POLITECNICO
MILANO 1863
DIPARTIMENTO DI ELETTRONICA
INFORMAZIONE E BIOINGEGNERIA

Control of industrial robots

Review of robot kinematics

22.09.2025 I Paolo Rocco

Contents

Position and orientation of a body in space

01

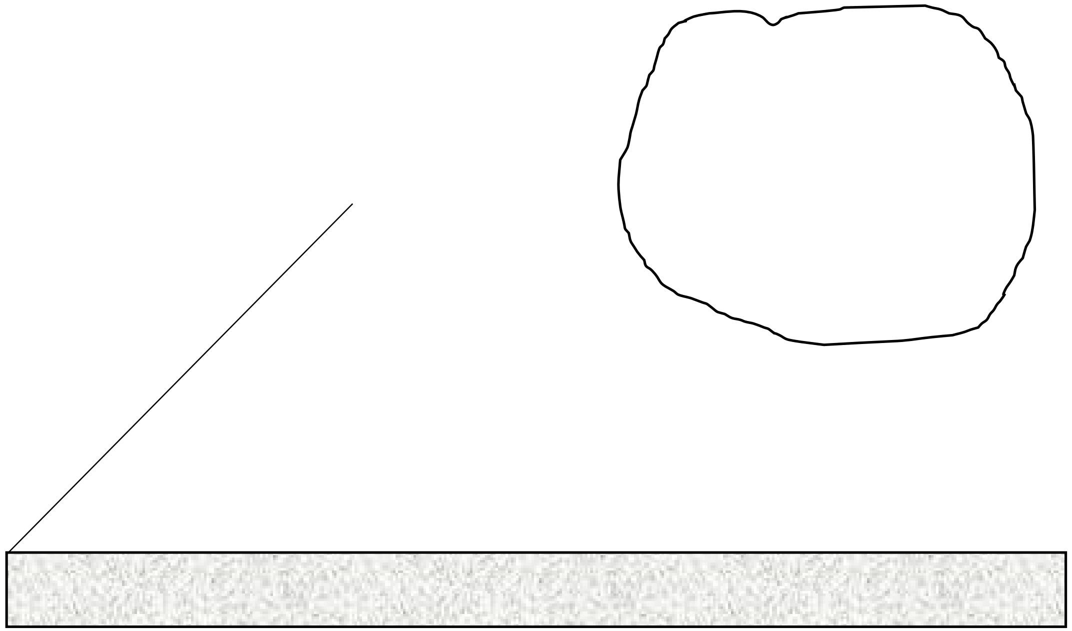
Introduction

- With these slides we will cover **basic elements in robot kinematics**.
- We will start from a basic problem of representation of a rigid body in space, and then proceed through the formal tools used in robotics till the definition of the direct, inverse and differential kinematics of the manipulator.
- All this material is well covered with better detail in any introductory Robotics course at BSc level. It is reviewed here for the sole purpose of making this course self-contained for students who lack this background.

Some of the pictures in these slides are taken from the textbook:
B. Siciliano, L. Sciavicco, L. Villani, G. Oriolo: *Robotics: Modelling, Planning and Control, 3rd Ed.*
Springer, 2009

Position and orientation of a body in space

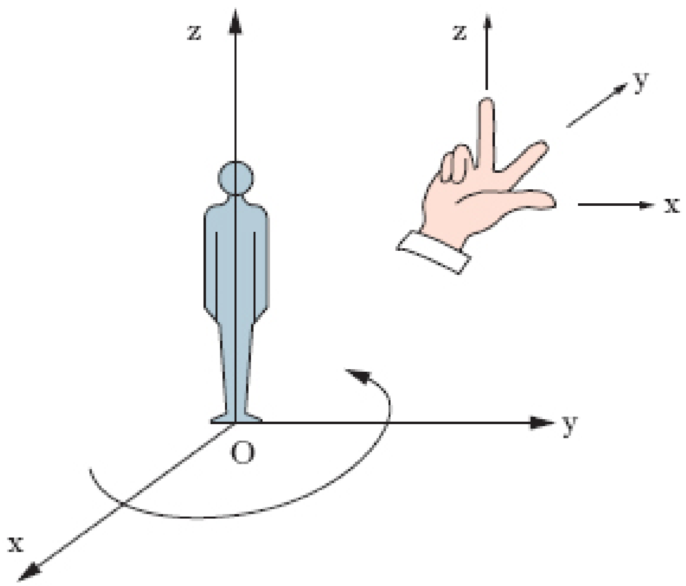
Let us consider a rigid body in space:



How can we characterize the position and orientation of the body in space?

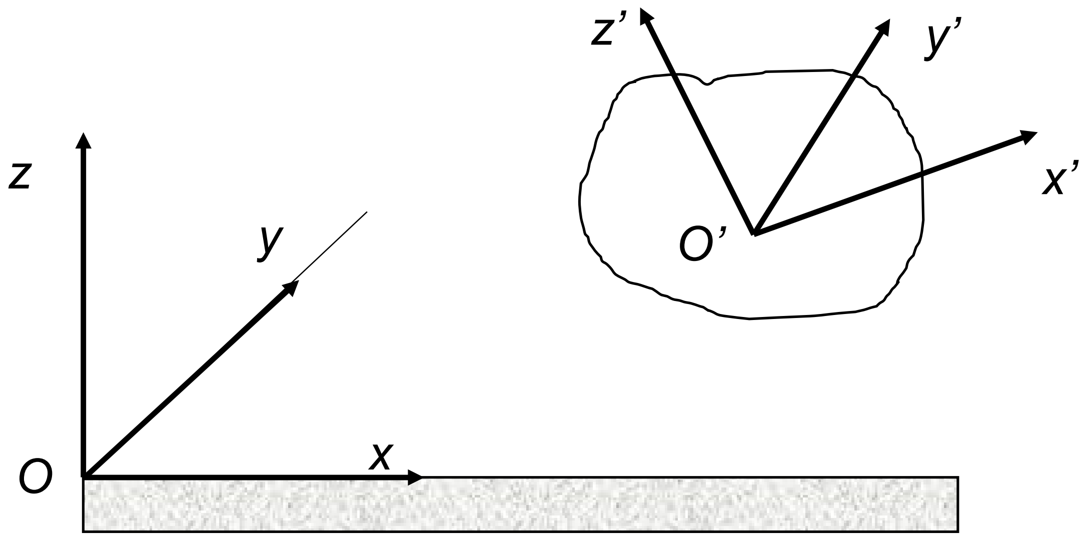
Position and orientation of a body in space

The study of kinematics of mechanical bodies is facilitated if Cartesian frames are introduced.
Each point in space has 3 coordinates (x, y, z) in the Cartesian frame.



Position and orientation of a body in space

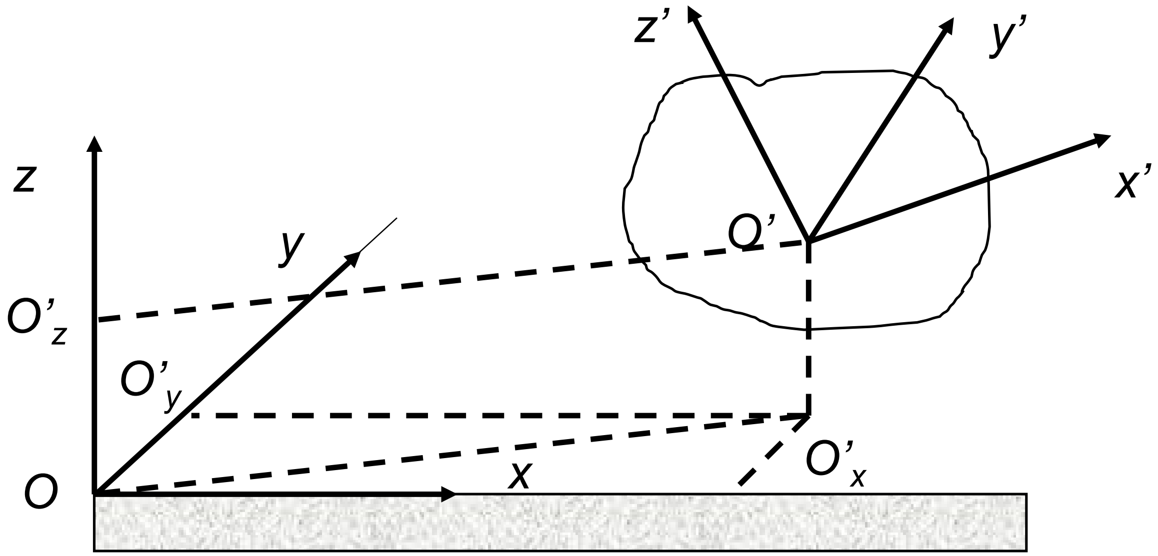
The best thing to do is to consider a reference frame and to attach a second frame to the body.



The problem is now how to characterize the position and orientation of a frame with respect to another one.

Position and orientation of a body in space

The representation of the position is just made with the components of the origin of the body-attached frame with respect to the reference frame:

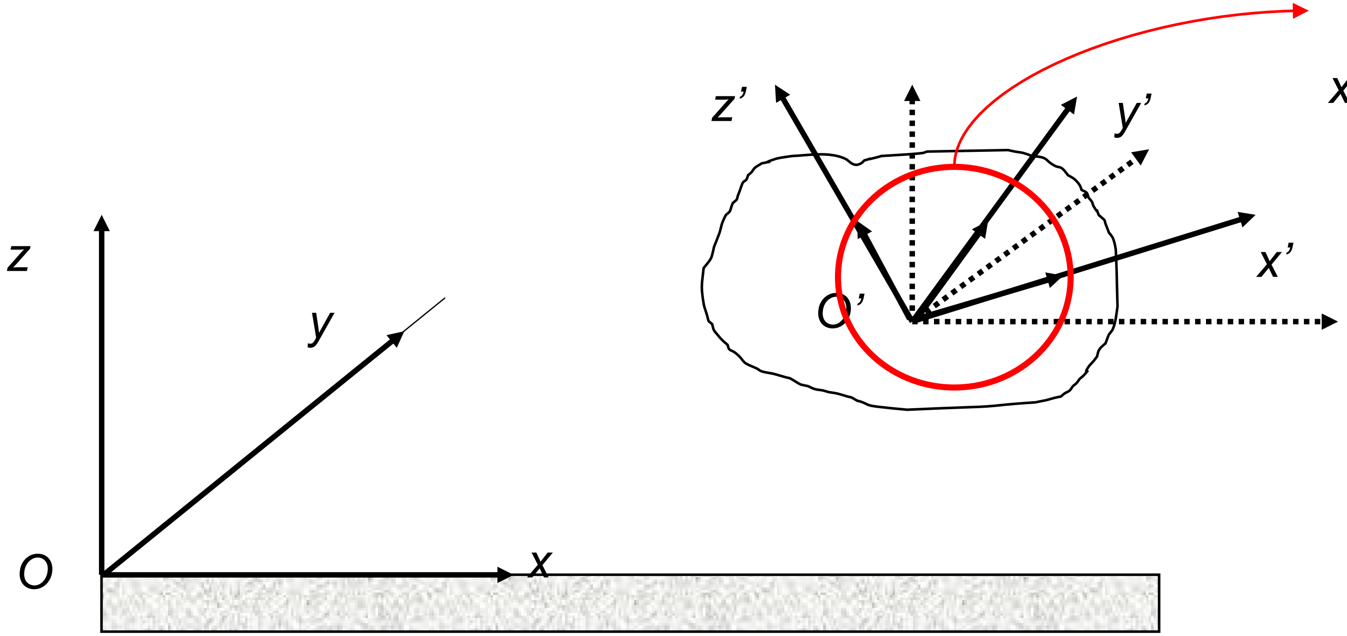


The three components can be conveniently gathered in a **vector**:

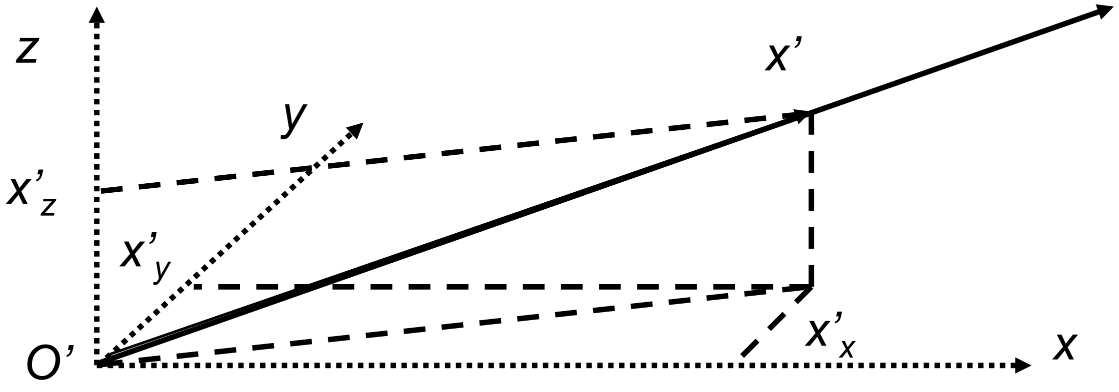
$$\mathbf{O}' = \begin{bmatrix} o'_x \\ o'_y \\ o'_z \end{bmatrix}$$

Position and orientation of a body in space

The representation of the orientation can be made considering unit length vectors along the axes of the rotated frame and evaluating their components in the reference frame:



For example, for the unit vector x':



We obtain three vectors:

$$\mathbf{x}' = \begin{bmatrix} x'_x \\ x'_y \\ x'_z \end{bmatrix}, \quad \mathbf{y}' = \begin{bmatrix} y'_x \\ y'_y \\ y'_z \end{bmatrix}, \quad \mathbf{z}' = \begin{bmatrix} z'_x \\ z'_y \\ z'_z \end{bmatrix}$$

Rotation matrix

We can gather the elements of $\mathbf{x}'$, $\mathbf{y}'$, $\mathbf{z}'$ in a matrix:

$$\mathbf{R} = [\mathbf{x}' \quad \mathbf{y}' \quad \mathbf{z}'] = \begin{bmatrix} x'_x & y'_x & z'_x \\ x'_y & y'_y & z'_y \\ x'_z & y'_z & z'_z \end{bmatrix} = \begin{bmatrix} \mathbf{x}'^T\mathbf{x} & \mathbf{y}'^T\mathbf{x} & \mathbf{z}'^T\mathbf{x} \\ \mathbf{x}'^T\mathbf{y} & \mathbf{y}'^T\mathbf{y} & \mathbf{z}'^T\mathbf{y} \\ \mathbf{x}'^T\mathbf{z} & \mathbf{y}'^T\mathbf{z} & \mathbf{z}'^T\mathbf{z} \end{bmatrix}$$

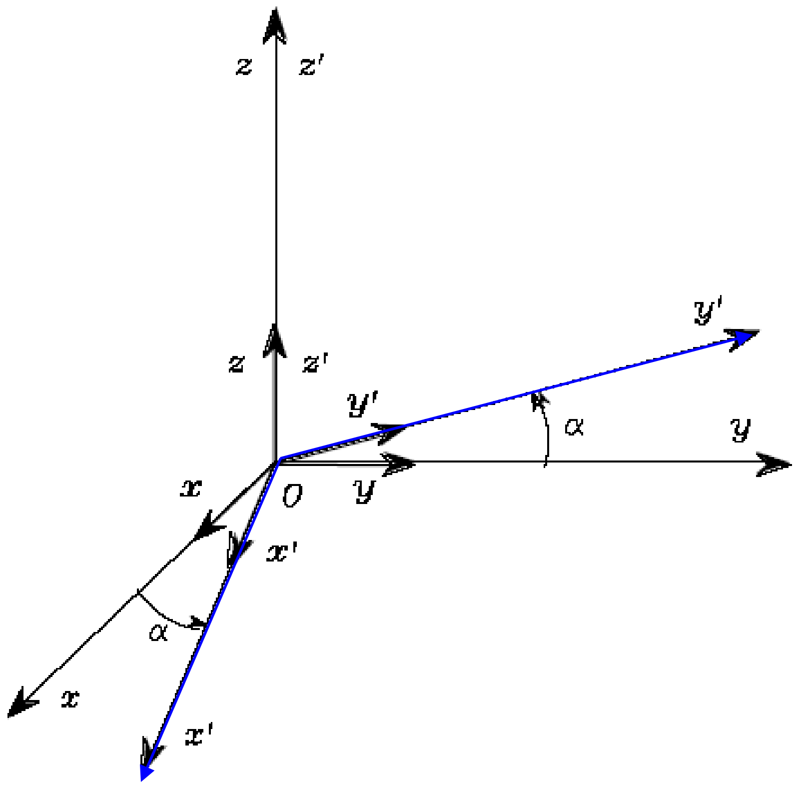
This matrix is called **rotation matrix** of the frame $(\mathbf{x}', \mathbf{y}', \mathbf{z}')$ with respect to the frame $(\mathbf{x}, \mathbf{y}, \mathbf{z})$.
Since the following relations hold:

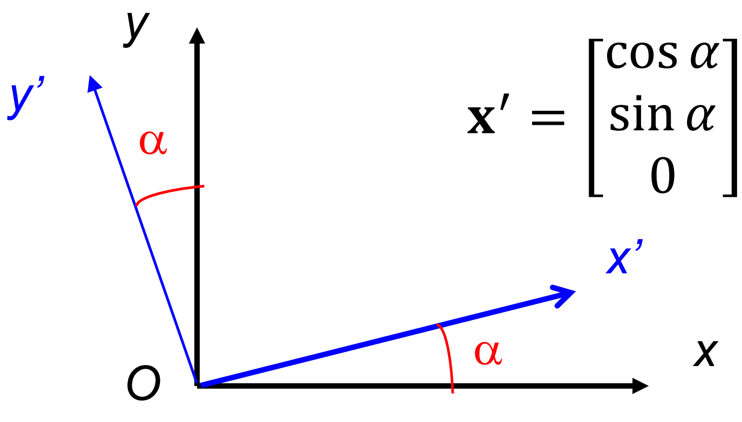
$$\mathbf{x}'^T\mathbf{x}' = 1, \quad \mathbf{y}'^T\mathbf{y}' = 1, \quad \mathbf{z}'^T\mathbf{z}' = 1$$
$$\mathbf{x}'^T\mathbf{y}' = 0, \quad \mathbf{y}'^T\mathbf{z}' = 0, \quad \mathbf{z}'^T\mathbf{x}' = 0$$

we have: $\mathbf{R}^T\mathbf{R} = \mathbf{I} \quad (\mathbf{R}^T = \mathbf{R}^{-1})$ **orthogonal matrix**

Elementary rotations

Let us consider a rotation by an angle α around $\boldsymbol{z}$ axis:





$$\mathbf{x}' = \begin{bmatrix} \cos\alpha \\ \sin\alpha \\ 0 \end{bmatrix}, \quad \mathbf{y}' = \begin{bmatrix} -\sin\alpha \\ \cos\alpha \\ 0 \end{bmatrix}, \quad \mathbf{z}' = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

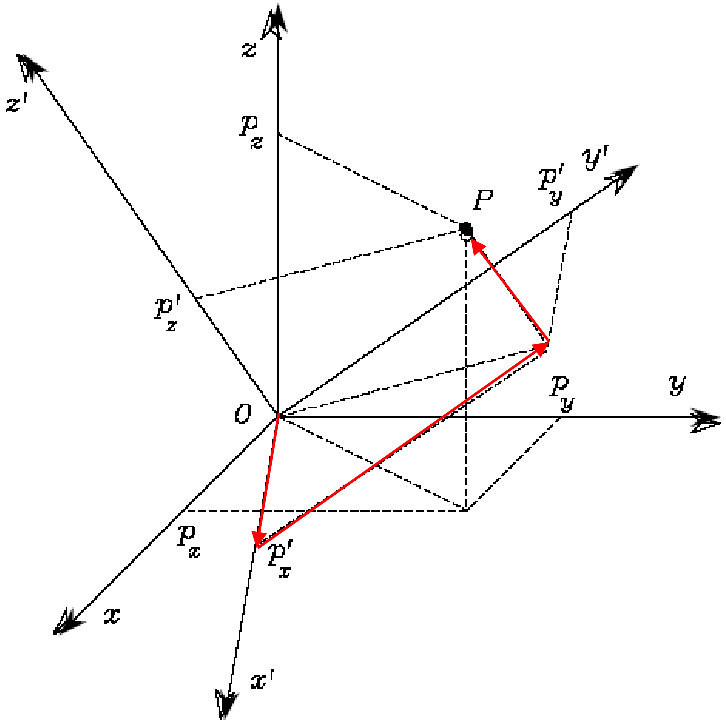
The rotation matrix is thus:

$$\mathbf{R}_z(\alpha) = \begin{bmatrix} \cos\alpha & -\sin\alpha & 0 \\ \sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Similarly for the rotations around the other axes.

Representation of a vector

Consider now a point P whose coordinates are expressed in two reference frames:



The coordinates of the **same point** in the two frames are:

$$\mathbf{p} = \begin{bmatrix} p_x \\ p_y \\ p_z \end{bmatrix}, \quad \mathbf{p}' = \begin{bmatrix} p'_x \\ p'_y \\ p'_z \end{bmatrix}$$

Therefore:

$$\mathbf{p} = p'_x\mathbf{x}' + p'_y\mathbf{y}' + p'_z\mathbf{z}' = [\mathbf{x}' \quad \mathbf{y}' \quad \mathbf{z}']\mathbf{p}' = \mathbf{R}\mathbf{p}'$$

The rotation matrix thus encodes the **transformation** which maps the coordinates expressed in the frame $(\mathbf{x}', \mathbf{y}', \mathbf{z}')$ into the coordinates expressed in frame $(\mathbf{x}, \ \mathbf{y}, \mathbf{z})$.

Inverse transformation: $\mathbf{p}' = \mathbf{R}^T\mathbf{p}$

Composition of rotation matrices

Let us consider three frames (denoted with 0, 1 and 2) with a common origin.
We denote with:

$\mathbf{R}_i^j$ the **rotation matrix** of frame i with respect to frame j

Thus:

$$\mathbf{R}_i^j = \left(\mathbf{R}_j^i\right)^{-1} = \left(\mathbf{R}_j^i\right)^T$$

The coordinates of the same point in the three frames can be expressed in different ways:

$$\mathbf{p}^1 = \mathbf{R}_2^1\mathbf{p}^2 \quad \mathbf{p}^0 = \mathbf{R}_1^0\mathbf{p}^1 \quad \mathbf{p}^0 = \mathbf{R}_2^0\mathbf{p}^2 \quad \Rightarrow \quad \mathbf{R}_2^0 = \mathbf{R}_1^0\mathbf{R}_2^1$$

Rotations can then be obtained by **composing partial rotations**.
Partial rotation matrices are multiplied **from left to right**.

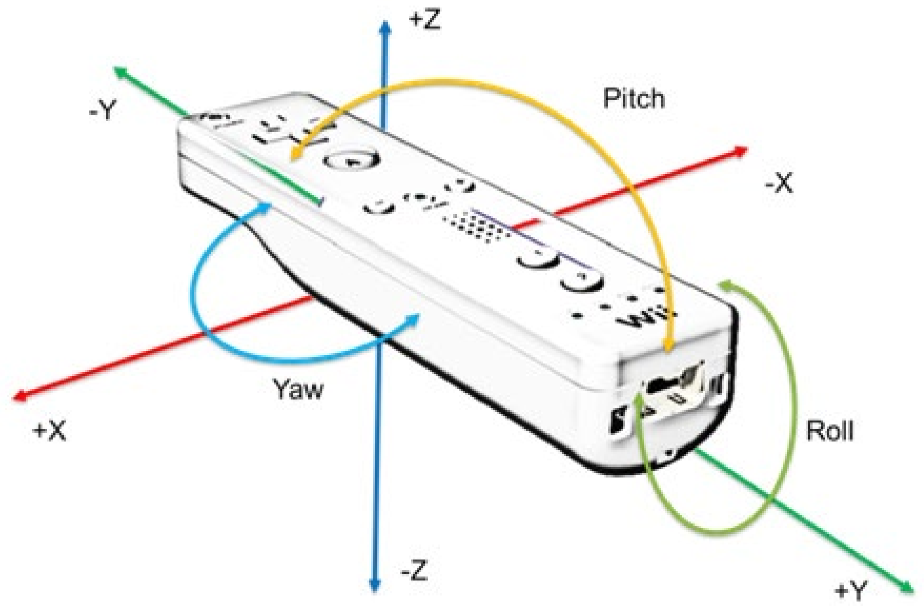
Minimal representation of the orientation

A rotation matrix represents the orientation of a frame with respect to another one by means of 9 parameters, among which 6 constraints exist.

In a **minimal representation** the orientation is described by means of 3 independent parameters.

Possible representations are:

- Euler angles (3 parameters)
- roll-pitch-yaw angles (3 parameters)
- axis/angle (4 parameters)
- quaternions (4 parameters)



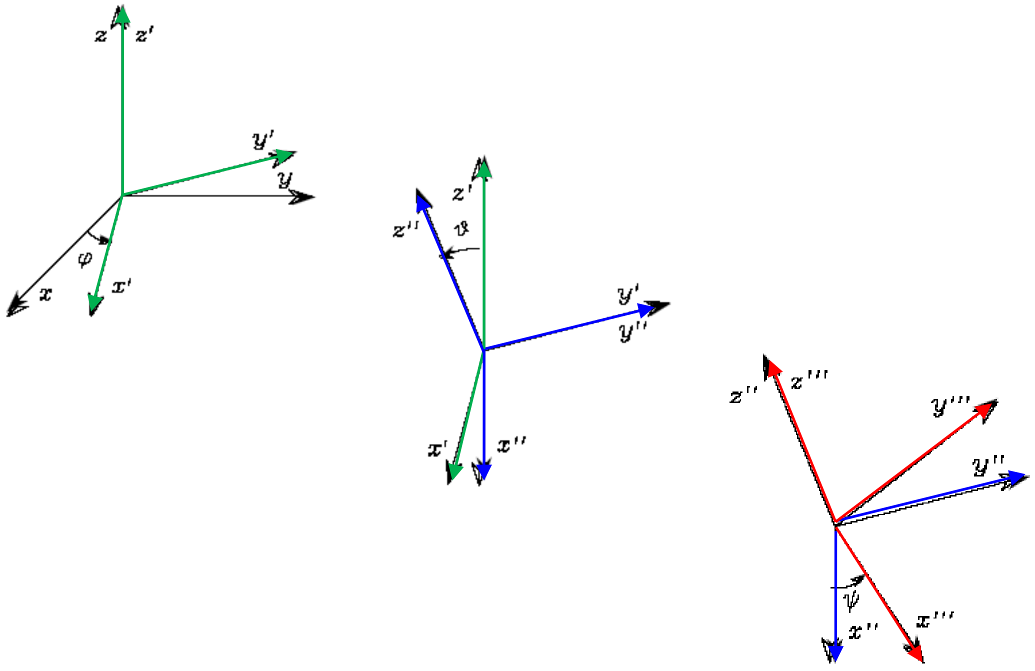
ZYZ Euler angles

With **ZYZ Euler angles** the sequence is composed as:

I) Rotation around Z (angle φ)

II) Rotation around Y' (angle ϑ)

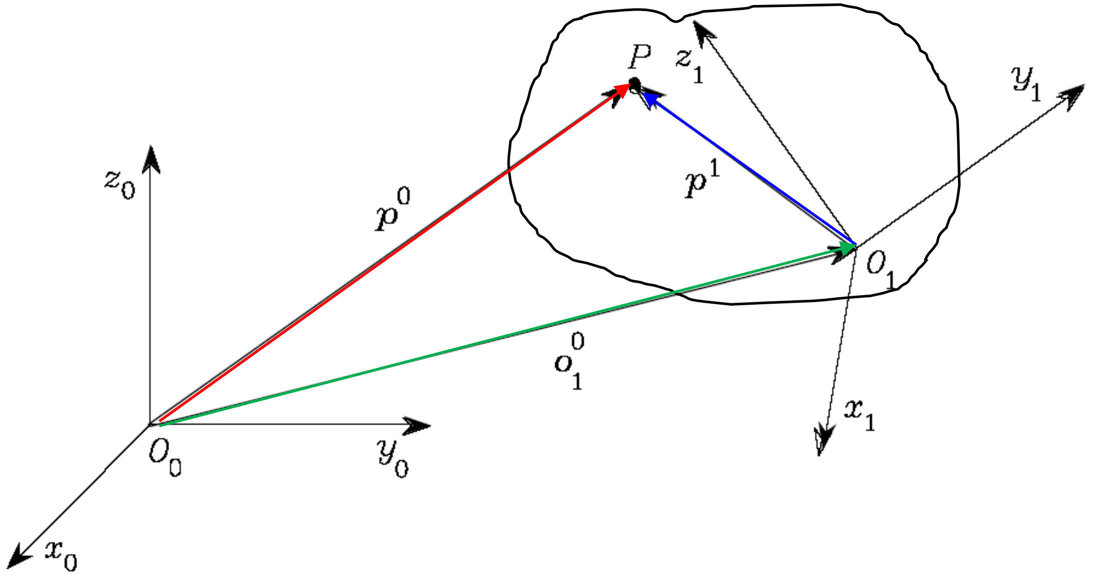
III) Rotation around Z'' (angle ψ)



$$\mathbf{R}=\mathbf{R}_z(\varphi)\mathbf{R}_{y'}(\vartheta)\mathbf{R}_{z''}(\psi)=\begin{bmatrix} c_\varphi c_\vartheta c_\psi - s_\varphi s_\psi & -c_\varphi c_\vartheta s_\psi - s_\varphi c_\psi & c_\varphi s_\vartheta \\ s_\varphi c_\vartheta c_\psi + c_\varphi s_\psi & -s_\varphi c_\vartheta s_\psi + c_\varphi c_\psi & s_\varphi s_\vartheta \\ -s_\vartheta c_\psi & s_\vartheta s_\psi & c_\vartheta \end{bmatrix}$$

$$c_\vartheta = \cos(\vartheta)$$
$$s_\vartheta = \sin(\vartheta)$$

Homogeneous representation



How can we express coordinates of point P in frame 0, based on its coordinates in frame 1?

$$\mathbf{p}^0 = \mathbf{o}_1^0 + \mathbf{R}_1^0 \mathbf{p}^1$$

Rotation matrix of frame 1 w.r.t. frame 0

Inverse transform:

$$\mathbf{p}^1 = -\mathbf{R}_0^1 \mathbf{o}_1^0 + \mathbf{R}_0^1 \mathbf{p}^0$$

In order to represent in a **compact form** these transformations, it is advisable to introduce a 4-dim vector:

$$\tilde{\mathbf{p}} = \begin{bmatrix} w\mathbf{p} \\ w \end{bmatrix}$$ **Homogeneous representation**

w is a scale factor which is always set to 1 in robotics (it is used in computer graphics)

Homogeneous transformations

We now introduce the **homogeneous transformation matrix** (size 4×4): $\mathbf{A}_1^0 = \begin{bmatrix} \mathbf{R}_1^0 & \mathbf{o}_1^0 \\ 0^T & 1 \end{bmatrix}$

The relationship:

$$\mathbf{p}^0 = \mathbf{o}_1^0 + \mathbf{R}_1^0 \mathbf{p}^1$$

can be expressed, in terms of homogeneous coordinates, as :

$$\tilde{\mathbf{p}}^0 = \mathbf{A}_1^0 \tilde{\mathbf{p}}^1$$

$\mathbf{A}_1^0$ relates the description (position/orientation) of a point on frame 1 with the description in frame 0.

The inverse transformation is:

$$\tilde{\mathbf{p}}^1 = \mathbf{A}_0^1 \tilde{\mathbf{p}}^0 = (\mathbf{A}_1^0)^{-1} \tilde{\mathbf{p}}^0 \qquad \mathbf{A}_0^1 = \begin{bmatrix} \mathbf{R}_0^1 & -\mathbf{R}_0^1 \mathbf{o}_1^0 \\ 0^T & 1 \end{bmatrix}$$

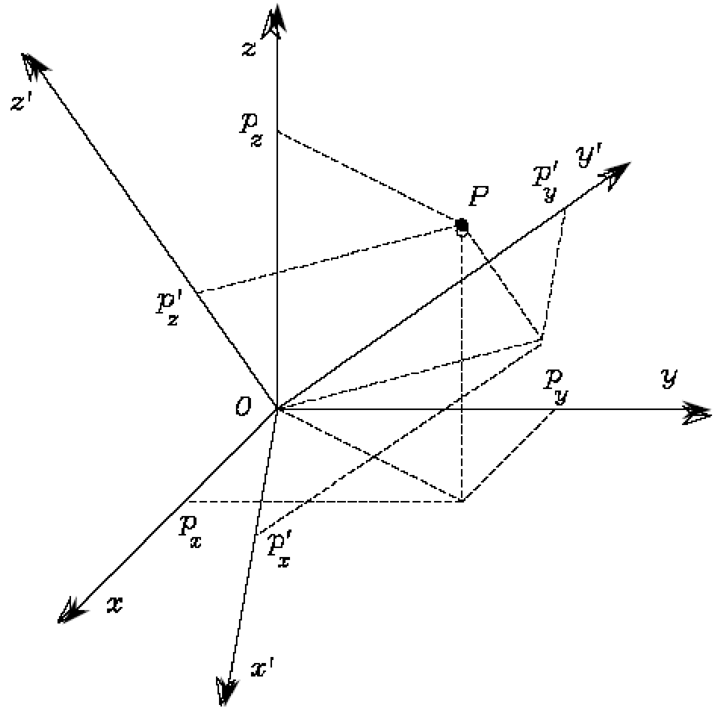
A is **not orthogonal**

Composing several transformations: $\tilde{\mathbf{p}}^0 = \mathbf{A}_1^0 \mathbf{A}_2^1 \dots \mathbf{A}_n^{n-1} \tilde{\mathbf{p}}^n$

Time dependent rotations

Suppose now that **rotation** of one frame with respect to the second one **changes with time**. Let us consider a point P attached to the rotating frame and expressed with the constant vector $\mathbf{p}'$. The coordinates of the same point in the stationary frame are:

$$\mathbf{p}(t) = \mathbf{R}(t)\mathbf{p}'$$



Take now the derivative with respect to time:

$$\dot{\mathbf{p}}(t) = \dot{\mathbf{R}}(t)\mathbf{p}'$$

How can we express the derivative of a rotation matrix?

Derivative of a rotation matrix

Since the rotation matrix is orthogonal, we have:

$$\mathbf{R}(t)\mathbf{R}^T(t) = \mathbf{I} \quad \Rightarrow \quad \dot{\mathbf{R}}(t)\mathbf{R}^T(t) + \mathbf{R}(t)\dot{\mathbf{R}}^T(t) = \mathbf{0}$$

If we define the new matrix:

$$\mathbf{S}(t) = \dot{\mathbf{R}}(t)\mathbf{R}^T(t)$$

It turns out that: $\mathbf{S}(t) + \mathbf{S}^T(t) = \mathbf{0}$ which means that matrix $\mathbf{S}$ is **skew symmetric**.

Matrix $\mathbf{S}$ then takes the following form:

$$\omega = \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix}, \quad \mathbf{S}(\omega) = \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix}$$

We conclude that the **derivative of a rotation matrix** is given by: $\dot{\mathbf{R}}(t) = \mathbf{S}\big(\omega(\mathbf{t})\big)\mathbf{R}(t)$

Relation with the angular velocity vector

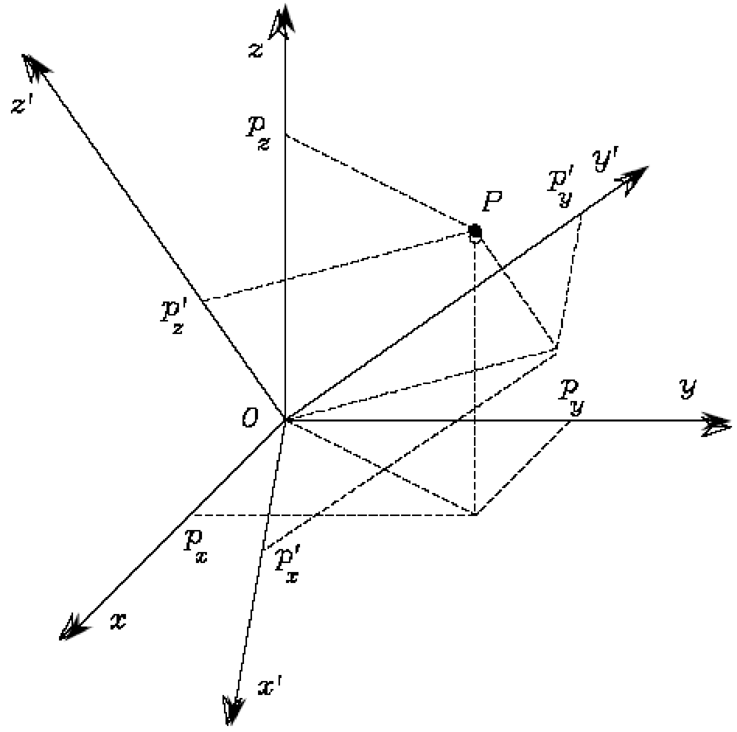
The derivative of vector $\mathbf{p}(t)$ can thus be expressed as :

$$\dot{\mathbf{p}}(t) = \dot{\mathbf{R}}(t)\mathbf{p}' = \mathbf{S}\big(\omega(\mathbf{t})\big)\mathbf{R}(t)\mathbf{p}' = \mathbf{S}\big(\omega(\mathbf{t})\big)\mathbf{p}(t)$$

On the other hand the same vector denotes the velocity of point P in the stationary frame:

$$\dot{\mathbf{p}}(t) = \omega(t) \times \mathbf{R}(t)\mathbf{p}' = \omega(t) \times \mathbf{p}(t)$$

- $\boldsymbol{\omega}$ is the **angular velocity vector** of the rotating frame
- symbol $\times$ denotes **cross product**

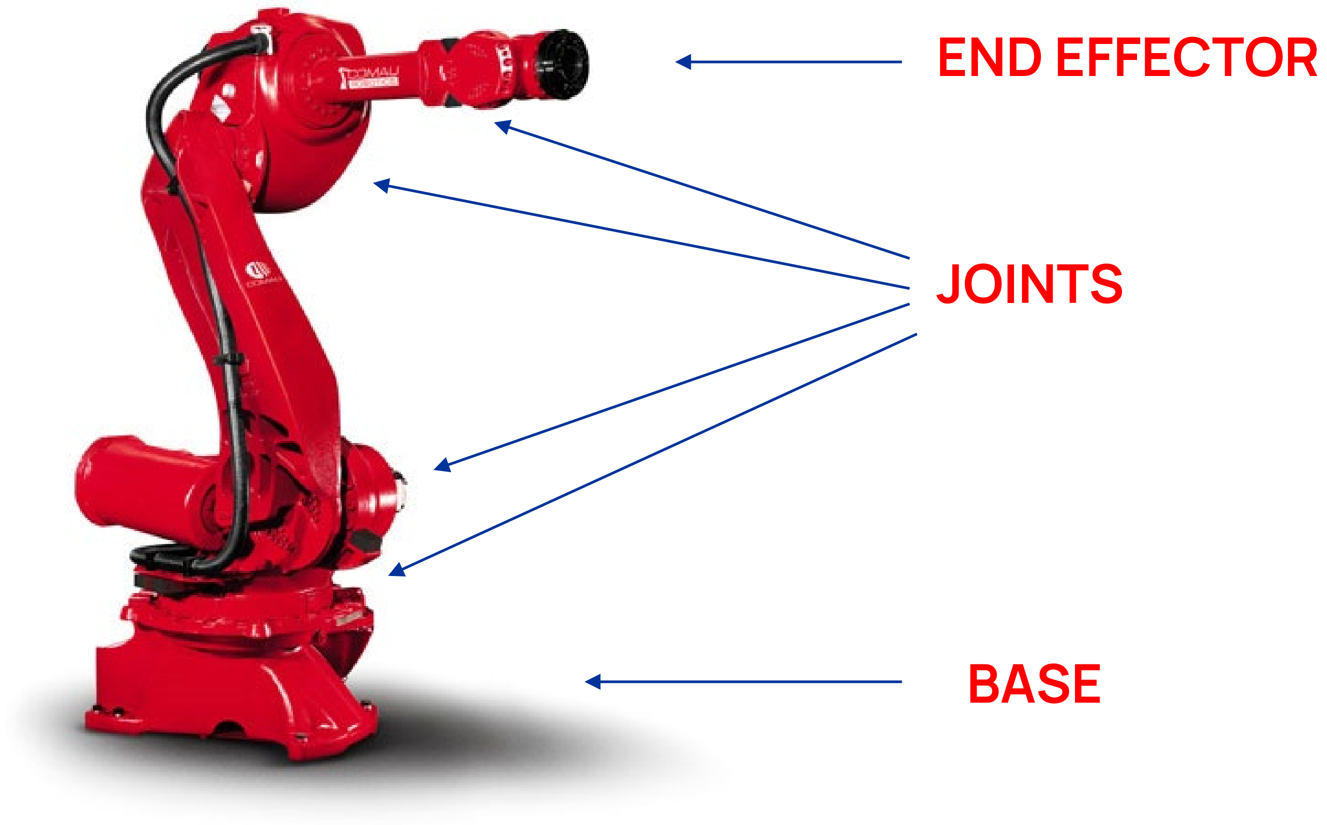


Thus the skew symmetric matrix $\mathbf{S}$ can be interpreted as the **operator** that computes the cross product.

Direct kinematics and the Denavit-Hartenberg method

02

How does all this relate to the robot?

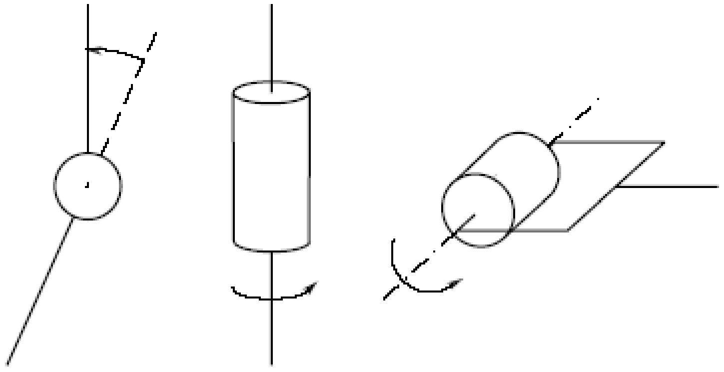


How does all this relate to the robot?

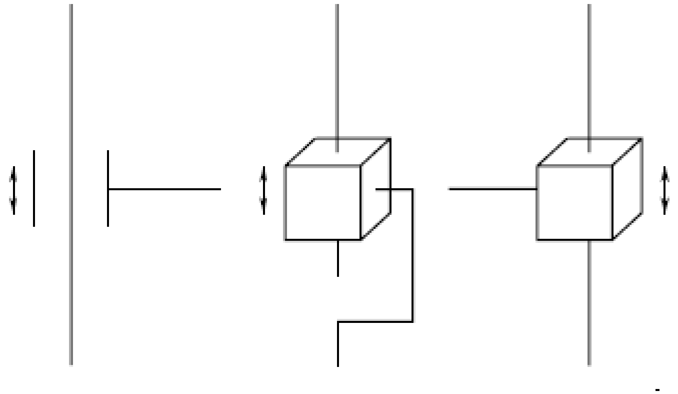
Each joint allows for one (and only one) *degree of freedom* between two links. We call **joint variable** the coordinate associated to such degree of freedom, and then we introduce the vector of joint variables:

$$\mathbf{q} = \begin{bmatrix} q_1 \\ \vdots \\ q_n \end{bmatrix}$$

Schematic draws of the joints:



ROTATIONAL JOINTS



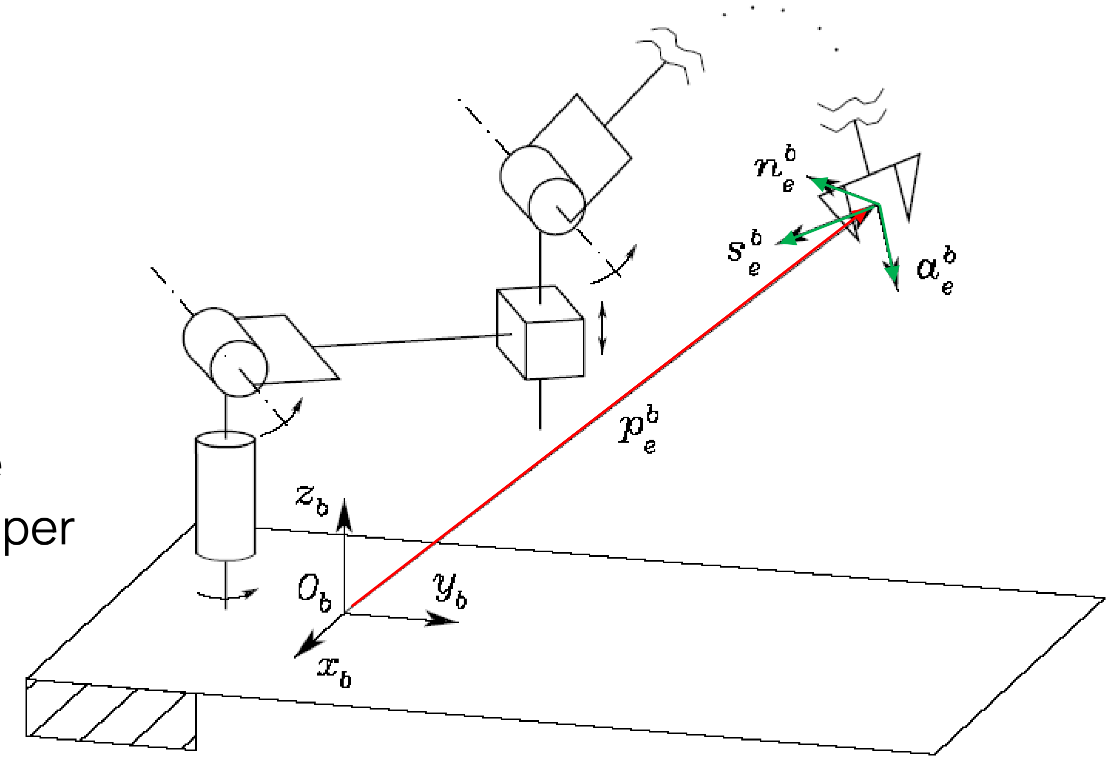
PRISMATIC JOINTS

Base frame and tool frame

Let us define a frame attached to the **base** and a frame attached to the **tool**.

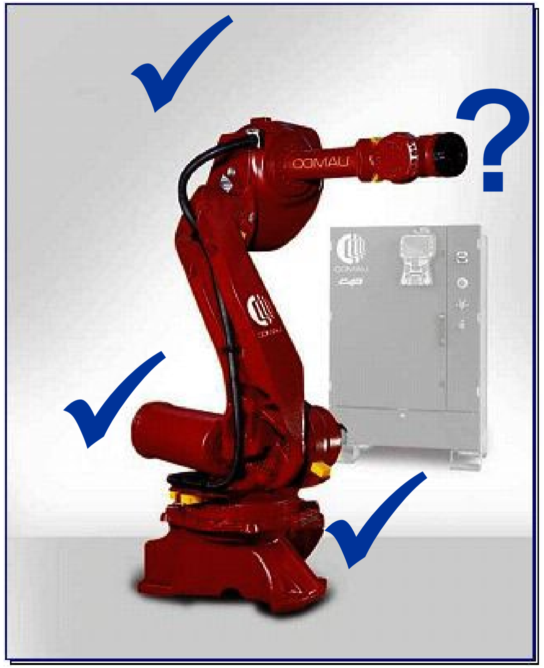
The tool frame is defined by means of **three unit vectors**:

$\mathbf{a}_e$ (approach): approach direction towards the work-piece
$\mathbf{s}_e$ (sliding): orthogonal to $\mathbf{a}_e$ in the sliding plane of the gripper
$\mathbf{n}_e$ (normal): orthogonal to both the other ones



$\mathbf{p}_e$ points to the origin of the tool frame (central point of the gripper).

Direct kinematics



The **direct kinematics problem** is to find position and orientation of the tool frame w.r.t. the base frame, as a function of the joint variables.

Direct kinematics

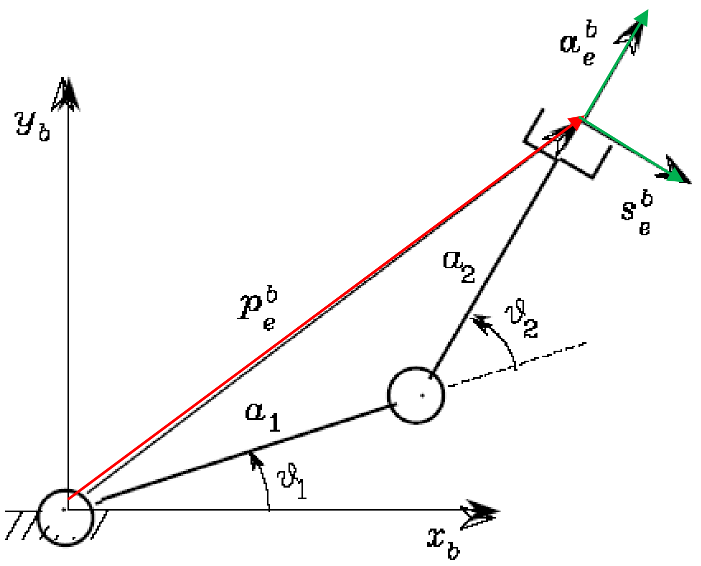
The **direct kinematic equation** can be expressed through the homogeneous transformation matrix of the tool frame with respect to the base frame.

$$\mathbf{T}_e^b(\mathbf{q}) = \begin{bmatrix} \mathbf{n}_e^b(\mathbf{q}) & \mathbf{s}_e^b(\mathbf{q}) & \mathbf{a}_e^b(\mathbf{q}) & \mathbf{p}_e^b(\mathbf{q}) \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

(homogeneous transformation matrix)

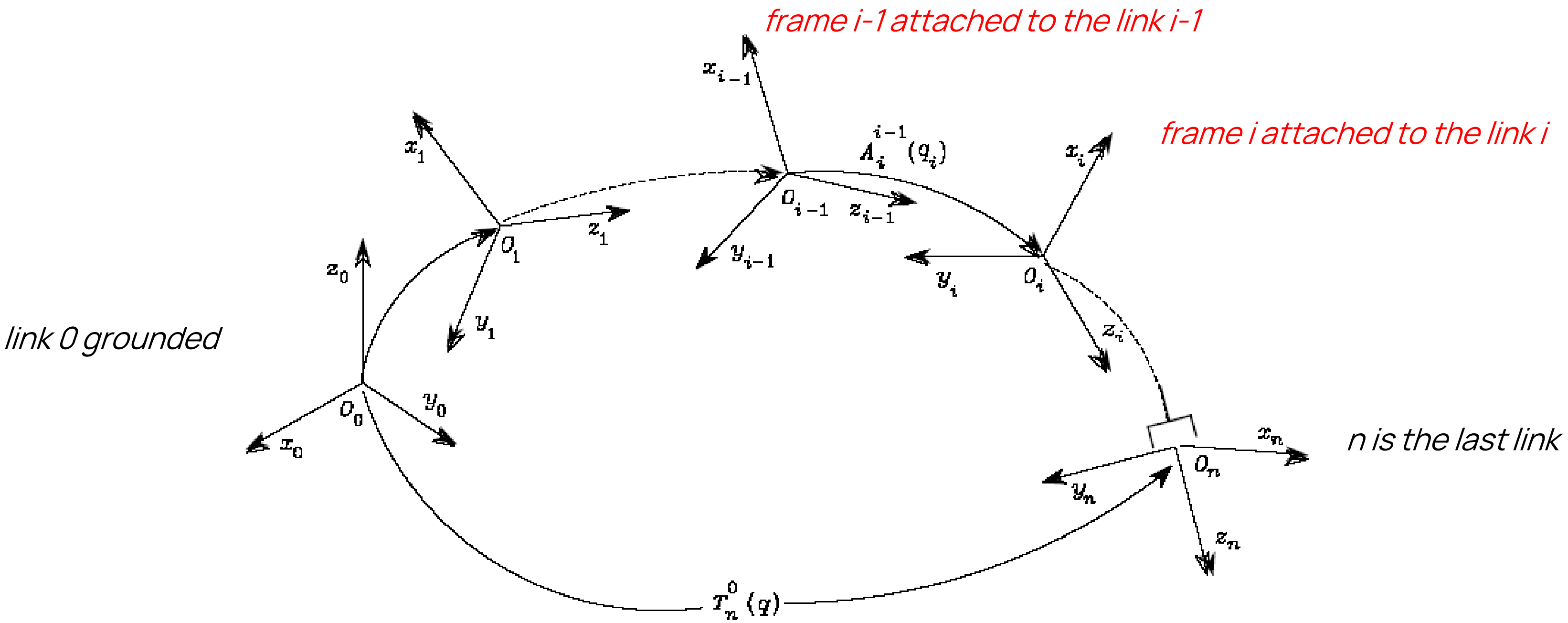
Example: planar two-link manipulator

$$\mathbf{T}_e^b(\mathbf{q}) = \begin{bmatrix} 0 & s_{12} & c_{12} & a_1 c_1 + a_2 c_{12} \\ 0 & -c_{12} & s_{12} & a_1 s_1 + a_2 s_{12} \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$



Direct kinematics

To proceed in a systematic way in the computation of the direct kinematics, a frame should be attached to each link:



$\mathbf{A}_i^{i-1}(q_i)$

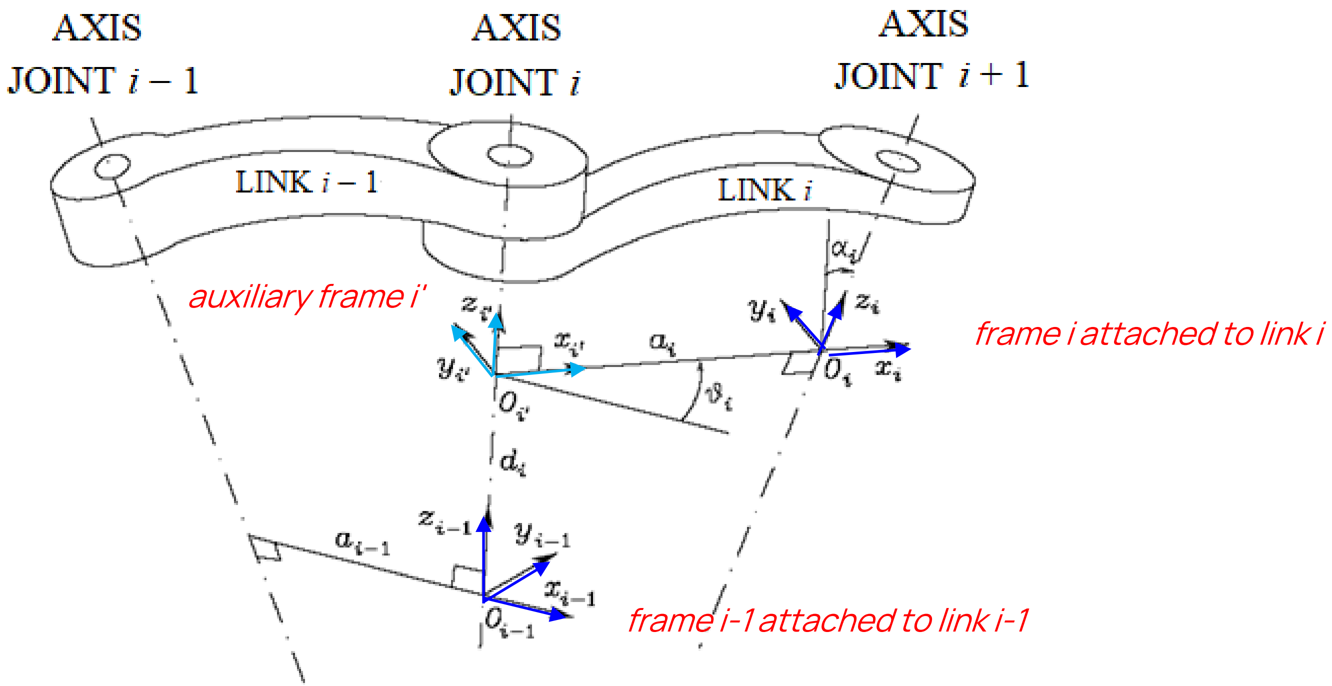
- It is the homogeneous transformation matrix of frame i with respect to frame $i-1$
- It only depends on the joint coordinate q_i
- **How to place the reference frames?**

Proceeding iteratively:

$$\mathbf{T}_n^0(\mathbf{q}) = \mathbf{A}_1^0(q_1)\mathbf{A}_2^1(q_2)\dots\mathbf{A}_n^{n-1}(q_n) \qquad \mathbf{T}_e^b(\mathbf{q}) = \mathbf{T}_0^b\mathbf{T}_n^0(\mathbf{q})\mathbf{T}_e^n$$

Denavit-Hartenberg convention

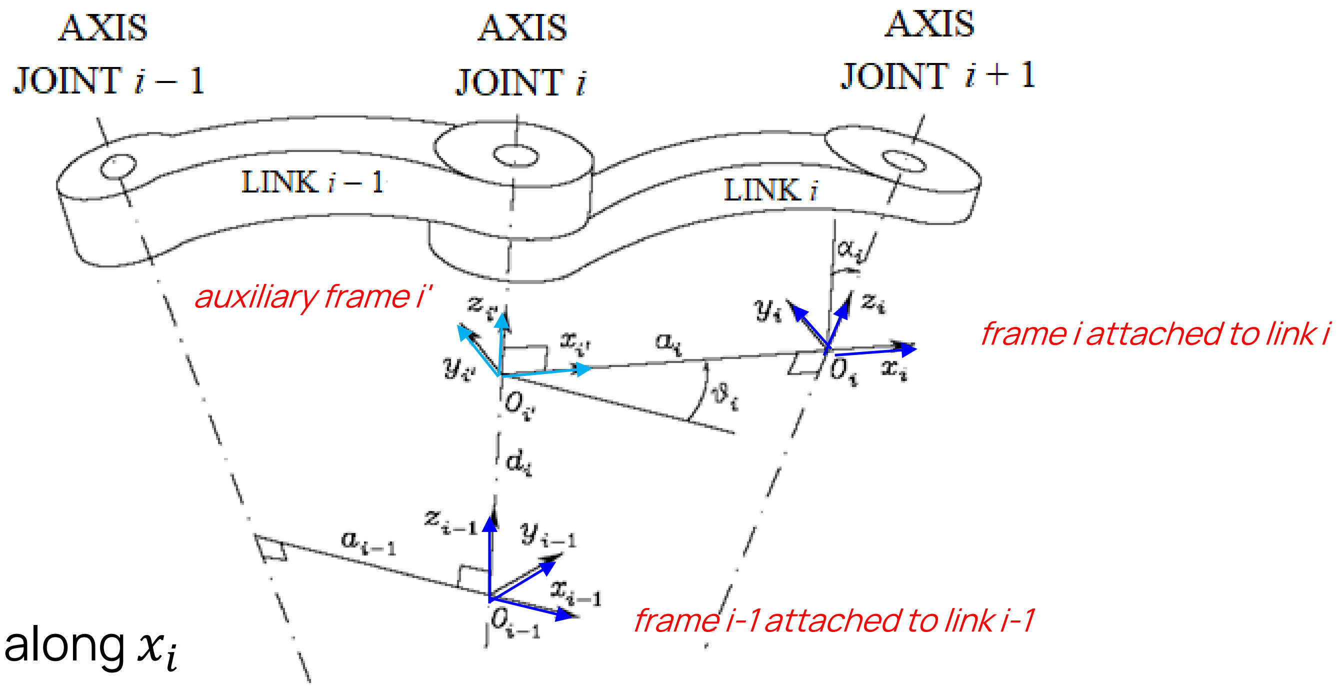
It is a **convention** for the selection of the frames attached to each link.



- z_i lies along the axis of joint $i+1$
- O_i is at the intersection of z_i axis with the common normal to axes z_i and z_{i-1}; we denote with O_i' the intersection of this common normal with axis z_{i-1}
- x_i is aligned with the common normal to axes z_i and z_{i-1}, with positive orientation from joint i to joint $i+1$
- y_i completes a right-handed frame

Denavit-Hartenberg parameters

In order to define a frame w.r.t. to the preceding one, **4 parameters** are needed.



- a_i distance of O_i from O_i' measured along x_i
- d_i coordinate on z_{i-1} of O_i'
- α_i angle around axis x_i between axis z_{i-1} and axis z_i computed as positive counter clockwise
- ϑ_i angle around axis z_{i-1} between axis x_{i-1} and axis x_i computed as positive counter clockwise

a_i and α_i are always constant, either ϑ_i or d_i is varying

Denavit-Hartenberg method illustrated

https://www.youtube.com/watch?v=rA9tm0gTln8

Homogeneous transformation matrix

How to build the **transformation matrix** from frame $i-1$ to frame i:

I) In order to superimpose frame $i-1$ to frame i' we translate the frame along axis z_{i-1} by a length d_i rotating by an angle ϑ_i around z_{i-1} :

$$\mathbf{A}_{i'}^{i-1} = \begin{bmatrix} c_{\vartheta_i} & -s_{\vartheta_i} & 0 & 0 \\ s_{\vartheta_i} & c_{\vartheta_i} & 0 & 0 \\ 0 & 0 & 1 & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

II) In order to superimpose frame i' to frame i we translate the frame along axis x_i' by a length a_i, rotating of an angle α_i around x_i' :

$$\mathbf{A}_{i}^{i'} = \begin{bmatrix} 1 & 0 & 0 & a_i \\ 0 & c_{\alpha_i} & -s_{\alpha_i} & 0 \\ 0 & s_{\alpha_i} & c_{\alpha_i} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\mathbf{A}_{i}^{i-1}(q_i) = \mathbf{A}_{i'}^{i-1}\mathbf{A}_{i}^{i'} = \begin{bmatrix} c_{\vartheta_i} & -s_{\vartheta_i}c_{\alpha_i} & s_{\vartheta_i}s_{\alpha_i} & a_i c_{\vartheta_i} \\ s_{\vartheta_i} & c_{\vartheta_i}c_{\alpha_i} & -c_{\vartheta_i}s_{\alpha_i} & a_i s_{\vartheta_i} \\ 0 & s_{\alpha_i} & c_{\alpha_i} & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Joint space and operational space

The **joint space** is defined by the vector of joint variables:

$$\mathbf{q} = \begin{bmatrix} q_1 \\ \vdots \\ q_n \end{bmatrix} \qquad \begin{aligned} q_i &= \vartheta_i \text{ (rotating joint)} \\ q_i &= d_i \text{ (prismatic joint)} \end{aligned}$$

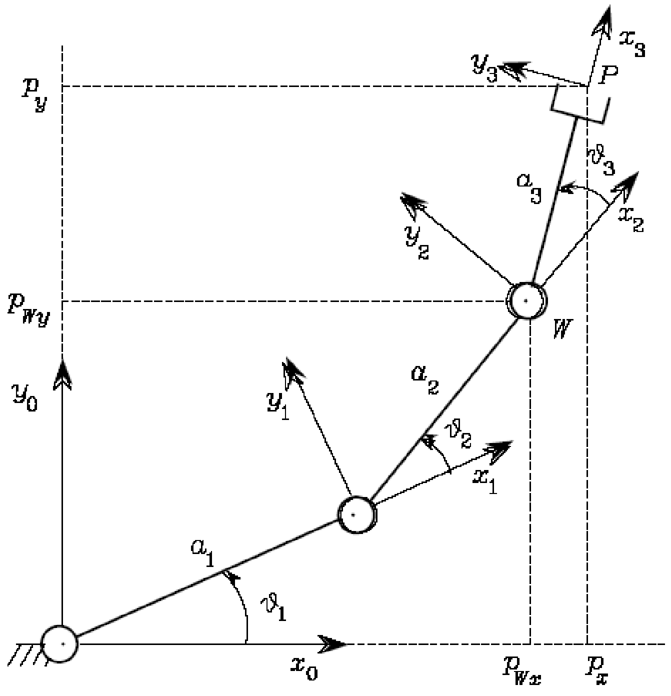
The **operational space** is the space where the task that the manipulator has to accomplish is specified. It is defined by the posture $\mathbf{x}$:

$$\mathbf{x} = \begin{bmatrix} \mathbf{p} \\ \phi \end{bmatrix} \qquad \begin{aligned} &\mathbf{p} \text{ (position)} \\ &\phi \text{ (minimal representation of the orientation)} \end{aligned}$$

m components

Direct kinematic relation: $\mathbf{x} = \mathbf{k}(\mathbf{q})$

Three d.o.f. planar manipulator



	a_i	α_i	d_i	ϑ_i
1	a_1	0	0	ϑ_1
2	a_2	0	0	ϑ_2
3	a_3	0	0	ϑ_3

red: joint variables

$$\mathbf{A}_1^0 = \begin{bmatrix} c_1 & -s_1 & 0 & a_1c_1 \\ s_1 & c_1 & 0 & a_1s_1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad \mathbf{A}_2^1 = \begin{bmatrix} c_2 & -s_2 & 0 & a_2c_2 \\ s_2 & c_2 & 0 & a_2s_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\mathbf{A}_3^2 = \begin{bmatrix} c_3 & -s_3 & 0 & a_3c_3 \\ s_3 & c_3 & 0 & a_3s_3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$c_i = \cos(\vartheta_i)$
$s_i = \sin(\vartheta_i)$

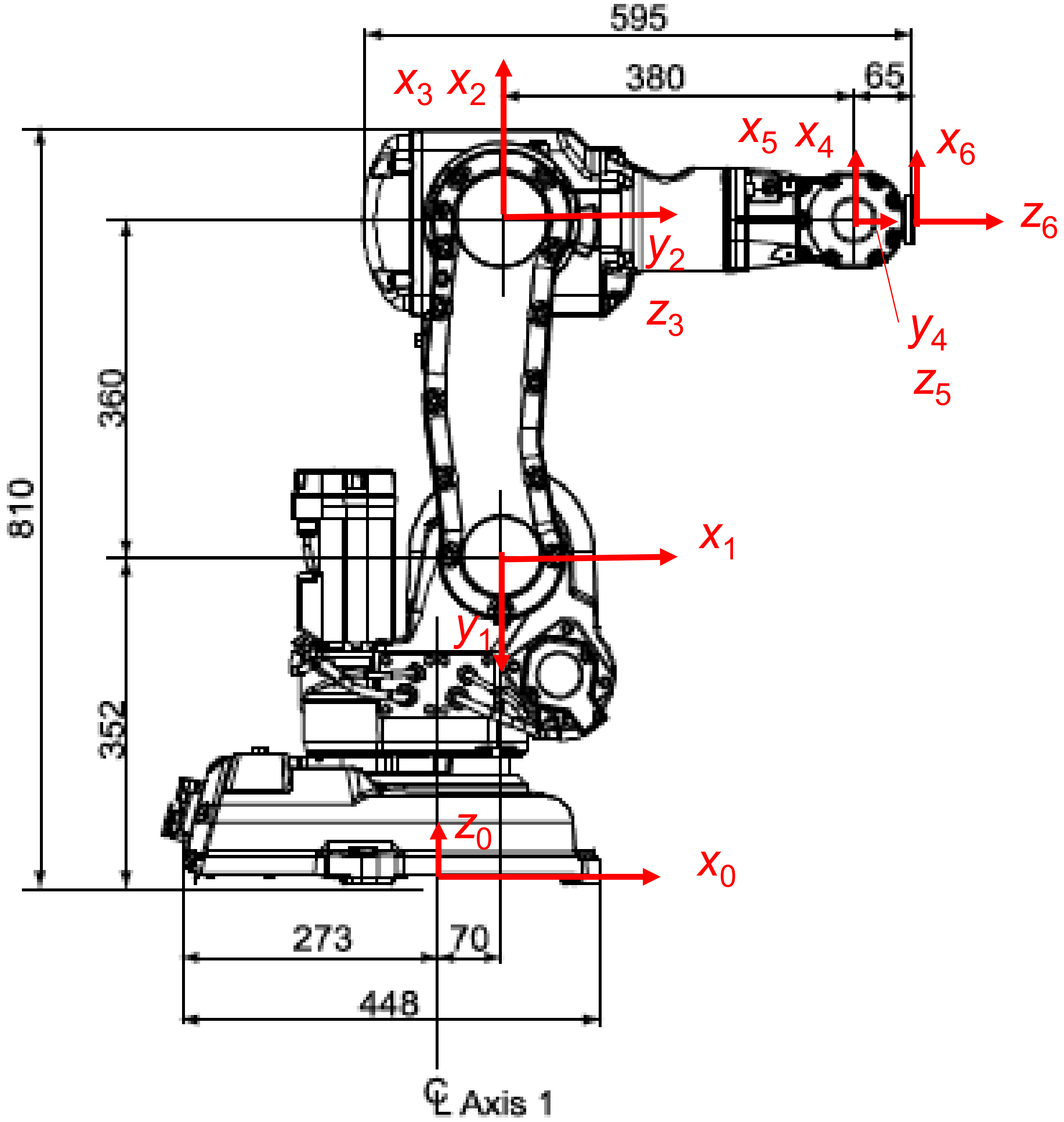
$$\mathbf{T}_3^0 = \mathbf{A}_1^0\mathbf{A}_2^1\mathbf{A}_3^2 = \begin{bmatrix} c_{123} & -s_{123} & 0 & a_1c_1 + a_2c_{12} + a_3c_{123} \\ s_{123} & c_{123} & 0 & a_1s_1 + a_2s_{12} + a_3s_{123} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

We can define the orientation with the angle φ formed by the end effector (vector x_3) with axis x_0

$$\mathbf{x} = \begin{bmatrix} p_x \\ p_y \\ \varphi \end{bmatrix} = \mathbf{k}(\mathbf{q}) = \begin{bmatrix} a_1c_1 + a_2c_{12} + a_3c_{123} \\ a_1s_1 + a_2s_{12} + a_3s_{123} \\ \vartheta_1 + \vartheta_2 + \vartheta_3 \end{bmatrix}$$

$c_{12} = \cos(\vartheta_1 + \vartheta_2)$
$s_{12} = \sin(\vartheta_1 + \vartheta_2)$
$c_{123} = \cos(\vartheta_1 + \vartheta_2 + \vartheta_3)$
$s_{123} = \sin(\vartheta_1 + \vartheta_2 + \vartheta_3)$

A six d.o.f. robot

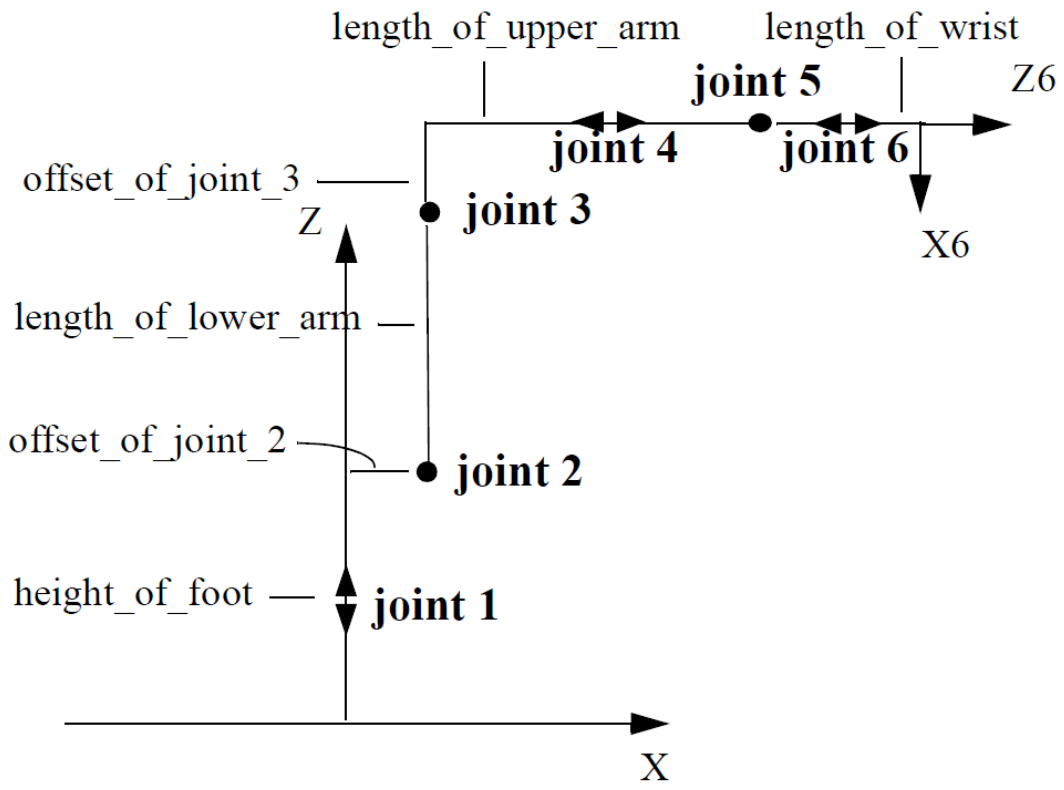


	a_i	α_i	d_i	ϑ_i
1	0.07	$-\frac{\pi}{2}$	0.352	ϑ_1
2	0.36	0	0	ϑ_2
3	0	$-\frac{\pi}{2}$	0	ϑ_3
4	0	$\frac{\pi}{2}$	0.38	ϑ_4
5	0	$-\frac{\pi}{2}$	0	ϑ_5
6	0	0	0.065	ϑ_6

Kinematic model from a user's perspective

The **robot manuals** usually do not give the kinematic model in terms of Denavit-Hartenberg parameters, rather in a simplified way.

Here is an example:



Source: ABB

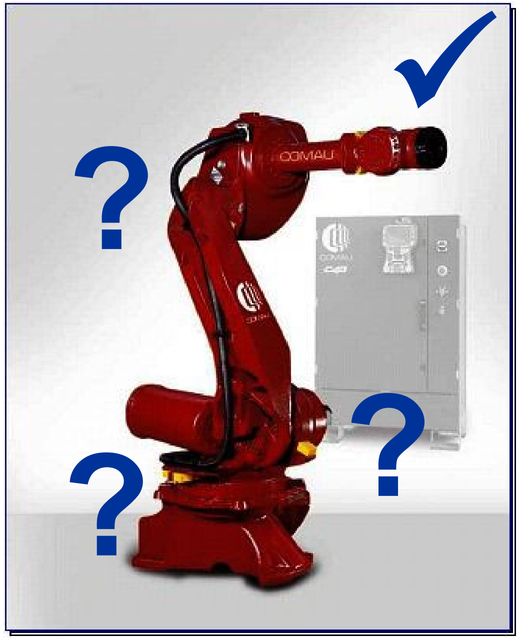
A description like this is specific to this robot and can be understood only with an accompanying sketch.

Denavit-Hartenberg representation is universal and can be understood as it is.

Inverse kinematics

03

Inverse kinematics



The **inverse kinematics problem** is to find joint variables given position and orientation of the tool frame w.r.t. the base frame.

Inverse kinematics

$\mathbf{T} \Rightarrow \mathbf{q}$

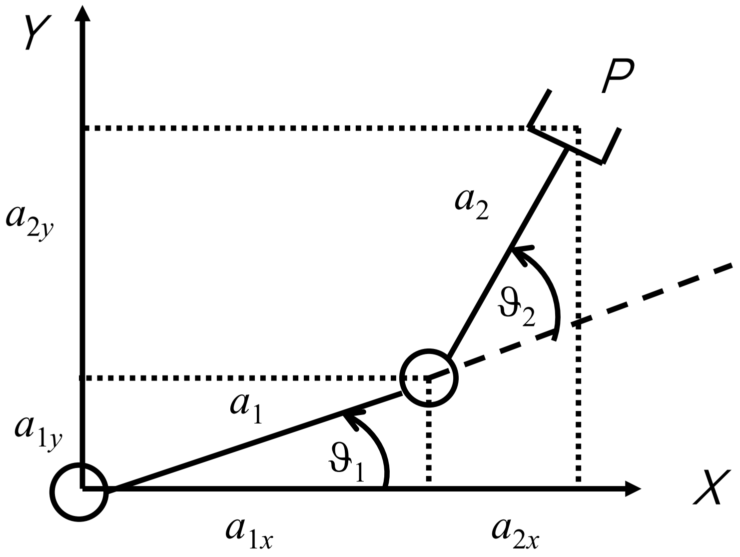
$\mathbf{x} \Rightarrow \mathbf{q}$

Given position and orientation of the tool frame, **find the corresponding joint variables**.

- The problem may admit no solutions (if position and orientation do not belong to the workspace of the manipulator)
- The analytical solution (in closed form) may not exist. In this case numerical techniques are used
- Multiple or an infinite number of solutions might exist

In general the solution is found without a systematic procedure, rather relying on intuition in manipulating the equations.

Two d.o.f. planar manipulator



$$p_x = a_{1x} + a_{2x} = a_1\cos(\vartheta_1) + a_2\cos(\vartheta_1 + \vartheta_2)$$
$$p_y = a_{1y} + a_{2y} = a_1\sin(\vartheta_1) + a_2\sin(\vartheta_1 + \vartheta_2)$$

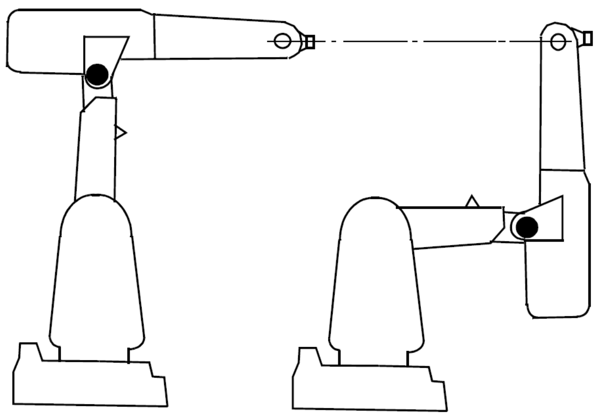
Squaring and summing:

$$c_2 = \frac{p_x^2 + p_y^2 - a_1^2 - a_2^2}{2a_1a_2}$$
$$s_2 = \pm\sqrt{1 - c_2^2} \quad \Rightarrow \quad \vartheta_2 = \mathrm{Atan2}(s_2, c_2)$$

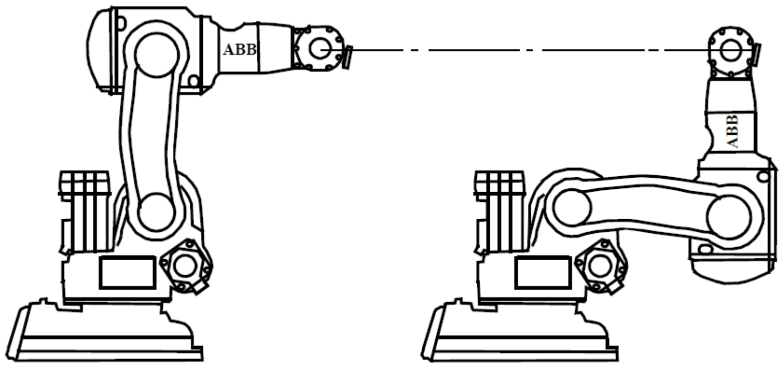
2 solutions

$$c_1 = \frac{(a_1 + a_2c_2)p_x + a_2s_2p_y}{p_x^2 + p_y^2}$$
$$s_1 = \frac{(a_1 + a_2c_2)p_y - a_2s_2p_x}{p_x^2 + p_y^2} \quad \Rightarrow \quad \vartheta_1 = \mathrm{Atan2}(s_1, c_1)$$

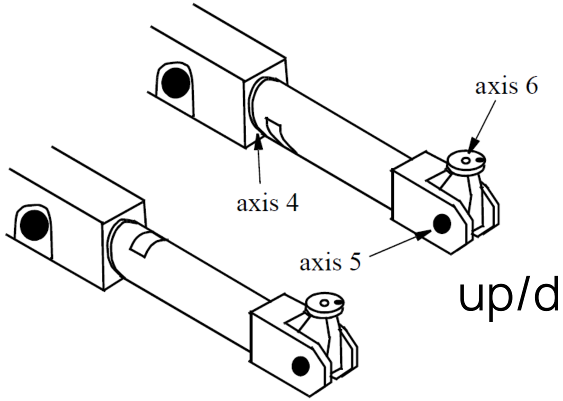
Anthropomorphic manipulator



right/left shoulder



up/down elbow



up/down wrist

Eight admissible configurations exist

Source: ABB

Differential kinematics and robot singularities

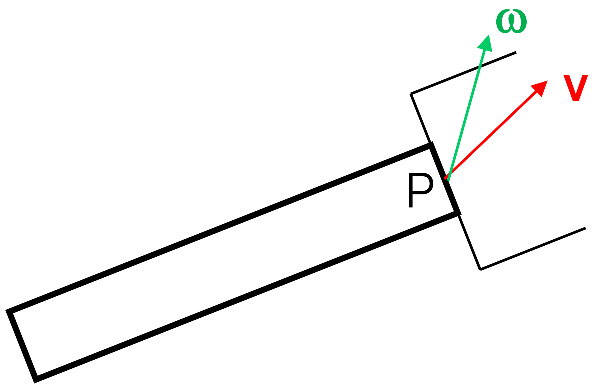
04

Differential kinematics: geometrical Jacobian

Let's introduce now the linear velocity and the angular velocity of the tool frame (attached to the tool): $\mathbf{v}$ and ω.

The goal of **differential kinematics** is to express these velocities in terms of the joint velocities.

$$\mathbf{v} = \dot{\mathbf{p}} = \mathbf{J}_P(\mathbf{q})\dot{\mathbf{q}}$$
$$\omega = \mathbf{J}_O(\mathbf{q})\dot{\mathbf{q}}$$



In a compact form: $\begin{bmatrix}\mathbf{v}\\ \boldsymbol{\omega}\end{bmatrix} = \begin{bmatrix}\dot{\mathbf{p}}\\ \boldsymbol{\omega}\end{bmatrix} = \mathbf{J}(\mathbf{q})\dot{\mathbf{q}}$

The (6×n) matrix: $\mathbf{J}(\mathbf{q}) = \begin{bmatrix}\mathbf{J}_P(\mathbf{q})\\ \mathbf{J}_O(\mathbf{q})\end{bmatrix}$

is called **geometrical Jacobian** of the manipulator.

Computation of the geometrical Jacobian

We partition the Jacobian in n columns, each one in turn partitioned in two vectors:

$$\mathbf{J} = \begin{bmatrix} \mathbf{j}_{P1} & & \mathbf{j}_{Pn} \\ & \cdots & \\ \mathbf{j}_{O1} & & \mathbf{j}_{On} \end{bmatrix}$$

contribution of joint i to the linear velocity

We have:

$$\mathbf{v} = \dot{\mathbf{p}} = \mathbf{j}_{P1}(\mathbf{q})\dot{q}_1 + \mathbf{j}_{P2}(\mathbf{q})\dot{q}_2 + \cdots + \mathbf{j}_{Pi}(\mathbf{q})\dot{q}_i + \cdots + \mathbf{j}_{Pn}(\mathbf{q})\dot{q}_n$$

$$\boldsymbol{\omega} = \mathbf{j}_{O1}(\mathbf{q})\dot{q}_1 + \mathbf{j}_{O2}(\mathbf{q})\dot{q}_2 + \cdots + \mathbf{j}_{Oi}(\mathbf{q})\dot{q}_i + \cdots + \mathbf{j}_{On}(\mathbf{q})\dot{q}_n$$

contribution of joint i to the angular velocity

There is a superposition of effects, that can be used to compute the single contributions

Computation of the geometrical Jacobian

Angular velocity

Joint i **prismatic**: a prismatic joint does not give any contribution of angular velocity.

$$\dot{q}_i \mathbf{j}_{Oi} = 0 \quad \Rightarrow \quad \mathbf{j}_{Oi} = 0$$

Joint i **rotational**: a rotational joint gives a contribution of angular velocity directed as the axis of the joint

$$\dot{q}_i \mathbf{j}_{Oi} = \dot{\vartheta}_i \mathbf{z}_{i-1} \quad \Rightarrow \quad \mathbf{j}_{Oi} = \mathbf{z}_{i-1}$$

In making these considerations, we consider the joints downward in the kinematic chain as «frozen»

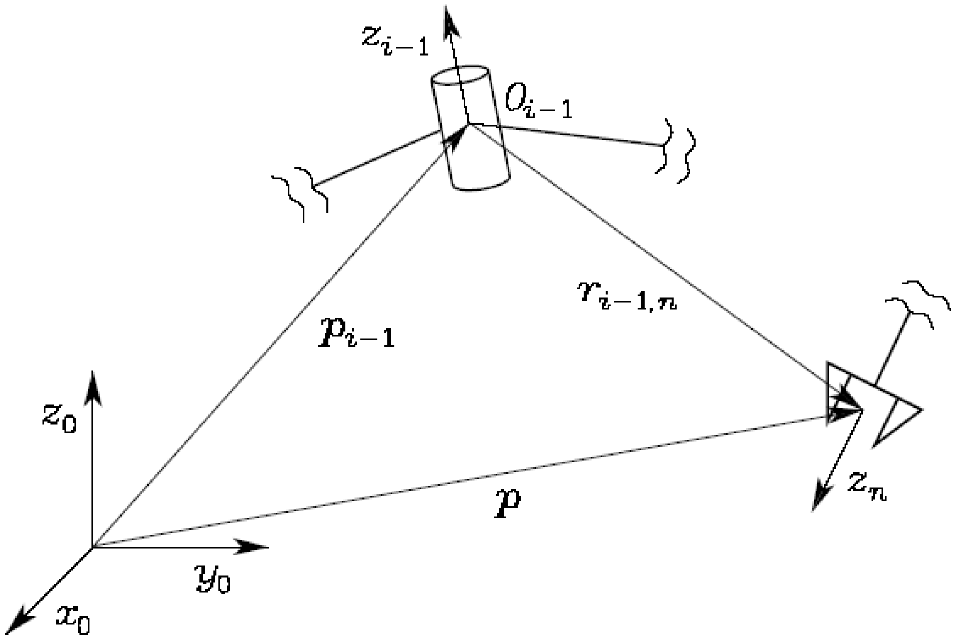
Computation of the geometrical Jacobian

Linear velocity

Joint i **prismatic**: a prismatic joint gives a contribution of linear velocity directed as the axis of the joint

$$\dot{q}_i \mathbf{j}_{Pi} = \dot{d}_i \mathbf{z}_{i-1} \quad \Rightarrow \quad \mathbf{j}_{Pi} = \mathbf{z}_{i-1}$$

Joint i **rotational**: a rotational joint gives a contribution of linear velocity that can be obtained with a cross product



$$\dot{q}_i \mathbf{j}_{Pi} = \dot{\vartheta}_i \mathbf{z}_{i-1} \times \mathbf{r}_{i-1,n} =$$
$$= \dot{\vartheta}_i \mathbf{z}_{i-1} \times (\mathbf{p} - \mathbf{p}_{i-1})$$
$$\Downarrow$$
$$\mathbf{j}_{Pi} = \mathbf{z}_{i-1} \times (\mathbf{p} - \mathbf{p}_{i-1})$$

Computation of the geometrical Jacobian

We can then compute the geometrical Jacobian column by column:

$$\begin{bmatrix} \mathbf{j}_{Pi} \\ \mathbf{j}_{Oi} \end{bmatrix} = \begin{cases} \begin{bmatrix} \mathbf{z}_{i-1} \\ \mathbf{0} \end{bmatrix} & \text{joint } i \text{ prismatic} \\ \begin{bmatrix} \mathbf{z}_{i-1} \times (\mathbf{p} - \mathbf{p}_{i-1}) \\ \mathbf{z}_{i-1} \end{bmatrix} & \text{joint } i \text{ rotational} \end{cases}$$

The matrices needed to compute these vectors can be determined through direct kinematics relations:

$$\mathbf{z}_{i-1} = \mathbf{R}_1^0(q_1) \dots \mathbf{R}_{i-1}^{i-2}(q_{i-1})\mathbf{z}_0$$

$$\tilde{\mathbf{p}} = \mathbf{A}_1^0(q_1) \dots \mathbf{A}_n^{n-1}(q_n)\tilde{\mathbf{p}}_0$$

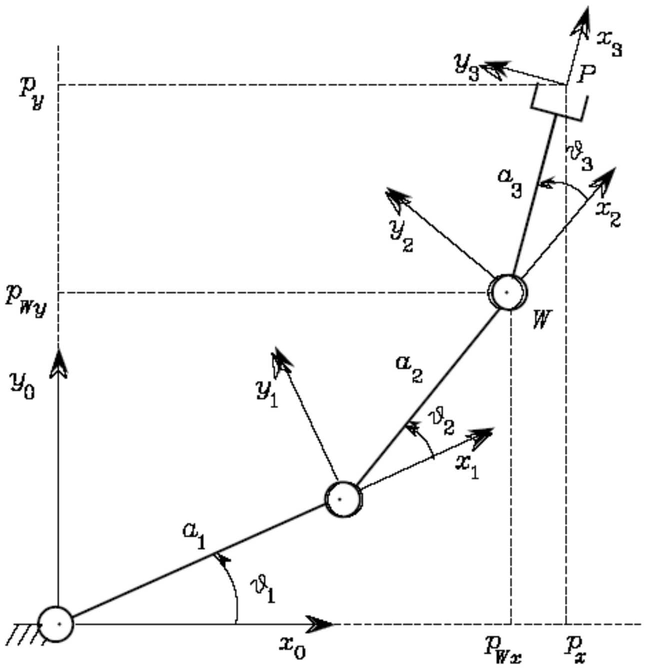
$$\tilde{\mathbf{p}}_{i-1} = \mathbf{A}_1^0(q_1) \dots \mathbf{A}_{i-1}^{i-2}(q_{i-1})\tilde{\mathbf{p}}_0$$

where:

$$\mathbf{z}_0 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}, \tilde{\mathbf{p}}_0 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}, \tilde{\mathbf{p}} = \begin{bmatrix} \mathbf{p} \\ 1 \end{bmatrix}$$

(homogeneous coordinates)

Computation of the geometrical Jacobian: example



$$\mathbf{J}(\mathbf{q}) = \begin{bmatrix} \mathbf{z}_0 \times (\mathbf{p}-\mathbf{p}_0) & \mathbf{z}_1 \times (\mathbf{p}-\mathbf{p}_1) & \mathbf{z}_2 \times (\mathbf{p}-\mathbf{p}_2) \\ \mathbf{z}_0 & \mathbf{z}_1 & \mathbf{z}_2 \end{bmatrix}$$

$$\mathbf{p}_0 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \quad \mathbf{p}_1 = \begin{bmatrix} a_1 c_1 \\ a_1 s_1 \\ 0 \end{bmatrix} \quad \mathbf{p}_2 = \begin{bmatrix} a_1 c_1 + a_2 c_{12} \\ a_1 s_1 + a_2 s_{12} \\ 0 \end{bmatrix}$$

$$\mathbf{p} = \begin{bmatrix} a_1 c_1 + a_2 c_{12} + a_3 c_{123} \\ a_1 s_1 + a_2 s_{12} + a_3 s_{123} \\ 0 \end{bmatrix} \quad \mathbf{z}_0 = \mathbf{z}_1 = \mathbf{z}_2 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

$$\mathbf{J} = \begin{bmatrix} -a_1 s_1 - a_2 s_{12} - a_3 s_{123} & -a_2 s_{12} - a_3 s_{123} & -a_3 s_{123} \\ a_1 c_1 + a_2 c_{12} + a_3 c_{123} & a_2 c_{12} + a_3 c_{123} & a_3 c_{123} \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix}$$

note:

$$\mathbf{a} \times \mathbf{b} = \begin{bmatrix} a_2 b_3 - a_3 b_2 \\ a_3 b_1 - a_1 b_3 \\ a_1 b_2 - a_2 b_1 \end{bmatrix}$$

Analytical Jacobian

Let's go back to the direct kinematic equation of a manipulator:

$$\mathbf{x} = \mathbf{k}(\mathbf{q}) = \begin{bmatrix} \mathbf{p}(\mathbf{q}) \\ \phi(\mathbf{q}) \end{bmatrix}$$

where ϕ is a minimal representation of the orientation. Differentiating w.r.t. time, we obtain:

$$\dot{\mathbf{x}} = \frac{\partial \mathbf{k}(\mathbf{q})}{\partial \mathbf{q}} \dot{\mathbf{q}} = \mathbf{J}_A(\mathbf{q})\dot{\mathbf{q}}$$

On the other hand:

$$\dot{\mathbf{x}} = \begin{bmatrix} \dot{\mathbf{p}} \\ \dot{\phi} \end{bmatrix} = \begin{bmatrix} (\partial \mathbf{p}(\mathbf{q})/\partial \mathbf{q})\dot{\mathbf{q}} \\ (\partial \phi(\mathbf{q})/\partial \mathbf{q})\dot{\mathbf{q}} \end{bmatrix} = \begin{bmatrix} \mathbf{J}_P(\mathbf{q}) \\ \mathbf{J}_\phi(\mathbf{q}) \end{bmatrix} \dot{\mathbf{q}}$$

Matrix: $\mathbf{J}_A(\mathbf{q}) = \begin{bmatrix} \mathbf{J}_P(\mathbf{q}) \\ \mathbf{J}_\phi(\mathbf{q}) \end{bmatrix}$ is called **analytical Jacobian** of the manipulator.

Analytical vs. geometrical Jacobian

The link between the angular velocity ω and the derivative of vector ϕ expressing the orientation is the following one:

$$\boldsymbol{\omega} = \mathbf{T}(\phi)\dot{\phi}$$

where $\mathbf{T}$ is a matrix that depends on the representation of the orientation:

$$\mathbf{T}(\phi) = \begin{bmatrix} 0 & -s_\phi & c_\phi s_\vartheta \\ 0 & c_\phi & s_\phi s_\vartheta \\ 1 & 0 & c_\vartheta \end{bmatrix}$$ (for the ZYZ Euler angles)

Let us thus express the velocity (linear and angular) of the tool frame in terms of the derivatives of $\mathbf{p}$ and ϕ :

$$\begin{bmatrix} \mathbf{v} \\ \boldsymbol{\omega} \end{bmatrix} = \begin{bmatrix} \dot{\mathbf{p}} \\ \boldsymbol{\omega} \end{bmatrix} = \begin{bmatrix} \dot{\mathbf{p}} \\ \mathbf{T}(\phi)\dot{\phi} \end{bmatrix} = \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{T}(\phi) \end{bmatrix} \begin{bmatrix} \dot{\mathbf{p}} \\ \dot{\phi} \end{bmatrix} = \mathbf{T}_A(\phi)\dot{\mathbf{x}} = \mathbf{T}_A(\phi)\mathbf{J}_A\dot{\mathbf{q}}$$

where: $\mathbf{T}_A(\phi) = \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{T}(\phi) \end{bmatrix}$

The **relation between analytical and geometrical Jacobian** follows: $\boxed{\mathbf{J} = \mathbf{T}_A(\varphi)\mathbf{J}_A}$

Kinematic singularities

The equation defining the geometrical Jacobian is:

$$\begin{bmatrix} \mathbf{v} \\ \boldsymbol{\omega} \end{bmatrix} = \begin{bmatrix} \dot{\mathbf{p}} \\ \boldsymbol{\omega} \end{bmatrix} = \mathbf{J}(\mathbf{q})\dot{\mathbf{q}}$$

The values of $\mathbf{q}$ for which matrix $\mathbf{J}$ is rank-deficient are called **kinematic singularities**. At a kinematic singularity we have:

1. Loss of mobility (it is not possible to impose arbitrary motion laws)
2. Possibility of infinite solutions to the kinematic inversion problem
3. High velocities in joint space (around the singularity)

The singularities may happen:

1. **At the borders** of the manipulator work-space
2. **Inside** the manipulator work-space

The latter are more problematic, since they can be incurred with trajectories planned in the operational space.

Kinematic singularities: example

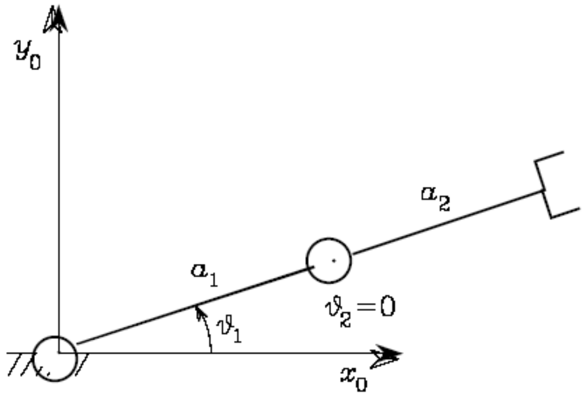
For a two-link manipulator the Jacobian is:

$$\mathbf{J} = \begin{bmatrix} -a_1 s_1 - a_2 s_{12} & -a_2 s_{12} \\ a_1 c_1 + a_2 c_{12} & a_2 c_{12} \end{bmatrix}$$

We can compute singularities:

These are singularities at the borders of the workspace.

$$\det(\mathbf{J}) = a_1 a_2 s_2 = 0 \quad \Leftrightarrow \quad \vartheta_2 = \begin{cases} 0 \\ \pi \end{cases}$$



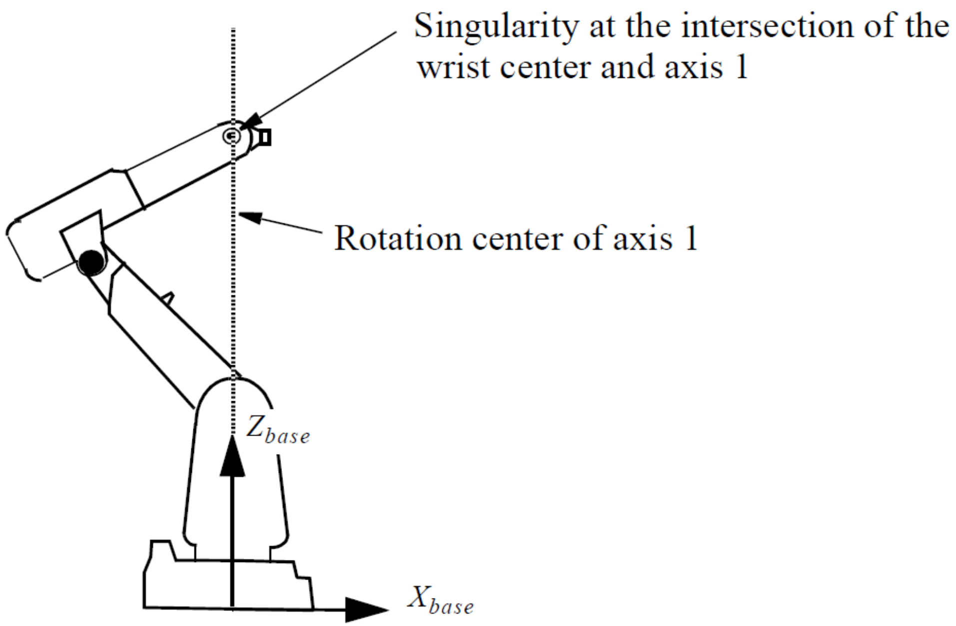
In these configurations the two columns of the Jacobian are not independent.

$\vartheta_2 = 0$

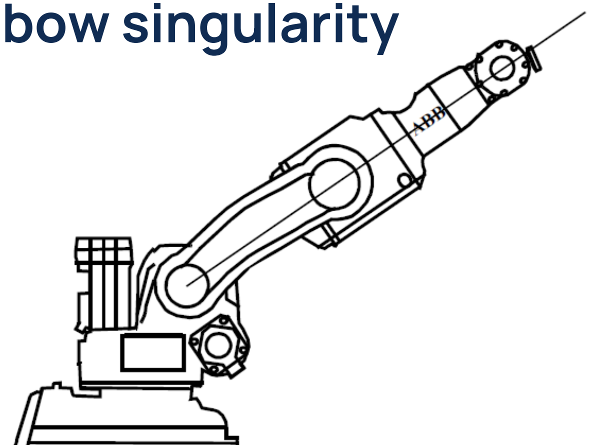
$$\mathbf{J} = \begin{bmatrix} -(a_1 + a_2) s_1 & -a_2 s_1 \\ (a_1 + a_2) c_1 & a_2 c_1 \end{bmatrix}$$

Kinematic singularities of a complete manipulator

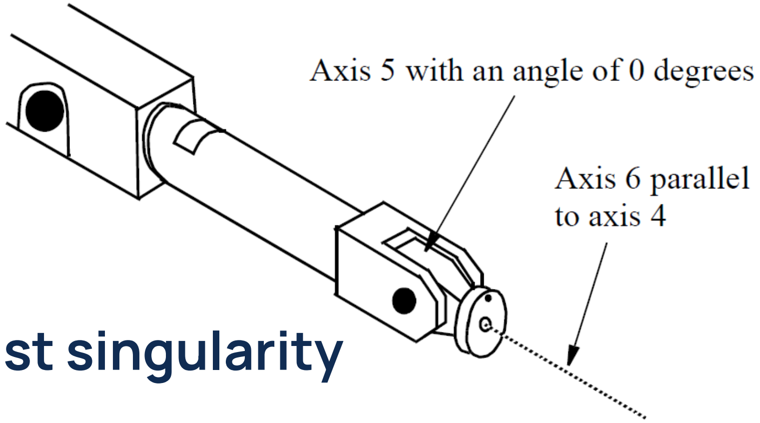
Arm singularity



Elbow singularity

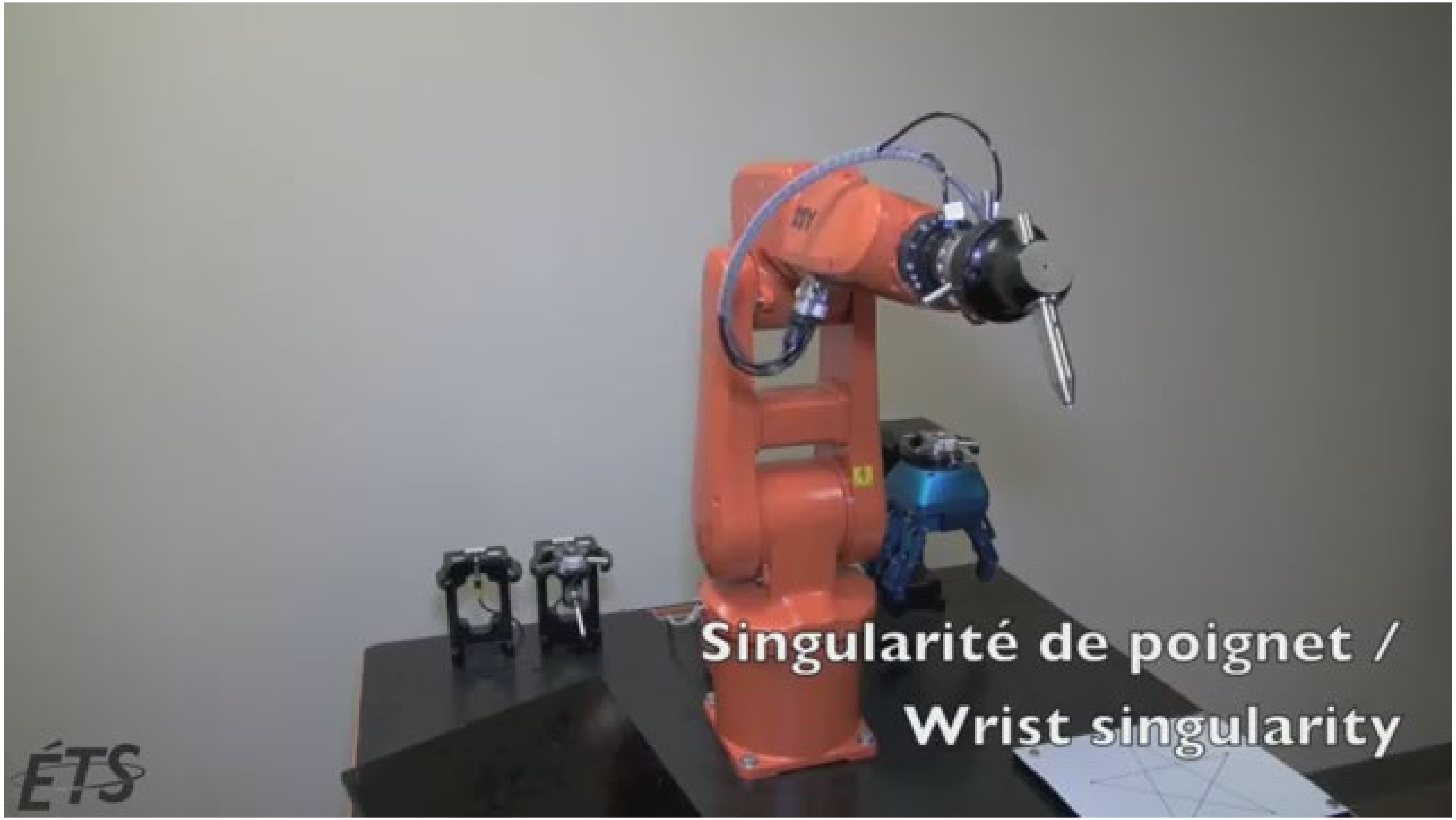


Wrist singularity



Source: ABB

Consequences of kinematic singularities on robot motion



Inversion of the differential kinematics

The differential kinematics is linear for a certain value of $\mathbf{q}$:

$$\dot{\mathbf{x}} = \mathbf{J}_A(\mathbf{q})\dot{\mathbf{q}}$$

Given a vector of desired coordinates in the operational space $\mathbf{x}_d$ and an initial condition on $\mathbf{q}$ we might solve the kinematic inversion problem by **inverting the differential kinematics** and then integrating. If the Jacobian is square $(n = 6)$:

$$\dot{\mathbf{q}} = \mathbf{J}_A^{-1}(\mathbf{q})\dot{\mathbf{x}}_d \quad \Rightarrow \quad \mathbf{q}(t) = \int_0^t \dot{\mathbf{q}}(\sigma)d\sigma + \mathbf{q}(0)$$

However, using this expression directly, **drifts** of the solution may occur.
The error in the operational space made by the kinematic inversion algorithm is then introduced:

$$\mathbf{e} = \mathbf{x}_d - \mathbf{x}$$

Inverse of the Jacobian

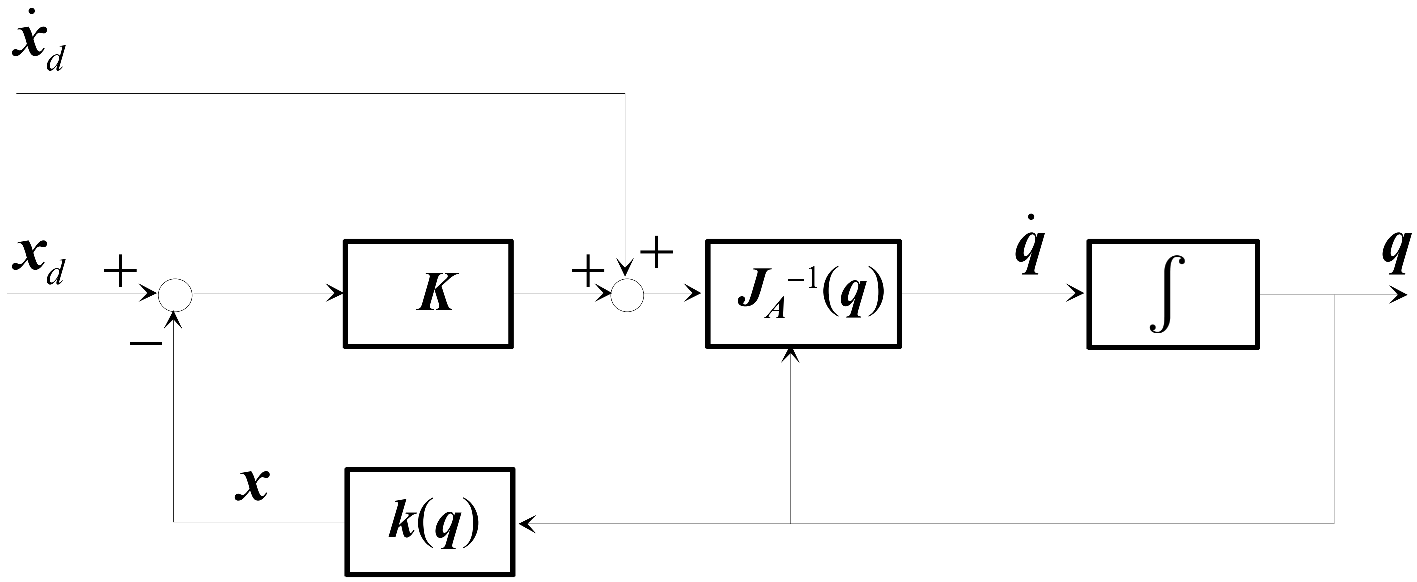
If we adopt the following dependence of $\dot{\mathbf{q}}$ from $\mathbf{e}$:

$$\dot{\mathbf{q}} = \mathbf{J}_A^{-1}(\mathbf{q})(\dot{\mathbf{x}}_d + \mathbf{K}\mathbf{e})$$

we obtain:

$$\dot{\mathbf{e}} + \mathbf{K}\mathbf{e} = 0$$

and the diagram:



Mathematically, this corresponds to solve the inverse kinematics problem through a **Gauss-Newton** iterative method.
Proof of convergence is trivial as:

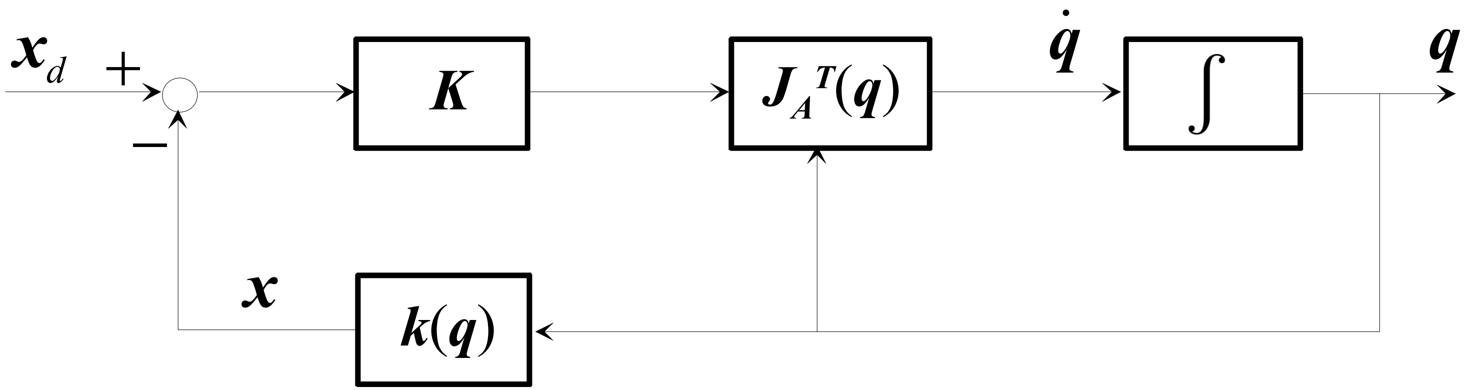
$$\dot{\mathbf{e}} + \mathbf{K}\mathbf{e} = 0$$

Transpose of the Jacobian

If we adopt the following (simpler) dependence:

$$\dot{\mathbf{q}} = \mathbf{J}_A^T(\mathbf{q})\mathbf{K}\mathbf{e}$$

we obtain the diagram:



Mathematically, this corresponds to solve the inverse kinematics problem through a **gradient descent** iterative method.
Proof of convergence can be obtained through a Lyapunov argument.

POLITECNICO MILANO 1863 | DIPARTIMENTO DI ELETTRONICA INFORMAZIONE E BIOINGEGNERIA

Contatti

Paolo Rocco
paolo.rocco@polimi.it