to (linear) vector spaces in R^3

Contribution of a prismatic joint

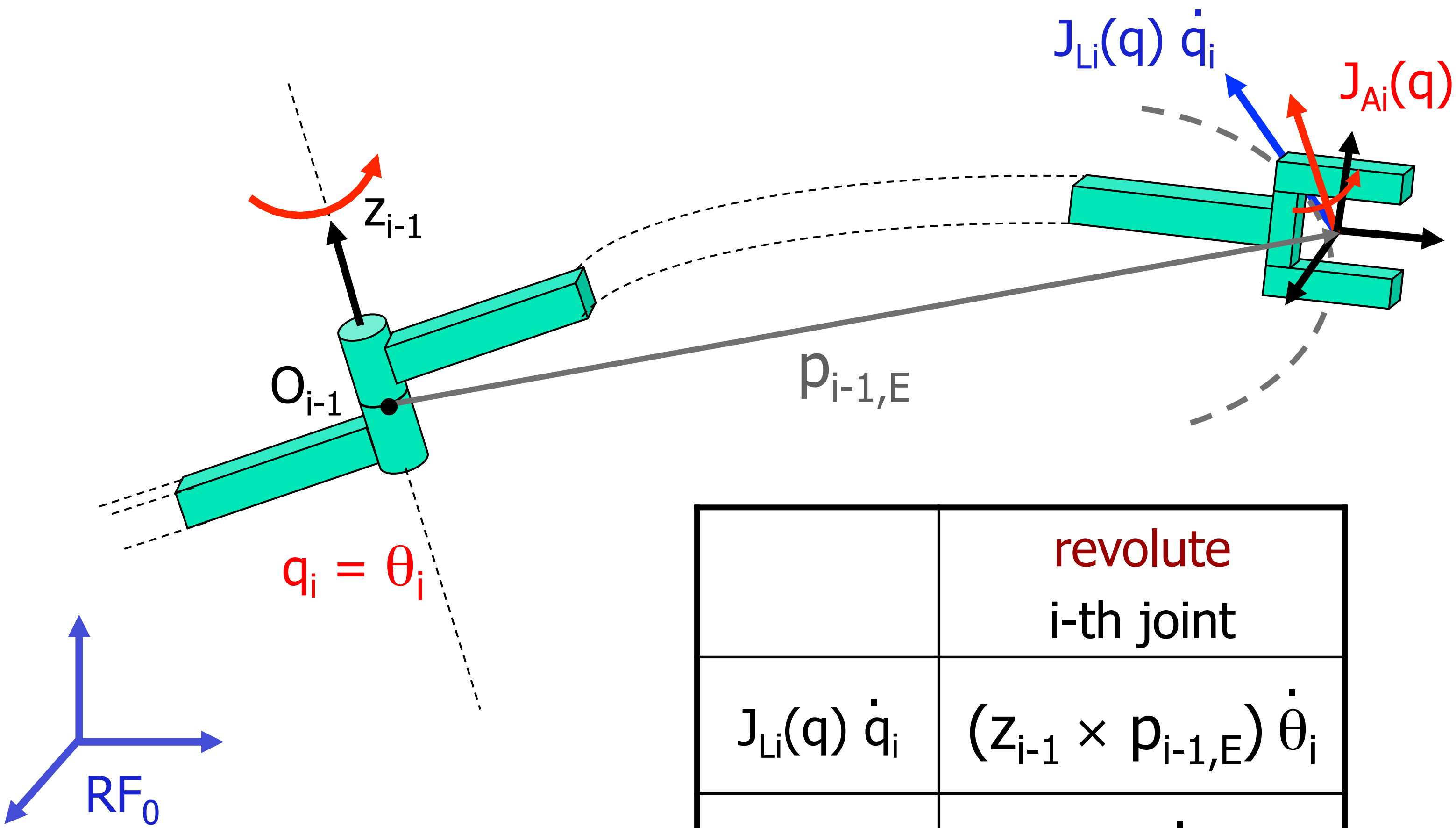
Note: joints beyond the i-th one are considered to be "frozen", so that the distal part of the robot is a single rigid body

$J_{Li}(q)\,\dot{q}_i = z_{i-1}\,\dot{d}_i$



	prismatic i-th joint
$J_{Li}(q)\,\dot{q}_i$	$z_{i-1}\,\dot{d}_i$
$J_{Ai}(q)\,\dot{q}_i$	0

Contribution of a revolute joint



	revolute i-th joint
$J_{Li}(q)\,\dot{q}_i$	$(z_{i-1} \times p_{i-1,E})\,\dot{\theta}_i$
$J_{Ai}(q)\,\dot{q}_i$	$z_{i-1}\,\dot{\theta}_i$

Expression of geometric Jacobian

$$\left(\begin{pmatrix}\dot{p}_{0,E}\\ \omega_E\end{pmatrix}=\right)\begin{pmatrix}v_E\\ \omega_E\end{pmatrix}=\begin{pmatrix}J_L(q)\\ J_A(q)\end{pmatrix}\dot{q}=\begin{pmatrix}J_{L1}(q) & \cdots & J_{Ln}(q)\\ J_{A1}(q) & \cdots & J_{An}(q)\end{pmatrix}\begin{pmatrix}\dot{q}_1\\ \vdots\\ \dot{q}_n\end{pmatrix}$$

	prismatic i-th joint	revolute i-th joint
$J_{Li}(q)$	z_{i-1}	$z_{i-1} \times p_{i-1,E}$
$J_{Ai}(q)$	0	z_{i-1}

this can be also computed as

$$= \boxed{\frac{\partial p_{0,E}}{\partial q_i}}$$

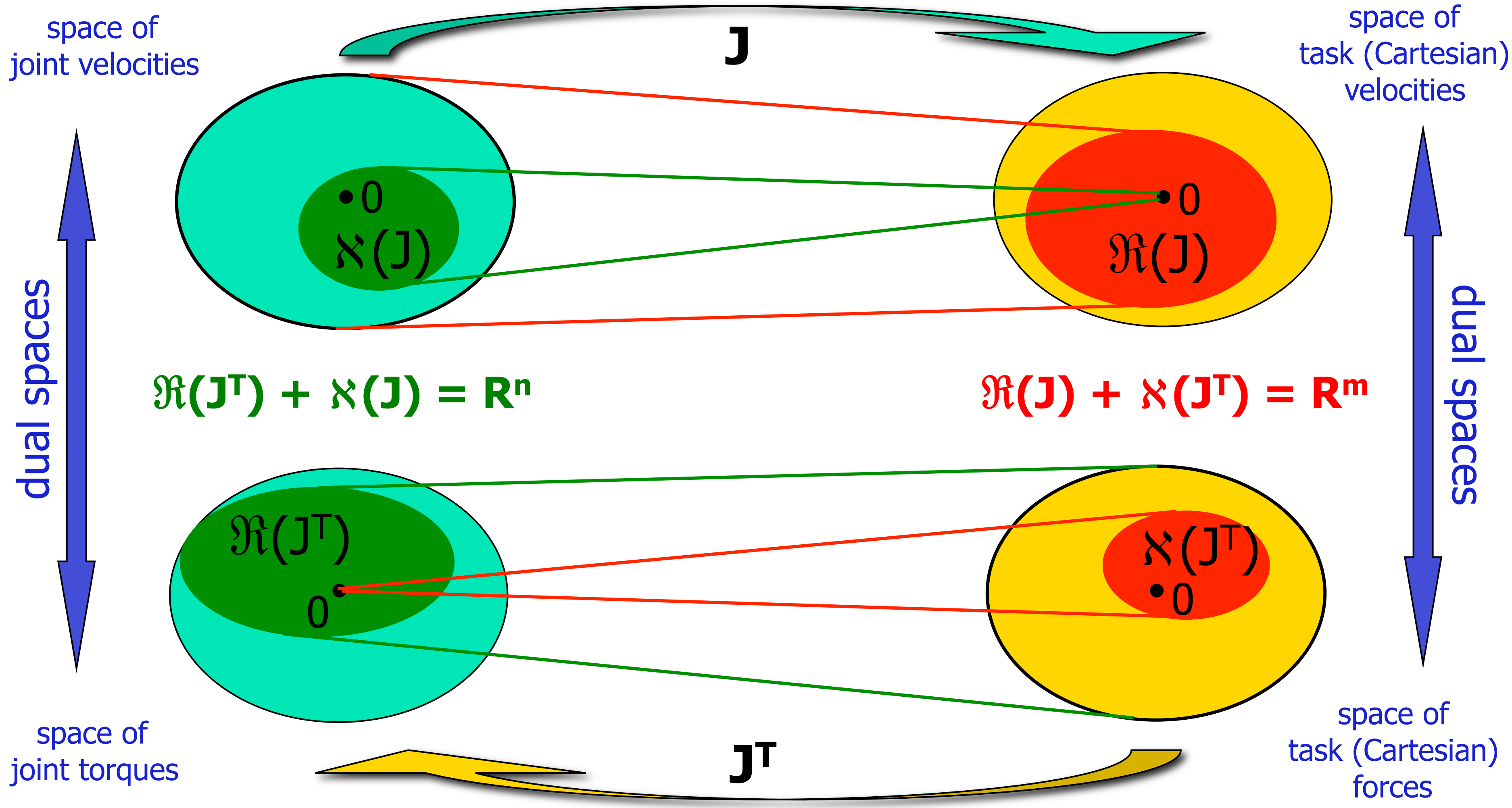
$$z_{i-1} = {}^0R_1(q_1)\ldots{}^{i-2}R_{i-1}(q_{i-1})\begin{pmatrix}0\\0\\1\end{pmatrix}$$

$$p_{i-1,E} = p_{0,E}(q_1,\ldots,q_n) - p_{0,i-1}(q_1,\ldots,q_{i-1})$$

all vectors should be expressed in the same reference frame (here, the base frame RF_0)

Robot Jacobian

decomposition in linear subspaces and duality



(in a given configuration q)

Mobility analysis

- $\rho(J) = \rho(J(q))$, $\Re(J) = \Re(J(q))$, $\aleph(J^T) = \aleph(J^T(q))$ are locally defined, i.e., they depend on the current configuration q
- $\Re(J(q))$ = subspace of all "generalized" velocities (with linear and/or angular components) that can be instantaneously realized by the robot end-effector when varying the joint velocities in the configuration q
- if J(q) has max rank (typically = m) in the configuration q, the robot end-effector can be moved in any direction of the task space R^m
- if $\rho(J(q)) < m$, there exist directions in R^m along which the robot end-effector cannot instantaneously move
  - these directions lie in $\aleph(J^T(q))$, namely the complement of $\Re(J(q))$ to the task space R^m, which is of dimension $m - \rho(J(q))$
- when $\aleph(J(q)) \neq \{0\}$ (this is always the case if m<n, i.e., in robots that are redundant for the task), there exist non-zero joint velocities that produce zero end-effector velocity ("self motions")

Kinematic singularities

- **configurations where the Jacobian loses rank**
  ⇔ loss of instantaneous mobility of the robot end-effector
- for m=n, they correspond in general to Cartesian poses that lead to a number of inverse kinematic solutions that differs from the "generic" case
- "in" a singular configuration, one cannot find a joint velocity that realizes a desired end-effector velocity in an arbitrary direction of the task space
- "close" to a singularity, large joint velocities may be needed to realize some (even small) velocity of the end-effector
- finding and analyzing in advance all singularities of a robot helps in avoiding them during trajectory planning and motion control
  - when $m = n$: find the configurations q such that $\det J(q) = 0$
  - when $m < n$: find the configurations q such that all $m \times m$ minors of J are singular (or, equivalently, such that $\det [J(q)\, J^T(q)] = 0$)
- finding all singular configurations of a robot with a large number of joints, or the actual "distance" from a singularity, is a hard computational task