

# **Modélisation 3D**

**MEA4**

**P. Fraisse, 15/09/2021**

# Plan du cours

1. Introduction
2. Outils mathématiques
3. Modélisation géométrique
4. Modélisation cinématique
5. Modélisation dynamique
6. Génération de mouvement

# Définitions

## Robot

- Le mot « Robot » a été crée en **1920** par Karel Capek auteur de la pièce de théâtre R.U.R. (Rossum Universal Robots) à partir du mot Tchèque Robots qui signifie : « travail, besogne, corvée ».
- **1956** : Premier robot industriel avec la société Unimation Inc. Créeée par Joseph Engelberger associé à Georges Devol détenteur du brevet du premier robot industriel Unimate.
- Cette société a été racheté par Westinghouse et renvendu ensuite à Stäubli (Société Suisse installée à Faverges, France).

# Définitions

- **Robot** - Merriam-Webster's dictionary:
  - a machine that resembles a living creature in being capable of moving independently (as by walking or rolling on wheels) and performing complex actions (such as grasping and moving objects).
  - a device that automatically performs complicated, often repetitive tasks (as in an industrial assembly line).
- **Cobot** : Collaborative robot, a robot intended to interact physically with humans in a shared workspace.

# Introduction

## Robotique de manipulation

**Human-robot collaborative assembly  
for a manufacturing application**

**A. Cherubini R. Passama P. Fraisse A. Crosnier**

**LIRMM - University of Montpellier CNRS  
Montpellier, France**

# Introduction

## Manipulation à deux bras

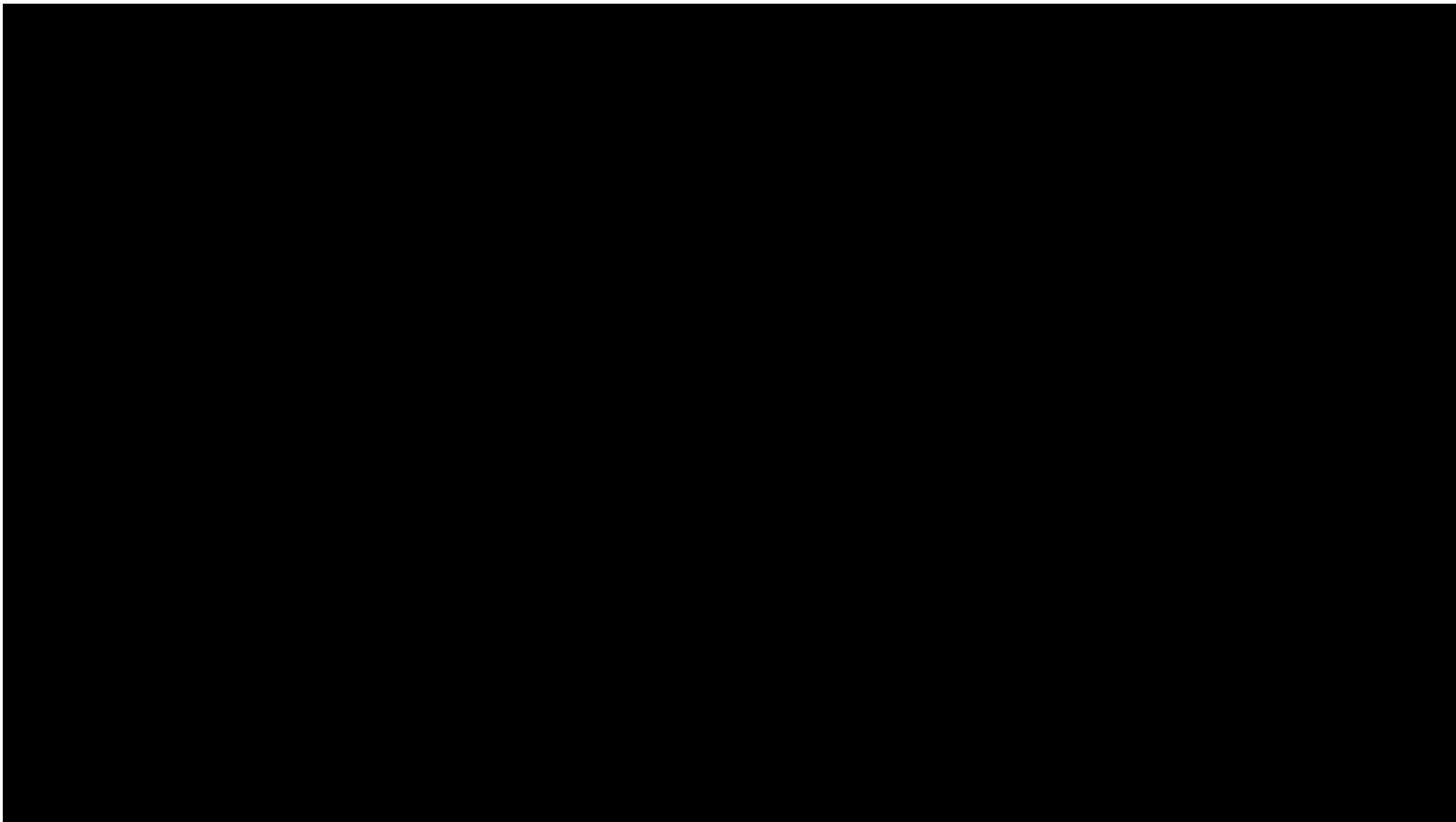
**Dual-arm relative tasks  
performance using  
sparse kinematic control**

S. Tarbouriech, B. Navarro, P. Fraisse, A. Crosnier, A. Cherubini



# **Introduction**

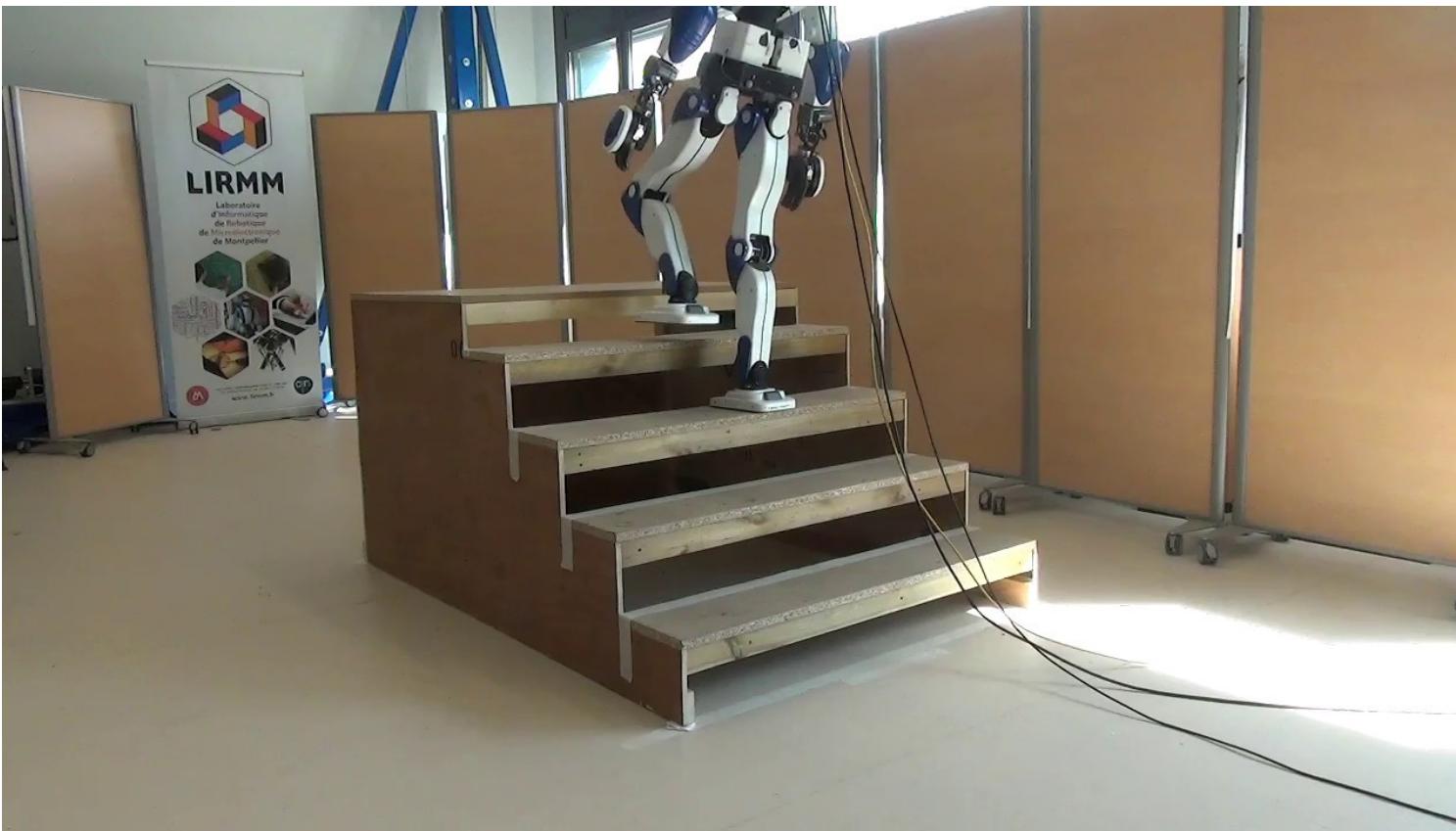
## **Evitement de collision**



# Interaction Humain-Robot



# Robotique humanoïde



# Introduction

## Modélisation du mouvement humain



# Définitions

## Robots souples

Design, Characterization, and  
Mechanical Programming of  
Fabric-Reinforced Textile Actuators  
for a Soft Robotic Hand

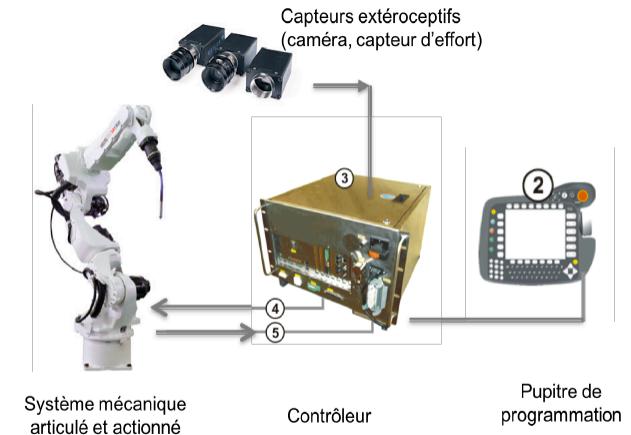
Pham H. Nguyen, Francisco L. Arellano, Wenlong Zhang\*,  
Panagiotis Polygerinos

2019



# Eléments constitutif d'un robot manipulateur

- Une structure mécanique poly-articulée actionnée,
- Un contrôleur, des capteurs extéroceptifs et proprioceptifs.
- Un outil.
- La structure mécanique est articulée à travers deux types de maisons différents :
- La liaison pivot qui est un mouvement de rotation autour d'un axe. Elle possède 1ddl. Elle est aussi appelée liaison Rotoïde.
- La liaison glissière qui est un mouvement de translation selon un axe. Elle possède 1ddl. Elle est aussi appelée liaison Prismatique.



# Robot manipulateur

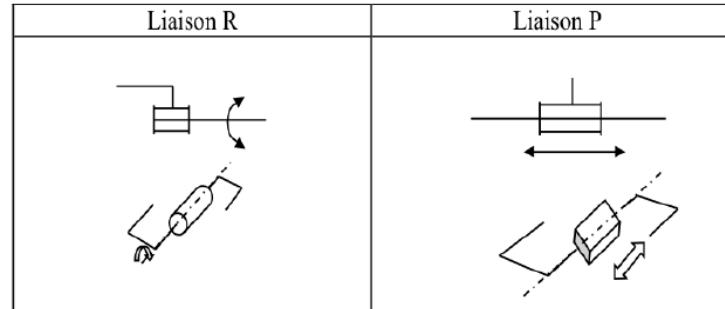


FIGURE 2.7 – Liaisons R et P

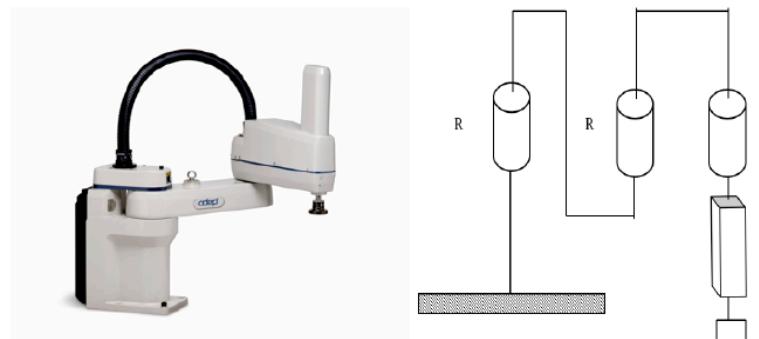


FIGURE 2.8 – Robot SCARA RRRP (Adept)

# Espace affine 3D

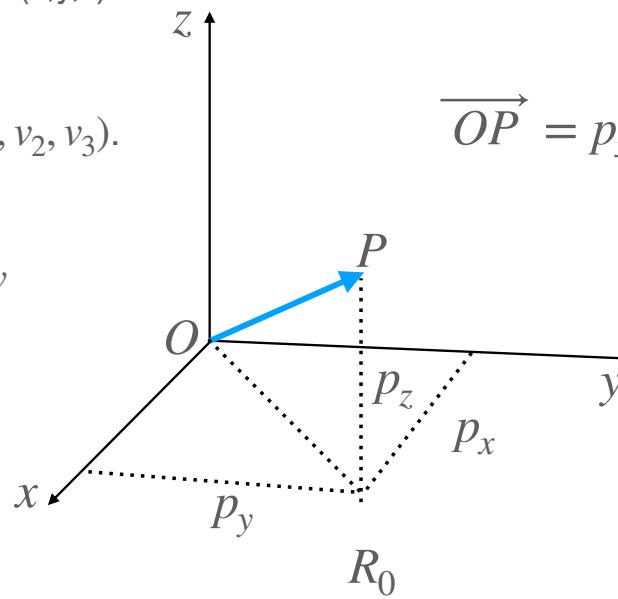
Un point d'origine O associé à une base orthonormée (x,y,z)

Exercice :

Soit les vecteurs  $\vec{u} = (u_1, u_2, u_3)$  et  $\vec{v} = (v_1, v_2, v_3)$ .

- 1) Calculer le produit scalaire :  $u \cdot v = u^T v$
- 2) Calculer le produit vectoriel :  $u \times v = S(u)v$
- 3) Déterminer la matrice  $S(u)$

$$\overrightarrow{OP} = p_x \vec{x} + p_y \vec{y} + p_z \vec{z}$$



# Matrice de rotation

- Une matrice de rotation est une matrice orthogonale de déterminant 1.
- L'ensemble des matrices de rotation de taille fixée forme un groupe appelé groupe spécial orthogonal (ex:  $\text{SO}(2)$ ).
- $Rot(\theta) = \begin{pmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{pmatrix}$
- Exercice 1 : Calculer l'expression de  $R(\theta)^{-1}$  ainsi que l'expression de  $R(\theta)^T$ . Conclusion.
- Exercice 2 : Calculer l'expression de  $R(\theta_1)R(\theta_2)$ . En déduire que  $R(\theta)^n = R(n\theta)$ .

# Matrice de rotation

- En dimension 3 on a :

$$\bullet \quad R_x(\theta) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos \theta & -\sin \theta \\ 0 & \sin \theta & \cos \theta \end{pmatrix}$$

- Calculer et représenter la rotation autour de x du vecteur u.
- En déduire  $R_y(\theta)$  et  $R_z(\theta)$ .
- $R_x(\theta)$  :  $R_x(\theta)$  tourne l'axe y vers l'axe z.
- $R_y(\theta)$  :  $R_y(\theta)$  tourne l'axe z vers l'axe x.
- $R_z(\theta)$  :  $R_z(\theta)$  tourne l'axe x vers l'axe y.

# Coordonnées et transformations homogènes

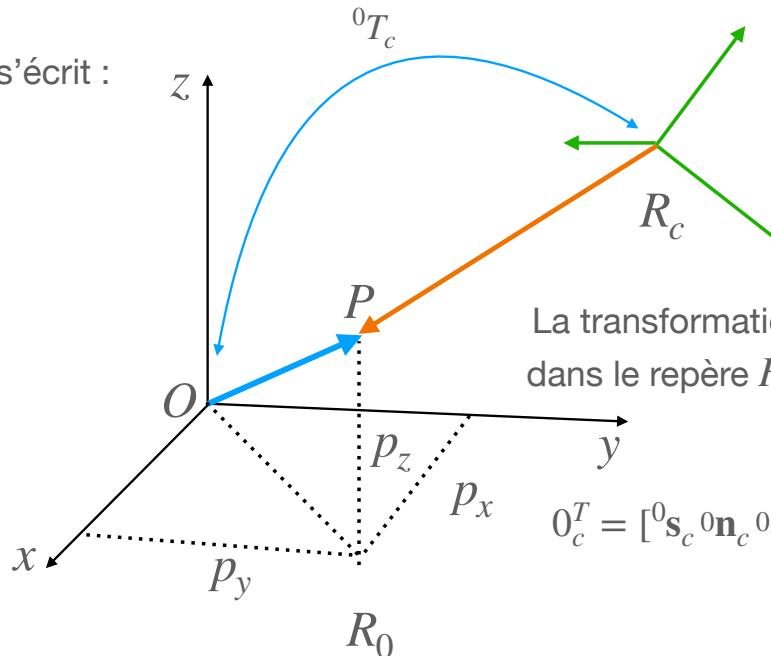
Un vecteur unitaire  $\vec{u}$  en coordonnées homogènes s'écrit :

$$u = \begin{pmatrix} u_x \\ u_y \\ u_z \\ 0 \end{pmatrix}$$

Un point  $p = (x \ y \ z)^T$  est représenté par :

$$p = \begin{pmatrix} p_x \\ p_y \\ p_z \\ 1 \end{pmatrix}$$

- 1) Ecrire la transformation du point P défini dans le repère R<sub>0</sub> vers le repère R<sub>c</sub>
- 2) Ecrire la transformation du vecteur u défini dans le repère R<sub>0</sub> et u<sub>c</sub> dans le repère R<sub>c</sub>



La transformation homogène exprimant le repère  $R_c$  dans le repère  $R_0$  s'exprime par :

$${}^0T_c = [{}^0\mathbf{s}_c \ {}^0\mathbf{n}_c \ {}^0\mathbf{a}_c \ {}^0P_c] = \begin{pmatrix} s_x & n_x & a_x & P_x \\ s_y & n_y & a_y & P_y \\ s_z & n_z & a_z & P_z \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$${}^0T_C = \begin{pmatrix} {}^0A_C & {}^0P_C \\ {}^0\mathbf{1}_{3 \times 3} & 1 \end{pmatrix}$$

# Composition des transformations

- Transformation homogène :  ${}^0T_C = \begin{pmatrix} {}^0A_C & {}^0P_C \\ 0_{1 \times 3} & 1 \end{pmatrix}$
- Transformations homogènes particulières :
  - Rotation pure :  ${}^0P_C = 0$
  - Translation pure :  ${}^0A_C = \mathbb{I}$
- Le produit des matrices homogènes n'est pas commutatif :
$${}^0T_2 = {}^0T_1 {}^1T_2 = \begin{pmatrix} {}^0A_1 & {}^0P_1 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} {}^1A_2 & {}^1P_2 \\ 0_{1 \times 3} & 1 \end{pmatrix} = ?$$

# Composition des transformations

- Calculer la translation d'un vecteur  $V=(Vx, Vy, Vz)$  en utilisant la matrice homogène suivante :  ${}^0T_1 = \begin{pmatrix} \mathbb{I} & t_{3 \times 1} \\ 0_{1 \times 3} & 1 \end{pmatrix}$  avec  $t_{3 \times 1} = \begin{pmatrix} t_x \\ t_y \\ t_z \end{pmatrix}$
- Le produit des matrices homogènes n'est pas commutatif :
- Exercice : Calculer l'inverse de la transformation homogène suivante :

- ${}^0T_C = \begin{pmatrix} {}^0A_C & {}^0P_C \\ 0_{1 \times 3} & 1 \end{pmatrix}$ ,  ${}^0T_C^{-1}=?$ , tel que

# Représentation des orientations

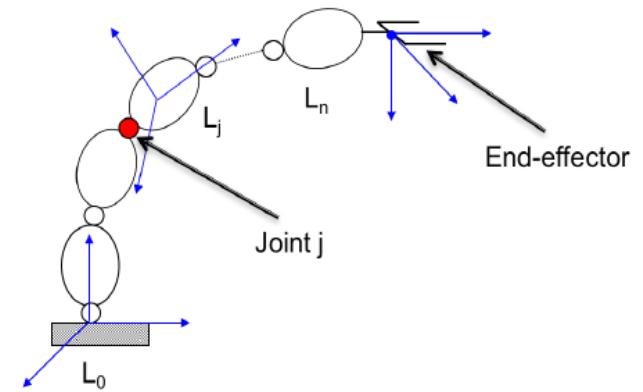
- Utilisation de la matrice de rotation orthonormée : matrice 3x3 avec 9 paramètres appelés cosinus directeurs.
- Représentation redondante et non minimale.
- Représentation de 3 angles de rotation autour des axes principaux.
- Représentation RTL : Roulis (Roll, x), Tangage (Pitch, y), Lacet (Yaw, z)
  - ${}^0A_n = Rot(z, \phi)Rot(y, \theta)Rot(x, \psi)$  : convention ZYX
  - Représentation d'Euler
  - ${}^0A_n = Rot(z, \phi)Rot(x, \theta)Rot(z, \psi)$  : convention ZXZ
  - Exercice : Ecrire la matrice de rotation en 3D utilisant la convention RTL

# Exercices

- Exercice 5 : Ecrire une fonction Matlab permettant de visualiser le repère  $R_C$  dans  $R_0$  à partir de la matrice homogène  ${}^0T_C$ . Etablir l'expression de la matrice inverse de  ${}^0T_C$  et écrire une fonction Matlab permettant le calcul de la matrice inverse à partir de  ${}^0T_C$ .
- Exercice 6 : Représenter en 3D l'effet des transformations élémentaires  $Rot(x, \phi)$ ,  $Rot(y, \theta)$ ,  $Rot(z, \psi)$ ,  $Trans(x, a)$ ,  $Trans(y, b)$ ,  $Trans(z, c)$ . Ecrire les fonctions Matlab correspondant à l'ensemble de ces transformations élémentaires.
- Représenter un vecteur dans un repère R0 et ensuite transformer ce vecteur dans un repère Rc ayant une translation selon x,y et z et 3 rotations (Convention RTL).

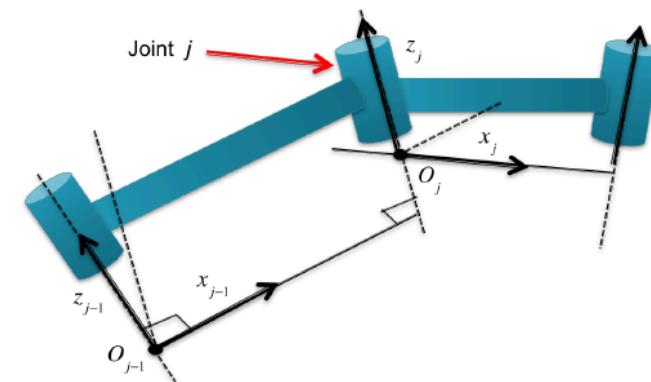
# Modélisation Géométrique

- Chaîne cinématique :
- Pour déterminer la pose de chaque corps,
- On exprime la position et l'orientation du repère  $R_j$  par rapport à  $R_{j-1}$ .
- On exprime la matrice homogène  ${}^{j-1}T_j$  décrivant le repère  $R_j$  dans  $R_{j-1}$
- 12 paramètres utiles.
- Il existe 2 descriptions principales pour spécifier  ${}^{j-1}T_j$  :
  - 1) Les paramètres de Denavit-Hartenberg (DH) standard ou modifié,
  - 2) Description URDF (Unified Robot Description Format)



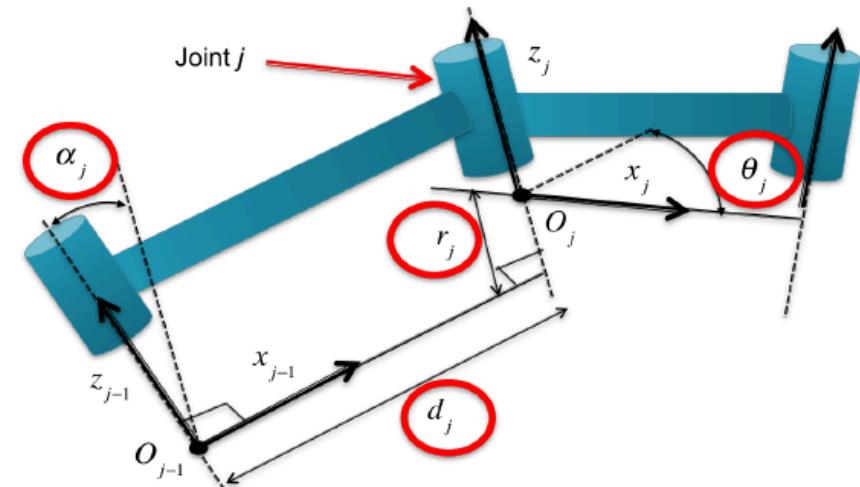
# Modélisation Géométrique

- Paramètres DH modifiés :
- L'axe  $z_j$  est orienté selon l'axe de l'articulation  $j$
- L'axe  $x_{j-1}$  est orienté selon la direction perpendiculaire aux axes  $z_{j-1}$  et  $z_j$ .
- Cette direction peut-être obtenue par le produit vectoriel  $z_{j-1} \times z_j$  lorsque les axes ne sont pas parallèles.



# Modélisation Géométrique

- $\alpha_j$  est l'angle entre les axes  $z_{j-1}$  et  $z_j$  et définit une rotation autour de  $x_{j-1}$
- $d_j$  est la distance entre les axes  $z_{j-1}$  et  $z_j$ , et définit une translation selon l'axe  $x_{j-1}$
- $\theta_j$  est l'angle entre les axes  $x_{j-1}$  et  $x_j$  et définit une rotation autour de l'axe  $z_j$
- $r_j$  est la distance entre les axes  $x_{j-1}$  et  $x_j$ , et définit une translation selon l'axe  $z_j$ .



# Modélisation Géométrique

- **Remarque importante :**
- 3 des 4 paramètres sont toujours constants :  $\alpha_j$ ,  $d_j$ , et  $\theta_j$  ou  $r_j$
- Le paramètre restant n'est pas constant et représente la variable articulaire.
- **Pour une articulation de type R**,  $\theta_j$  est variable et on a :  $q_j = \theta_j$
- **Pour une articulation de type P**,  $r_j$  est variable et on a :  $q_j = r_j$

# Modélisation Géométrique

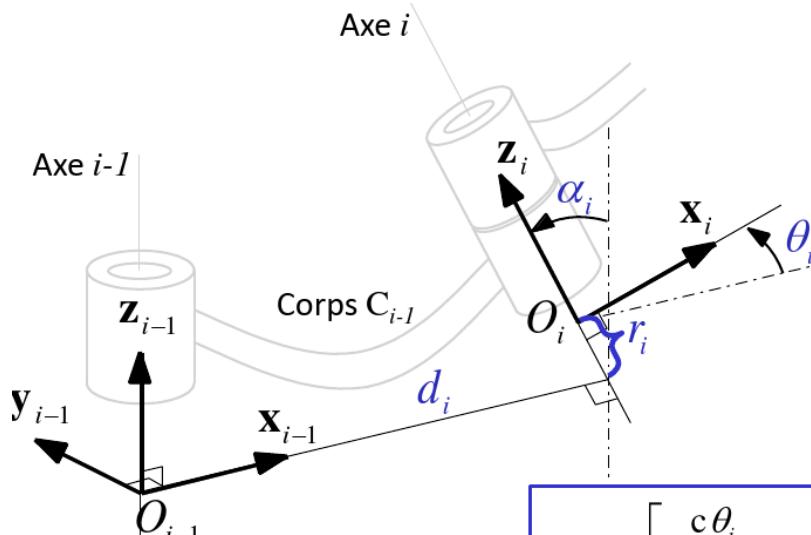
- Expression de la matrice de transformation  ${}^{j-1}T_j$  :
- 1) On applique la rotation  $Rot(x, \alpha_j)$  qui permet d'aligner l'axe  $z_{j-1}$  sur l'axe  $z_j$
- 2) On applique la translation  $Trans(x, d_j)$  qui permet de mettre en coincidence l'axe courant  $z$  avec  $z_j$
- 3) On applique la rotation  $Rot(z, \theta_j)$  qui permet d'aligner l'axe courant  $x = x_{j-1}$  avec  $x_j$
- 4) On applique la translation  $Trans(z, r_j)$  qui permet de mettre en coincidence l'axe courant  $x$  avec  $x_j$  afin que l'origine du repère coincide avec le point  $O_j$

# Modélisation Géométrique

- ${}^{j-1}T_j = Rot(x, \alpha_j)Trans(x, d_j)Rot(z, \theta_j)Trans(z, r_j)$
- Calculer l'expression de  ${}^{j-1}T_j$
- Exercice 9. Cf Polycop p. 32.
- Exercice 10. Cf. Polycop. P.33.

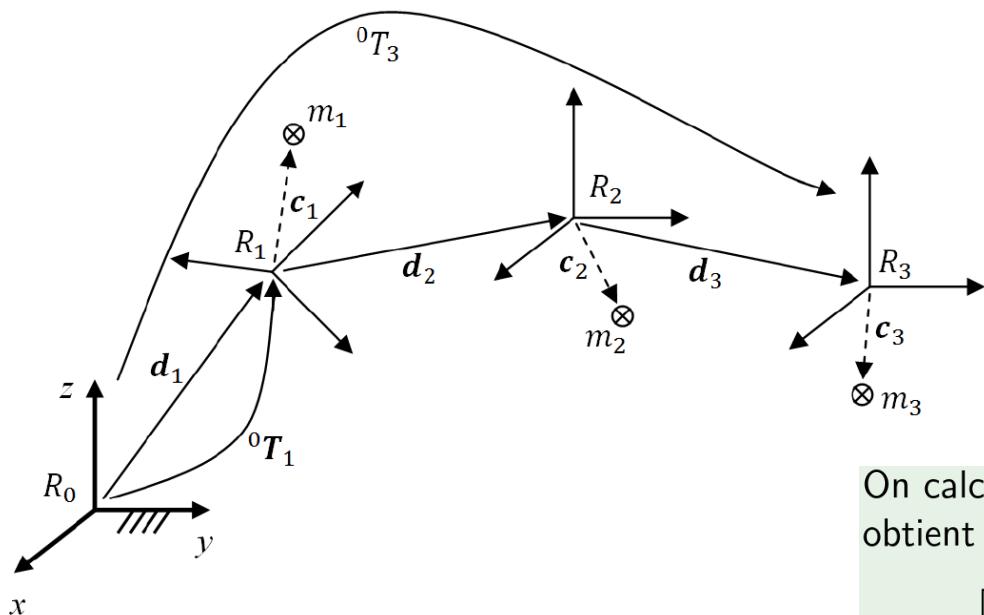
# Modélisation Géométrique

- $\alpha_j$  est l'angle entre les axes  $z_{j-1}$  et  $z_j$  et définit une rotation autour de  $x_{j-1}$
- $d_j$  est la distance entre les axes  $z_{j-1}$  et  $z_j$ , et définit une translation selon l'axe  $x_{j-1}$
- $\theta_j$  est l'angle entre les axes  $x_{j-1}$  et  $x_j$  et définit une rotation autour de l'axe  $z_j$
- $r_j$  est la distance entre les axes  $x_{j-1}$  et  $x_j$ , et définit une translation selon l'axe  $z_j$ .



$${}^{i-1}\mathbf{T}_i = \begin{bmatrix} c\theta_i & -s\theta_i & 0 & d_i \\ c\alpha_i s\theta_i & c\alpha_i c\theta_i & -s\alpha_i & -r_i s\alpha_i \\ s\alpha_i s\theta_i & s\alpha_i c\theta_i & c\alpha_i & r_i c\alpha_i \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

# Modélisation Géométrique : CdM



$$\vec{OCdM} = \frac{1}{M} \sum_{i=1}^N m_i \vec{Oc_i}$$

On calcule en utilisant les Matrices Homogènes l'expression du CdM. On obtient :

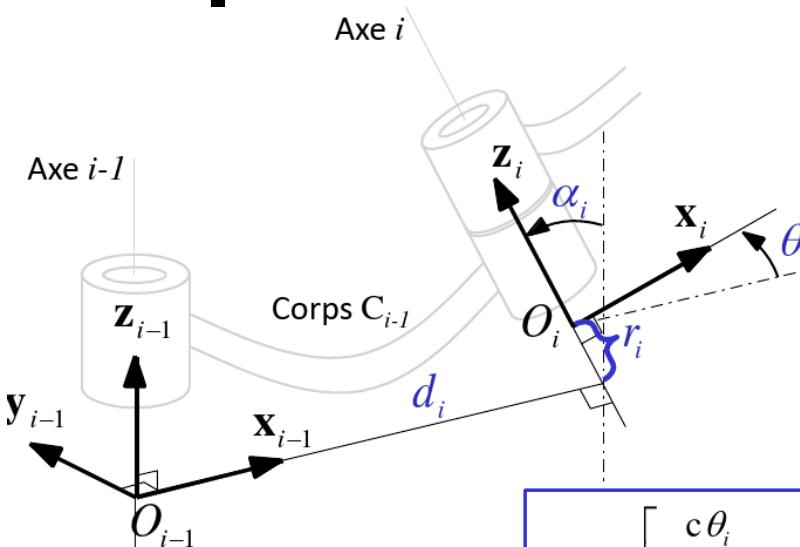
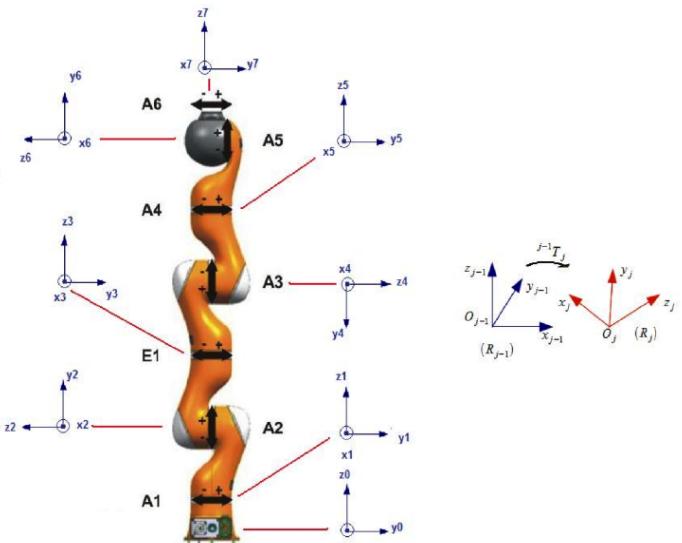
$$\begin{bmatrix} \mathbf{C}_{\text{CdM}} \\ 1 \end{bmatrix} = \frac{m_1}{M} \mathbf{T}_1 \begin{bmatrix} \mathbf{c}_1 \\ 1 \end{bmatrix} + \frac{m_2}{M} \mathbf{T}_1 \mathbf{T}_2 \begin{bmatrix} \mathbf{c}_2 \\ 1 \end{bmatrix} + \frac{m_3}{M} \mathbf{T}_1 \mathbf{T}_2 \mathbf{T}_3 \begin{bmatrix} \mathbf{c}_3 \\ 1 \end{bmatrix}$$

Démontrer que le CdM peut s'exprimer comme :

$$\mathbf{C}_{\text{CdM}} = \mathbf{r}_1 + \mathbf{A}_1 \mathbf{r}_2 + \mathbf{A}_1 \mathbf{A}_2 \mathbf{r}_3 + \mathbf{A}_1 \mathbf{A}_2 \mathbf{A}_3 \mathbf{r}_4$$

Déterminer l'expression des  $r_i$ .

# Modélisation Géométrique : Kuka LWR



$${}^{i-1}\mathbf{T}_i = \begin{bmatrix} c\theta_i & -s\theta_i & 0 & d_i \\ c\alpha_i s\theta_i & c\alpha_i c\theta_i & -s\alpha_i & -r_i s\alpha_i \\ s\alpha_i s\theta_i & s\alpha_i c\theta_i & c\alpha_i & r_i c\alpha_i \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Articulation	$\alpha$	$d(m)$	$\theta$	$r(m)$	$\theta_{min}$	$\theta_{max}$	$\dot{\theta}_{max}$
A1	0	0	$\theta_1$	0,3105	-170°	170°	112,5°/s
A2	$\frac{\pi}{2}$	0	$\theta_2$	0	-120°	120°	112,5°/s
E1	$-\frac{\pi}{2}$	0	$\theta_3$	0,4	-170°	170°	112,5°/s
A3	$-\frac{\pi}{2}$	0	$\theta_4$	0	-120°	120°	112,5°/s
A4	$\frac{\pi}{2}$	0	$\theta_5$	0,39	-170°	170°	180°/s
A5	$\frac{\pi}{2}$	0	$\theta_6$	0	-120°	120°	112,5°/s
A6	$-\frac{\pi}{2}$	0	$\theta_7$	0,078	-170°	170°	112,5°/s

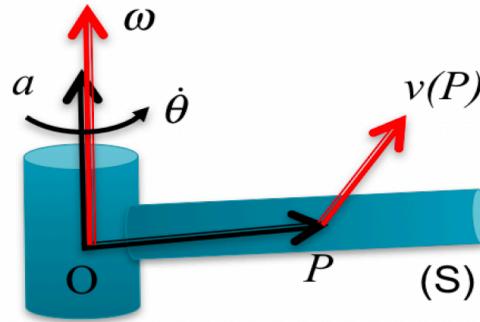
```

alpha=[0.0,pi/2,-pi/2,-pi/2,pi/2,pi/2,-pi/2];
d=zeros(7);
r=[0.3105,0.0,0.4,0.0,0.39,0.0,0.078]
theta=zeros(7);
m=[2.7,2.7,2.7,2.7,2.7,2.7,0.3];
c=zeros(4,7)
c[2,:]=[-8.70e-3, 8.7e-3, 8.7e-3, -8.7e-3,-8.2e-3, -7.6e-3, 0.0];
c[3,:]=[-1.461e-2,1.461e-2,-1.461e-2, 1.461e-2,-3.48e-2, 1.363e-3, 0.0];
c[4,:]=[1.0,1.0,1.0,1.0,1.0,1.0,1.0];

```

# Cinématique

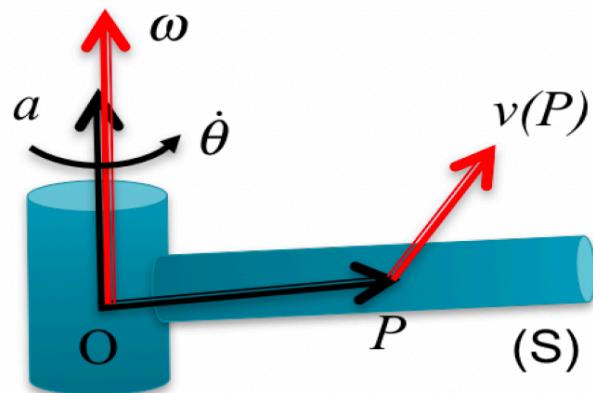
- $\omega = \vec{a}\dot{\theta}$



- Relation entre vitesse angulaire et dérivées des angles d'orientations :
  - $\omega = \omega_\phi + \omega_\theta + \omega_\psi$ ,
  - Exercice : Donner l'expression de  $\omega$  en fonction de  $\dot{\phi}, \dot{\theta}, \dot{\psi}$

# Cinématique

- Soit le point P situé sur un corps S :  $v(P) = v(O) + \omega \times \overrightarrow{OP}$



- Avec  $v(O)$  la vitesse linéaire du point O.

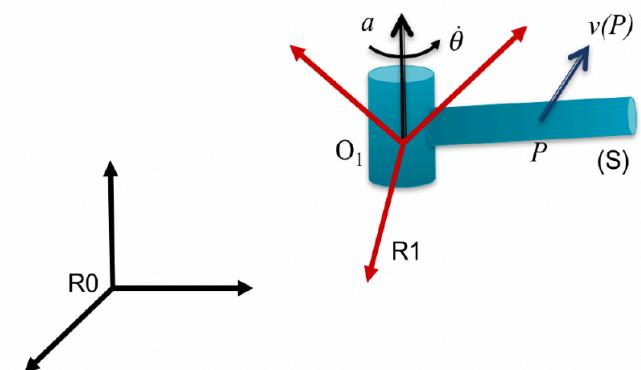
# Cinématique

- Exercice 7 :
- On considère un solide (S) dont l'pose est caractérisée par le repère R1 dans le repère absolu R0, et défini par la matrice homogène :

- ${}^0T_1 = \begin{pmatrix} {}^0A_1 & {}^0O_1 \\ 0 & 1 \end{pmatrix}$ , avec  ${}^0A_1 = \begin{pmatrix} 0 & 0 & -1 \\ 1 & 0 & 1 \\ 0 & -1 & 0 \end{pmatrix}$  et  ${}^0O_1 = (1 \ 1 \ 1)^T$ .

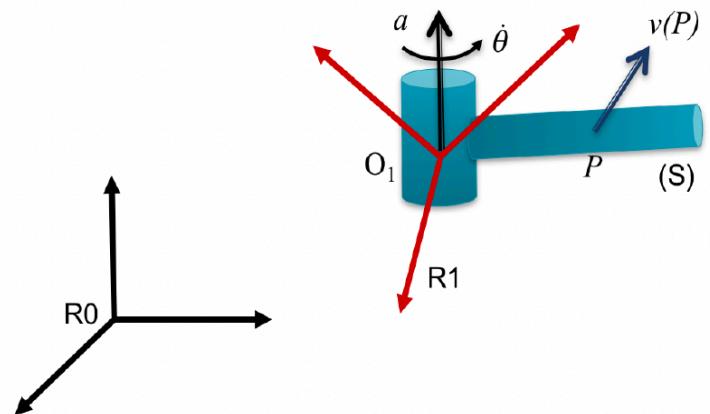
- Le solide subit une rotation autour de l'axe :

- ${}^1a = (0 \ 0 \ 1)^T$  défini dans le repère R1 à la
- Vitesse  $\dot{\theta} = 0.5 \text{ rd/s}$



# Cinématique

- Exercice 7 :
- Calculer la vitesse angulaire du solide dans le repère R0.
- Calculer dans R0 la vitesse linéaire en  ${}^1P = (0.5 \ 0.5 \ 0)^T$  défini en R1
- Sachant que la vitesse de l'origine O1
- $v(O_1) = (1 \ 2 \ 0.5)^T$  m/s dans R0



# Cinématique directe

- Modèle cinématique direct :

- Avec  $J_X(q)$  est la matrice **jacobienne analytique**       $\dot{X} = \frac{\partial f}{\partial q}(q) \dot{q} = J_X(q) \dot{q}$
- $J_X(q)$  est de dimension  $m \times n$ ,  $X = \begin{pmatrix} P \\ \Phi \end{pmatrix} = f(q)$ , qui est le MGD.

- Matrice Jacobienne géométrique :

- Pour une articulation de type R on a :  $\begin{pmatrix} v_j(P) \\ \omega_j \end{pmatrix} = \begin{pmatrix} a_j \times \vec{O_j P} \\ a_j \end{pmatrix} \dot{q}_j = J_j \dot{q}_j$
- Pour une articulation de type P on a :  $\begin{pmatrix} v_j(P) \\ \omega_j \end{pmatrix} = \begin{pmatrix} a_j \\ 0 \end{pmatrix} \dot{q}_j = J_j \dot{q}_j$
- Avec le point P est le point du TCP (la plupart du temps).

# Cinématique directe

- Modèle cinématique direct :

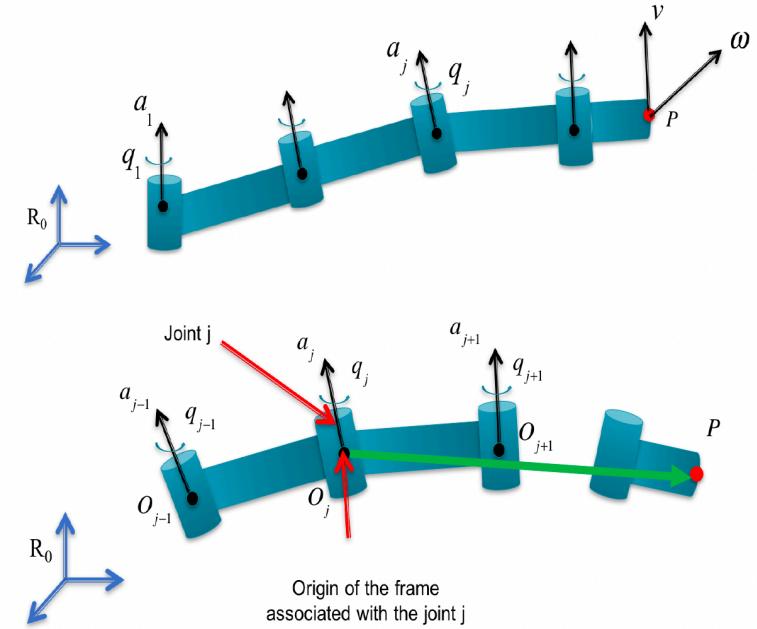
$$\begin{pmatrix} v_j(P) \\ \omega_j \end{pmatrix} = \begin{pmatrix} a_j \times \vec{O_j P} \\ a_j \end{pmatrix} \dot{q}_j = J_j \dot{q}_j$$

- Avec  $J = (J_1 \ J_2 \ J_3 \ \dots \ J_n)_{6 \times n}$
- Pour le calcul de la matrice Jacobienne, on a :

$$\cdot {}^0T_j = \begin{pmatrix} {}^0R_i & {}^0P_j \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

- $a_j = z_j = {}^0 R_j(1 : 3,3)$
- $\overrightarrow{O_j P} = \overrightarrow{O_0 P} - \overrightarrow{O_0 O_j} = {}^0 T_p(1 : 3,4) - {}^0 T_j(1 : 3,4)$

- Ecrire l'algorithme calculant la matrice Jacobienne géométrique définie par les paramètres DH modifiés.



# Solutions Robot Kuka LWR

```
q0=[0.143019, -0.109465, -0.011994, -1.1788, -0.154233, 0.93555, 0.264868];
```

$${}^0T_7 = \begin{pmatrix} -0.408983 & 0.05293 & -0.911006 & -0.366889 \\ 0.119445 & 0.992833 & 0.00406101 & -0.0379161 \\ 0.904691 & -0.107154 & -0.412374 & 0.863414 \\ 0.0 & 0.0 & 0.0 & 1.0 \end{pmatrix}$$

```
P=[-0.3668886207137053, -0.03791609269591508, 0.8634137719059174]
```

```
J=[0.03791609269591508 -0.547268634867933 0.04629863851078837 0.15391724468871776 0.0035146716514660122 0.031869001420461625 0.0;  
-0.3668886207137053 -0.07880787050173274 -0.4244798706382025 0.020839474015372682 -0.06212358325627413 0.011280815633467329 0.0;  
0.0 -0.3685470167025494 0.001612955847352644 0.4123919922152176 -0.008376301611551125 -0.07029302915016773 -0.0;  
0.0 0.14253193627295377 0.10813113392675715 -0.13072147848444948 -0.8694426715096779 0.05598019916465056 -0.9110055435337664;  
0.0 -0.9897902035998754 0.015571117832765668 0.9914182660471231 -0.1140033300362337 -0.9894780817028502 0.004061011347561494;  
1.0 6.123233995736766e-17 0.9940146870973043 -0.001310271328961469 0.48070019939538616 -0.13341417889839827 -0.41237411150262804]
```

# Cinématique inverse

- Cinématique inverse :
  - Soit  $\dot{X} = J(q)\dot{q}$ , avec dimension de  $J(q)$  est  $m \times n$ . On peut calculer la cinématique inverse telle que :
  - $\dot{q} = J(q)^{-1}\dot{X}$
  - Les conditions d'inversion de  $J(q)$  nécessite que  $m = n$ , et que  $\det(J(q)) \neq 0$ .
  - Lorsque  $\det(J(q)) = 0$ , on parle de singularités.
  - Lorsque  $\dim(J(q)) = m \times n$  avec  $m < n$ , on parle de robot redondant ou de système sous-déterminé.

# Cinématique inverse

## Systèmes sous-déterminés

Soit le système algébrique :

$$Ax = b \quad (8)$$

Avec  $A$  une matrice de dimension  $m \times n$  et  $x$  un vecteur de dimension  $n \times 1$  et  $b$  de dimension  $m \times 1$ .

On considère le cas sous-déterminé avec :  $m < n$  et  $\text{rang}(A) = m$ . Le système présente un nombre d'équations ( $m$ ) inférieur aux nombres d'inconnus ( $n$ ).

Le nombre de solution est infinie. On propose alors le choix de minimiser la norme euclidienne du vecteur  $x$ . Dans ce cas essayons de trouver quelle est la relation qui permet de réaliser cette minimisation. Soit :

$$\begin{aligned} & \underset{x}{\operatorname{argmin}} \frac{1}{2} \|x\|_2^2 \\ & \text{s.t. : } Ax - b = 0 \end{aligned} \quad (9)$$

# Cinématique inverse

## Systèmes sous-déterminés

On propose de déterminer sa solution en utilisant les multiplicateurs de Lagrange. Soit une nouvelle fonction objectif définie par :

$$\xi(x) = \frac{1}{2} \|x\|_2^2 + \lambda^T(Ax - b)$$

Avec un vecteur de dimension  $m$ . On a maintenant un problème de minimisation non-constraint avec  $m + n$  variables, les  $m$  composantes de  $b$  et les  $n$  composantes de  $x$ . On peut les regrouper dans le vecteur :

$$y = [x^T \lambda^T]^T$$

On peut maintenant établir la solution de ce problème par :

$$\frac{\partial \xi(x)}{\partial y} = 0$$

# Cinématique inverse

## Systèmes sous-déterminés

On peut décomposer cette expression en deux solutions :

$$\frac{\partial \xi}{\partial x} = x + A^T \lambda = 0_n$$

et

$$\frac{\partial \xi}{\partial x} = Ax - b = 0_m$$

. On a :  $x = -A^T \lambda$  et  $AA^T \lambda - b = 0$  De ces deux expressions on en déduit :

$$x = A^T (AA^T)^{-1} b \quad (10)$$

Avec :

$$A^+ = A^T (AA^T)^{-1} \quad (11)$$

$A^+$  est appelée la pseudo-inverse de Moore-Penrose de la matrice  $A$  ou inverse généralisée.

# Cinématique inverse

## propriétés de la pseudo-inverse

En considérant la relation :

$$Ax = b$$

avec  $x = A^+b$  et  $A^+ = A^T(AA^T)^{-1}$ , on obtient les propriétés suivantes :

$$AA^+A = A$$

$$A^+AA^+ = A^+$$

$$(A^+A)^T = A^+A$$

Une solution particulière  $x_p$  peut-être trouver en utilisant le projecteur dans le noyau de la matrice  $A$ . Soit :

$$x_p = A^+b + (I - A^+A)z \quad (12)$$

Avec  $z$  un vecteur quelconque.

# Cinématique inverse

propriétés de la pseudo-inverse

Démonstration :

$$x_p = A^+b + (I - A^+A)z$$

$$Ax_p = AA^T(AA^T)^{-1}b + A(I - A^T(AA^T)^{-1}A)z$$

$$Ax_p = AA^T(AA^T)^{-1}b + Az - AA^T(AA^T)^{-1}Az$$

$$Ax_p = b + Az - Az$$

soit :

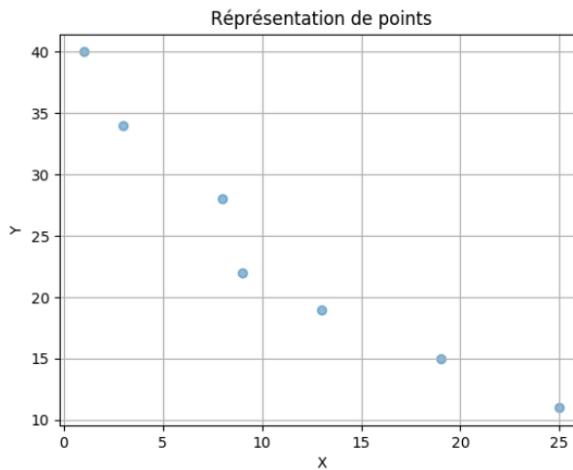
$$Ax_p = b$$

# Cinématique inverse

## Exercice : système sur-déterminé

Soit l'ensemble des points suivants :

- ①  $X = [1 \ 3 \ 8 \ 9 \ 13 \ 19 \ 25]$  et  $Y = [40 \ 34 \ 28 \ 22 \ 19 \ 15 \ 11]$
- ② Déterminer la droite d'équation  $Y = a * X + b$  minimisant l'écart entre la distribution des points et la droite au sens des moindres carrés.

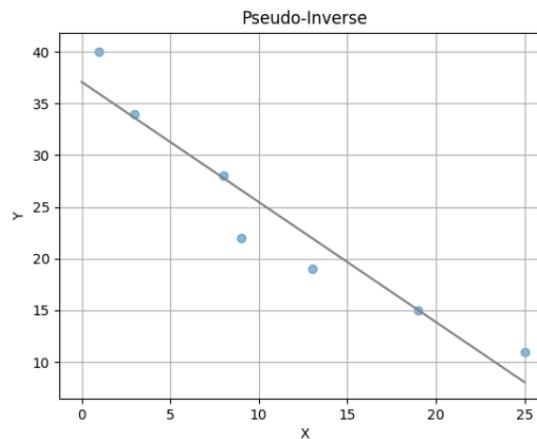


# Cinématique inverse

Exercice : système sur-déterminé

Soit l'ensemble des points suivants :

$$\begin{pmatrix} a \\ b \end{pmatrix} = \begin{pmatrix} X_1 & 1 \\ X_2 & 1 \\ \dots & \\ X_7 & 1 \end{pmatrix}^+ Y^T = \begin{pmatrix} -1.1617 \\ 37.0875 \end{pmatrix} \quad (13)$$



# Cinématique inverse

Exercice : système sous-déterminé

Soit le système suivant à 2 équations et 4 inconnues :

$$\begin{pmatrix} 0.5 & -1 & 3 & 0.1 \\ 2 & 1 & 1.5 & 2 \end{pmatrix} \begin{pmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \end{pmatrix} = \begin{pmatrix} 12 \\ 25 \end{pmatrix}$$

on obtient :

$$\begin{pmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \end{pmatrix} = \begin{pmatrix} 0.5 & -1 & 3 & 0.1 \\ 2 & 1 & 1.5 & 2 \end{pmatrix}^+ \begin{pmatrix} 12 \\ 25 \end{pmatrix} = \begin{pmatrix} 4.38 \\ 1.98 \\ 3.77 \\ 4.30 \end{pmatrix}$$

# Modèle géométrique inverse

- Modèle géométrique inverse
  - Avec le MGD on a :  $X = f(q)$
  - Le Modèle Géométrique Inverse (MGI) :  $q = g(X)$
  - Le MGI est intéressant lorsque l'on souhaite résoudre le problème de la commande d'un robot.
  - Il existe plusieurs méthodes :
    - 1)  $Find \ q^* \in argmin \ ||X - f(q)||$
    - 2) Identification des relations géométriques

# Modèle géométrique inverse

- Méthode de Paul [Paul81] :
- On considère le robot décrit par la transformation suivante :

$${}^0T_n = {}^0T_1(q_1) {}^1T_2(q_2) \cdots {}^{n-1}T_n(q_n)$$

- La méthode de Paul consiste à séparer les variables q1, q2 jusqu'à qn en calculant de façon itératives les matrices suivantes :

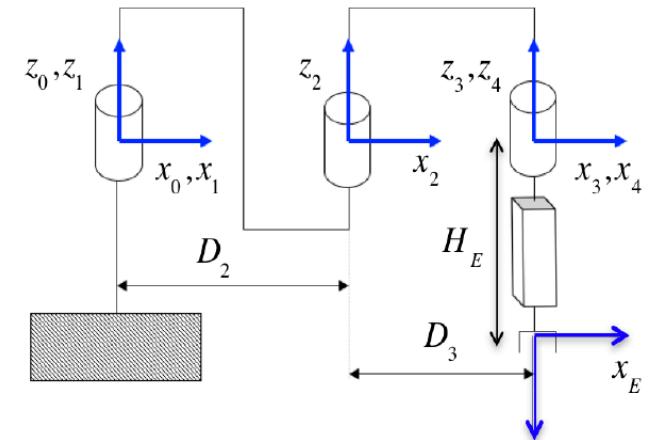
$$U_j = {}^jT_{j-1} U_{j-1}, \quad j = 1, \dots, n-1$$

- Exemples pour un robot à 6ddl :

$$\begin{aligned} U_1 &= {}^1T_0(q_1) U_0 = {}^1T_2(q_2) \cdots {}^5T_6(q_6) \\ U_2 &= {}^2T_1(q_2) U_1 = {}^2T_3(q_3) \cdots {}^5T_6(q_6) \\ U_3 &= {}^3T_2(q_3) U_2 = {}^3T_4(q_4) \cdots {}^5T_6(q_6) \\ U_4 &= {}^4T_3(q_4) U_3 = {}^4T_5(q_5) {}^5T_6(q_6) \\ U_5 &= {}^5T_4(q_5) U_4 = {}^5T_6(q_6) \end{aligned}$$

# Exercice

- Soit le robot Scara suivant :
- Donner les paramètres DH modifiés.



- Déterminer l'expression du MGD du robot :  ${}^0T_4$
- En déduire l'expression de la matrice Jacobienne géométrique.
- Calculer l'expression du MGI du robot SCARA.

# **Commande en espace libre et constraint**

# Introduction

- La génération de mouvement se décompose en 2 étapes :
- 1) La planification du chemin,
- 2) La génération e trajectoires.
- 2 types de schéma de commande :
  - A) Dans l'espace articulaire,
  - B) Dans l'espace de la tâche.

# Génération de mouvement

- **Espace articulaire :**
- Le mouvement entre la variable  $q^i$  (position initiale) et  $q^f$  (position finale) en fonction du temps  $t$  est décrit par l'équation suivante :
  - $q(t) = \sum_{i=1}^n a_i t^i = q^i + r(t)(q^f - q^i)$
  - Avec  $r(t_i) = 0$  et  $r(t_f) = 1$
  - **Interpolation linéaire** :  $q(t) = q^i + \frac{t}{t_f}(q^f - q^i)$  : discontinuité en vitesse.

# Génération de mouvement

- **Espace articulaire :**
- **Polynôme de degré 3 :**
- $q(t) = a_0 + a_1t + a_2t^2 + a_3t^3$
- Exercice : si l'on considère que  $q(0) = q^i$ ,  $q(t_f) = q^f$ ,  $\dot{q}(0) = 0$  et  $\ddot{q}(t_f) = 0$ , déterminer les coefficients  $a_i$ .
- Tracer  $q(t)$ ,  $\dot{q}(t)$  et  $\ddot{q}(t)$ .

# Génération de mouvement

- **Espace articulaire :**
- **Polynôme de degré 5 :**
- $q(t) = a_0 + a_1t + a_2t^2 + a_3t^3 + a_4t^4 + a_5t^5$
- Exercice : si l'on considère que  $q(0) = q^i$ ,  $q(t_f) = q^f$ ,  $\dot{q}(0) = 0$  et  $\dot{q}(t_f) = 0$  ainsi que  $\ddot{q}(0) = 0$  et  $\ddot{q}(t_f) = 0$ , déterminer les coefficients  $a_i$ .
- Tracer  $q(t)$ ,  $\dot{q}(t)$  et  $\ddot{q}(t)$

# Génération de mouvement

- **Espace articulaire :**
- **Calcul de la trajectoire à temps minimum (loi bang-bang) :**
- Il s'agit d'une phase d'accélération constante puis d'un phase de décélération constante.
- Exercice : si l'on considère que  $q(0) = q^i$ ,  $q(t_f) = q^f$ ,  $\dot{q}(0) = 0$  et  $\dot{q}(t_f) = 0$ , déterminer la loi horaire.
- Tracer  $q(t)$ ,  $\dot{q}(t)$  et  $\ddot{q}(t)$

# Génération de mouvement

- **Espace articulaire :**
- **Calcul de la trajectoire à temps minimum avec palier de vitesse (loi trapèze) :**
- Il s'agit d'une phase d'accélération constante puis d'un phase de décélération constante.
- Exercice : si l'on considère que  $q(0) = q^i$ ,  $q(t_f) = q^f$ ,  $\dot{q}(0) = 0$  et  $\ddot{q}(t_f) = 0$ , déterminer la loi horaire.
- Tracer  $q(t)$ ,  $\dot{q}(t)$  et  $\ddot{q}(t)$

# Génération de mouvement dans l'espace de la tâche

- Dans l'espace de la tâche on a :

- ${}^0T_E^i = \begin{pmatrix} A^i & P^i \\ 0 & 0 & 0 & 1 \end{pmatrix}$  et  ${}^0T_E^f = \begin{pmatrix} A^f & P^f \\ 0 & 0 & 0 & 1 \end{pmatrix}$

- ${}^0T_E^i$  la position et l'orientation initiale
- ${}^0T_E^f$  la position et l'orientation finale.
- Avec  $P(t) = P^i + r(t)(P^f - P^i)$
- Et  $A(t) = Rot(z, \phi^i + r(t)\phi)Rot(y, \theta^i + r(t)\theta)Rot(x, \psi^i + r(t)\psi)$
- Avec  $\phi = \phi^f - \phi^i$ ,  $\theta = \theta^f - \theta^i$  et  $\psi = \psi^f - \psi^i$

# Génération de mouvement dans l'espace de la tâche

- Dans l'espace de la tâche on a :

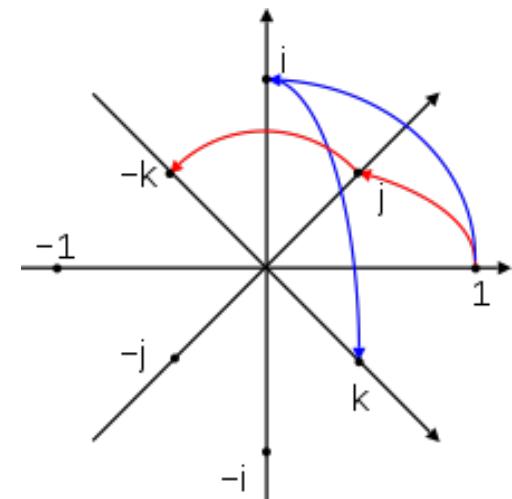
$$\bullet \quad {}^0T_E^i = \begin{pmatrix} A^i & P^i \\ 0 & 0 & 0 & 1 \end{pmatrix} \text{ et } {}^0T_E^f = \begin{pmatrix} A^f & P^f \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

- ${}^0T_E^i$  la position et l'orientation initiale
- ${}^0T_E^f$  la position et l'orientation finale.
- Avec  $P(t) = P^i + r(t)(P^f - P^i)$
- Et  $A^i Rot(u, \alpha) = A^f$ ,  $\mathbf{A}(t) = A^i Rot(u, r(t), \alpha)$

# Génération de mouvement dans l'espace de la tâche

- **Quaternions**

- $q = q_0 + iq_1 + jq_2 + kq_3$
- $i^2 = j^2 = k^2 = -1$
- $q^* = q_0 - iq_1 - jq_2 - kq_3$
- $q^{-1} = \frac{q^*}{||q||^2}$
- $||q||^2 = q_0^2 + q_1^2 + q_2^2 + q_3^2$



$$\begin{aligned} ij &= k \\ ji &= -k \\ ij &= -ji \end{aligned}$$

# Génération de mouvement dans l'espace de la tâche

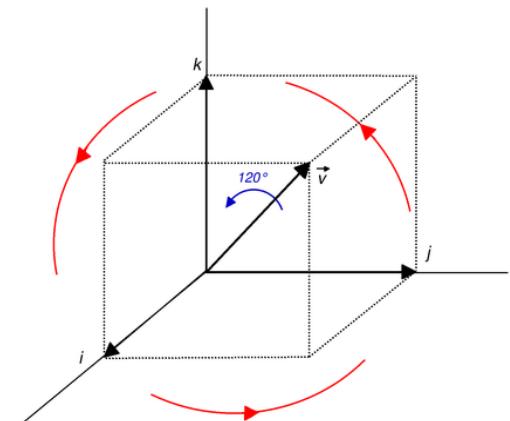
- **Quaternions**

- $q = q_0 + iq_1 + jq_2 + kq_3 = q_0 + \begin{pmatrix} q_1 \\ q_2 \\ q_3 \end{pmatrix} = \cos\left(\frac{\alpha}{2}\right) + \vec{u} \sin\left(\frac{\alpha}{2}\right)$

- **Calculer  $q \vec{v} q^{-1}$**

- **Calculer le quaternion unitaire  $q$  tel que  $\alpha = \frac{2\pi}{3}$**

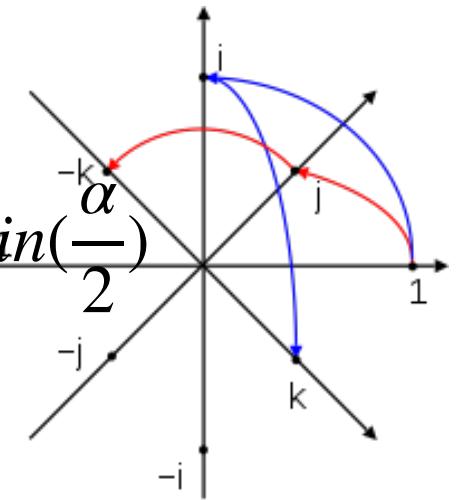
- **Réaliser la rotation du vecteur  $\vec{v} = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}$  d'un angle  $\alpha = 2\frac{\pi}{3}$ .**



# Génération de mouvement dans l'espace de la tâche

- **Quaternions**

- $q = q_0 + iq_1 + jq_2 + kq_3 = q_0 + \begin{pmatrix} q_1 \\ q_2 \\ q_3 \end{pmatrix} = \cos\left(\frac{\alpha}{2}\right) + \overrightarrow{u} \sin\left(\frac{\alpha}{2}\right)$



- **Calculer  $\overrightarrow{v} q^{-1}$**

- $q^* = q_0 - iq_1 - jq_2 - kq_3$

- **Calculer le quaternion unitaire q tel que  $\alpha = \frac{2\pi}{3}$**

$$\begin{aligned} ij &= k \\ ji &= -k \\ ij &= -ji \end{aligned}$$

# Génération de mouvement dans l'espace de la tâche

- Représentation Euler-Rodrigues
- Les paramètres peuvent être considérés comme les coefficients d'un quaternion :
  - $q = a + ib + jc + kd$
  - Avec :
    - $a = \cos\left(\frac{\theta}{2}\right); b = u_x \sin\left(\frac{\theta}{2}\right); c = u_y \sin\left(\frac{\theta}{2}\right); d = u_z \sin\left(\frac{\theta}{2}\right)$
    - On obtient la matrice de rotation  $\text{Rot}(u, \theta)$  tel que :

# Génération de mouvement dans l'espace de la tâche

- Dans l'espace de la tâche on a :

$$\bullet \quad {}^0T_E^i = \begin{pmatrix} A^i & P^i \\ 0 & 0 & 0 & 1 \end{pmatrix} \text{ et } {}^0T_E^f = \begin{pmatrix} A^f & P^f \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

- ${}^0T_E^i$  la position et l'orientation initiale
- ${}^0T_E^f$  la position et l'orientation finale.
- Avec  $P(t) = P^i + r(t)(P^f - P^i)$
- Et  $A^i Rot(u, \alpha) = A^f$ ,  $\mathbf{A(t)} = A^i Rot(u, r(t), \alpha)$

# Génération de mouvement dans l'espace de la tâche

- Représentation Euler-Rodrigues

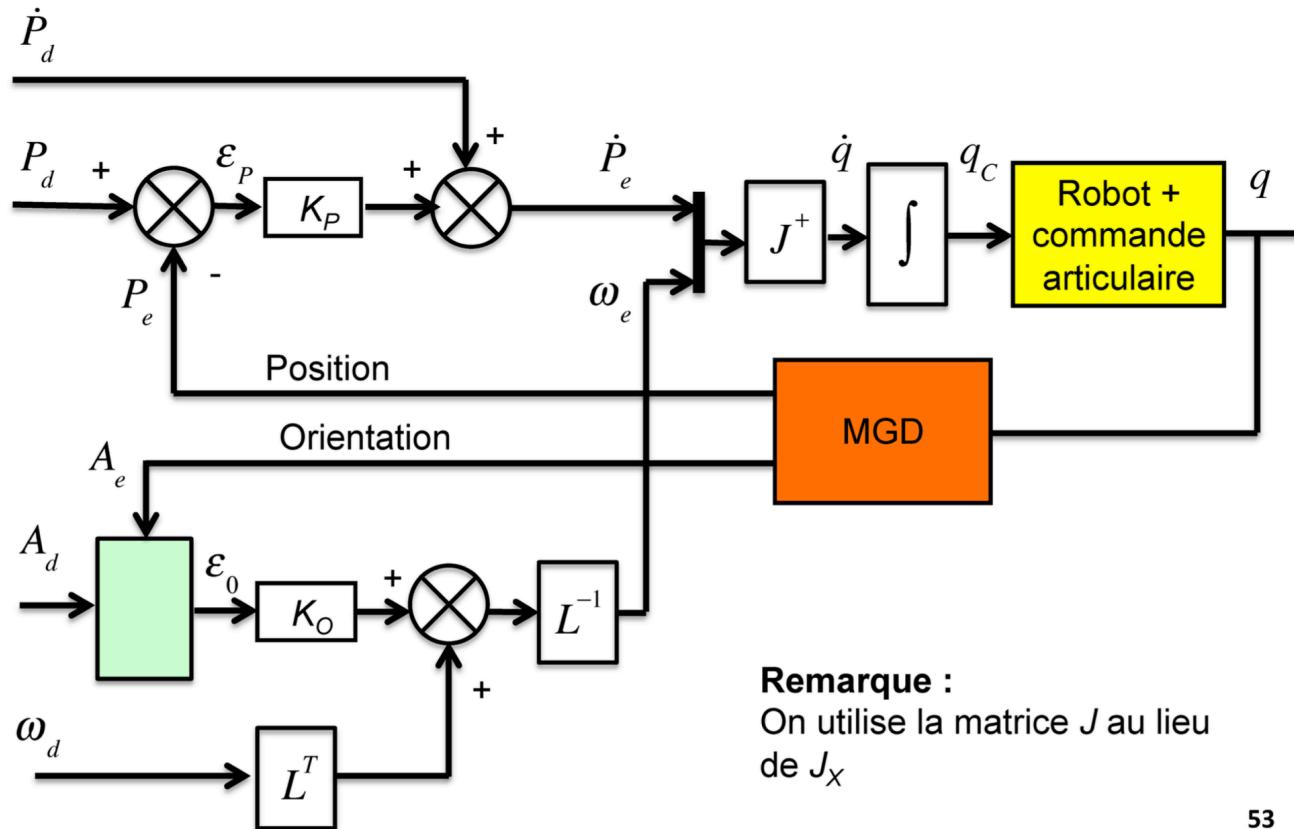
$$R(u, \theta) = \begin{pmatrix} a^2 + b^2 - c^2 - d^2 & 2(bc - ad) & 2(bd + ac) \\ 2(bc + ad) & a^2 + c^2 - b^2 - d^2 & 2(cd - ab) \\ 2(bd - ac) & 2(cd + ab) & a^2 + d^2 - b^2 - c^2 \end{pmatrix}$$

$$R(u, \theta) = \begin{pmatrix} u_x^2(1 - \cos(\theta)) + \cos(\theta) & u_x u_y (1 - \cos(\theta)) - u_z \sin(\theta) & u_x u_z (1 - \cos(\theta)) + u_y \sin(\theta) \\ u_x u_y (1 - \cos(\theta)) + u_z \sin(\theta) & u_y^2(1 - \cos(\theta)) + \cos(\theta) & u_y u_z (1 - \cos(\theta)) - u_x \sin(\theta) \\ u_x u_z (1 - \cos(\theta)) - u_y \sin(\theta) & u_y u_z (1 - \cos(\theta)) + u_x \sin(\theta) & u_z^2(1 - \cos(\theta)) + \cos(\theta) \end{pmatrix}$$

# Génération de mouvement dans l'espace de la tâche

- Représenter le schéma de commande dans l'espace opérationnel basé sur le MGI.
- Représenter le schéma de commande dans l'espace opérationnel basé sur la cinématique inverse.

# Commande cinématique inverse



# Commande cinématique inverse

Rappel soit la matrice de rotation :  $T = \begin{pmatrix} s_x & n_x & a_x & 0 \\ s_y & n_y & a_y & 0 \\ s_z & n_z & a_z & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$

Soit l'erreur de position du vecteur X :

$$e_p = [dX_p^T dX_r^T]^T$$

Avec l'erreur de position :

$$dX_p = {}^0 P_n^d - {}^0 P_n$$

L'erreur d'orientation s'exprime :

$$dX_r = \frac{1}{2} \begin{pmatrix} n_z - a_y \\ a_x - s_z \\ s_y - n_x \end{pmatrix} \text{ tel que ;}$$

# Commande cinématique inverse

L'erreur d'orientation s'exprime :

$$\varepsilon_o = \vec{r} \sin \theta = \frac{1}{2} \begin{pmatrix} a_{32} - a_{23} \\ a_{13} - a_{31} \\ a_{21} - a_{12} \end{pmatrix} = \frac{1}{2} (s_e \times s_d + n_e \times n_d + a_e \times a_d) \text{ avec } \|\vec{r}\|=1$$

La dérivée de l'erreur d'orientation est liée aux vitesses angulaires  $\omega_e$  et  $\omega_d$  par la relation (non démontré) :

$$\dot{\varepsilon}_o = L^T \omega_d - L \omega_e \quad \text{avec} \quad L = -\frac{1}{2} (\hat{s}_e \hat{s}_d + \hat{n}_e \hat{n}_d + \hat{a}_e \hat{a}_d)$$

Opérateur qui associe au vecteur  $v$  la matrice  $\hat{v}$

$$\hat{v} = \begin{pmatrix} 0 & -v_z & v_y \\ v_z & 0 & -v_x \\ -v_y & v_x & 0 \end{pmatrix}$$

# Commande cinématique inverse

L'erreur d'orientation s'exprime :

$$\epsilon_0 = \vec{r} \sin \theta = \frac{1}{2} \begin{pmatrix} n_z - a_y \\ a_x - s_z \\ s_y - n_x \end{pmatrix} = \frac{1}{2} (s_e \times s_d + n_e \times n_d + a_e \times a_d) \quad \text{Avec } \| r \| = 1$$