

# Robotique Industrielle Avancée

Ce cours aborde les concepts et les méthodes suivants :

- Utilisation et intérêt de la Jacobienne
  - MGI numérique
  - Résolution du MDI (cas robots non-redondants)
  - Résolution de systèmes linéaires
  - Espace des vitesses et indice de manipulabilité
  - Modèle statique et dualité vitesse-effort
- Les robots redondants
  - Généralités et intérêt
  - Méthodes de résolution
  - Tâches prioritaires
- Modèle dynamique d'un robot manipulateur
- Génération de mouvements

# Modèle Géométrie Inverse

Pour une position/orientation  $\mathbf{X} = (X_p, X_r)^t$  données, trouver les valeurs de  $\mathbf{q}$   
 $\mathbf{X}$  peut définir l'O.T. mais aussi définir la tâche robotique  
MGI : Trouver  $\mathbf{q}$  solution de  $\mathbf{X} = f(\mathbf{q})$  pour  $\mathbf{X}$  donné.

## Problème non-linéaire

- existence ou non de solution (notion de workspace)
- unicité/multiplicité de solutions
- méthodes analytiques ou numériques

## Calcul numérique du MGI

Pourquoi : si la résolution de  $\mathbf{X} = f(\mathbf{q})$  est trop compliquée ou impossible  
Outils : optimisation non-linéaire en utilisant la Jacobienne  $J(\mathbf{q})$

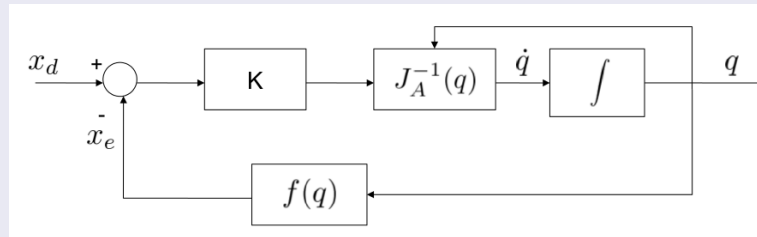


FIGURE – Schéma correspondant à la méthode de Newton

$$\mathbf{q}_{k+1} = \mathbf{q}_k + J(\mathbf{q}_k)^{-1} \cdot (\mathbf{x}_d - F(\mathbf{q}_k))$$

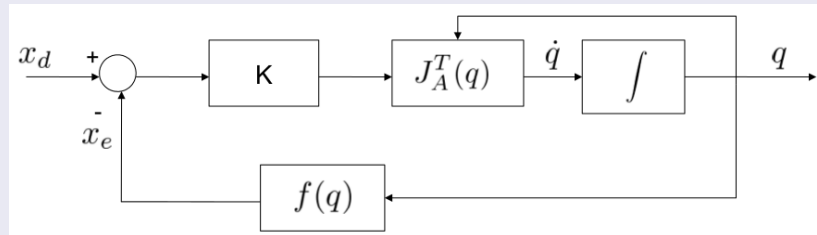


FIGURE – Schéma correspondant à la méthode du gradient

$$\mathbf{q}_{k+1} = \mathbf{q}_k + \alpha_k \cdot J(\mathbf{q}_k)^t \cdot (\mathbf{x}_d - F(\mathbf{q}_k))$$

Singularité et noyau de  $J(\mathbf{q}_k)^t$  ? Exemple du 2R plan.

## Position du problème

- Rappel: MDD =  $dq \rightarrow dx$ ?
- MDI:  $dx \rightarrow dq$ ?

On désire faire une variation de la situation de l'OT, notée  $dx$ , pour une situation connue,  $x$ , de l'OT. Comment faire varier les variables de configuration (ou de commande)  $q$  ?

## Hypothèse

- $dx$  and  $x$  sont connus
- $q$  est calculé à partir de  $x \Rightarrow$  plusieurs solutions (tableau)

## But

- Trouver  $dq$  pour un  $x$  et un  $dx$  données, tel que:  $dx = J \cdot dq$
- Il y a autant de solution au MDI qu'au MGI

# Introduction

## Méthodes de calcul

- Différentiation du MGI
  - Trop complexe
  - Rarement utilisé
- Inverser le MDD  $dx = J dq$ 
  - Il faut résoudre un système linéaire
  - $J$  n'est pas toujours inversible
    - Matrice  $J$  non carrée
    - $\text{Rang}(J)$  non plein

## Objectif :

- Exprimer le  $dq$  permettant de réaliser le  $dx$  imposé **au voisinage d'une situation  $x$  donnée**
- **But :** Trouver  $dq$  connaissant  $dx$ ,  $x$  et  $q$

## Principe de calcul

- Idée : inverser le processus de calcul du MDD
- Etapes de calcul du MDI
  1.  $dx \longrightarrow dp_{(0)} \text{ et } d\varphi_{(0)}$
  2.  $dp_{(0)} \text{ et } d\varphi_{(0)} \longrightarrow dp_{(j)} \text{ et } d\varphi_{(j)} \longrightarrow \delta$
  3. Résoudre  $J_{i(j)} dq = \delta$

## Idée: Faire l'inverse du calcul du MDD

1. Calculer  $\delta = \mathbf{J}_{p+1(p)} \cdot dq$
2. Calculer  $dp_{(p)}$  et  $d\varphi_{(p)}$  en fonction de  $\delta$
3. Calculer  $dp_{(0)}$  et  $d\varphi_{(0)}$  à partir de  $dp_{(p)}$  et  $d\varphi_{(p)}$
4. Calculer  $dx$  à partir de  $dp_{(0)}$  et  $d\varphi_{(0)}$

## MDI

1. Calculer  $dp_{(0)}$  et  $d\varphi_{(0)}$  à partir de  $dx$
2. Calculer  $dp_{(p)}$  et  $d\varphi_{(p)}$  à partir de  $dp_{(0)}$  et  $d\varphi_{(0)}$
3. Calculer  $\delta$  en fonction de  $dp_{(p)}$  et  $d\varphi_{(p)}$
4. Résoudre  $\delta = \mathbf{J}_{p+1(p)} \cdot dq$

## Etape 1 : calcul de $dp$ et $d\phi$

Calcul de  $dp, d\phi$  en fonction de  $dx = \begin{pmatrix} dx_P \\ dx_R \end{pmatrix}$

1. Calcul de  $d\phi_{(0)} = (d\phi_{x_0} \quad d\phi_{y_0} \quad d\phi_{z_0})^T$  pour  $dx_R = \begin{pmatrix} dx_n \\ dz_n \end{pmatrix}$

$\left. \begin{array}{l} dx_n = d\phi \times x_n \\ dz_n = d\phi \times z_n \end{array} \right\}$  En projetant dans base 0, on a 6 équations à 3 inconnues avec 3 contraintes.  
 On montre que :  $d\phi = (-y_n \cdot dz_n) \cdot x_n + x_n \times dx_n$

2. Calcul de  $dp_{(0)} = (dp_{x_0} \quad dp_{y_0} \quad dp_{z_0})^T$  pour  $dx_P = dp_{0,n+1(0)} = \begin{pmatrix} dx \\ dy \\ dz \end{pmatrix}$

$$dp_{(0)} = dp_{0,n+1(0)} - d\phi_{(0)} \wedge P_{n,n+1(0)}$$



## Etape 2 et 3 : calcul de $\delta$

*Etape 1 : cf. cours précédent*

$$\begin{pmatrix} dp_{(0)} \\ d\phi_{(0)} \end{pmatrix} = \begin{pmatrix} R_{01} & 0_{3 \times 3} \\ 0_{3 \times 3} & R_{01} \end{pmatrix} \begin{pmatrix} R_{12} & 0_{3 \times 3} \\ 0_{3 \times 3} & R_{12} \end{pmatrix} \dots \begin{pmatrix} R_{j-1,j} & 0_{3 \times 3} \\ 0_{3 \times 3} & R_{j-1,j} \end{pmatrix} \begin{pmatrix} I_{3 \times 3} & -\hat{P}_{in(i)} \\ 0_{3 \times 3} & I_{3 \times 3} \end{pmatrix} J_{i(j)} dq$$

CONNU

*Etape 2 et 3 : Déterminer  $\delta$*

$$\begin{pmatrix} I_{3 \times 3} & \hat{P}_{in(i)} \\ 0_{3 \times 3} & I_{3 \times 3} \end{pmatrix} \begin{pmatrix} R_{j,j-1} & 0_{3 \times 3} \\ 0_{3 \times 3} & R_{j,j-1} \end{pmatrix} \dots \begin{pmatrix} R_{21} & 0_{3 \times 3} \\ 0_{3 \times 3} & R_{21} \end{pmatrix} \begin{pmatrix} R_{10} & 0_{3 \times 3} \\ 0_{3 \times 3} & R_{10} \end{pmatrix} \begin{pmatrix} dp_{(0)} \\ d\phi_{(0)} \end{pmatrix} = J_{i(j)} dq$$

$\delta$

*Etape 4 : Résoudre le système suivant :*  $\delta = J_{i(j)} dq$

# Résolution du MDI pour des robots non-redondants

On a :  $\dot{\mathbf{X}} = J(\mathbf{q}).\dot{\mathbf{q}}$  ou  $\delta = J_{p+1(p)}(\mathbf{q}).\dot{\mathbf{q}}$  (Jacobienne ou Jacobienne préférencielle)

On peut montrer que  $\text{rang}(J(\mathbf{q})) = \text{rang}(J_{p+1(p)}(\mathbf{q}))$

Hors singularité ( $\det(J_{p+1(p)}) \neq 0$ ) on sait que  $\dot{\mathbf{q}} = J_{p+1(p)}^{-1}.\delta$  ( ou  $J^{-1}.\dot{\mathbf{X}}$  )

Lorsque  $\det(J_{p+1(p)}) = 0$  il faut vérifier si le système est compatible ou non,  
 $\text{rang}(J_{p+1(p)}) = \text{rang}([J_{p+1(p)}|\delta])$

Cette notion de compatibilité traduit le fait que le bras manipulateur ne peut effectuer des mouvements infinitésimaux de son O.T. que si ceux-ci sont compatibles avec le mouvement permis par la structure mécanique dans la configuration autour de laquelle se trouve le bras.

Exemple : 2R

# Résolution des systèmes linéaires

## Système compatible

$Y$  un vecteur  $(m, 1)$ ,  $A$  une matrice  $(m, n)$  et  $X$  un vecteur inconnu  $(n, 1)$

$Y = A.X$  admet au moins une solution  $\iff$  le système est compatible

$Y = A.X$  admet au moins une solution  $\iff \text{rang}([A \mid Y]) = \text{rang}(A)$

- Si le système est compatible
  - Si  $m > n$  et  $r = n$  alors  $X = A^+.Y$
  - Si  $m = n$  et  $r = n$  alors  $X = A^{-1}.Y$
  - Si  $m < n$  et  $r = m$  alors toutes les solutions s'écrivent  
 $X = A^+.Y + (I - (A^+.A)).z$  avec  $z \in \mathbb{R}^n$
- Si le système n'est pas compatible
  - Il n'y a pas de solution exacte
  - $X = A^+.Y$  donne la solution la plus proche au sens des moindres carrés (  $\min \|Y - A.X\|^2$  )

## Inverse généralisée

Soit  $A$  une matrice  $(m, n)$  de rang  $r \leq \min(m, n)$

- Inverse généralisée  $A^\#$

$A^\#$  de dimension  $(n, m)$  est une inverse généralisée de  $A \iff A.A^\#.A = A$

## Pseudo-Inverse

Soit  $A$  une matrice  $(m, n)$  de rang  $r \leq \min(m, n)$

- Pseudo-inverse  $A^+$

$A^+$  est la pseudo-inverse de  $A \iff$

$$A.A^+.A = A, A^+.A.A^+ = A^+, (A^+.A)^t = A^+.A, (A.A^+)^t = A.A^+$$

- Si  $m \geq n$  et  $r = n$  alors  $A^+ = (A^t.A)^{-1}.A^t$  (inverse à gauche)
- Si  $m \leq n$  et  $r = m$  alors  $A^+ = A^t.(A.A^t)^{-1}$  (inverse à droite)
- Cas général : calcul de  $A^+$  par SVD (décomposition en valeurs singulières)

# Singular Value Decomposition (SVD)

$J$  : matrice  $m \times n$ . On peut décomposer  $J$  en un produit de 3 matrices :

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

avec  $U \in \mathbb{R}^{m \times m}$   $V \in \mathbb{R}^{n \times n}$  et  $\Sigma$  matrice des valeurs singulières de  $J$ .

$U \triangleq$  matrice des vect. propres de  $J \cdot J^t$  (orthogonale)

$V \triangleq$  matrice des vect. propres de  $J^t \cdot J$  (orthogonale)

$$\Sigma = \begin{pmatrix} \sigma_1 & 0 & \dots & 0 & 0 & 0 \\ 0 & \sigma_2 & 0 & \dots & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & \dots & \sigma_r & 0 & \dots & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

avec  $\sigma_1 \geq \sigma_2 \geq \dots \sigma_r > 0$  (racines carrées des  $r$  val. propres  $> 0$  de  $J^t \cdot J$ )

## $J^+$ à partir de Singular Value Decomposition (SVD)

A partir de la décomposition en SVD de  $J = U.\Sigma.V^t$   
on a  $J^+$ , matrice ( $n \times m$ ) :

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

avec

$$\Sigma^+ = \begin{pmatrix} \frac{1}{\sigma_1} & 0 & \dots & 0 & 0 & 0 \\ 0 & \frac{1}{\sigma_2} & 0 & \dots & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & \dots & \frac{1}{\sigma_r} & 0 & \dots & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

# Exemple de calcul du MDI du RRPRRR

Etape 1 supposée réalisée

$$dx = \begin{pmatrix} dx_P \\ dx_R \end{pmatrix} \text{ pour } \underline{x} \text{ et } \underline{q} \text{ connues} \Rightarrow \begin{matrix} dp_{(0)} = \begin{pmatrix} dp_{x_0} & dp_{y_0} & dp_{z_0} \end{pmatrix}^T \\ d\varphi_{(0)} = \begin{pmatrix} d\varphi_{x_0} & d\varphi_{y_0} & d\varphi_{z_0} \end{pmatrix}^T \end{matrix} \left| \begin{matrix} \longrightarrow \text{connus} \end{matrix} \right.$$

Rappel du MDD

$$\begin{pmatrix} dp_{(0)} \\ d\varphi_{(0)} \end{pmatrix} = \begin{pmatrix} R_{01} & 0_{3 \times 3} \\ 0_{3 \times 3} & R_{01} \end{pmatrix} \begin{pmatrix} R_{12} & 0_{3 \times 3} \\ 0_{3 \times 3} & R_{12} \end{pmatrix} \begin{pmatrix} R_{23} & 0_{3 \times 3} \\ 0_{3 \times 3} & R_{23} \end{pmatrix} \begin{pmatrix} I_{3 \times 3} & -\hat{P}_{46(3)} \\ 0_{3 \times 3} & I_{3 \times 3} \end{pmatrix} J_{4(3)} d\mathbf{q}$$

## Etape 2 : Calcul de $X_3$ et $X_2$

Calcul de  $X_3$  en fonction de  $dp_{(0)}$  et  $d\varphi_{(0)}$

$$X_3 = \begin{pmatrix} R_{10} & 0_{3 \times 3} \\ 0_{3 \times 3} & R_{10} \end{pmatrix} \begin{pmatrix} dp_{(0)} \\ d\varphi_{(0)} \end{pmatrix} = \begin{pmatrix} c_1 dp_{x_0} + s_1 dp_{y_0} \\ -s_1 dp_{x_0} + c_1 dp_{y_0} \\ dp_{z_0} \\ c_1 d\varphi_{x_0} + s_1 d\varphi_{y_0} \\ -s_1 d\varphi_{x_0} + c_1 d\varphi_{y_0} \\ d\varphi_{z_0} \end{pmatrix} = \begin{pmatrix} E_1 \\ E_2 \\ dp_{z_0} \\ E_3 \\ E_4 \\ d\varphi_{z_0} \end{pmatrix}$$

## Etape 2 : Calcul de $X_2$

Calcul de  $X_2$  en fonction de  $dp_{(0)}$  et  $d\varphi_{(0)}$

$$X_2 = \begin{pmatrix} R_{21} & 0_{3 \times 3} \\ 0_{3 \times 3} & R_{21} \end{pmatrix} X_3 = \begin{pmatrix} c_2 E_1 + s_2 dp_{z_0} \\ -s_2 E_2 + c_2 dp_{z_0} \\ -E_2 \\ c_2 E_3 + s_2 d\varphi_{z_0} \\ -s_2 E_4 + c_2 d\varphi_{z_0} \\ -E_4 \end{pmatrix} = \begin{pmatrix} E_5 \\ E_6 \\ -E_2 \\ E_7 \\ E_8 \\ -E_4 \end{pmatrix}$$



## Etape 2 : Calcul de $X_1$ et $\delta$

Calcul de  $X_1$  en fonction de  $dp_{(0)}$  et  $d\varphi_{(0)}$

$$X_1 = \begin{pmatrix} R_{32} & 0_{3 \times 3} \\ 0_{3 \times 3} & R_{32} \end{pmatrix} X_2 = \begin{pmatrix} E_5 \\ -E_2 \\ -E_6 \\ E_7 \\ -E_4 \\ -E_8 \end{pmatrix}$$

$$X_1 = \begin{pmatrix} I_{3 \times 3} & -\hat{P}_{46(3)} \\ 0_{3 \times 3} & I_{3 \times 3} \end{pmatrix} J_{4(3)} dq \Rightarrow \begin{pmatrix} I_{3 \times 3} & \hat{P}_{46(3)} \\ 0_{3 \times 3} & I_{3 \times 3} \end{pmatrix} X_1 = \delta = J_{4(3)} dq$$

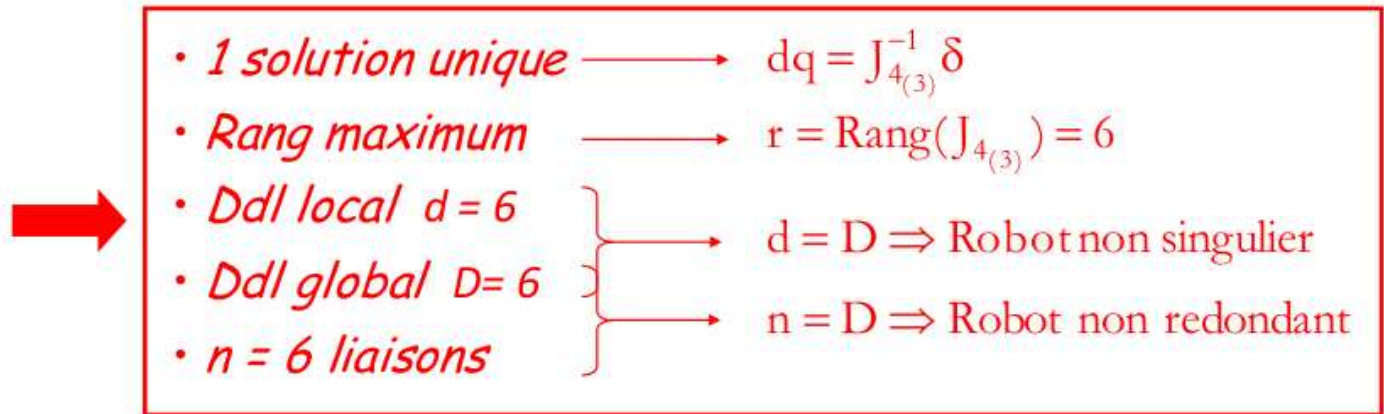
$$X_1 = \delta = J_{4(3)} dq \text{ puisque } \hat{P}_{46(3)} = 0_{3 \times 3}$$



Système à résoudre :  $X_1 = \delta = J_{4(3)} dq$   
avec  $X_1$  et  $J_{4(3)}$  connus

## Etape 3 : résolution du système

Si  $J_{4(3)}$  est inversible (i.e.,  $q_3 s_2 s_5 \neq 0$ ):




*NB : Dans ce cas :*

- *le système est toujours compatible*
- *Résolution détaillée : tableau*

## Etape 3 : résolution lorsque $J_{4(3)}$ n'est pas inversible

Si  $J_{4(3)}$  n'est pas inversible :

- 
- 1 infinité de solutions ssi le système est compatible
  - Pas de solution exacte sinon
  - Rang non maximum  $\longrightarrow r = \text{Rang}(J_{4(3)}) < 6$
  - Ddl local  $d < 6$
  - Ddl global  $D = 6$
- }  $\longrightarrow d < D \Rightarrow \text{Robot singulier}$

Dans ce cas, pour chaque configuration pour laquelle  $\det(J_{4(3)}) = 0$  :

- ① Calculer  $J_{4(3)}$  et étudier son rang
- ② Exprimer  $[J_{4(3)} \mid \delta]$  et étudier son rang
- ③ En déduire la condition de compatibilité
- ④ Résoudre le système du MDI lorsque cette condition est vérifiée

# Définition des espaces des vitesses

- Relation entre  $\dot{\mathbf{X}}$  et  $\dot{\mathbf{q}}$  
$$\dot{\mathbf{X}} = J(\mathbf{q}) \cdot \dot{\mathbf{q}} = \sum_{i=1}^r \sigma_i \cdot u_i \cdot v_i^t \cdot \dot{\mathbf{q}}$$
  
L'action des vitesses des axes ( $\dot{\mathbf{q}}$ ) se fait au travers de  $[\sigma_i \cdot u_i \cdot v_i^t]$ .

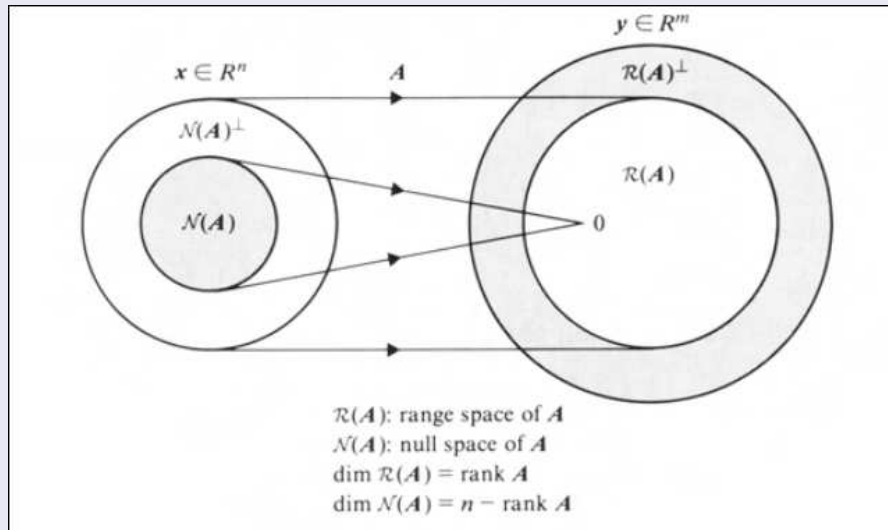


FIGURE – Nakamura-Redundancy and Optimization

- Signification de ces espaces ?
- Conséquence des singularités de  $J(q)$  ?

## Définition de l'ellipsoïde de vitesse

- Elle traduit la qualité de la transmission de vitesse

$$\dot{\mathbf{q}}_{min} \leq \dot{\mathbf{q}} \leq \dot{\mathbf{q}}_{max} \implies \dot{\mathbf{X}}_{min} \leq \dot{\mathbf{X}} \leq \dot{\mathbf{X}}_{max}$$

- Exemple polytope, hyperphère

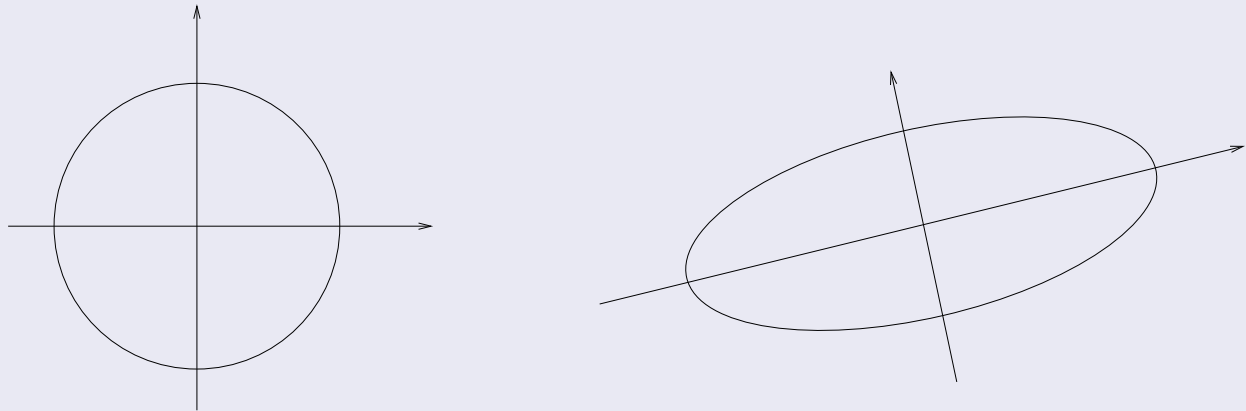


FIGURE – Ellipsoïde de vitesse

# Indice de manipulabilité

## Ellipsoïde de vitesse

Volume de l'ellipsoïde de vitesse traduit la mesure de la capacité du robot à engendrer des vitesses.

## Indice de manipulabilité en vitesse

$$\omega(\mathbf{q}) = \sqrt{\det(J(\mathbf{q}).J(\mathbf{q})^t)} = \prod_1^r \sigma_i$$

## Mesure de distance par rapport aux singularités

$$\kappa(\mathbf{q}) = \frac{\sigma_1}{\sigma_r}$$

# Efforts statiques

L'interaction du robot avec l'environnement produit des efforts (forces/couples) au niveau de l'outil  $\implies$  produit des couples au niveau des articulations du robot.

Puissance instantanée qui est transmise par l'outil (lié à l'O.T) sur l'environnement  
= puissance instantanée fournie par les actionneurs.

L'application du principe des travaux virtuels donne :

$$\tau = J(q)^t F_e$$

avec  $F$  = vecteur  $6 \times 1$  des efforts (forces et couples) agissant sur l'outil.

Relation vraie à vitesse nulle ou constante (accélération est nulle)

Remarque :  $J(q)$  et  $F$  doivent être exprimé dans  $R_o$ .

## Manipulabilité en effort

- Dualité vitesse effort

# Dualité vitesse effort

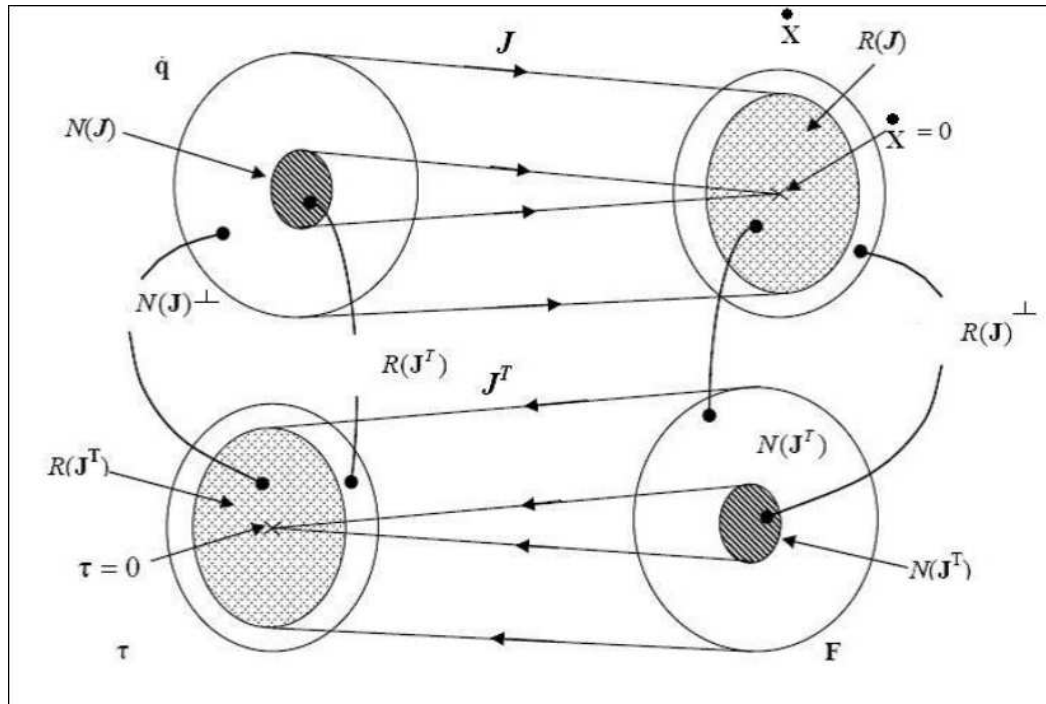


FIGURE – Nakamura-Redundancy and Optimization



# Robotique Avancée : les robots redondants

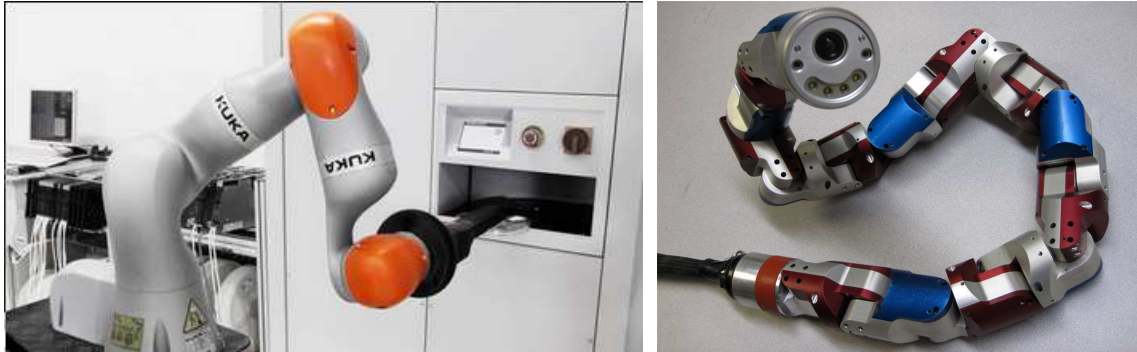


FIGURE – Exemples de robot redondant



FIGURE –

# Robots séries redondants

Robots avec  $n$  liaisons (R ou P)  $\Rightarrow$  indice de mobilité =  $n$

Nombre de degré de liberté de l'O.T (ou de la tâche) =  $m$

Un BM est redondant si  $m < n$

Si  $m = n$  (non redondant) alors à chaque  $\mathbf{x}$  correspond un nombre fini de  $\mathbf{q}$

Si  $n > m$  (redondant) alors à chaque  $\mathbf{x}$  peut correspond nombre fini ou infini de  $\mathbf{q}$ .

Le degré de redondance est  $(n - m)$ .

## Interêt de la redondance

- augmenter le volume de travail de l'OT,
- augmenter la dextérité (bras humain ?),
- éviter les configurations singulières,
- avoir des d.d.l. libres au niveau des liaisons par rapport au d.d.l. de la tâche,
- prendre en compte les butées articulaires,
- répartir les efforts des articulations

# Robots séries redondants

La relation entre les espaces des vitesses articulaires et opérationnelles est définie par  $\dot{\mathbf{x}}(t) = J(\mathbf{q}).\dot{\mathbf{q}}(t)$ .

Dans le cas de robot non redondant il suffit d'inverser  $J$

Avec un robot redondant  $J$  non carrée  $\implies$  pseudo-inverse de  $J$

Solution générale :

$$\dot{\mathbf{q}} = J^+.\dot{\mathbf{x}} + (Id - J^+.J).\dot{\mathbf{q}}_0$$

Exemple : robot 2R

Infinité de solution : comment faire un choix de la *meilleure* solution ?

Comment calculer le  $\dot{\mathbf{q}}$  pour une tâche désirée ?.

Idée : rajouter des contraintes ou prendre en compte d'autres tâches

# Robots séries redondants

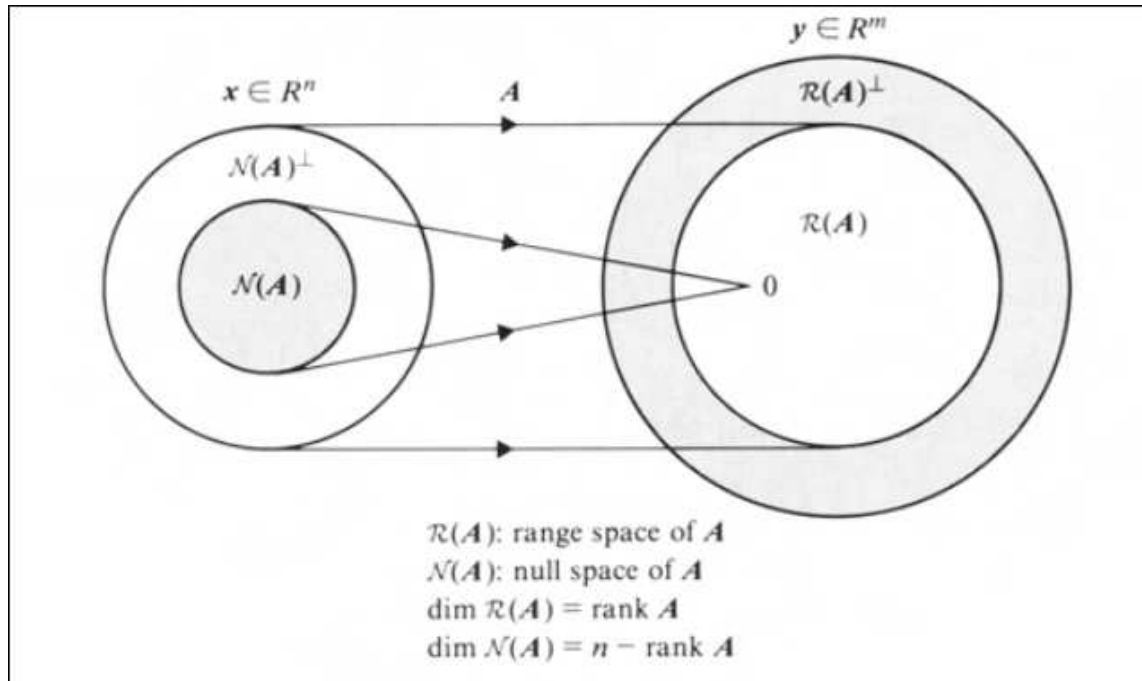


FIGURE – Nakamura-Redundancy and Optimization

Signification de ces espaces ?

# Robots séries redondants

$$R(J) = \{\dot{\mathbf{x}} \in \mathbb{R}^m | \exists \dot{\mathbf{q}} \in \mathbb{R}^n \dot{\mathbf{x}}(t) = J(\mathbf{q}).\dot{\mathbf{q}}(t)\}$$

$$R(J)^\perp = \{\dot{\mathbf{x}} \in \mathbb{R}^m | \exists \dot{\mathbf{x}}' \in R(J) \quad \dot{\mathbf{x}}'.\dot{\mathbf{x}} = 0\}$$

$$N(J) = \{\dot{\mathbf{q}} \in \mathbb{R}^n | J(\mathbf{q}).\dot{\mathbf{q}} = 0\}$$

$$N(J)^\perp = \{\dot{\mathbf{q}} \in \mathbb{R}^n | \exists \dot{\mathbf{q}}' \in N(J) \quad \dot{\mathbf{q}}^t.\dot{\mathbf{q}}' = 0\}$$

$$\dim(R(J)) + \dim(N(J)) = n$$

Signification de ces espaces ?

L'existence d'un sous-espace  $N(J) \neq 0$  pour un robot redondant permet de gérer la redondance.

# Redondance

Modèle géométrique direct de la tâche :  $\mathbf{x} = f(\mathbf{q})$

$$f : Q \longrightarrow X$$

Dimension de l'espace de la tâche  $X = m$

Dimension de l'espace articulaire  $Q = n$

Un robot est redondant pour la tâche si  $n > m$

$\mathbf{x}$  peut contenir la position et/ou l'orientation de l'O.T.

La redondance est un concept relatif à la tâche.

- Tâche en position dans le plan  $\longrightarrow m = 2$
- Tâche en position dans l'espace 3D  $\longrightarrow m = 3$
- Tâche en position et orientation dans l'espace 3D  $\longrightarrow m = 6$
- Tâche de pointage dans l'espace 3D  $\longrightarrow m = ?$

Trois classes de méthode de résolution de  $\dot{\mathbf{x}}(t) = J(\mathbf{q})\dot{\mathbf{q}}(t)$ .

- ① Méthodes basées sur la Jacobienne (en général analytique) :  
une solution parmi une infinité est choisie (minimization d'une norme,...)
- ② Méthodes basées sur l'espace Nul :  
un terme est ajouté à la solution précédente qui n'affecte pas la réalisation de la trajectoire
- ③ Méthodes d'augmentation de tâche :  
la redondance est supprimée ou réduite en ajoutant une(des) tâche(s) supplémentaire(s).

# 1 - Méthodes de résolution basées sur la Jacobienne

Tâche :  $\dot{\mathbf{x}}(t) = J(\mathbf{q}).\dot{\mathbf{q}}(t)$  implique de calculer  $\dot{\mathbf{q}}(t) = K(\mathbf{q}).\dot{\mathbf{x}}(t)$

$\forall \dot{\mathbf{x}} \in I(J) \Rightarrow J(\mathbf{q}).K(\mathbf{q}).\dot{\mathbf{x}} = J.K.J.\dot{\mathbf{q}} = J.\dot{\mathbf{q}} = \dot{\mathbf{x}}$

Il suffit d'avoir :  $J(\mathbf{q}).K(\mathbf{q}).J(\mathbf{q}) = J(\mathbf{q})$

Exemple :  $J$  avec bloc inversible

## 1 Pseudo-inverse

On prend  $K = J^+$

Si  $J$  de rang plein ( $r = m$ ) alors  $J^+ = J^t.(J.J^t)^{-1}$  sinon utiliser a SVD.

On a  $\dot{\mathbf{q}} = J^+(\mathbf{q}).\dot{\mathbf{x}}$  la seule solution qui minimise la norme de  $\dot{\mathbf{q}}$

$$\min_{\dot{\mathbf{q}}} \frac{1}{2} \|\dot{\mathbf{q}}\|^2$$

sous la contrainte

$$\dot{\mathbf{x}}(t) - J(\mathbf{q}).\dot{\mathbf{q}}(t) = 0$$

Si  $\dot{\mathbf{x}} \in I(J)$  ? Sinon ?



# 1 - Méthodes de résolution basées sur la Jacobienne

## 2 Pseudo-inverse pondérée :

On cherche à optimiser

$$\min_{\dot{\mathbf{q}}} \frac{1}{2} \dot{\mathbf{q}}^t \cdot \mathbf{W} \cdot \dot{\mathbf{q}}$$

sous la contrainte

$$\dot{\mathbf{x}}(t) - \mathbf{J}(\mathbf{q}) \cdot \dot{\mathbf{q}}(t) = 0$$

$$\dot{\mathbf{q}} = \mathbf{W}^{-1} \cdot \mathbf{J}^t \cdot [\mathbf{J} \cdot \mathbf{W}^{-1} \mathbf{J}^t]^{-1} \cdot \dot{\mathbf{x}} = \mathbf{J}_W^+ \cdot \dot{\mathbf{x}}$$

⚠ :  $\mathbf{J}_W^+$  appelé pseudo-inverse pondérée de  $\mathbf{J}$  n'est pas une pseudo-inverse.

Intérêt ?

# 1 - Méthodes de résolution basées sur la Jacobienne

## 3 Moindres carrés amortis : Damped Least Square (DLS)

On cherche à optimiser

$$\min_{\dot{\mathbf{q}}} \frac{\mu^2}{2} \|\dot{\mathbf{q}}\|^2 + \frac{1}{2} \|\dot{\mathbf{x}} - \mathbf{J}.\dot{\mathbf{q}}\|^2 = \min_{\dot{\mathbf{q}}} H(\dot{\mathbf{q}})$$

On obtient :

$$\dot{\mathbf{q}} = \mathbf{J}^t.[\mathbf{J}.\mathbf{J}^t + \mu^2.Id]^{-1}.\dot{\mathbf{x}} = \mathbf{J}_{DLS}.\dot{\mathbf{x}}$$

$$\mathbf{J}_{DLS} = \mathbf{V}.\Sigma_{DLS}.\mathbf{U}^t$$

Intérêt : robuste proche des singularités.

Que vaut l'erreur  $\dot{\mathbf{e}} = \dot{\mathbf{x}} - \mathbf{J}.\dot{\mathbf{q}}$  ?

## Interprétation géométrique de l'effet de $\mu$

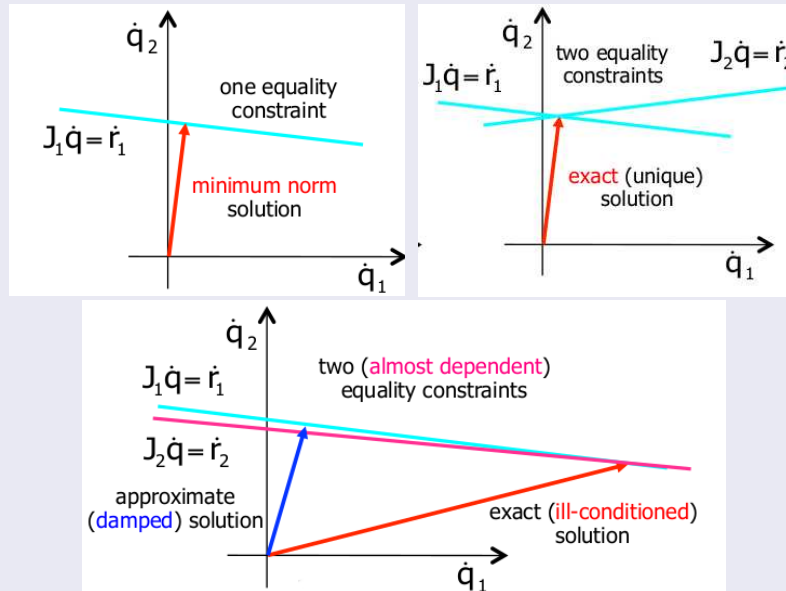


FIGURE – Alessandro De Luca- Sapienza

# Comparaison Pseudoinverse et DLS

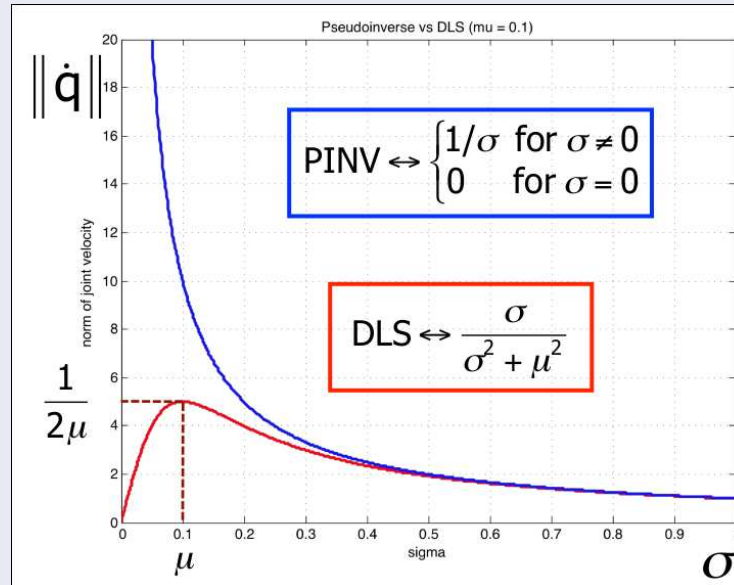


FIGURE – Alessandro De Luca- Sapienza

## Conclusion sur les méthodes basées sur la Jacobienne

- Pas de garantie d'éviter les singularités
- Pas de garantie de cycle dans l'espace des  $\mathbf{q}$

## 2 -Méthodes de résolution basées sur le N(J)

La solution de  $\dot{\mathbf{x}} = J(\mathbf{q}).\dot{\mathbf{q}}$  est

$$\dot{\mathbf{q}} = J^+.\dot{\mathbf{x}} + (Id - J^+.J).\dot{\mathbf{q}}_0$$

. On utilise les propriétés du projecteur  $(Id - J^+.J)$  :

- symétrique
- idempotent
- $(Id - J^+.J) = (Id - J^+.J)^+$
- $J^+.\dot{\mathbf{x}}$  est  $\perp$  à  $(Id - J^+.J).\dot{\mathbf{q}}_0$

Choix de  $\dot{\mathbf{q}}_0$  qui satisfait un critère (on veut privilégier  $\dot{\mathbf{q}}_0$ )

$$\min_{\dot{\mathbf{q}}} \frac{1}{2}(\dot{\mathbf{q}} - \dot{\mathbf{q}}_0)^T.W.(\dot{\mathbf{q}} - \dot{\mathbf{q}}_0)$$

sous la contrainte

$$\dot{\mathbf{x}}(t) - J(\mathbf{q}).\dot{\mathbf{q}}(t) = 0$$

$$\dot{\mathbf{q}} = \dot{\mathbf{q}}_0 - W^{-1}.J^T.(J.W^{-1}.J^T)^{-1}.(J.\dot{\mathbf{q}}_0 - \dot{\mathbf{x}})$$

.

$$\dot{\mathbf{q}} = J_W^\#.\dot{\mathbf{x}} + (I - J_W^\#.J).\dot{\mathbf{q}}_0$$

### 3 - Méthode de résolution par tâche augmentée

On va ajouter une tâche auxiliaire à la tâche principale  $\dot{\mathbf{x}} = J(\mathbf{q}).\dot{\mathbf{q}}$

$\implies$  on ajoute des contraintes

$$\mathbf{X}_c = h(\mathbf{q}) \text{ de dimension } s = n - m \implies \dot{\mathbf{X}}_c = \frac{\partial h}{\partial \mathbf{q}}.\dot{\mathbf{q}} = J_h(\mathbf{q}).\dot{\mathbf{q}}$$

Jacobienne augmentée  $J_A$

$$J_A(\mathbf{q}) = \begin{pmatrix} J(\mathbf{q}) \\ J_h(\mathbf{q}) \end{pmatrix} \implies \begin{pmatrix} \dot{\mathbf{x}} \\ \dot{\mathbf{x}}_c \end{pmatrix} = J_A(\mathbf{q}).\dot{\mathbf{q}}$$

$$\text{Si } J_A \text{ est non singulière alors } \dot{\mathbf{q}} = J_A^{-1}(\mathbf{q}).\begin{pmatrix} \dot{\mathbf{x}} \\ \dot{\mathbf{x}}_c \end{pmatrix}$$

Questions :

- Comment choisir les  $(n - m)$  contraintes ?
- $J_A$  est-il inversible si  $\text{rang}(J(\mathbf{q})) = m$  ?
- Unicité de la solution ?

# Tâches prioritaires

Idée : définir une tâche primaire  $\mathbf{X}_1$  et utiliser les d.d.l restant pour réaliser une tâche  $\mathbf{X}_2$  de priorité moindre.

La tâche  $\mathbf{X}_1$  sera réalisée en priorité et la tâche  $\mathbf{X}_2$  sera réalisée au mieux (pas de garantie).

$$\mathbf{x}_1 = F_1(\mathbf{q}) \text{ de dimension } m_1 \implies \dot{\mathbf{x}}_1 = J_1(\mathbf{q}).\dot{\mathbf{q}}$$

$$\mathbf{x}_2 = F_2(\mathbf{q}) \text{ de dimension } m_2 \implies \dot{\mathbf{x}}_2 = J_2(\mathbf{q}).\dot{\mathbf{q}}$$

Tâche prioritaire 1 : la solution est  $\dot{\mathbf{q}} = J_1^+.\dot{\mathbf{x}}_1 + (Id - J_1^+.J_1).\dot{\mathbf{q}}_1$

Pour la tâche 2, on a maintenant :  $\dot{\mathbf{x}}_2 = J_2(\mathbf{q}).\dot{\mathbf{q}}$

$$\dot{\mathbf{x}}_2 = J_2(\mathbf{q}).(J_1^+.\dot{\mathbf{x}}_1 + (Id - J_1^+.J_1).\dot{\mathbf{q}}_1)$$

$$\dot{\mathbf{x}}_2 - J_2.J_1^+.\dot{\mathbf{x}}_1 = J_2.(Id - J_1^+.J_1).\dot{\mathbf{q}}_1$$

# Méthodes de résolution

Equation du type :  $\dot{\epsilon} = \beta(\mathbf{q}).\dot{\mathbf{q}}_1 \implies$  calculer  $\dot{\mathbf{q}}_1$  qui satisfait au mieux cette nouvelle tâche.

On calcule à nouveau une pseudo-inverse de  $\beta$ .

$$\dot{\mathbf{q}}_1 = \beta^+ . (\dot{\mathbf{x}}_2 - J_2 . J_1^+ . \dot{\mathbf{x}}_1) + (Id - \beta^+ . \beta) . \dot{\mathbf{q}}_2$$

On obtient donc :

$$\dot{\mathbf{q}} = J_1^+ . \dot{\mathbf{x}}_1 + (Id - J_1^+ . J_1) . (\beta^+ . (\dot{\mathbf{x}}_2 - J_2 . J_1^+ . \dot{\mathbf{x}}_1) + (Id - \beta^+ . \beta) . \dot{\mathbf{q}}_2)$$

Propriété (projecteur symétrique/idempotent) :

$$(Id - J_1^+ . J_1) . \beta^+ = (Id - J_1^+ . J_1) . (J_2 . (Id - J_1^+ . J_1))^+ = (J_2 . (Id - J_1^+ . J_1))^+$$

Comment choisir  $\dot{\mathbf{q}}_2$  ?



# Projection dans les noyaux des tâches

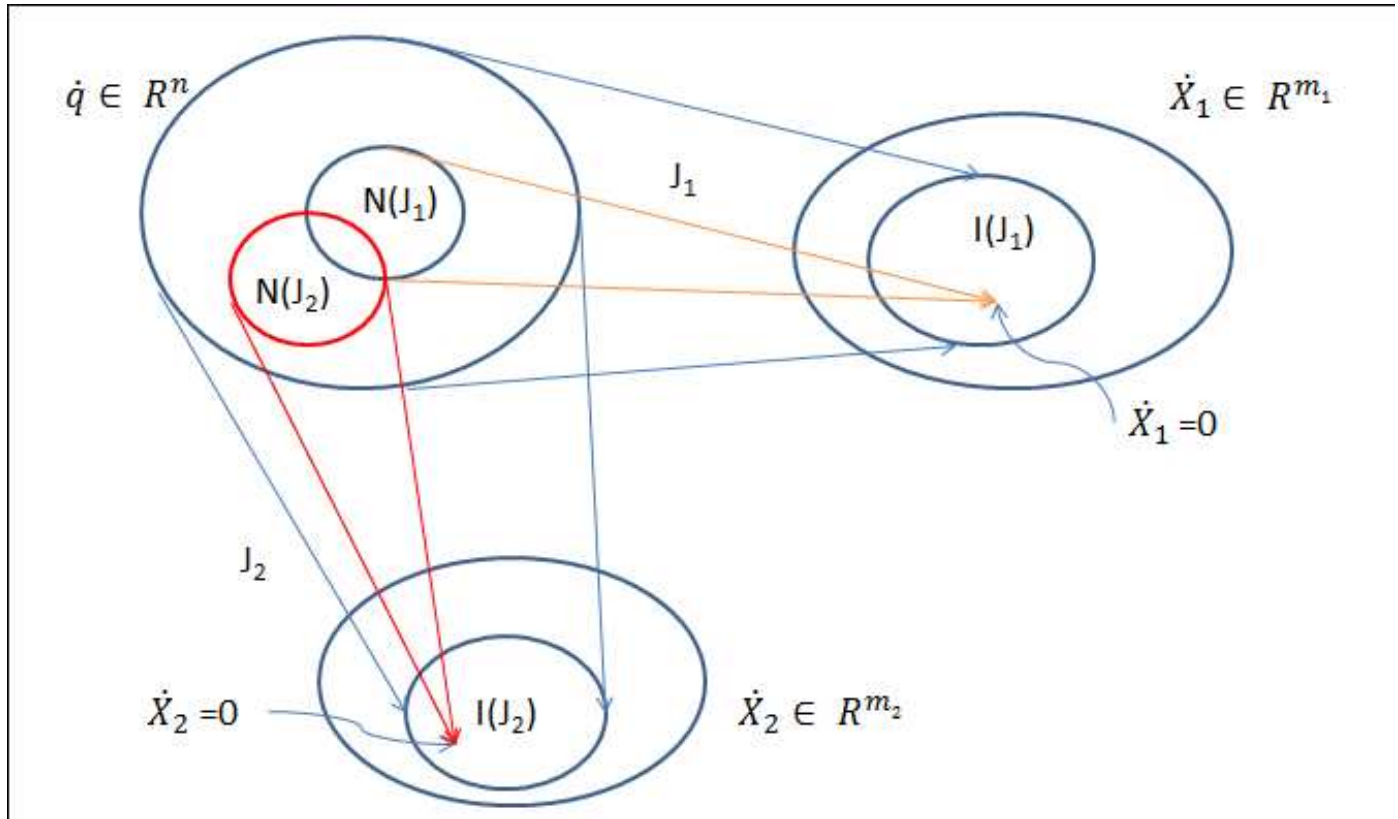


FIGURE – Illustration des projections dans les noyaux des tâches (Nakamura)

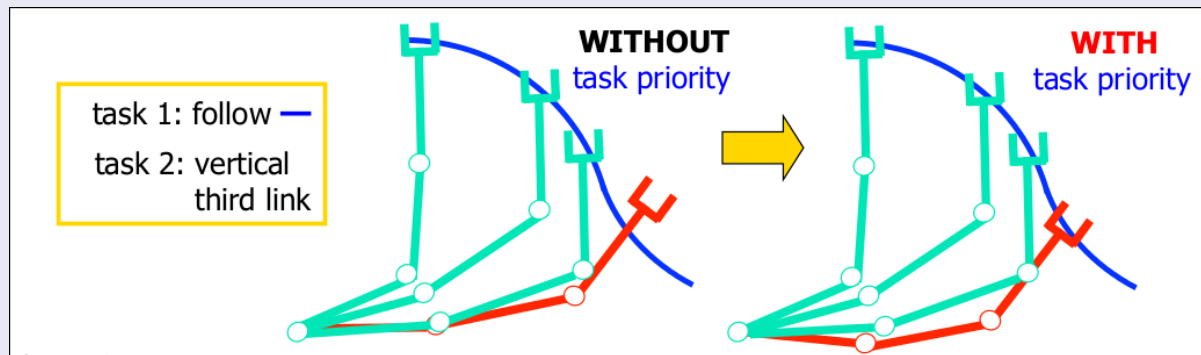


FIGURE – Exemple de priorité avec 2 tâches (source : Alessandro De Luca)

# Robots séries redondants

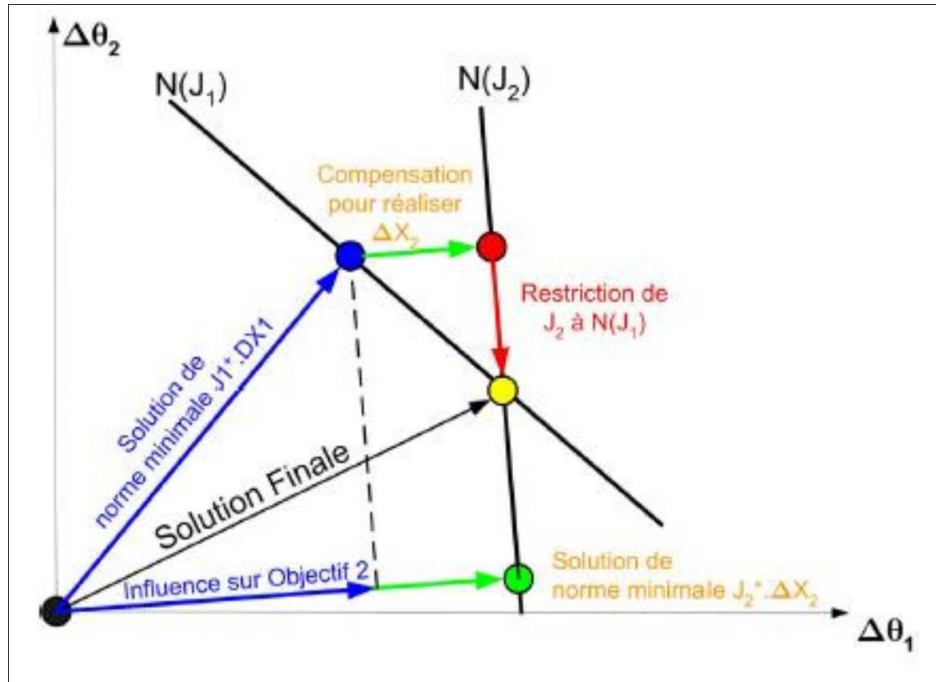


FIGURE – Illustration de la projection (source : R. Boulic)

Signification ?

# Équations de la dynamique : application en robotique

Etablir les équations liant les couples et forces exercées par les actionneurs ( $\tau$ ) avec les actions dues aux déplacement des axes ( $\ddot{q}_i$ ) et aussi les actions dues à l'interaction avec l'environnement  $F_e$ .

## Modèle dynamique (inverse)

$$\tau = fct(q, \dot{q}, \ddot{q}, F_e)$$

## Modèle dynamique direct

$$\ddot{q} = fct(q, \dot{q}, \tau, F_e)$$

Applications :

- Simulation
- Commande de systèmes robotique (ou SMA)
- Dimensionnement des actionneurs, identification des inerties,...

## Efforts statiques

Puissance instantanée qui est transmise par l'outil (lié à l'O.T) sur l'environnement  
= puissance instantanée fournie par les actionneurs.

L'application du principe des travaux virtuels donne :

$$\tau = J(q)^t F_e$$

avec  $F$  = vecteur  $6 \times 1$  des efforts (forces et couples) agissant sur l'outil.

- Équations d'Euler-Lagrange
- Equations de Newton-Euler

## Équation dynamique d'un robot

$$A(\mathbf{q})\ddot{\mathbf{q}} + C(\mathbf{q}, \dot{\mathbf{q}}).\dot{\mathbf{q}} + G(\mathbf{q}) + F_v.\dot{\mathbf{q}} - F_S.sgn(\dot{\mathbf{q}}) = \tau - (\mathbf{q})^T . \mathbf{f}_e$$

$$A(\mathbf{q})\ddot{\mathbf{q}} + B(\mathbf{q}, \dot{\mathbf{q}}) = \tau - J(\mathbf{q})^T . \mathbf{f}_e$$

# Equations d'Euler-Lagrange

$$\frac{d}{dt} \frac{\partial L}{\partial \dot{q}_i} - \frac{\partial L}{\partial q_i} = \tau_i$$

avec :

$L = K - V$  : Lagrangien du système,

$K$  : énergie cinétique,

$V$  : énergie potentielle,

$q_i$  : coordonnée généralisée de la  $i^{\text{ème}}$  liaison,

$\tau_i$  : force généralisée appliquée à la  $i^{\text{ème}}$  liaison (soit un couple si la liaison est rotoïde, soit une force si la liaison est prismatique).

## Energie cinétique

$$K = \sum_{i=1}^n \frac{1}{2} m_i v_i^t v_i + \frac{1}{2} \omega_i^t I_i \omega_i$$

avec :

$I_i$  : matrice d'inertie du corps  $i$ ,

$m_i$  : masse du corps  $i$ ,

$v_i$  : vitesse linéaire du centre de gravité du corps  $i$ ,

$\omega_i$  : vitesse angulaire du corps  $i$ .

## Energie cinétique dans le cas d'un robot

$$v_i = J_{v_i(q)} \dot{q} \qquad \omega_i = J_{\omega_i(q)} \dot{q}$$

- $J_{v_i(q)}$  : matrice jacobienne reliant la variation de vitesse du centre de gravité du corps  $i$  aux vitesses articulaires
- $J_{\omega_i(q)}$  : matrice jacobienne reliant la variation de vitesse de rotation du corps  $i$  aux vitesses articulaires.

## Matrice d'inertie du robot

$$K = \frac{1}{2} \dot{q}^t \sum_{i=1}^n (m_i J_{v_i(q)}^t J_{v_i(q)} + J_{\omega_i(q)}^t I_i J_{\omega_i(q)}) \dot{q}$$

$$K = \frac{1}{2} \dot{q}^t A(q) \dot{q} = \frac{1}{2} \sum_{i,j}^n a_{ij}(q) \dot{q}_i \dot{q}_j$$

La matrice d'inertie  $A(q)$  est une matrice symétrique définie positive de dimension  $n \times n$  qui est fonction de la configuration  $q$  du robot.

La matrice d'inertie  $A(q)$  varie en fonction de  $q$ .

## Energie potentielle

Origine énergie potentielle : gravité

$$V = g^t \sum_{i=1}^n O_{gi(0)} m_i$$

$g$  : vecteur de gravité exprimé dans le repère de base  $R_0$

$O_{gi(0)}$  : coordonnées du centre de gravité du corps  $i$  dans  $R_0$



## Lagrangien dans le cas d'un robot

L'énergie potentielle ne dépend que de la configuration  $q$  du robot, le Lagrangien s'écrit :

$$L = K - V = \frac{1}{2} \sum_{i,j}^n a_{ij}(q) \dot{q}_i \dot{q}_j - V(q)$$

## Equations d'Euler-Lagrange dans le cas d'un robot

$$\frac{\partial L}{\partial \dot{q}_k} = \sum_j a_{kj}(q) \dot{q}_j$$

$$\frac{d}{dt} \frac{\partial L}{\partial \dot{q}_k} = \sum_j a_{kj}(q) \ddot{q}_j + \sum_j \frac{d}{dt} a_{kj}(q) \dot{q}_j = \sum_j a_{kj}(q) \ddot{q}_j + \sum_{i,j} \frac{\partial a_{kj}}{\partial q_i} \dot{q}_i \dot{q}_j$$

et

$$\frac{\partial L}{\partial q_k} = \frac{1}{2} \sum_{i,j} \frac{\partial a_{ij}}{\partial q_k} \dot{q}_i \dot{q}_j - \frac{\partial V}{\partial q_k}$$

Les équations d'Euler-Lagrange pour chaque liaison  $k$  ( $k = 1, \dots, n$ ) deviennent :

$$\sum_j a_{kj}(q) \ddot{q}_j + \sum_{ij} \left( \frac{\partial a_{kj}}{\partial q_i} - \frac{1}{2} \frac{\partial a_{ij}}{\partial q_k} \right) \dot{q}_i \dot{q}_j + \frac{\partial V}{\partial q_k} = \tau_k$$

Pour simplifier l'écriture on utilise le fait que :

$$\sum_{ij} \frac{\partial a_{kj}}{\partial q_i} \dot{q}_i \dot{q}_j = \frac{1}{2} \sum_{ij} \left( \frac{\partial a_{kj}}{\partial q_i} + \frac{\partial a_{ki}}{\partial q_j} \right) \dot{q}_i \dot{q}_j$$

On obtient alors :

$$\sum_{ij} \left( \frac{\partial a_{kj}}{\partial q_i} - \frac{1}{2} \frac{\partial a_{ij}}{\partial q_k} \right) \dot{q}_i \dot{q}_j = \sum_{ij} \frac{1}{2} \left( \frac{\partial a_{kj}}{\partial q_i} + \frac{\partial a_{ki}}{\partial q_j} - \frac{\partial a_{ij}}{\partial q_k} \right) \dot{q}_i \dot{q}_j$$

On appelle  $c_{ijk}(q) = \frac{1}{2} \left( \frac{\partial a_{kj}}{\partial q_i} + \frac{\partial a_{ki}}{\partial q_j} - \frac{\partial a_{ij}}{\partial q_k} \right)$  symbole de Christoffel.

$$c_{ijk} = c_{jik}$$

$$\text{On note } \Phi(q)_k = \frac{\partial V}{\partial q_k}$$

Pour chaque liaison  $k$  ( $k = 1, \dots, n$ ) on obtient :

$$\sum_j a_{kj}(q)\ddot{q}_j + \sum_{ij} c_{ijk}(q)\dot{q}_i\dot{q}_j + \Phi(q)_k = \tau_k$$

Ecriture sous forme matricielle :

$$\mathbf{A}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q},\dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}) = \boldsymbol{\tau}$$

$r_{kj}$  le  $(k,j)$ <sup>ème</sup> élément de  $\mathbf{C}$  est défini par  $r_{kj} = \sum_i c_{ijk}(q)\dot{q}_i$

Les termes impliquant un produit  $\dot{q}_i^2$  sont appelés centrifuge

Les termes impliquant un produit  $\dot{q}_i\dot{q}_j$  ( $i \neq j$ ) sont appelés Coriolis

En général les forces non conservatives  $\boldsymbol{\tau}$  ne correspondent pas seulement aux couples des actionneurs. Nous avons :

- $\tau_m$  couples des actionneurs,
- $F_c$  forces de contact
- $\gamma_f(q, \dot{q})$  de frottement

$$\mathbf{A}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q},\dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}) = \tau_m - \gamma_f(q, \dot{q}) + J^t(q).F_c$$



# Génération de mouvements

- Introduction
- Calcul de GM dans l'espace généralisé
  - Loi polynomiale
  - Schéma trapézoïdal
  - Synchronisation des axes
  - Cas de points de passage
- Calcul de GM dans l'espace opérationnel

# Introduction

Les modèles = passage entre les espaces  $Q$  et  $X$

La commande = mouvement des liaisons en fonction d'une consigne à respecter (position, vitesse) pour atteindre  $\underline{X}$  ou  $\underline{Q}$

Problème : comment calculer la consigne ? (tableau)

Trouver la relation entre la trajectoire spatiale qui va relier la trajectoire et son exécution temporelle

Plusieurs types de problème: (tableau)

- \*  $\underline{X}_i$  et  $\underline{X}_f$  (ou  $\underline{Q}_i$  et  $\underline{Q}_f$ ) imposées (trajectoire libre)
- \*  $\underline{X}_i$  et  $\underline{X}_f$  (ou  $\underline{Q}_i$  et  $\underline{Q}_f$ ) imposées + points de passage
- \*  $\underline{X}_i$  et  $\underline{X}_f$  (ou  $\underline{Q}_i$  et  $\underline{Q}_f$ ) + trajectoire  $\underline{X}(u)$  (ou  $\underline{Q}(u)$ ) imposées
- \*  $\underline{X}_i$  et  $\underline{X}_f$  (ou  $\underline{Q}_i$  et  $\underline{Q}_f$ ) +  $\underline{X}(u)$  (ou  $\underline{Q}(u)$ ) +  $u(t)$  imposées (loi de mvt)

# Définitions

Trajectoire= courbe décrite par un objet en mouvement.

Mouvement =déplacement d'un corps par rapport à un point fixe de l'espace et à un moment déterminé.

Faire l'étude du mouvement d'un point mobile consiste à rechercher deux types d'informations: une information sur la trajectoire du point mobile d'une part et une information sur la rapidité avec laquelle cette trajectoire est parcourue.

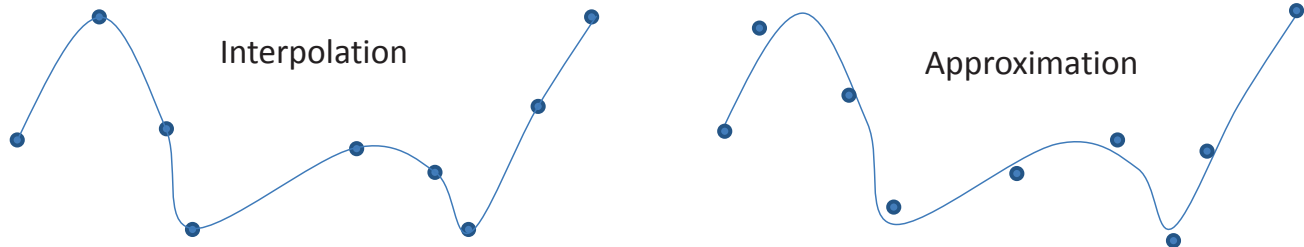
Le problème de la Génération de Mouvement (GM) est de calculer les consignes (position/vitesse/accélération) de référence, en fonction du temps, pour permettre à l'OT du robot de suivre une trajectoire.

# Contraintes

L'utilisateur spécifie généralement un certain nombre de contraintes pour décrire la trajectoire désirée.

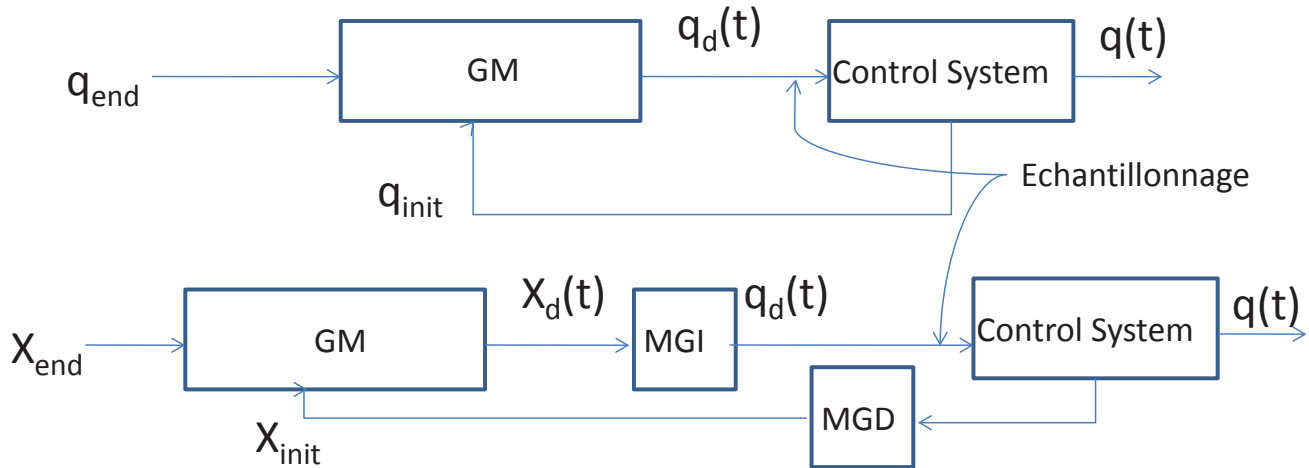
La GM consiste à générer une séquence temporelle des valeurs à atteindre de la trajectoire désirée qui respecte les contraintes.

Contraintes: spatiale/temporelle/saturation( $K_v$ ,  $K_a$ )/continuité/.....





# Schéma général



## Espace articulaire

- Facilité pour « points » de passage
- Pas de problème de singularité
- Peu de calcul
- Impossible de suivre une trajectoire définie dans  $X$

## Espace operationel

- Suivi de forme
- Couteux en temps de calcul
- Problème de discontinuité

Passage entre Q et X: tableau

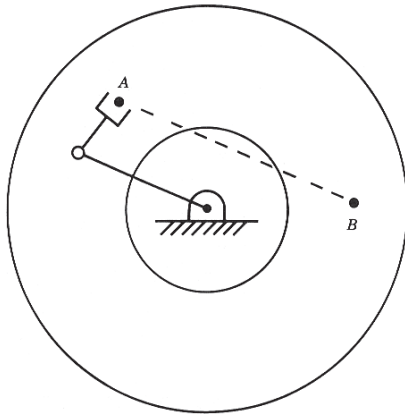
# Problèmes dans l'espace opérationnel

## Point non atteignable:

point de la trajectoire en dehors du volume de travail du robot

Exemple:

Mouvement en ligne droite

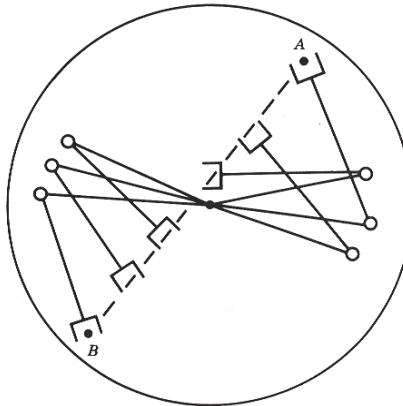


## Singularité: si $q$ s'approche

d'une singularité, alors une articulation peut avoir une vitesse qui tend vers l'infini

Exemple:

Vitesse linéaire constante

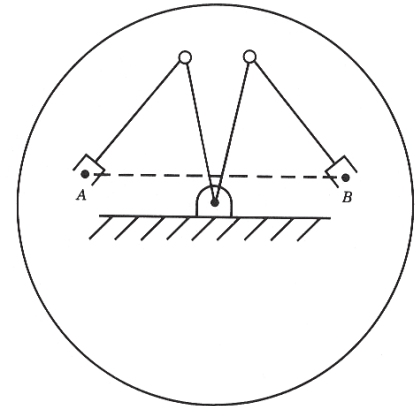


## Début et fin dans #

### type de solutions:

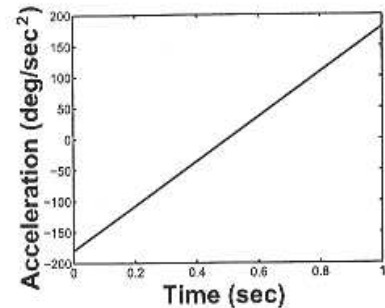
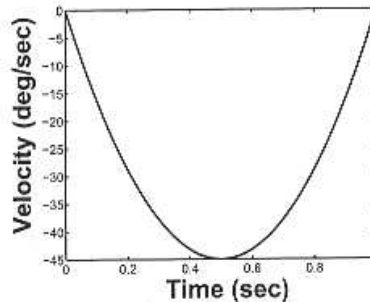
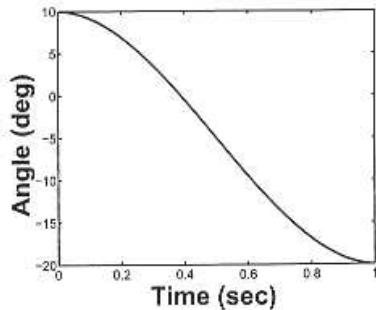
limites articulaires  $\Rightarrow$  # solutions du MGI

Exemple: tous les points atteignables mais dans des configurations de bras différentes



# Loi polynomiale

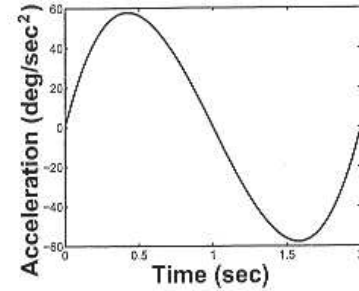
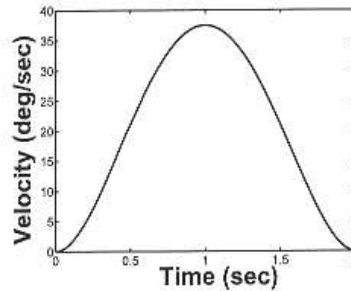
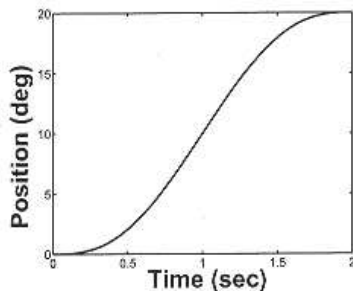
- Etude sur un axe  $q$
- $q_i$ ,  $q_f$ ,  $\dot{q}_i$  et  $\dot{q}_f$  fixées
- $K_v$  et  $K_a$
- GM : calculer les  $a_i$
- Profil de vitesse parabolique
- Profil d'accélération linéaire



- Choix de  $t_f = ?$ 
  - Arbitraire
  - temps minimum  $\Rightarrow K_v$  and  $K_a$
- $K_v \Rightarrow$  maximum pour  $t = t_f/2 \quad \Rightarrow$
- $K_a \Rightarrow$  maximum for  $t = 0$  and  $t = t_f \quad \Rightarrow$
- $t_{fi} = \max(t_{fa}, t_{fv})$
- Pour tous les axes ( $i=1,..,n$ ):  $t_f = \max(t_{fi})$

- Discontinuité en accélération: dépend de la rigidité de la structure mécanique.
- Si nécessaire => 6 contraintes

$$q(t) = a_5.t^5 + a_4.t^4 + a_3.t^3 + a_2.t^2 + a_1.t + a_0$$



# Profil de vitesse trapezoidal

Objectif: minimiser le temps (Bang-bang)

⇒ Accélération/décélération constantes

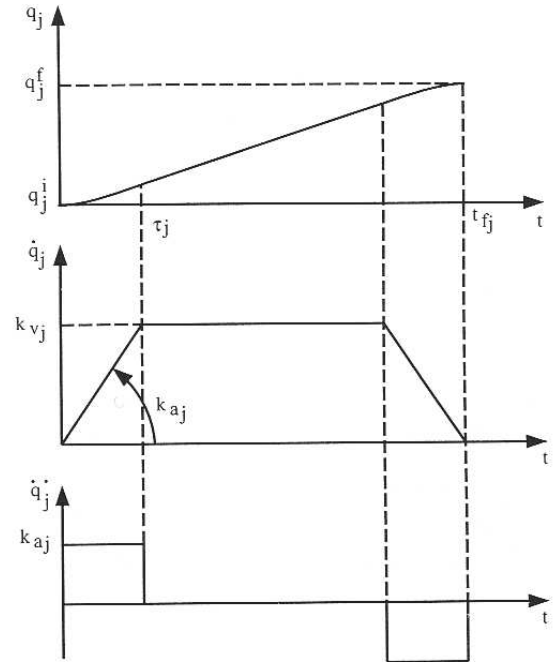
⇒ trajectoire = segment linéaire et deux segments paraboliques

Problème:

- calculer les instants de commutation?
- calculer le temps final?

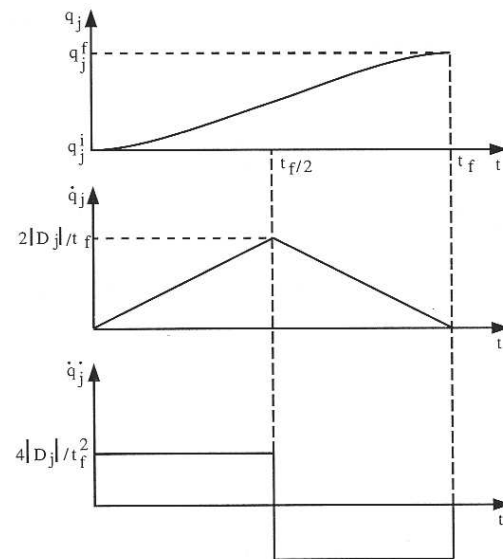
Input =  $q_i$ ,  $q_f$ ,  $K_a$ ,  $K_v$

Output =  $\tau$ ,  $t_f$  and  $\dot{q}(t)$ ,  $\ddot{q}(t)$





- Cas vitesse max <  $K_v$  ?



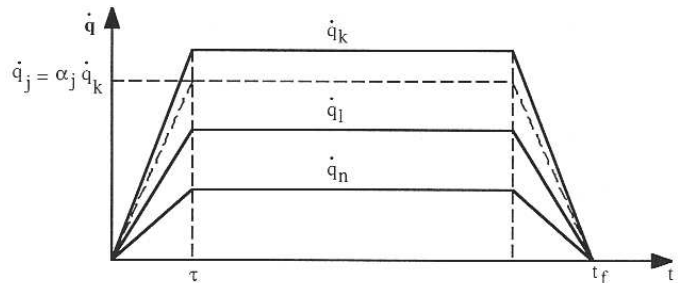
- Discontinuité en accélération => jerk



# 3 - Synchronisation entre les axes

- Tous les axes bougent en même temps
- Même temps de début et de fin
  - Trajectoire identique: A -> B and B -> A
  - minimum nombre d'axes à vitesse maximale
- L'axe qui prend le plus de temps impose le temps final.
- Cas simple: tous les temps d'accélération sont égaux
  - Axes k,  $\Delta q_k \Rightarrow t_f$
  - Pour chaque  $i \neq k$  calculer  $\dot{q}_i(t)$

$$\Rightarrow \dot{q}_i(t) = \frac{\Delta q_i}{(t_f - \tau)}$$



- Cas général : les temps d'accélération sont différents

$$\Rightarrow \tau_i \neq \tau_j$$

- Pour tous les axes  $j$  : vitesse maximum =  $\lambda_j \cdot Kv_j$  avec  $0 \leq \lambda_j \leq 1$   
accélération maximale =  $v_j \cdot Ka_j$  avec  $0 \leq \gamma_j \leq 1$
- Problème: calculer  $\tau$  pour chaque axe  $\Rightarrow t_f$  minimum pour la trajectoire

- Exemple avec 2 axes

$$t_f^1 = \tau_1 + \frac{|\Delta 1|}{Kv_1} = \frac{Kv_1}{Ka_1} + \frac{|\Delta 1|}{Kv_1} \qquad t_f^2 = \tau_2 + \frac{|\Delta 2|}{Kv_2} = \frac{Kv_2}{Ka_2} + \frac{|\Delta 2|}{Kv_2}$$

$$t_f = \max \left( t_f^1, t_f^2 \right)$$

$$\Rightarrow t_f = \frac{\lambda_1 \cdot K v_1}{\gamma_1 \cdot K a_1} + \frac{|\Delta 1|}{\lambda_1 \cdot K v_1} = \frac{\lambda_2 \cdot K v_2}{\gamma_2 \cdot K a_2} + \frac{|\Delta 2|}{\lambda_2 \cdot K v_2} \Rightarrow \left\{ \begin{array}{l} \tau = \frac{\lambda_1 \cdot K v_1}{\gamma_1 \cdot K a_1} = \frac{\lambda_2 \cdot K v_2}{\gamma_2 \cdot K a_2} \\ \lambda_2 = \lambda_1 \cdot \frac{K v_1}{K v_2} \cdot \frac{|\Delta 2|}{|\Delta 1|} \end{array} \right.$$

$$\Rightarrow \gamma_2 = \gamma_1 \cdot \frac{K a_1}{K a_2} \cdot \frac{|\Delta 2|}{|\Delta 1|}$$

$$0 \leq \lambda_j \leq 1, 0 \leq \gamma_j \leq 1 \Rightarrow \left\{ \begin{array}{l} 0 \leq \lambda_1 \leq \frac{K v_2}{K v_1} \cdot \frac{|\Delta 1|}{|\Delta 2|} \\ 0 \leq \gamma_1 \leq \frac{K a_2}{K a_1} \cdot \frac{|\Delta 1|}{|\Delta 2|} \end{array} \right.$$

$$\lambda_1 = \min \left( 1, \frac{K v_2}{K v_1} \cdot \frac{|\Delta 1|}{|\Delta 2|} \right)$$

$$\gamma_1 = \min \left( 1, \frac{K a_2}{K a_1} \cdot \frac{|\Delta 1|}{|\Delta 2|} \right)$$

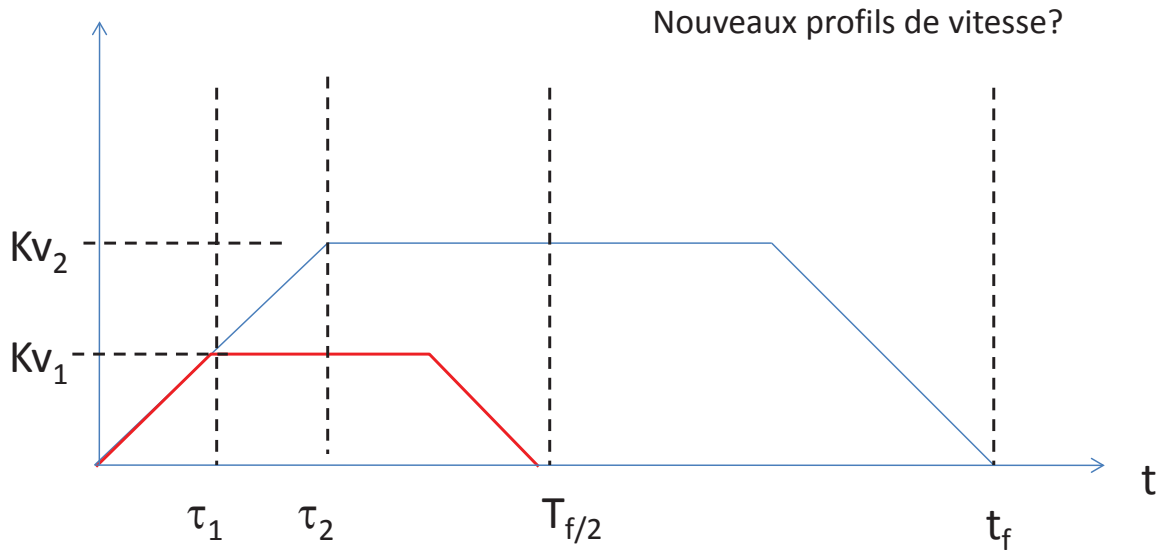


$$\tau = \frac{\lambda_1}{\gamma_1} \cdot \frac{K v_1}{K a_1}$$



Généralisation  
pour les n axes

Exemple:  $Ka_1=Ka_2$ ,  $2.Kv_1= Kv_2$ ,  $\Delta 2 = 4.\Delta 1$



4 – Points de passage