Commande de robots de manipulation

Chapitre 3 : **Modélisation d’un robot manipulateur**

Modèle cénimatique

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

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

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

2. Jacobien d'un robot

2.2. Calcul du Jacobien : vitesse linéaire

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

$$^0\boldsymbol{v}_{0n} = {^0\boldsymbol{v}}_{0n}^{O_n} = {^0\boldsymbol{v}}_{01}^{O_n} + {^0\boldsymbol{v}}_{12}^{O_n} + \cdots + {^0\boldsymbol{v}}_{n-1\,n}^{O_n} = \sum_{i=1}^{n} {^0\boldsymbol{v}}_{i-1\,i}^{O_n}$$

  - ✓ Si l'axe q_i est prismatique $\quad {^0\boldsymbol{v}}_{i-1\,i}^{O_n} = {^0\boldsymbol{z}}_{i-1}\dot{q}_i$
  - ✓ Si l'axe q_i est rotoïde:

$$\begin{aligned} {^0\boldsymbol{v}}_{i-1\,i}^{O_n} &= {^0\boldsymbol{\Omega}}_{i-1\,i} \times {^0}\left(O_{i-1}O_n\right) \\ &= \dot{q}_i\, {^0\boldsymbol{z}}_{i-1} \times \left(R_{0\,i-1}\, {^{i-1}}O_n\right) \end{aligned}$$

2. Jacobien d'un robot

2.2. Calcul du Jacobien : vitesse angulaire

- On applique la loi de composition des vitesses angulaires :
  - On à: $^0\boldsymbol{\Omega}_{0n} = {^0\boldsymbol{\Omega}_{01}} + {^0\boldsymbol{\Omega}_{12}} + \cdots + {^0\boldsymbol{\Omega}_{n-1\,n}} = \sum_{i=1}^{n} {^0\boldsymbol{\Omega}_{i-1\,i}}$
  - Si l'axe q_i est prismatique : $^0\boldsymbol{\Omega}_{i-1\,i} = 0$
  - Si l'axe q_i est rotoïde : $^0\boldsymbol{\Omega}_{i-1\,i} = {^0z_{i-1}}\dot{q}_i$

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

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

2. Jacobien d'un robot
2.2. Calcul du Jacobien : synthèse

- 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 $\boldsymbol{z}$ soient colinéaires aux axes du robot.

2. Jacobien d'un robot
2.3. Exemple 1 : 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 : $\boldsymbol{q} = \begin{bmatrix} d & \alpha & \beta \end{bmatrix}^T \triangleq \begin{bmatrix} q_1 & q_2 & q_3 \end{bmatrix}^T$
- **1ère méthode : calcul direct**
  - En dérivant le modèle géométrique en translation, on en déduit le Jacobien.

2. Jacobien d’un robot
2.3. Exemple 1 : robot plan

❖ **2ième méthode : composition**

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

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

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

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 : $$\boldsymbol{p}(\boldsymbol{q}) = \begin{bmatrix} \boldsymbol{T}(\boldsymbol{q})\} \text{ 3 coordonnées de translation} \\ \Theta(\boldsymbol{q})\} \text{ 3 coordonnées de rotation} \end{bmatrix}$$
- La représe on doit être précisée.
- On a : $$\dot{\boldsymbol{p}} = \begin{bmatrix} \dot{\boldsymbol{T}} \\ \dot{\Theta} \end{bmatrix} \triangleq J_a(\boldsymbol{q})\dot{\boldsymbol{q}}$$
- *Ja* 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 *Ja* à celle de *J*.

2. Jacobien d'un robot
2.5. Exemple 3 : Jacobien analytique

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

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

  - ✓ On a:

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

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

  - ✓ Donc :

$$J = \begin{bmatrix} \boldsymbol{I}_3 & \boldsymbol{0}_3 \\ \boldsymbol{0}_3 & J_{rtl} \end{bmatrix} J_a$$

3. Jacobien inverse d'un robot
3.1. Définition

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

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

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

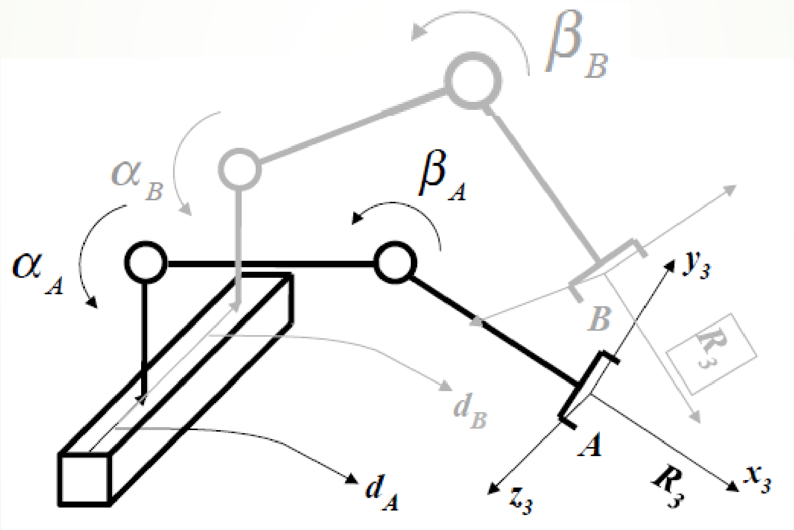
- Détection des singularités :
  - En testant certaines variables articulaires. Nécessite d’avoir fait l’inventaire de toutes les singularités (voir par exemple le modèle géométrique inverse d’un 6R anthropomorphe).
  - En testant le facteur de conditionnement du Jacobien.
- A proximité de la singularité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

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 ?

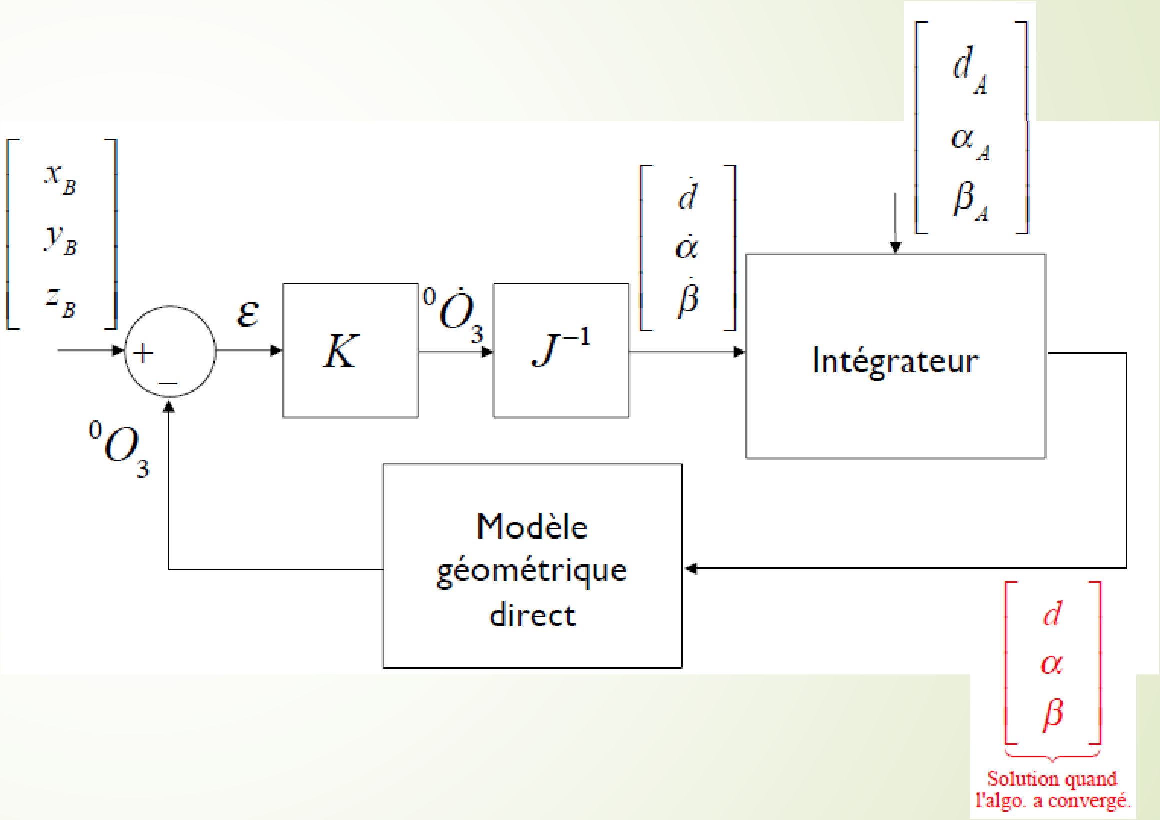


$$B : {}^{0}O_3 = \begin{bmatrix} x_B \\ y_B \\ z_B \end{bmatrix}$$

$$A : {}^{0}O_3 = \begin{bmatrix} x_A \\ y_A \\ z_A \end{bmatrix}$$

4. Applications
4.1. Inversion numérique du modèle géométrique

- Boucle d'optimisation :



- Condition d'arrêt : $\varepsilon <$ Seuil

❖ **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.

4.2. Efforts statiques

- On a donc : $F^T\, {}^0C_{0n} = T^T \dot{q}$

Avec :

- ✓ $\boldsymbol{F}$ le vecteur des efforts sur l'environnement
- ✓ ${}^0\boldsymbol{C}_{\boldsymbol{0n}}$ le torseur cinématique de l'effecteur
- ✓ $\boldsymbol{T}$ le vecteur des efforts articulaires
- ✓ $\dot{q}$ le vecteur des vitesses articulaires

- Or : ${}^0C_{0n} = J\dot{q} \Rightarrow F^T J\dot{q} = T^T \dot{q} \xrightarrow[\dot{q}\to 0]{} F^T J = T^T$
- D'où : $T = J^T F$
- **Remarques**
  - ✓ Comme J est exprimé dans R_0 , $\boldsymbol{F}$ 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.