Control of industrial robots

Review of robot kinematics

Prof. Paolo Rocco (paolo.rocco@polimi.it)

Politecnico di Milano

Dipartimento di Elettronica, Informazione e Bioingegneria

Introduction

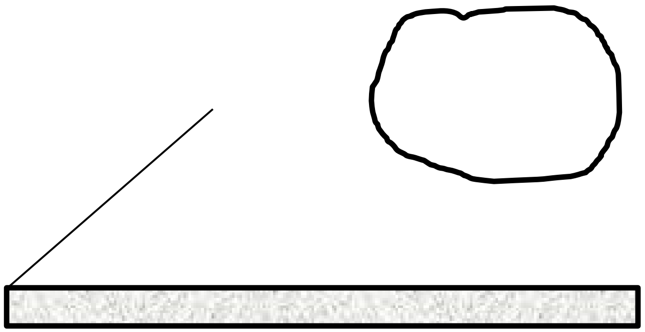
- With these slides we will cover basic elements in robot kinematics.
- We will start from a basic problem of representation of a rigid body in space, and then proceed through the formal tools used in robotics till the definition of the direct, inverse and differential kinematics of the manipulator.
- 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.

Most 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

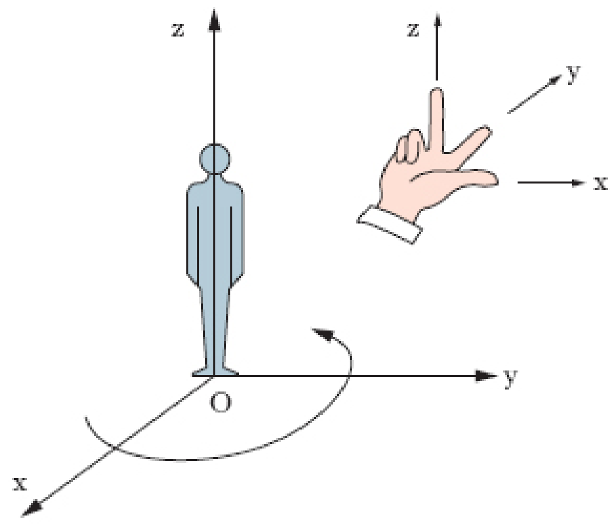
Position and orientation of a body in space

Let us consider a rigid body in space:



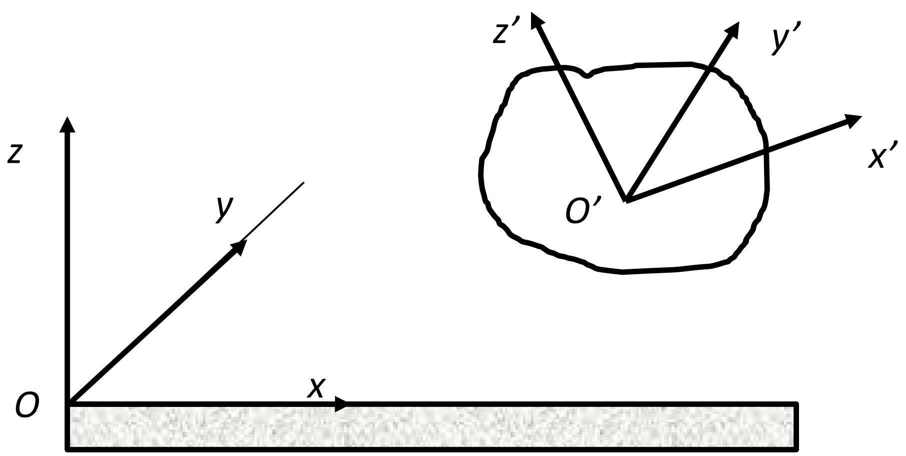
How can we characterize the position and orientation of the body in space?

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

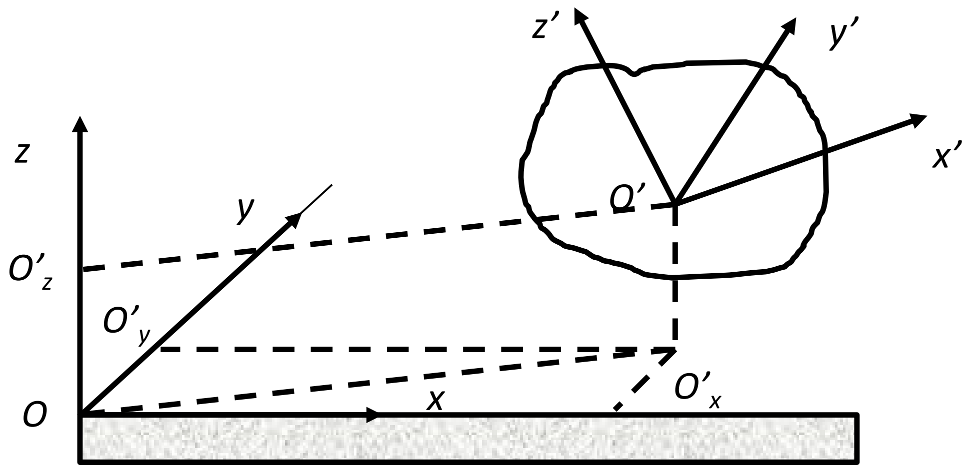
The best thing to do is to consider a reference frame and to attach a second frame to the body.



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:

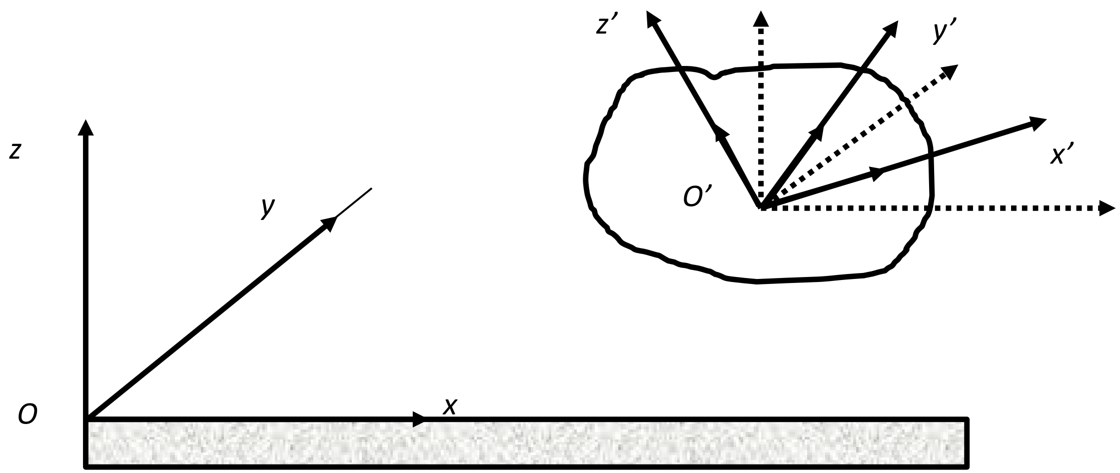


The three components can be conveniently gathered in a **vector**:

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

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:



We obtain three vectors:

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

Rotation matrix

We can gather the elements of ***x*** ', ***y*** ', ***z*** ' in a matrix:

$$\boldsymbol{R} = \begin{bmatrix} \boldsymbol{x}' & \boldsymbol{y}' & \boldsymbol{z}' \end{bmatrix} = \begin{bmatrix} x'_x & y'_x & z'_x \\ x'_y & y'_y & z'_y \\ x'_z & y'_z & z'_z \end{bmatrix} = \begin{bmatrix} \boldsymbol{x}'^T\boldsymbol{x} & \boldsymbol{y}'^T\boldsymbol{x} & \boldsymbol{z}'^T\boldsymbol{x} \\ \boldsymbol{x}'^T\boldsymbol{y} & \boldsymbol{y}'^T\boldsymbol{y} & \boldsymbol{z}'^T\boldsymbol{y} \\ \boldsymbol{x}'^T\boldsymbol{z} & \boldsymbol{y}'^T\boldsymbol{z} & \boldsymbol{z}'^T\boldsymbol{z} \end{bmatrix}$$

This matrix is called **rotation matrix** of the frame (***x*** ', ***y*** ', ***z*** ') with respect to the frame (***x***, ***y***, ***z***) .

Since the following relations hold:

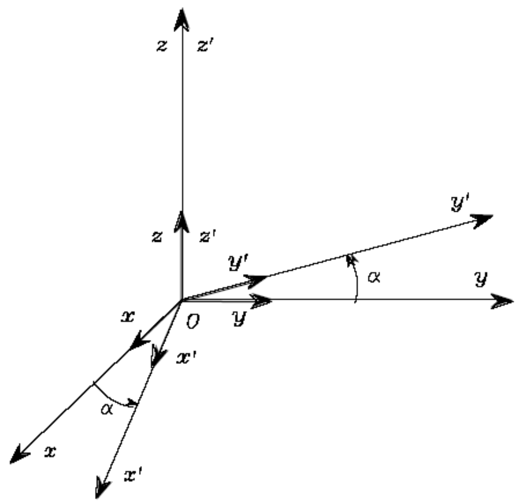
$$\boldsymbol{x}'^T\boldsymbol{x}' = 1, \quad \boldsymbol{y}'^T\boldsymbol{y}' = 1, \quad \boldsymbol{z}'^T\boldsymbol{z}' = 1$$

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

we have: $\boldsymbol{R}^T\boldsymbol{R} = \boldsymbol{I} \quad \left(\boldsymbol{R}^T = \boldsymbol{R}^{-1}\right)$ orthogonal matrix

Elementary rotations

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



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

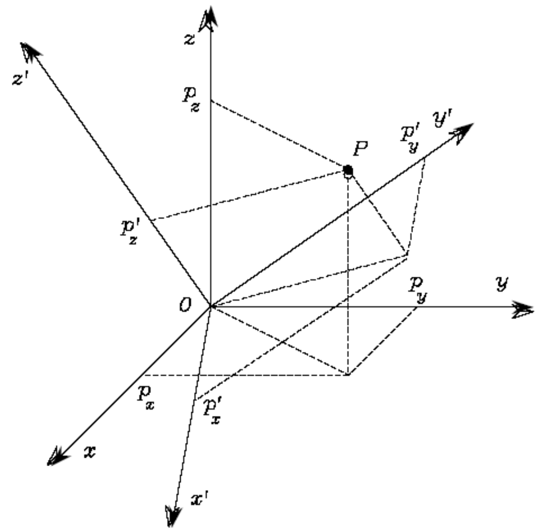
The rotation matrix is thus:

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

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:

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

Therefore:

$$\boldsymbol{p} = p'_x \boldsymbol{x}' + p'_y \boldsymbol{y}' + p'_z \boldsymbol{z}' = \begin{bmatrix} \boldsymbol{x}' & \boldsymbol{y}' & \boldsymbol{z}' \end{bmatrix} \boldsymbol{p}' = \boldsymbol{R}\boldsymbol{p}'$$

The rotation matrix thus contains the transformation which maps the coordinates expressed in the frame (***x***', ***y*** ', ***z*** ') into the coordinates expressed in frame (***x***, ***y***, ***z***).

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

Composition of rotation matrices

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

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

Thus:

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

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

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

Rotations can be obtained by composing partial rotations.

Partial rotation matrices are multiplied from left to right.

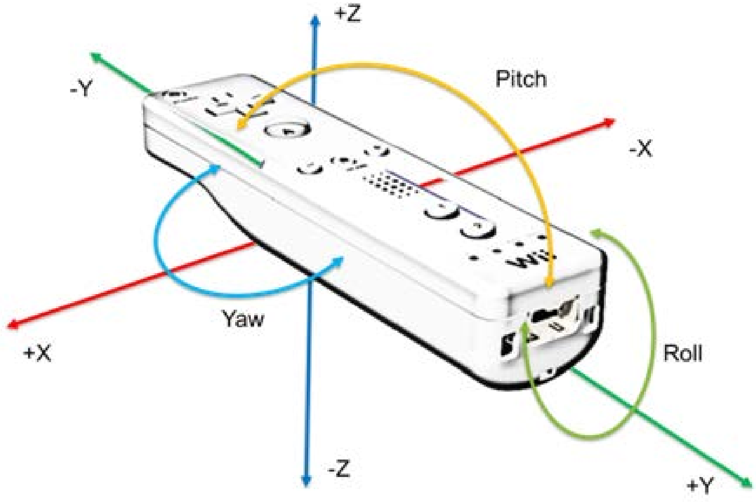
Minimal representation of the orientation

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)



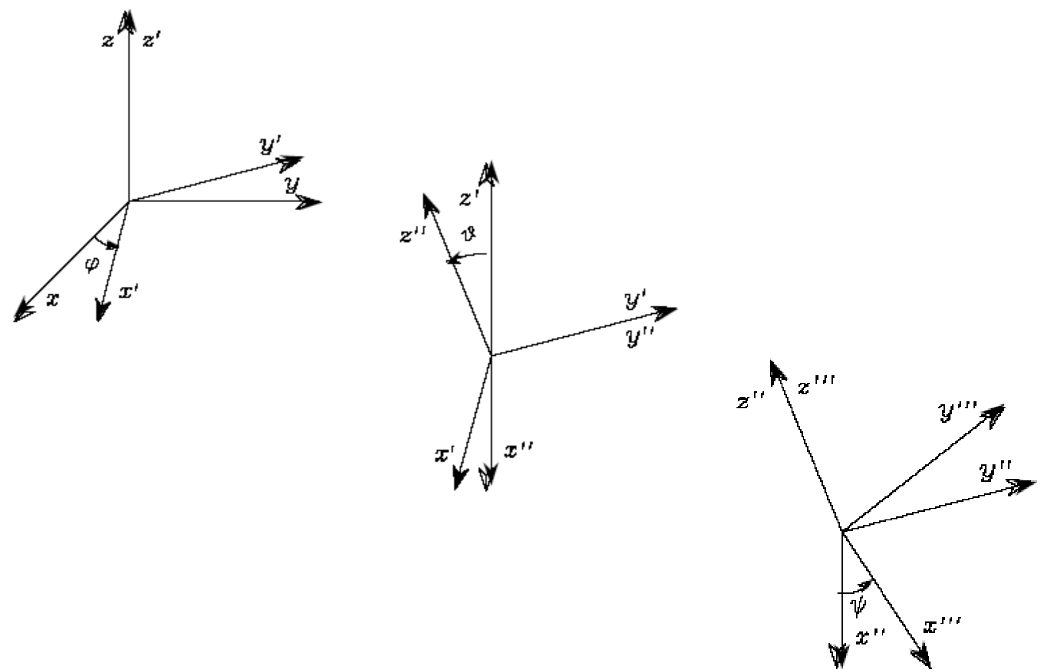
ZYZ Euler angles

With ZYZ Euler angles the sequence is composed as:

I) Rotation around Z (angle φ)

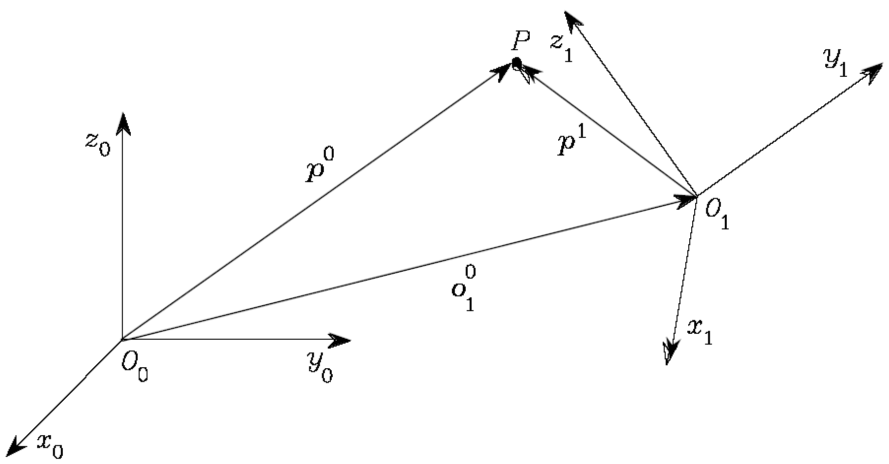
II) Rotation around Y' (angle ϑ)

III) Rotation around Z'' (angle ψ)



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

Homogeneous representation



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

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

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

Inverse transform:

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

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

$$\tilde{\boldsymbol{p}} = \begin{bmatrix} w\boldsymbol{p} \\ w \end{bmatrix}$$

Homogeneous representation

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

Homogeneous transformations

Let us introduce the homogeneous transformation matrix (size 4×4):

$$\boldsymbol{A}_1^0 = \begin{bmatrix} \boldsymbol{R}_1^0 & \boldsymbol{o}_1^0 \\ \mathbf{0}^T & 1 \end{bmatrix}$$

The relationship:

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

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

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

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

The inverse transformation is:

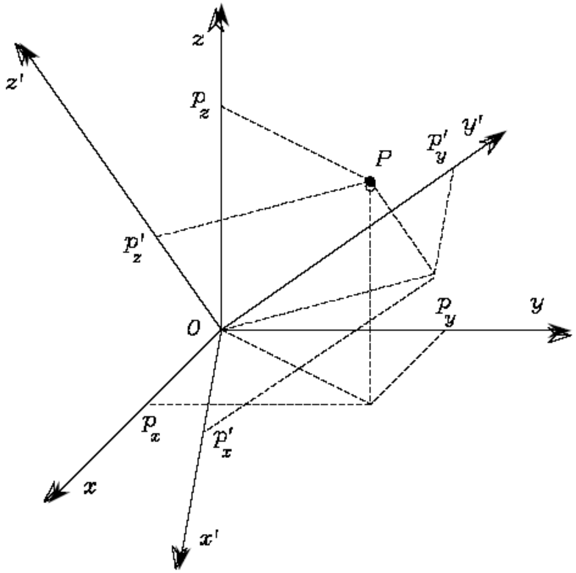
$$\tilde{\boldsymbol{p}}^1 = \boldsymbol{A}_0^1 \tilde{\boldsymbol{p}}^0 = (\boldsymbol{A}_1^0)^{-1} \tilde{\boldsymbol{p}}^0 \qquad \boldsymbol{A}_0^1 = \begin{bmatrix} \boldsymbol{R}_0^1 & -\boldsymbol{R}_0^1 \boldsymbol{o}_1^0 \\ \mathbf{0}^T & 1 \end{bmatrix}$$

N.B. $\boldsymbol{A}$ is not orthogonal

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

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 $\boldsymbol{p}'$. The coordinates of the same point in the stationary frame are:



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

Take now the derivative with respect to time:

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

How can we express the derivative of a rotation matrix?

Derivative of a rotation matrix

Since the rotation matrix is orthogonal we have:

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

If we define the new matrix:

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

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

Matrix ***S*** then takes the following form:

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

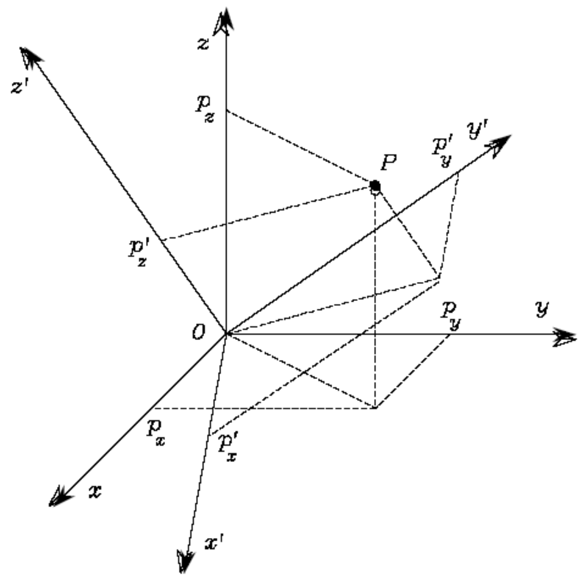
We conclude that the derivative of a rotation matrix is given by:

$$\dot{\boldsymbol{R}}(t) = \boldsymbol{S}(\omega(\boldsymbol{t}))\boldsymbol{R}(t)$$

Skew symmetric matrix

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

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



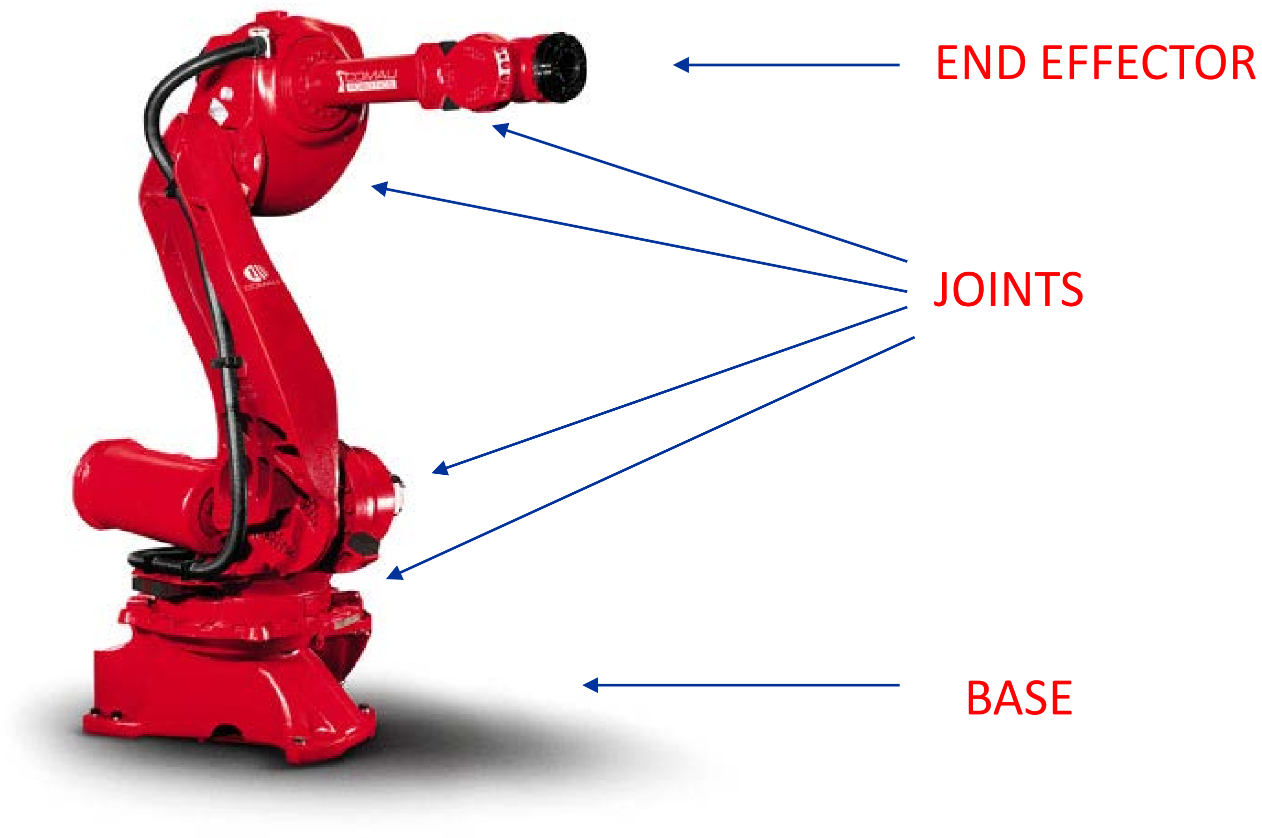
On the other hand the same vector denotes the velocity of point *P* in the stationary frame:

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

- ω is the angular velocity vector of the rotating frame
- symbol $\times$ denotes cross product

Thus the skew symmetric matrix ***S*** can be interpreted as the operator that computes the cross product.

How does all this relate to the robot?

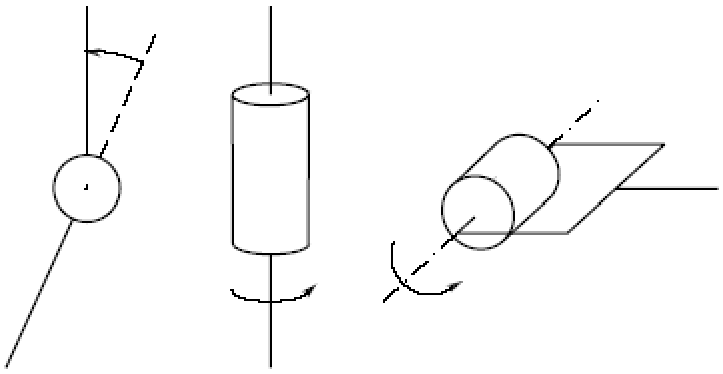


The joints

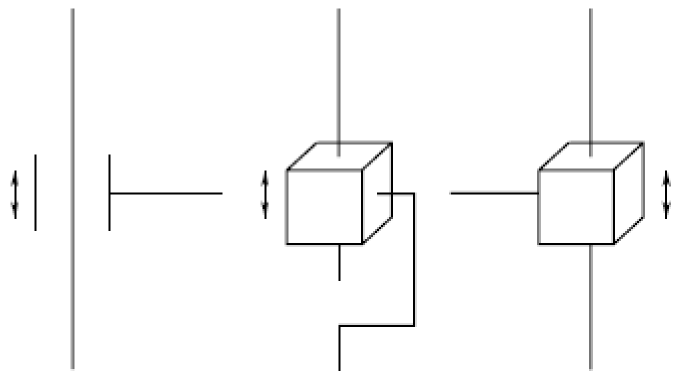
Each joint allows for one (and only one) *degree of freedom* between two links. We call joint variable the coordinate associated to such degree of freedom, and then we introduce the vector of joint variables:

$$\boldsymbol{q} = \begin{bmatrix} q_1 \\ q_2 \\ \vdots \\ q_n \end{bmatrix}$$

Schematic draws of the joints:



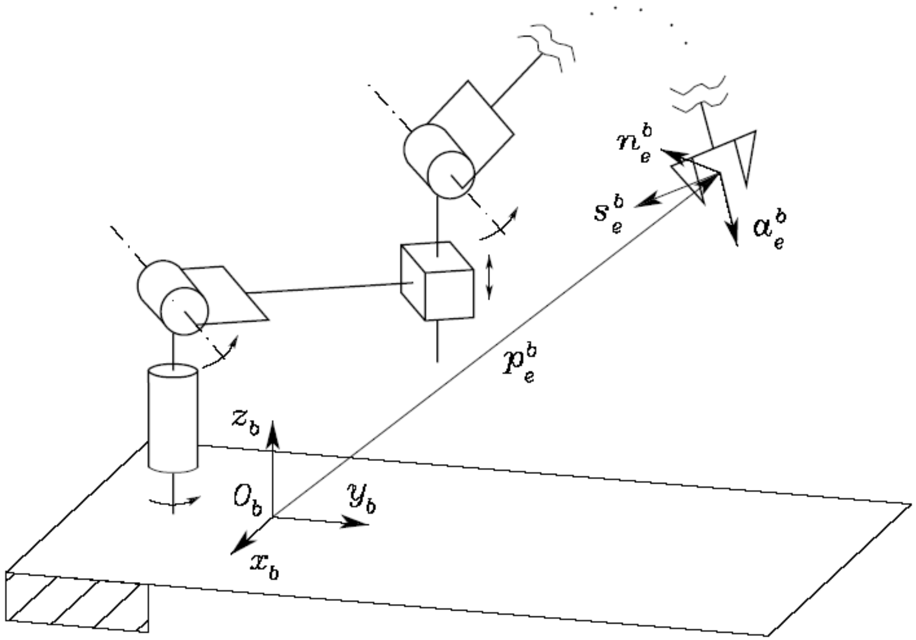
ROTATIONAL JOINTS



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:

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

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

Direct kinematics

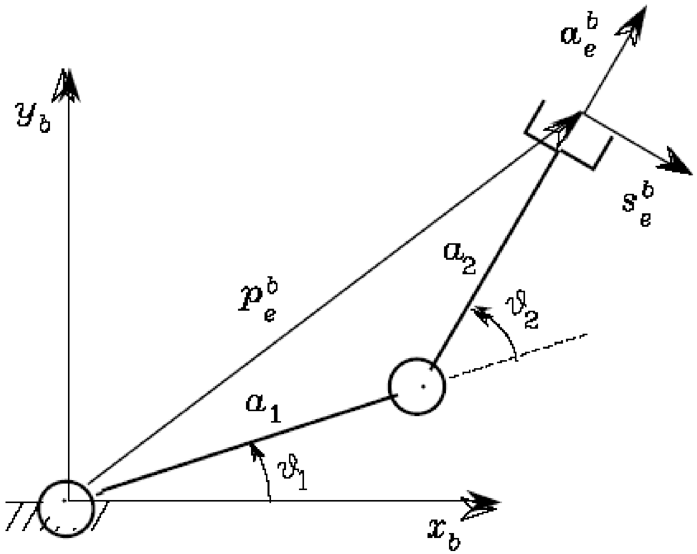
The direct kinematic equation gives position and orientation of the tool frame w.r.t. the base frame, as a function of the joint variables.

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

(homogeneous transformation matrix)

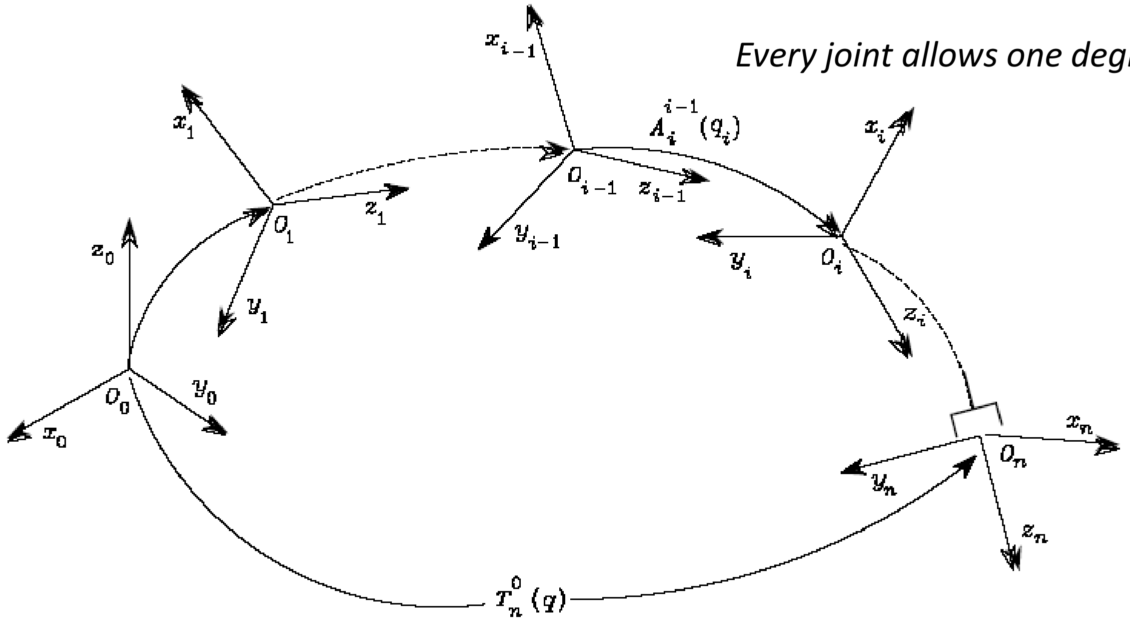
Example: planar two-link manipulator

$$\boldsymbol{T}_e^b(\boldsymbol{q}) = \begin{bmatrix} 0 & s_{12} & c_{12} & a_1c_1 + a_2c_{12} \\ 0 & -c_{12} & s_{12} & a_1s_1 + a_2s_{12} \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$



Direct kinematics

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

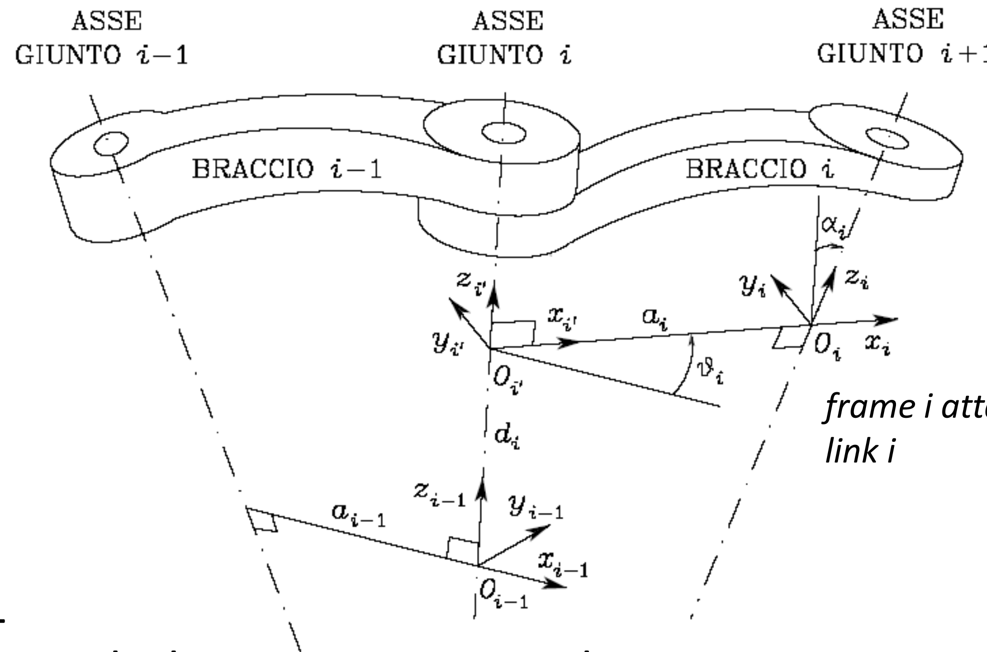


Proceeding iteratively:

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

Denavit-Hartenberg convention

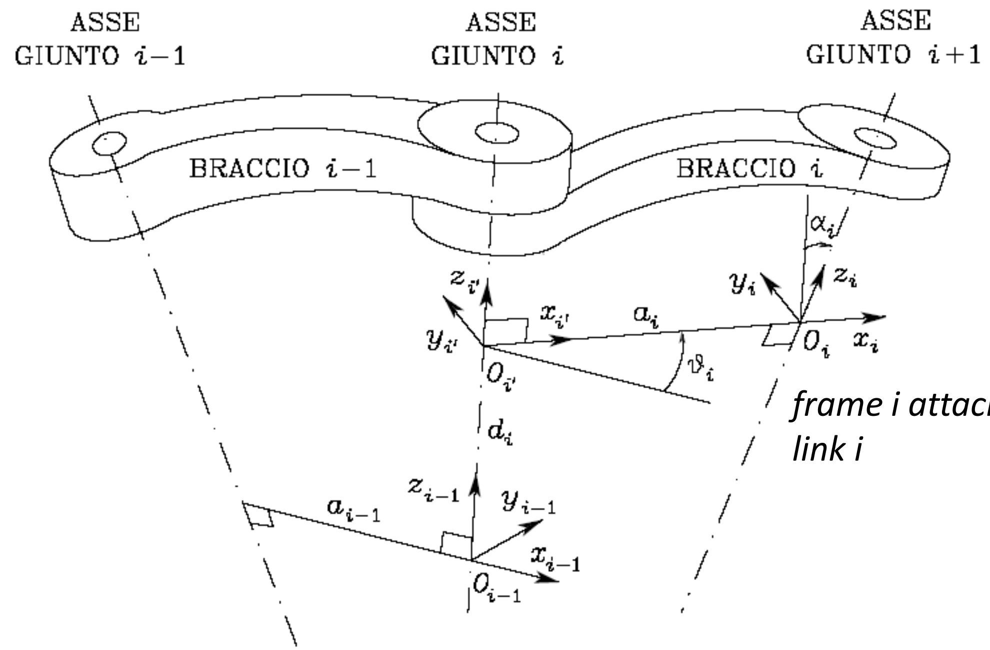
It is a convention for the selection of the frames attached to each link.



- 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 e z_{i-1}; we denote withO_i ' the intersection of this common normal with axis z_{i-1}
- x_i is aligned with the common normal to axes z_i e 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.



frame i attached to link i

- a_i distance of O_i from O_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=rA9tm0gTIn8

Homogeneous transformation matrix

How to construct 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}:

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

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 ' :

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

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

Joint space and operational space

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

$$\boldsymbol{q} = \begin{bmatrix} q_1 \\ \vdots \\ q_n \end{bmatrix} \qquad \begin{array}{l} q_i = \vartheta_i \text{ (rotating joint)} \\ \\ q_i = d_i \text{ (prismatic joint)} \end{array}$$

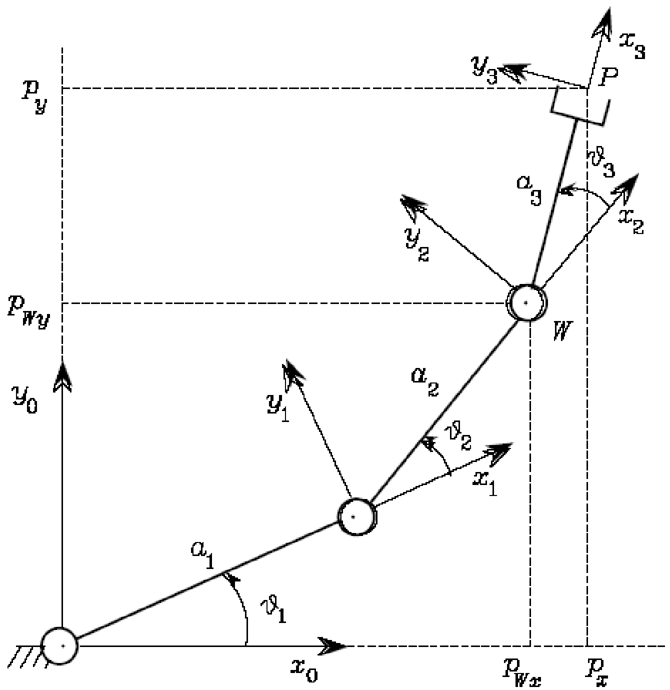
The **operational space** is the space where the task that the manipulator has to accomplish is specified. It is defined by the posture $\boldsymbol{x}$:

$$\boldsymbol{x} = \begin{bmatrix} \boldsymbol{p} \\ \phi \end{bmatrix} \qquad \begin{array}{l} \boldsymbol{p} \text{ (position)} \\ \phi \text{ (minimal representation of the orientation)} \end{array}$$

m components

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

Three d.o.f. planar manipulator



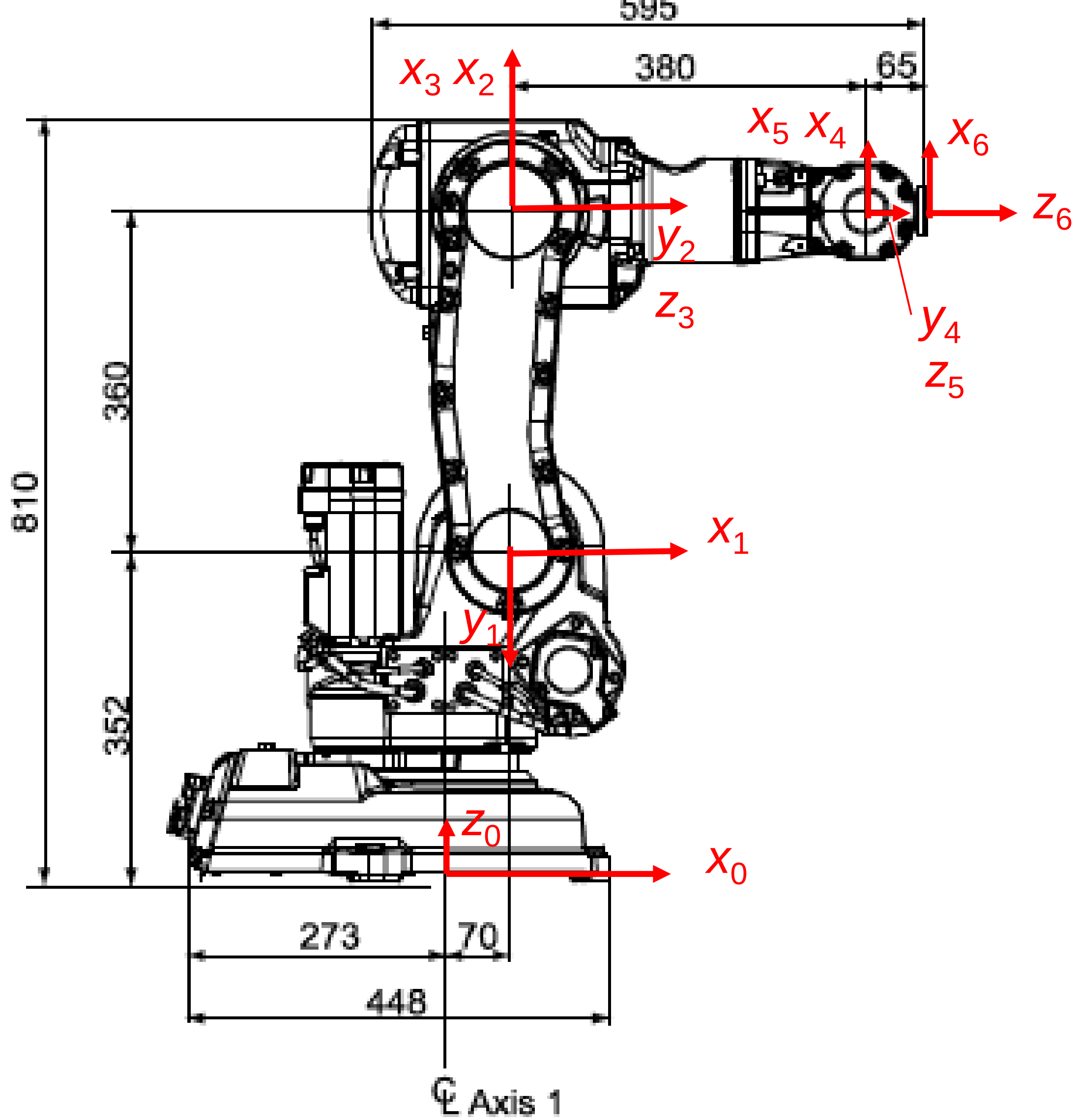
	a_i	α_i	d_i	ϑ_i
1	a_1	0	0	ϑ_1
2	a_2	0	0	ϑ_2
3	a_3	0	0	ϑ_3

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

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

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

A six d.o.f. robot



Inverse kinematics problem

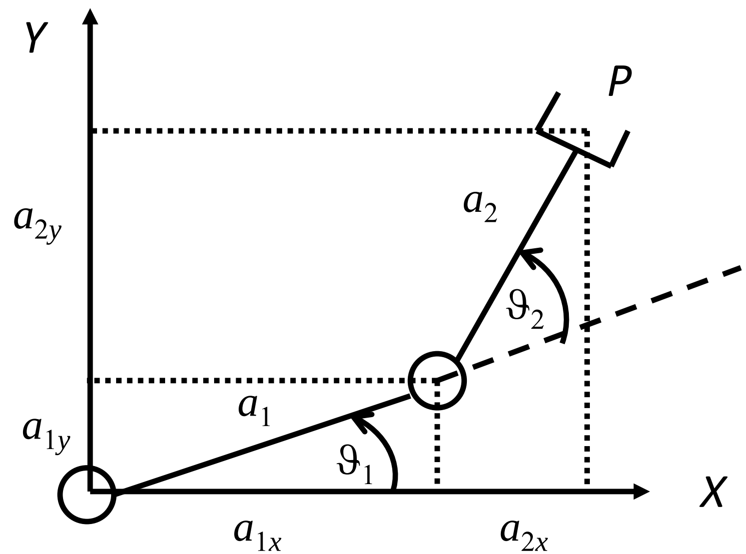
$$\boldsymbol{T} \Rightarrow \boldsymbol{q}$$
$$\boldsymbol{x} \Rightarrow \boldsymbol{q}$$

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:

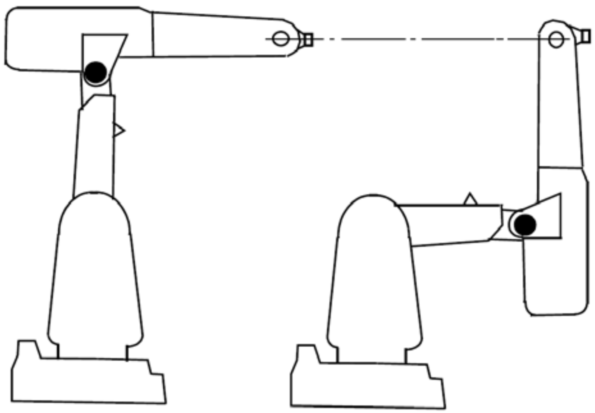
$$c_2 = \frac{p_x^2 + p_y^2 - a_1^2 - a_2^2}{2a_1a_2} \quad \Rightarrow \quad \vartheta_2 = \mathrm{Atan}\,2(s_2, c_2)$$
$$s_2 = \pm\sqrt{1 - c_2^2}$$

2 solutions

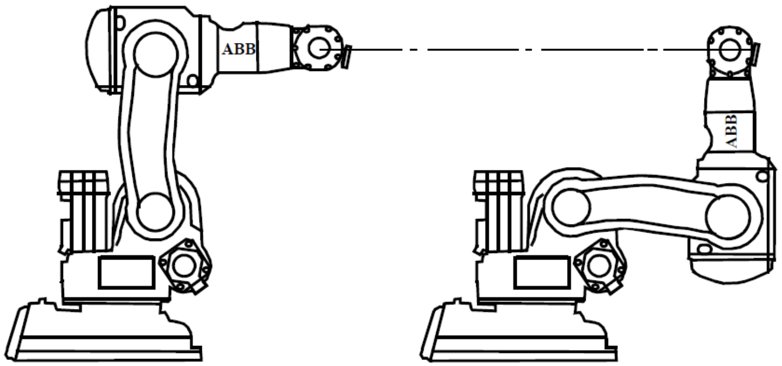
$$c_1 = \frac{(a_1 + a_2c_2)p_x + a_2s_2p_y}{p_x^2 + p_y^2}$$
$$s_1 = \frac{(a_1 + a_2c_2)p_y - a_2s_2p_x}{p_x^2 + p_y^2} \quad \Rightarrow \quad \vartheta_1 = \mathrm{Atan}\,2(s_1, c_1)$$

COMPLICATED!

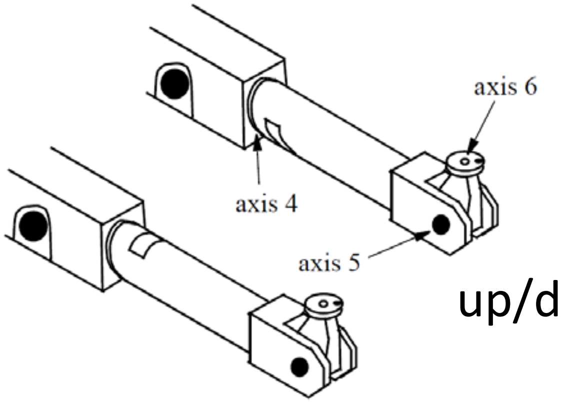
Anthropomorphic manipulator



right/left shoulder



up/down elbow



up/down wrist

Eight admissible configurations exist

Differential kinematics: geometrical Jacobian

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

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

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

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

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

is called **geometrical Jacobian** of the manipulator.
Systematic methods exist to compute the Jacobian.

Analytical Jacobian

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

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

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

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

On the other hand:

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

Matrix: $\boldsymbol{J}_A(\boldsymbol{q}) = \begin{bmatrix} \boldsymbol{J}_P(\boldsymbol{q}) \\ \boldsymbol{J}_\phi(\boldsymbol{q}) \end{bmatrix}$ 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:

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

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

$$\boldsymbol{T}(\phi) = \begin{bmatrix} 0 & -s_\varphi & c_\varphi s_\vartheta \\ 0 & c_\varphi & s_\varphi s_\vartheta \\ 1 & 0 & c_\vartheta \end{bmatrix}$$

(for the ZYZ Euler angles)

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

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

The relation between analytical and geometrical Jacobian follows: $\boldsymbol{J} = \boldsymbol{T}_A(\phi)\boldsymbol{J}_A$

Kinematic singularities

The equation defining the geometrical Jacobian is:

$$\boldsymbol{v} = \boldsymbol{J}(\boldsymbol{q})\dot{\boldsymbol{q}}$$

The values of $\boldsymbol{q}$ for which matrix $\boldsymbol{J}$ 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.

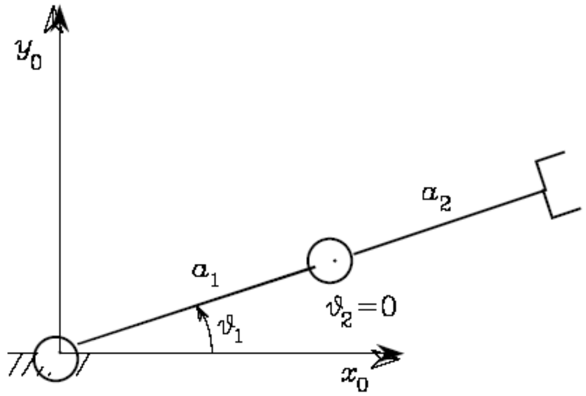
Kinematic singularities: example

For a two-link manipulator the Jacobian is:

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

We can compute singularities:

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

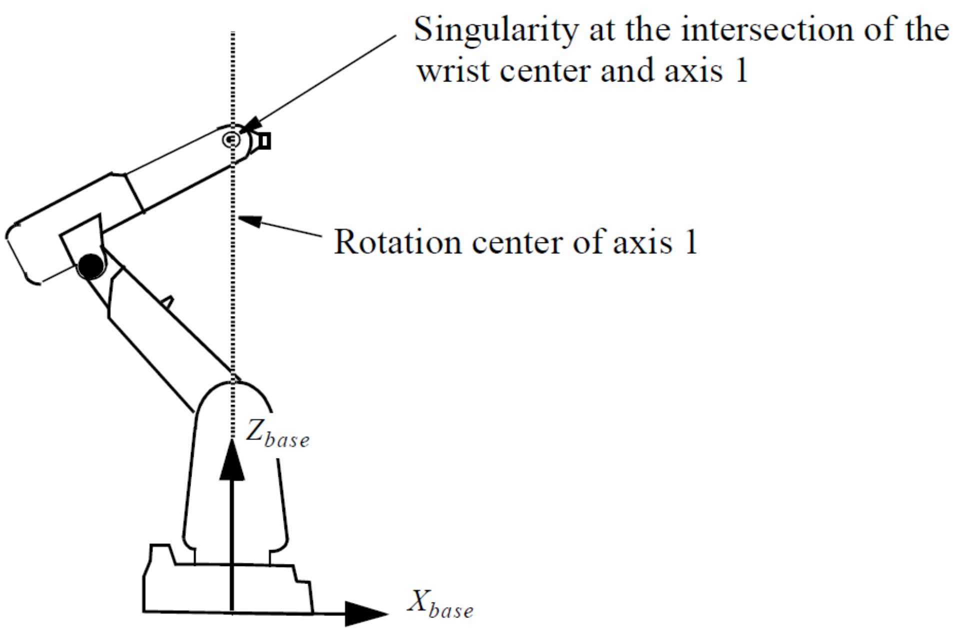


These are singularities at the borders of the workspace.

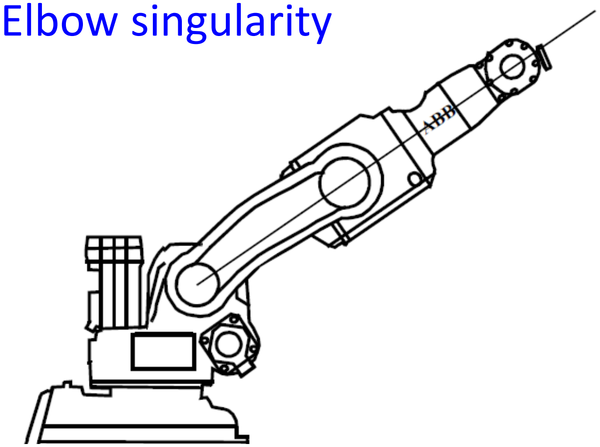
In these configurations the two columns of the Jacobian are not independent.

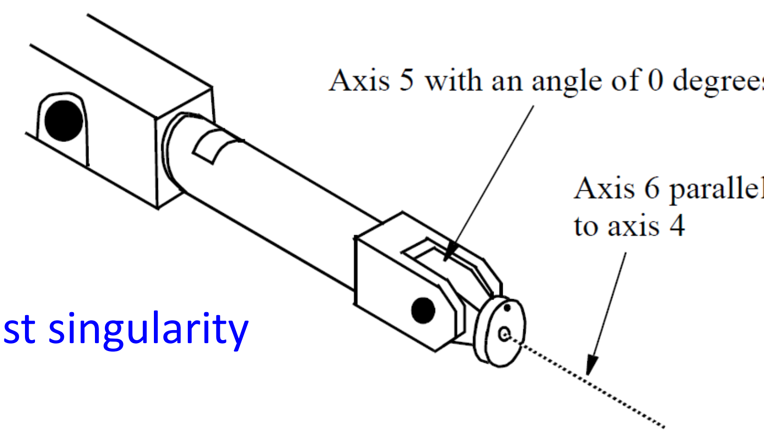
Kinematic singularities of a complete manipulator

Arm singularity



Elbow singularity





Wrist singularity

Source: ABB

Consequences of kinematic singularities on robot motion

https://www.youtube.com/watch?v=zlGCurgsqg8

Inversion of the differential kinematics

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

$$\boldsymbol{v} = \boldsymbol{J}(\boldsymbol{q})\dot{\boldsymbol{q}}$$

Given a velocity $\boldsymbol{v}$ in the operational space and an initial condition on $\boldsymbol{q}$ we might solve the kinematic inversion problem by inverting the differential kinematics and then integrating. If the Jacobian is square ($n = r$, number of coordinates in the operational space needed to describe a task):

$$\dot{\boldsymbol{q}} = \boldsymbol{J}^{-1}(\boldsymbol{q})\boldsymbol{v} \quad \Rightarrow \quad \boldsymbol{q}(t) = \int_0^t \dot{\boldsymbol{q}}(\sigma)d\sigma + \boldsymbol{q}(0)$$

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:

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

Inverse of the Jacobian

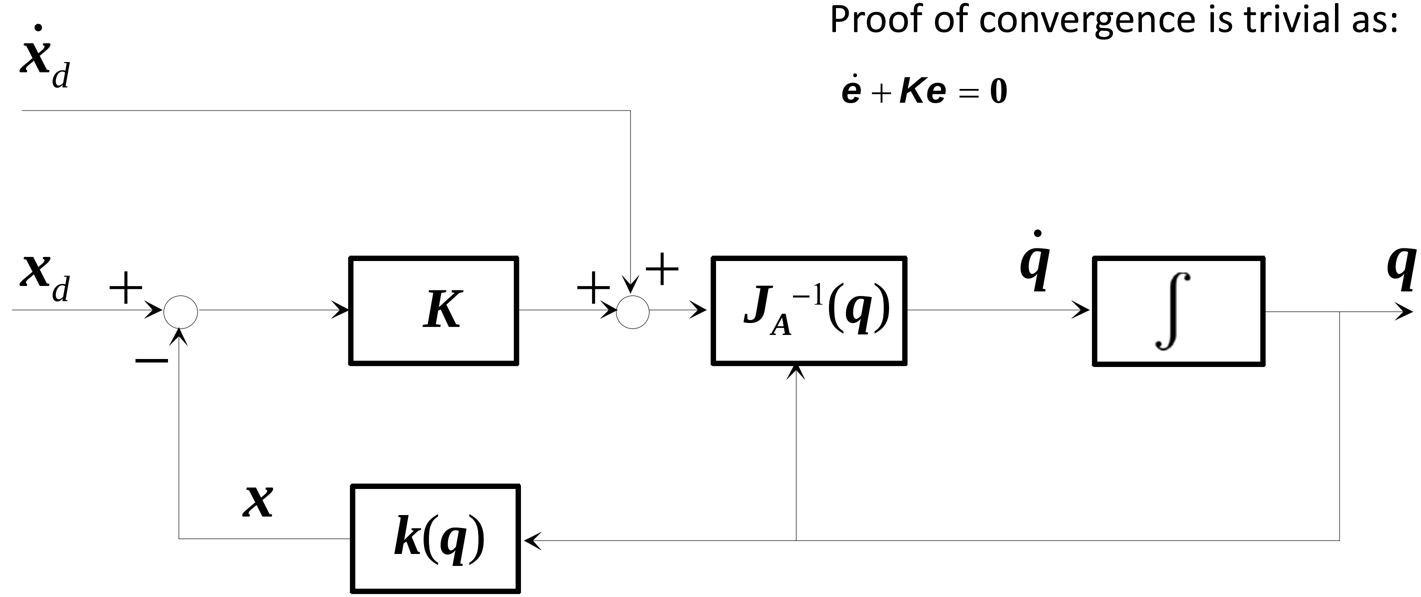
If we adopt the following dependence of $\dot{\boldsymbol{q}}$ from $\boldsymbol{e}$:

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

we obtain:

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

and the diagram:



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

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

Transpose of the Jacobian

If we adopt the following (simpler) dependence:

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

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.