Transformation models for robotics

Computation of some mathematical models for design and control of robots, such as :

- Transformation models between the joint space (in which the configuration of the robot is defined) and the task space (in which the location of the end-effector is specified) :
  - direct and inverse geometric models
    giving the location of the end-effector X as a function of the joint variables of the mechanism q and vice versa ;
  - direct and inverse kinematic models
    giving the velocity of the end-effector $\dot{X}$ as a function of the joint velocities $\dot{q}$ and vice versa.
- dynamic models giving the relations between the input torques or forces of the actuators Γ and the positions q, velocities $\dot{q}$ and accelerations $\ddot{q}$ of the joints.

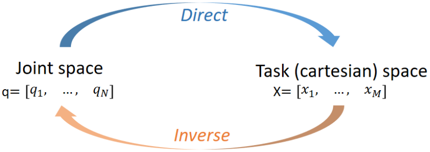
Transformation models for robotics

Computation of some mathematical models for design and control of robots, such as :

- Transformation models between the joint space (in which the configuration of the robot is defined) and the task space (in which the location of the end-effector is specified) :
  - direct and inverse geometric models
    giving the location of the end-effector X as a function of the joint variables of the mechanism q and vice versa ;
  - direct and inverse kinematic models
    giving the velocity of the end-effector $\dot{X}$ as a function of the joint velocities $\dot{q}$ and vice versa.
- dynamic models giving the relations between the input torques or forces of the actuators Γ and the positions q, velocities $\dot{q}$ and accelerations $\ddot{q}$ of the joints.

Remark :
In this course of introduction to robotics, only the case of simple open-tree structures will be studies. For example, in [13] and [21], the cases of parallel robots and multiple open-tree structures are studied.

Formulation of geometric models



- Case of the direct geometric model of the manipulator $X = f(q)$:
  - Expression of the position and orientation X of the end-effector as a function of joint variables q ;
  - Methodology :
    1. assign frames $\mathscr{R}_i$ to each rigid body $\mathscr{C}_i$ of the chain ;
    2. description of their relative position/orientation using one dedicated setting (convention used in this course : Modified *Denavit-Hartenberg* - MDH -, called *Khalil-Kleinfinger* [14]) ;
    3. use of homogeneous transformation matrices to describe change transformation $\overline{g}_{(i-1)i}$ between two adjacent bodies $\mathscr{C}_{i-1}$ and $\mathscr{C}_i$;
    4. forward kinematic of the complete kinematic chain from $\mathscr{C}_0$ to $\mathscr{C}_n$ obtained recursively.
- Case of the inverse geometric description of the manipulator $q = f^{-1}(X)$:
  - Joint coordinates q needed to bring the end-effector in a prescribed position and orientation X ;
  - Nature and number of equations raising the issues about *existence* and *multiplicity* of the solution.

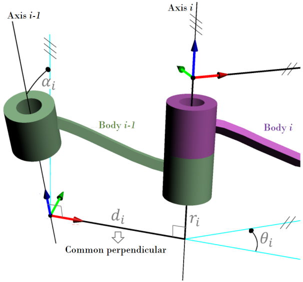
Khalil-Kleinfinger convention

Rigid body given by the relative position of its axes $i-1$ and i :

- distance d_i (length of the common perpendicular to the two axes) ;
- twisting angle α_i (angle of the rotation around the common perpendicular that brings the two axes parallel) ;

Relative position of two successive bodies thanks to two parameters :

- angle θ_i (angle of the needed rotation around axis i to bring the common perpendicular of body $i-1$ parallel to the one of body i) ;
- distance r_i which must then be translated along the i axis to bring them into coincidence..



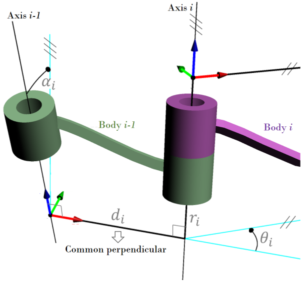
Khalil-Kleinfinger convention

Rigid body given by the relative position of its axes $i-1$ and i :

- distance d_i (length of the common perpendicular to the two axes) ;
- twisting angle α_i (angle of the rotation around the common perpendicular that brings the two axes parallel) ;

Relative position of two successive bodies thanks to two parameters :

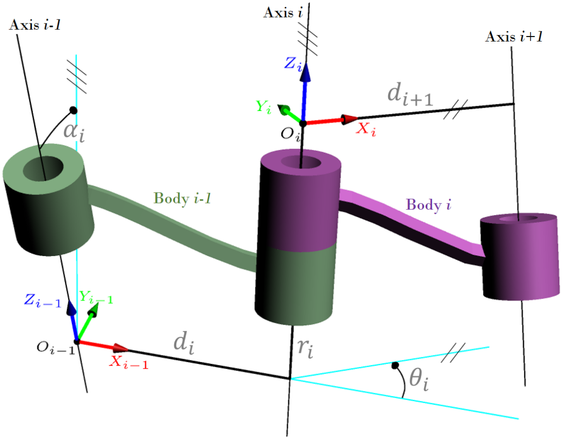
- angle θ_i (angle of the needed rotation around axis i to bring the common perpendicular of body $i-1$ parallel to the one of body i) ;
- distance r_i which must then be translated along the i axis to bring them into coincidence..



Remarks :

1. θ_i or r_i : joint variables
   - $q_i = \theta_i$ for a revolute joint ;
   - $q_i = r_i$ for a prismatic joint.
2. $(\alpha_i, d_i, \theta_i, r_i)$: four parameters constituting a minimal geometrical characterization of a body with the following body.

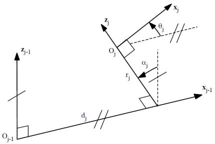
Khalil-Kleinfinger convention



Methodology for computing the DGM

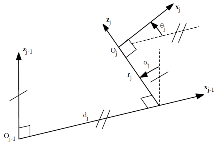
1. Assignation of orthonormal frame $\mathscr{R}_i$ to each body $\mathscr{C}_i$:

- the axis Z_i is along the axis of joint connecting body $\mathscr{C}_{i-1}$ to body $\mathscr{C}_i$;
- the axis X_i is aligned with the common normal between Z_i and Z_{i+1} (i.e. $X_i = Z_i \wedge Z_{i+1}$) ;
  - if Z_i and Z_{i+1} are collinear, then X_i is not unique and can be taken in any plane perpendicular to them ;
  - if Z_i and Z_{i+1} are parallel, then X_i is not unique and is in the plane defined by them ;
  - in the case of intersecting joint axes, X_i is normal to the plane defined by them and passing through their intersection point ;
- the Y_i axis is formed by the right-hand rule to complete the coordinate system (X_i, Y_i, Z_i) (i.e. $Y_i = Z_i \wedge X_i$).



Methodology for computing the DGM

1. Assignation of orthonormal frame $\mathscr{R}_i$ to each body $\mathscr{C}_i$:
   - the axis Z_i is along the axis of joint connecting body $\mathscr{C}_{i-1}$ to body $\mathscr{C}_i$;
   - the axis X_i is aligned with the common normal between Z_i and Z_{i+1} (i.e. $X_i = Z_i \wedge Z_{i+1}$) ;
     - if Z_i and Z_{i+1} are collinear, then X_i is not unique and can be taken in any plane perpendicular to them ;
     - if Z_i and Z_{i+1} are parallel, then X_i is not unique and is in the plane defined by them ;
     - in the case of intersecting joint axes, X_i is normal to the plane defined by them and passing through their intersection point ;
   - the Y_i axis is formed by the right-hand rule to complete the coordinate system (X_i, Y_i, Z_i) (i.e. $Y_i = Z_i \wedge X_i$).



2. Transformation between succesive bodies $\mathscr{C}_{i-1}$ and $\mathscr{C}_i$ given by the following four parameters :
   - α_i : angle between axes Z_{i-1} and Z_i around axis X_{i-1} ;
   - d_i : distance between axes Z_{i-1} and Z_i along axis X_{i-1} ;
   - θ_i : angle between axes X_{i-1} and X_i around axis Z_i ;
   - r_i : distance between axes X_{i-1} and X_i along axis Z_i ;

Methodology for computing the DGM

3. Change transformation between two adjacent bodies :
   - Transformation matrix defining the frame $\mathscr{R}_i$ relative to frame $\mathscr{R}_{i-1}$

$$
\begin{aligned}
\overline{g}_{(i-1)i} =& R_{x,\alpha_i}\, Trans\,(x, d_i)\, R_{z,\theta_i}\, Trans\,(z, r_i) \\
=& \begin{bmatrix} cos\,(\theta_i) & -sin\,(\theta_i) & 0 & d_i \\ cos\,(\alpha_i)\, sin\,(\theta_i) & cos\,(\alpha_i)\, cos\,(\theta_i) & -sin\,(\alpha_i) & -r_i sin\,(\alpha_i) \\ sin\,(\alpha_i)\, sin\,(\theta_i) & sin\,(\alpha_i)\, cos\,(\theta_i) & cos\,(\alpha_i) & r_i cos\,(\alpha_i) \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
=& \begin{bmatrix} R_{(i-1)i} & p_{(i-1)i} \\ 0_{1\times 3} & 1 \end{bmatrix}
\end{aligned}
$$

Methodology for computing the DGM

3. Change transformation between two adjacent bodies :
   - Transformation matrix defining the frame $\mathscr{R}_i$ relative to frame $\mathscr{R}_{i-1}$

$$
\begin{aligned}
\overline{g}_{(i-1)i} =& R_{x,\alpha_i}\, Trans\,(x, d_i)\, R_{z,\theta_i}\, Trans\,(z, r_i) \\
=& \begin{bmatrix} \cos(\theta_i) & -\sin(\theta_i) & 0 & d_i \\ \cos(\alpha_i)\sin(\theta_i) & \cos(\alpha_i)\cos(\theta_i) & -\sin(\alpha_i) & -r_i \sin(\alpha_i) \\ \sin(\alpha_i)\sin(\theta_i) & \sin(\alpha_i)\cos(\theta_i) & \cos(\alpha_i) & r_i \cos(\alpha_i) \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
=& \begin{bmatrix} R_{(i-1)i} & p_{(i-1)i} \\ 0_{1\times 3} & 1 \end{bmatrix}
\end{aligned}
$$

   - Inverse transformation matrix defining the frame $\mathscr{R}_{i-1}$ relative to frame $\mathscr{R}_i$

$$
\begin{aligned}
\overline{g}^{-1}_{(i-1)i} =& \overline{g}_{i(i-1)} \\
=& Trans\,(z, -r_i)\, R_{z,-\theta_i}\, Trans\,(x, -d_i)\, R_{x,-\alpha_i} \\
=& \begin{bmatrix} R^t_{(i-1)i} & -R^t_{(i-1)i} p_{(i-1)i} \\ 0_{1\times 3} & 1 \end{bmatrix} \\
=& \begin{bmatrix} & & & -d_i \cos(\theta_i) \\ & R^t_{(i-1)i} & & d_i \sin(\theta_i) \\ & & & -r_i \\ 0 & 0 & 0 & 1 \end{bmatrix}
\end{aligned}
$$

Methodology for computing the DGM

4. DGM of the whole kinematic chain :
   Representation of DGM thanks to the transformation matrix $\overline{g}_{0n}$ in the case of a simple open-tree chain

$$X = f(q) \quad \text{where} \quad f = \overline{g}_{0N}$$

- Computation of $\overline{g}_{0n}$ obtained recursively

$$\overline{g}_{0N}(q) = \overline{g}_{01}(q_1)\dots\overline{g}_{(i-1)i}(q_i)\dots\overline{g}_{(N-1)N}(q_N)$$

- $q = [q_1 \dots q_N]^t$ being the vector of joint variables ;
- $X = [x_1 \dots x_M]^t$ being the vector of cartesian variables.
  - Several possibilities exist for parameterizing the orientation from vector X (using, for example, *Euler* angles computed from the direction cosines of R_{0n} to obtain $X = [p^x \, p^y \, p^z \, \alpha \beta \gamma]$).

Recommendations for choosing the frames :

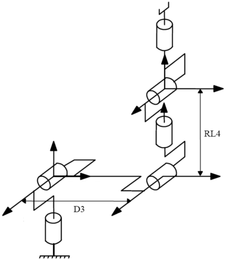
- $\mathscr{R}_0$ confounded with $\mathscr{R}_1$ when $q_1 = 0$ (cancellation of parameters α_1, d_1) ;
- X_N (free since Z_{N+1} does not exist) chosen colinear to X_{N-1} when $q_N = 0$;
- X_i chosen such that $r_i = 0$ or $r_{i+1} = 0$ when Z_i is parallel to Z_{i+1}.

DGM of serial robot

Example : case of robot RX130L from *Staubli*^{TM}

- Assignation of frames ;
- Parameterization according to *Khalil-Kleinfinger* convention ;
- Computation of $\overline{g}_{(i-1)i}$ for $i = 1, \ldots, 4$.

i	α_i	d_i	θ_i	r_i
1	?	?	?	?
2	?	?	?	?
3	?	?	?	?
4	?	?	?	?
5	?	?	?	?
6	?	?	?	?



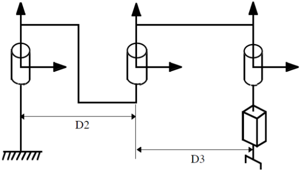
Example

DGM of serial robot

Example : case of robot SCARA IRB 910SC from ABB^{TM}

- Assignation of frames ;
- Parameterization according to *Khalil-Kleinfinger* convention.

i	α_i	d_i	θ_i	r_i
1	?	?	?	?
2	?	?	?	?
3	?	?	?	?
4	?	?	?	?



Example

Introduction

Definition

The direct kinematic model (DKM) of a robot manipulator gives the velocity of the end-effector $\dot{X}$ as a function of joint velocities $\dot{q}$:

$$\dot{X} = J(q)\,\dot{q}$$

where $J(q)$ denotes the Jacobian matrix of dimensions $M \times N$ and given by $\frac{\partial X}{\partial q}$.

- The same Jacobian matrix also appears in the direct differential model, which provides the differential displacement of the end-effector dX in terms of the differential variation of the joint variables dq :

$$dX = J(q)\,dq$$

- Interests of the Jacobian matrix for robots :
  - usefulness for singularities analysis and dimension of the reachable workspace of the robot ;
  - usefulness for numerically computing the solutions to the problem of Inverse Geometric Model (IGM) ;
  - usefulness for establishing in static the relationship between wrench exerted by the end-effector and forces/torques at the actuators level.
- Main methods to compute the Jacobian matrix :
  1. through direct derivation of DGM ;
  2. through velocities composition.

1- Jacobian matrix as derivative of the DGM

- The Jacobian matrix can be obtained by differentiating the DGM $X = f(q)$ using its partial derivatives, such that :

$$
{}^{0}J(q) = \begin{bmatrix} \frac{\partial f_1}{\partial q_1} & \cdots & \frac{\partial f_1}{\partial q_N} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_M}{\partial q_1} & \cdots & \frac{\partial f_M}{\partial q_N} \end{bmatrix}
$$

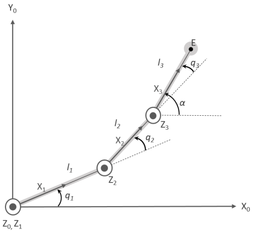
  - let note that the DGM f is given in frame $\mathscr{R}_0$ in this case ;
  - functions f_i are differentiable since it is formed by affines and trigonometric functions.
- let note that the Jacobian matrix J is configuration-dependent.
- Specificities of this approach :
  - convenient for simple robots having a reduced number of degrees of freedom ;
  - less practical for a general N degree-of-freedom robot.

1- Jacobian matrix as derivative of the DGM

Examples :

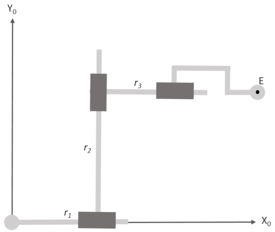
- Computation of homogeneous transformation matrix of end-effector in $\mathscr{R}_0$;
- Computation of ${}^0J$ through the derivation of the DGM.

Case of planar RRR robot



Example

Case of planar PPP robot



Example

1- Jacobian matrix as derivative of the DGM

Main steps for computing ${}^0J(q)$

$${}^0\mathscr{V}_{0N}(0_n) = \begin{bmatrix} {}^0V_{0N}(O_N) \\ {}^0\omega_{0N} \end{bmatrix} = \begin{bmatrix} {}^0J_v(q) \\ {}^0J_\omega(q) \end{bmatrix} \dot{q} = {}^0J(q)\,\dot{q}$$

where ${}^0V_{0N}(O_N)$ and ${}^0\omega_{0N}$ are the twist components of the end-effector velocity given in base frame.

- Linear velocity part :
  1. Computation of the homogeneous transformation matrix $\overline{g}_{0N}$ (used for DGM computation) ;
  2. Extraction of the position vector $p_{0N}(q)$ from the last column of $\overline{g}_{0N}$;
  3. Derivative of $p_{0N}(q)$ providing the upper submatrix of ${}^0J(q)$:

$${}^0V_{0N}(O_N) = \frac{d}{dt} p_{0N}(q) = \frac{d}{dt} O_0O_N = {}^0J_v(q)\,\dot{q}.$$

- Angular velocity part :
  1. Extraction of the matrix $R_{0N}(q)$ from $\overline{g}_{0N}$;
  2. Computation of ${}^0\widehat{\omega}_{0N}$ according to ${}^0\widehat{\omega}_{0N} = \dot{R}_{0N}R_{N0} = \left(\frac{\partial R_{0N}}{\partial q_1}\dot{q}_1 + \ldots + \frac{\partial R_{0N}}{\partial q_N}\dot{q}_N\right) R_{N0}$
  3. Term-by-term identification of ${}^0\omega_{0N} \in \mathbb{R}^3$ from the preproduct matrix ${}^0\widehat{\omega}_{0N}$ to compute the lower submatrix of ${}^0J(q)$:

$${}^0\omega_{0N} = {}^0J_\omega(q)\,\dot{q}$$

2- Jacobian matrix using velocity composition rule

Effect of the i^{th} joint on the end-effector velocity

- Expression of the cartesian velocity ${}^{i}\mathscr{V}_{iN}\left(O_N\right) = \begin{bmatrix} {}^{i}V_{iN}\left(O_N\right) \\ {}^{i}\omega_{iN} \end{bmatrix} \in \mathbb{R}^6$ of end-effector $\mathscr{R}_N$ due to the i^{th} joint of the poly-articulated chain :
  - ${}^{i}V_{iN}\left(O_N\right)$: velocity of origin O_N relative to $\mathscr{R}_i$, such that ${}^{i}V_{iN}\left(O_N\right) = \frac{dp_{iN}}{dt}$ with $p_{iN} = O_iO_N$, element of translation in the homogeneous transformation $\overline{g}_{iN}$;
  - ${}^{i}\omega_{iN}$: angular velocity of frame $\mathscr{R}_N$ relative to $\mathscr{R}_i$.
- Type of joint :
  - Case of a prismatic joint ($q_i = r_i$)

$$ {}^{i}\mathscr{V}_{iN}\left(O_N\right) = \begin{bmatrix} {}^{i}V_{iN}\left(O_N\right) \\ {}^{i}\omega_{iN} \end{bmatrix} = \begin{bmatrix} Z_i \\ 0_{3\times 1} \end{bmatrix} \dot{q}_i $$

  - Case of a revolute joint ($q_i = \theta_i$)

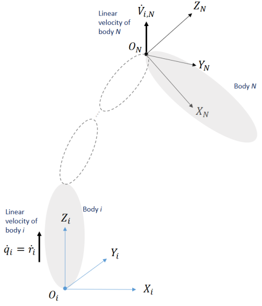
$$ {}^{i}\mathscr{V}_{iN}\left(O_N\right) = \begin{bmatrix} {}^{i}V_{iN}\left(O_N\right) \\ {}^{i}\omega_{iN} \end{bmatrix} = \begin{bmatrix} Z_i \times p_{iN} \\ Z_i \end{bmatrix} \dot{q}_i $$

- Remarks :
  - According to *Khalil-Kleinfinger* convention, the joint axis is along the vector $Z_i = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}^t$.
  - All the chain between joint i and extremity N is supposed to constitute one single rigid body.

2- Jacobian matrix using velocity composition rule

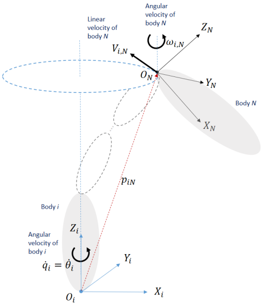
Effect of the i^{th} joint on the end-effector velocity

Case of a prismatic joint



$$\begin{bmatrix} {}^iV_{iN}(O_N) \\ {}^i\omega_{iN} \end{bmatrix} = \underbrace{\begin{bmatrix} Z_i \\ 0_{3\times 1} \end{bmatrix}}_{={}^iJ_i(q)} \dot{q}_i$$

Case of a revolute joint



$$\begin{bmatrix} {}^iV_{iN}(O_N) \\ {}^i\omega_{iN} \end{bmatrix} = \underbrace{\begin{bmatrix} Z_i \times p_{iN} \\ Z_i \end{bmatrix}}_{={}^iJ_i(q)} \dot{q}_i$$

2- Jacobian matrix using velocity composition rule

Effect of the i^{th} joint on the end-effector velocity

- Projection of cartesian velocity in base frame $\mathscr{R}_0$
  - Linear velocity ${}^0V_{iN}$ of the origin O_N w.r.t. $\mathscr{R}_i$ given in $\mathscr{R}_0$:

$$ {}^0V_{iN}\left(O_N\right) = R_{0i}\,{}^iV_{iN}\left(O_N\right) $$

  - Angular velocity ${}^0\omega_{iN}$ of $\mathscr{R}_N$ w.r.t. $\mathscr{R}_i$ given in $\mathscr{R}_0$:

$$ {}^0\omega_{iN}\left(O_N\right) = R_{0i}\,{}^i\omega_{iN}\left(O_N\right) $$

i.e.

$$ {}^0\mathscr{V}_{iN}\left(O_N\right) = \begin{bmatrix} {}^0V_{iN}\left(O_N\right) \\ {}^0\omega_{iN} \end{bmatrix} = {}^0J_i\left(q\right)\dot{q}_i $$

with

$$ {}^0J_i\left(q\right) = \begin{bmatrix} R_{0i} & 0_{3\times3} \\ 0_{3\times3} & R_{0i} \end{bmatrix} {}^iJ_i\left(q\right) $$

Remarks about dependency of ${}^0J_i$:

- with the reference frame in which the velocities are given (possibility to project the cartesian velocity in another frame if required) ;
- with the configuration q of manipulator.

2- Jacobian matrix using velocity composition rule

Methodology for the practical computation of ${}^{0}J\left(q\right)$

Using composition of the differents twists ${}^{0}\mathscr{V}_{iN}\left(O_N\right)$ for $i=1,\dots,N$, the Jacobian matrix that maps joint velocities $\dot{q}$ of the robot to the cartesian velocities $\dot{X}$ given in inertial reference frame $\mathscr{R}_0$ is given by :

$$
{}^{0}J\left(q\right)=\left[\begin{array}{ccccc} {}^{0}J_1\left(q\right) & \dots & {}^{0}J_i\left(q\right) & \dots & {}^{0}J_N\left(q\right) \end{array}\right]
$$

where ${}^{0}J_i\left(q\right)$ defines the cartesian velocity due to the action of the i^{th} joint given in frame $\mathscr{R}_0$, according to the type of joint :

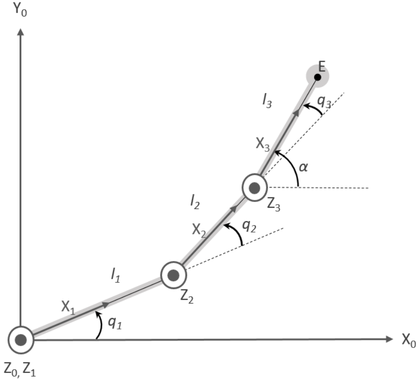
$$
{}^{0}J_i\left(q\right)=\begin{cases} \begin{bmatrix} R_{0i}Z_i \\ 0_{3\times 1} \end{bmatrix} & \text{if the } i^{th} \text{ joint is prismatic} \\ \\ \begin{bmatrix} R_{0i}\left(Z_i\times p_{iN}\right) \\ R_{0i}Z_i \end{bmatrix} & \text{if the } i^{th} \text{ joint is revolute} \end{cases}
$$

with :

- R_{0i} : rotation matrix of frame $\mathscr{R}_i$ w.r.t. $\mathscr{R}_0$;
- Z_i : unit vector of the i^{th} joint given in frame $\mathscr{R}_i$;
- p_{iN} : position vector from origin of frame $\mathscr{R}_i$ to origin of frame $\mathscr{R}_N$.

2- Jacobian matrix using velocity composition rule

Example : returning to the previous case of the planar RRR robot
Computation of the Jacobian matrix in base frame ${}^{0}J$



Example

Kinematic model associated with the task coordinates representation

Definition of analytical Jacobian

Let the end-effector position X_p and orientation X_r be specified in terms of a minimal number of parameters in the operational space (for example using *Euler* angles, RPY, etc.) and are given by $X = \begin{bmatrix} X_p & X_r \end{bmatrix}^T$ for representing the pose of frame $\mathscr{R}_N$ relative to frame $\mathscr{R}_0$, such that :

$$\begin{bmatrix} \dot{X}_p \\ \dot{X}_r \end{bmatrix} = \underbrace{\begin{bmatrix} \Omega_p & \mathbb{O}_{3\times 3} \\ \mathbb{O}_{3\times 3} & \Omega_r \end{bmatrix}}_{\Omega} \begin{bmatrix} {}^0V_{0N}\left(O_N\right) \\ {}^0\omega_{0N} \end{bmatrix}.$$

The *analytical Jacobian* $J_X\left(q\right)$ can be related to the previously computed Jacobian ${}^0J\left(q\right)$ (referred as *geometric* Jacobian) as follows :

$$\begin{bmatrix} \dot{X}_p \\ \dot{X}_r \end{bmatrix} = \underbrace{\Omega\,{}^0J\left(q\right)}_{J_X(q)}\dot{q}.$$

- This relationship shows that $J_X\left(q\right)$ and ${}^0J\left(q\right)$, in general, differ. However, when the dof cause rotations of the end-effector all about the same fixed axis in space (as it is the case for the previously studied RRR planar robot), the two Jacobian are the same.
- The matrix Ω_p is equal to $\mathbb{I}_{3\times 3}$ when the position of frame $\mathscr{R}_N$ is described by the Cartesian coordinates.
- In the example given in the previous chapter, let recall that we have already calculated $\Omega_{r_{\text{Euler}}}$ for *Euler* orientation representation :

$$\Omega_r = \begin{bmatrix} -sin(\alpha)cotg(\beta) & cos(\alpha)cotg(\beta) & 1 \\ cos(\alpha) & sin(\alpha) & 0 \\ \frac{sin(\alpha)}{sin(\beta)} & -\frac{cos(\alpha)}{sin(\beta)} & 0 \end{bmatrix}$$

Dimension of the task space for simply open-tree robot

Redundancy

For a given joint configuration q, the rank r of the Jacobian matrix $J \in \mathbb{R}^{M \times N}$ corresponds to the degrees-of-freedom of the robot task space (associated to the end-effector frame), i.e. to the dimension of the reachable task space in this *particular configuration* q.

The maximum rank r_{max} that takes the Jacobian matrix in all its possible configurations is the number of degrees-of-freedom M of the task space of a robot.

- If $M = N$ (N being the number of degrees-of-freedom of the robot, equal to the number of motorized joint in the case of a simply open-tree robot), then the robot is said to be *non-redundant*.
  - However, let note that a robot which is not redundant ($M = N$) may be *locally redundant* or *redundant w.r.t. a particular task* whose number of degrees of freedom, r, is less than the number of degrees of freedom of the robot N.
- If $M < N$, then the robot is *redundant* of order $N - M$.

Dimension of the task space for simply open-tree robot

Analysis of the range and null of the Jacobian matrix J

- The Jacobian describes the linear mapping from the joint velocity space to the end-effector velocity space.
  - the *range* of J is the subspace $\mathscr{R}(J)$ in $\mathbb{R}^M$ of the end-effector velocities that can be generated by the joint velocities, in the given manipulator posture ;
  - the *null* of J is the subspace $\mathscr{N}(J)$ in $\mathbb{R}^N$ of the joint velocities that do not produce any end-effector velocity, in the given manipulator posture.
- The following relation holds independently of the rank r of the matrix J :

$$\boxed{\dim(\mathscr{R}(J)) + \dim(\mathscr{N}(J)) = N}$$

and the dimension of the null of J, equal to $N - \dim(\mathscr{R}(J))$, is an indicator of the order of redundancy.

  - if the Jacobian has *full* rank, one has :

$$\dim(\mathscr{R}(J)) = M, \quad \dim(\mathscr{N}(J)) = N - M$$

    and the range of J spans the entire space $\mathbb{R}^M$.
  - Instead, if the Jacobian degenerates at a *singularity*, the dimension of the range space decreases ($\dim(\mathscr{R}(J)) = r < M$) while the dimension of the null space increases.

Example : returning to the previous case of planar PPP robot
Analysis of the rank and the null space of the Jacobian matrix ${}^0J$

Example

Dimension of the task space for simply open-tree robot

Singularity

Singularities are solutions of :

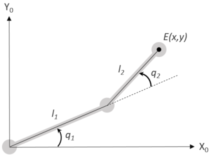
$$\begin{cases} det\left(J\right) = 0 & \text{for the non-redundant case (square matrix } J\text{),} \\ det\left(JJ^t\right) = 0 & \text{for the redundant case (non-square matrix } J\text{)} \end{cases}$$

where $det\left(\bullet\right)$ denotes the determinant of matrix $\bullet$.

- For certain joint configurations, it may happen that the rank r is inferior to M (the robot being redundant or not) : the number of degrees of freedom of the end-effector becomes less than the dimension of the task space. We say that the robot posseses a singularity or a local redundancy of order $M - r$.
  - this rank loss means that it is impossible to generate velocities along or around certain directions ;
  - close to this singularity, small velocities in the task space may imply significant (infinite) velocities in joint space.
- Types of singularities :
  1. Boundary singularity corresponding to points located at the border of the reachable workspace, i.e. robot in *extended* or *folded* configuration (possibility to avoid easily these configurations in bringing the manipulator away to the border of the reachable workspace)
  2. Internal singularity appearing inside the reachable workspace and generally caused by an alignment of two or more axes (more tricky problem as singular configurations can be reached in the workspace for a planned trajectory in the operational space)

Dimension of the task space for simply open-tree robot

Case study of the workspace of the planar RR robot (see previous example)

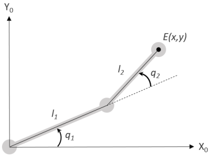


- Representation of its workspace considering no mechanical stops (assuming $L_1 > L_2$)
- Representation of singularity branches

Example

Dimension of the task space for simply open-tree robot

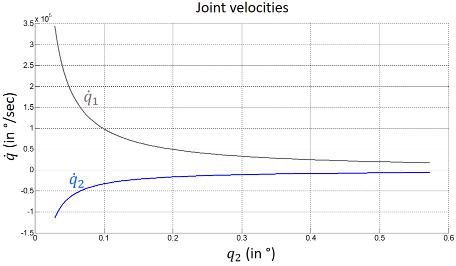
Case study of the workspace of the planar RR robot (see previous example)



- Representation of its workspace considering no mechanical stops (assuming $L_1 > L_2$)
- Representation of singularity branches

Example

- Influence of the closeness of singular configuration on the joint velocity



Evolution of joint velocity $\dot{q} = {}^0J^{-1}(q)\,\dot{X}_d$ enabling to reach the cartesian velocity $\dot{X}_d = \begin{bmatrix} -1 & -1 \end{bmatrix}^t$ around the joint configuration $q_1 = 0$ and $0,03^\circ < q_2 < 0,55^\circ$

Transmission of velocities between joint and task spaces

Recall on decomposition in singular values (function *SVD* in *Matlab*^{TM})
Let consider J of dimensions $m \times n$ and of rank r for a given configuration. There exists orthogonal matrices $U \in \mathbb{R}^{m \times m}$ and $V \in \mathbb{R}^{n \times n}$, such that

$$J = U \Sigma V^t$$

The matrix $\Sigma \in \mathbb{R}^{m \times n}$ takes the following form :

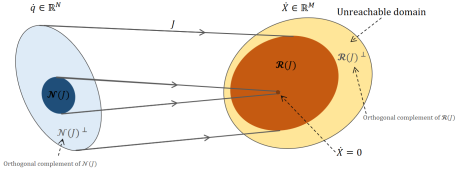
$$\Sigma = \begin{bmatrix} S & 0_{r \times (n-r)} \\ 0_{(m-r) \times r} & 0_{(m-r) \times (n-r)} \end{bmatrix}$$

$S \in \mathbb{R}^{r \times r}$ is the diagonal matrix computed with the non-null singular values σ_i of J, ordered in a decreasing way : $\sigma_1 \geq \ldots \geq \sigma_r$.

- The singular values of J are equal to the square roots of the eigenvalues λ_i of the product $J^t J$: $\sigma_i = \sqrt{\lambda_i \left(J^t J\right)}$;
- The matrix V is constituted by the eigenvectors of $J^t J$;
- The matrix U is constituted by the eigenvectors of JJ^t ;
- Since $\sigma_i = 0$ for $i > r$, the direct kinematic model becomes : $\dot{X} = \sum_{i=1}^{r} \sigma_i U_i V_i^t \dot{q}$.

Transmission of velocities between joint and task spaces

Null and range spaces of J



- Orthonormal basis for the subspace of $\dot{q}$ generating an end-effector velocity : $(V_1, \ldots, V_r)$;
- Orthonormal basis of null space of J giving $\dot{X} = 0$: $(V_{r+1}, \ldots, V_n)$;
- Orthonormal basis for the set of the achievable end-effector velocities $\dot{X}$, defining the range space of J : $(U_1, \ldots, U_r)$;
- Orthonormal basis for the subspace composed of the set of $\dot{X}$ that cannot be generated by the robot, defining the complement of the range space : $(U_{r+1}, \ldots, U_m)$.

Transmission of velocities between joint and task spaces

Velocity transmission performance

Assuming the joint velocity norm defined by $\dot{q}^t\dot{q} \leq 1$ (unit hyper-sphere), the resulting velocity in task space are given by the quadratic form (representing an ellipsoid in $\mathbb{R}^{\dot{X}}$)[30] :

$$\dot{X}^t \left(JJ^t\right)^{-1} \dot{X} \leq 1$$

The ellipsoid has a form and an orientation defined through JJ^t :

- the principal axes of the ellipsoid are given by the vectors $U_1, \ldots, U_m$ (i.e. eigenvectors of de JJ^t) ;
- the lengths of the principal axes are determined by the singular values $\sigma_1 \geq \ldots \geq \sigma_m$ (i.e. root square of the eigenvalues of JJ^t) ;
- its volume is an indicator of the robot capacity to generate velocity (velocity manipulability), given by : $\mathscr{W} = \sqrt{det\left(J\left(q\right)J^t\left(q\right)\right)}$ (or $|det\left(J\left(q\right)\right)|$ in the non-redundant case), thus $\mathscr{W} = \prod_{i=1}^{r} \sigma_i \geq 0$.

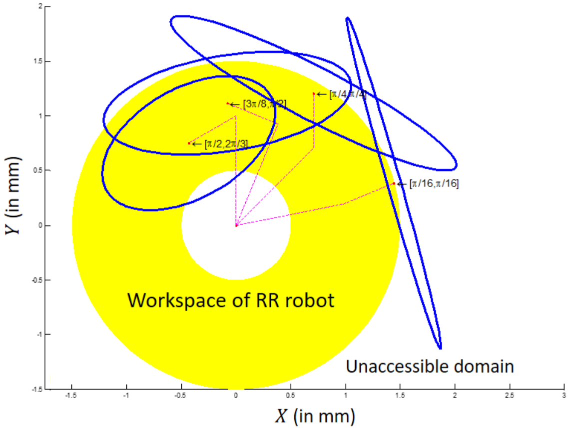
From the SVD decomposition of J, it follows that :

$$J = U\Sigma V^t \Rightarrow JJ^t = U\Sigma^2 U^t \Rightarrow \left(JJ^t\right)^{-1} = U\Sigma^{-2} U^t$$

Transmission of velocities between joint and task spaces

Velocity transmission performance

Case study of the RR robot workspace



Transmission of forces/torques between joint and task spaces

Mapping of an external wrench into joint torques

The interaction forces between the robot and its environment can be calculated directly using the kinematic Jacobian matrix. The joint forces/torques Γ_e resulting from the application of the wrench ${}^0\mathscr{F}_e$ giving the forces applied at the end of the poly-articulated chain is equal to

$$\Gamma_e = {}^0J\left(q\right)^t\,{}^0\mathscr{F}_e$$

where ${}^0\mathscr{F}_e$ denotes the static wrench applied by the robot end-effector on the environment given in coordinates frame $\mathscr{R}_0$:

$${}^0\mathscr{F}_e = \left[f_x, f_y, f_z, m_x, m_y, m_z\right]^t$$

and ${}^0J$ the Jacobian matrix computed at the application point of the external wrench on the robot.

▸ Demonstration

- ${}^0\mathscr{F}_e$ is a vector of dimension $M \times 1$ ($M = 6$ in general) composed by forces f and torques m applied to the manipulator ;
- Γ_e is a vector of dimension $M \times 1$ composed by forces and/or torques applied to the n joints.

Transmission of forces/torques between joint and task spaces

Performance of forces/torques transmission [28]
Analogously, we can study the force transmission performance using a force manipulability ellipsoid, which corresponds to the set of achievable wrench in the task space $\mathbb{R}^M$ corresponding to the constraint $\Gamma^t\Gamma \leq 1$. The force ellipsoid is defined by :

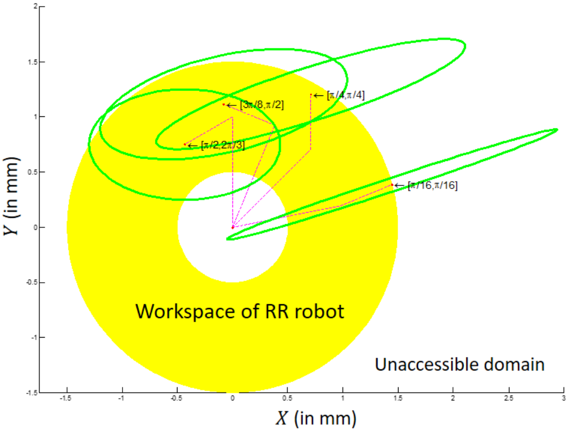
$$\mathscr{F}^t \left(JJ^t\right) \mathscr{F} \leq 1.$$

- Characteristics of the ellipsoid :
  - Same principal axes of the ellipsoids in velocity and force : $U_1, \ldots, U_m$ (eigenvectors of JJ^t) ;
  - Length of axes inversely proportional to singular values σ_i of J
    The velocity is controlled most accurately in the direction where the robot can resist large force disturbances, and force is most accurately controlled in the direction where the robot can rapidly adapt its motion.
- Let note that the singular configurations for applying cartesian velocity are identical to those for applying cartesian force (since $Rank\,(J) = Rank\,(J^t)$).

Transmission of forces/torques between joint and task spaces

Performance of forces/torques transmission

Case study of the RR robot workspace



Transmission of forces/torques between joint and task spaces

Duality force/velocity

- Subspaces analysis
  - The torques of the actuators are uniquely determined for an arbitrary wrench ${}^{0}\mathscr{F}_e$; the range space of J^T, denoted as $\mathscr{R}\left(J^T\right)$, is the set of Γ_e balancing the static wrench ${}^{0}\mathscr{F}_e$ according to $\Gamma_e = {}^{0}J(q)^t\,{}^{0}\mathscr{F}_e$;
  - For a zero Γ_e, the corresponding static wrench can be non-zero ; we thus define the null space of J^T, $\mathscr{N}\left(J^T\right)$, as the set of static wrenches that do not require actuator torques in order to be balanced.
- Following the previous Singular Values Decomposition (SVD) of matrix J for velocity analysis, the SVD of matrix J^T for force analysis leads to :

$$J^T = V\Sigma U^T$$

so that,

$$\mathbb{R}^M = \mathscr{R}(J) + \mathscr{R}(J)^{\perp} = \mathscr{R}(J) + \mathscr{N}\left(J^T\right),$$

$$\mathbb{R}^N = \mathscr{R}\left(J^T\right) + \mathscr{R}\left(J^T\right)^{\perp} = \mathscr{R}\left(J^T\right) + \mathscr{N}(J).$$

Physical insights

- $\mathscr{N}\left(J^T\right) = \mathscr{R}(J)^{\perp}$: in this configuration, the endpoint wrench is borne by the structure of the robot without requiring balancing actuator torques, while it also refers to the set of directions along which the robot cannot generate velocity.
- $\mathscr{R}\left(J^T\right)^{\perp} = \mathscr{N}(J)$: in this configuration, some joint torques cannot be compensated by the external wrench, while it also refers to the joint velocities that cannot generate cartesian velocities.