



Robotique

Manipulation et commande

Université de Strasbourg

Telecom Physique Strasbourg, option ISAV

Master IRIV, parcours AR

Chapitre 3 – Modélisation cinématique

Plan du chapitre

- 1. Introduction
- 2. Jacobien d'un robot
- 3. Jacobien inverse
- 4. Applications

I. Introduction

- Un robot est actionné au niveau articulaire. On contrôle la vitesse à ce niveau.
- Les tâches sont réalisées par l'effecteur et doivent donc être spécifiées dans un repère qui lui est associé.
- Certaines tâches nécessitent d'imposer une vitesse à l'effecteur. Pour pouvoir réaliser ce mouvement, il faut connaître la relation entre vitesses articulaires et vitesses opérationnelles.

2. Jacobien d'un robot

2.1. Définition

- **Jacobien**

- Soit C le torseur cinématique de l'effecteur.
- Soit \dot{q} le vecteur des vitesses articulaires.
- Soit J le *Jacobien* du robot ou *Jacobien géométrique*.
- Alors on a :

$$C = J\dot{q}$$

- Remarque : en général, le Jacobien dépend des positions articulaires q , c'est pourquoi on le note parfois $J(q)$.

2. Jacobien d'un robot

2.2. Calcul du Jacobien : vitesse linéaire

- On applique la loi de composition des vitesses linéaires (cf chapitre 1) :
 - Soit R_n le repère de l'effecteur et O_n son origine.
 - Loi de composition des vitesses :

$${}^0\mathbf{v}_{0n} = {}^0\mathbf{v}_{0n}^{O_n} = {}^0\mathbf{v}_{01}^{O_n} + {}^0\mathbf{v}_{12}^{O_n} + \dots + {}^0\mathbf{v}_{n-1n}^{O_n} = \sum_{i=1}^n {}^0\mathbf{v}_{i-1i}^{O_n}$$

- Si l'axe q_i est prismatique : ${}^0\mathbf{v}_{i-1i}^{O_n} = {}^0\mathbf{z}_{i-1}\dot{q}_i$
- Si l'axe q_i est rotoïde:
$$\begin{aligned} {}^0\mathbf{v}_{i-1i}^{O_n} &= {}^0\boldsymbol{\Omega}_{i-1i} \times {}^0(O_{i-1}O_n) \\ &= \dot{q}_i {}^0\mathbf{z}_{i-1} \times (R_{0i-1}^{i-1}O_n) \end{aligned}$$

2. Jacobien d'un robot

2.2. Calcul du Jacobien : vitesse angulaire

- On applique la loi de composition des vitesses angulaires :

- On a : ${}^0\Omega_{0n} = {}^0\Omega_{01} + {}^0\Omega_{12} + \dots + {}^0\Omega_{n-1n} = \sum_{i=1}^n {}^0\Omega_{i-1i}$

- Si l'axe q_i est prismatique : ${}^0\Omega_{i-1i} = 0$

- Si l'axe q_i est rotoïde : ${}^0\Omega_{i-1i} = {}^0z_{i-1}\dot{q}_i$

2. Jacobien d'un robot

2.2. Calcul du Jacobien : synthèse

- Des 2 transparents précédents on tire :

$$J = \begin{bmatrix} J_1 & J_2 & \cdots & J_n \end{bmatrix}$$

- Avec :

$$J_i = \begin{bmatrix} {}^0\mathbf{z}_{i-1} \\ \mathbf{0}_3 \end{bmatrix} \text{ si l'axe est prismatique}$$

- Et :

$$J_i = \begin{bmatrix} {}^0\mathbf{z}_{i-1} \times \left(R_{0\ i-1}^{i-1} O_n \right) \\ {}^0\mathbf{z}_{i-1} \end{bmatrix} \text{ si l'axe est rotoïde}$$

2. Jacobien d'un robot

2.2. Calcul du Jacobien : synthèse

- **Remarques**

- Cette méthode systématique permet de trouver J avec la seule connaissance du modèle géométrique.
- Cette méthode ne nécessite aucune dérivée, elle peut donc être facilement implémentée dans un langage informatique.
- Cette méthode suppose que le modèle géométrique ait été modélisé avec la convention de DH et notamment que les axes z soient colinéaires aux axes du robot.

2. Jacobien d'un robot

2.3. Exemple I : robot plan

- **Hypothèses**

- Le robot plan n'a que 3 degrés de liberté en translation. Pour le Jacobien on se limite donc à la partie « translation », à savoir les 3 premières lignes.

- On définit : $\mathbf{q} = \begin{bmatrix} d & \alpha & \beta \end{bmatrix}^T \triangleq \begin{bmatrix} q_1 & q_2 & q_3 \end{bmatrix}^T$

- **1^{ère} méthode : calcul direct**

- En dérivant le modèle géométrique en translation, on en déduit le Jacobien.

2. Jacobien d'un robot

2.3. Exemple I : robot plan

- D'après le chapitre 2, on a :

$$T_x = a_3 \cos(q_2 + q_3) + a_2 \cos q_2 + a_1$$

$$T_y = a_3 \sin(q_2 + q_3) + a_2 \sin q_2$$

$$T_z = q_1$$

- D'où :

2. Jacobien d'un robot

2.3. Exemple 1 : robot plan

- **2^{ème} méthode : composition**

$$J = \begin{bmatrix} J_1 & J_2 & J_3 \end{bmatrix}$$

avec $J_i = \begin{bmatrix} {}^0\mathbf{z}_{i-1} \end{bmatrix}$ si l'axe i est prismatique

et $J_i = \begin{bmatrix} {}^0\mathbf{z}_{i-1} \times \left(R_{0\ i-1}^{i-1} O_n \right) \end{bmatrix}$ si l'axe i est rotoïde

- L'axe 1 est prismatique :

- L'axe 2 est rotoïde :

2. Jacobien d'un robot

2.3. Exemple I : robot plan

2. Jacobien d'un robot

2.3. Exemple I : robot plan

2. Jacobien d'un robot

2.3. Exemple I : robot plan

- L'axe 3 est rotoïde : $J_3 = \left[{}^0z_2 \times (R_{02} {}^2O_3) \right]$

- On a :

$$R_{02} = R_{01}R_{12} = R_{12} = \begin{bmatrix} c_2 & -s_2 & 0 \\ s_2 & c_2 & 0 \\ 0 & 0 & 1 \end{bmatrix} \Rightarrow {}^0z_2 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

- De plus :

$$M_{23} = \begin{bmatrix} c_3 & -s_3 & 0 & a_3c_3 \\ s_3 & c_3 & 0 & a_3s_3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \Rightarrow {}^2O_3 = \begin{bmatrix} a_3c_3 \\ a_3s_3 \\ 0 \end{bmatrix}$$

2. Jacobien d'un robot

2.3. Exemple 1 : robot plan

$$\Rightarrow R_{02} {}^2O_3 = \begin{bmatrix} c_2 & -s_2 & 0 \\ s_2 & c_2 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} a_3 c_3 \\ a_3 s_3 \\ 0 \end{bmatrix} = \begin{bmatrix} a_3 c_{23} \\ a_3 s_{23} \\ 0 \end{bmatrix}$$

$$\Rightarrow \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} a_3 c_{23} \\ a_3 s_{23} \\ 0 \end{bmatrix} = \begin{bmatrix} -a_3 s_{23} \\ a_3 c_{23} \\ 0 \end{bmatrix}$$

2. Jacobien d'un robot

2.3. Exemple I : robot plan

- Et finalement :

$$J(\mathbf{q}) = \begin{bmatrix} 0 & -a_3 s_{23} - a_2 s_2 & -a_3 s_{23} \\ 0 & a_3 c_{23} + a_2 c_2 & a_3 c_{23} \\ 1 & 0 & 0 \end{bmatrix}$$

- **Conclusions**

- On vérifie qu'on obtient bien le même résultat qu'avec la méthode par dérivation.
- Pour le Jacobien en translation la méthode basée sur les compositions de vitesses est plus fastidieuse.
- Par contre, cette dernière méthode ne fait appel à aucune dérivation et peut donc être programmée.

2. Jacobien d'un robot

2.4. Exemple 2 : rotule

- **Hypothèses**

- Dans le chapitre 2, on a modélisé le poignet d'un robot anthropomorphe. On reprend ici ce modèle.
- On renomme $q_1 q_2 q_3$ les 3 variables articulaires.
- Pour le calcul du Jacobien, on ne s'intéresse cette fois qu'à la partie « rotations ».
- Rappel du modèle géométrique obtenu au ch. 2 :

$$R_{01} = \begin{bmatrix} c_1 & 0 & -s_1 \\ s_1 & 0 & c_1 \\ 0 & -1 & 0 \end{bmatrix} \quad R_{12} = \begin{bmatrix} c_2 & 0 & s_2 \\ s_2 & 0 & -c_2 \\ 0 & 1 & 0 \end{bmatrix} \quad R_{23} = \begin{bmatrix} c_3 & -s_3 & 0 \\ s_3 & c_3 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

2. Jacobien d'un robot

2.4. Exemple 2 : rotule

- 1^{ère} méthode : dérivation
- On a (cf chap. I) :

$$AS({}^0\mathbf{\Omega}_{0n}) = \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix} = \frac{dR_{0n}}{dt} R_{n0}$$

- De plus :

$$\frac{dR_{0n}}{dt} R_{n0} = \left(\frac{\partial R_{0n}}{\partial q_1} \dot{q}_1 + \frac{\partial R_{0n}}{\partial q_2} \dot{q}_2 + \dots + \frac{\partial R_{0n}}{\partial q_n} \dot{q}_n \right) R_{n0}$$

2. Jacobien d'un robot

2.4. Exemple 2 : rotule

- On utilise la *toolbox* Maple™ de Matlab™ :

```
clear all;
syms t;
q1=sym('q1(t)');
q2=sym('q2(t)');
q3=sym('q3(t)');
R01=[    cos(q1) 0    -sin(q1);
        sin(q1) 0    cos(q1);
        0        -1 0    ];
R12=[    cos(q2) 0    sin(q2);
        sin(q2) 0    -cos(q2);
        0        1 0    ];
R23=[    cos(q3) -sin(q3)    0;
        sin(q3) cos(q3)    0;
        0        0        1    ];
R03=simple(R01*R12*R23);
ASW=simple(diff(R03,t)*R03. ');
OMEGA=[    ASW(3,2);    ASW(1,3);    ASW(2,1)    ]
```

2. Jacobien d'un robot

2.4. Exemple 2 : rotule

- Le script renvoie le résultat suivant :

```
OMEGA =  
[  
    /d      \  
[sin(q2(t)) cos(q1(t)) |-- q3(t) | - sin(q1(t)) |-- q2(t) |]  
[  
    \dt      /      \dt      /]  
[  
    /d      \  
[cos(q1(t)) |-- q2(t) | + sin(q2(t)) sin(q1(t)) |-- q3(t) |]  
[  
    \dt      /      \dt      /]  
[  
    /d      \  
[  
    cos(q2(t)) |-- q3(t) | + |-- q1(t) |  
[  
    \dt      /      \dt      /]  
]
```

- On en déduit :

$${}^0\Omega_{03} = \begin{bmatrix} -s_1\dot{q}_2 + s_2c_1\dot{q}_3 & c_1\dot{q}_2 + s_2s_1\dot{q}_3 & \dot{q}_1 + c_2\dot{q}_3 \end{bmatrix}^T$$

2. Jacobien d'un robot

2.4. Exemple 2 : rotule

- D'où :

$${}^0\mathbf{\Omega}_{03} = J \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \end{bmatrix}$$

- Avec :

$$J = \begin{bmatrix} 0 & -s_1 & s_2 c_1 \\ 0 & c_1 & s_2 s_1 \\ 1 & 0 & c_2 \end{bmatrix}$$

2. Jacobien d'un robot

2.4. Exemple 2 : rotule

- **2^{ème} méthode : composition**

- On a : $J = \begin{bmatrix} J_1 & J_2 & J_3 \end{bmatrix}$

- Avec : $J_i = \begin{bmatrix} {}^0 z_{i-1} \end{bmatrix}$ (axes rotoïdes)

- Donc :

- Et :

2. Jacobien d'un robot

2.4. Exemple 2 : rotule

- Et :

- On en déduit :
$$J = \begin{bmatrix} 0 & -s_1 & c_1 s_2 \\ 0 & c_1 & s_1 s_2 \\ 1 & 0 & c_2 \end{bmatrix}$$

- **Conclusions**

- On retrouve le même résultat qu'en dérivant.
- Pour la rotation, la composition est plus efficace.

2. Jacobien d'un robot

2.5. Jacobien analytique

- Soit p , un vecteur d'attitude à 6 composantes et q le vecteur des coordonnées articulaires.

- On a : $p(q) = \left[\begin{array}{l} T(q) \} 3 \text{ coordonnées de translation} \\ \Theta(q) \} 3 \text{ coordonnées de rotation} \end{array} \right]$

- La représentation de la rotation doit être précisée.

2. Jacobien d'un robot

2.5. Jacobien analytique

- On a :
$$\dot{\mathbf{p}} = \begin{bmatrix} \dot{\mathbf{T}} \\ \dot{\Theta} \end{bmatrix} \triangleq J_a(\mathbf{q})\dot{\mathbf{q}}$$
- J_a est appelé le *Jacobien analytique*.
- La partie translation est identique à celle du *Jacobien géométrique* J .
- Seule la partie rotation diffère. Son expression dépend de la représentation. Un Jacobien relie la partie rotation de J_a à celle de J .

2. Jacobien d'un robot

2.5. Exemple 3 : Jacobien analytique

- Soit une attitude $p(q)$ décrite avec la représentation roulis, tangage, lacet.
- On rappelle l'expression générale de la matrice de rotation roulis, tangage, lacet :

$$R = \begin{bmatrix} c\theta_l c\theta_t & -s\theta_l c\theta_r + c\theta_l s\theta_t s\theta_r & s\theta_l s\theta_r + c\theta_l s\theta_t c\theta_r \\ s\theta_l c\theta_t & c\theta_l c\theta_r + s\theta_l s\theta_t s\theta_r & -c\theta_l s\theta_r + s\theta_l s\theta_t c\theta_r \\ -s\theta_t & c\theta_t s\theta_r & c\theta_t c\theta_r \end{bmatrix}$$

2. Jacobien d'un robot

2.5. Exemple 3 : Jacobien analytique

- On a : $AS(\mathbf{\Omega}) = \dot{R}R^T \Rightarrow \mathbf{\Omega} = \begin{bmatrix} -\dot{\theta}_t s\theta_l + \dot{\theta}_r c\theta_t c\theta_l \\ \dot{\theta}_t c\theta_l + \dot{\theta}_r c\theta_t s\theta_l \\ \dot{\theta}_l - \dot{\theta}_r s\theta_t \end{bmatrix}$

- D'où : $\mathbf{\Omega} = \underbrace{\begin{bmatrix} c\theta_t c\theta_l & -s\theta_l & 0 \\ c\theta_t s\theta_l & c\theta_l & 0 \\ -s\theta_t & 0 & 1 \end{bmatrix}}_{J_{rtl}} \begin{bmatrix} \dot{\theta}_r \\ \dot{\theta}_t \\ \dot{\theta}_l \end{bmatrix}$

- Donc : $J = \begin{bmatrix} \mathbf{I}_3 & \mathbf{0}_3 \\ \mathbf{0}_3 & J_{rtl} \end{bmatrix} J_a$

3. Jacobien inverse d'un robot

3.1. Définition

- Soit J le Jacobien d'un robot à n axes.
- On a : ${}^0C_{0n} = J\dot{q}$
- Le *Jacobien inverse* permet d'obtenir, si elles existent, les vitesses articulaires en fonction du torseur cinématique de l'effecteur :

$$\dot{q} = J^+ {}^0C_{0n} \text{ avec } J^+ = \left(J^T J \right)^{-1} J^T$$

3. Jacobien inverse d'un robot

3.2. Singularités

- Dans le cas particulier où le robot a autant d'axes que de degrés de liberté, J^+ est égal à J^{-1} .
- Le Jacobien inverse est indéfini si le robot est dans une configuration singulière.
- Comme pour le modèle géométrique inverse, il existe 2 types de singularités :
 - En limite d'espace de travail
 - A l'intérieur de l'espace de travail
 - Les configurations singulières pour le modèle géométrique inverse le sont aussi pour le Jacobien inverse.

3. Jacobien inverse d'un robot

3.3. Gestion des singularités

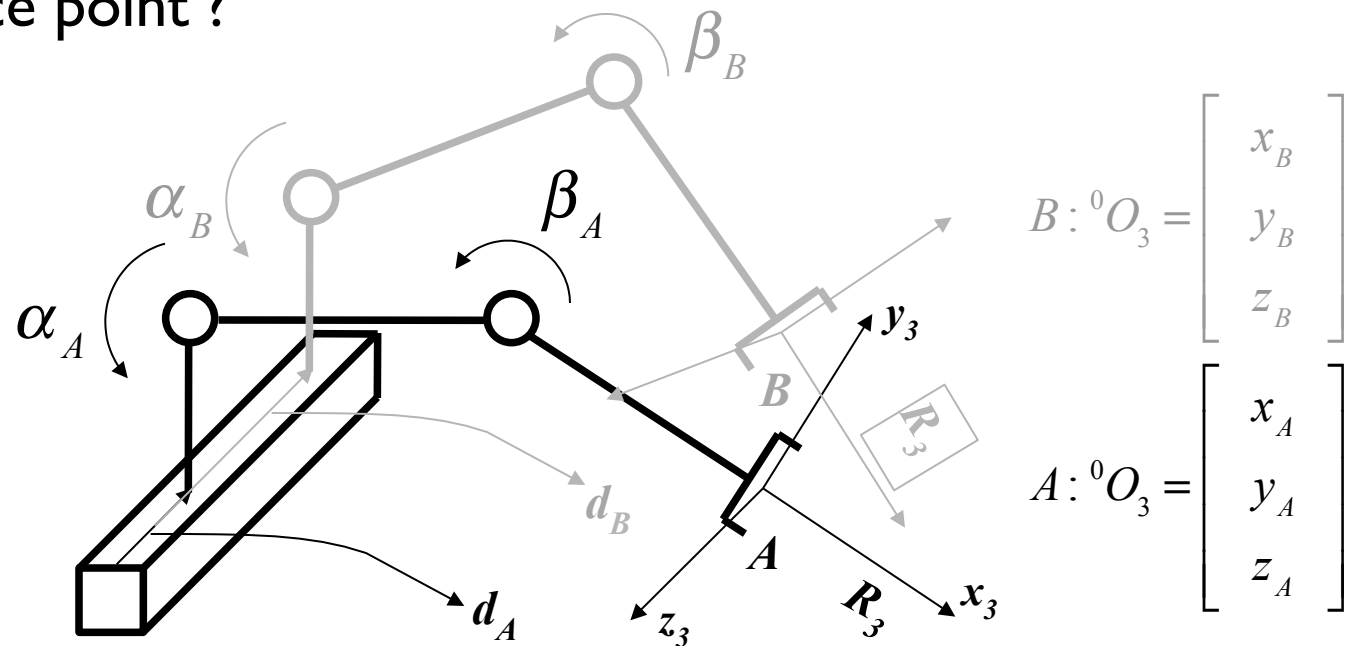
- **Détection des singularités :**
 - En testant certaines variables articulaires. Nécessite d'avoir fait l'inventaire de toutes les singularités (voir par exemple le modèle géométrique inverse d'un 6R anthropomorphe).
 - En testant le facteur de conditionnement du Jacobien.
- **A proximité de la singularité :**
 - Calcul du Jacobien pour des positions articulaires modifiées : on calcule $J^+(\mathbf{q} + \Delta\mathbf{q})$ au lieu de $J^+(\mathbf{q})$.

4. Applications

4.1. Inversion numérique du modèle géométrique

• Problème

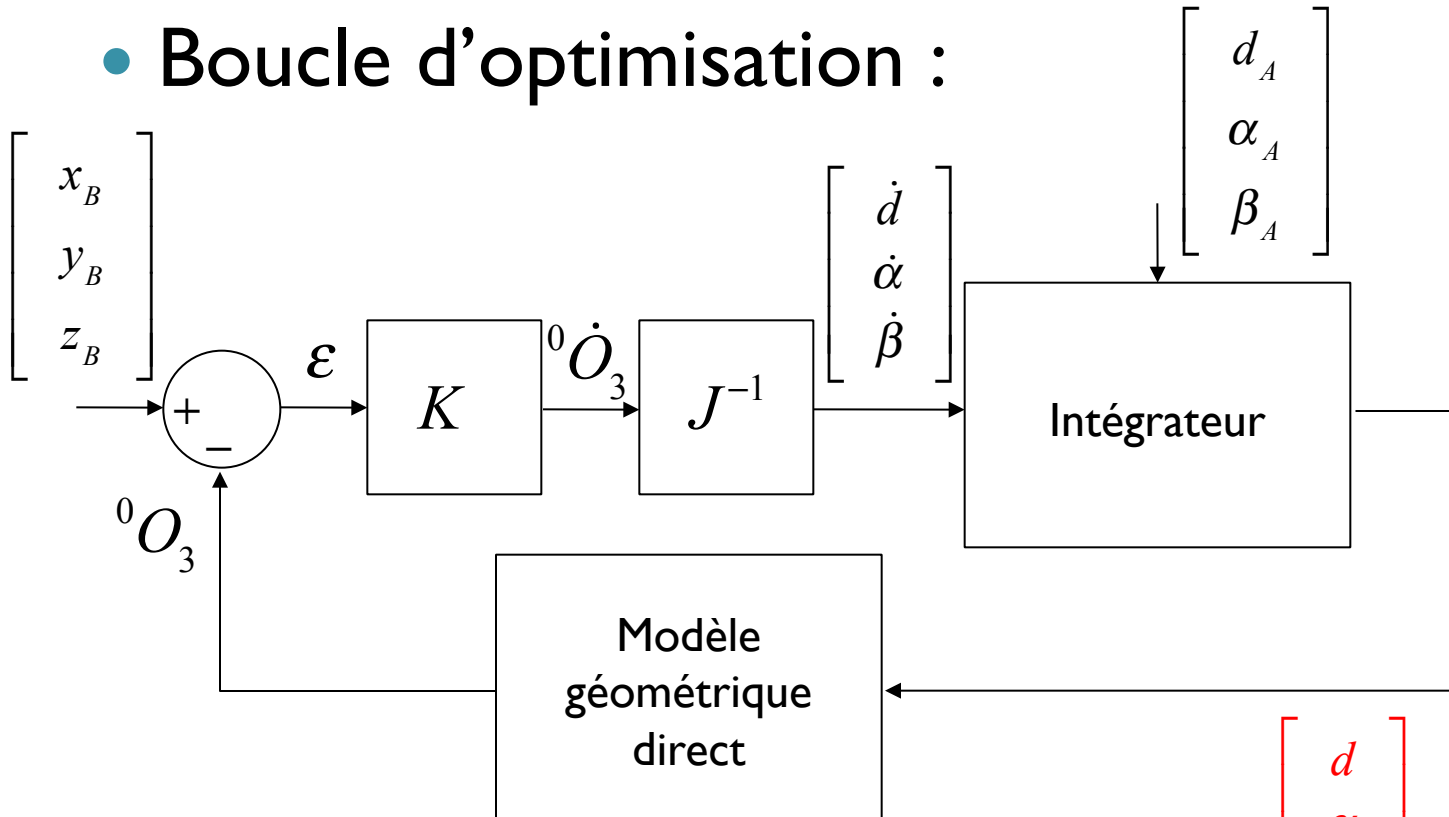
- Le robot est en position A . On veut l'amener en position B . Que valent les positions articulaires en ce point ?



4. Applications

4.1. Inversion numérique du modèle géométrique

- Boucle d'optimisation :



- Condition d'arrêt : $\varepsilon < \text{Seuil}$

$\begin{bmatrix} d \\ \alpha \\ \beta \end{bmatrix}$
Solution quand
l'algo. a convergé.

4. Applications

4.2. Efforts statiques

- **Problème**

- Trouver les efforts articulaires qui génèrent un effort donné de l'organe terminal sur l'environnement lorsque le robot est statique.

- **Principe des puissances virtuelles :**

- Lorsque le robot est statique, en contact avec l'environnement par l'effecteur, si on néglige les pertes (pertes par frottements) alors la somme des puissances virtuelles fournie par les actionneurs est égale à la puissance virtuelle transmise à l'environnement via l'effecteur.

4. Applications

4.2. Efforts statiques

- On a donc :

$$F^T {}^0C_{0n} = T^T \dot{q}$$

Avec :

F le vecteur des efforts sur l'environnement

${}^0C_{0n}$ le torseur cinématique de l'effecteur

T le vecteur des efforts articulaires

\dot{q} le vecteur des vitesses articulaires

- Or : ${}^0C_{0n} = J\dot{q} \Rightarrow F^T J\dot{q} = T^T \dot{q} \xrightarrow{\dot{q} \rightarrow 0} F^T J = T^T$

4. Applications

4.2. Efforts statiques

- D'où :

$$\mathbf{T} = \mathbf{J}^T \mathbf{F}$$

- **Remarques**

- Comme \mathbf{J} est exprimé dans R_0 , \mathbf{F} doit aussi être exprimé dans ce repère.
- Pour certains robots, les frottements secs peuvent significativement détériorer la précision de cette relation.
- Cette relation reste valable aux très basses vitesses.